

Personal System/2®
Hardware Interface
Technical Reference

00F9809



Personal System/2®
Hardware Interface
Technical Reference

First Edition (May 1988)

The following paragraph does not apply to the United Kingdom or any country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.

It is possible that this publication may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

THE PUBLICATION OF THE INFORMATION CONTAINED HEREIN IS NOT INTENDED TO AND DOES NOT CONVEY ANY RIGHTS OR LICENSES, EXPRESS OR IMPLIED, UNDER ANY IBM PATENTS, COPYRIGHTS, TRADEMARKS, MASK WORKS OR ANY OTHER INTELLECTUAL PROPERTY RIGHTS.

Requests for copies of this publication and for technical information about IBM products should be made to your IBM Authorized Dealer or your IBM Marketing Representative.

Personal System/2 is a registered trademark of the International Business Machines Corporation.

© Copyright International Business Machines Corporation 1988. All rights reserved.
No part of this work may be reproduced or distributed in any form or by any means without prior permission in writing from the IBM Corporation.

Preface

This technical reference provides hardware and software interface information specifically for IBM Personal System/2 products. This manual contains both a PS/2™ family overview and system-specific information, and should be used with the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference.*

This manual is divided into the following:

"Micro Channel™ Architecture" describes the Micro Channel signals, timing, electrical characteristics, level-sensitive interrupts, and multidevice arbitration.

"Programmable Option Select" describes adapter setup, system configuration utilities, and adapter description files.

"Micro Channel Adapter Design" provides information such as adapter dimensions, power requirements, and design guidelines.

"Microprocessors and Instruction Sets" describes the various processors used in the Personal System/2 family. Also included is a quick reference for microprocessor assembly instruction sets.

"System Board I/O Controllers" describes the input and output interfaces of the system board controllers.

"Keyboards (101- and 102-Key)" has layouts of the 101- and 102-key keyboards. Keyboard scan-code sets and keyboard specifications are also provided.

"Characters and Keystrokes" supplies the decimal and hexadecimal values for ASCII characters.

"Power Supply" provides the electrical input/output specifications and describes the theory of operations.

Micro Channel and PS/2 are trademarks of the International Business Machines Corporation.

"Compatibility" provides hardware and software information to take into consideration, and provides suggestions to aid you in developing your programs.

System-specific information concerning hardware implementation and performance is also included.

Note: Information added to the system-specific area of this manual may have new information about subjects covered in other parts of this manual. Refer to the system-specific area for information that could affect your hardware or software development decisions.

A Bibliography is also provided.

Technical Reference Library

The technical reference library is intended for those who develop hardware and software products for IBM PS/2 systems and who understand computer architecture and programming concepts.

The technical reference library for Personal System/2 products that incorporate the Micro Channel architecture consists of the following:

- *IBM Personal System/2 Hardware Interface Technical Reference*: provides information to support the functions and architecture common to multiple models of the PS/2 family. In this manual, Type 1 refers to the initial hardware design level. Subsequent levels are designated as Type 2, Type 3, and so on.
- System-specific technical references: provide information concerning hardware implementation and performance for a given model.
- *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference*: provides BIOS and Advanced BIOS interface information.
- Option and Adapter Technical References: provide hardware and programming information about individual PS/2 options and adapters.

Suggested Reading:

- *BASIC for the IBM Personal Computer*
- IBM *Disk Operating System (DOS)*
- IBM *Operating System/2*™
- *Macro Assembler for the IBM Personal Computer*
- *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference.*

Additional publications relating to the information contained in this manual are listed in the Bibliography.

Warning: In this technical reference, the term "Reserved" is used to describe certain signals, bits, and registers. Use of reserved areas can cause compatibility problems, loss of data, or permanent damage to the hardware.

When modifying a register, the state of the reserved bits must be preserved. When possible, read the register first and change only the bits required.

Operating System/2 is a trademark of the International Business Machines Corporation.

Notes:

Setup

Reference Diskette 1
Option Diskette 2
System Configuration Utilities 2
  Automatic Configuration Utility 3
  Change Configuration Utility 5
  View Configuration Utility 5
  Backup and Restore Configuration Utilities 5
  Copy an Option Diskette Utility 6
Adapter Description Files 7
  Format 7
  Syntax 7
  Example 17
Adapter Description Program 20
  Syntax for ADPs 21
  System Memory ADPs 21
  Nonsystem Memory ADPs 22
  Program Call 23
  ADP Functions 28
    Configure the Adapter 28
    Configure All Adapters 29
    Deallocate Resources for an Adapter 29
    Deallocate Resources for All Adapters 29
    Deduce Adapter Configuration 29
    Disable Block of System Memory 30
    Remap System Memory to Above 16MB 31
    Remap System Memory to Below 16MB 31
    Configure System Memory on an Adapter 31
    Configure System Memory on All Adapters 31
    Deallocate System Memory on an Adapter 32
    Deallocate System Memory on All Adapters 32
    Deduce System Memory Configuration on an Adapter 32
    Check Validity of Choice 32
  Resource Query Functions 34
    Request Memory Block 35
    Check Memory Block 36
    Deallocate Memory Block 37
    Enable an Adapter 37
    Disable an Adapter 37
    Enable Memory by Prompt 38
    Get Number of Slots 39
    Get Slot Information 39
  Error Codes 40

© Copyright IBM Corp. 1989, 1990

ADP Example 41
Initialization Programs 49

Index 51

Figures

1. Files with the ADF Extension 1
2. Syntax Symbol Key 7
3. Adapter Description File Syntax 8
4. ADP Parameter Block 24
5. ADP Functions 28
6. Check Validity of ADF Choice Function. 33
7. Resource Query Function Calls. 34
8. Resource Query Parameter Block 35
9. Request Memory Block Function 36
10. Check Memory Block Function 36
11. Deallocate Memory Block Function 37
12. Enable an Adapter Function 37
13. Disable an Adapter Function 38
14. Enable Memory by Prompt Function 38
15. Get Number of Slots Function 39
16. Get Slot Information Function 39
17. Error Codes 40

© Copyright IBM Corp. 1989, 1990

Notes:

The integrated I/O functions of the system board use POS information during the setup procedure. The bit assignments and functions may vary from system to system (refer to the system board setup information in the system-specific technical reference for the system you are dealing with). The setup routine (Automatic Configuration) obtains adapter information from adapter description files and programs.

Reference Diskette

Each system has a Reference Diskette containing the System Configuration and diagnostic utilities. The diagnostic utilities perform customer and service testing on the system, its components, and adapters made by IBM. The System Configuration utilities identify the installed hardware and interpret the system resources (I/O ports, memory, interrupt levels, and arbitration levels) for each device.

All the files on the Reference Diskette must be copied onto a backup copy, and the backup copy is used for configuring the system and running diagnostic routines. As options are added, the files needed for configuration and problem determination of the new options are merged onto this backup copy.

The following table shows the naming convention for files used by adapters during configuration. The CardID is the POS adapter ID in hex.

File Name	Description	
@CardID.ADF	Adapter Description File	Required
CCardID.ADF	Adapter Description Program	Optional
ICardID.ADF	Initialization Program	Optional

Figure 1. Files with the ADF Extension

Option Diskette

The Option Diskette supplied with the adapter should contain the adapter description file (ADF) and, if needed by the adapter, the adapter description program (ADP) or the initialization program.

Before a new adapter is installed, its ADF and other associated files are merged onto a backup copy of the Reference Diskette. These files are merged by selecting the Copy an Option Diskette program from the Main Menu and following the instructions on the screen.

Additional files may be required on the Option Diskette to support keywords or functions used in the ADF or ADP. For more information on the content of an Option Diskette, contact Developer Assistance at 1-800-IBM-7763 (this number is for developers who are registered with IBM).

Note: Developers of adapters and their customers are granted permission to use, but not to modify, the Set Configuration program to support their adapters.

System Configuration Utilities

The System Configuration utilities are contained within the Set Configuration program. This program enables the user to configure the system in one of two ways:

- Automatically after a configuration error is displayed
- Manually by selecting Set Configuration from the Main Menu.

The Set Configuration program uses information contained in the adapter description file (ADF) to track and allocate system resources. Each ADF describes the resources that can be allocated to a specific adapter and the POS setting used to indicate those resource assignments. When more than one device is configured to the same resource and that resource cannot be shared, only one of the conflicting devices is enabled.

Each adapter contains a 16-bit adapter ID and four 8-bit POS registers. Adapter IDs and the POS information are stored in CMOS or nonvolatile RAM by the Set Configuration program. The Set Configuration program determines the system type and handles differences between systems such as the nonvolatile RAM storage and the number of adapter slots available.

Automatic Configuration Utility

Automatic Configuration can be run after a configuration error has occurred or by selecting Run Automatic Configuration from the Set Configuration menu. Each time the system is powered on, POST compares the configuration of the system to the configuration indicated by nonvolatile RAM. If differences between the two are detected, an error is displayed and logged in system RAM. If the Reference Diskette is installed, an expanded message is displayed that further explains the POST error.

If a configuration error is caused by a loss o f configuration data, Automatic Configuration is run immediately after the POST error message is displayed. If the error is caused by a change to the configuration, the user may be given a choice to either run Automatic Configuration, restore configuration, or continue to the Main Menu. If the user continues with the Main Menu, the changed areas of the system are configured and nonvolatile RAM is updated when Set Configuration is selected from the Main Menu.

Depending on the cause of the error, Automatic Configuration either reconfigures the entire system or configures only the areas of the system that have been changed since the last time a configuration was performed. The following POST errors cause the system to be completely reconfigured:

- 161 (battery failure)
- 162 (CRC error in nonvolatile RAM).
- 169 (processor complex error).
- 173 (CRC error in nonvolatile RAM).

The following POST errors cause only the areas of the system that have changed to be reconfigured:

- 162 (system configuration error not caused by a CRC error)
- 164 (memory configuration)
- 165 (adapter configuration).
- 174 (adapter configuration).

An adapter is considered previously configured when the POS data stored in nonvolatile RAM matches a set of POS settings in the appropriate ADF. If an ADF for an installed adapter cannot be found, the user is prompted to insert the Option Diskette. If system configuration is continued without its ADF being found, the adapter is not configured and is not enabled.

During Automatic Configuration, devices are configured to the first nonconflicting values as defined in the ADF. Adapters are configured in the order of the channel position in which they are installed. The system board is configured first, followed by each adapter slot starting with slot 1. Interrupt levels are a shared resource and are not considered to be in conflict if more than one device is assigned the same level. If a named item in the ADF contains a list of interrupt levels to choose from, the Automatic Configuration program assigns the level that is assigned to the fewest devices at that time.

Automatic Configuration will backtrack to previously configured adapters to resolve conflicts in nonsystem memory resources. If Automatic Configuration cannot resolve a conflict, the conflict must be resolved by choosing a nonconflicting resource option through the Change Configuration utility. Any adapter having a resource conflict that cannot be resolved by the Set Configuration program is disabled; the program sets bit 0 of POS Register 2 (hex 0102) to 0 in nonvolatile RAM.

Change Configuration Utility

The Change Configuration utility allows the user to change the current configuration settings to resolve unusual conflicts or to customize configuration.

Resource conflicts are indicated by an asterisk (*) next to the conflicting items and also by the "* Conflicts" string in the upper right corner of the Change Configuration window. Conflicts with fixed resources have the asterisk (*) to the left of the slot number of the adapter. Adapters with conflicts are disabled; the program sets bit 0 of POS Register 2 (hex 0102) to 0.

The user can scroll or page through the various screens. Changes are made by rotating field value names through a set of choices using the F5 (Previous) and F6 (Next) keys. Changes are not saved in nonvolatile RAM until the F10 (Save) key is pressed. Help text is provided for each item by pressing the F1 (Help) key.

Note: Automatic Configuration will restore all customized settings to the first setting listed in the ADF that does not conflict.

View Configuration Utility

The View Configuration utility is provided to view the configuration. This is the Change Configuration utility with the change capabilities disabled.

Backup and Restore Configuration Utilities

The Backup Configuration utility provides a means to back up the configuration data to a file on the Reference Diskette. If the battery fails or the battery is changed, the user can use the Restore utility to restore the configuration.

Note: A copy of the Reference Diskette that is not write protected is needed for this backup and restore process.

Copy an Option Diskette Utility

The Copy an Option Diskette utility is a separate program from the Set Configuration program and is accessible through the Main Menu. This utility is used to merge the files with the extension of ADF[1] from an Option Diskette onto a backup copy of the Reference Diskette. The files found on this diskette are compared to the files on the backup copy of the Reference Diskette. If the file does not exist on the Reference Diskette, it is copied. If the file already exists on the Reference Diskette, the dates of the two files are compared, and the file is copied if it has a date later than, or the same as, the corresponding file on the backup diskette.

The Option Diskette must be a DOS formatted diskette.

1 The following files are also copied by this utility; however, the file names are reserved for use by IBM: *.DGS, *.PEP, COMMAND.COM, DIAGS.COM, CMD.COM, SC.EXE, UPDATE.EXE, and SYSCONF.

Adapter Description Files

Adapter description files (ADFs) provide POS information and system resource information for system configuration. The ADFs also provide text for the System Configuration utilities, help screens, and prompts. This section provides guidelines for developing the adapter description files.

Format

- File names: @CardID.ADF (high byte of the adapter ID first).
- Type of file: ASCII text.
- Not case sensitive: Key words can be lowercase, uppercase, or mixed. The case is preserved within the user interface text strings.
- Blanks, tabs, new lines: These are considered as white space and ignored, except when in text strings for the Change Configuration user interface.
- Comments: Lines beginning with semicolons are comments and are ignored.

Syntax

The following table shows the meaning of special symbols used in the adapter description file syntax. Values shown as hexadecimal can be specified as decimal or hexadecimal using the Intel format. Values shown as binary must be specified as binary.

Symbol	Meaning
{ }	an optional item
{ }*	0,1,2,... items allowed
{ }+	1,2,3,... items allowed
x | y	either x or y allowed
x{n}	n x's required
[0-9]+	one or more decimal digits

Figure 2. Syntax Symbol Key

```
adf_file => card_id card_name nbytes {fixed_resources}
            {system_memory} {initial_program} {lock_enable}
            {address_width} {programmed_io} {items}*
```

This statement defines the contents of an ADF. The following definitions describe each parameter in detail.

```
card_id => AdapterId number
```

Each ADF must contain the card_id parameter. The character string AdapterId is a keyword and must be present. The Set Configuration program looks for this ID, which matches the ID used in the filename.
Example: `AdapterId 0DEFFh`

```
card_name => AdapterName string
```

Each ADF must contain the card_name parameter. The character string AdapterName is a keyword and must be present.
The string following the AdapterName keyword is displayed as the adapter name in the Change Configuration and View Configuration screens. The length of the AdapterName string is limited to (74 - (length of 'SlotX - ')) characters.
(The length for US English is 66 characters)
Example: `AdapterName "IBM Multiprotocol Communications Adapter"`

```
nbytes => NumBytes number
```

Each ADF must contain the nbytes parameter; it defines the number of POS bytes used by the adapter. The character string NumBytes is a keyword. The number of POS bytes used includes all bytes from POS [0] to the last POS byte used. (If POS [3] is the only POS byte defined, NumBytes should be set to 4.)
Example: `NumBytes 4`

```
fixed_resources => FixedResources pos_setting resource_setting
```

The fixed_resources parameter is optional, and it defines those resources required by an adapter and the corresponding POS data. The character string FixedResources is a keyword and must be present if the adapter has a specific system resource that it must use, such as a hard-wired I/O address or interrupt level.
Note:
Using fixed resources is not recommended because conflicts cannot be resolved automatically and can result in the adapter being disabled.
Example: `FixedResources  POS[1]=XXXX01XXb int 3`

Figure 3 (Part 1 of 9). Adapter Description File Syntax

```
system_memory => SysMem granularity {address_size}
```

The system_memory parameter is required for adapters that use an ADP to configure their system memory. The character string SysMem is the keyword.
The granularity field is required and specifies the smallest block of memory on the adapter that can be disabled. This block size must be divisible by 4KB. The value used in this field specifies the number of 1KB blocks.
The address_size field specifies the number of address bits supported by the adapter, 24 or 32 bits. The default size is 24.
Note:
This keyword requires an ADP.
Example: `SysMem 1024 32`

```
initial_program => InitProg attribute
```

The initial_program parameter is required for adapters that have an initialization program associated with them. The character string InitProg is the keyword. The initialization programs are loaded onto the fixed disk and executed before the operating system is booted from the fixed disk.
The attribute is a number from 0 to 127 and determines the order in which the initialization programs are run. The lower numbers are run first. Values outside the range from 30 to 80 are reserved.
Example: `InitProg 30`
(See "Initialization Programs" on page 49 for more information.)

```
lock_enable => Lock
```

The lock_enable parameter is optional. The character string Lock is the keyword, and its presence enables the Lock function.
An adapter that wishes to communicate with other masters through a memory semaphore protocol (an indivisible test and set operation, such as the Exchange instruction) must indicate this to the Set Configuration program by the Lock keyword.
Example: `Lock`

```
items => {{named_item | input_item} | {begin_device}}*
```

The items parameter is optional and can be any combination of the named_item and input_item parameters, or it can be any number of begin_device parameters.

Figure 3 (Part 2 of 9). Adapter Description File Syntax

```
named_item => NamedItem prompt {named_choice}+  help
```

The named_item parameter is optional, and it defines a field that provides a choice of one or more resources. Each Choice field sets specified POS bits to a unique setting used to identify resources assigned to the adapter. The character string NamedItem is a keyword and must be present if the adapter can be configured to use different resources. The adapter determines the resources to which it is configured by how the POS bytes are set. When a NamedItem is defined, it must be accompanied by a prompt, at least one named_choice, and a help field.

To provide flexibility during configuration, each named_item should contain more than one choice.

Note:

Each ADF is limited to 10 named_items per adapter.

Example:

```
NamedItem
Prompt "Communications Port"
choice "SDLC_1"    pos[0]=XXX1000Xb  io 0380h-038ch  int 3 4
choice "SDLC_2"    pos[0]=XXX1001Xb  io 03a0h-03ach  int 3 4
choice "BISYNC_1" pos[0]=XXX1100Xb  io 0380h-0389h  int 3 4
choice "BISYNC_2" pos[0]=XXX1101Xb  io 03a0h-03a9h  int 3 4
choice "SERIAL_1" pos[0]=XXX0000Xb  io 03f8h-03ffh  int 4
choice "SERIAL_2" pos[0]=XXX0001Xb  io 02f8h-02ffh  int 3
choice "SERIAL_3" pos[0]=XXX0010Xb  io 3220h-3227h  int 3
choice "SERIAL_4" pos[0]=XXX0011Xb  io 3228h-322fh  int 3
choice "SERIAL_5" pos[0]=XXX0100Xb  io 4220h-4227h  int 3
choice "SERIAL_6" pos[0]=XXX0101Xb  io 4228h-422fh  int 3
choice "SERIAL_7" pos[0]=XXX0110Xb  io 5220h-5227h  int 3
choice "SERIAL_8" pos[0]=XXX0111Xb  io 5228h-522fh  int 3
Help
"This port can be assigned as the primary (SDLC1) or
secondary (SDLC2) sdlc, the primary (BISYNC1) or secondary
(BISYNC2) bisync, or as a serial port (Serial 1 through Serial
8).  Use the F5=Previous and the F6=Next keys to change this
assignment in the 'Change configuration' window.  Conflicting
assignments are marked with an asterisk and must be changed
to use the adapter."
```

Figure 3 (Part 3 of 9). Adapter Description File Syntax

```
input_item => Input prompt pos_byte_setting range help
```

The input_item parameter is optional; it defines a field that accepts operator-entered data to initialize a single POS byte, or to define specific bits within that byte. The character string Input is the keyword. The operator can enter up to a three-digit positive integer. The bits to be altered by the input are defined using 0 or 1. The X specifies bits that remain unchanged by the input. The operator input is aligned with the least-significant bit of the pos_byte_setting field not defined as X. The two values are then ORed before the mask is applied and the result is stored. The range defines the limit of the values acceptable as user input. The first number listed in the range will be used as the input for the adapter during automatic configuration.

Example:
```
Input prompt "Enter value"pos[0]=XXXX1111b 0 - 15
      help "Enter a value from 0 - 15."
```

```
prompt => Prompt string
```

The prompt is required when a named_item or input_item is defined. The prompt is used to define a title for a NamedItem or an Input field. The character string Prompt is a keyword and must be present when a NamedItem or Input parameter is present. The string following the Prompt keyword appears after the adapter name in the Change Configuration and View Configuration screens.

Following the prompt string is a field that can be toggled in the Change Configuration screen if two or more named_choices are defined. The length of the prompt cannot exceed 38 characters.

Example: (See the example for named_item).

```
named_choice => Choice {valid_check} choice_name pos_setting
      resource_setting
```

At least one named_choice is required when a NamedItem is defined. The character string Choice is a keyword and must be present when a NamedItem is present. The named_choice field is specified after the prompt. Each Choice must contain the choice_name, which describes the current choice. The choice_name string must not exceed 28 characters. Each Choice must define a pos_setting for at least one POS byte that uniquely identifies the resource_setting defined in the named_choice.

Example:
```
Choice "IO 3F80" pos[1]=00111111b pos[2]=10000000b
      io 3f80-3f87
```

Figure 3 (Part 4 of 9). Adapter Description File Syntax

```
valid_check => Vcheck
```

The valid_check parameter invokes the ADP to check the specified choice for validity. The character string Vcheck is the keyword. The ADP ensures that the choice is valid for the system configuration. If the Set Configuration program is unable to load the ADP, the choice is treated as invalid. If all the choices within a named_item are invalid, the Set Configuration program deletes the named_item from the list of items for that adapter.

```
help => Help string
```

The help field is a string of text used to give the user assistance at a prompt. This text is displayed in the Change Configuration and View Configuration screens when the cursor is at the associated prompt and the F1 key is pressed. The character string Help is a keyword and must be present when the NamedItem or Input parameter is present. The string following the Help keyword is the text that describes the prompt. The length of the help string is limited to 1000 characters.
Example: (See the example of named_item.)

```
pos_setting => {pos_byte_setting}+
```

The pos_setting parameter must contain at least one pos_byte_setting. See the definition of pos_byte_setting.

```
pos_byte_setting => POS[number]=pos_bit{8}b
```

Each pos_byte_setting field defines one POS byte.
The character string POS is a keyword and precedes each POS byte defined. The number of the POS byte is specified in brackets and refers to the POS bytes as shown below:

- number = 0, POS byte 0 at port 102h
- number = 1, POS byte 1 at port 103h
- number = 2, POS byte 2 at port 104h
- number = 3, POS byte 3 at port 105h

The closing bracket must be followed by an equal sign, and then a bit definition of the POS byte (See pos_bit for more). The bit definition must define all 8 bits of the byte. Bit 0 of pos[0] and bit 7 of pos[3] should always be defined as X.
Example: `pos[0]=XXX1001Xb`

```
pos_bit => x | X | 0 | 1
```

A pos_bit can be defined as a mask bit (x or X), a clear bit (0), or a set bit (1).
Example: `pos[0]=XXX1001Xb`

Figure 3 (Part 5 of 9). Adapter Description File Syntax

```
resource_setting => {ioblock_list} {interrupt_list} {arb_list}
                    {memaddr_list} {exec_adp}
```

The resource_setting parameter defines a list of system resources. The resources can be fixed resources required by the adapter, or they can be resources the adapter uses when configured to a specific choice in a named_item. The resources can consist of the following:

- Range of I/O addresses (limited to 16 ranges).
- List of interrupt levels (limited to 16).
- List of arbitration levels (limited to 16).
- Range of memory addresses (limited to 2 ranges).
- An ADP call for nonsystem memory allocation.

Example: (See the specific resource definition).

```
ioblock_list => IO {range}+
```

The ioblock_list parameter is a list of one or more ranges of I/O addresses. The character string IO is a keyword and must be present in the ioblock_list.
Example: `IO 4226h-4227h`

```
interrupt_list => Int {number}+
```

The interrupt_list parameter is a list of one or more interrupt levels that are used concurrently. The character string Int is a keyword and must be present in the interrupt_list.
Example: `Int 3 4`

```
arb_list => Arb {shared_arb}{number}+
```

The arb_list parameter is a list of one or more arbitration levels that are used concurrently. The character string Arb is a keyword and must be present in the arb_list.
Example: `Arb 1`

```
shared_arb => SharedArb
```

The character string SharedArb is the keyword and indicates that, in a controlled environment, the arbitration level in the named_item parameter can be shared with the arbitration level of other devices that support sharing of arbitration levels.

Example: `Arb ShareArb 1`

Figure 3 (Part 6 of 9). Adapter Description File Syntax

```
memaddr_list => Mem {range}+
```

The memaddr_list parameter is a list of one or more ranges of RAM or ROM addresses. The character string Mem is a keyword and must be present in the memaddr_list. This keyword is used to allocate the memory address space in the hex 000C0000 through hex 000DFFFF and the nonsystem memory above 1MB.
Include the address for each Choice in the choice_name field so that the address is also displayed during Change Configuration.
Note:
Because of the limited space available, each adapter should limit the address space used in segments 0C000 and 0D000 to 16KB.
Example: `Mem 0CC000h - 0CDFFFh 0E00000h - 0EFFFFFh`

```
exec_adp => Exec
```

The exec_adp parameter is required for adapters containing nonsystem memory that can be initialized by an ADP. The character string Exec is the keyword that defines the ADP and should be the last Choice in the list of named_choices.
The Exec keyword indicates to the Set Configuration program that the adapter has an ADP.
Note:
This keyword requires an ADP.
See the example on page 22.

```
address_width => Address 24|32
```

The address_width field specifies the number of address bits supported by the adapter, 24 or 32 bits. The character string Address is the keyword and the default size is 24.
Example: `Address 32`

```
programmed_I/O => PIO
```

The programmed_I/O parameter is used to specify whether an adapter includes address bits 24 through 31 in its address decode. If this parameter is not specified, the adapter does respond to these address bits.
Note:
A device is address limited if the ADF does not have the PIO parameter and the slot or the adapter is address limited.
Example: `PIO`

Figure 3 (Part 7 of 9). Adapter Description File Syntax

```
begin_device =>
  Begin
  {device} {rs422} {DMA_not_used}
  {named_item |input_item}*
  End
```

The begin_device parameter is optional, and is used to define a device block on an adapter. A device block is a series of items that define POS settings for like devices on an adapter, but are not associated with any other devices that might be on the adapter.

```
device =>  Device device_id {implement_id {revision_level}}
```

The device parameter is optional and defines the device types in ABIOS for the individual devices on an adapter or the system board. The character string Device is the keyword. The device_id parameter is the ABIOS device ID, which is described in the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference*.

The implement_id parameter is the implementation identifier, and the revision_level parameter is the implementation revision level.
The implementation identifier and revision level are optional and default to 0 if not specified; however, the revision level cannot be specified without the implementation identifier.
Example: Device 2 1

```
rs422 => RS422
```

The rs422 parameter is optional, and indicates that the serial device uses a RS422-type interface. The character string RS422 is the keyword.
Example: RS422

```
DMA_not_used => NoDMA
```

The DMA_not_used parameter indicates that arbitration levels for the named_items within the begin_device do not require a DMA channel. The character string NoDMA is the keyword. If this parameter is specified, the Set Configuration program does not indicate that a DMA channel is required when an arbitration level is used. If this parameter is not specified, the Set Configuration program indicates that a DMA channel is required when an arbitration level is used.
Note:
Bus masters should use the NoDMA keyword.

Figure 3 (Part 8 of 9). Adapter Description File Syntax

```
range => number - number

number => [0-9]+ {d} |  [0-9a-f]+ h |  [0-9A-F]+ H

string => " [ascii except for "]+ "
```

The string field is a set of ASCII characters beginning with a double quote (") and ending with a double quote.
Example:

```
"This port can be assigned as:  primary (SDLC1) or
secondary (SDLC2) sdlc port, primary (BISYNC1) or secondary
(BISYNC2) bisync port, or as a serial port (Serial 1 through
Serial 8).  Use the F5=Previous and the F6=Next keys to change
this assignment in the 'Change configuration' window.
Conflicting assignments are marked with an asterisk and must
be changed to use the adapter."
```

Figure 3 (Part 9 of 9). Adapter Description File Syntax

Example

The following is an example of an adapter description file for the IBM Personal System/2 Multiprotocol Communications Adapter/A. The name of the file for this adapter is @DEFF.ADF. An explanation of each numbered item begins on page 18.

```
AdapterId 0DEFFh 1

AdapterName "IBM Multiprotocol Communications Adapter" 2

NumBytes 2 3

NamedItem 4

Prompt "Communications Port"

choice "SDLC_1"   pos[0]=XXX1000Xb  io 0380h-038ch  int 3 4
choice "SDLC_2"   pos[0]=XXX1001Xb  io 03a0h-03ach  int 3 4
choice "BISYNC_1" pos[0]=XXX1100Xb  io 0380h-0389h  int 3 4
choice "BISYNC_2" pos[0]=XXX1101Xb  io 03a0h-03a9h  int 3 4
choice "SERIAL_1" pos[0]=XXX0000Xb  io 03f8h-03ffh  int 4
choice "SERIAL_2" pos[0]=XXX0001Xb  io 02f8h-02ffh  int 3
choice "SERIAL_3" pos[0]=XXX0010Xb  io 3220h-3227h  int 3
choice "SERIAL_4" pos[0]=XXX0011Xb  io 3228h-322fh  int 3
choice "SERIAL_5" pos[0]=XXX0100Xb  io 4220h-4227h  int 3
choice "SERIAL_6" pos[0]=XXX0101Xb  io 4228h-422fh  int 3
choice "SERIAL_7" pos[0]=XXX0110Xb  io 5220h-5227h  int 3
choice "SERIAL_8" pos[0]=XXX0111Xb  io 5228h-522fh  int 3

Help

"This port can be assigned as the primary (SDLC1) or
secondary (SDLC2) sdlc port, the primary (BISYNC1) or secondary
(BISYNC2) bisync port, or as a serial port (Serial 1 through
Serial 8).  Use the F5=Previous and the F6=Next keys to
change this assignment.  Conflicting assignments are marked
with an asterisk and must be changed."
```

```
NamedItem 5

Prompt "Arbitration Level for SDLC"

choice "Level_1"   pos[1]=XXXX0001b  arb  1
choice "Level_0"   pos[1]=XXXX0000b  arb  0
choice "Level_2"   pos[1]=XXXX0010b  arb  2
choice "Level_3"   pos[1]=XXXX0011b  arb  3
choice "Level_4"   pos[1]=XXXX0100b  arb  4
choice "Level_5"   pos[1]=XXXX0101b  arb  5
choice "Level_6"   pos[1]=XXXX0110b  arb  6
choice "Level_7"   pos[1]=XXXX0111b  arb  7
choice "Level_8"   pos[1]=XXXX1000b  arb  8
choice "Level_9"   pos[1]=XXXX1001b  arb  9
choice "Level_10"  pos[1]=XXXX1010b  arb 10
choice "Level_11"  pos[1]=XXXX1011b  arb 11
choice "Level_12"  pos[1]=XXXX1100b  arb 12
choice "Level_13"  pos[1]=XXXX1101b  arb 13
choice "Level_14"  pos[1]=XXXX1110b  arb 14

Help

"This assignment need only be changed if it is in conflict
with another assignment.  Conflicting assignments are marked
with an asterisk.  Use the F5=Previous and the F6=Next keys
to change arbitration level assignments.  Using arbitration
levels, this adapter accesses memory directly without
burdening the computer's main microprocessor.  An
arbitration level of 0 has the highest priority, and
increasing levels have corresponding decreased priority"
```

1 The card_id for this adapter is hex 0DEFF. This is an ASCII representation of the ID generated by the adapter. The high byte is followed by the low byte. The card_id is required for all ADFs.

2 The card_name is "IBM Multiprotocol Communications Adapter." The card_name is required for all ADFs.

3 The nbytes in this file is 2, which indicates the adapter uses two POS bytes located at hex 0102 and 0103.

4 This is the first named_item for the adapter. The title of the field is "Communications Port." The user can toggle between the 12 named_choices. Each named_choice has a unique pos_setting assigned to it in bit locations 1 through 4 of POS byte hex 0102 (pos [0]). Also shown is a resource_setting that corresponds to the

pos_setting of the named_choice. The resources allocated in this named_item are I/O addresses and interrupt levels. A help string for this named_item is provided below the last named_choice.

5 This is the second named_item for the adapter. The title of the field is "Arbitration Level for SDLC." The user can toggle between the 15 named_choices. Each named_choice has a unique pos_setting assigned to it in bit locations 0 through 3 of POS byte hex 0103 (pos [1]). Also shown is a resource_setting that corresponds to the pos_setting of the named_choice. The resources allocated in this named_item are arbitration levels. A help string for this named_item is provided below the last named_choice.

Adapter Description Program

| The adapter description program (ADP) is program used to configure
| memory space on an adapter that cannot be handled by normal ADF
| syntax. Memory adapters that require initialization of more than their
| POS values, however, should use an initialization program, not an
| ADP. The ADP is called when the item selected in the ADF contains
| the appropriate keyword.

Two basic types of memory can be assigned addresses within the memory address space: system memory and nonsystem memory.

System memory is memory that is managed and allocated by the primary operating system. It remains assigned to or fixed in the 4GB physical address space, and its contents can be accessed or modified only by an independent master (bus master, system master, or DMA controller).

Nonsystem memory is memory that is not managed or allocated by the primary operating system. It is made up of memory-mapped I/O devices, memory on an adapter that can be directly modified by the adapter, or memory that can be relocated within the address space, such as bank switched and EMS (expanded memory specifications).

The ADP initializes its POS data in the POS array and updates the parameter block. It must be in EXE program format, should not exceed 10KB, and start its code at offset hex 100. The filename for the ADP is CCardID.ADF (CardID is the adapter ID in hex).

Notes:

1. Memory address space is the only system resource that can be allocated by an ADP.
2. The Set Configuration program does not include support for memory adapters that have translation tables.
3. ADPs should not output directly to the display or prompt the user for input; unpredictable results will occur.
4. To ensure system integrity, ADPs must not write directly to locations in nonvolatile RAM.

Syntax for ADPs

There are three keywords in the ADF syntax that define an ADP: SysMem, Exec, and Vcheck. SysMem is used to assign address space to system memory, and Exec is used to assign address space to nonsystem memory.

System Memory ADPs

The SysMem keyword in the ADF indicates that the adapter contains system memory and that an ADP is used to configure the adapter's POS data and the amount of its system memory.

Adapters should not split their memory between the 1MB to 16MB range and the 16MB to 4GB range unless the memory is contiguous across the 16MB address.

System memory is the first resource allocated by the Set Configuration program. The system memory ADPs are run before other resources are allocated and in the following priority:

1. System board memory
2. Memory address size (24-bit address followed by 32-bit)
3. Memory granularity (larger block size first)
4. Slot number starting with connector 1.

If the ADP is not found by the Set Configuration program, the user is prompted to insert the appropriate diskette into drive A. If the ADP still is not found, the adapter is disabled.

Nonsystem Memory ADPs

An ADP can be used by adapters that want to map nonsystem memory into the address space above 1MB. The memory on these adapters does not get included as part of the top of memory value stored in CMOS RAM. This memory must be mapped starting at the highest addressable location and counting down.

The Exec keyword, included within a NamedItem in the ADF, indicates that the adapter has an ADP. It should be the last Choice of a NamedItem list. The ADP is run at the time that Choice is selected. If the ADP cannot be found by the Set Configuration program, the user is prompted to insert the appropriate diskette into drive A. If the ADP still cannot be found, the adapter will be disabled if that Choice is selected. When the Exec keyword is used, POS bits that are set by the ADP are indicated by X's. The following is an example of how the Exec keyword is used.

```
NamedItem Prompt "Adapter RAM"
     Choice "C000"
          pos[2]=XXXX0000b
          pos[1]=11000000b  mem 0c0000h-0c7fffh

     Choice "C800"
          pos[2]=XXXX0000b
          pos[1]=11001000b  mem 0c8000h-0cffffh

     Choice "D000"
          pos[2]=XXXX0000b
          pos[1]=11010000b  mem 0d0000h-0d7fffh

     Choice "D800"
          pos[2]=XXXX0000b
          pos[1]=11011000b  mem 0d8000h-0dffffh

     Choice "Extended Memory"
          pos[2]=XXXXXXXXb
          pos[1]=XXXXXXXXb  Exec

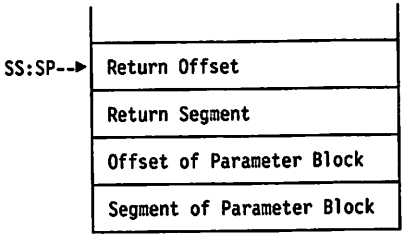
     Help
          "Select a memory address space."
```

Program Call

When the Set Configuration program calls the ADP, the address of the ADP parameter block is passed on the stack. This parameter block contains pointers to data structures that are used by the ADP.

The Set Configuration program checks the parameter block before and on return from the ADP; however, the ADP must maintain the integrity of these data structures during its execution.

When the ADP is given control, the stack is set up as follows:



The ADP must preserve the contents of the DS, DI, SI, and BP registers as well as the stack and stack pointers. The ADP ends using a Far Return with the stack set up as shown.

The following figure shows the format of the parameter block.

00H	ADP Function	Slot
02H	Offset to System Memory (< 16MB)	
04H	Segment to System Memory (< 16MB)	
06H	Offset to System Board Memory (< 16MB)	
08H	Segment to System Board Memory(< 16MB)	
0AH	Offset to POS Array	
0CH	Segment to POS Array	
0EH	Offset to POS Mask	
10H	Segment to POS Mask	
12H	Offset to Adapter ID List	
14H	Segment to Adapter ID List	
16H	Offset to ADP Function Call Entry	
18H	Segment to ADP Function Call Entry	
1AH	Offset to Resource Query Parameters	
1CH	Segment to Resource Query Parameters	
1EH	Offset to System Memory (16MB to 4GB)	
20H	Segment to System Memory (16MB to 4GB)	
22H	Offset of System Board Memory (16MB to 4GB)	
24H	Segment of System Board Memory (16MB to 4GB)	

Figure 4. ADP Parameter Block

The following is a description of the parameter block values and the data structures that are pointed to.

ADP Function: This byte specifies the operation that the ADP will perform. The functions are listed on page 28. The ADP uses this byte to return the results of the operation, successful (0) or unsuccessful (not 0). The ADP returns a hex 01 in this byte if the operation was not successfully completed, and returns a hex FF if the function is not supported.

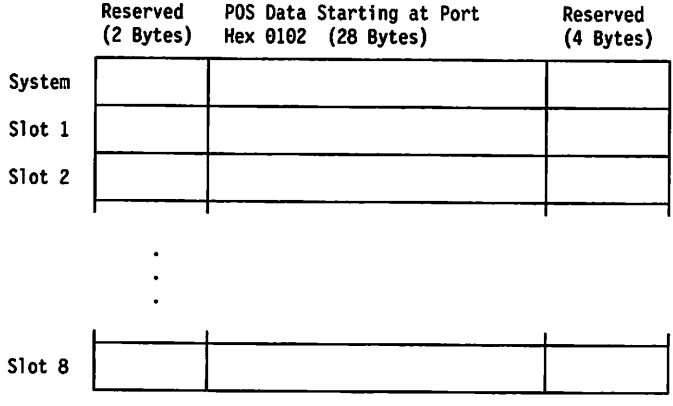
Slot: This is the slot number of the adapter to be configured. The value is not required for all ADP functions.

System Memory Below 16MB: This is a pointer to the current count of all system memory below 16MB. The count is a doubleword value and is the number of 1KB blocks of memory below 16MB already allocated as system memory. (This count includes the count in System Board Memory Below 16MB.) This value is adjusted by ADPs as they map their system memory into the system address space.

The value is initially set to 0 before any configuring is started. An ADP must not increase this value to the point that memory gets mapped beyond the 16MB address.

System Board Memory Below 16MB: This is a pointer to the count of system memory on the system board below 16MB. The count is a doubleword value and is the number of 1KB blocks of system board memory below 16MB. The memory from 640KB to 1MB is included in this count. This value is set by the system description program and must not be adjusted by any other program.

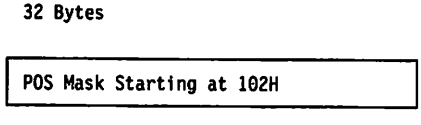
POS Array: This is a pointer to a table of the POS data for the system board and the I/O slots. The ADP should initialize only those locations that correspond to its adapters, and only the four POS data bytes from hex 0102 to 0105. The following is the format of the table and each entry.



POS Mask: This is a pointer to the 32-byte field containing the POS mask for adapter POS information. The POS mask data is initialized by an ADP, and is applied to POS data for all slots with that adapter ID.

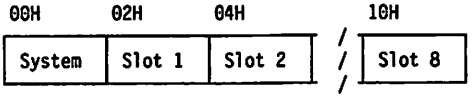
The mask bits are set to 1 for all POS bit positions that are initialized by the ADP. The mask bits are set to 0 for those POS bits that are not used, or that are initialized by statements in the ADF.

The following is the format of the POS mask data pointed to by this field.



The POS mask must be set every time the ADP is called.

Adapter ID List: This is a pointer to a list containing the system and adapter IDs that are currently installed in the system. The adapter IDs are stored as words at the following offsets.



ADP Function Call Entry: This pointer is the program entry point for calls by the ADP to the Set Configuration program when using the Resource Query function calls (see "Resource Query Functions" on page 34). The ADP does a Far Call when calling the Set Configuration program.

Resource Query Parameters: This pointer is the address for the Resource Query parameter block that the ADP uses to pass parameters when calling a Resource Query function. The address for this parameter block is allocated by the Set Configuration program and should not be changed by the ADP.

System Memory Between 16MB and 4GB: This is a pointer to the current count of all system memory between 16MB and 4GB. The count is a doubleword value and is the number of 1KB blocks from 16MB to below 4GB already allocated as system memory. This value is adjusted by ADPs as they map their system memory into the system address space.

The value is initially set to 0 before any configuring is started. An ADP must not increase this value to the point that system memory gets mapped to the 4GB address (must be less than 4GB).

System Board Memory Between 16MB and 4GB: This is a pointer to the count of system memory on the system board between 16MB and 4GB. The count is a doubleword value and is the number of 1KB blocks of system memory on the system board from 16MB to 4GB. This value is set by the system description program and should not be adjusted by other description programs.

ADP Functions

When control is passed to an ADP, a function code is also passed. This function code indicates the operation to be performed by the ADP. The ADPs do not need to support all functions. If a function code is passed to an ADP that is not supported, the ADP should return a hex FF in the Function byte.

The following functions are supported by the Set Configuration program.

Function (Hex)	Description
01	Configure the adapter in the given slot.
02	Configure all adapters corresponding to the ADP.
03	Deallocate resources for the adapter in the given slot.
04	Deallocate resources for all adapters corresponding to the ADP.
05	Deduce the configuration of the adapter in the given slot.
06	Disable a block of system memory.
07	Remap the system memory to above the 16MB address.
08	Remap the system memory to below the 16MB address.
09	Configure the system memory on the adapter in the given slot.
0A	Configure the system memory on all adapters corresponding to the ADP.
0B	Deallocate the system memory on the adapter in the given slot.
0C	Deallocate the system memory on all adapters corresponding to the ADP.
0D	Deduce the configuration of the system memory on the adapter in the given slot.
80	Check the validity of an ADF choice

Figure 5. ADP Functions

Configure the Adapter

This function is indicated by 01 in the Function byte. The ADP configures the adapter in the slot specified, and updates the appropriate system memory and POS array values in the parameter table before returning to the Set Configuration program.

All ADPs defined by the SysMem or Exec keywords must support this function.

Configure All Adapters

This function is indicated by 02 in the Function byte. All adapters associated with that ADP are configured. If more than one adapter is being configured during this function, they are configured in the ascending order of their slot number. The ADP updates the appropriate system memory and POS array values in the parameter table before returning to the Set Configuration program.

All ADPs defined by the SysMem keyword must support this function.

Deallocate Resources for an Adapter

This function is indicated by 03 in the Function byte. The resources allocated to the slot specified are deallocated. An ADP can deallocate only those resources it has allocated. The ADP updates the appropriate system memory and POS array values in the parameter table before returning to the Set Configuration program.

All ADPs defined by the SysMem or Exec keywords must support this function.

Note: The Set Configuration program can deallocate memory space, previously allocated by an ADP, without calling the ADP.

Deallocate Resources for All Adapters

This function is indicated by 04 in the Function byte. The ADP deallocates all resources for all its adapters. The ADP can affect only those resources and adapters that it changed during allocation. The ADP updates the appropriate system memory and POS array values in the parameter table before returning to the Set Configuration program.

All ADPs defined by the SysMem or Exec keywords must support this function.

Deduce Adapter Configuration

This function is indicated by 05 in the Function byte. The ADP interprets the data in the POS array for the slot specified and determines what memory space is allocated to the adapter.

If the ADP was defined by the Exec keyword, it uses the Request Memory Block function to request the address space again. If the address space is not allocated, the ADP leaves the POS data unchanged and returns with an error in the Function byte.

| If the ADP was defined by the SysMem keyword, the current top of
| memory is checked to determine if this adapter was previously
| assigned the next address space. If not, memory on another adapter
| may have been deallocated, and the ADP should reconfigure its
| adapter to the current top of memory. The ADP then updates the
| current count of system memory in the parameter block before
| returning.

| When the ADP detects a difference between the actual memory on the
| adapter and its configuration, the ADP calls the Enable Memory
| Prompt function and handles memory according to the user response.

| All ADPs defined by the SysMem or Exec keywords must support this
| function.

| If the Set Configuration program is deducing the configuration of an
| adapter and a match cannot be found between the data stored in
| CMOS/NVRAM and any of the ADF choice POS definitions, the
| adapter will be configured to the first choice in the list and the
| CMOS/NVRAM will be updated to reflect this. If the adapter is in
| conflict with any other adapters when configured to the first choice,
| the adapter will be disabled. The only indication that the user will
| have of this change is by looking at the Change or View Configuration
| screen. A message indicating that changes were made will also be
| displayed when the user exits the Set Configuration menu.

| Disable Block of System Memory

| This function is indicated by 06 in the Function byte. The ADP
| disables the smallest amount of memory that it possibly can for the
| specified slot. The ADP also decreases the current count of memory
| value affected (either in the below 16MB value or the 16MB to 4GB
| value), and updates the POS array to reflect the disabled memory.

| The ADP can interpret the POS data to determine where its memory
| was previously mapped. If its memory is not currently mapped as the
| top of memory, the ADP does not change any values and indicates a
| successful operation. If its memory is currently mapped as the top of
| memory, but the ADP cannot disable the memory, the ADP returns a
| hex 01 in the ADP Function byte indicating the operation was not
| successfully completed.

| The ADP should support multiple calls to disable memory. For each
| call, an additional block of memory should be disabled until all
| memory on the adapter is disabled. If all memory has been disabled

| and the ADP is called again, the ADP returns an error indicating that
| all memory for the slot specified is already disabled.

| All ADPs defined by the SysMem keyword must support this function.

| **Remap System Memory to Above 16MB**

| This function is indicated by 07 in the Function byte. All memory is
| remapped to the area above 16MB. The ADP updates its POS values
| in the POS array and also updates both top of memory values to
| indicate the changes. If its memory is not currently mapped as the
| top of memory, the ADP does not change any values and indicates a
| successful operation.

| All ADPs defined by the SysMem keyword must support this function;
| adapters that have specified 24-bit addressing will not receive this
| request.

| **Remap System Memory to Below 16MB**

| This function is indicated by 08 in the Function byte. All memory for
| this adapter is remapped to the area below 16MB. The ADP updates
| its POS values in the POS array and also updates both top of memory
| values to reflect the changes. If its memory is not currently mapped
| as the top of memory, the ADP does not change any values and
| indicates a successful operation.

| All ADPs defined by the SysMem keyword must support this function.

| **Configure System Memory on an Adapter**

| This function is indicated by 09 in the Function byte and is used only
| when the ADP is defined with both the SysMem and Exec keywords.
| The ADP configures only the system memory on the adapter in the
| slot specified. The ADP updates the appropriate system memory and
| POS array values in the parameter table before returning to the Set
| Configuration program.

| **Configure System Memory on All Adapters**

| This function is indicated by hex 0A in the Function byte and is used
| only when the ADP is defined with both the SysMem and Exec
| keywords. The ADP configures only the system memory on all
| adapters associated with the ADP. If more than one adapter is being
| configured, they are configured in the ascending order of their slot
| number. The ADP updates the appropriate system memory and POS

| array values in the parameter table before returning to the Set
| Configuration program.

| Deallocate System Memory on an Adapter

| This function is indicated by hex 0B in the Function byte and is used
| only when the ADP is defined with both the SysMem and Exec
| keywords. The system memory previously allocated to the adapter
| by the ADP is deallocated. The ADP updates the appropriate system
| memory and POS array values in the parameter table before
| returning to the Set Configuration program.

| Deallocate System Memory on All Adapters

| This function is indicated by hex 0C in the Function byte and is used
| only when the ADP is defined with both the SysMem and Exec
| keywords. The ADP deallocates system memory that it previously
| allocated for all its adapters. The ADP updates the appropriate
| system memory and POS array values in the parameter table before
| returning to the Set Configuration program.

| Deduce System Memory Configuration on an Adapter

| This function is indicated by hex 0D in the Function byte and is used
| only when the ADP is defined with both the SysMem and Exec
| keywords. The ADP checks the current top of memory to determine if
| the adapter was previously assigned the next address space. If not,
| memory on another adapter may have been deallocated and the ADP
| should reconfigure the starting address of its adapter to the current
| top of memory. The ADP then updates the current count of system
| memory in the parameter table before returning.

| When an ADP detects a difference between the actual memory on the
| adapter and its configuration, the ADP uses the Enable Memory by
| Prompt function and handles memory according to the user response.

| Check Validity of Choice

| This function is indicated by hex 80 in the function byte. It indicates
| whether the choice is valid or not for the current system
| configuration. The parameter block for this function is different than
| those defined previously. The following figure shows the parameter
| block for this function.

00H	Hex 80	Hex 00
02H	Starting Address of the First I/O Address Range (0 if no I/O Address Range)	
04H	First Interrupt Level (0 if no Interrupt Level)	
06H	First Arbitration Level (Hex 0FFFF if no Arbitration Level)	
08H	Starting Address of the First Memory Address Range (0 if no Memory Address Range)	
0CH	Offset of the Adapter IDs List.	
0EH	Segment of the Adapter IDs List.	
10H	Offset of the ADP Function Call Entry Point	
12H	Segment of the ADP Function Call Entry Point	
14H	Offset of the Resource Query Parameters	
16H	Segment of the Resource Query Parameters	

Figure 6. Check Validity of ADF Choice Function.

If the choice is valid, the ADP returns a 0 in the function byte. If the choice is not valid, the named_item is not included in the list of available settings during the configuration process.

All ADPs that use the Vcheck keyword must support this function.

Resource Query Functions

The Resource Query functions allows the ADP to communicate with the Set Configuration program. The ADP can query the memory resource-allocation table and request a block of addresses for an adapter. Before calling, the ADP loads the specific values into the Resource Query parameter block for that function and restores the DS register if it was changed.

A Resource Query function is initiated by making a far call to the program entry point passed to the ADP in the parameter block. Most Resource Query functions are designed to allocate address space for nonsystem memory. ADPs configuring system memory are generally concerned with only the current top of memory value.

Code (Hex)	Description
00	Request Memory Block
01	Check Memory Block
02	De-allocate Memory Block
03	Enable an Adapter
04	Disable an Adapter
05	Enable Memory with a Prompt
06	Get Number of Slots
07	Get Slot Information

Figure 7. Resource Query Function Calls.

The allocation of nonsystem is handled by the Set Configuration program. The ADP simply requests a block of memory within a specified range. The ADP should try to map its memory into the highest block possible.

Note: The size of the block and high end of the range plus 1 must be divisible by 4KB.

The Set Configuration program responds with a return code that indicates whether the request was successful or unsuccessful. If the request is successful, the Set Configuration program has also returned the starting address of the memory block allocated. The ADP then sets the appropriate bits in the POS array.

The Set Configuration program attempts to allocate the highest block of memory within the requested range. If the highest block is not available, the starting address is decreased by the size of the block requested until either a block of memory space is available, or the starting address is below the requested range.

When the ADP returns control to the Set Configuration program, it must indicate in the Function byte whether or not the configuration was successful.

The parameter block for a Resource Query function is defined as follows:

00H	Function	Slot
02H	Start_add_high	
04H	Start_add_low	
06H	End_add_high	
08H	End_add_low	
0AH	Size_high	
0CH	Size_low	

Figure 8. Resource Query Parameter Block

Request Memory Block

This function is used to request a block of memory space for adapters with nonsystem memory. The ADP provides the range where its memory can be mapped and the number of bytes to be mapped.

The ADP can only allocate memory to an adapter with its ID. Once a memory block has been allocated, it remains allocated until that ADP deallocates the space, or until automatic configuration is run.

The following figure shows the values for the parameter passed between the ADP and the Set Configuration program.

```
        Function = 00
        Slot = Slot number of adapter
        Start_addr_high = Beginning of range (High Word)
        Start_addr_low = Beginning of range (Low Word)
        End_addr_high = End of range (High Word)
        End_addr_low = End of range (Low Word)
        Size_high = Requested block size in bytes (High Word)
        Size_low = Requested block size in bytes (Low Word)

    Return:
        Successful:
            Function = 00
            Start_addr_high = Beginning of the block allocated
            Start_addr_low = Beginning of the block allocated
        Unsuccessful:
            Function = Error code (01, 02, 03, 06, 08 or 09)
                    (refer to "Error Codes" on page 40)
            All other values unchanged.
```

Figure 9. Request Memory Block Function

Check Memory Block

This function allows the ADP to get the current memory block assigned to an adapter in a given slot. An ADP can check the allocated memory blocks for any slot.

Because a list of memory blocks can be allocated to an adapter, the ADP must specify which block in the list to use. An ADP can check the entire list by starting at number 0 and checking the next block until an error code is returned.

```
        Function = 01
        Slot = Slot number of adapter
        End_addr_high = Block number

    Return:
        Successful:
            Function = 00
            Start_addr_high = Beginning of block (High Word)
            Start_addr_low = Beginning of block (Low Word)
            Size_high = Allocated block size in bytes (High Word)
            Size_low = Allocated block size in bytes (Low Word)
        Unsuccessful:
            Function = Error code (01, 05 or 09)
                    (refer to "Error Codes" on page 40)
            All other values unchanged.
```

Figure 10. Check Memory Block Function

Deallocate Memory Block

This function allows the ADP to deallocate a block of memory for a given slot. The ADP can deallocate only the memory space assigned to adapters with its ID. The ADP specifies only the beginning address of the memory block.

```
      Function = 02
      Slot = Slot number of adapter
      Start_addr_high = Beginning of memory block (High Word)
      Start_addr_low = Beginning of memory block (Low Word)

Return:
      Successful:
           Function = 00
      Unsuccessful:
           Function = Error code (01, 04 or 09)
                    (refer to "Error Codes" on page 40)
           All other values unchanged.
```

Figure 11. Deallocate Memory Block Function

Enable an Adapter

This function allows the ADP to enable and temporarily configure one of its adapters. Because the data in the POS mask array is written when the adapter is enabled, the ADP must initialize the POS mask before enabling the adapter.

```
      Function = 03
      Slot = Slot number of the adapter

Return:
      Successful:
           Function = 00
      Unsuccessful:
           Function = Error code (01)
                    (refer to "Error Codes" on page 40)
           All other values unchanged.
```

Figure 12. Enable an Adapter Function

Disable an Adapter

This function disables the adapter. Configuration information is not required because only the adapter enable bit is changed.

```
        Function = 04
        Slot = Slot number of the adapter

    Return:
        Successful:
            Function = 00
        Unsuccessful:
            Function = Error code (01)
                    (refer to "Error Codes" on page 40)
            All other values unchanged.
```

Figure 13. Disable an Adapter Function

Enable Memory by Prompt

This function allows an ADP to prompt the user for a decision when the amount of memory on the adapter does not match configuration. When this function is called, the Set Configuration program displays a message and waits for the user to type a Y or N. The message asks the user if the memory should be enabled and the adapter reconfigured. If a Y is typed, the Set Configuration program returns a 01 in the Function byte and the ADP enables memory and reconfigures the adapter. If an N is typed, the Set Configuration program returns a 00 in the Function byte, the ADP disables the changed memory and does not change the configuration.

This function should be used only when the ADP was called with a Deduce function. An example of when this function is used is if memory was added to a configured adapter.

Note: The return code specified in the Function byte for Enable Memory by Prompt function is different from the error code described in "Error Codes" on page 40.

Architecture Note:

If the slot number specified in the parameter block is not valid, the Set Configuration program also returns a 01 in the Function byte.

```
        Function = 05
        Slot = Slot number of the adapter

    Return:
        Enable memory:
            Function = 01
        Leave memory disabled:
            Function = 00
```

Figure 14. Enable Memory by Prompt Function

Get Number of Slots

This function allows the ADP to determine the number of channel connectors in the system, which can be used to determine the size of the POS array, adapter ID list, and other arrays.

```
     Function = 06

Return:
     Function = 00
     Slot = Number of channel connectors in the system
```

Figure 15. Get Number of Slots Function

Get Slot Information

This function requests information about a particular channel connector. The ADP can use this information to determine the address size of the connector and whether or not the connector has a video extension.

```
     Function = 07
     Slot = Slot number to get information about

Return:
     Successful:
          Function = 00
          Slot = address size of slot
          Start_addr_high:
             bit 0 = 1, slot contains video extension
                     0, slot does not contain video extension
             bits 1 - 15 = reserved (set to 0)
     Unsuccessful:
          Function = Error code (01)
                   (refer to "Error Codes" on page 40)
          All other values unchanged.
```

Figure 16. Get Slot Information Function

Error Codes

The following is a list of the error codes used by the ADP and Set Configuration program. The error codes are the same for the ADP functions and the Resource Query functions.

Error Code	Description
01	The slot number specified was not valid.
02	The size specified is not valid.
03	The resource is already allocated to an adapter.
04	Attempted to deallocate resources that were not allocated.
05	A list value is not valid.
06	The memory range specified is not valid.
07	The function passed by the ADP is not valid.
08	Cannot allocate any more memory blocks to this adapter.
09	All memory for this adapter has already been disabled.
FF	Operation is not supported.

Figure 17. Error Codes

ADP Example

The following is an example of an adapter description program that is defined by the Exec keyword.

```
;-------------------------------------------------------------------;
;                                                                   ;
;  Sample ADP defined by the EXEC keyword.                          ;
;                                                                   ;
;  Adapter specifications:                                          ;
;       Adapter has a 1M buffer of nonsystem memory that can be     ;
;       mapped on any 1M boundary in the area from 2G to 4G - 1.    ;
;                                                                   ;
;       Adapter POS registers:                                      ;
;               register 100h = 77h                                 ;
;               register 101h = 77h                                 ;
;               register 102h =                                     ;
;                       bit 0 = adapter enable bit                  ;
;                       bit 1 = nonsystem memory enable             ;
;               register 103h = low byte of high word of assigned   ;
;                               memory block.                       ;
;               register 104h = high byte of high word of assigned  ;
;                               memory block.                       ;
;                       (Low word of memory block always 0).        ;
;                                                                   ;
;-------------------------------------------------------------------;
.286C
.286p

;-------------------------------------------------------------------;
;  Constants:                                                       ;
;-------------------------------------------------------------------;

adapter_id              equ     7777h   ;ID for this adapter
begin_block_h           equ     8000h   ;high word for start of 2G
begin_block_l           equ     0       ;low word for start of 2G
end_block_h             equ     0ffffh  ;high word for end of 4G - 1
end_block_l             equ     0ffffh  ;low word for end of 4G - 1
size_h                  equ     10h     ;high word for size of 1M
size_l                  equ     0       ;low word for size of 1M
mem_enable_mask         equ     2       ;mask for memory enable bit in POS 102h
not_mem_enable          equ     0fdh    ;complement of mem_enable_mask

max_pos_bytes           equ     32      ;number of POS bytes in pos array
pos_array_size          equ     34      ;number of total bytes in pos array

;ADP functions
configure               equ     1       ;configure an adapter
deallocate              equ     3       ;deallocate resources for an adapter
deallocate_all          equ     4       ;deallocate resources for all adapters
deduce                  equ     5       ;deduce the adapter configuration
```

```
;ADP function calls - resource query
request_mem_block       equ     0       ;request a block of nonsystem memory
check_mem_block         equ     1       ;check memory block allocated to adapter
deallocate_mem_block    equ     2       ;de-allocate a block of nonsystem memory

;ADP parameter block
adp_pb          struc
function        db      ?               ;ADP function
slot            db      ?               ;slot being configured
top_mem_os      dw      ?               ;pointer to top of system memory below
top_mem_seg     dw      ?               ;       16M
top_pmem_os     dw      ?               ;pointer to top of planar system memory
top_pmem_seg    dw      ?               ;       below 16M
pos_array_os    dw      ?               ;pointer to POS array
pos_array_seg   dw      ?
pos_mask_os     dw      ?               ;pointer to POS mask
pos_mask_seg    dw      ?
id_list_os      dw      ?               ;pointer to list of adapter IDs
id_list_seg     dw      ?
res_query_os    dw      ?               ;resource query entry point
res_query_seg   dw      ?
adp_pb_os       dw      ?               ;pointer to ADP function call parameter
adp_pb_seg      dw      ?               ;       block
top_mem32_os    dw      ?               ;pointer to top of system memory from
top_mem32_seg   dw      ?               ;       16M to below 4G
top_pmem32_os   dw      ?               ;pointer to top of planar system memory
top_pmem32_seg  dw      ?               ;       from 16M to below 4G
adp_pb          ends

;Resource Query parameter block
res_query       struc
rqfunction      db      ?               ;ADP function call
rqslot          db      ?               ;slot to perform function on
start_addr_high dw      ?               ;high word of starting address
start_addr_low  dw      ?               ;low word of starting address
end_addr_high   dw      ?               ;high word of ending address
end_addr_low    dw      ?               ;low word of ending address
size_high       dw      ?               ;high word of memory block size
size_low        dw      ?               ;low word of memory block size
res_query       ends
```

```
;-------------------------------------------------------------------;
;  Begin program:
;-------------------------------------------------------------------;

code    segment para    public 'code'
start   proc    far
        assume  cs:code,ds:code,ss:code
        org     100h                    ;program starts at offset 100h

        push    bp                      ;save base pointer on stack
        mov     bp,sp                   ;initialize base pointer to stack pointer
        push    es                      ;save registers
        push    ds
        push    si
        push    di
        jmp     prog                    ;jump around data area

;-------------------------------------------------------------------;
;  Data area:
;-------------------------------------------------------------------;

data_seg        dw      ?               ;holds value of DS for Setup program


prog:   mov     cs:data_seg,ds          ;save value of DS from Setup program
        mov     ax,[bp+8]               ;get segment of ADP parameter block
        mov     ds,ax                   ;put ADP parameter block segment in DS
        mov     di,[bp+6]               ;put offset of ADP parameter block in DI

;-------------------------------------------------------------------;
;  Determine what ADP function is to be performed.
;-------------------------------------------------------------------;

        mov     al,[di].function        ;get ADP function
        cmp     al,configure            ;is function configure?
        je      do_configure            ;yes, branch
        cmp     al,deallocate           ;is function deallocate?
        je      do_deallocate           ;yes, branch
        cmp     al,deallocate_all       ;is function deallocate all?
        je      do_deallocate_all       ;yes, branch
        cmp     al,deduce               ;is function deduce?
        jne     bad_command
        jmp     do_deduce               ;yes, branch
bad_command:
        mov     al,0ffh                 ;function not supported return code
        jmp     exit                    ;exit the ADP
```

```
;-------------------------------------------------------------------;
;  Configure the adapter.
;-------------------------------------------------------------------;

do_configure:
        call    get_rq_pb               ;point ES:SI to resource query param blk
        mov     es:[si].rqfunction,request_mem_block    ;function to perform
        mov     al,[di].slot            ;get slot of adapter being configured
        mov     es:[si].rqslot,al                       ;slot to configure
        mov     es:[si].start_addr_high,begin_block_h   ;beginning of allowable
        mov     es:[si].start_addr_low,begin_block_l    ;       range
        mov     es:[si].end_addr_high,end_block_h       ;end of allowable range
        mov     es:[si].end_addr_low,end_block_l
        mov     es:[si].size_high,size_h                ;size of block required
        mov     es:[si].size_low,size_l
        call    rqfunction_call         ;do the resource query function call
        cmp     es:[si].rqfunction,0    ;was the function successful?
        je      set_pos_bits            ;yes, set the appropriate POS bits
        jmp     unsuccessful

set_pos_bits:
        mov     bx,es:[si].start_addr_high              ;get assigned address
        mov     al,[di].slot            ;slot value to get pos array pointer for
        call    point_to_pos            ;point ES:SI to pos array
        mov     es:[si+3],bl            ;set POS value for register 103h
        mov     es:[si+4],bh            ;set POS value for register 104h
        jmp     successful

;-------------------------------------------------------------------;
;  De-allocate resources for the adapter.
;-------------------------------------------------------------------;

do_deallocate:
        mov     al,[di].slot            ;get slot of adapter being configured
        call    deallocate_mem          ;call proc to de-allocate the memory
        call    point_to_pos            ;get pointer to pos array for adapter
        and     byte ptr es:[si+2],not_mem_enable       ;clear memory enable bit
        mov     al,0                    ;successful error code
        jmp     exit                    ;exit ADP
```

```
;-------------------------------------------------------------------;
;  De-allocate resources for all adapters.
;-------------------------------------------------------------------;

do_deallocate_all:
        mov     cx,8                    ;max number of adapter slots per system
        mov     ax,[di].id_list_seg     ;get segment of list of adapter IDs
        mov     es,ax
        mov     si,[di].id_list_os      ;get offset of list of adapter IDs
        mov     al,1                    ;first slot to check

deallocate_lup:
        add     si,2                    ;point to next ID in list (skip planar)
        cmp     es:[si],adapter_id      ;does current slot have this adapter?
        jne     no_adapter              ;no, skip dealloction of memory
        call    deallocate_mem          ;call proc to deallocate the memory
        call    point_to_pos            ;get pointer to pos array for adapter
        and     byte ptr es:[si+2],not_mem_enable       ;clear memory enable bit

no_adapter:
        inc     al                      ;next slot to check
        loop    deallocate_lup

        mov     al,0                    ;successful return code
        jmp     exit                    ;exit ADP

;-------------------------------------------------------------------;
;  Deduce current configuration of adapter.
;-------------------------------------------------------------------;

do_deduce:
        mov     al,[di].slot
        call    point_to_pos            ;point ES:SI to pos array
        mov     al,es:[si+2]            ;read value of POS register 102h
        and     al,mem_enable_mask      ;see if the memory is enabled
        jz      unsuccessful            ;if mem enable bit not set, branch

        mov     bl,es:[si+3]            ;read value of POS register 103h
        mov     bh,es:[si+4]            ;read value of POS register 104h
        call    get_rq_pb               ;point ES:SI to resource query param blk
        mov     es:[si].rqfunction,request_mem_block    ;function to perform
        mov     al,[di].slot            ;get slot of adapter being deduced
        mov     es:[si].rqslot,al                       ;slot to deduce
        mov     es:[si].start_addr_high,bx              ;beginning of allocated
        mov     es:[si].start_addr_low,begin_block_l    ;       range
        add     bx,size_h - 1           ;determine end value for this request
        mov     es:[si].end_addr_high,bx                ;end of allocated range
        mov     es:[si].end_addr_low,end_block_l
        mov     es:[si].size_high,size_h                ;size of block required
        mov     es:[si].size_low,size_l
        call    rqfunction_call         ;do the resource query function call
        cmp     es:[si].rqfunction,0    ;was the function successful?
        je      successful              ;yes, branch
        jmp     unsuccessful
```

```
;-------------------------------------------------------------------;
;  ADP function was successful.
;-------------------------------------------------------------------;

successful:
        mov     al,[di].slot
        call    point_to_pos            ;get pointer to pos array
        or      byte ptr es:[si+2],mem_enable_mask      ;set memory enable bit
        mov     al,0                    ;successful return code for ADP function
        jmp     exit                    ;exit the ADP

;-------------------------------------------------------------------;
;  ADP function was unsuccessful.
;-------------------------------------------------------------------;

unsuccessful:
        mov     al,[di].slot
        call    point_to_pos            ;get pointer to pos array
        and     byte ptr es:[si+2],not_mem_enable       ;clear memory enable bit
        mov     al,1                    ;unsuccessful return code for ADP function

;-------------------------------------------------------------------;
;  Exit the ADP.
;-------------------------------------------------------------------;

exit:   mov     [di].function,al        ;save return code of ADP function
        call    set_pos_mask            ;initialize the pos mask bits
        pop     di                      ;restore registers
        pop     si
        pop     ds
        pop     es
        pop     bp
        ret
start   endp
```

```
;-----------------------------------------------------------------------;
;  DEALLOCATE_MEM
;       Perform deallocate ADP function on the slot defined in the
;       AL register.
;-----------------------------------------------------------------------;

deallocate_mem  proc    near
        push    ax
        push    es
        push    cx
        push    si
        mov     bl,al                   ;save slot number in BL register

del_begin:
        call    get_rq_pb               ;point ES:SI to resource query param blk
        mov     es:[si].rqfunction,check_mem_block      ;function to perform
        mov     es:[si].rqslot,bl       ;slot to check
        mov     es:[si].end_addr_high,0 ;always check first block in list
        call    rqfunction_call         ;do the resource query function call
        cmp     es:[si].rqfunction,0    ;is there a memory block assigned?
        jne     del_done                ;no, done de-allocating memory

        call    get_rq_pb               ;point ES:SI to resource query param blk
        mov     es:[si].rqfunction,deallocate_mem_block ;function to perform
        call    rqfunction_call         ;do the resource query function call
        jmp     del_begin

del_done:
        pop     si
        pop     cx
        pop     es
        pop     ax
        ret
deallocate_mem  endp

;-----------------------------------------------------------------------;
;  GET_RQ_PB
;       Initializes ES:SI to point to the ADP function call
;       (resource query) parameter block.
;-----------------------------------------------------------------------;

get_rq_pb       proc    near
        push    ax
        mov     ax,[di].adp_pb_seg      ;put the segment of the ADP function
        mov     es,ax                   ;       call parameter block in ES
        mov     si,[di].adp_pb_os       ;put the offset in SI
        pop     ax
        ret
get_rq_pb       endp
```

```
;-----------------------------------------------------------------------;
;  RQFUNCTION_CALL
;       Makes an ADP function call.
;-----------------------------------------------------------------------;

rqfunction_call proc    near
        push    ds                      ;save registers
        push    es
        push    si
        push    di
        push    bx
        push    cx

        mov     bx,di                   ;set BX to start of ADP parameter block
        add     bx,res_query_os         ;ES:BX points to ADP function call entry
        mov     ax,cs:data_seg          ;restore the Setup data segment
        mov     ds,ax
        call    dword ptr es:[bx]       ;make the ADP function call

        pop     cx                      ;restore the registers
        pop     bx
        pop     di
        pop     si
        pop     es
        pop     ds
        ret
rqfunction_call endp

;-----------------------------------------------------------------------;
;  POINT_TO_POS
;       Initializes ES:SI to point to the pos array for the slot
;       number defined in the AL register.
;-----------------------------------------------------------------------;

point_to_pos    proc    near
        push    ax
        push    bx

        mov     bl,al                   ;slot number to get pos array pointer for
        mov     ax,[di].pos_array_seg   ;put the pointer to the beginning of the
        mov     es,ax                   ;       pos array into ES
        mov     si,[di].pos_array_os    ;put pointer to beginning in SI
        mov     ax,pos_array_size       ;get the size of each adapter pos array
        mul     bl                      ;multiply the size by the slot number
        add     si,ax                   ;set SI to the start of the pos array
                                        ;       for the specified slot
        pop     bx
        pop     ax
        ret
point_to_pos    endp
```

```
;-------------------------------------------------------------------;
;  SET_POS_MASK
;       Initializes the bits in the POS mask.
;-------------------------------------------------------------------;

set_pos_mask    proc    near
        mov     ax,[di].pos_mask_seg    ;put the segment of the pos mask into
        mov     es,ax                   ;       ES
        mov     si,[di].pos_mask_os     ;put segment of pos mask in SI
        mov     cx,max_pos_bytes        ;set CX to the number of POS bytes
        push    si
lup2:   mov     byte ptr es:[si],0      ;set all pos mask values to 0
        inc     si
        loop    lup2
        pop     si
        mov     byte ptr es:[si],mem_enable_mask        ;set mem enable bit for
                                                        ;   POS register 102h
        mov     byte ptr es:[si+1],0ffh ;set all bits for POS register 103h
        mov     byte ptr es:[si+2],0ffh ;set all bits for POS register 104h
        ret
set_pos_mask    endp

code    ends
        end     start
```

Initialization Programs

An initialization program is a memory image program that is loaded onto track 0, head 0 of drive hex 80. The master boot record executes the initialization programs before booting the operating system from the fixed disk (the initialization program is bypassed if the operating system is booted from the diskette drive). It is an alternate method of initializing an adapter that does not require the use of ROM or jumpers and switches on the adapter.

The Set Configuration program, selected from the Main Menu of the Reference Diskette, installs the initialization programs onto drive C. (Automatic Configuration does not install these programs.) The Set Configuration program determines that an adapter has an initialization program by checking the adapter ADF. An adapter with a corresponding initialization program has the statement, InitProg <attribute>, in its ADF. The InitProg statement is placed after the FixedResources statement and before the NamedItem statement.

The attribute is a value from 0 to 127 that selects the order in which the initialization programs are run; the lowest value is run first. Attributes from 30 to 80 are available; attributes outside of this range are reserved. An example of the use of this attribute is to ensure that a specific adapter is initialized before another if both are installed in the same system.

The following rules apply to all initialization programs:

- The initialization program is always loaded beginning at offset hex 7C00. Any data variables referenced in the code may need their offsets adjusted to hex 7C00.
- The initialization program must begin with a 20 byte header. Execution of the initialization program begins on the 21st byte.
- The 16th byte of the program header is the revision level of the initialization program. The value in this byte should be incremented if a new revision of the program is released. When Set Configuration is run, the installation process checks this byte to ensure that the program with the latest revision level is installed. The other bytes in the header are set by the Set Configuration program.
- The initialization program cannot be more than 1024 bytes in size, including the 20 byte header.
- While executing, the initialization program should not assume that memory is available beyond its own programming size.
- The initialization program is shipped as a DOS file so that it can be merged onto the Reference Diskette in the same way as the ADFs. The name of the file is based on the ID of the adapter, and is in the form ICardID.ADF (CardID is the adapter ID in hex).
- When an initialization program finishes executing, it performs a Near Return to pass control back to the calling program. The stack will then be in the same state it was when the initialization program was called.

A revision to an initialization program is installed the same way as the original is installed. The revision to the program has a higher value in the revision level byte than previous versions. The Set Configuration program compares the revision levels of initialization programs installed on the fixed disk drive with those on the Reference Diskette. If the program on the Reference Diskette has a greater revision level than the one on the fixed disk drive, it updates the program on the fixed disk drive.

Index

A

adapter configuration call 29
adapter configuration, order of 4
adapter description program (ADP) 20
  error codes 40
  example 41
  function call entry 27
  function codes 28
  memory above 16MB 27
  nonsystem memory 22
  POS mask 26
  program call 23
  query parameters 27
  resource query 34
  syntax 21
  system memory 21
adapter identification 2
adapter init programs 49
adapter location 39
adapterid keyword 8
adaptername keyword 8
address keyword 14
arb keyword 13
array, POS 26
assistance, developer 2
automatic configuration 1, 3

B

backup configuration utility 5
battery failure 3
begin_device parameter 15

C

call, ADP function 23
change configuration utility 5
check memory block 36
check validity of choice 32
choice keyword 11
CMOS RAM, card ID bytes 2
configuration determination call 29
configuration error 3
configure system memory 31
configure the adapters 28
copy an option diskette utility 6
CRC error 3

D

deallocate memory block 37
deallocate resources 29
deallocate system memory 32
deduce configuration 29
deduce system memory configuration 32
developer assistance 2
device keyword 15
disable adapter 37
disable block of memory 30
diskettes, reference and option 1

E

enable adapter 37
enable memory by prompt 38
entry point, function call 27
error codes 40
example, ADP 41
exec keyword 14

F

file naming 1
fixedresources keyword 8
function call entry point, ADP 27
function call, ADP 23
functions, ADP 28
  check validity 32
  configure system memory 31
  configure the adapters 28
  deallocate resources 29
  deallocate system memory 32
  deduce configuration 29

© Copyright IBM Corp. 1989, 1990

functions, ADP *(continued)*
deduce system memory 32
disable memory 30
remap memory 31
functions, query 34
check memory block 36
deallocate memory block 37
enable memory prompt 38
enable/disable adapter 37
get slot data 39
request memory block 35

G

get slot information 39

H

help keyword 12
help text 5

I

implement_id parameter 15
information on adapter slots 39
initialization programs 49
initprog keyword 9
input keyword 11
int keyword 13
io keyword 13
items parameter 9

L

location, adapter 39
lock keyword 9

M

masking POS 26
mem keyword 14
memory above 16MB 27
memory and ADPs 21
memory below 16MB 25
memory call, disable 30
memory call, remap 31
memory prompt 38
memory, system and
nonsystem 20

N

nameditem keyword 10
naming convention, file 1
nodma keyword 15
nonsystem memory 20, 22
nonvolatile RAM, card ID bytes 2
numbytes keyword 8

O

option diskette 2

P

parameter table 25
pio keyword 14
POS array 26
pos keyword 12
POS mask 26
POST error message files 3
program call, ADP 23
prompt keyword 11
prompt to enable memory 38

Q

query parameters 27

R

reference diskette 1
remap memory 31
request memory block 35
resource query functions 34
check memory block 36
deallocate memory block 37
enable memory prompt 38
enable/disable adapter 37
error codes 40
get slot data 39
request memory block 35
resource query parameters 27
resource_setting parameter 13

restore configuration utility 5
rs422 keyword 15

S

set configuration program 2
setup, system board 1
sharedarb keyword 13
syntax for ADPs 21
sysmem keyword 9
system board memory above 16MB 27
system board memory below 16MB 25
system board setup 1
system configuration utilities 1
system memory 20
system memory above 16MB 27
system memory ADP 21
system memory assignment 32
system memory below 16MB 25
system memory call, configure 31
system memory, deallocate 32
system resources 2

V

validity check 12, 32
vcheck keyword 12
view configuration 5

Notes:

Micro Channel Architecture

Description 1
Micro Channel Participants 2
Masters 2
Slaves 3
Micro Channel Structure 5
Addressing Model 6

Micro Channel Buses and Signals 7
Arbitration Bus and Associated Signals 7
Address Bus and Associated Signals 9
Address Signal Combinations 12
Data Bus and Associated Signals 13
Interrupt Signals 18
Other Micro Channel Signals 19
Channel Signal Groups (Digital) 20
Matched-Memory Extension Signal Descriptions 23
Auxiliary Video Extension Signal Descriptions 23
Micro Channel Architecture Base and Extensions 24
32-Bit Connector and Extensions 25
16-Bit Connector and Extensions 26

Micro Channel Procedures 27
Arbitration 27
Channel Ownership 27
Central Arbitration Control Point 28
Local Arbiters 30
Burst Transfer 32
Preemption 33
Fairness Feature and the Inactive State 34
Arbitration Timing 35
Data Transfer 39
Basic Transfer 40
Basic Transfer Procedure 40
Default Cycle Timing (200 ns Minimum) 43
Synchronous-Extended Cycle Timing (300 ns Minimum) 47
Asynchronous-Extended Cycle Timing (≥300 ns) 50
Data Steering 53
Data Steering Control 54
Master Replication of Data Bytes 55

© Copyright IBM Corp. 1989, 1990

Channel Support 55
Address Bus Translator 55
Central Data Steering 56
TR 32 56
Central Steering Logic 56
Central Translator Logic 56
-BE(0-3) 56
Streaming Data Transfer 58
Streaming Data Procedure (16- or 32-Bit) 59
Streaming Data Procedure (64-Bit) 61
Streaming Data Transfer Rates 63
Address Bus Management 63
Address Boundary Alignment 64
Streaming Data Cycles 66
64-Bit Streaming Data Cycles 68
Starting the Cycle 70
Deferring the Start of the Cycle 70
Pacing the Data 72
Terminating the Cycle 74
Streaming Data Timing 77
Direct Memory Access Procedure 83
Read and Write Operations 83
DMA Slave Selection 84
Address During DMA Operation 84
Transfer Data Length 84
DMA Procedure 84
Single Transfer 85
Burst Transfer (Multiple Transfer Cycles) 86
DMA Terminating Conditions 86
Use of -BURST 87
DMA Timings 88
Single DMA Transfer (DMA-Controller Controlled) 88
Burst DMA Transfer (DMA-Controller Terminated) 90
Burst DMA Transfer (DMA-Slave Terminated - Default Cycle, 200 ns) 91
Burst DMA Transfer (DMA-Slave Terminated - Synchronous-Extended Cycle, 300 ns) 94
Burst DMA Transfer (DMA Slave Terminated - Asynchronous-Extended Cycle, ≥300 ns) 96
Interrupt Procedures 98
Refresh 100
System Configuration 101
Multiple Card Adapters 102

Adapter Configuration Data 103
Required Configuration Fields 103
Adapter ID Field 104
Card Enable Field 104
Channel Check Field 104
Conditionally-Required Configuration Fields 104
Adapter I/O Address Select Field 105
Adapter ROM/RAM Address Select Field 105
Arbitration Level Field 105
Auto-Incrementing Enable Field 105
Channel Check Exception Reporting Field 106
Channel Check Status Indicator Field 106
Data and Address Parity Enable Field 106
Device I/O Address Assignment 106
Matched Memory Enable Field 107
Selected Feedback Return Exception Enable Field 107
Streaming Data Enable Field 108
Optional Configuration Fields 108
Option Select Data Field 108
Fairness Feature Enable Field 108
POS Subaddress Extension Field 108
Status Information Field 111
System Configuration Procedure 112
Address Bus During Setup Cycles 112
Card Enable Field During Setup Cycles 113
Channel Check Status During Setup Cycles 113
-CD SETUP During Setup Cycles 113
CD CHRDY During Setup Cycles 113
System Configuration Timing Diagrams 114
Exception Condition Reporting and Handling 118
Exception Conditions 118
Channel Data and Address Parity Exceptions 119
Data Parity 120
Read Operations - Basic Transfer 120
Read Operations - Streaming Data Transfer 122
Write Operations - Basic Transfer 122
Write Operations - Streaming Data Transfer 124
Data Parity Exception Handling 126
Bus Master 126
System Master 126
DMA Controller 126
Slaves 126
Address Parity 127

Channel Non-Parity Exceptions 128
Invalid -DS 16 RTN and -DS 32 RTN Combination 128
Selected Feedback Return Exception 129
Master and Slave Dependent Exceptions 130
Channel Timeout Exception 130
Exception Reporting 131
Exception Reporting using -CHCK 131
Synchronous -CHCK 132
Asynchronous -CHCK 137
Exception Reporting Using Interrupts 138
Exception Handling 138
Bus Master 138
System Master 138
DMA Controller 139
Slaves 139
Select Feedback Exception Handling 140
Bus Master 140
System Master 140
DMA Controller 140
Channel Reset 141

Connector Definition 142
| Micro Channel Connector 144
| Matched-Memory Extension 148
| Auxiliary Video Extension 149
| Base Video Extension 149
| Adapter Physical Specifications 150
| Adapter Size 151
Connector Spacing 151
Power 152
Voltage Regulation 153

Index 154

Figures

1. Master-Slave Relationships 4
2. Micro Channel Structure 5
3. I/O and Memory Address Space 6
4. Address Bus Support - Memory Cycle 12
5. Address Bus Support - I/O Cycle 12
6. Port Size of Selected Slave 16
7. Streaming Data Rates 17
8. I/O and Memory Transfer Controls 17
9. Driver and Receiver Requirements and Options 21
10. Signal Groups, Driver Types, and Loading Specifications . 22
| 11. The 32-Bit Connector and the Optional Extensions 25
| 12. The 16-Bit Connector and the Optional Extensions 26
13. Arbitration Block Diagram 29
14. Local Arbiter Example 31
15. Burst Transfer 32
16. Preempted Burst Transfer 33
| 17. Exiting from the Inactive State 35
18. Arbitration Cycle 36
19. Arbitration Cycle Timing Specifications 36
20. First Cycle After Grant 38
21. First-Cycle-After-Grant Timing Specifications 38
22. Basic Transfer Procedure - Overview 42
23. Default Cycle (200 ns Minimum) 44
24. Default Cycle (200 ns Minimum) Timing Specifications . . . 44
25. Synchronous-Extended Cycle (300 ns Minimum) 48
26. Synchronous-Extended Cycle Timing Specifications (300 ns Minimum) 48
27. Asynchronous-Extended Cycle Timing Specifications (≥300 ns) 50
28. Asynchronous Extended Cycle (≥ 300 ns) 51
29. Steering Control 53
30. Steering Responsibility (Basic Transfer, DMA, and Setup Procedures) 54
31. Steering Responsibility (Streaming Data Procedure) 55
32. Data Bus Steering Implementation 57
33. Streaming Data Transfer Rates 63
34. Data Block Example 65
35. Streaming Data Signal Combinations 66
36. 16- and 32-Bit Streaming Data Cycle Overview 67

© Copyright IBM Corp. 1989, 1990

37. 64-Bit Streaming Data Cycle Overview 68
38. Streaming Data Deferred Start Overview 71
39. CD CHRDY and Data Pacing 73
40. Master-Terminated Streaming Data Cycle 74
41. Master-Terminated Streaming Data Cycle (Slave Not Ready) 75
42. Slave-Terminated Streaming Data Cycle 76
43. Streaming Data Cycle 77
44. Streaming Data Cycle - Deferred Start 78
45. Streaming Data Transfer Timing Specifications 78
46. Streaming Data Clocking (16- and 32-Bit) 80
47. Streaming Data Transfer Timing Specifications (16- and 32-Bit) 80
48. Streaming Data Clocking (64-Bit) 82
49. Streaming Data Transfer Timing Specifications (64-Bit) . . . 83
50. Single DMA Transfer (DMA-Controller Controlled) 88
51. Single DMA Transfer (DMA-Controller Controlled) Timing Specifications 89
52. Burst DMA Transfer (DMA-Controller Terminated) 90
53. Burst DMA Transfer (DMA-Controller Terminated) Timing Specifications 90
54. Burst DMA Transfer (DMA-Slave Terminated - Default Cycle, 200 ns) 92
55. Burst DMA Transfer (DMA-Slave Terminated - Default Cycle, 200 ns) Timing Specifications 92
56. Burst DMA Transfer (DMA-Slave Terminated - Synchronous-Extended Cycle) 94
57. Burst DMA Transfer (DMA-Slave Terminated - Synchronous-Extended Cycle, 300 ns) Timing Specifications 95
58. Burst DMA Transfer (DMA-Slave Terminated - Asynchronous-Extended Cycle, ≥300 ns) 96
59. Burst DMA Transfer (DMA-Slave Terminated - Asynchronous-Extended Cycle, ≥300 ns) Timing Specifications 97
60. Typical Adapter Interrupt Sharing Implementation 98
61. POS I/O Address Space 103
62. Subaddressing Model 110
63. POS I/O Address Decode 112
64. Setup Cycle Overview 114
65. System Configuration Timing 115
66. System Configuration Timing Specifications 116
67. Data Parity Checking Read Cycle - Basic Transfer 121
68. Data Parity Checking Read Timing Specifications 121
69. Data Parity Read Cycle - Streaming Data Transfer 122

70. Data Parity Checking Write Cycle - Basic Transfer 123
71. Data Parity Checking Write Timing Specifications 123
72. Data Parity Write Cycle - Streaming Data Transfer 125
73. Data Parity Checking Write Timing Specifications 125
74. Address Parity 127
75. Address Parity Timing Specifications 127
76. Channel-Check - Default Cycle 133
77. Channel-Check Timing Specifications - Default Cycle ... 133
78. Channel-Check - Extended Cycle 134
79. Channel-Check Timing Specifications - Extended Cycle . 134
80. Channel Check - Streaming Data Read Cycle 135
81. Channel-Check Timing Specifications - Streaming Data Read Cycle 135
82. Channel Check - Streaming Data Write Cycle 136
83. Channel-Check Timing Specifications - Streaming Data Write Cycle 136
84. Micro Channel Bus and Signal Status after a CHRESET . 141
| 85. Micro Channel Connectors 143
| 86. Micro Channel Connector 145
| 87. Micro Channel Connector - continued 146
| 88. Micro Channel Connector - continued 147
89. Channel Connector Voltage and Signal Assignments (Matched-Memory Extension) 148
90. Auxiliary Video Extension 149
| 91. Base Video Extension 150
| 92. Minimum current available for the Type 3 adapter. 152
| 93. Minimum current available for the Type 5 adapter 152
94. Channel Voltage Regulation 153

Notes:

Description

The Micro Channel™ architecture consists of an address bus, a data bus, an arbitration bus, a set of interrupt signals, and support signals. It uses synchronous and asynchronous procedures for data transfer between memory, I/O devices, and a controlling master. The controlling master can be a DMA controller, the system master (system processor), or a bus master. The features of the Micro Channel architecture are:

- I/O data transfers of 8-, 16-, 24-, or 32-bits within a 64KB[1] address space (16-bit address width).
- Memory data transfers of 8-, 16-, 24-, or 32-bits within a 16MB[1] (24-bit address width) or 4GB[1] (32-bit address width) address space.
- An arbitration procedure that enables up to 15 devices and the system master to bid for control of the channel.
- A basic transfer procedure that allows data transfers between masters and slaves.
- A direct memory access (DMA) procedure that supports multiple DMA channels. Additionally, this procedure allows a device to transfer data in bursts.
- An optional streaming data procedure that provides a faster data-transfer rate than the basic transfer procedure and allows 64-bit data transfers.
- Address- and data-parity enable and detect procedures.
- Interrupt sharing on all levels.
- A flexible system-configuration procedure that uses programmable registers.

Micro Channel is a trademark of the IBM Corporation

1 KB equals 1024 bytes; MB equals 1,048,576 bytes; GB equals 1,073,741,824 bytes.

© Copyright IBM Corp. 1989, 1990

- An adapter interface to the channel using:
  - A 16-bit connector with a 24-bit address bus and a 16-bit data bus
  - A 32-bit connector with a 32-bit address bus and a 32-bit data bus
  - An optional matched-memory extension
  - An optional video extension.
- Support for audio signal transfer (audio voltage-sum node).
- Support for both synchronous and asynchronous data transfer.
- An exception condition reporting procedure.
- Improved electromagnetic characteristics.

Micro Channel Participants

All Micro Channel participants are either masters or slaves. There are three types of masters and three types of slaves. (See Figure 1 on page 4).

An adapter can incorporate either a master function, a slave function, or a combination of both. For example, an adapter might be designed to operate primarily as a DMA slave. However, it would also respond to certain I/O read and I/O write operations from the system master, making it an I/O slave. If the adapter contains RAM or ROM that is in the memory address space, it would be a memory slave when that memory was accessed.

Masters

A master is a participant that drives the address bus and data transfer control signals that cause data transfer to or from a slave.

The channel supports up to 16 masters. The central arbitration control point and the Micro Channel arbitration procedure control ownership of the channel. The central arbitration control point grants ownership at the end of the arbitration procedure to highest-priority requester.

The three types of masters are:

- System master

  A system master controls and manages the system configuration. It arbitrates for use of the channel. A system master can also be a default master. The default master is the master that owns the channel when no other master requires the channel.

  The system master supports data transfers with an I/O slave or a memory slave.

- Bus master

  A bus master arbitrates for use of the channel. A bus master supports data transfers with an I/O slave or a memory slave.

- DMA controller

  A DMA controller does not initiate arbitration for the channel; a DMA slave initiates and completes the arbitration procedure. The DMA controller monitors the arbitration bus to detect the DMA slave's arbitration level.

  A DMA controller supports data transfers with DMA slaves and memory slaves.

Slaves

A slave is a participant that sends and receives data under the control of a master. The slave responds to signals that are driven by the master. A slave is selected by the controlling master using Micro Channel procedures.

The channel supports 8-, 16- and 32-bit data bus size in a slave. The data port size (8-, 16-, or 32-bits) describes the maximum width of the data transfer. For example, a 32-bit slave can support 8-, 16- 24-, or 32-bit data transfers. During a data transfer cycle, an 8-bit data port on a 32-bit slave acts like an 8-bit slave.

The three types of slaves are:

- I/O slave

  I/O slaves are selected by their address within the I/O address space.

- Memory slave

  Memory slaves are selected by their address within the memory address space.

- DMA slave

  A DMA slave is selected by arbitration or, optionally, by its address within the I/O address space. A DMA slave initiates a request for the channel and arbitrates for the channel with arbitrating masters and other DMA slaves. The DMA controller provides the DMA slave with the address and data transfer signals required for data transfers. During a DMA write operation, the DMA slave provides the data, and the memory slave stores it. During a DMA read operation, the memory slave provides the data, and the DMA slave stores it.

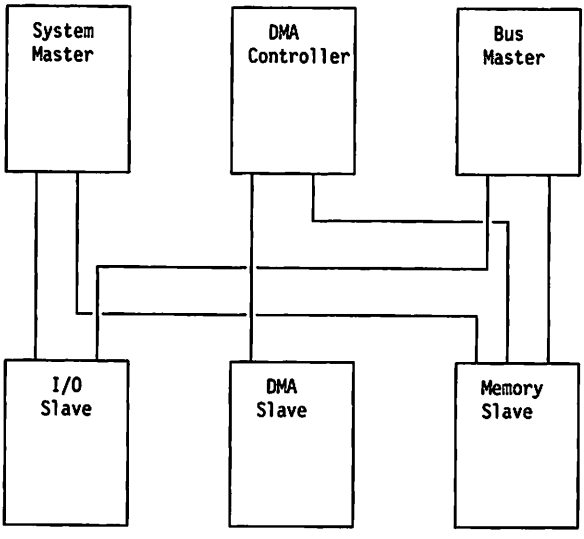
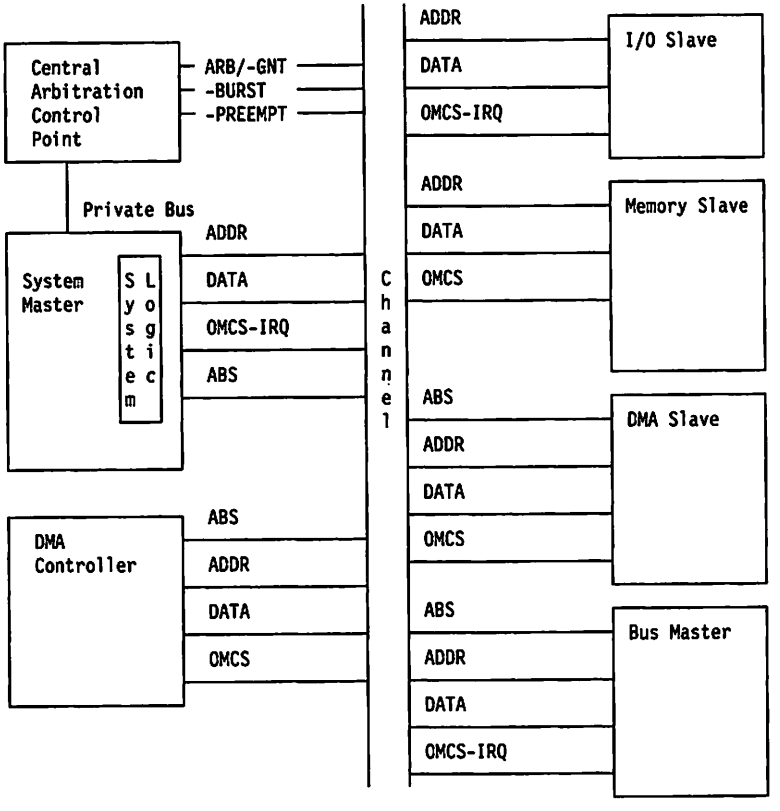


Figure 1. Master-Slave Relationships

Micro Channel Structure

The following shows the participants, the central arbitration control point, and the Micro Channel buses and signals that they drive or receive. "Micro Channel Buses and Signals" on page 7 describes the buses and signals of the Micro Channel architecture.



ABS - Arbitration Bus and Associated Signals
ADDR - Address Bus and Associated Signals
DATA - Data Bus and Associated Signals
IRQ - Interrupt Request Signals
OMCS - Other Micro Channel Signals

Figure 2. Micro Channel Structure

Figure 9 on page 21 shows which Micro Channel participant drives and receives each bus and signal, and specifies the requirements for each driver and receiver.

Addressing Model

The Micro Channel addressing model consists of a memory address space and an I/O address space. During an I/O cycle, the 64KB I/O address space is addressed by the low-order 16 bits of the address bus. During a memory cycle, the memory address space is addressed by the address bus.

The signals used for an address of up to 4GB are MADE 24, M/-IO, and A0 through A31 of the address bus. An address of up to 16MB uses A0 through A23 plus MADE 24 and M/-IO.

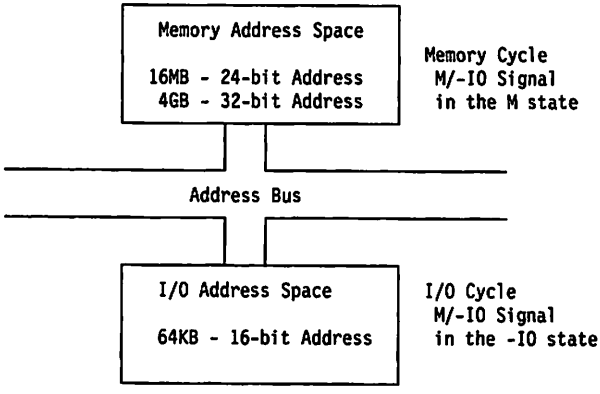


Figure 3. I/O and Memory Address Space

Micro Channel Buses and Signals

The Micro Channel architecture consists of:

- An arbitration bus and associated signals
- An address bus and associated signals
- A data bus and associated signals
- Interrupt signals
- Other Micro Channel signals
- Optional extensions for:
  - Matched memory extension signals
  - Video extension signals.

Throughout this document, a minus sign (-) in front of a signal name indicates that the signal is active when it is at a low-voltage level. When no minus sign appears, the signal is active when it is at a high-voltage level. For example, -CMD specifies the signal is active low. Also, ARB/-GNT is in the ARB state when it is at a high level and is in the -GNT state when it is at a low level.

Arbitration Bus and Associated Signals

The arbitration bus and associated signals allow arbitrating participants (the system master, bus masters, and DMA slaves) to request and gain ownership of the channel. The resolution of multiple arbitration requests results in granting ownership of the channel to the highest-priority requester. See "Arbitration" on page 27 for more information.

ARB0 – ARB3: Arbitration Bus: These signals make up the arbitration bus. They are driven by the system master, bus masters, and DMA slaves to present their arbitration level when requesting ownership of the channel. ARB0 (least significant) through ARB3 (most-significant) support up to 16 arbitration levels.

The highest value of the arbitration bus (hex F) has the lowest priority, and the lowest value (hex 0) has the highest priority. To

participate in the arbitration procedure, an arbitrating participant must present its arbitration level immediately after the rising edge of ARB/-GNT. All arbitrating participants monitor the arbitration bus, and those with lower priority arbitration levels withdraw them by not driving less-significant arbitration bits.

The arbitration level of the highest-priority requester is valid on the arbitration bus after a settling time. After the channel is granted to the highest-priority requester, that requester continues to drive its arbitration level on the bus.

ARB/-GNT: Arbitrate/-Grant: Only the central arbitration control point drives this signal. The negative-to-positive transition of ARB/-GNT initiates an arbitration cycle. When in the ARB state, this signal indicates an arbitration cycle is in progress. When in the -GNT state, this signal indicates the acknowledgment from the central arbitration control point to the arbitrating participants and the DMA controller that channel ownership has been granted. This signal is driven to the ARB state by the central arbitration control point following the end of transfer (EOT). (See page 28 for a description of the end of transfer.)

Note: The system master can perform data transfers during arbitration (ARB/-GNT in the ARB state).

-BURST: -Burst: This signal is optionally driven by the winning arbitrating participant or the DMA controller after ARB/-GNT is driven to the -GNT state. This signal indicates to the central arbitration control point that the controlling master will use the channel for one or more consecutive data transfer cycles. This type of data transfer is called *burst transfer*.

-PREEMPT: -Preempt: This signal is used by arbitrating participants to request use of the channel through arbitration. Any arbitrating participant that requires ownership of the channel drives -PREEMPT active, causing an arbitration cycle to occur. When a participant is granted control of the channel, it stops driving -PREEMPT. All arbitrating participants that have not been granted ownership keep their requests pending by continuing to drive -PREEMPT active. All masters and DMA slaves that use burst transfer must receive -PREEMPT.

Address Bus and Associated Signals

The address bus and the associated signals are used by the controlling master to assert the memory address or the I/O address (M/-IO), to enable a slave to latch the address and status signals (-S0, -S1), and to indicate that the memory address is greater than 16MB.

A0 – A23: Address Bits 0 through 23: These lines, along with A24 through A31, make up the address bus. These lines are driven by the controlling master to address memory, I/O slaves, and, optionally, DMA slaves. A0 is the least-significant bit and A23 is the most-significant bit. These 24 address lines allow access of up to 16MB of memory. Only the lower 16 address lines (A0 through A15) are for I/O operations, and all 16 lines must be decoded by the I/O slave. A0 through A23 are generated by the controlling master. Valid addresses, generated by the controlling master, are unlatched on the channel and, if required, must be latched by the slaves using either the leading or trailing edge of -ADL or the leading edge of -CMD.

A24 – A31: Address Bits 24 through 31: These lines, along with A0 through A23 are driven by the controlling master to address memory attached to the channel. A0 is the least-significant bit and A31 is the most-significant bit. These additional address lines allow access of up to 4GB of memory. A24 through A31 are generated by the controlling master. Valid addresses, generated by the controlling master, are unlatched on the channel and, if required, must be latched by the slaves using either the leading or trailing edge of -ADL or the leading edge of -CMD.

Note: A0 - A31 are used to transfer data during a 64-bit streaming data cycle.

-ADL: -Address Decode Latch: This signal, driven by the controlling master, is provided as a convenient way for the slave to latch valid address decodes and status bits.

-APAREN: -Address Parity Enable: This optional signal is driven active by a master when the master places an address on the bus. This signal indicates to a slave that the address parity signals are valid.

APAR0 – APAR3: Address Parity Bits 0 through 3: These optional signals are driven by a master when an address is placed on the address bus. These signals represent the odd parity of the address bits on the address bus during both read and write operations. (Odd parity is the condition where the total number of 1s in a byte of data, including the parity bit, is odd.)

APAR(0) represents the odd parity of A(0-7).
APAR(1) represents the odd parity of A(8-15).
APAR(2) represents the odd parity of A(16-23).
APAR(3) represents the odd parity of A(24-31).

During both read and write operations, a master generates a parity bit for each valid address byte, and the receiving slave optionally performs the parity checking to ensure the integrity of the address.

Note: APAR0 - APAR3 represent data parity during 64-bit streaming data cycle when -DPAREN is active.

APAR(0) represents the odd parity of D(32-39).
APAR(1) represents the odd parity of D(40-47).
APAR(2) represents the odd parity of D(48-55).
APAR(3) represents the odd parity of D(56-63).

-CD SFDBK (n): -Card Selected Feedback: This signal is driven active by the selected slave as a positive acknowledgement of the slave's selection. The (n) indicates this signal is unique to each channel connector (one independent signal per connector). This signal is unlatched and driven by the selected slave. The slave does not drive -CD SFDBK during the configuration procedure (-CD SETUP active).

Note: Memory that contains diagnostic code must not drive -CD SFDBK during the diagnostic operation.

MADE 24: Memory Address Enable 24: This signal is driven by the controlling master and decoded by all memory slaves, regardless of the size of their address-space. When this signal is active, A24 - A31 are undefined.

- A master driving only A0 - A23 drives MADE 24 active.
- A master driving A0 - A31 drives MADE 24:
  - Active when all bits in A24 - A31 are 0
  - Inactive when any bit in A24 - A31 is 1.

M/-IO: Memory/-Input Output: This signal is driven by the controlling master and decoded by all slaves. This signal selects a memory cycle or an I/O cycle. When this signal is in the M state, a memory cycle is selected. When this signal is in the -IO state, an I/O cycle is selected.

-SFDBKRTN: Selected Feedback Return: This optional signal is generated by the system logic from the AND of the -CD SFDBK(n) signals being driven by slaves. This signal is a positive acknowledgement to the master from the slave that the slave is at the address specified by the master. Masters that support address parity must receive this signal.

Address Signal Combinations

The following tables show the valid addressing signal combinations for:

- the address bus (A0 - A23 and A24 - A31)
- M/-IO
- MADE 24
- -APAR(0-3).

Addressing Signals	16-Bit Connector	32-Bit Connector	
	24-Bit Address	< 16MB Address	≥ 16MB Address
M/-IO	1	1	1
MADE 24	1	1	0
A0 - A23	Valid	Valid	Valid
APAR0 - APAR2 (Optional)	N/A	Valid	Valid
A24 - A31	N/A	X	Valid
APAR3 (Optional)	N/A	X	Valid
X - Don't Care	N/A - Not applicable		

Figure 4. Address Bus Support - Memory Cycle

Addressing Signals	16-Bit Connector	32-Bit Connector
M/-IO	0	0
MADE 24	X	X
A0 - A15	Valid	Valid
APAR0 - APAR1 (Optional)	N/A	Valid
X - Don't Care	N/A - Not applicable	

Figure 5. Address Bus Support - I/O Cycle

Data Bus and Associated Signals

The data bus is used to transfer either 8, 16, 24, or 32 bits of data. The associated signals indicate the amount of data transferred by the master in a single transfer cycle, the size of the slave's data port, and the type (read or write) of the data transfer.

D0 – D15: Data Bits 0 through 15: These lines, along with D16 - D31, make up the data bus. The data bus is driven by any master or slave that is transferring data. These lines (D0 - D15) provide data bits 0 through 15. D0 is the least-significant bit; D15 is the most-significant bit. The 16-bit transfers from the controlling master to an 8-bit slave are converted by the controlling master to two 8-bit transfers, and are transmitted on lines D0 through D7. An 8-bit slave must use D0 through D7 to communicate with the controlling master.

D16 – D31: Data Bits 16 through 31: These lines, along with D0 - D15, make up the data bus. The data bus is driven by any master or slave that is transferring data. These lines (D16 - D31) provide data bits 16 through 31. D0 is the least-significant bit; D31 is the most-significant bit. The 32-bit transfers from the controlling master to an 8-bit slave are converted to four 8-bit transfers by the controlling master, and are transmitted on lines D0 through D7. The 32-bit transfers from the controlling master to a 16-bit slave are converted to two 16-bit transfers by the controlling master, and are transmitted on lines D0 through D15.

-BE0 – -BE3: -Byte Enable 0 through 3: These signals are used during data transfers with 32-bit slaves to indicate which data bytes are valid on the data bus. Data transfers of 8, 16, 24, or 32 contiguous bits are controlled by -BE0 through -BE3 during transfers involving 32-bit slaves only. These signals are driven by the controlling master when TR 32 is inactive, and by the central translator logic (for those operations involving a 16-bit master with a 32-bit slave) when TR 32 is active. These signals are not latched on the bus and, if required, are latched by 32-bit slaves. (See "Channel Support" on page 55 for additional information.)

-CD DS 16 (n): -Card Data Size 16: This signal is driven by 16-bit and 32-bit slaves to indicate a 16-bit or 32-bit data port at the location addressed. The (n) indicates this signal is unique to each channel connector (one independent signal per connector). This signal is derived from a valid unlatched address decode. -CD DS 16 is not driven by 8-bit slaves and is inactive for an 8-bit data port.

-CD DS 32 (n): -Card Data Size 32: This signal, along with -CD DS 16, is driven by 32-bit slaves to indicate a 32-bit data port at the location addressed. The (n) indicates this signal is unique to a channel connector position (one independent signal per connector). -CD DS 32 is derived from a valid unlatched address decode. -CD DS 32 is inactive for an 8- or 16-bit data port.

CD CHRDY (n): Channel Ready: This signal is normally active (ready) and is driven inactive (not ready) by a slave to allow additional time to complete a channel cycle. The (n) indicates this signal is unique to each channel connector (one independent signal per connector).

During a read cycle, a slave ensures that data is valid within the time specified after releasing the signal to a ready state. The slave also holds the data long enough for the controlling master to sample the data. A slave can also use this signal during a write cycle if more time is needed to store the data. This signal is initially driven from a valid unlatched address decode and status active.

CHRDYRTN: Channel Ready Return: This signal is the AND of CD CHRDY (n). It is driven by the system logic. If all slaves drive CD CHRDY active, this signal is active. CHRDYRTN allows the controlling master to monitor the ready information.

-CMD: -Command: This signal is driven by the controlling master and is used to define when data on the data bus is valid. The trailing edge of this signal indicates the end of the bus cycle. This signal indicates to the slave how long data is valid. During write cycles, the data is valid as long as -CMD is active. During read cycles, the data is valid after the leading edge, but before the trailing edge, of -CMD and is held on the bus until after -CMD goes inactive. Slaves can latch address and status information with the leading edge of -CMD.

-DPAREN: -Data Parity Enable: This optional signal is driven active by the participant when data is placed on the data bus. This signal indicates that the data parity signals are valid.

Note: APAR(0) - APAR(3) represent data parity during 64-bit streaming data cycles when -DPAREN is active.

DPAR0 – DPAR1: Data Parity Bits 0 and 1: These optional signals are driven by the participant when data is placed on the data bus. These signals represent the odd parity of the data bits on the data bus during both read and write operations. (Odd parity is the condition where the total number of 1s in a byte of data, including the parity bit, is odd.)

DPAR(0) represents the odd parity of D(0-7).
DPAR(1) represents the odd parity of D(8-15).

During write operations, a master generates a parity bit for each data byte being transferred, and the receiving slave optionally performs the parity checking to ensure the integrity of the data. During read operations, a slave generates a parity bit for each valid data byte, and the receiving master performs the parity checking to ensure the integrity of the data.

DPAR2 – DPAR3: Data Parity Bits 2 and 3: These optional signals are driven by the participant when data is placed on the data bus. These signals represent the odd parity of the data bits on the data bus during both read and write operations.

DPAR(2) represents the odd parity of D(16-23).
DPAR(3) represents the odd parity of D(24-31).

During write operations, a master generates a parity bit for each data byte being transferred, and the receiving slave optionally performs the parity checking to ensure the integrity of the data. During read operations, a slave generates a parity bit for each valid data byte, and the receiving master performs the parity checking to ensure the integrity of the data.

-DS 16 RTN: -Data Size 16 Return: This signal is driven by the system logic. This signal is the AND of -CD DS 16 (n) from each channel connector. If any slave drives its -CD DS 16 active, this signal is active. This signal allows the controlling master to monitor the information about the selected slave's data port size (see Figure 6).

-DS 32 RTN: -Data Size 32 Return: This signal is driven by the system logic. This signal is the AND of -CD DS 32 (n) from each channel connector. If any slave drives its -CD DS 32 active, this signal is active. This signal allows the controlling master to monitor the information about the selected slave's data port size.

-DS 16 RTN	-DS 32 RTN	Data Port Size
1	1	8-Bit Data Port
1	0	Not Valid
0	1	16-Bit Data Port
0	0	32-Bit Data Port

Figure 6. Port Size of Selected Slave

-MSDR: -Multiplexed Streaming Data Request: This signal is driven by a slave to indicate to the controlling master that the slave is capable of 64-bit streaming data transfers.

-SBHE: -System Byte High Enable: This signal is driven by the controlling master to indicate and enable transfers of data on D8 - D15. It is used with A0 to distinguish between high-byte transfers (D8 – D15) and low-byte transfers (D0 – D7) and double-byte (16-bit) transfers to 16-bit data ports. All 16-bit slaves receive this signal.

-SD STROBE: -Streaming Data Strobe: The controlling master and the slave use this signal to clock data on and off the data bus. This optional signal is driven by the controlling master. This signal also indicates to the slave that the controlling master supports streaming data.

-SDR(0): -Streaming Data Request 0: This optional signal is driven by a slave to indicate to the controlling master that the slave is capable of streaming data, and also indicates the maximum clocking rate the slave supports.

-SDR(1): -Streaming Data Request 1: This optional signal is driven by a slave to indicate to the controlling master that the slave is capable of streaming data, and also indicates the maximum clocking rate the slave supports.

-SDR(0)	-SDR(1)	Decoded Streaming Rate
1	1	Basic Transfer Cycle
0	1	10 MHz maximum (100 ns minimum cycle)
1	0	Reserved
0	0	Reserved

Figure 7. Streaming Data Rates

-S0, -S1: -Status 0, -Status 1: These status signals are driven by the controlling master to indicate the start of a data transfer cycle and also define the type of data transfer. When used with M/-IO, memory read or write cycles are distinguished from I/O read or write cycles. These signals are latched by the slave, as required, using the leading edge of -CMD, or the leading or trailing edge of -ADL.

Data is transferred to or from the data bus based on -CMD and a latched decode of the address, the status lines (-S0 exclusive-OR -S1), and M/-IO.

Slaves must support a full decode of -S0 and -S1. The following table shows the states of M/-IO, -S0, and -S1 in decoding I/O and memory read and write commands.

M/-IO	-S0	-S1	Function
0	0	0	Reserved
0	0	1	I/O Write Command
0	1	0	I/O Read Command
0	1	1	Inactive
1	0	0	Reserved
1	0	1	Memory Write Command
1	1	0	Memory Read Command
1	1	1	Inactive

Figure 8. I/O and Memory Transfer Controls

An I/O write command instructs an I/O slave to receive the data from the data bus. An I/O read command instructs an I/O slave to drive its data onto the data bus.

A memory write command instructs a memory slave to receive the data from the data bus. A memory read command instructs a memory slave to drive its data onto the data bus.

-TC: -Terminal Count: This signal is driven by the DMA controller and provides a pulse during a read or write command to the DMA slave to indicate that the terminal count of the current DMA channel has been reached. This indicates to the DMA slave that this is the last cycle to be performed. -TC is driven active on the channel during DMA operations only.

TR 32: Translate 32: This signal is driven inactive by 32-bit controlling masters and received by the central translator logic. The signal indicates to the central translator logic that the controlling master is performing data steering. TR 32 can also be received by any 32-bit slave. See "Data Steering" on page 53 for information about the central translator logic.

Interrupt Signals

-IRQ 3 – 7, -IRQ 9 – 12, and -IRQ 14 – 15: -Interrupt Request: An interrupt request is generated when an I/O slave drives one of the 'interrupt request' signals low. These signals make up the set of interrupt signals. The polarity of these signals makes it possible for multiple slaves to concurrently share the same interrupt level.

Other Micro Channel Signals

AUDIO: Audio Sum Node: This analog signal is the sum of all the audio signals being driven. It is used to drive audio signals from an adapter to the system audio output and between adapters. The frequency response of the audio signal is 50 Hz to 10 kHz ± 3 dB. The maximum signal amplitude is 2.5 Vac peak-to-peak, at a dc offset of 0.0 Vdc ± 50 millivolts. The noise level is limited to a maximum of 50 millivolts peak-to-peak.

AUDIO GND: Audio Ground: This is the analog ground return signal for the audio subsystem.

-CD SETUP (n): -Card Setup: This signal is driven by system logic to individually select channel connectors. The (n) indicates this signal is unique to each channel connector (one independent signal per connector). When this signal is activated, a specific channel connector is selected, and, if an adapter is present, access is gained to configuration data at that connector. The adapter ID and configuration data is obtained by an I/O read during a setup cycle; the configuration data is stored by an I/O write during a setup cycle.

-CHCK: -Channel Check: This signal is driven active by a slave to report an exception condition, and optionally, it can be driven by a master.

CHRESET: Channel Reset: This signal is generated by the system logic to reset or initialize all adapters at power-on or when a low voltage condition is detected. The system can also activate this signal under program control.

OSC: Oscillator: This signal is a high-speed clock driven by the system logic, with a frequency of 14.31818 MHz ± 0.01%. The high-level pulse width (more than 2.3 Vdc) and the low-level pulse width (less than 0.8 Vdc) must not be less than 20 nanoseconds each.

-REFRESH: -Refresh: This signal is driven by the system logic and is used to indicate that a memory refresh operation is in progress. Memory slaves that do not need to perform refresh operations do not need to receive this signal.

Channel Signal Groups (Digital)

The following figure lists the digital Micro Channel signals and shows what type of driver or receiver is required. The 'audio' and 'audio ground' signals are analog signals. (For further information about these signals, see page 19.)

When a Micro Channel adapter combines the functions of bus masters and slaves, the signals driven and received by the adapter are a superset of the selected functions.

For example:

- An adapter that combines a slave function with a bus master function must drive and receive all signals required by a slave and all signals required by a bus master.
- An adapter that combines an I/O slave function with a memory slave or DMA slave function, must drive and receive all signals required by both slaves.

  Note: All adapters must support setup.

The central arbitration control point, central steering logic, and central translator logic are packaged within the system logic. The system logic also includes circuitry for memory refresh, the oscillator, and channel reset.

The signals for the 16-bit adapter are a subset of the signals for the 32-bit adapter.

Signal Name	System Master	DMA Cntlr	Bus Master	DMA Slave	Memory Slave	I/O Slave	Driver Type
	D/R	D/R	D/R	D/R	D/R	D/R	Signal Group
A(0-15)	D/-	D/-	D/-	-/0	-/R	-/R	TS (1)
A(16-23)	D/-	D/-	D/-	-/-	-/R	-/-	TS (1)
A(24-31)	*/-	*/-	*/-	-/-	-/*	-/-	TS (1)
-ADL	D/-	D/-	D/-	-/0	-/0	-/0	TS (1)
APAR(0-3)	0/-	0/-	0/-	-/0	-/0	-/0	TS (2)
-APAREN	0/-	0/-	0/-	-/0	-/0	-/0	TS (2)
ARB(0-3)	0/0	-/R	D/R	D/R	-/-	-/-	OC (5)
ARB/-GNT	D/-	-/R	-/R	-/R	-/-	-/-	BD (4)
-BE(0-3)	*/-	*/-	*/-	-/*	-/*	-/*	TS (1)
-BURST	0/R	D/R	0/-	0/-	-/0	-/0	OC (5)
CD CHRDY (n)	-/R	-/-	-/-	0/-	0/-	0/-	TP (3)
-CD DS 16 (n)	-/R	-/-	-/-	$/-	$/-	$/-	TP (3)
-CD DS 32 (n)	-/*	-/-	-/-	*/-	*/-	*/-	TP (3)
-CD SETUP (n)	D/-	-/@	-/@	-/@	-/@	-/@	TP (8)
-CD SFDBK (n)	-/R	-/-	-/-	D/-	D/-	D/-	TP (3)
-CHCK	-/R	0/R	0/R	0/-	0/-	0/-	OC (6)
CHRDYRTN	D/-	-/R	-/R	-/-	-/-	-/-	BD (4)
CHRESET	D/-	-/R	-/R	-/R	-/R	-/R	BD (4)
-CMD	D/-	D/-	D/-	-/R	-/R	-/R	TS (1)
D(0-7)	D/R	D/R	D/R	D/R	D/R	D/R	TS (2)
D(8-15)	D/R	D/R	D/R	$/$	$/$	$/$	TS (2)
D(16-31)	*/*	*/*	*/*	*/*	*/*	*/*	TS (2)
DPAR(0)	0/0	0/0	0/0	0/0	0/0	0/0	TS (2)
DPAR(1)	0/0	0/0	0/0	0/0	0/0	0/0	TS (2)
DPAR(2-3)	0/0	0/0	0/0	0/0	0/0	0/0	TS (2)
-DPAREN	0/0	0/0	0/0	0/0	0/0	0/0	TS (2)
-DS 16 RTN	D/-	-/R	-/R	-/-	-/-	-/-	BD (4)
-DS 32 RTN	*/-	-/*	-/*	-/-	-/-	-/-	BD (4)
-IRQ (N)	0/R	0/-	0/0	0/-	0/-	0/-	OC (6)
M/-IO	D/-	D/-	D/-	-/R	-/R	-/R	TS (1)
MADE 24	D/-	D/-	D/-	-/-	-/R	-/-	TS (1)
-MSDR	-/0	-/0	-/0	0/-	0/-	0/-	TS (1)
OSC	D/-	-/0	-/0	-/0	-/0	-/0	CD (7)
-PREEMPT	D/R	-/-	D/0	D/0	-/-	-/-	OC (5)
-REFRESH	D/-	-/-	-/-	-/-	-/0	-/-	TS (1)
-S0,-S1	D/*	D/-	D/-	-/R	-/R	-/R	TS (1)
-SBHE	D/*	D/-	D/-	-/#	-/#	-/#	TS (1)
-SFDBKRTN	0/-	-/0	-/0	-/-	-/-	-/-	BD (4)
-SD STROBE	0/-	0/-	0/-	-/0	-/0	-/0	TS (1)
-SDR(0)	-/0	-/0	-/0	0/-	0/-	0/-	TS (1)
-SDR(1)	-/0	-/0	-/0	0/-	0/-	0/-	TS (1)
-TC	-/-	D/-	-/-	-/R	-/-	-/-	TS (1)
TR 32	*/R	*/-	*/-	-/0	-/0	-/0	TS (1)

- KEY -

D = Drive Enabled
O = Optional
R = Receive Enabled
- = Not Implemented
CD = Clock Driver

OC = Open Collector
TS = Tri-State
TP = Totem Pole
BD = Bus Driver
N = IRQ(9-12,14,15,3-7)

= Required for 16-bit devices
$ = Required for 16- or 32-bit devices
* = Required for 32-bit devices
@ = -CD SETUP is received by all adapters

Figure 9. Driver and Receiver Requirements and Options

The following table and notes apply to the driver and receiver options listed on page 21.

Signal Group	Driver Type	Sinking Capacity	Capacitive Loading	Total Capacitance
1	Tri-State	24 mA	20 pF	240 pF
2	Tri-State	24 mA	20 pF	240 pF
3	Totem Pole	6 mA	20 pF	40 pF
4	Bus Driver	24 mA	20 pF	240 pF
5	Open Collector	24 mA	15 pF	200 pF
6	Open Collector	24 mA	20 pF	240 pF
7	Clock Driver	24 mA	15 pF	200 pF
8	Totem Pole or Tri-State	6 mA	50 pF	Not Applicable

Figure 10. Signal Groups, Driver Types, and Loading Specifications

Notes:

1. Pull-up resistors are provided by system logic for all signals other than -SDR0, -SDR1, -MSDR, and -SD STROBE. Adapters that receive these signals must provide a pull-up resistor for each signal received. (A minimum of 20 kΩ pulled to 5 Vdc is recommended.)
2. The maximum loading current is 1.6 mA per channel connector, except signal group 5. The maximum loading current of group 5 is 1.0 mA per channel connector.
3. An open collector can be either an open-collector device or a tri-state device wired with the input grounded and using the enable line of the tri-state device to control the output.
4. For good signal quality, the net length on the adapter for any control signal driven or received on the adapter should not exceed 2 inches.

Matched-Memory Extension Signal Descriptions

The matched-memory extension can, in some cases, improve the data transfer capabilities between the system master and channel-resident memory. The matched-memory signal-sequence and signal-timing specifications are system dependent. For more information, see the technical manuals for the systems that support the matched-memory signals.

Auxiliary Video Extension Signal Descriptions

| The auxiliary video extension allows optional video adapters to share
| video resources with the base video subsystem. The base video can
| be on the system board or on an adapter with the base video
| extension. When the base video subsystem is on the system board,
| support of the auxiliary video extension is direct. When the base
| video subsystem is on an adapter, support of the auxiliary video
| extension is through the base video extension.

| The signals and signal timings for the auxiliary and base video
| extensions are system dependent. For information on these signals,
| see the technical references for your specific system.

| The auxiliary video extension is shown in Figure 90 on page 149.
| The base video extension is shown in Figure 91 on page 150.

Micro Channel Architecture Base and Extensions

| Figure 11 on page 25 shows the buses and signals available for use
| with the 32-bit connector. The optional extensions signals are shown
| at the top of the figure and the optional connector pins at the bottom
| of the figure.

| Figure 12 on page 26 shows the buses and signals available for use
| with the 16-bit connector. The optional extensions signals are shown
| at the top of the figure and the optional connector pins at the bottom
| of the figure. The 16-bit connector is a subset of the 32-bit connector.

"Connector Definition" on page 142 contains the physical specifications for the connectors.

32-Bit Connector and Extensions

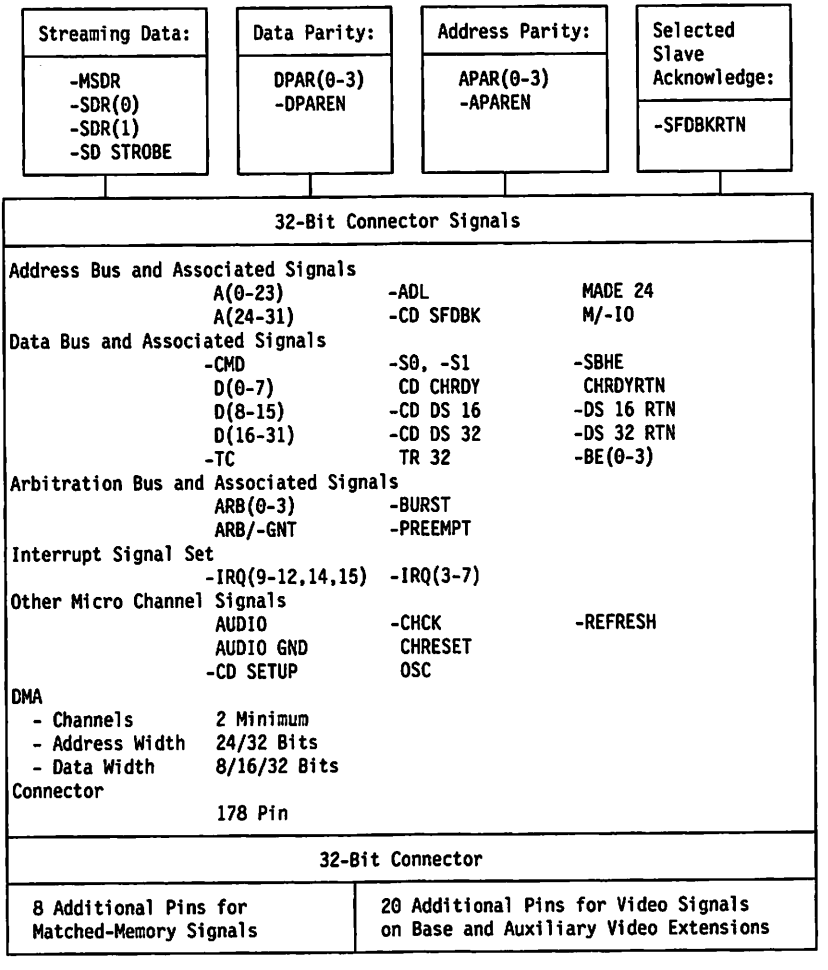


Figure 11. The 32-Bit Connector and the Optional Extensions

16-Bit Connector and Extensions

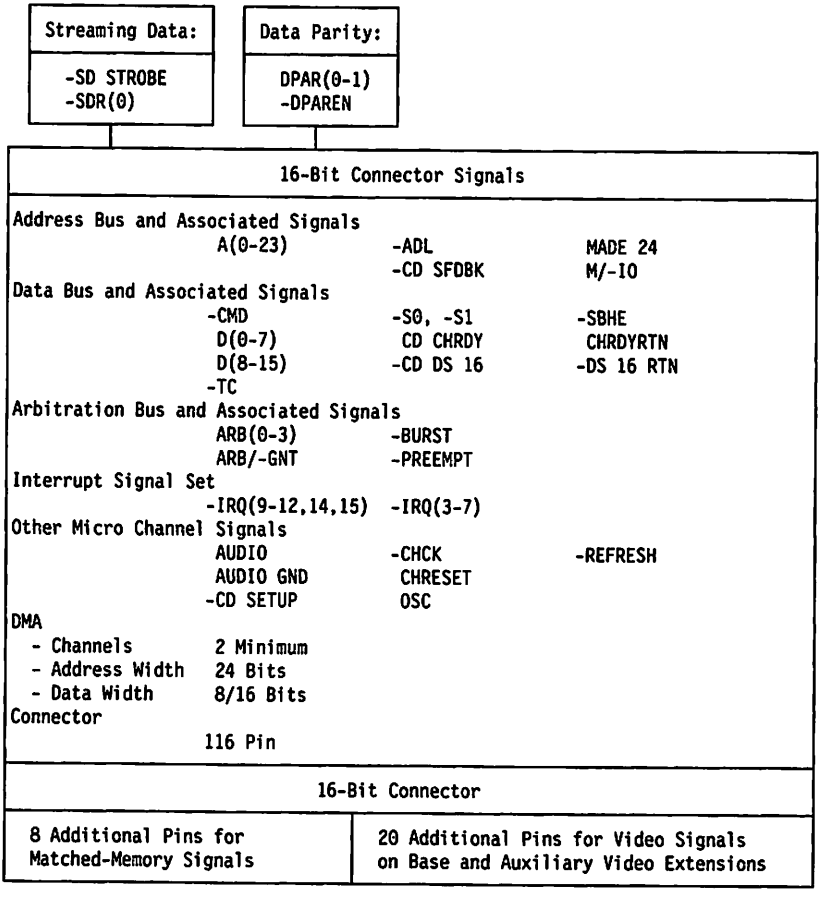


Figure 12. The 16-Bit Connector and the Optional Extensions

Micro Channel Procedures

The Micro Channel procedures, as well as data steering, are described in the following pages.

Arbitration

The arbitration procedure provides the functions to resolve multiple requests for control of the channel. Arbitration consists of local arbiters, the arbitration bus and the associated signals, and a central arbitration control point.

The arbitration bus is a parallel bus that consists of four lines ARB(0-3), supporting up to 16 arbitration levels. These arbitration levels are hex 0 through hex F; level hex 0 is the highest priority, level hex F is the lowest priority. The master with level hex F is always the default master.

If no -PREEMPT is driven, the arbitration level defaults to hex F and the default master gains control of the channel.

Channel Ownership

Channel ownership is controlled by the central arbitration control point based on prioritized arbitration of up to 16 devices. These arbitrating devices can be the system master, bus masters, or DMA slaves. If either a bus master or the system master wins the arbitration, it controls the bus and becomes the controlling master. If a DMA slave wins the arbitration, the supporting DMA controller becomes the controlling master.

If a system master is also the default master, the system master controls the Micro Channel when the channel is not controlled by any other master.

Central Arbitration Control Point

The central arbitration control point controls arbitration cycles using three arbitration signals: -PREEMPT, ARB/-GNT, and -BURST.

Arbitrating participants (local arbiters) drive -PREEMPT active to request use of the channel. An arbiter can activate -PREEMPT asynchronously. However, an arbiter must have -PREEMPT activated *before* the beginning of an arbitration cycle to participate in that cycle. If an arbiter drives -PREEMPT active when an arbitration cycle is in progress, it must wait for the *next* arbitration cycle before driving its arbitration level onto the arbitration bus.

The central arbitration control point initiates an arbitration cycle by driving ARB/-GNT to the ARB state. The requesting local arbiters then drive their 4-bit arbitration level onto the arbitration bus. When a local arbiter sees a more-significant bit low on the arbitration bus (other than those driven low by itself), it stops driving its less-significant bits onto the arbitration bus. The local arbiter driving the lowest arbitration level (highest priority) thereby wins control of the channel when ARB/-GNT goes to the -GNT state.

Arbitrating participants with multiple data transfers to perform signal the central arbitration control point by driving -BURST active. -BURST is held active until all transfers have been completed or until another participant drives -PREEMPT active. When preempted, the controlling master or DMA slave must release the channel and signal the end of transfer within 7.8 microseconds.

The end of transfer (EOT) is indicated by the trailing edge of -S0, -S1, -BURST, or -CMD, whichever occurs last. When the central arbitration control point detects an EOT condition, it drives ARB/-GNT to the ARB state, initiating the next arbitration cycle.

When ARB/-GNT is in the ARB state, bus masters and DMA controllers disable all of their channel drivers except -PREEMPT and ARB(0-3). Slaves, including bus masters and DMA controllers responding as slaves, must be capable of driving the channel when addressed.

When a master gains control of the channel, it is required to initiate a transfer cycle. If the master must abort the transfer cycle, it activates -S(0,1) with the minimum pulse width of T2A, and does *not* activate -ADL and -CMD (see Figure 24 on page 45 for the T2A timing specification).

The T2A time allows the central arbitration control point to recognize EOT and allows adapters to exit the inactive state.

During exception conditions, the central arbitration control point can gain control of the channel by driving ARB/-GNT to the ARB state independently of an EOT condition.

Note: While ARB/-GNT is in the ARB state, the system master can perform data transfers and system logic can perform refresh operations. Selected slaves must respond independently of the state of ARB/-GNT

The following is a block diagram of the interaction between the central arbitration control point and the local arbiters.

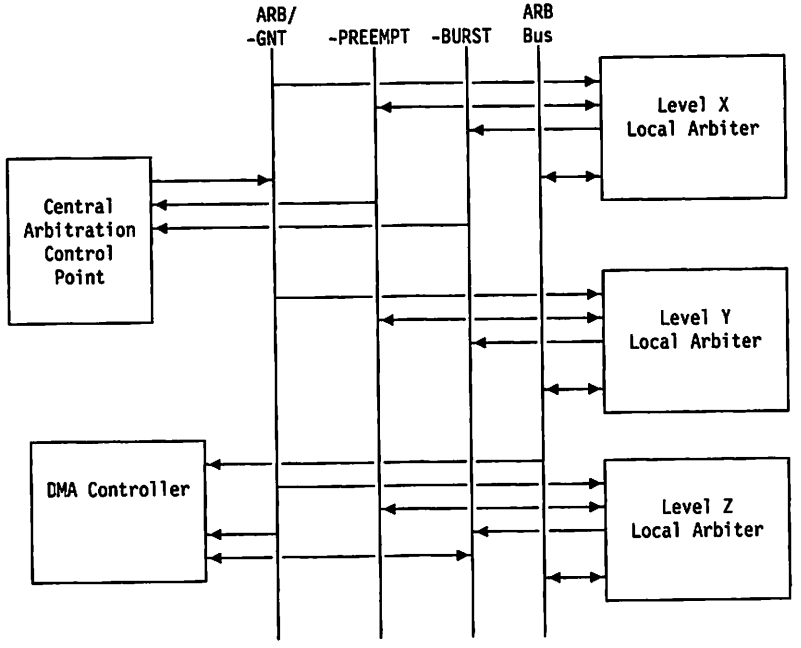


Figure 13. Arbitration Block Diagram

Local Arbiters

Participants requesting the use of the channel implement logic that allows all competing participants to recognize the winner. This logic is known as a local arbiter. An arbitrating participant competes for control of the channel only if it has driven -PREEMPT active before ARB/-GNT goes to the ARB state. A competing local arbiter drives its arbitration level onto the arbitration bus, then compares its arbitration level, on a bit-by-bit basis, with the value appearing on the arbitration bus beginning with the most significant bit, ARB3. If the competing local arbiter detects a mismatch on one of the bits, it immediately stops driving its less-significant bits. If the local arbiter subsequently recognizes a match on that bit, it resumes driving its less-significant bits until another mismatch is detected. The following is an example of an arbitration cycle.

1. Two local arbiters with arbitration levels 1010 and 0101 (hex A and 5) compete for the channel. Both local arbiters drive their arbitration levels on the bus, which now appears as 0000.
2. The first local arbiter (1010) detects a mismatch on ARB3 and stops driving all less-significant arbitration bits (ARB2 and ARB0 in this case).
3. The second local arbiter (0101) detects a mismatch on ARB2 and stops driving its less-significant arbitration bit (ARB1, in this case). The arbitration bus now shows 0111.
4. The second local arbiter now sees a match on ARB2 and resumes driving ARB1 of the arbitration bus.
5. The arbitration bus now shows a value of 0101, and the second local arbiter wins the arbitration cycle.

The following is a simplified example of a local arbiter using combinational logic. If ARB/-GNT goes to the ARB state, +BUS WON is not true.

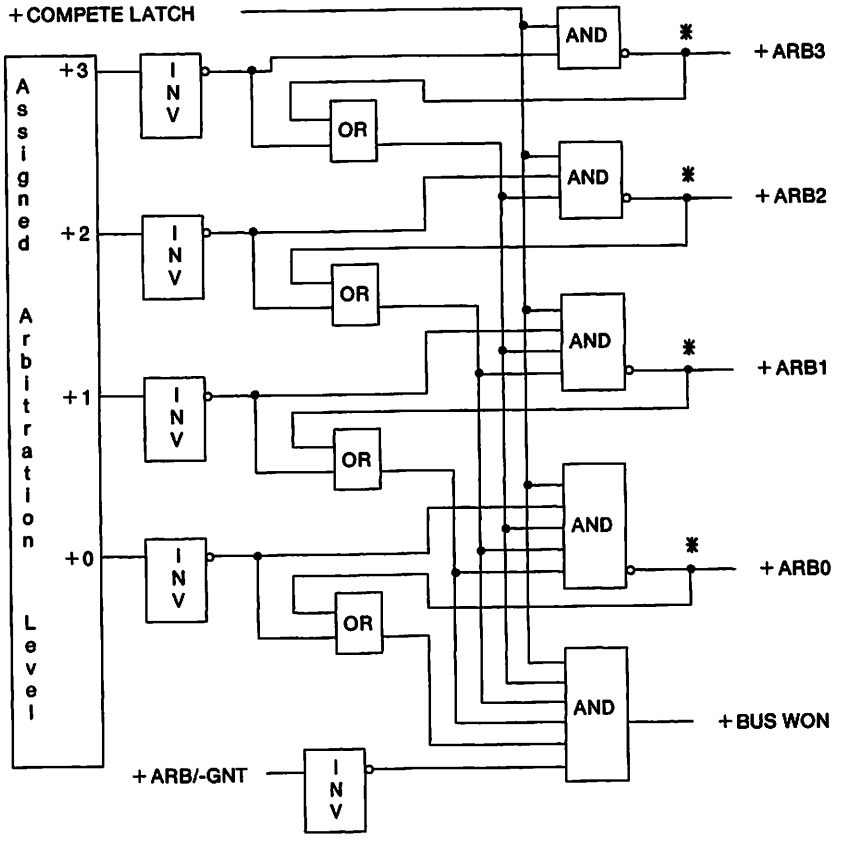


Figure 14. Local Arbiter Example

Burst Transfer

To use burst transfer, a controlling master or a DMA slave activates -BURST after being granted the channel. -BURST is held active until after the start of the last cycle of the burst transfer. The following diagram shows a burst transfer.

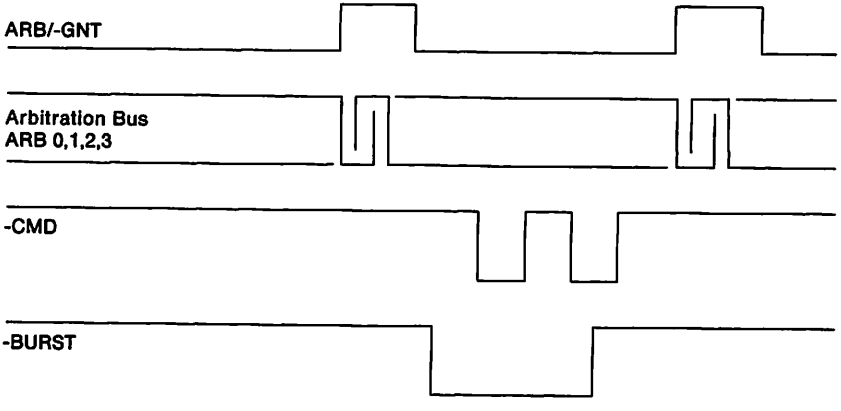


Figure 15. Burst Transfer

Preemption

When an arbitrating participant requires the channel, it activates -PREEMPT. The following diagram shows -PREEMPT occurring during a burst transfer.

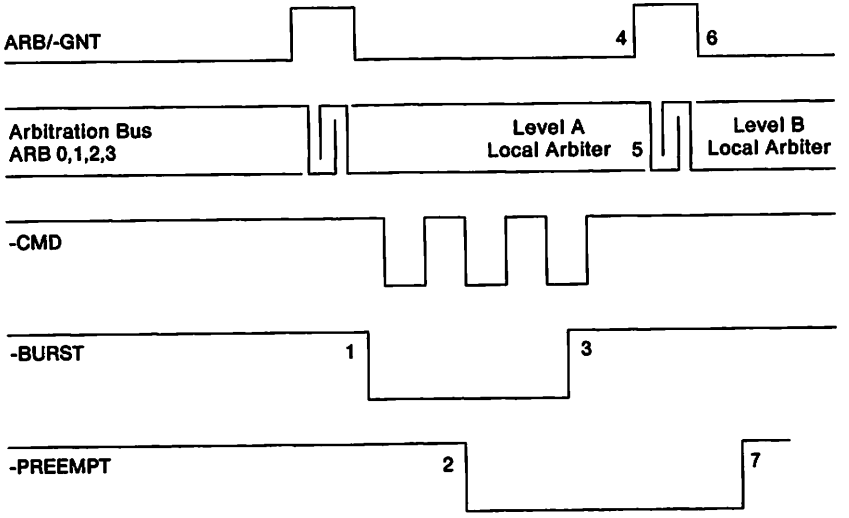


Figure 16. Preempted Burst Transfer

The preempted burst transfer sequence is as follows:

1. Participant A has control of the channel and has driven -BURST active.
2. Participant B drives -PREEMPT active.
3. Participant A, still in control of the channel, recognizes -PREEMPT active and signals the EOT condition within 7.8 microseconds. Participant A does not participate in the next arbitration cycle if its fairness feature is enabled. (See "Fairness Feature and the Inactive State" on page 34.)
4. When the central arbitration control point recognizes the EOT, it drives ARB/-GNT to the ARB state.
5. The arbitration cycle begins.
6. When ARB/-GNT is in the -GNT state, Participant B gains control of the channel.
7. Participant B, the preempting participant, removes -PREEMPT in response to gaining control of the channel.

Arbitrating masters should maximize use of the channel within the 7.8 microsecond limit from -PREEMPT going active. If a master or a DMA slave *does not* release the channel by presenting EOT within 7.8 microseconds after -PREEMPT is driven active, a channel time-out can occur, creating an exception condition.

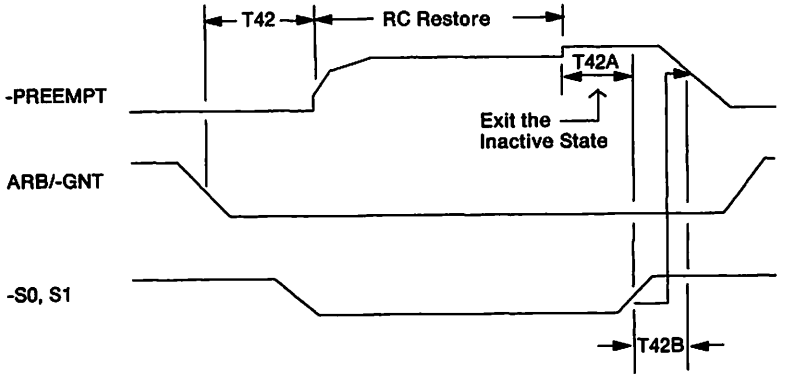
Fairness Feature and the Inactive State

| All bursting participants must support the fairness feature. Fairness
| forces the bursting participant to equitably share the channel time
| with the other participants. An optional configuration field in
| programmable option select (POS) can be used to disable the
| fairness feature.

When a bursting participant with control of the channel senses an active -PREEMPT, it must release the channel within the specified time (even if the fairness feature is disabled). If its fairness feature is enabled, the participant must enter the *inactive state* unless -PREEMPT was inactive while that participant owned the channel. If -PREEMPT was inactive while that participant had control, it is not required to enter the inactive state but must release the channel. In the inactive state, the bursting participant will not drive -PREEMPT active or participate in arbitration.

To exit the inactive state, the arbitrating participant must wait for
| -PREEMPT to go inactive. The participant then waits for a specified
| time (T42B) after the trailing edge of status (status active to inactive)
| before driving -PREEMPT active.

The following figure shows exiting from inactive state.



| Figure 17. Exiting from the Inactive State

Note: The participant exits the inactive state independently of the state of ARB/-GNT.

Arbitration Timing

The following figures show the signal sequences and the timing specifications of the arbitration procedure.

After ARB/-GNT goes to the -GNT state and until EOT, the data transfer procedures are used to transfer data. See "Basic Transfer" on page 40, "Streaming Data Transfer" on page 58, and "DMA Procedure" on page 84 for the signal sequences and timing specifications.

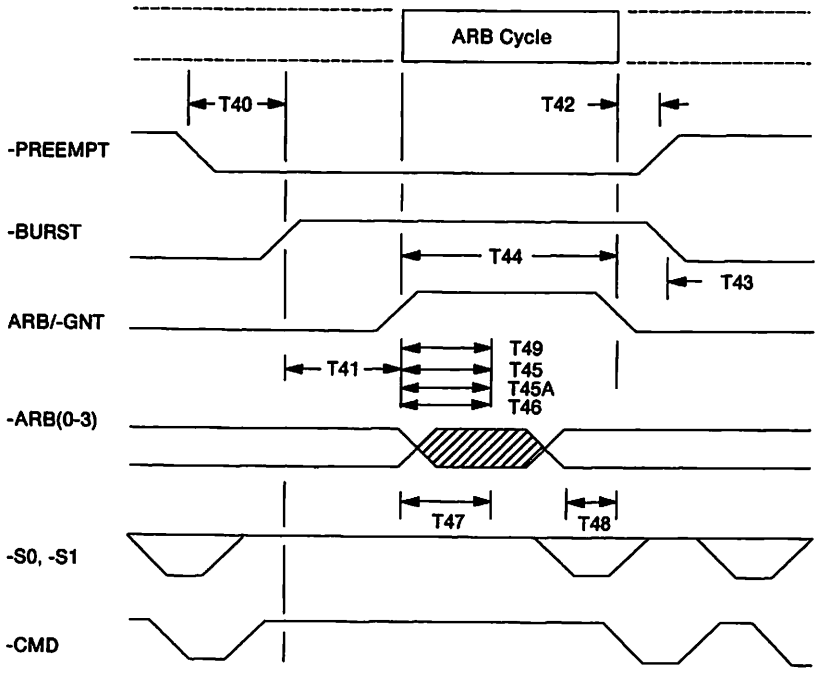


Figure 18. Arbitration Cycle

	Timing Parameter	Min/Max	Note
T40	-PREEMPT active to EOT (channel release)	0 / 7.8 μs	1
T41	ARB/-GNT in the ARB state from EOT (channel release)	30 / - ns	6
T42	-PREEMPT inactive from ARB/-GNT in -GNT state	0 / 50 ns	8
T42A	-PREEMPT inactive to status inactive	20 / - ns	5
T42B	-PREEMPT active from status inactive (Compete for the channel after the inactive state)	10 / - ns	
T43	-BURST active from ARB/-GNT in -GNT state	- / 50 ns	4
T44	ARB/-GNT in ARB state	100 / - ns	2
T45	Driver turn-on delay from ARB/-GNT in ARB state	0 / 50 ns	3
T45A	Driver turn-on delay from lower priority line	0 / 50 ns	3
T46	Driver turn-off delay from ARB/-GNT in ARB state	0 / 50 ns	3
T47	Driver turn-off delay from higher priority line	0 / 50 ns	3
T48	Arbitration bus stable before ARB/-GNT in -GNT state	10 / - ns	
T49	Tri-state drivers from ARB/-GNT in ARB state	- / 50 ns	7

Figure 19. Arbitration Cycle Timing Specifications

Notes:

1. This parameter applies to all masters and DMA slaves driving -BURST active. This parameter ensures that time-critical processes (such as refresh) can gain control of the channel resources. The value of the parameter is a requirement for the adapter designer and not a channel timeout requirement.
2. The value shown applies only to special case implementation involving the central arbitration control point and is provided for pulse width consideration only.
3. T45, T45A, T46 and T47 must be satisfied by the ARB(0-3) drivers of all arbitrating participants.
4. This parameter applies to all DMA slaves and bus masters when they are granted control of the channel.
5. This parameter is met indirectly by meeting T43D and T42 and includes the RC restore time on the bus, assuming no other participant is driving -PREEMPT active during this time. This delay allows participants to detect inactive -PREEMPT and subsequently exit the inactive state.
6. Because no maximum is specified, each non-default master must maintain the EOT on the channel until ARB/-GNT is in the ARB state.
7. This timing applies only to the controlling master as it is releasing control of the channel.
8. The RC restore of -PREEMPT will only occur at this time if no other participant is driving -PREEMPT active. If the system master performs cycles when ARB/-GNT is in the ARB state, -PREEMPT might be released while ARB/-GNT is in the ARB state.

The following are the timing parameters for the first cycle after ARB/-GNT goes to the -GNT state.

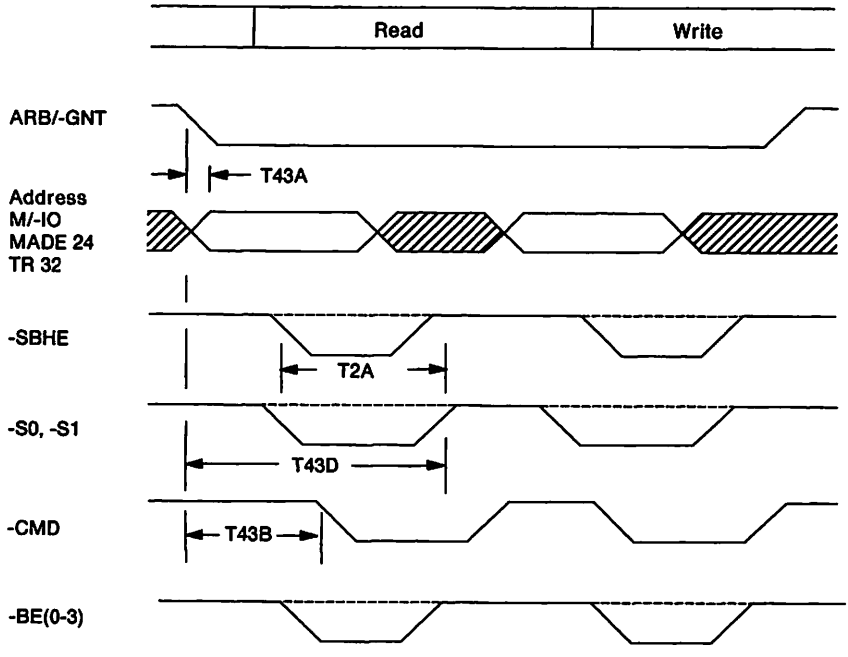


Figure 20. First Cycle After Grant

	Timing Parameter	Min/Max	Note
T2A	Status pulse width of an aborted cycle	85 / - ns	
T43A	Address bus valid from ARB/-GNT in the -GNT state	0 / - ns	
T43B	-CMD active from ARB/-GNT in the -GNT state	115 / - ns	
T43C	DMA controller delay	30 / - ns	1
T43D	Status inactive from ARB/-GNT in the -GNT state	145 / - ns	

Figure 21. First-Cycle-After-Grant Timing Specifications

Notes:

1. A DMA controller must allow a minimum time (T43C) after the grant for a slave to generate an internal acknowledgment that it has been selected. During the first cycle, the DMA controller must allow this additional time before sampling -DS 16/32 RTN and CHRDYRTN if it places an address on the address bus within 30 nanoseconds after grant. However, if the DMA controller drives the address 30 nanoseconds after grant, the additional delay is not needed.

Data Transfer

The data transfer procedures are used to transfer data between a controlling master and the selected slave. The three types of data transfer procedures are:

- Basic Transfer
- Streaming Data Transfer
- DMA Transfer.

Each procedure defines the signal sequences and the signal timing specifications used in the procedure. Data transfer occurs only during the data transfer cycle defined by the procedure. A cycle is one or more data transfers, occurring during the time between the leading edge of -CMD and the trailing edge of -CMD. The data transfer cycle address selects either an I/O-address-space address or a memory-address-space address, under the control of M/-IO.

A controlling master or a DMA slave can drive -BURST, to indicate to the central arbitration control point that one or more data transfer cycles will be used before the EOT occurs. Refer to "Burst Transfer" on page 32 for more information.

This section describes each procedure, the signal sequence for each procedure, and the signal timing specifications.

The system-specific technical manuals contain the signal sequence and timing specifications for matched-memory signals and auxiliary-video signals.

Exception condition handling and reporting descriptions are contained in "Exception Condition Reporting and Handling" on page 118.

Basic Transfer

The three types of basic transfer cycles are:

- Default Cycle
- Synchronous Extended Cycle
- Asynchronous Extended Cycle.

Basic Transfer Procedure

All masters and DMA controllers transfer data with the same control sequence. The following covers the case of a write cycle. The signals appear in the following sequence:

1. The master drives the address bus, MADE 24, M/-IO, -APAREN, APAR(0-3) and -REFRESH (if applicable) valid, beginning the cycle.
2. In response to an unlatched decode of address, MADE 24, and M/-IO, the selected slave returns:
   - -CD SFDBK
   - -CD DS 16 (if the addressed data port is capable of 16-bit operations)
   - -CD DS 16 and -CD DS 32 (if the addressed data port is capable of 32-bit operations)
3. The master drives -S0 active (-S1 inactive).
4. In response to an unlatched decode of address, MADE 24, M/-IO, and status, the selected slave drives CD CHRDY inactive if the cycle is to be extended.
5. The master drives -ADL active. A slave can latch decodes of address, status, and M/-IO at the leading or trailing edge of -ADL. A DMA slave can also be selected by a latched decode of the same signals, using the arbitration level in place of the address.
6. The master drives the write data onto the data bus and, if appropriate, drives -DPAREN active and DPAR(0-3) valid.
7. The master drives -CMD active and -ADL inactive. A slave must latch the decodes of address, status, and M/-IO with the leading edge of -CMD if they were not latched with -ADL.

8. If CD CHRDY has been driven inactive, the cycle is extended until CD CHRDY is driven active. The signal must not be held inactive longer than specified.
9. The status signals become inactive and the address bus can change in preparation for the next cycle.
10. In response to an address change, the slave updates the state of:
    - -CD SFDBK
    - -CD DS 16
    - -CD DS 32.
11. The status signals and M/-IO can become valid in preparation for the next cycle.
12. -CMD goes inactive and the selected slave receives the data from the data bus. The data transfer cycle is complete.

Note: The changes to the address bus and status (-S0, -S1) can be overlapped with the preceding cycle to minimize the impact of the memory access time.

The sequence for the basic transfer procedure is as follows.

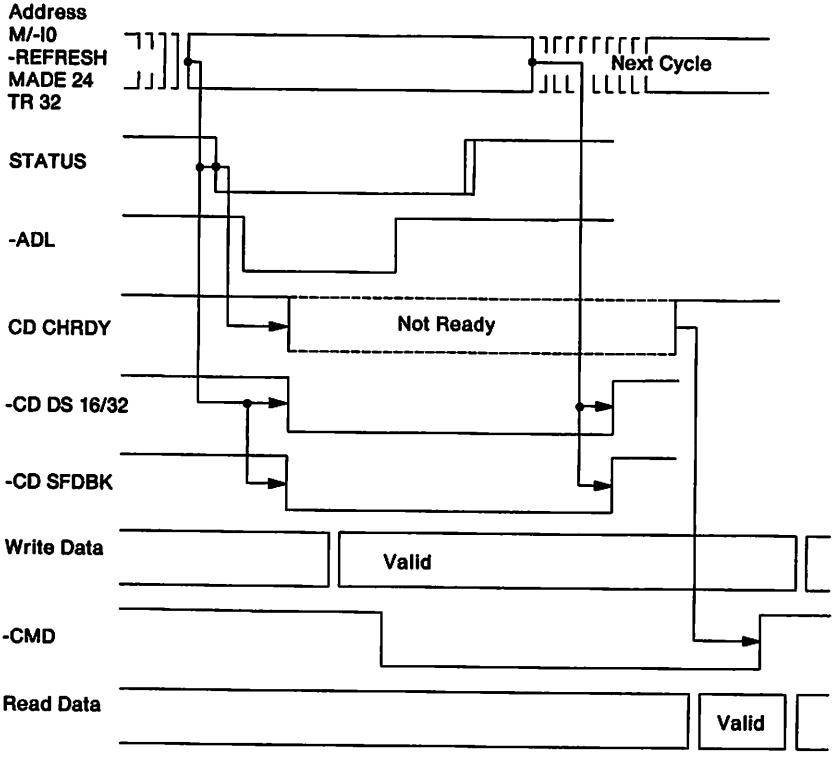


Figure 22. Basic Transfer Procedure - Overview

Whether a default, a synchronous-extended, or an asynchronous-extended cycle is performed depends on how a slave uses CD CHRDY.

- A default cycle occurs when a slave does not drive CD CHRDY inactive.
- A synchronous-extended cycle occurs when a slave drives CD CHRDY inactive, then releases CD CHRDY synchronously within the specified time after the leading edge of -CMD. The slave provides the read data within the specified time from -CMD active.
- An asynchronous-extended cycle occurs when a slave drives the CD CHRDY inactive, then releases CD CHRDY asynchronously.

If a master begins a transfer cycle but must abort that cycle, -ADL and -CMD must *not* be activated, and -S0, -S1 must be activated with a minimum pulse width T2A (see Figure 24 on page 45 for the T2A timing specification). -BURST, if active, is deactivated with status deactivation. Slaves must not latch CD CHRDY inactive until after the appearance of an active -ADL signal, and if the cycle is aborted, must release CD CHRDY with the deactivation of status. The selected slave degates the data bus and data transfer control signals when it is no longer selected (status and -CMD are inactive).

Default Cycle Timing (200 ns Minimum)

This section provides the specifications for the timing parameters for the I/O and memory default transfer cycle.

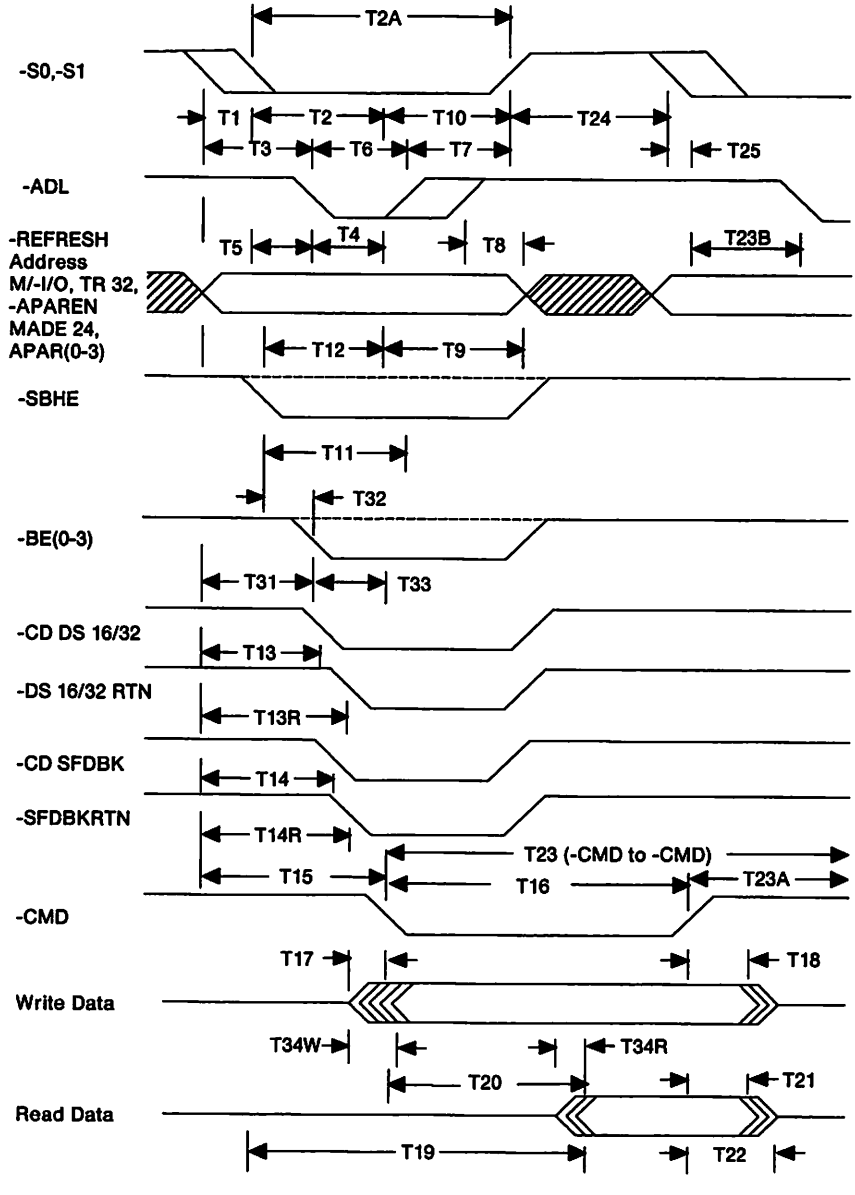


Figure 23. Default Cycle (200 ns Minimum)

	Timing Parameter	Min/Max	Note
T1	Status active from Address, M/-IO, -APAREN, -REFRESH, MADE 24, TR 32, and APAR(0-3) valid	10 / - ns	13
T2	-CMD active from status active	55 / - ns	4
T2A	Status pulse width of an aborted cycle	85 / - ns	10
T3	-ADL active from Address, M/-IO, -APAREN, -REFRESH, MADE 24, TR 32, and APAR(0-3) valid	45 / - ns	
T4	-ADL active to -CMD active	40 / - ns	
T5	-ADL active from status active	12 / - ns	
T6	-ADL pulse width	40 / - ns	
T7	Status hold from -ADL inactive	25 / - ns	4,11
T8	Address, M/-IO, -REFRESH, -SBHE, -BE(0-3), MADE 24, TR 32, APAR(0-3), -APAREN hold from -ADL inactive	25 / - ns	4,13
T9	Address, M/-IO, -REFRESH, -SBHE, -BE(0-3), MADE 24, TR 32, APAR(0-3), -APAREN hold from -CMD active	30 / - ns	2,3,4, 13
T10	Status hold from -CMD active	30 / - ns	4,11
T11	-SBHE setup to -ADL inactive	40 / - ns	4
T12	-SBHE setup to -CMD active	40 / - ns	4
T13	-CD DS 16/32 active from Address Bus and M/-IO valid	- / 55 ns	2,6
T13R	-DS 16/32 RTN active from Address Bus and M/-IO valid	- / 75 ns	2
T14	-CD SFDBK active from Address Bus and M/-IO valid	- / 60 ns	1,2
T14R	-SFDBKRTN active from Address Bus and M/-IO valid	- / 80 ns	2
T15	-CMD active from Address valid	85 / - ns	4
T16	-CMD pulse width	90 / - ns	
T17	Write Data setup to -CMD active	0 / - ns	5,12
T18	Write Data hold from -CMD inactive	30 / - ns	5
T19	Status active to Read Data valid (Access Time)	- / 125 ns	5,8
T20	Read Data valid from -CMD active	- / 60 ns	5
T21	Read Data hold from -CMD inactive	0 / - ns	5
T22	Read data bus tri-state from -CMD inactive	- / 40 ns	5
T23	-CMD active to next -CMD active	190 / - ns	
T23A	-CMD inactive to next -CMD active	80 / - ns	
T23B	-CMD inactive to next -ADL active	40 / - ns	
T24	Next status active from status inactive	30 / - ns	
T25	Next status active to -CMD inactive	- / 20 ns	
T31	-BE(0-3) active from Address valid (32-bit masters only)	- / 40 ns	
T32	-BE(0-3) active from -SBHE, A0, A1 active	- / 30 ns	
T33	-BE(0-3) active to -CMD active	10 / - ns	
T34R	Centralized steering logic Read Data buffer delay	0 / 15 ns	9
T34W	Centralized steering logic Write Data buffer delay	0 / 15 ns	9

Figure 24. Default Cycle (200 ns Minimum) Timing Specifications

Notes:

1. Slaves drive -CD SFDBK when selected by either arbitration or address. The slave does not drive -CD SFDBK active when selected by -CD SETUP.
2. The master must hold the address bus and associated signals valid until -DS 16 RTN, -DS 32 RTN, and -SFDBKRTN are sampled, in addition to meeting T8 and T9. The slave must continue to drive -CD DS 16/32 and -CD SFDBK as long as its address is valid.
3. A master can deactivate status and address concurrently. This can result in changes on the address bus while status is still active at the slave. CD CHRDY can change states during changes on the address bus.
4. It is recommended that slaves use transparent latches to latch information with the leading edge of -CMD, or with the leading or trailing edge of -ADL.
5. When data parity is supported, the implemented data parity signals *must be* active and stable on the bus for the data to be considered valid.
6. -CD DS 16/32 and -CD SFDBK must be driven by unlatched address decodes because the next address can be driven valid in the current cycle.
7. When address parity is supported, the implemented address parity signals *must be* active and stable for the address to be considered valid.
8. T19 is a valid timing parameter *only* when T2 is less than 65 nanoseconds.
9. The centralized steering delay applies when a 16-bit master accesses a 32-bit data port. The slave must compensate for the centralized steering delay, as required.
10. Masters must provide this minimum timing for the central arbitration control point and for inactive state exit. Other participants must not depend on this minimum during an abort cycle. (See "Central Arbitration Control Point" on page 28 for more information.)
11. A bus master must deactivate -S(0,1) as soon as possible after the hold time specified by T7 and T10, and prior to the deactivation of

-CMD. Additional rules apply for extended cycles and streaming data cycles.

12. If data parity is used, see T96 timing in Figure 71 on page 123.
13. TR 32 can be held inactive between cycles by a 32-bit master. TR 32 timings are provided to define when it must be valid on the first and last cycle.

Synchronous-Extended Cycle Timing (300 ns Minimum)

A synchronous-extended cycle occurs when a slave drives CD CHRDY inactive, then releases CD CHRDY synchronously within the specified time after the leading edge of -CMD.

The timing sequence is shown in the following figure.

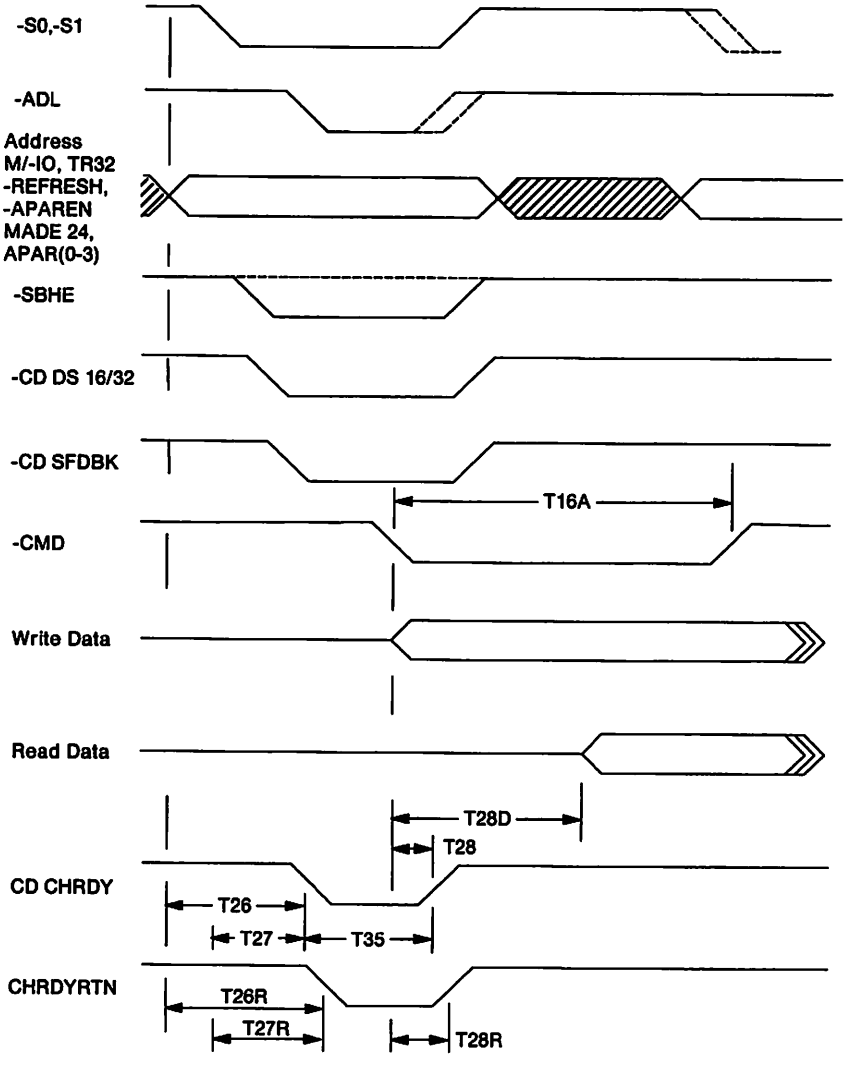


Figure 25. Synchronous-Extended Cycle (300 ns Minimum)

	Timing Parameter	Min/Max	Note
T16A	-CMD pulse width	190 / - ns	
T26	CD CHRDY inactive from Address Bus valid	- / 60 ns	3
T26R	CHRDYRTN inactive from Address Bus valid	- / 80 ns	3
T27	CD CHRDY inactive from status valid and previous -CMD inactive	0 / 30 ns	3,4
T27R	CHRDYRTN inactive from status valid and previous -CMD inactive	0 / 50 ns	3,4
T28	CD CHRDY active from -CMD active	0 / 30 ns	1
T28R	CHRDYRTN active from -CMD active	0 / 50 ns	
T28D	Read Data valid from -CMD active (used with T28)	0 / 160 ns	
T35	CD CHRDY active from CD CHRDY inactive	0 / 3.5μs	5

This table shows only the parameters that are additional to those of the default cycle. All other parameters are the same as the default cycle.

Figure 26. Synchronous-Extended Cycle Timing Specifications (300 ns Minimum)

Notes:

1. CD CHRDY is released by a slave performing a 300 nanoseconds extended cycle within the specified time after the leading edge of -CMD.
2. A master can deactivate status and address concurrently. This can result in changes on the address bus while status is still active at the slave. CD CHRDY can change states during changes on the address bus.
3. T27 and T27R apply *only* when T1 is greater than 30 nanoseconds. T26 and T26R applies *only* when T1 is less than or equal to 30 nanoseconds.
4. Masters are responsible for controlling their own bus activity. When a master uses overlapped cycles, it *must* be capable of handling (or ignoring) CD CHRDY from the next slave while terminating the cycle with the current slave. Once the current cycle is terminated, the master must act appropriately to an inactive CD CHRDY signal. Masters must also recognize and handle any added delay in a slaves deactivation of CD CHRDY caused by overlapped cycles.
5. A bus master must deactivate -S0,-S1 as specified in the default cycle, and this deactivation must be independent of CHRDYRTN. Slaves should not depend on the removal of -S0,-S1 for the deactivation of CD CHRDY.

6. When address parity is supported, the address parity signals (APAR 0-3) *must* be active and stable for the address to be considered valid.

Asynchronous-Extended Cycle Timing (≥300 ns)

An asynchronous-extended cycle occurs when a slave drives CD CHRDY inactive, then releases CD CHRDY asynchronously. The timing sequence is shown in the following figures.

	Timing Parameter	Min/Max	Note
T16A	-CMD pulse width	190 / - ns	
T26	CD CHRDY inactive from Address Bus valid	- / 60 ns	3
T26R	CHRDYRTN inactive from Address Bus valid	- / 80 ns	3
T27	CD CHRDY inactive from status valid and previous -CMD inactive	0 / 30 ns	3,4
T27R	CHRDYRTN inactive from status valid and previous -CMD inactive	0 / 50 ns	3,4
T29S	Read Data from slave valid from CD CHRDY active	- / 60 ns	1
T29M	Read Data valid to master from CHRDYRTN active	- / 60 ns	1
T29A	-CMD inactive from CHRDYRTN active	60 / - ns	
T35	CD CHRDY active from CD CHRDY inactive	0 /3.5μs	5

This table shows only the parameters that are additional to those of the default cycle. All other parameters are the same as the default cycle.

Figure 27. Asynchronous-Extended Cycle Timing Specifications (≥300 ns)

Notes:

1. CD CHRDY is released asynchronously by a slave performing a ≥300-nanosecond cycle. The slave must present the read data within the time specified after the release of CD CHRDY.
2. A master can deactivate status and address concurrently. This can result in changes on the address bus while status is still active at the slave. CD CHRDY can change states during changes on the address bus.
3. T27 and T27R apply *only* when T1 is greater than 30 nanoseconds. T26 and T26R apply *only* when T1 is less than or equal to 30 nanoseconds.

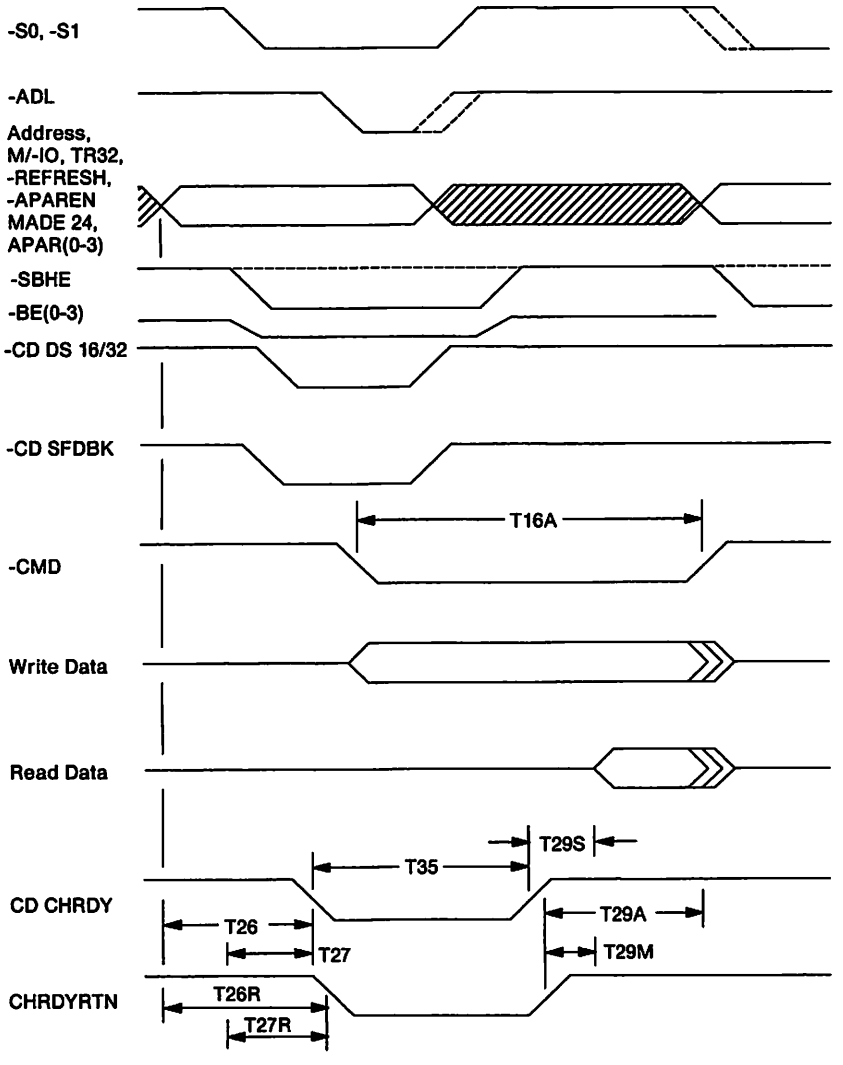


Figure 28. Asynchronous Extended Cycle (≥ 300 ns)

4. Masters are responsible for controlling their own bus activity. When a master uses overlapped cycles, it *must* be capable of handling (or ignoring) CD CHRDY from the next slave while terminating the cycle with the current slave. Once the current cycle is terminated, the master must act appropriately to an inactive CD CHRDY signal. Masters must also recognize and handle the potential added delay in a slave's deactivation of CD CHRDY due to overlapped cycles.
5. A bus master must deactivate -S0, -S1 as specified in the default cycle, and this deactivation must be independent of CHRDYRTN. Slaves should not have dependencies on removal of -S0, -S1 for the deactivation of CD CHRDY.
6. When address parity is supported, the implemented address parity signals (APAR 0-3) *must* be active and stable for the address to be considered valid.

Data Steering

During data transfer procedures, data can be transferred between a 16-bit or 32-bit master, and an 8-, 16-, or 32-bit slave. The following table shows the relationship between the data bus, associated signals, and the number of cycles required for the combinations of data transfer and steering.

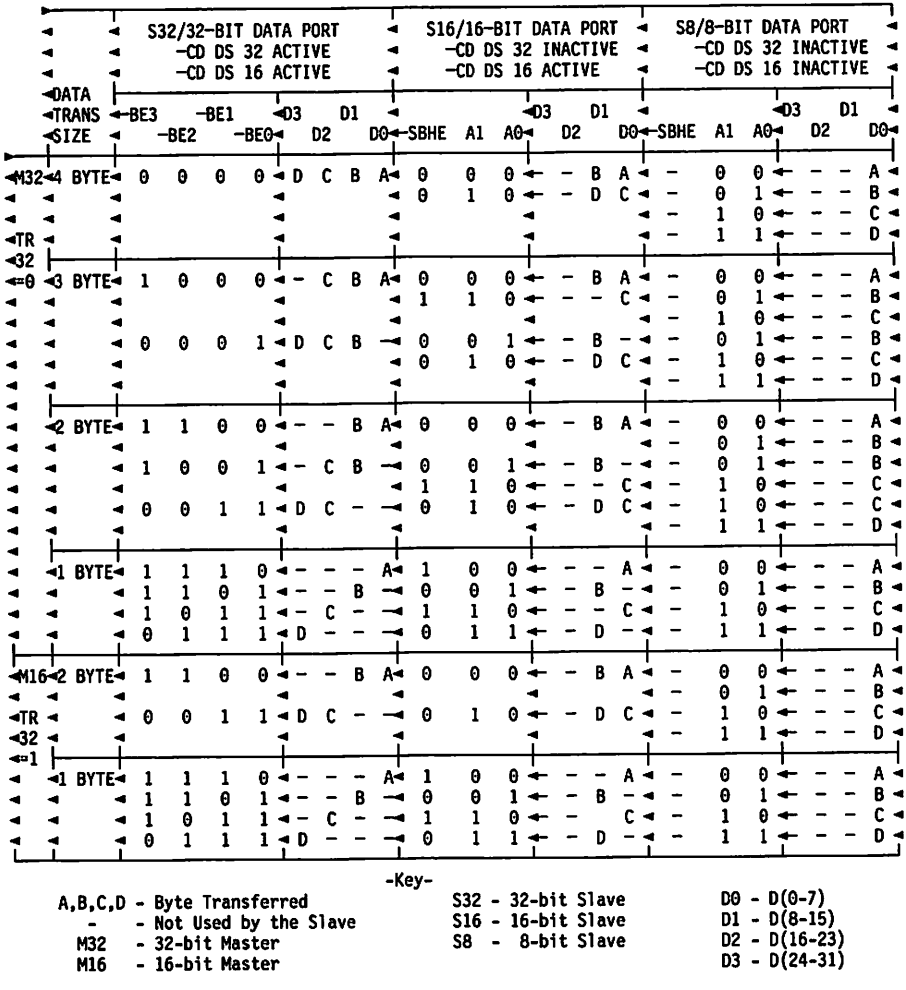


Figure 29. Steering Control

Notes:

1. Masters are responsible for generating and controlling the extra cycles needed to complete the transfer.
2. If TR 32 = 1, the central translator logic drives -BE(0-3) signals. The central steering logic steers data bus byte D(0-7) to or from data bus byte D(16-23) and data bus byte D(8-15) to or from data bus byte D(24-31).
3. If TR 32 = 0, the 32-bit master (M32) drives -BE(0-3) signals.
4. A -BE0, -BE1, -BE2, or -BE3 signal equal to 0 indicates a valid data byte in the corresponding data bus byte D(0-7), D(8-15), D(16-23), or D(24-31).
5. A 1 denotes the most-positive logic state of the signal. A 0 denotes the least-positive logic state of the signal.
6. Combinations of -SBHE, A1, A0 not shown in the table are not permitted.

Data Steering Control

The following table shows which participant is responsible for data-steering control during basic transfer, DMA, and setup procedures.

	S32	S16	S8
M16	Central Steering	No Steering	M16
M32	No Steering	M32	M32

M32 - 32-bit Master
M16 - 16-bit Master

S32 - 32-bit Slave
S16 - 16-bit Slave
S8 - 8-bit Slave

Figure 30. Steering Responsibility (Basic Transfer, DMA, and Setup Procedures)

The following table shows which participant is responsible for data-steering control during streaming data procedures.

	S32	S16
M16	S32	No Steering
M32	No Steering	M32
	M32 - 32-bit Master M16 - 16-bit Master	S32 - 32-bit Slave S16 - 16-bit Slave

Figure 31. Steering Responsibility (Streaming Data Procedure)

Master Replication of Data Bytes

The following are the requirements for master replication of data bytes during a write operation:

- A 16 or 32-bit master transferring one byte of data on D(8-15) replicates D(8-15) on D(0-7).
- A 32-bit master transferring two bytes of data on D(8-15) and D(16-23) or three bytes of data on D(8-15), D(16-23), and D(24-31) replicates D(8-15) on D(0-7).
- A 32-bit master transferring one byte of data on D(16-23) replicates D(16-23) on D(0-7).
- A 32-bit master transferring one byte of data on D(24-31) replicates D(24-31) on D(0-7) and D(8-15).
- A 32-bit master transferring two bytes of data on D(16-23) and D(24-31) replicates D(16-23) on D(0-7) and replicates D(24-31) on D(8-15).

Channel Support

The 32-bit data bus requires unique system logic to permit 16-bit masters to communicate with slaves with 32-bit data ports.

Address Bus Translator

A 32-bit slave uses signals -BE(0-3) as part of its address instead of A0, A1, and -SBHE. A 16-bit master does not provide -BE(0-3); the central translation logic generates them when a 16-bit master has control of the channel.

Central Data Steering

A 32-bit slave writes data to and reads data from data bits 0 through 31. A 16-bit master does not use data bits 16 through 31; the central steering logic must cross data over from the low 16 data lines (D0 through D15) to the high data lines (D16 through D31) and back at the appropriate times. The 32-bit slave must compensate for the added delay.

TR 32

TR 32 is driven inactive by 32-bit masters only. When TR 32 is active, it is used by:

- The central translator logic to drive -BE(0-3)
- The central steering logic to perform data steering.

When TR 32 is active, 32-bit slaves can use it to detect that the controlling master is not a 32-bit master and compensate for additional delay attributable to the central steering logic.

Central Steering Logic

Central steering logic uses A1, -SBHE, -S(0,1), -CD DS 16, and -CD DS 32 to steer data in support of 16-bit masters communicating with 32-bit slaves.

Central Translator Logic

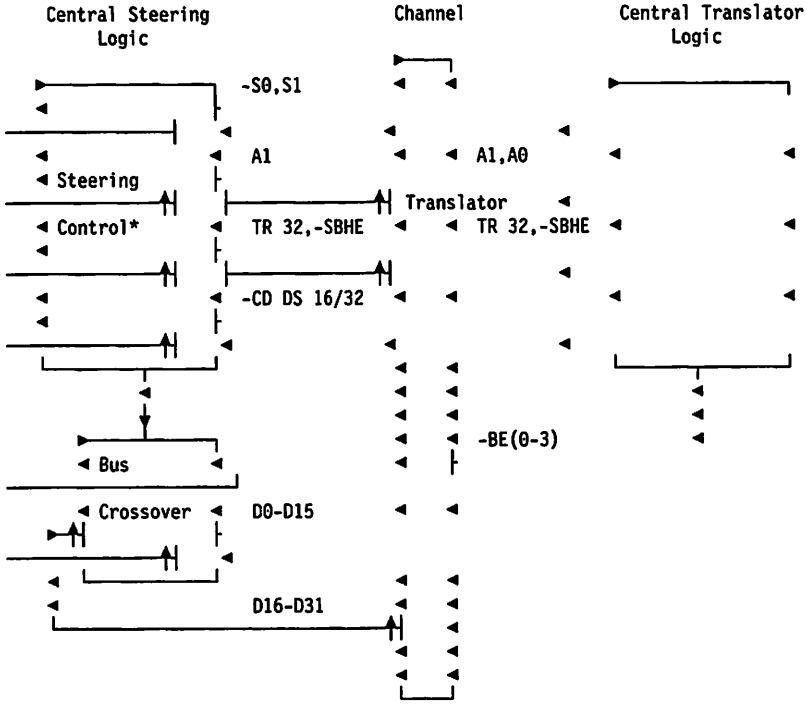
Central translator logic translates A0, A1, and -SBHE to -BE(0-3), when TR 32 is active to support transfers between 16-bit masters and 32-bit slaves.

-BE(0-3)

Signals -BE(0-3) are:

- Driven by a 32-bit controlling master.
- Created by the central translator logic when a 16-bit master is the controlling master.
- Used by 32-bit slaves only.

The following block diagram shows the implementation of data steering.



* For 16-bit devices to 32-bit devices

Figure 32. Data Bus Steering Implementation

Streaming Data Transfer

The streaming data procedure provides performance improvements over basic transfer procedures for block transfers, and supports data-transfer rates of up to 80MB per second. It provides for the transfer of a data block by using a single address followed by multiple 16-, 32-, or 64-bit data transfers within a single streaming data cycle. The data transfers are clock-synchronous and incorporate automatic speed matching between the controlling master and slave.

The streaming data procedure can be used for high speed data transfer between a controlling master and the selected slave. The streaming data procedure is transparent to devices that are not selected. Streaming data participants **must** support the basic transfer procedure to operate with nonstreaming-data participants.

Special rules apply to streaming data devices to ensure compatibility with nonstreaming-data participants. These include:

- A streaming-data-enable bit must be supported in the adapter POS space.

  Note: Streaming data participants must support a fully-functional operation with nonstreaming-data participants, independent of the state of the Streaming Data Enable field.

- After CHRESET, the default state of the streaming-data-enable bit is disabled.
- Both 16- and 32-bit streaming data transfers must start on four-byte address boundaries.
- A 64-bit streaming data transfer must be aligned on an eight-byte address boundary.

Data is validated by using CD CHRDY to indicate ready or not ready, so that the slave has the ability to pace the data transfer for 16- or 32-bit transfers. Either the master or slave can terminate the cycle at any time.

Implementing and using streaming data transfer is optional to masters. Implementing and indicating streaming data transfer capabilities is optional to slaves.

Streaming Data Procedure (16- or 32-Bit)

A 16- or 32-bit streaming data transfer cycle is initiated as either a 16- or 32-bit basic data transfer cycle, and all rules associated with those procedures apply. The following is an example of a 16- or 32-bit streaming data write cycle. The signals appear in the following sequence:

1. The master drives the address bus, MADE 24, M/-IO, -APAREN, and APAR(0-3) valid to begin the cycle.
2. In response to an unlatched decode of address, MADE 24, and M/-IO, the selected slave returns:
   - -CD SFDBK
   - -CD DS 16 (if the addressed data port is capable of 16-bit operations)
   - -CD DS 16 and -CD DS 32 (if the addressed data port is capable of 32-bit operations).
3. The master drives -S0 active (-S1 inactive).
4. In response to an unlatched decode of address, MADE 24, M/-IO, and status, the selected slave drives CD CHRDY inactive, if the cycle is to be extended.
5. A 32-bit master (or the central translator logic for a 16-bit master in a 32-bit system) drives the byte enable signals. -BE(0-3) is driven to binary 0011 by the central translator logic, or -BE(0-3) is driven to binary 0000 by a 32-bit master.

   Note: Because 32-bit operations are aligned on four-byte boundaries, A0, A1, and -SBHE equal a binary 000.
6. The master drives -ADL valid. A slave can latch decodes of address, status, and M/-IO at the leading or trailing edge of -ADL.
7. The slave drives -SDR(0,1) active.

   Note: When the slave **does not** drive -SDR(0,1) active (indicating that it is defaulting to basic transfer procedures), the controlling master will drive -S(0,1) inactive after the hold time specified by timing parameter T7 and T10 (see Figure 24 on page 45).
8. The master drives the write data onto the bus and, if supported, drives -DPAREN active and DPAR(0-3) valid.

9. The master drives -CMD active and -ADL inactive. A slave must latch the decodes of address, status, and M/-IO with the leading edge of -CMD if they were not latched with -ADL.
10. If CD CHRDY has been driven inactive, the cycle is extended until CD CHRDY is driven active. CD CHRDY must not be held inactive longer than specified.
11. The controlling master indicates the ability to perform streaming data by starting -SD STROBE.

    Note: If CHRDYRTN is active, -SD STROBE can be activated concurrently with -CMD. -SD STROBE is used by both the controlling master and slave to transfer data, with data being clocked on and off the bus on the falling clock edge and clocked off the bus on the next falling clock edge. The operation proceeds with new data being placed on the bus every time -SD STROBE makes a high-to-low transition.
12. The master drives status inactive.
13. The slave drives -SDR(0,1) inactive.
14. The address bus can change in preparation for the next cycle.
15. In response to an address change, the slave updates the state of:
    - -CD SFDBK
    - -CD DS 16
    - -CD DS 32.
16. The status signals and M/-IO can become valid in preparation for the next cycle.
17. The master stops driving -SD STROBE.
18. On the last transfer, the controlling master drives -CMD inactive, and the selected slave gates the data off the bus. The transfer cycle is complete.

Streaming Data Procedure (64-Bit)

A 64-bit streaming data transfer cycle is initiated as a 32-bit basic transfer cycle, and all the rules associated with this procedure apply. The following is an example of a 64-bit streaming data write cycle. The signals appear in the following sequence:

1. The master drives the address bus, MADE24, M/-IO, -APAREN, and APAR(0-3) valid to begin the cycle.
2. In response to an unlatched decode of address, MADE24, and M/-IO, the selected slave returns:
   - -CD SFDBK
   - -CD DS 16
   - -CD DS 32.
3. The master drives -S0 active (-S1 inactive).
4. In response to an unlatched decode of address, MADE24, M/-IO, and status, the selected slave drives CD CHRDY inactive, if the cycle is to be extended.
5. A 32-bit master drives -BE(0-3) to binary 0000 (64-bit streaming data transfers are not performed by 16-bit masters).

   Note: Because 64-bit operations are aligned on an eight-byte boundary, A0, A1, and -SBHE equal binary 000.
6. The master drives -ADL active. A slave can latch decodes of address, status, and M/-IO at the leading or trailing edge of -ADL.
7. The slave drives -SDR(0,1) to indicate its transfer speed capability and drives -MSDR active to indicate 64-bit capability.

   Note: When the slave does not drive -SDR(0,1), and -MSDR active (indicating that it is defaulting to a basic transfer procedure), the controlling master drives -S(0,1) inactive after the hold time specified by timing parameters T7 and T10, and prior to driving -CMD active (see Figure 24 on page 45.)
8. The master drives the write data onto the data bus and, if supported, drives -DPAREN active and DPAR(0-3) valid.
9. The master drives -CMD active and -ADL inactive. A slave must latch the decodes of address, status, and M/-IO with the leading edge of -CMD if they were not latched with -ADL.

10. If CD CHRDY has been driven inactive, the cycle is extended until CD CHRDY is driven active. CD CHRDY must not be held inactive longer than specified.

11. The controlling master indicates the ability to perform streaming data transfers by activating -SD STROBE. The 64-bit streaming data transfer capability is indicated by driving -BE(0-3) inactive, driving the four most-significant bytes of the eight byte transfer on the address bus, and, if -DPAREN is active, driving APAR(0-3) valid. -APAREN is driven inactive.

    Note: -SD STROBE is used by both the controlling master and the slave to transfer data; data is clocked on and off the address bus and the data bus on the falling clock edge and clocked off the address bus and the data bus on the next falling clock edge. The operation proceeds with the new data being placed on the address bus and the data bus each time -SD STROBE makes a high-to-low transition.

12. Both the address bus and data bus are used for transferring data for the remainder of the cycle.

13. The status signals are driven inactive on the last high-to-low transition of -SD STROBE. The slave drives -SDR(0,1) inactive.

14. In response to data on the address bus, other slaves can drive -CD SFDBK, -CD DS 16, -CD DS 32, and CD CHRDY valid during the data transfer. The master should not monitor the state of these signals during the data transfer.

15. On the last transfer, the controlling master drives -CMD inactive and the slave gates the data off the address bus and the data bus. The cycle is complete.

Streaming Data Transfer Rates

The amount of data transferred per -SD STROBE period is constant during streaming data transfer. The number of bytes transferred in one transfer cycle are listed below.

Streaming Data Master Size	16-Bit Slave	32-Bit Slave	64-Bit Slave
16-Bit	2	2	2
32-Bit	2	4	4
64-Bit	2	4	8

Figure 33. Streaming Data Transfer Rates

Note: When a slave requests a streaming data transfer, it must be prepared to support a minimum of two transfers. When a master initiates a streaming data transfer, it must be prepared to support a minimum of one transfer.

Address Bus Management

During streaming data transfers, the address on the address bus is neither incremented nor decremented. The controlling master and the slave must provide memory-address-space address management and I/O-address-space address management as follows:

- Memory-address-space addresses

  The controlling master must keep a count of the number of bytes transferred, so that if early termination of the block transfer occurs, the master can assert the correct address (original address plus the number of bytes transferred) at the start of the next data transfer procedure.

  The slave must manage the address, to ensure that the data being clocked off the bus is gated into the correct memory addresses.

- I/O-address-space addresses

  If early termination of the block transfer occurs, the master must present the same (original) address on the address bus when restarting the streaming data procedure.

  The slave must accommodate the streaming data transfer to the same I/O address.

During 64-bit streaming data transfers (after the address is removed from the address bus), the controlling master must ignore CHRDYRTN, -DS 16 RTN, -DS 32 RTN, and -SFDBKRTN. Slaves (except the selected slave) are decoding the address bus and responding as if selected.

Address Boundary Alignment

If the starting address is not a four-byte address boundary and a 16- or 32-bit streaming data procedure is going to be used, then the basic transfer procedure must be used to transfer data until the address has the correct boundary alignment. The controlling master must execute the basic transfer procedure to obtain address boundary alignment.

A slave can indicate streaming data capability, by driving -SDR(0,1) active, without checking address boundary alignment. Then the controlling master and the slave can use the streaming data procedure. The basic transfer procedure might also be required to complete the data transfer, even if the initial data address was a four-byte address boundary.

If the starting address is not on an eight-byte address boundary and a 64-bit streaming transfer procedure is going to be used basic transfer cycles are used to transfer data until the address has a correct boundary alignment. The basic transfer procedure might also be needed to complete the data transfer, even if the initial data address was a eight-byte address boundary.

The following is an example of a basic transfer procedure occurring before and after a 16- or 32-bit streaming data procedure. The controlling master should drive -BURST active to indicate that one or more consecutive data transfer cycles will occur. Figure 34 on page 65 shows the data block to be transferred.

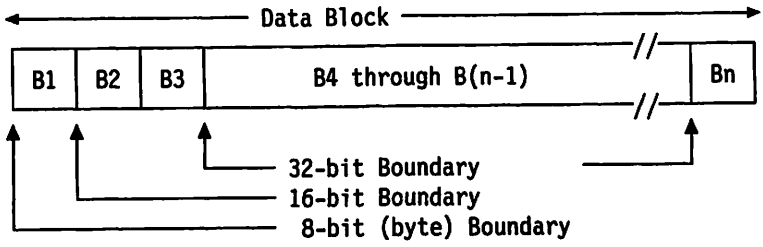


Figure 34. Data Block Example

In the above example, the data transfer procedures used to achieve address boundary alignment between a controlling master and the selected slave are:

- Byte 1 (B1) is transferred using a basic transfer procedure.
- Bytes 2 (B2) and 3 (B3) are transferred using a basic transfer procedure.

  The data address is now on a 32-bit boundary.
- Bytes B4 through B(n-1) are transferred using the streaming data procedure. Each period of -SD STROBE transfers two or four bytes of data. The streaming data procedure continues until all data, except byte Bn, is transferred.
- The streaming data procedure is terminated.
- Byte Bn, the last byte is transferred using a basic transfer procedure.
- The session between the controlling master and the slave is terminated.

Streaming Data Cycles

A master can perform streaming data transfers at 10 MHz when -SDR(0,1) is a binary 00, 01, or 10, and can perform the transfer at less than the requested width. -SDR(1) is reserved and slaves should not activate it. The valid streaming data signal combinations from the selected slave are shown in the following table.

-SDR 0 1	-CD DS 16/32	-MSDR	Type of Cycle Requested by Slave
0 0	X X	X	Reserved
0 1	0 0	0	64-bit Streaming Data Transfer
0 1	0 0	1	32-bit Streaming Data Transfer
0 1	0 1	1	16-bit Streaming Data Transfer
1 0	X X	X	Reserved
1 1	X X	X	Basic Transfer

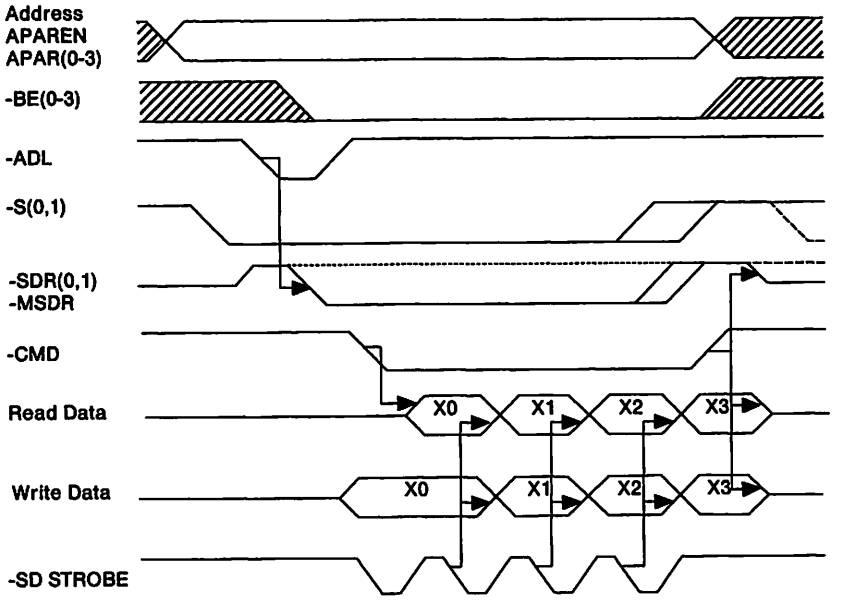
Figure 35. Streaming Data Signal Combinations

Both 16- and 32-bit masters can use streaming data procedures with either 16- or 32-bit slaves. However, because the central translator logic does not perform data steering during a streaming data transfer, a 32-bit slave must perform the steering when streaming with a 16-bit master. (The 32-bit master does the steering for all data transfers with slaves having a smaller data port.)

"Streaming Data Procedure (16- or 32-Bit)" on page 59 and "Streaming Data Procedure (64-Bit)" on page 61 contain the general signal sequence overview.

The following describe how:

- Masters and slaves select streaming data or basic transfer procedures, and the number of bits in each transfer.
- Deferred cycles are performed.
- Data is paced during streaming data procedures.
- Streaming data can be terminated by either the master or the slave.



X0, X1, X2, and X3 represent a data transfer of 16 or 32 bits.

Figure 36. 16- and 32-Bit Streaming Data Cycle Overview

64-Bit Streaming Data Cycles

The 64-bit streaming data transfer can only be used between 64-bit streaming masters and 64-bit streaming slaves. An example of a 64-bit cycle is shown in Figure 37. The 64-bit streaming transfer is similar to the streaming data transfer shown in Figure 36 on page 67, but uses both the data bus and the address bus to achieve the 64-bit data transfer width.

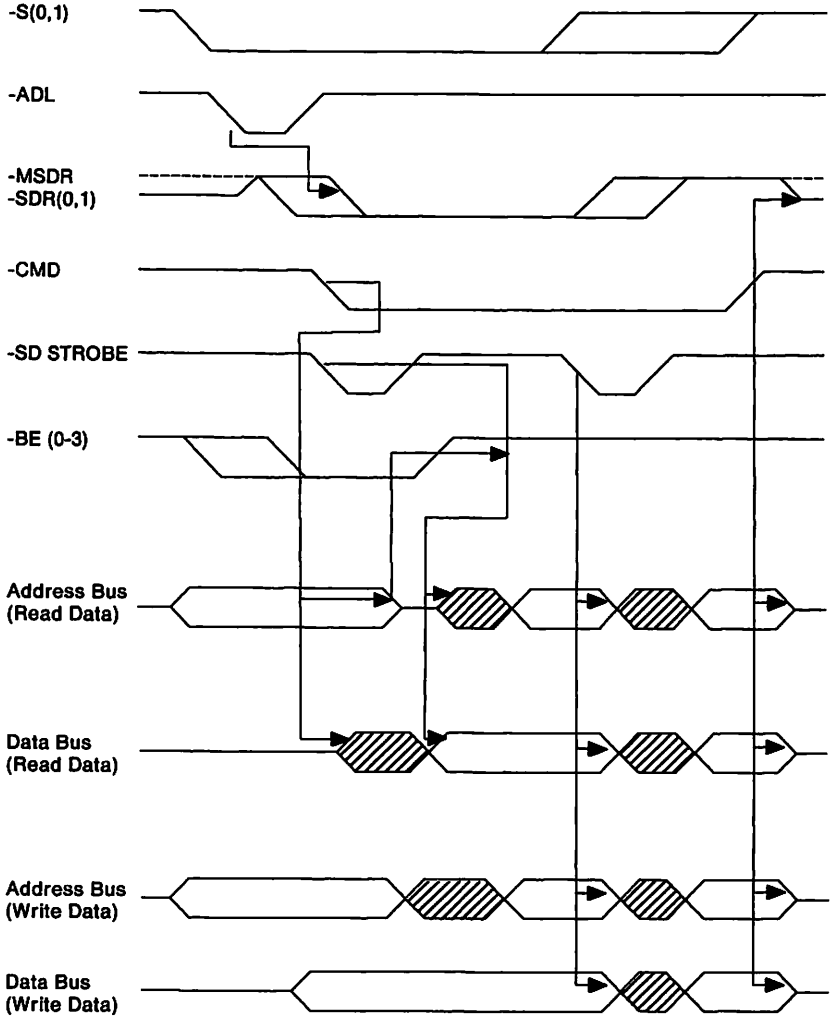


Figure 37. 64-Bit Streaming Data Cycle Overview

The 64-bit cycle begins as a 32-bit basic transfer cycle. The selected slave responds by driving -CD DS 16, -CD DS 32, -SDR(0,1) and -MSDR valid to indicate that it supports 64-bit streaming operations. -BE(0-3) are driven inactive by the master to indicate a 64-bit streaming-data procedure will be used. During a transfer to the slave, the master starts -SD STROBE and gates the data onto the data and address buses. During a transfer to the master, the master tri-states the address bus after driving -CMD active, and, after the trailing edge of -BE(0-3), the slave gates the data onto the data and address buses. The least significant byte is D(0-7) and the most significant byte is A(24-31).

The 64-bit streaming transfer then proceeds like a 32-bit streaming data transfer.

If -DPAREN has been driven active, the parity bits for data bytes zero through three are contained in DPAR(0-3), the parity bits for data bytes four through seven are contained in APAR(0-3).

Deferred start of the 64-bit streaming transfer is supported, but data pacing is not available during 64-bit data streaming transfers.

Starting the Cycle

Streaming masters can drive -SD STROBE active at the same time as -CMD to minimize the overhead in starting the streaming data cycle. The T70 time plus propagation delay for the signals to and from the slave can delay the sample point of -SDR(0,1) and -MSDR until after -CMD and -SD STROBE have been activated. Therefore, if a slave indicates it is capable of performing a streaming data transfer, the master must perform at least one streaming data transfer. Requiring one streaming data transfer prevents the slave from confusing a 32-bit basic transfer with a 64-bit streaming data transfer.

Deferring the Start of the Cycle

During a streaming data transfer, the slave may need more time at the start of the cycle to prepare for the transfer. The slave can defer the start of the transfer by driving CD CHRDY inactive. The controlling master delays the start of -SD STROBE until after CHRDYRTN goes active.

If -SDR(0,1) is not active within the time specified (T70A), the controlling master continues the cycle using basic transfer procedures. All specifications for the basic transfer procedure apply.

If -SDR(0,1) is active, after CHRDYRTN becomes active, the streaming data procedure can be invoked as shown in Figure 38 on page 71.

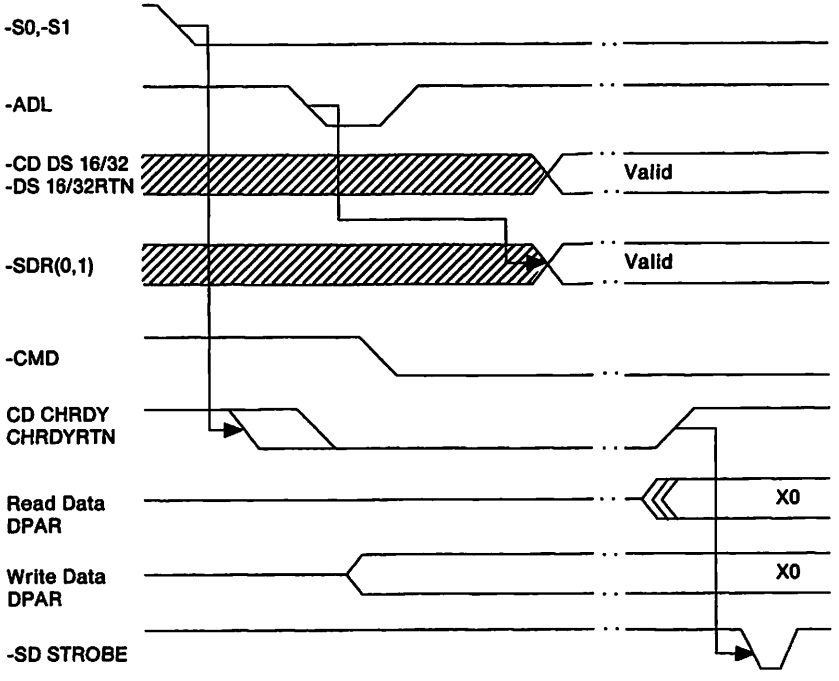


Figure 38. Streaming Data Deferred Start Overview

Pacing the Data

After a 16- or 32-bit streaming data cycle is started, CD CHRDY and CHRDYRTN pace the data transfer, allowing slaves to introduce momentary pauses in the data transfer. CD CHRDY is clocked simultaneously with data and makes state transitions following the high-to-low transition of -SD STROBE On each falling edge, the controlling master must inspect the state of CHRDYRTN. When CHRDYRTN is inactive during a read operation, it indicates that the slave did not have valid data in this clock period and that the transfer must be repeated. When CHRDYRTN is inactive during a write operation, it indicates the slave did not accept the data and that the transfer must be repeated.

Note: Data pacing by the slave is not supported during 64-bit streaming data cycles.

An example of data pacing using CD CHRDY is shown in Figure 39 on page 73. The example shows two idle (CD CHRDY inactive) clock periods, one in the middle of the data transfer, and one at the end. If the slave has driven CD CHRDY inactive, it must not deactivate -SDR(0,1).

When -MSDR is active, the slave cannot pace the transfer. The slave can use CD CHRDY only to delay the initiation of 64-bit transfers. Slaves that cannot meet the timing specifications for the 64-bit transfer must terminate the streaming data cycle. After -SD STROBE is being driven, the master ignores CHRDYRTN.

In this example, as the controlling source of CD CHRDY, the slave internally clocks its bus interface registers with the inversion of -SD STROBE. At the end of the cycle, the bus interface registers are clocked when -CMD goes high. The internal interface clock of the controlling master is conditional on CHRDYRTN being active.

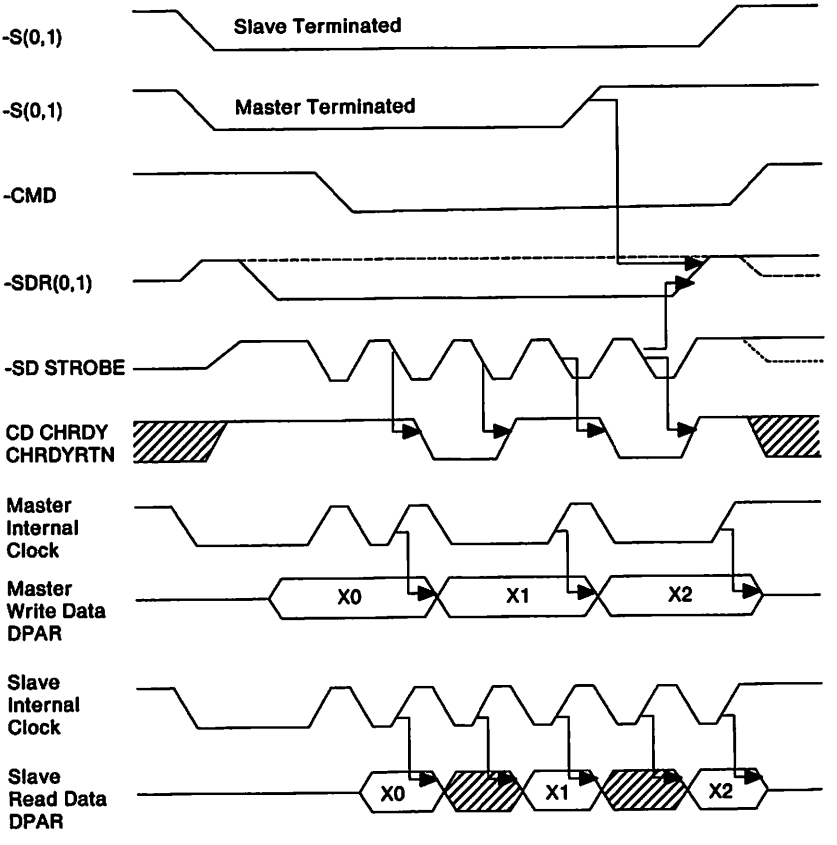


Figure 39. CD CHRDY and Data Pacing

To perform data transfer pacing, streaming masters can introduce momentary pauses in the data transfer by extending -SD STROBE. Data is transferred on the falling edge of -SD STROBE. The strobe period can be extended at either the high or low level, or both. This is done asynchronously to the edges of -SD STROBE.

Whenever the slave is not-ready (CD CHRDY inactive), the master must continue to pulse -SD STROBE. The actual data transfer will occur on the falling edge of -SD STROBE once the slave becomes ready by driving CD CHRDY active.

Terminating the Cycle

Either the controlling master or the slave can terminate the streaming data procedure. If -PREEMPT has become active, the controlling master terminates the streaming data procedure within 7.8 microseconds and releases the channel at EOT. If the fairness feature is active, the preempted master enters the inactive state. See "Fairness Feature and the Inactive State" on page 34 for more information.

Figure 40 shows streaming data cycle termination by the controlling master. Figure 41 on page 75 shows streaming data cycle termination by the controlling master when the slave is not ready. Figure 42 on page 76 shows streaming data cycle termination by a streaming slave.

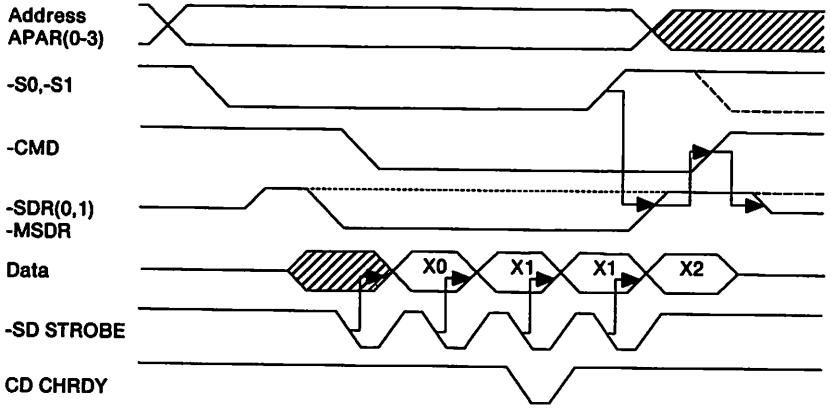


Figure 40. Master-Terminated Streaming Data Cycle

The controlling master can terminate a streaming data cycle by deactivating -S(0,1) on the data transfer before the desired stopping point. This can be done concurrently with the high-to-low transition of -SD STROBE. If -SDR(0,1) is inactive, the controlling master drives -CMD inactive at the point where the next -SD STROBE would have been. The slave **must not** drive -SDR(0,1) inactive while driving CD CHRDY inactive during 16- or 32-bit transfers. If -SDR(0,1) is active and CD CHRDY is inactive, it indicates that the slave is not ready, and the controlling master cannot terminate the cycle.

Figure 41 shows streaming data cycle termination by the master, when the slave is not ready. The master continues to pulse -SD STROBE until it detects the deactivation of -SDR(0,1), which implies that CD CHRDY has been activated. Following the slaves deactivation of -SDR(0,1), the master proceeds to terminate the streaming data cycle. The slave must drive -SDR(0,1), according to T71, on the last -SD STROBE falling edge, similar to a slave-terminated cycle.

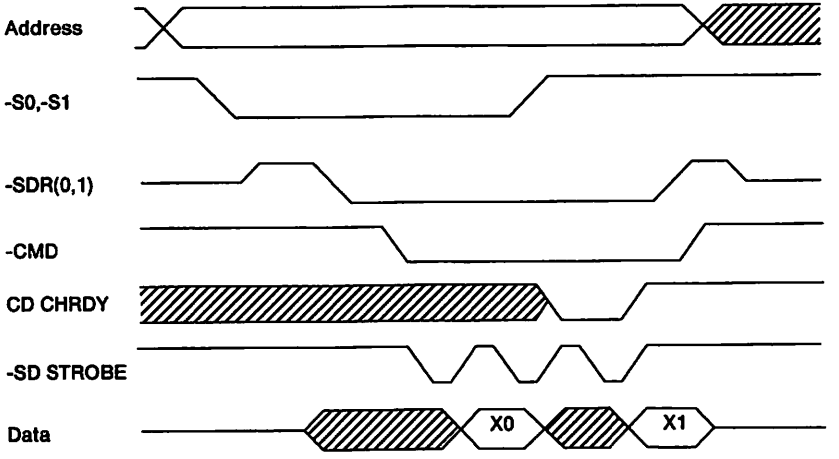


Figure 41. Master-Terminated Streaming Data Cycle (Slave Not Ready)

Figure 42 shows how a streaming data cycle can be terminated by a slave. However, if the master has not transferred all of its data, the master can immediately initiate an additional data transfer procedure.

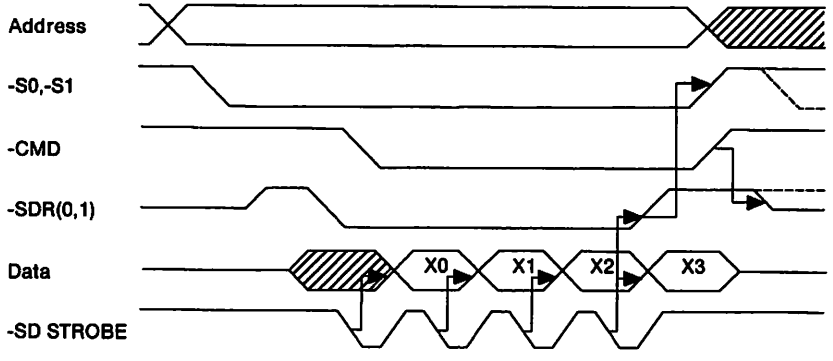


Figure 42. Slave-Terminated Streaming Data Cycle

The slave can terminate a streaming data cycle by driving -SDR(0,1) inactive following the last desired high-to-low transition of the -SD STROBE. The slave must not drive -SDR(0,1) inactive while CD CHRDY is inactive. The controlling master stops -SD STROBE and deactivates -S0, -S1 and -CMD at the point where the next -SD STROBE would have been.

A slave that does not drive CD CHRDY inactive (in response to slave selection) must not drive -SDR(0,1) or -MSDR inactive with the intention of executing a slave-terminated single-cycle operation. This ensures that the master has adequate time to sample the state of -SDR(0,1) before it is deactivated. The slave must be able to execute **two or more** data transfers with the master. The slave can sample TR32 to determine if two or four bytes will be transferred with each -SD STROBE. A master can perform a single streaming-data cycle operation. A master that performs a single streaming-data cycle must make sure that it meets the basic timing requirements (T10) as well as all master-terminated streaming-data requirements, including T71A.

When -CMD is driven inactive, the slave tri-states -SDR(0,1), -MSDR and, if performing a read operation, the data bus. The termination sequence results in restoring the streaming data signals to their inactive state prior to their being tri-stated, eliminating uncertainties about the state of these signals in succeeding channel cycles.

Streaming Data Timing

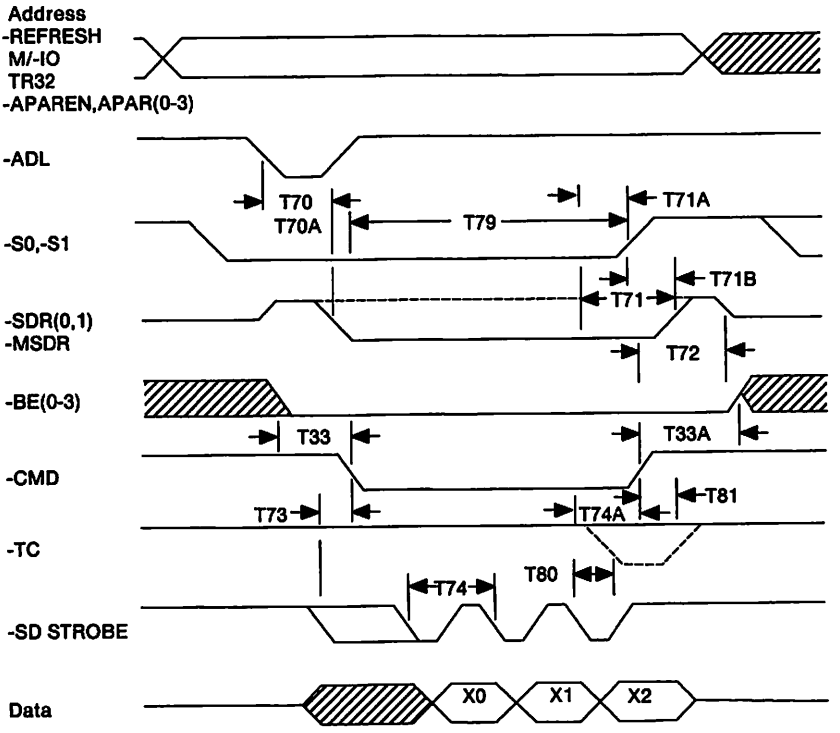


Figure 43. Streaming Data Cycle

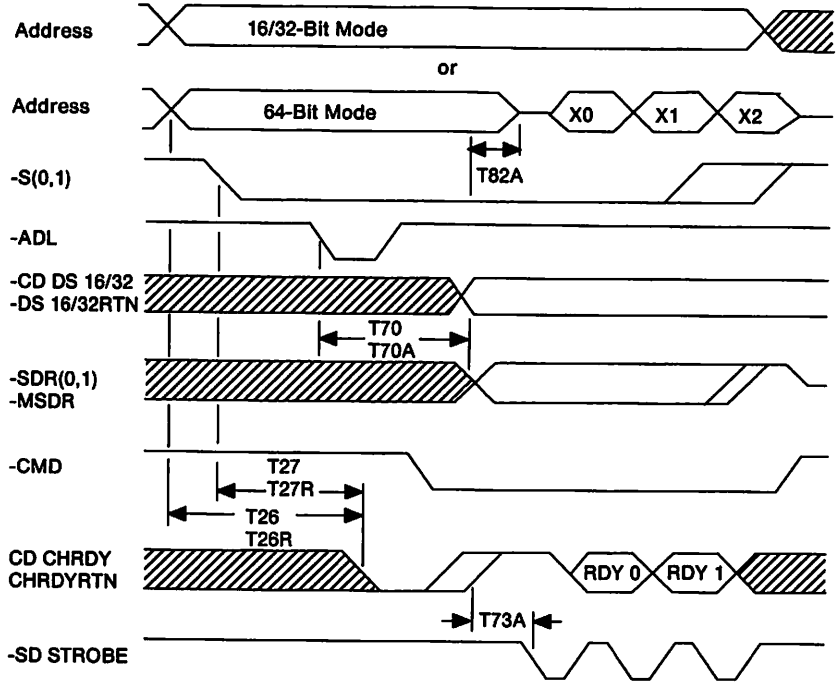


Figure 44. Streaming Data Cycle - Deferred Start

Note: CD CHRDY is not supported during 64-bit streaming data transfers after the address has been removed.

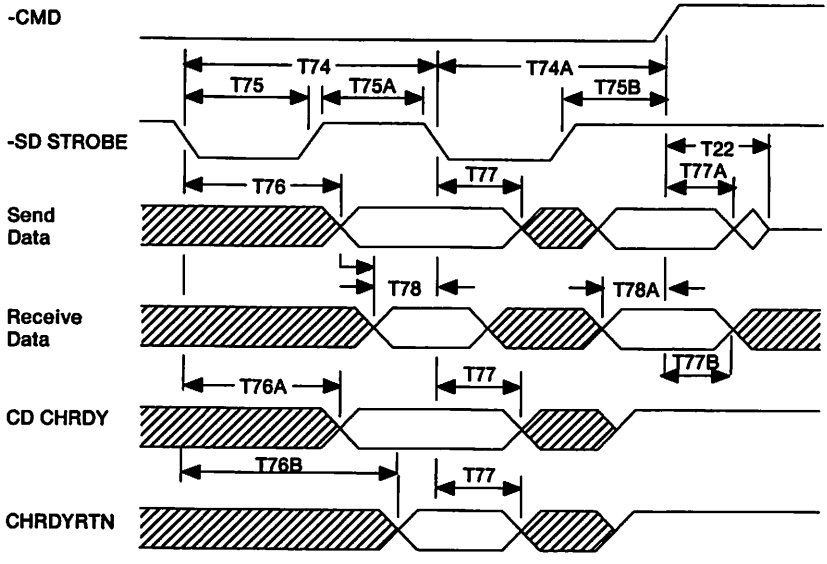
	Timing Parameter	Min/Max	Note
T22	Data bus tri-state from -CMD inactive	0 / 40 ns	
T33	-BE(0-3) active to -CMD active	10 / - ns	
T33A	-BE(0-3) hold from -CMD inactive	10 / - ns	
T70	-SDR(0,1) and -MSDR valid from -ADL active (slave only)	0 / 40 ns	7
T70A	-SDR(0,1) and -MSDR valid from -ADL active (masters only, for deferred cycle use)	0 /115 ns	
T71	-SDR(0,1) and -MSDR inactive from last -SD STROBE fall (slave terminated cycle)	0 / 40 ns	
T71A	-S0, -S1 inactive from last -SD STROBE fall (master terminated cycle)	- / 10 ns	4,1
T71B	-SDR(0,1) and -MSDR inactive from -S0, -S1 inactive (master terminated; slave ready)	0 / 40 ns	
T72	-SDR(0,1) and -MSDR tri-state from -CMD inactive (master-terminated cycle)	0 / 40 ns	
T73	-SD STROBE active to -CMD active	- / 10 ns	3
T73A	-SD STROBE active from CHRDYRTN active	0 / - ns	
T74	-SD STROBE period	100/ - ns	
T74A	-CMD inactive from last -SD STROBE fall	100 / - ns	
T79	Status inactive from -CMD active (master-terminated cycle)	- / 7.8 μs	6
T80	-TC active from last -SD STROBE fall	- / 60 ns	1
T81	-TC hold from -CMD inactive	10 / - ns	
T82A	A(0-31) tri-stated from CD CHRDY active	0 / - ns	5

Figure 45. Streaming Data Transfer Timing Specifications

Notes:

1. Streaming masters can increase T71A and T80 by the amount that the -SD STROBE clock period exceeds T74 minimum.
2. For streaming data cycles that default to the basic transfer cycles, the rules for deactivation of -S0, -S1 apply, as specified by the basic transfer procedure.
3. -SD STROBE can be driven active concurrently or after -CMD active, and can become active just before -CMD active due to driver skew and loading.
4. -S0, -S1 are deactivated concurrently with the negative transition of -SD STROBE. Normal skew from circuits and net loading can result in -S0, -S1 being deactivated prior to the negative transition of -SD STROBE in order to meet the T71A timing requirement.

5. This timing applies only to the 64-bit streaming data transfer.
6. The master must signal termination of the streaming data transfer within 7.8 microseconds from -CMD active. However, if -PREEMPT goes active, the master can continue to hold the channel for 7.8 microseconds from -PREEMPT.
7. A valid address decode must be completed before driving these signals.



See notes for definition of Send and Receive Data

Figure 46. Streaming Data Clocking (16- and 32-Bit)

	Timing Parameter	Min/Max	Note
T74	-SD STROBE period	100 / - ns	
T74A	-CMD inactive from last -SD STROBE fall	100 / - ns	
T75	-SD STROBE active	35 / - ns	
T75A	-SD STROBE inactive	35 / - ns	
T75B	-SD STROBE inactive to -CMD inactive	35 / - ns	
T76	Send Data valid from -SD STROBE fall	- / 60 ns	1
T76A	CD CHRDY valid from -SD STROBE fall	3 / 25 ns	
T76B	CHRDYRTN valid from -SD STROBE fall	3 / 45 ns	
T77	Send Data and CD CHRDY hold from -SD STROBE fall	10 / - ns	
T77A	Write data hold from -CMD inactive	11 / - ns	
T77B	Read data hold from -CMD inactive	7 / - ns	
T78	Receive Data valid before -SD STROBE fall	25 / - ns	
T78A	Receive Data valid before -CMD inactive	25 / - ns	

Figure 47. Streaming Data Transfer Timing Specifications (16- and 32-Bit)

Notes:

1. Streaming masters can increase T76 by the amount that the -SD STROBE clock period exceeds T74 minimum.
2. To perform data transfer pacing, streaming masters can introduce momentary pauses in the data transfer by extending the active or inactive levels of the streaming data strobe.
3. Send Data refers to write data at the connector for the master and read data at the connector for the slave. Receive Data refers to read data at the connector for the master and write data at the connector for the slave.
4. The value of (T74 minus T76 minus T78) represents system timing margins to accommodate bus skew, propagation delays, and other signal transmission delays.

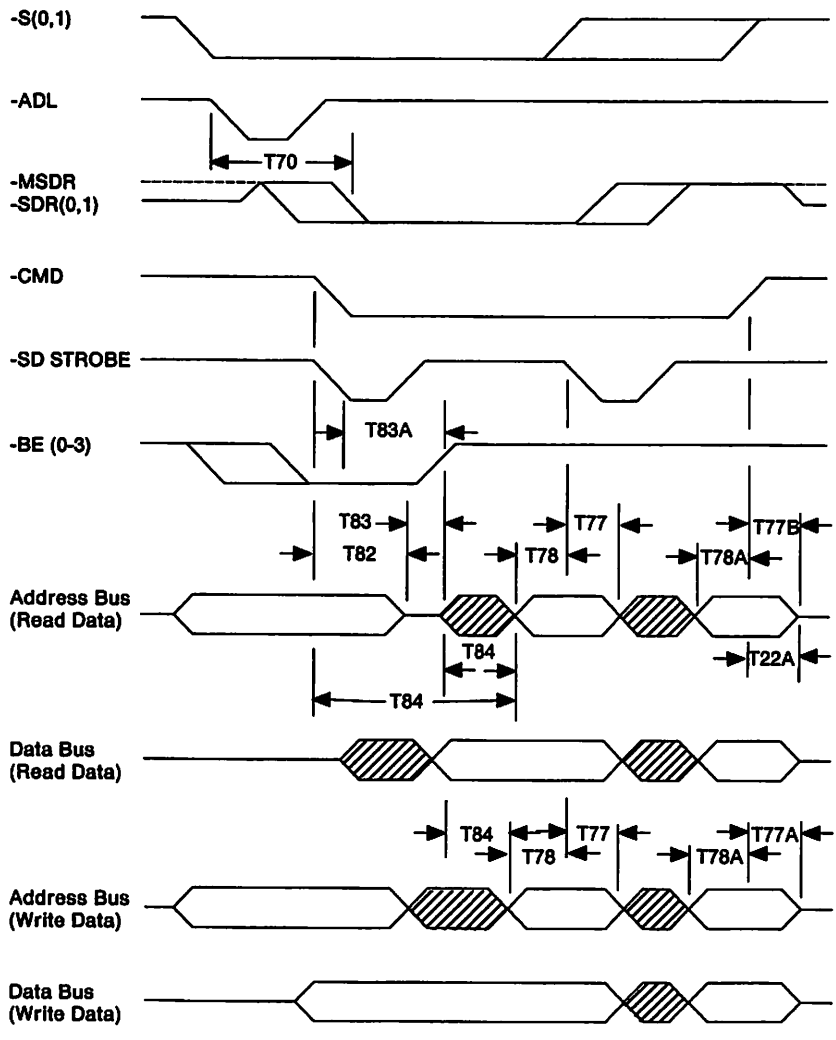


Figure 48. Streaming Data Clocking (64-Bit)

	Timing Parameter	Min/Max	Note
T22A	A(0-31) as D(32-63) tri-state from -CMD inactive	0 / 40 ns	2
T82	A(0-31) tri-stated from -CMD active	30 / - ns	1
T83	-BE(0-3) inactive from A(0-31) tri-stated	0 / - ns	3
T83A	-BE(0-3) inactive from -SD STROBE active	20 / - ns	1
T84	A(0-31) valid as D(32-63) from -BE(0-3) inactive and -SD STROBE active	0 / 60 ns	4

Figure 49. Streaming Data Transfer Timing Specifications (64-Bit)

Notes:

1. This timing applies only to a master during a read operation.
2. This timing applies only to a slave during a read operation.
3. During write operations, the master must meet the setup and hold times for the 16- and 32-bit streaming data transfer.
4. Streaming masters can increase T84 by the amount that the -SD STROBE clock period exceeds T74 minimum.

Direct Memory Access Procedure

A direct memory access (DMA) operation is a specialized sequence of basic transfer procedure. It consists of a single transfer or a burst transfer between a DMA slave and a memory slave. The following lists only those definitions, signal sequences, and timing specifications that are different from or additional to those of a basic transfer cycle.

Read and Write Operations

During a DMA write operation, the DMA slave provides the data and the memory slave stores it. During a DMA read operation, the memory slave provides the data and the DMA slave stores it. If the slave is unable to provide or store data within the time specified for a normal operation, it extends the cycle by driving CD CHRDY inactive. To ensure the read data is valid, the slaves storing the data sample the data lines with reference to the trailing edge of the -CMD.

DMA Slave Selection

A DMA slave is selected by either its arbitration level or optionally by its I/O address, but not by both. The default selection is by decode of the arbitration level, status (-S0 exclusive-OR -S1), and an I/O cycle (M/-IO in the -IO state). Selection by I/O address is by decode of the I/O address (A0-A15), status (-S0, -S1), and an I/O cycle (M/-IO in the -IO state). When selection by I/O address is implemented for a DMA slave, a bit located in either configuration-data space or I/O address space is used to control the method of selection. All DMA slaves drive -CD SFDBK during the I/O portion of the DMA cycle for both selection methods.

Address During DMA Operation

During a memory cycle, the DMA controller drives the memory address on the address bus. During an I/O cycle, the DMA controller must drive a known I/O address on the bus. The I/O address must be that of the DMA slave, or hex FFFC or 0000.

Transfer Data Length

Normally, the DMA transfer length is an integral number of transfers of a width determined by the DMA slave's data port width. The data port address for a 16-bit DMA slave must be on an even address boundary; the address for a 32-bit DMA slave must be on a four-byte address boundary. The burden of supporting odd-length transfers is the responsibility of the DMA slave.

DMA Procedure

The two types of DMA procedures are:

- Single Transfer
- Burst Transfer.

Single Transfer

During a data transfer, the requesting DMA slave initiates the DMA operation by having its arbiter drive -PREEMPT active and arbitrate for the channel. On the low-to-high transition edge of ARB/-GNT (start of arbitration cycle), the DMA slave's priority level is placed on the arbitration bus. The DMA slave competes for use of the channel as defined in the arbitration procedure.

On the high-to-low transition edge of ARB/-GNT the DMA controller monitors the arbitration bus to determine if its DMA slave gained control of the channel. A match initiates a serial DMA operation in which the DMA controller drives the address lines during both cycles, as well as driving -S0, -S1, -CMD, and other control signals. The DMA controller also monitors -BURST, finds it inactive, and then activates -BURST (either on the channel, or through the system logic) to notify the central arbitration control point that consecutive cycles are required.

The DMA controller also manages the storage of data during the DMA read or DMA write operations for the consecutive cycles required by serial DMA transfers. Data count for data gated on or off the bus by the DMA slave is also maintained by the DMA controller.

Notes:

1. The DMA controller might be required to perform multiple cycles for the data transfer to the memory slave, if the data port width of the memory slave is less than that of the DMA slave.
2. DMA slaves using the single-transfer-type DMA procedure *will not* receive more data than is required to support their data port width (for example, 2 bytes to a 16-bit port).
3. Consecutive transfers can be made with the DMA slave (as opposed to the serial sequences shown in the remainder of this section). The memory read or write cycles might not appear on the channel.

Burst Transfer (Multiple Transfer Cycles)

The method for transferring a block of data is similar to a single transfer DMA cycle, except that the local arbiter for the requesting DMA slave activates -BURST to request continued channel ownership for a burst operation. The central arbitration control point will not drive ARB/-GNT to the ARB state while -BURST is active. As in single-transfer DMA operations, the DMA controller drives the address, -S0, -S1, -CMD, and data transfer signals, as well as manages addresses and data transfer counts. The DMA controller also drives -BURST (either on the channel or through the system logic) jointly with the DMA slave for DMA write cycles. It does not drive -BURST active for burst DMA read cycles.

DMA Terminating Conditions

The DMA transfer operation is terminated by the EOT. At this time, a DMA slave can again request DMA service by arbitrating for use of the channel.

The terminating conditions for DMA transfers are:

- During a single transfer, the DMA controller can terminate the DMA transfer by releasing -BURST.
- During burst transfers, either the DMA slave or the DMA controller can terminate the DMA operation.
  - The DMA slave terminates the transfer by driving -BURST inactive. The DMA slave will terminate the transfer when it reaches its buffer limit, or after -PREEMPT is driven active by another arbitrating participant. Arbitrating slaves should maximize use of the channel within the 7.8 microsecond time limit.
  - The DMA controller terminates the transfer by driving -TC active when the transfer count goes to zero. When the DMA slave receives -TC, it drives -BURST inactive.

Use of -BURST

When a bursting DMA slave wins the arbitration for the channel, it drives -BURST active. During default cycles, the DMA slave drives -BURST inactive when it detects status active for the last I/O cycle of this burst transfer. However, on sensing -SBHE and/or -BE(0-3), the DMA slave might discover that it has one more cycle to perform. If so, it redrives -BURST active to continue the transfer. See "Burst DMA Transfer (DMA-Slave Terminated - Default Cycle, 200 ns)" on page 91.

During extended cycles, the DMA slave drives -BURST inactive within the specified time after the leading edge of -CMD.

If a burst transfer is terminated by the DMA controller, the DMA slave drives -BURST inactive within the specified time after the leading edge of -TC.

For DMA write cycles using burst transfers, the DMA controller drives -BURST active at the leading edge of -CMD for each I/O read and drives it inactive when it detects status active for the next memory write.

DMA Timings

The following are the timing parameters for the DMA procedure.

Single DMA Transfer (DMA-Controller Controlled)

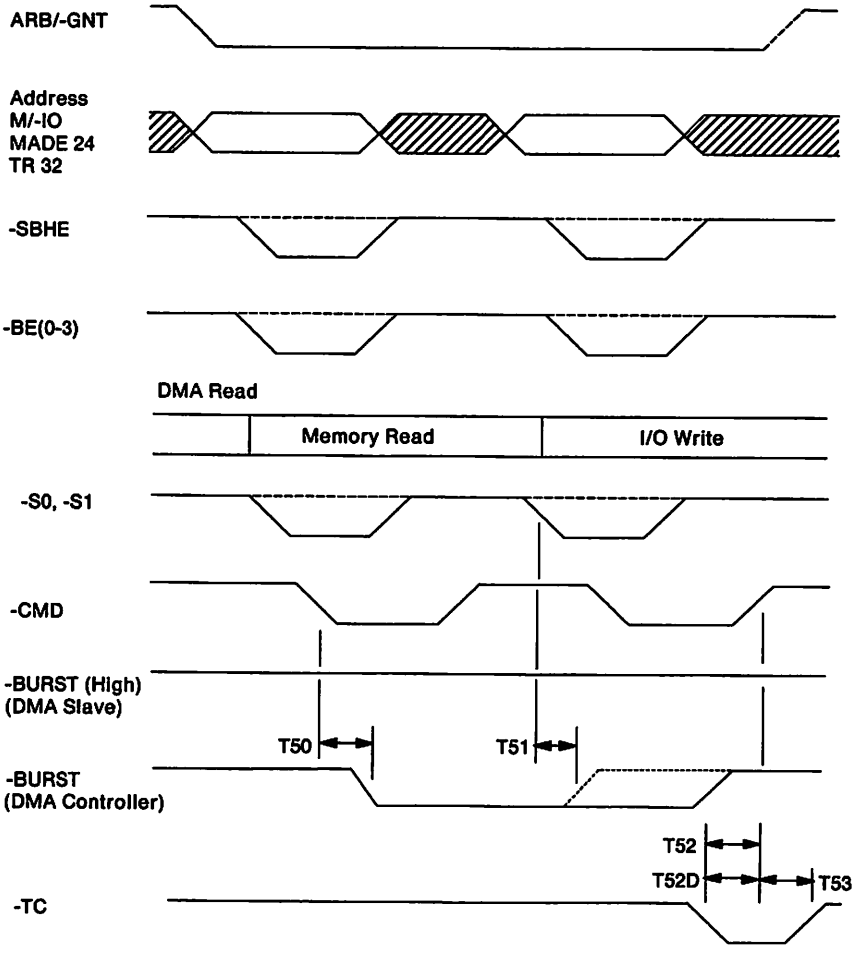


Figure 50 (Part 1 of 2). Single DMA Transfer (DMA-Controller Controlled)

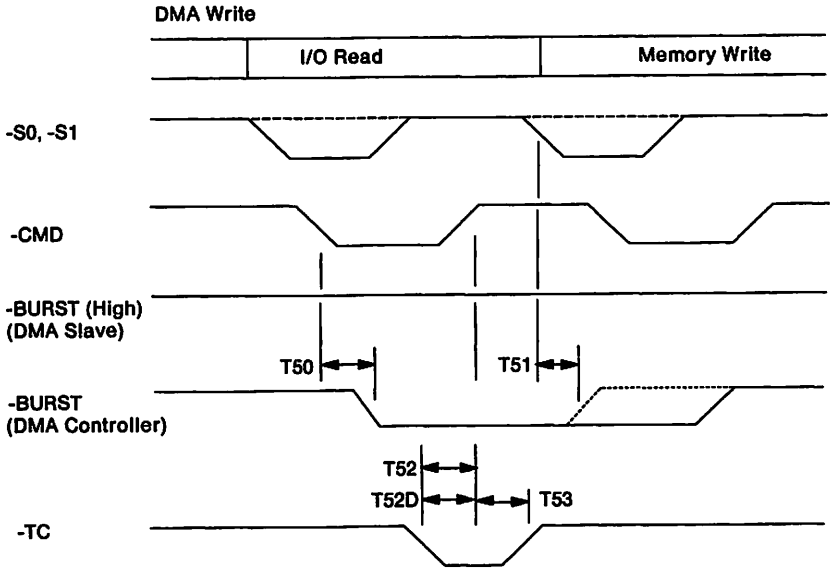


Figure 50 (Part 2 of 2). Single DMA Transfer (DMA-Controller Controlled)

	Timing Parameter	Min/Max	Note
T50	-BURST activated by the DMA controller from the first -CMD active	- / 30 ns	
T51	-BURST released by the DMA controller from the last status active	- / 40 ns	
T52	-TC setup to -CMD inactive	30 / - ns	2
T52D	-TC setup to -CMD inactive	15 / - ns	2
T53	-TC hold from -CMD inactive	10 / - ns	3

Figure 51. Single DMA Transfer (DMA-Controller Controlled) Timing Specifications

Notes:

1. Only the timing parameters that are additional to those specified for the basic transfer cycle are included here.
2. T52D applies to participants operating at 200-nanosecond cycles *only*. Devices operating at cycle rates slower than 200 nanoseconds *must* meet T52.
3. For streaming data procedures, T80 and T81 also apply. (See "Streaming Data Timing" on page 77).

Burst DMA Transfer (DMA-Controller Terminated)

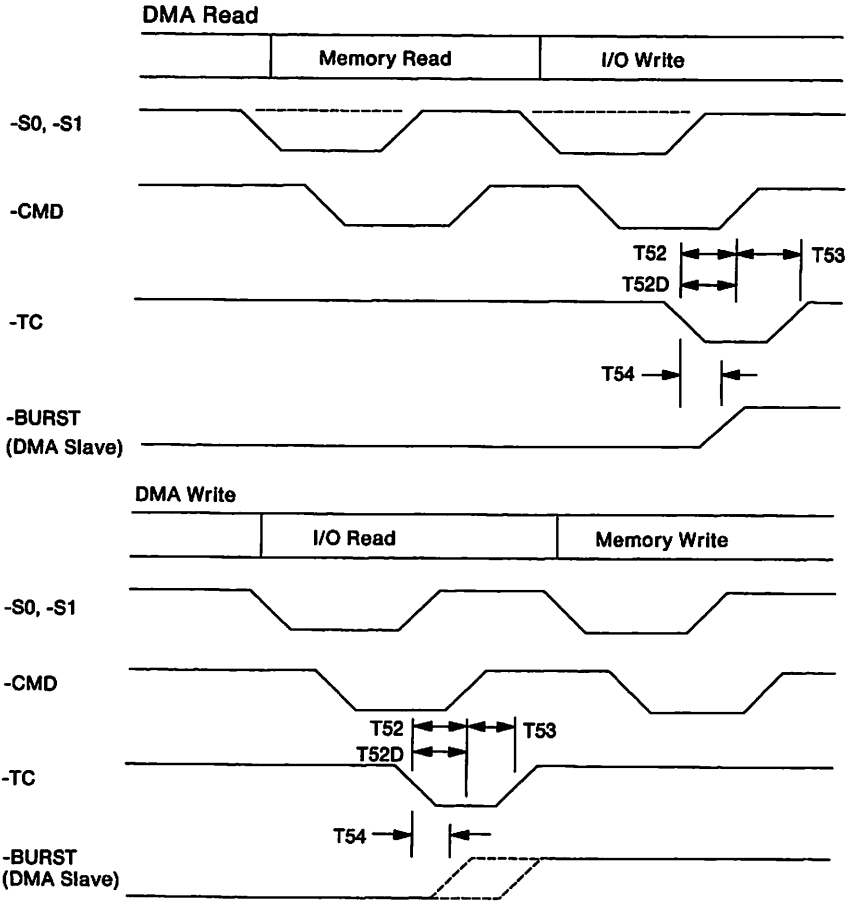


Figure 52. Burst DMA Transfer (DMA-Controller Terminated)

	Timing Parameter	Min/Max	Note
T52	-TC setup to -CMD inactive	30 / - ns	1
T52D	-TC setup to -CMD inactive	15 / - ns	1
T53	-TC hold from -CMD inactive	10 / - ns	3
T54	-BURST released by the DMA slave from -TC active	- / 30 ns	

Figure 53. Burst DMA Transfer (DMA-Controller Terminated) Timing Specifications

Notes:

1. T52D applies to devices operating at 200-nanosecond cycles *only*. Devices operating at cycle rates slower than 200 nanoseconds *must* meet T52.
2. During a DMA read, the DMA controller must not initiate another cycle after issuing -TC, even if the DMA slave is not fast enough to drive -BURST inactive before the end of -CMD.
3. For the streaming data procedure, T80 and T81 also apply. (See "Streaming Data Timing" on page 77).

Burst DMA Transfer (DMA-Slave Terminated - Default Cycle, 200 ns)

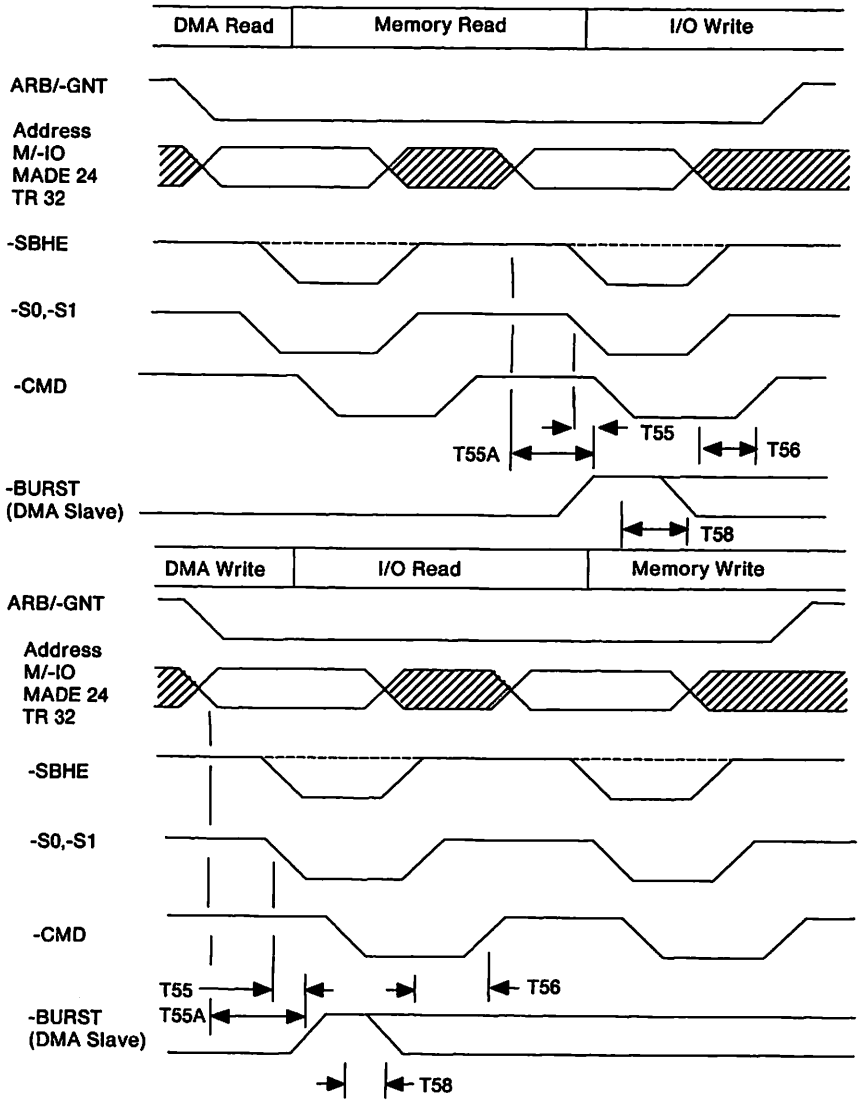


Figure 54. Burst DMA Transfer (DMA-Slave Terminated - Default Cycle, 200 ns)

	Timing Parameter	Min/Max	Note
T55	-BURST released by the DMA slave from the last I/O cycle status active (default cycle only)	- / 40 ns	1
T55A	-BURST released by the DMA slave from I/O address valid for the last I/O cycle (default cycle only)	- / 70 ns	1
T56	-BURST inactive to -CMD inactive	35/ - ns	2
T58	-BURST redrive from -CMD active	- / 30 ns	1

Figure 55. Burst DMA Transfer (DMA-Slave Terminated - Default Cycle, 200 ns) Timing Specifications

Notes:

1. After releasing -BURST and on receiving -SBHE and/or -BE(0-3), if the DMA slave determines that it requires another cycle, it can redrive -BURST active.
2. The setup time for -BURST inactive to the end of -CMD (T56) must be guaranteed during the last I/O write cycle to prevent the DMA controller from starting the next cycle. This setup time (T56) is guaranteed by the sum of -BURST release time (T55/T55A controlled by the DMA slave) and the -BURST RC restoration time. The RC restoration time must not exceed 70 nanoseconds. T56 is the same for default and extended cycles.

Burst DMA Transfer (DMA-Slave Terminated - Synchronous-Extended Cycle, 300 ns)

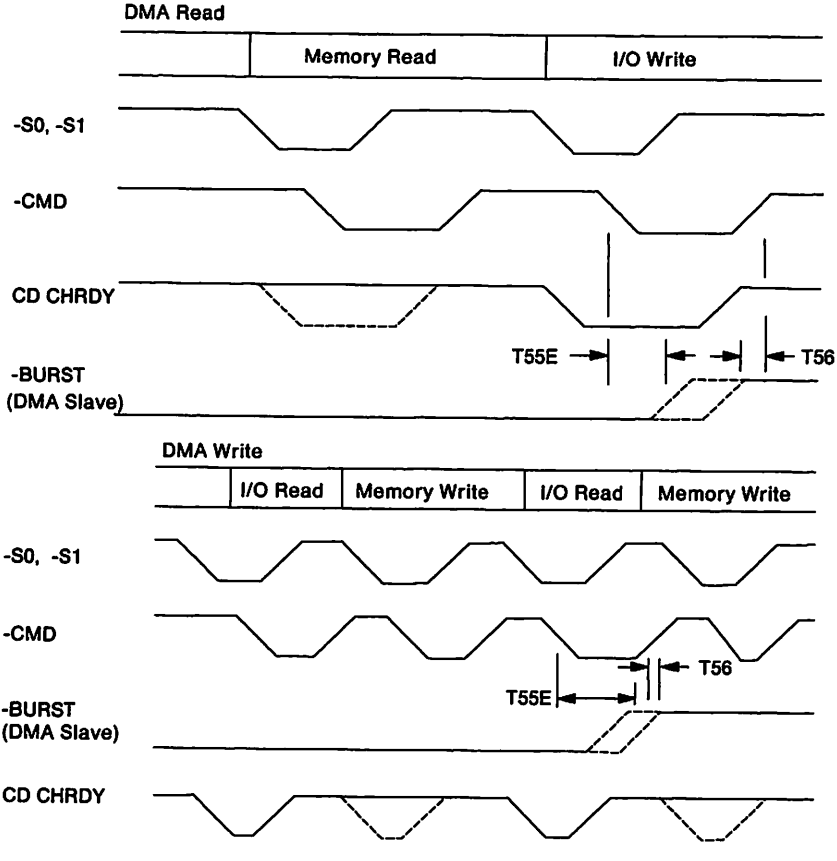


Figure 56. Burst DMA Transfer (DMA-Slave Terminated - Synchronous-Extended Cycle)

	Timing Parameter	Min/Max	Note
T55E	-BURST released by the DMA slave from the last I/O -CMD active (extended cycles only)	- / 80 ns	2
T56	-BURST inactive to -CMD inactive	35 / - ns	1

Figure 57. Burst DMA Transfer (DMA-Slave Terminated - Synchronous-Extended Cycle, 300 ns) Timing Specifications

Notes:

1. The setup time for -BURST inactive to the end of -CMD (T56) must be guaranteed during the last I/O write cycle to prevent the DMA controller from starting the next cycle. This setup time (T56) is guaranteed by the sum of -BURST release time (T55E controlled by the DMA slave) and the -BURST RC restoration time. The RC restoration time must not exceed 70 nanoseconds. T56 is the same for default and extended cycles.
2. -BURST cannot be redriven.

Burst DMA Transfer (DMA Slave Terminated - Asynchronous-Extended Cycle, ≥300 ns)

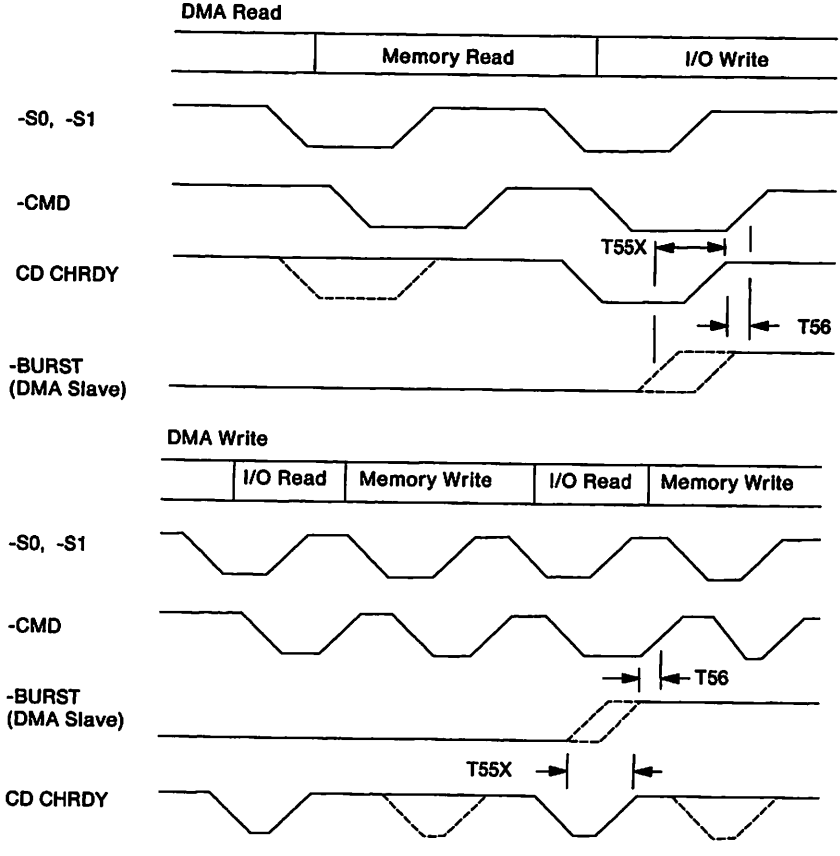


Figure 58. Burst DMA Transfer (DMA-Slave Terminated - Asynchronous-Extended Cycle, ≥300 ns)

	Timing Parameter	Min/Max	Note
T55X	-BURST released by the DMA slave before CD CHRDY active (asynchronous-extended cycles only)	50 / - ns	2
T56	-BURST inactive to -CMD inactive	35 / - ns	1

Figure 59. Burst DMA Transfer (DMA-Slave Terminated - Asynchronous-Extended Cycle, ≥300 ns) Timing Specifications

Notes:

1. The setup time for -BURST inactive to the end of -CMD (T56) must be guaranteed during the last I/O write cycle to prevent the DMA controller from starting the next cycle. This setup time (T56) is guaranteed by the sum of -BURST release time (T55X controlled by the DMA slave) and the -BURST RC restoration time. The RC restoration time must not exceed 70 nanoseconds. T56 is the same for default and extended cycles.
2. -BURST cannot be redriven.

Interrupt Procedures

This section describes how all devices use interrupts. The procedure for sharing interrupts uses a request for interrupt services that is detected by the level of the interrupt request signal (level sensitive). These procedures involve the interaction between the hardware and an interrupt service routine.

To initiate an interrupt request, a device drives its -IRQ(N) active (N represents the assigned interrupt level for the device) and holds the signal active until it is reset by the interrupt service routine.

Each device provides an interrupt-pending bit within its address space. This bit is set by the device when it has an interrupt request pending and is reset by the interrupt handler when the interrupt request is serviced.

Note: The drivers for the interrupt signal must be disabled when the device is disabled.

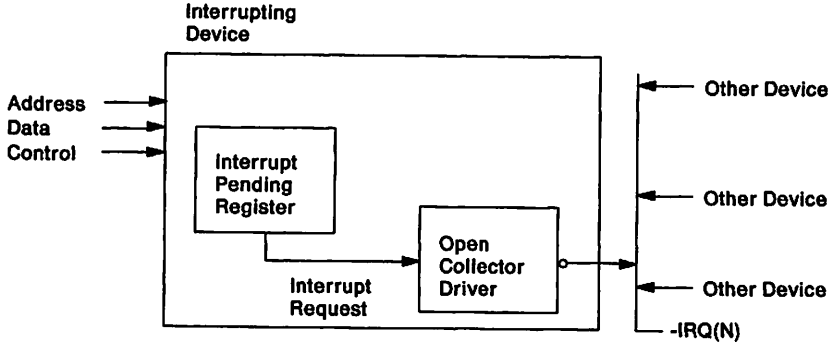


Figure 60. Typical Adapter Interrupt Sharing Implementation

An example of the sequence of the hardware and interrupt service routine interaction is shown in the following.

Hardware Operation

1. An interrupt condition causes the hardware to drive -IRQ(N) active and sets an interrupt-pending latch, which can be read by software.

Software Operation

2. A master begins executing code at the beginning of the appropriate chain of interrupt handlers.
3. The interrupt handler reads the interrupt-pending latch of the first device in the chain. If the latch is not set, the next device in the chain is tested. When the reporting device is detected, the handler executes the service routine.
4. The interrupt service routine operates the device hardware.

Hardware Operation

5. The device hardware resets the interrupt-pending latch and the interrupt request because of interrupt-service-routine actions.

Software Operation

6. The interrupt service routine completes the interrupt processing.

Hardware Operation

7. If another interrupt is pending (-IRQ(N) driven active by another device), the sequence starts again at 1.

Refresh

The timing of the refresh operation is performed by system logic. -REFRESH is driven active during a memory-read operation to indicate that a refresh cycle is in progress. The refresh cycle is a basic transfer default cycle with -REFRESH active. The timing for -REFRESH is the same as that of the address bus.

Note: Adapters with registers that have destructive read-outs and are mapped into the memory address space should include -REFRESH in their address decode.

During a refresh operation, the nine low-order address bits are incremented for each refresh cycle to address the next byte to be refreshed. (A0 changes after each refresh cycle; A8 changes after 256 cycles.)

The address bits, A31 through A9, might not change, but these bits will be stable during the refresh cycle.

The refresh operation can occur in periodic intervals or in short bursts of multiple operations. The refresh rate is a minimum of 128 refresh cycles every 2 milliseconds.

Note: Time periods between refresh cycles can vary and should not be used as a timing mechanism.

Regardless of whether it decodes -REFRESH or not, a memory slave can perform a memory read operation during a refresh cycle and drive its data onto the data bus, provided its memory address is on the address bus.

Memory slaves that do not need to perform refresh operations do not need to receive this signal.

System Configuration

The adapter configuration procedure consists of:

- Adapter ID and adapter configuration data
- Programmable option select (POS) registers
- Nonvolatile memory (for system configuration data)
- Signal timing specifications.

The system master configures a system using setup cycles. The system configuration procedure supports the identification of the adapters that reside within a system (by reading the adapter IDs) and the configuration of those adapters (by writing configuration data). Configuration data consists of the following programmable fields:

- Required Fields
  - Adapter ID - the two byte adapter ID
  - Card Enable - a bit that enables or disables the output drivers on the adapter
  - Channel Check - a bit that indicates if the adapter has driven -CHCK active
- Conditionally Required Fields
  - Adapter I/O Address Select - if two or more of the same type of adapter can be installed in a system
  - Adapter ROM/RAM Address Select - if the adapter uses part of the memory address space at hex 000C0000 to 000DFFFF
  - Arbitration level - if the adapter is an arbitrating participant
  - Auto-Incrementing Enable - if the adapter uses POS Registers 6 and 7 as an automatically-incremented register
  - Channel Check Exception Reporting - if the adapter supports synchronous -CHCK for channel non-parity or slave-dependent exceptions
  - Channel Check Status Indicator - if the adapter supports -CHCK
  - Data and Address Parity Enable - if the adapter supports data or address parity

- Device I/O Address Assignment - if the total I/O-address-space required for a type of adapter is greater than 32 bytes
- Matched Memory Enable - if the adapter supports the matched-memory extension
- Selected Feedback Return Exception Enable - if the bus master receives -SFDBKRTN
- Streaming Data Enable - if the adapter supports streaming data procedures

- Optional Fields
  - Option Select Data - customized, adapter-unique configuration data
  - Fairness Enable - optional if the adapter drives -BURST
  - POS Subaddress Extension
  - Status Information - contains the channel-check status or a pointer to the channel-check status.

See the specific-system technical manuals for system configuration information.

Multiple Card Adapters

Multiple-card adapters are adapters that consist of a primary card and one or more secondary cards. The secondary cards are installed in the channel connectors adjacent to the primary card and can be on either, or both, sides of the primary card. The number of secondary cards is limited only by the number of connectors available.

The multiple-card adapter provides only one set of POS registers. Only the primary card responds to -CD SETUP.

The multiple-card adapter operates as a single adapter, and is completely enabled or disabled by the single set of POS registers.

Adapter Configuration Data

After initialization, the system must be able to access the configuration data written into adapter POS registers.

Adapters must always respond to setup cycles. After initialization, the ability to change the configuration data in the POS registers for an adapter is adapter dependent.

Figure 61 shows both the address and field definition. When its -CD SETUP is active, the adapter must provide access to its configuration data. This data is contained in its POS registers and is divided into multiple device-unique, read/write fields.

The following table shows the organization of the I/O address space used for POS operations. Setup functions respond to these addresses only when -CD SETUP is active.

Address (Hex)	Function
XXX0	POS Register 0 - Adapter Identification Byte (Low Byte)
XXX1	POS Register 1 - Adapter Identification Byte (High Byte)
XXX2	POS Register 2 - Option Select Data Byte 1 Bit 0 is designated as Card Enable.
XXX3	POS Register 3 - Option Select Data Byte 2
XXX4	POS Register 4 - Option Select Data Byte 3
XXX5	POS Register 5 - Option Select Data Byte 4 Bit 7 is designated as channel check. Bit 6 is designated as channel-check-status indicator.
XXX6	POS Register 6 - Subaddress Extension (Low Byte)
XXX7	POS Register 7 - Subaddress Extension (High Byte)

Figure 61. POS I/O Address Space

Bits 6 and 7 of POS Register 5 and bit 0 of POS Register 2 are defined by the architecture; all other bits within POS Registers 2 through 5 are available for adapter use.

Required Configuration Fields

The following configuration fields are required for all adapters.

Note: When CHRESET is driven active, adapters must set all fields to the default state.

Adapter ID Field

The Adapter ID field is a two-byte, read-only field contained in POS Register 0 and 1. Each adapter is assigned a unique two-byte adapter ID that is used for adapter identification. An adapter ID of hex 0000 indicates that the adapter is not ready; if an adapter is not present, the adapter ID will be read as hex FFFF.

Card Enable Field

The card enable field is a one-bit, read/write field; it is bit 0 in POS Register 2. When this field is set to 0, the adapter disables the Micro Channel interface except when responding to its setup cycles. When this field is set to 1, the adapter enables all of its Micro Channel interface. The default state of this field is 0.

Channel Check Field

The Channel Check field is a one-bit, read/write field that indicates the adapter has driven -CHCK active; it is bit 7 of POS Register 5. The adapter sets this field to 0 to indicate it has an exception condition, and the system master resets this field to 1 to reset the adapter after handling the error condition. If the adapter has driven -CHCK active, it must set this field to 0 except for address parity exceptions. If the system master resets this field to 1 and the adapter is still driving -CHCK active, the adapter must drive -CHCK inactive.

Note: If this field is set to 0 during a setup cycle, the adapter should not drive -CHCK active.

Conditionally-Required Configuration Fields

The following are the configuration fields that are required when the adapter supports a specified function.

The Channel Check Status Indicator field has a specific location and state because the system master must handle the channel-check condition. Other conditionally-required fields do not have a specific location within POS address space or a specific state.

Note: When CHRESET is driven active, adapters must set all fields to the default state.

Adapter I/O Address Select Field

The Adapter I/O Address Select field is a one- to three-bit field that selects the adapter address decode from the possible address ranges. These bits allow I/O address resolution for up to eight devices of one type in a system. However, if the total I/O address space needed to support the maximum number of devices exceeds 32 bytes, use the Device I/O Address Assignment field (see page 106).

Adapter ROM/RAM Address Select Field

The Adapter ROM/RAM Address Select field is a four-bit, read/write field that determines the starting address of the adapter ROM/RAM address space for the adapter. The adapter address space is defined as the memory address space from hex 000C0000 to 000DFFFF. The four bits represent address bits A16 through A13, which corresponds to 16 consecutive 8KB blocks.

Note: Adapters that require more than 16KB in this memory space can cause addressing conflicts that cannot be resolved during system configuration.

Arbitration Level Field

The Arbitration Level field is a four-bit, read/write field that selects the arbitration level. All adapters that use the arbitration procedure (bursting and nonbursting) must support a selectable arbitration level.

Auto-Incrementing Enable Field

The Auto Incrementing Enable field is a one-bit, read/write field that enables or disables the automatic incrementing of POS Registers 6 and 7. The default state for this field is disable, which causes the adapter to not perform auto-incrementing. When enabled, it causes POS Registers 6 and 7 to automatically increment for each access to POS Register 3 or 4 (see "POS Subaddress Extension Field" on page 108).

Channel Check Exception Reporting Field

The Channel Check Exception Reporting field is a one-bit, read/write field that selects either synchronous or asynchronous -CHCK exception reporting for channel non-parity or slave-dependent exception conditions. When enabled, these exceptions are reported with synchronous -CHCK. When disabled, these exceptions are reported by asynchronous -CHCK. The default state for this field is disabled (see "Exception Reporting" on page 131 for more information on exception reporting).

Channel Check Status Indicator Field

The Channel Check Status Indicator field is a one-bit, read-only field that indicates if channel-check status is available; it is bit 6 of POS Register 5. The default state of this field is 1. This field is required if the Channel Check field is supported by the adapter.

When this field is a 1, it indicates that status for the channel-check condition is not available. When the field is 0 and the Channel Check field is 0, it indicates that POS Registers 6 and 7 contain the channel-check status or a pointer to the status. (If an adapter does not support -CHCK, bit 6 of POS register 5 can be used for other configuration information.)

Data and Address Parity Enable Field

The Data and Address Parity Enable field is a one-bit, read/write field that is required if the adapter supports address or data parity. This field enables address or data parity on the channel. The default state for this field is disabled. While disabled, the device does not drive either -DPAREN or -APAREN active, and does not check parity from the channel. This field can be a two-bit field to individually enable data and address parity.

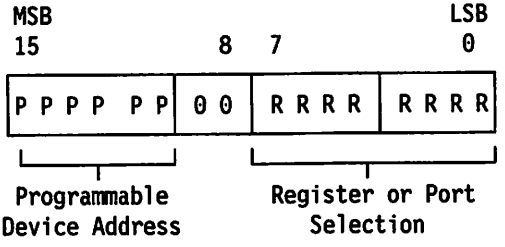
Device I/O Address Assignment

The Device I/O Address Assignment field is a six-bit field that selects the I/O address of the adapter. The six bits represent bits 15 through 10 of the I/O address as shown in the following figure. All I/O devices with a total I/O-address-space requirement greater than 32 bytes must support this field.

The total I/O-address-space requirement for a particular type of device is:

> The number of bytes of I/O address space required by one device multiplied by the maximum number of devices of the same type supported in one system.

The I/O address assigned to the device is:



Address bits 15, 14, and 13 must be in POS registers 2 through 5. Address bits 12, 11, and 10 can be in POS registers 2 through 5 or in the POS extension. The address bits in the POS extension must default to 1 at power on.

Matched Memory Enable Field

The Matched Memory Enable field is a one-bit, read/write field that enables or disables support for the matched-memory signals. The default state of this field is disable, which causes the memory adapter to use the default cycle of the Basic Transfer procedures. When the field is in the enable state, the adapter will support the matched-memory signals.

Selected Feedback Return Exception Enable Field

The Selected Feedback Return Exception Enable field is a one-bit, read/write field that selects whether or not a master reports select-feedback-return exceptions. When the field is in the enable state, the adapter reports select-feedback-return exceptions. When the field is in the disable state, the adapter does not report a select-feedback-return exception. The default state of this field is disable.

Streaming Data Enable Field
The Streaming Data Enable field is a one-bit, read/write field that enables support for streaming data procedures. The default state of this field is disable, which causes the adapter to use the basic data transfer procedures. Adapters that support streaming data procedures must support this bit. When disabled, a slave must tri-state its streaming data signals, and a master must tri-state -SD STROBE. When enabled, the streaming data procedure is supported.

Optional Configuration Fields

The following configuration fields are optional.

Note: When CHRESET is driven active, adapters must set all fields to the default state.

Option Select Data Field

The Option Select Data field is four bytes of adapter unique information in POS Registers 2 through 5 that allow developers to customize configuration data for the specific adapter. Bit 0 of POS Register 2 and bits 7 and 6 of POS Register 5 are defined by the architecture; all other bits within the Option Select Data field are available for adapter configuration.

Fairness Feature Enable Field

The Fairness Feature Enable field is a one-bit, read/write field that enables or disables the fairness feature. The default state for this bit is enabled (see "Fairness Feature and the Inactive State" on page 34).

POS Subaddress Extension Field

The POS Subaddress Extension field is an optional, two-byte field that provides a method to extend the POS register space. It is contained in POS Registers 6 and 7 and is used as an index for all setup-cycle accesses to POS Registers 3 and 4. This field allows the subaddressing of up to 128KB of additional information for each adapter.

The POS extension is an extension to POS Registers 3 and 4 and is made up of two one-byte registers. An index of hex 0000 results in

setup cycles accessing POS Register 3 and 4. All adapters supporting the POS extension must set the index to hex 0000 when CHRESET is driven active. The default value for this field is hex 0000.

Adapters supporting POS extension can use POS Registers 6 and 7 as an index register that is automatically incremented for each access to POS Register 3 or 4. Adapters supporting this feature use the Auto-Incrementing field to enable the feature.

When bits 6 and 7 in POS Register 5 are set to 0, reading POS Registers 6 and 7 will return the channel-check-status information. This information can be status or a pointer to the status.

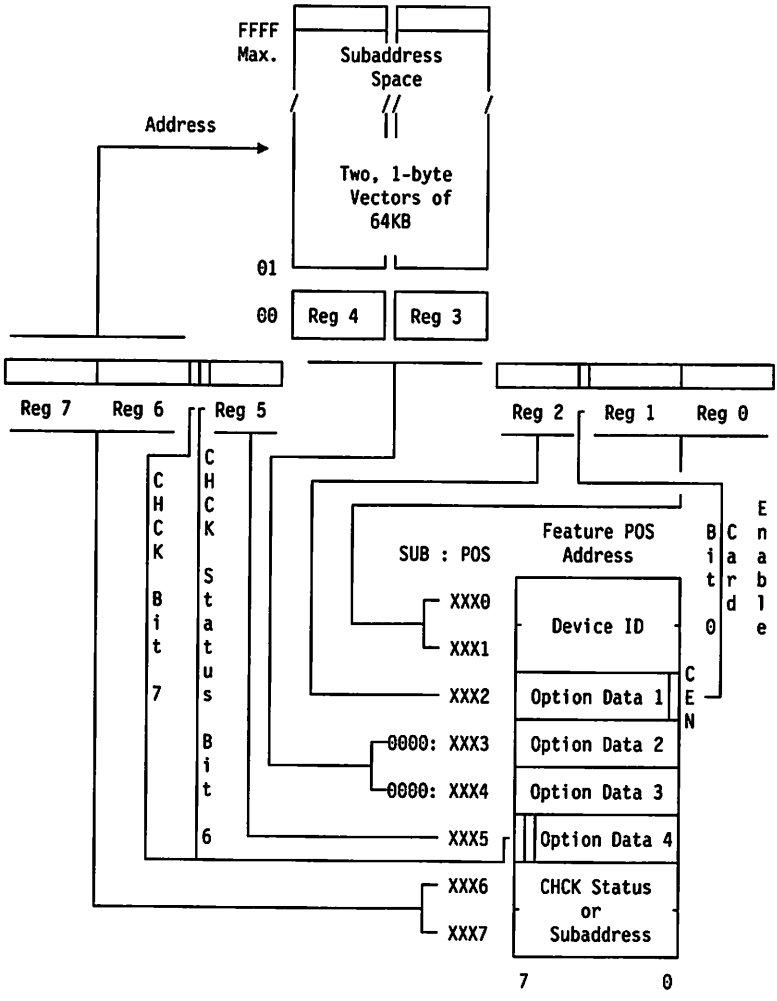


Figure 62. Subaddressing Model

The following is an example of how subaddressing could be used to store or access data in the POS extension.

- POS Register 6 is set to hex 01
- POS Register 7 is set to hex 00
- Access to the POS extension is through POS Register 4
- POS-extension data is in consecutive locations
- Extended POS addressing starts at hex 0001, unless specified by the adapter description file.

Note: The amount of POS-extension information that can be supported is system dependent.

Status Information Field

The Status Information field is contained in POS Registers 6 and 7. The field is used by an adapter to indicate its channel-check status or as a pointer to channel-check status.

System Configuration Procedure

The system configuration procedure uses setup cycles to transfer adapter ID and adapter configuration data between the system master and adapters. The system configuration procedure is similar to the basic transfer procedure, except for the following:

- An adapter is selected by an active -CD SETUP (not by an address decode or arbitration level).
- An adapter must not drive -CHCK active because of a setup cycle.
- A valid POS register address is driven on the address bus; only address bits A(0-2) are decoded by the adapter.
- M/-IO must be in the -IO state during the setup cycle, and adapters must decode M/-IO in the -IO state.
- The adapter does not drive -CD SFDBK active.
- All configuration data transfers are single-byte (8-bit) transfers. All transfers to an 8-bit data port occur on D(0-7). If a 16-bit or 32-bit slave drives -CD DS 16 or -CD DS 32, the 8-bit transfer occurs on the data lines indicated by -SBHE, A0, and A1, or by -BE(0-3) (see Figure 29 on page 53).
- The minimum setup cycle is 300 nanoseconds; adapters can extend the cycle be driving CD CHRDY inactive.

Address Bus During Setup Cycles

During a setup cycle, a valid POS-register address is driven on the address bus; only address bits A(0-2) are decoded by the adapter. The address driven on the bus must not be the I/O address of any adapter.

The following figure shows the organization of the address space used by POS during setup.

-CD SETUP	M/-IO	Address A2 A1 A0	Function
0	0	0 0 0	Adapter ID (Low Byte)
0	0	0 0 1	Adapter ID (High Byte)
0	0	0 1 0	Option Select Data Byte 1
0	0	0 1 1	Option Select Data Byte 2
0	0	1 0 0	Option Select Data Byte 3
0	0	1 0 1	Option Select Data Byte 4
0	0	1 1 0	Subaddress Extension/Channel Check Status (Low Byte)
0	0	1 1 1	Subaddress Extension/Channel Check Status (High Byte)

Figure 63. POS I/O Address Decode

Card Enable Field During Setup Cycles

An adapter selected by -CD SETUP must enable its drivers, regardless of the state of the Card Enable field.

Channel Check Status During Setup Cycles

When -CHCK is driven active, the Channel Check field indicates that the adapter drove -CHCK active (except for address parity). The Channel Check, Channel Check Status Indicator, and Status Information fields (and channel-check status, if the field is used as a pointer) must remain unchanged until the system master resets the Channel Check field to 1 or until CHRESET is driven active.

-CD SETUP During Setup Cycles

During a setup cycle, only one adapter at a time is selected. The adapter is selected by an active -CD SETUP, not by an address decode or arbitration level.

CD CHRDY During Setup Cycles

CD CHRDY can be used to extend the setup cycle, as in the basic transfer procedure (see "Asynchronous-Extended Cycle Timing (≥300 ns)" on page 50 for usage of CD CHRDY in an extended cycle).

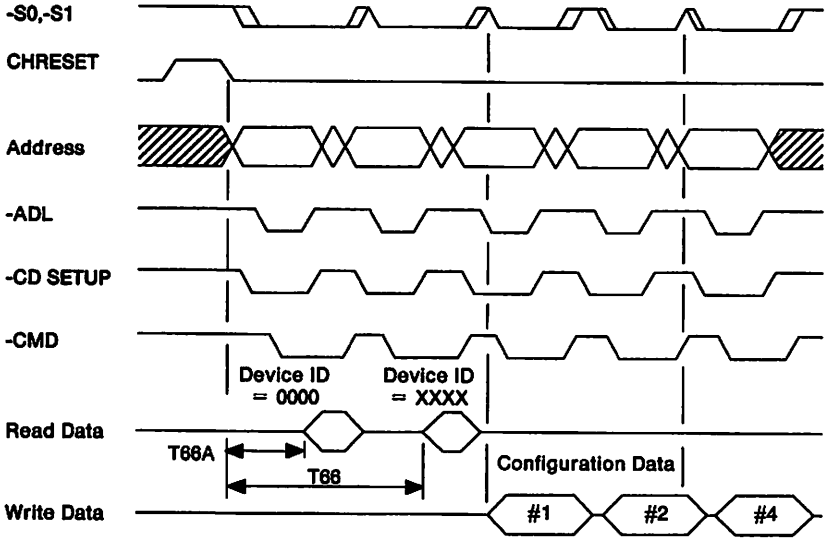


Figure 64. Setup Cycle Overview

System Configuration Timing Diagrams

The following provides the specification for the signal timings used for the system configuration procedure.

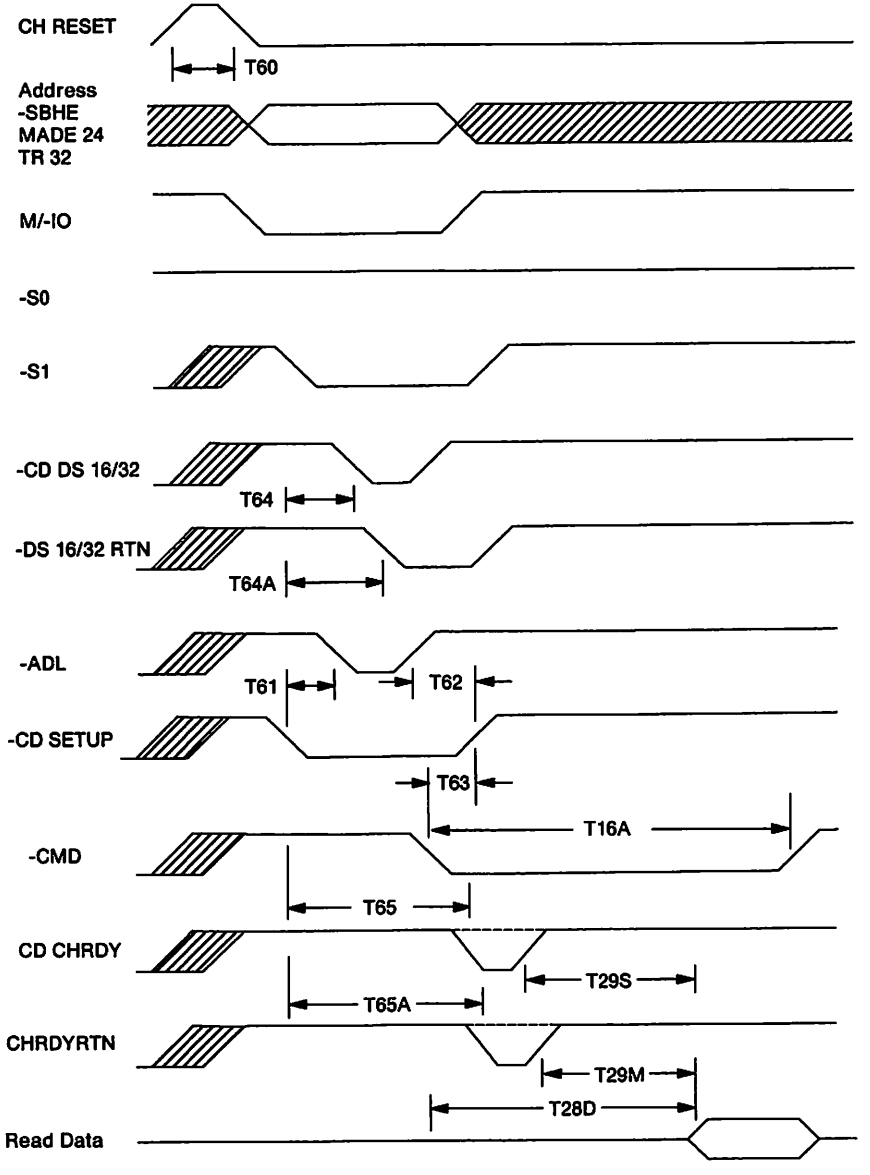


Figure 65. System Configuration Timing

	Timing Parameter	Min/Max	Note
T16A	-CMD Pulse width	190 / - ns	1
T28D	Read data valid from -CMD active	- / 160 ns	1
T29S	Read data from slave valid from CD CHRDY active	- / 60 ns	1
T29M	Read data valid to master from CHRDYRTN active	- / 60 ns	1
T60	CHRESET active pulse width	100 / - ms	
T61	-CD SETUP active to -ADL active	15 / - ns	
T62	-CD SETUP hold from -ADL inactive	25 / - ns	
T63	-CD SETUP hold from -CMD active	30 / - ns	
T64	-CD DS 16 active from -CD SETUP active	- / 25 ns	
T64A	-DS 16 RTN active from -CD SETUP active	- / 45 ns	
T64	-CD DS 32 active from -CD SETUP active	- / 25 ns	5
T64A	-DS 32 RTN active from -CD SETUP active	- / 45 ns	5
T65	CD CHRDY inactive from -CD SETUP active and Status active	- / 100 ns	3
T65A	CHRDYRTN inactive from -CD SETUP active and Status active	- / 120 ns	
T66	Adapter ID valid from trailing edge of CHRESET	- / 1 s	4
T66A	Adapter ID access from trailing edge of CHRESET	1 / - ms	4

Figure 66. System Configuration Timing Specifications

Notes:

1. See "Synchronous-Extended Cycle Timing (300 ns Minimum)" on page 47 and "Asynchronous-Extended Cycle Timing (≥300 ns)" on page 50 for specific details. The following are the timing parameters for the system configuration procedure that are different or additional to those specified for the basic transfer procedure.
2. The minimum setup cycle is 300 nanoseconds. A valid non-card-selecting address must be present on the address bus during setup. During setup, the address on the address bus must not be an I/O address of any adapter.
3. A slave can extend the setup cycle beyond 300 nanoseconds with CD CHRDY.

4. All adapters must present their adapter ID within 1 second after CHRESET is driven inactive. If an adapter is not ready to present its complete adapter ID before T66 time, it must present an ID of hex 0000. If an adapter continues to present an ID of hex 0000 for more than 1 second after CHRESET, it is considered defective.
5. Signals and timings only supported by the 32-bit connector.

Exception Condition Reporting and Handling

An exception condition is an event (including errors) that disrupts normal processing by the master. Resolving an exception usually requires system intervention. Exception conditions are reported to the controlling master or the system master using either -CHCK or an interrupt. Methods for reporting exceptions are described in "Exception Reporting" on page 131.

Note: An adapter whose master function detects exception conditions reports the condition through its I/O slave function by driving -IRQ active.

Exception Conditions

The following types of exception conditions are supported:

- Channel data and address parity exceptions
- Channel non-parity exceptions
- Master- and slave-dependent exceptions
- Channel-timeout exceptions.

Descriptions of each type are included in the following sections.

Channel Data and Address Parity Exceptions

Data and address parity support are optional features on the channel. This type of exception can occur during read or write operations. All participants that support data and address parity checking provide a POS bit (Data and Address Parity Enable field) to enable and disable this feature. The default state of this field is disable. (See "Data and Address Parity Enable Field" on page 106.)

When the Data and Address Parity Enable field is in the disable state, slaves disable checking data and address parity and do not activate -DPAREN during a data-transfer read cycle. Masters disable activation of -DPAREN and -APAREN and do not check data parity on the channel.

Note: Slaves do not disable the driving of -CHCK for non-parity exceptions (independent of the Data and Address Parity Enable field) and masters do not disable the detection of -CHCK.

The description of data and address parity support follows.

Data Parity

Data parity support is optional and allows the use of parity and non-parity adapters on the channel. For exceptions to be detected and reported, both the master and slave involved in the operation must support data parity. -DPAREN is activated by masters on write operations and by slaves on read operations. (-DPAREN is activated by the data source only, and indicates the support of data parity.) The data parity bits, DPAR (0-3), are provided at the same time as data on both write and read cycles, when parity is supported by the master and slave. Only the data bytes used during the data transfer need to have correct parity. For 64-bit streaming, the address parity bits are used during data transfers. -DPAREN is used to indicate that parity is valid on the address bus during 64-bit streaming data transfers. -APAREN remains inactive or is deactivated during the data portion of the transfer.

Read Operations - Basic Transfer

A read cycle with parity checking is shown in Figure 67 on page 121. It is a basic-transfer read cycle that adds data parity checking. During a read cycle, slaves supporting data parity drive -DPAREN active with the data following the activation of -CMD. When -CMD is deactivated, slaves must drive -DPAREN inactive then tri-state it. This ensures that -DPAREN is inactive before a write cycle begins. If a slave fails to restore -DPAREN, adjacent read and write cycles issued from a controlling master that does not support data parity could cause a slave that supports data parity to erroneously assume the write cycle included parity.

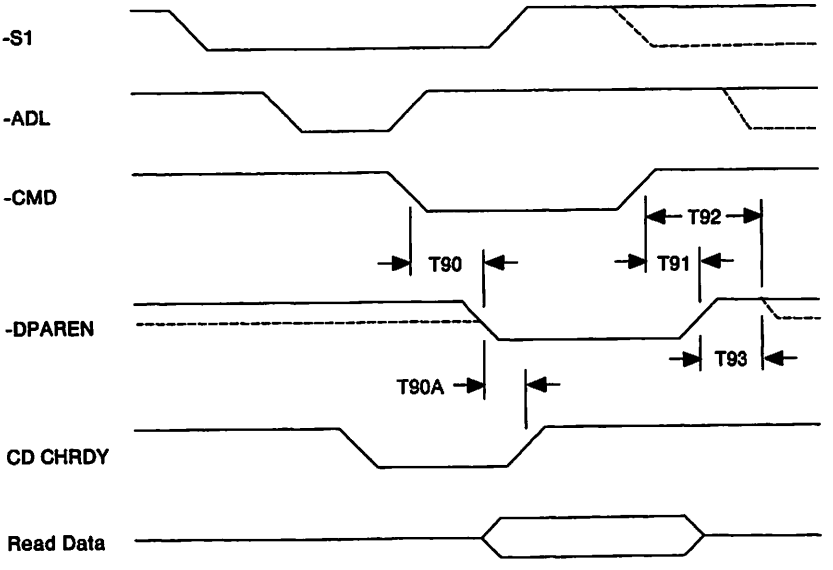


Figure 67. Data Parity Checking Read Cycle - Basic Transfer

	Timing Parameter	Min/Max	Note
T90	-DPAREN active from -CMD active	0 / 60 ns	1
T90	-DPAREN active from -CMD active	0 /160 ns	2
T90	-DPAREN active from -CMD active	0 / - ns	3
T90A	-DPAREN active to CD CHRDY active	- / 0 ns	3
T91	-DPAREN inactive from -CMD inactive	0 / 40 ns	4
T92	-DPAREN tri-stated from -CMD inactive	0 / 80 ns	4
T93	-DPAREN tri-stated from -DPAREN inactive	20 / - ns	

Figure 68. Data Parity Checking Read Timing Specifications

Notes:

1. Default cycle timing only.
2. Synchronous-extended cycle timing only.
3. Asynchronous-extended cycle timing only.
4. -DPAREN must be driven inactive before tri-stating.

Masters supporting data parity must check data parity on read operations, if -DPAREN is activated. If the master detects a parity error, the slave is not informed that the error occurred.

Read Operations - Streaming Data Transfer

Timing of -DPAREN for streaming-data read operations is shown in the following figure.

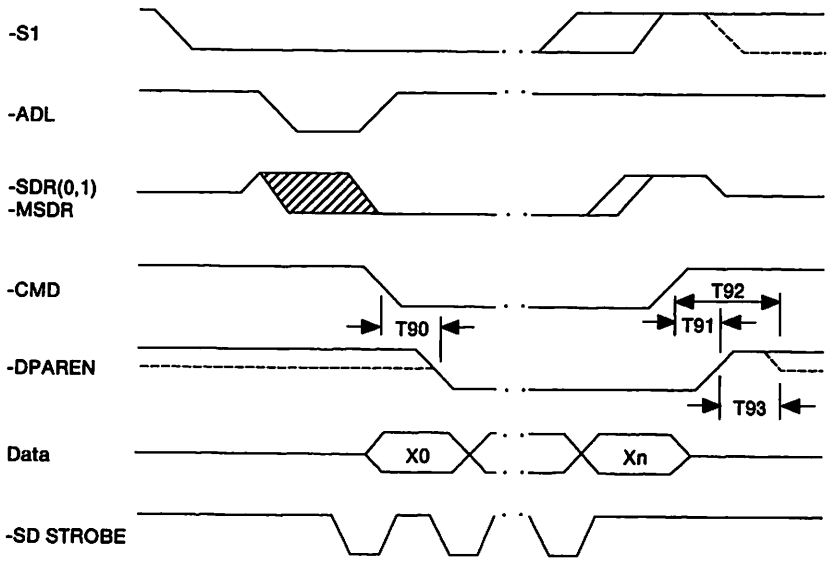


Figure 69. Data Parity Read Cycle - Streaming Data Transfer

Notes:

1. See Figure 68 on page 121 for timing specifications not listed here. For Streaming Data Read Cycles (when CD CHRDY is not driven inactive) T90 is the same as T90 for the basic-transfer cycle.

Write Operations - Basic Transfer

The timing of -DPAREN during write cycles is shown in Figure 70 on page 123. The figure is a basic-transfer write cycle but includes data parity. A master that supports data parity activates -DPAREN during write cycles to indicate that parity is associated with the data. To aid in parity checking at the slave, masters supporting parity are required to provide the data setup time with respect to -CMD. See Figure 70 on page 123 for -DPAREN and T96 timing specifications.

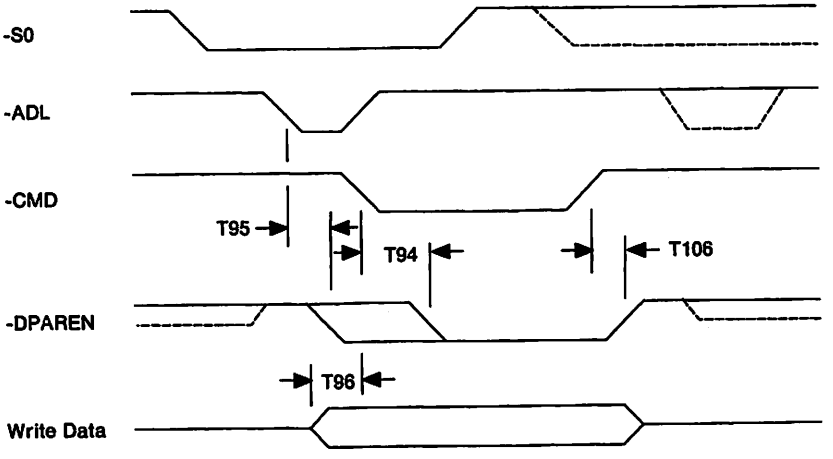


Figure 70. Data Parity Checking Write Cycle - Basic Transfer

	Timing Parameter	Min/Max	Note
T94	-DPAREN active from -CMD active	- / 20 ns	1
T95	-DPAREN active from -ADL active	40 / - ns	2,4
T96	-CMD active from write data setup	15 / - ns	2
T106	-DPAREN inactive from -CMD inactive	20 / - ns	3

Figure 71. Data Parity Checking Write Timing Specifications

Notes:

1. Controlling masters can drive -DPAREN active concurrently with -CMD during write operations. T94 specifies the maximum allowable skew.
2. Applies only to masters supporting data parity.
3. When a write cycle follows a basic-transfer write cycle, the master can optionally hold -DPAREN active between the cycles. When a write cycle follows a streaming data write cycle, the master must drive -DPAREN inactive between cycles. If a master does not hold -DPAREN active between cycles, it must drive -DPAREN inactive following the write cycle. The master must ensure -DPAREN has had time to be restored to the inactive state before sampling its parity check logic during the read cycle. (The master should allow for a 90 ns RC recovery of -DPAREN without an active restore.)

4. Applies only for a master executing a write cycle preceded by a read cycle.
5. See Figure 68 on page 121 for timing specifications not listed here.

Slaves that support data parity, check data parity when -DPAREN is active. In the event of a data parity error during a write, the slave activates -CHCK during the cycle to indicate to the master that an error occurred. Bit 7 in POS Register 5 is set to 0 in the slave when it activates -CHCK. See "Exception Reporting using -CHCK" on page 131 for more information. Slaves do not post an interrupt as a result of detecting a parity error. The slave determines what to do with the data. When sending data to a DMA slave, the DMA controller drives -TC active after detecting -CHCK. Alternatively, the DMA controller can drive -TC active on a subsequent operation with the DMA slave. When the DMA slave detects -TC, it terminates the data transfer and interrupts the system master.

Write Operations - Streaming Data Transfer

The timing of -DPAREN for streaming-data write operations is shown in Figure 72 on page 125. -DPAREN must remain active longer at the end of the cycle. The last data word is not valid until just before the rise of -CMD, and additional time is required for the slave to respond with -CHCK, if a parity error occurred. Slaves use -DPAREN to gate -CHCK onto the channel on streaming-data write operations, and the master must hold -DPAREN active until it can successfully sample -CHCK. Masters that support streaming data transfers with data parity drive -BURST on each transfer and keep it active to delay EOT. The EOT delay prevents conflicts between -DPAREN of the controlling master and the system master, which can execute cycles when ARB/-GNT is in the arbitration state.

The participant receiving the data checks parity on the bytes that are signaled as valid by the data transfer control signals A0, -SBHE or -BE(0-3). The participant driving the data bus (and address bus for 64 bit streaming data) includes parity with the valid data bytes. When the master replicates data bytes for steering purposes, the replicated bytes contain valid parity also. When the system provides data steering the steering includes parity.

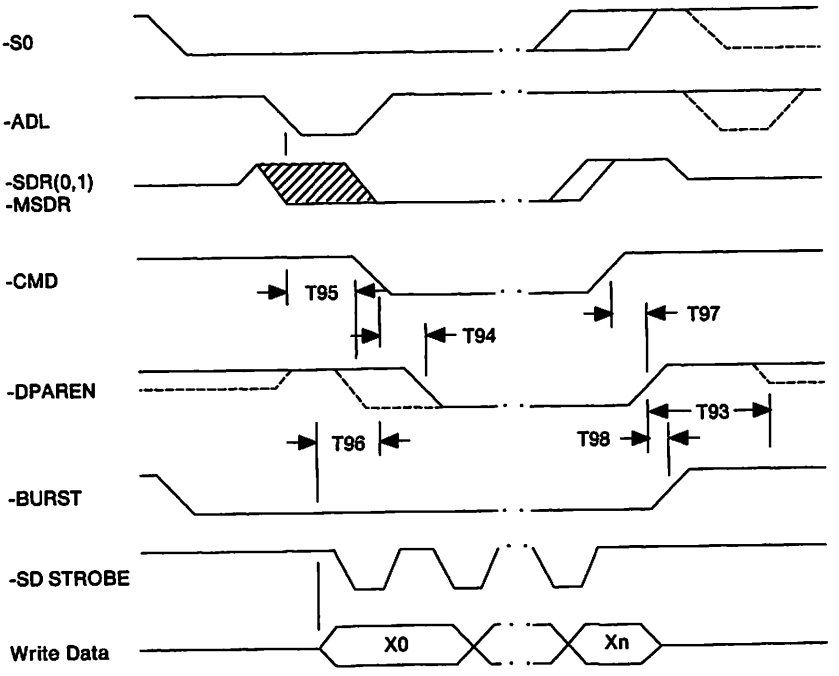


Figure 72. Data Parity Write Cycle - Streaming Data Transfer

	Timing Parameter	Min/Max	Note
T97	-DPAREN hold from -CMD inactive	40 / - ns	1
T98	-BURST inactive from -DPAREN inactive	0 / -	

Figure 73. Data Parity Checking Write Timing Specifications

Notes:

1. Masters must drive -DPAREN inactive before tri-stating it. (See T93 in Figure 67 on page 121).
2. See Figure 67 on page 121 and Figure 70 on page 123 for timing specifications not listed here.

Data Parity Exception Handling

The following is a summary of how participants handle data-parity exception conditions.

Bus Master

During a read cycle a master that supports data parity checking and detects a parity error should do one of the following:

- Suspend the data transfer and post an interrupt reporting a read data parity exception. The master determines what to do with the data.
- Complete the data transfer and post an interrupt reporting a read data parity exception. The master determines what to do with the data.

System Master

When a system master owns the channel and is enabled to support data parity, it must handle data parity errors.

DMA Controller

When a DMA controller that supports data parity detects a data parity error during a read cycle, it should record read data parity exception status and, if the DMA slave is selected, drive -TC active. Alternatively, the DMA controller can drive -TC active on a subsequent operation with the DMA slave.

The system master reads the DMA controller and DMA slave exception status at the end of the DMA operation to determine if the operation completed successfully.

Slaves

When a selected slave detects a parity error during a write cycle, it should activate -CHCK and set bit 7 in POS Register 5 to 0 to indicate that an exception condition occurred (see "Exception Reporting using -CHCK" on page 131). The slave determines what to do with the data.

Address Parity

Address parity support is optional and allows the use of parity and non-parity adapters on the channel. For the exception to be detected and reported, both the master and slave involved in the operation must support address parity.

A controlling master that supports address parity drives -APAREN active to indicate that the parity is associated with the address. When a slave that supports address parity finds -APAREN active and an address parity error is detected, the slave is not selected. The slave includes address parity checking in its address decode. The following figure shows address parity timing requirements.

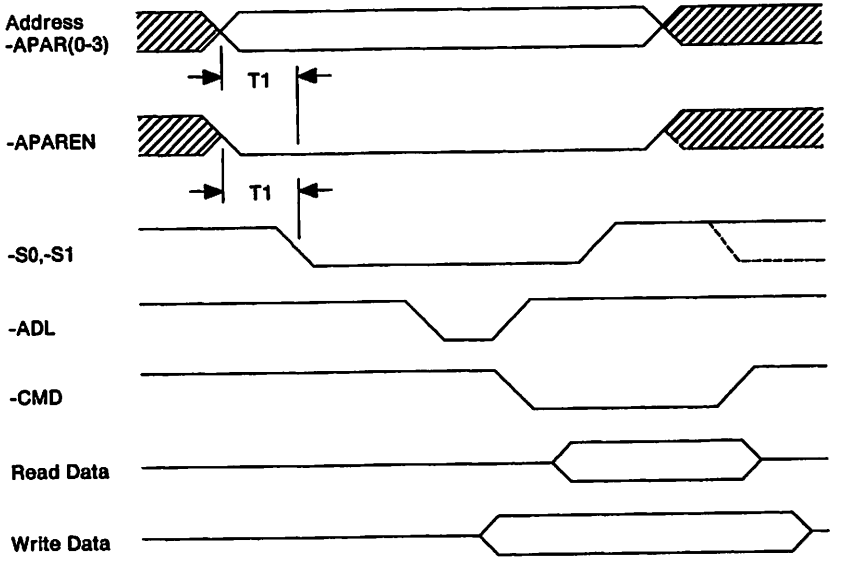


Figure 74. Address Parity

	Timing Parameter	Min/Max	Note
T1	Status active from Address Bus, M/-IO, -APAREN, and -REFRESH valid	10 / - ns	

Figure 75. Address Parity Timing Specifications

-APAREN and address parity bits are valid during the same interval that the address is valid (see Figure 23 on page 44). Only those address bytes used in the data-transfer cycle need to have correct parity. (See data parity for use of address parity bits during 64-bit streaming-data transfers).

Slaves that support address parity must include parity checking in their address decode logic so that, when parity errors occur, they do not become selected and do not drive -CD SFDBK. Slaves check address parity for all data transfers that have -APAREN active. Slaves that support address parity check each address and drive -CHCK active when an address parity error is detected. I/O slaves check parity on all I/O addresses, and memory slaves check parity on all memory addresses. They do not set the channel check status bit in POS Register 5.

Note: A slave treating the address bus as a command should ensure address parity is good before allowing the command to execute.

Channel Non-Parity Exceptions

These exceptions are channel exceptions other than data and address parity exceptions. These exceptions can be signaled by the current slave or detected by the master.

Invalid -DS 16 RTN and -DS 32 RTN Combination

If the system master or a bus master detects an inactive -DS 16 RTN and an active -DS 32 RTN, it suspends the data transfer and posts an interrupt, indicating with interrupt status that the operation was not valid. Detecting the invalid operation is optional.

The DMA controller drives -TC active during the cycle of the detected exception when the DMA slave is selected, and records the status of the invalid operation. Alternatively, the DMA controller can drive -TC active on a subsequent operation with the DMA slave. The timings for -TC are listed in Figure 45 on page 78, Figure 51 on page 89, and Figure 53 on page 90.

Selected Feedback Return Exception

Bus masters monitor -SFDBKRTN under control of a Selected Feedback Return Exception Enable field. Masters that do not support -SFDBKRTN do not need this POS bit. When the Selected Feedback Return Exception Enable field is enabled and -SFDBKRTN is inactive, the bus master sets appropriate status, suspends the data transfer and interrupts the system master.

The DMA controller monitors -SFDBKRTN under control of a Selected Feedback Return Exception Enable field. DMA controllers that do not support -SFDBKRTN do not need this POS bit. When the Selected Feedback Return Exception Enable field is enabled and -SFDBKRTN is inactive, the DMA controller sets the appropriate status and, if a DMA slave is selected, drives -TC active for that cycle. Alternatively, the DMA controller can drive -TC active on a subsequent operation with the DMA slave.

Systems can monitor the -CD SFDBK signals, or -SFDBKRTN, under control of the Selected Feedback Return Exception Enable field. When the Selected Feedback Return Exception Enable field is enabled and a -CD SFDBK is not returned, the system monitor should set appropriate status and generate an interrupt.

These types of exceptions can occur during either read or write operations.

Note: Masters must record the state of -SFDBKRTN when a channel check is detected regardless of the state of the Select Feedback Return Exception Enable field.

Master and Slave Dependent Exceptions

Master- and slave-dependent exceptions are not related to the channel but to the participants. Reporting of this type of exception is specific to each participant and to the specific exception condition.

This type of exception can occur during either read or write operations. A channel check can also occur. Therefore, a master must monitor -CHCK for exceptions during both read and write operations. This is required regardless of whether address parity or data parity are supported.

Master- and DMA slave-dependent exceptions should be reported with an interrupt. However, certain slave-dependent exceptions might require using -CHCK to report the exception. Examples of exceptions that could require using -CHCK are:

- Page fault
- Internal parity error (known bad data without an interrupt procedure to report the error)
- Command queue overflow
- Catastrophic exception condition.

Channel Timeout Exception

If a controlling master or DMA slave fails to release the channel within 7.8 microseconds of -PREEMPT going active, the central arbitration control point can drive ARB/-GNT to the ARB state. In this state, the controlling master degates the address and data buses and all data transfer signals.

Note: Reporting and handling of this type of exception is system dependent.

Exception Reporting

A slave normally reports exceptions with synchronous -CHCK. A master normally reports exceptions with an interrupt and termination of current bus ownership. Exception reporting using -CHCK is described in the following information.

Exception Reporting using -CHCK

Exceptions detected by a slave or channel monitor are usually reported by activating a synchronous -CHCK. If -CHCK is activated, the system master or a bus master suspends processing and generates an interrupt. This interrupt invokes a utility program to handle the exception condition.

Any adapter that activates -CHCK sets bit 7 in POS Register 5 to a 0, to indicate the source of the exception in all cases except address parity errors. If multiple conditions can set -CHCK, the adapter can optionally set status information indicating the cause of the exception. The status or a pointer to it should be placed in POS Registers 6 and 7. Bit 6 in POS Register 5 is then set to 0 to indicate that status information is available. The system master can reset the -CHCK indication by setting a logical 1 into bit 7 in POS Register 5. Setting the bit to 0 will produce unpredictable results. Adapters that do not support -CHCK must ensure bit 7 in POS Register 5 is interpreted as a logical 1.

Synchronous and asynchronous reporting of exceptions using the -CHCK signal are supported. Both reporting methods are described in the following sections.

Synchronous -CHCK

Synchronous -CHCK is signaled by a slave or channel monitor with a pulse on -CHCK. The conditions that are reported with synchronous -CHCK are:

- Data parity errors
- Address parity errors
- Channel non-parity exceptions
- Slave-dependent exceptions.

Note: Slaves or channel monitors that report channel non-parity or slave-dependent exceptions with synchronous -CHCK must support the Channel Check Exception Reporting field (see "System Configuration" on page 101 for a description of this field).

Figure 76 shows a synchronous-exception presentation for the basic transfer cycles.

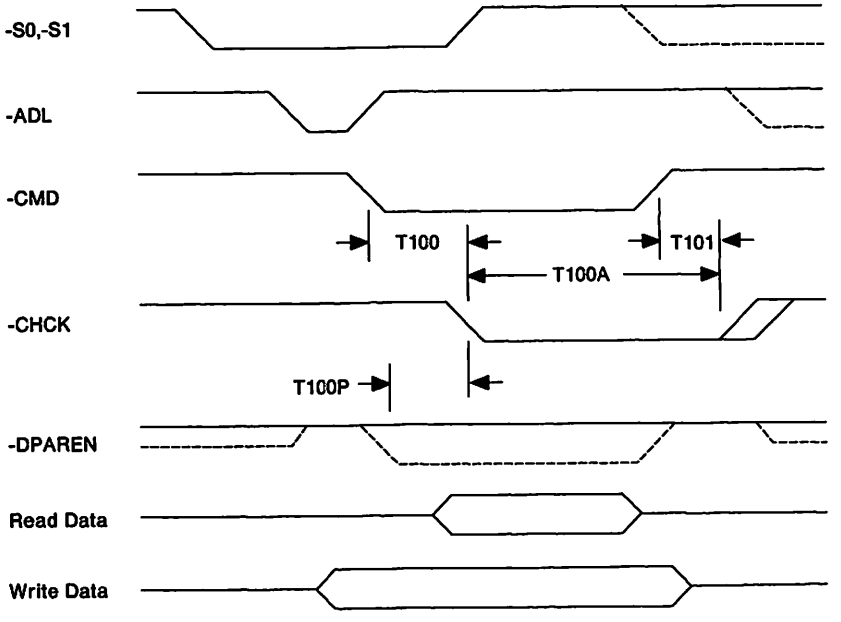


Figure 76. Channel-Check - Default Cycle

	Timing Parameter	Min/Max	Note
T100	-CHCK active from -CMD active	0 / 50 ns	1
T100A	-CHCK pulse width	50 / - ns	1
T100P	-CHCK active from -CMD and -DPAREN active	0 / 50 ns	2
T101	-CHCK disabled from -CMD inactive	0 / 40 ns	

Figure 77. Channel-Check Timing Specifications - Default Cycle

Notes:

1. Slave exception reporting for exceptions other than channel data parity.
2. Activation of -CHCK because of a write-data parity error is timed from -CMD and -DPAREN active, whichever occurs last.
3. Using synchronous -CHCK to report an exception during a transfer that is caused by a condition other than those listed is timed from the falling edge of -CMD.

In a default cycle, the slave activates -CHCK from the activation of -CMD or -CMD and -DPAREN. If the slave requires additional time, it can extend the cycle by deactivating CD CHRDY. Figure 78 shows an extended bus cycle.

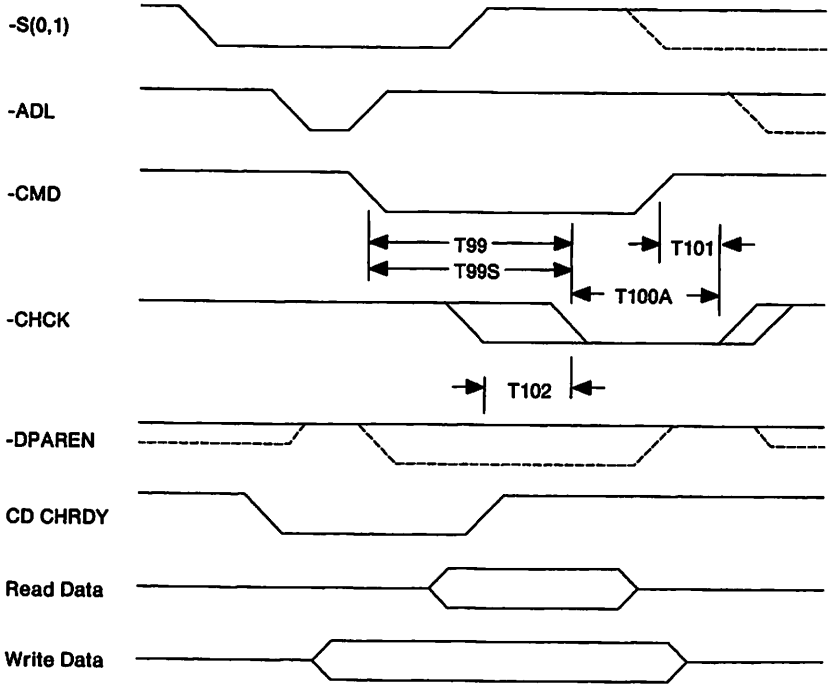


Figure 78. Channel-Check - Extended Cycle

	Timing Parameter	Min/Max	Note
T99	-CHCK active from -CMD active	0 / - ns	2
T99S	-CHCK active from -CMD active	0 /150 ns	1
T102	-CHCK active from CD CHRDY active	- / 50 ns	2

Figure 79. Channel-Check Timing Specifications - Extended Cycle

Notes:

1. For a synchronous-extended cycle only.
2. For an asynchronous-extended cycle and streaming data.
3. See Figure 77 on page 133 for timing specifications not listed here.

Presentation of exceptions for streaming data cycles is similar to a basic transfer cycle. Since exceptions can also occur on any of the data cycles, -CHCK can be activated at any time during the cycle. During read operations, -CHCK is gated by -CMD and is required to be valid coincident with the final data word (Figure 80). During streaming data write operations, -CHCK is gated by -DPAREN and this signal is held active beyond the rise of -CMD to allow time for -CHCK to be activated on the channel (see Figure 72 on page 125).

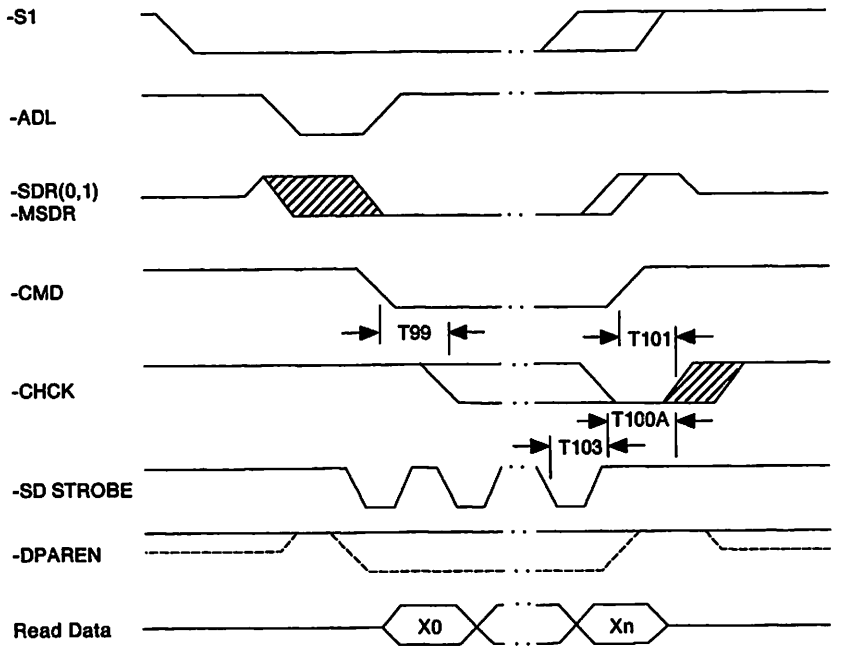


Figure 80. Channel Check - Streaming Data Read Cycle

	Timing Parameter	Min/Max
T103	-CHCK active from last -SD STROBE active	- / 50 ns

Figure 81. Channel-Check Timing Specifications - Streaming Data Read Cycle

Note: See Figure 77 on page 133 and Figure 79 on page 134 for timing specifications not listed here.

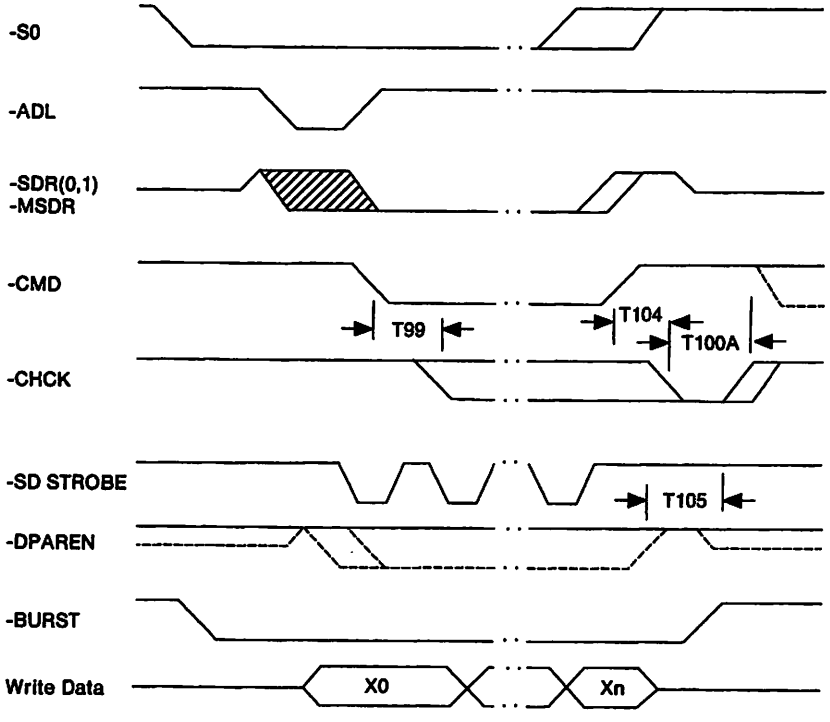


Figure 82. Channel Check - Streaming Data Write Cycle

	Timing Parameter	Min/Max
T104	-CHCK active from -CMD inactive	- / 40 ns
T105	-CHCK disabled from -DPAREN inactive	0 / 40 ns

Figure 83. Channel-Check Timing Specifications - Streaming Data Write Cycle

Note: T100A, T101, and T103 apply to -CHCK for all conditions other than data parity exceptions.

Asynchronous -CHCK

An asynchronous -CHCK is indicated by driving -CHCK active asynchronously to a data transfer. -CHCK remains active until reset by the system master. The adapter that activates an asynchronous -CHCK sets bit 7 in POS Register 5 to a 0. The asynchronous -CHCK is reset by the system master setting bit 7 in POS Register 5, to a 1 (in the adapter that activated -CHCK). The time that asynchronous -CHCK remains active is system master dependent.

Because asynchronous exceptions are latched, -CHCK is detected during the reading of the adapter configuration registers to determine the source of the exception. System masters should ignore -CHCK while collecting status associated with the signaling of -CHCK.

The conditions that can be reported with asynchronous -CHCK are:

- Channel non-parity exceptions
- Slave-dependent exceptions.

If the exception cannot be reported with an interrupt or a synchronous -CHCK and the condition would otherwise go undetected, the exception can be reported with an asynchronous -CHCK. **Using asynchronous -CHCK can cause momentary suspension of all I/O activity and could require a system reset.**

Exception Reporting Using Interrupts

If an interrupt is used to report an exception, status must be supplied to indicate the type of exception being reported.

- The minimum status for bus masters is:
  - Data parity error during a read operation
  - -CHCK state
  - -SFDBKRTN state (if received).
- The minimum status for slaves is:
  - Data parity error during a write operation
  - Internal error.

Exception Handling

The following describes how participants handle -CHCK conditions.

Bus Master

The bus master monitors -CHCK when it is the controlling master. When the bus master detects -CHCK, it should do one of the following:

- Suspend the data transfer and post an interrupt.
- Complete the data transfer and post an interrupt.

Data sent by the bus master after it detects -CHCK might not be received. Data received by the bus master after it detects -CHCK might not be valid.

System Master

The system master monitors -CHCK when it is the controlling master. If -CHCK is detected, the system master will handle the exception condition. The system master can monitor -CHCK at all other times also. The action taken is system dependent.

The method for handling exceptions is software dependent and will include setting the -CHCK status bit to 1 in participants that have activated -CHCK, if exception recovery is part of the action taken.

DMA Controller

A DMA controller monitors -CHCK when it is the controlling master. When it detects -CHCK active, the DMA controller should record channel-check status, record the state of -SFDBKRTN, and, if the DMA slave is selected, drive -TC active. Alternatively, the DMA controller can drive -TC active on a subsequent operation with the DMA slave.

The system master reads the DMA controller and DMA slave exception status, when the end of the DMA operation is signaled, to determine if the operation completed successfully.

Slaves

When a slave activates -CHCK, it must set bit 7 in POS Register 5 to a 0 (except for address parity checking). The system master, upon resolving the -CHCK exception, will reset the bit to a 1.

Select Feedback Exception Handling

A -SFDBKRTN exception occurs when the controlling master does not detect -SFDBKRTN active (or -CD SFDBK for system masters). Reporting of this exception is optional. However, all masters that receive -SFDBKRTN must record its state when an exception occurs. Reporting -SFDBKRTN exceptions is controlled by the Select Feedback Return Exception Enable field. If the master does not receive the signal, the POS field and the monitoring of the signal is not required.

Note: All masters that support address parity must receive -SFDBKRTN.

The following describes how participants handle a -SFDBKRTN exception.

Bus Master

A bus master that is enabled to report a -SFDBKRTN exception and does not receive a -SFDBKRTN should do one of the following:

- Suspend the data transfer and post an interrupt
- Complete the data transfer and post an interrupt.

The status provided with the interrupt includes the lack of -SFDBKRTN. Data sent by the bus master after it detects the lack of -SFDBKRTN might not be received. Data received by the bus master after it detects the lack of -SFDBKRTN might not be valid.

System Master

Reporting a -SFDBKRTN exception and handling the exception is system dependent.

DMA Controller

A DMA controller that is enabled to report a -SFDBKRTN exception and receives no -SFDBKRTN should record status and, if the DMA slave is selected, drive -TC active. Alternatively, the DMA controller can drive -TC active on a subsequent operation with the DMA slave.

At the end of the DMA operation the system master reads the DMA controller and DMA slave status to determine if the operation completed successfully.

Channel Reset

CHRESET is used to force all system functions and channel participants to a defined state. The following situations cause a channel reset.

- A programmed-controlled channel-reset request
- A system power-on reset, which occurs:
  - When the system is powered on
  - When a low-voltage condition is detected at the input of the power supply.

CHRESET is active for a minimum amount of time, after all voltages in the system are within operating limits. See Figure 66 on page 116 for the T60 timing specifications.

After a channel reset or a system power-on reset, the state of the Micro Channel buses and signals are as follows:

	System Power-On Reset		Channel Reset	
Buses and Signals	**Adapter**	**System**	**Adapter**	**System**
Address Bus and Associated Signals	Tri-state	Initial	Tri-state	Unknown
Data Bus and Associated Signals	Tri-state	Initial	Tri-state	Unknown
Arbitration Bus and Associated Signals	Tri-state	Initial	Tri-state	Unknown
IRQ signal	Inactive	Initial	Inactive	Initial
Channel Check Signal	Inactive	Initial	Inactive	Initial
All Others	Inactive	Initial	Inactive	Unknown

Figure 84. Micro Channel Bus and Signal Status after a CHRESET

Connector Definition

The channel provides all signal, power, and ground signals to adapters through 50-mil channel connectors.

The channel provides two basic types of connectors:

- 16-bit, which support 8- and 16-bit operations
- 32-bit, which support 8-, 16-, 24-, and 32-bit operations.

Pins 01 through 45 support 8-bit operations. Pins 46 and 47 are keys (physical notch). Pins 48 through 58 provide additional power and signals to support 16-bit operations. For 32-bit adapters that function in 16-bit connectors, pins 59 and 60 are keys; for 32-bit adapters that cannot function in 16-bit connectors, pins 59 and 60 are reserved. Pins 61 through 89 are used with pins 01 through 58 to support 32-bit operations.

The power and ground pins on side A of each connector are offset from side B by 2 pins, and every fourth pin on either side of each connector is an ac ground.

Video and matched-memory extensions to the basic 16- and 32-bit connectors are implemented on a system-by-system basis. For more information, see the technical reference for the specific system.

Note: Adapter designs should not extend the card-edge connector beyond the basic 16- or 32-bit connector unless the signals provided by the extension are used by the adapter.

All of the logic signal lines are transistor-transistor logic (TTL) compatible.

© Copyright IBM Corp. 1989, 1990

The following figure shows the basic types of channel connectors with optional extensions.

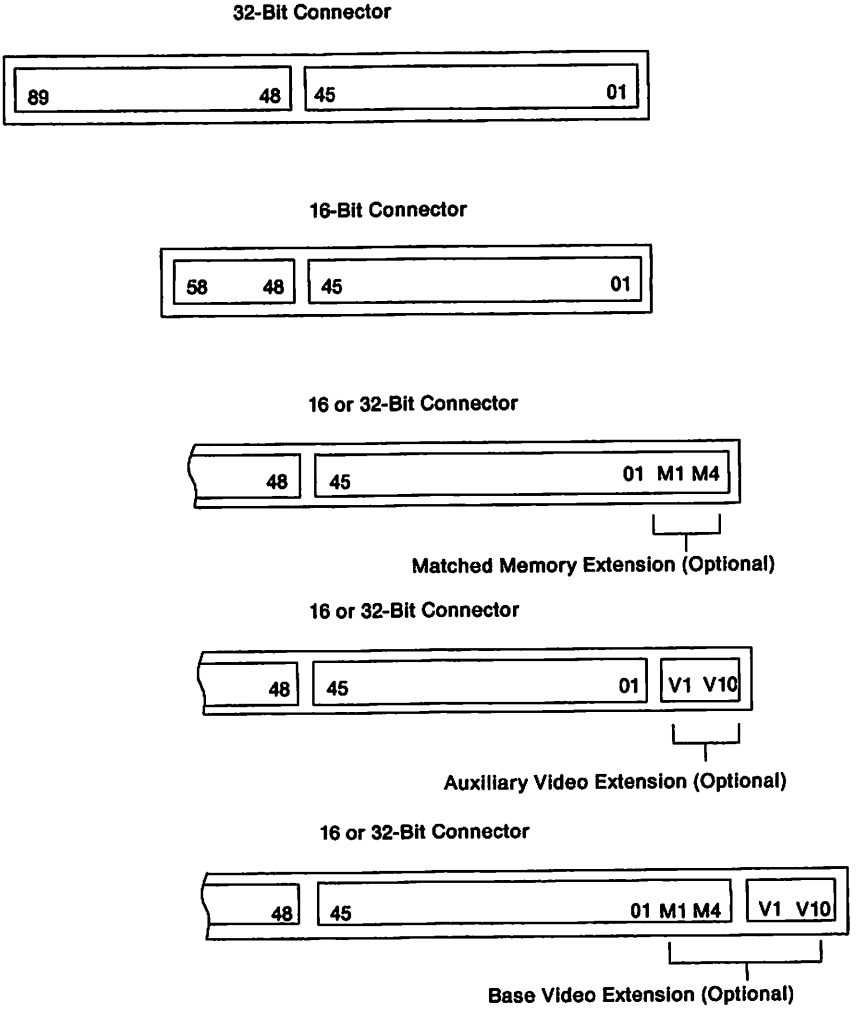


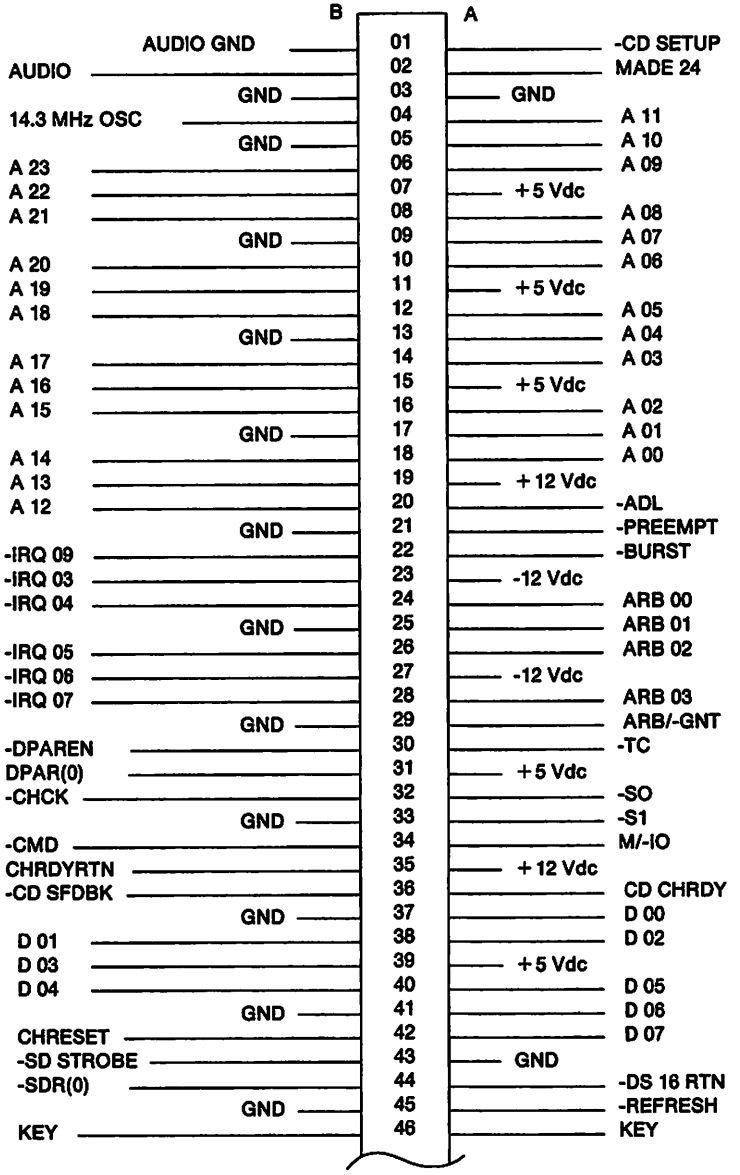
Figure 85. Micro Channel Connectors

Warning: Any signals shown or described as "Reserved" should not be driven or received. These signals are reserved to allow compatibility with future implementations of the channel interface. Serious compatibility problems, loss of data, or permanent damage can result to features or the system, if these signals are misused.

Micro Channel Connector

The following three figures show the signals and the voltages assigned to the 32-bit channel connector. The 16-bit connector is a subset of the 32-bit connector consisting of pins 1 through 58. A key is provided at pin locations 46 and 47 for mechanical alignment.

Rear of the System Board



| Figure 86. Micro Channel Connector

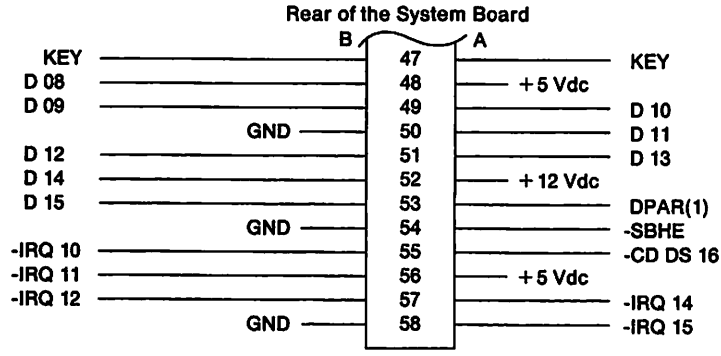
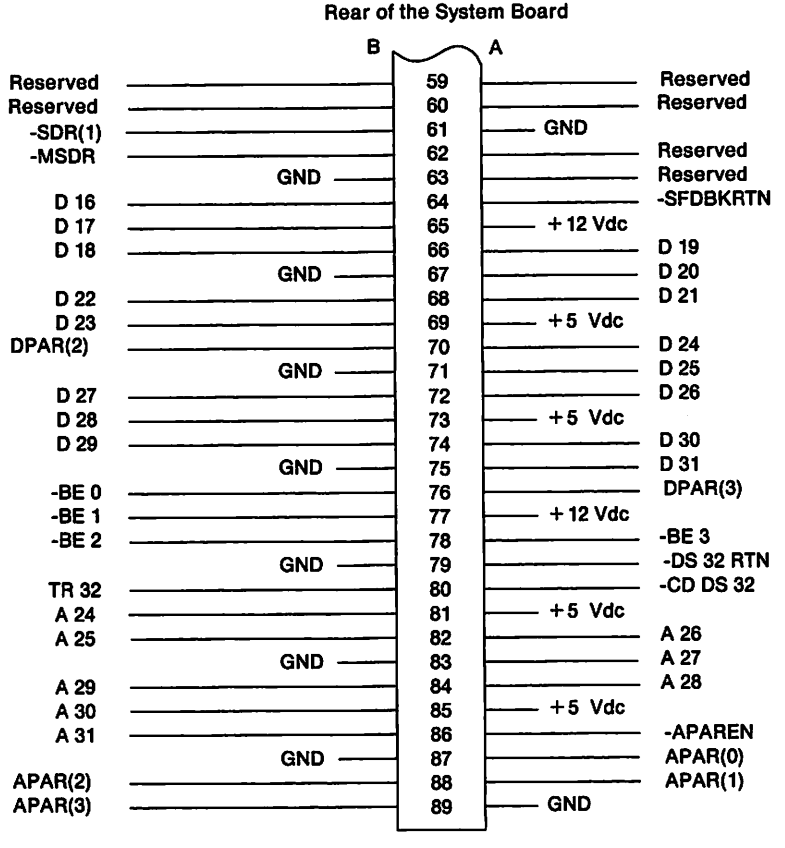


Figure 87. Micro Channel Connector - continued



| Figure 88. Micro Channel Connector - continued

Matched-Memory Extension

This extension provides additional signals to accommodate matched-memory cycles. The following figure shows a connector with a typical set of matched-memory signals. Refer to the system-specific technical reference for the system you are dealing with for further information.

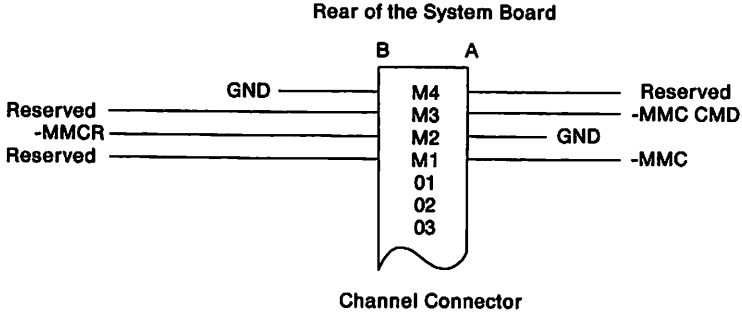


Figure 89. Channel Connector Voltage and Signal Assignments (Matched-Memory Extension)

Auxiliary Video Extension

This extension provides a video adapter with access to resources of the base video subsystem and allows it to extend or supplement the functions provided.

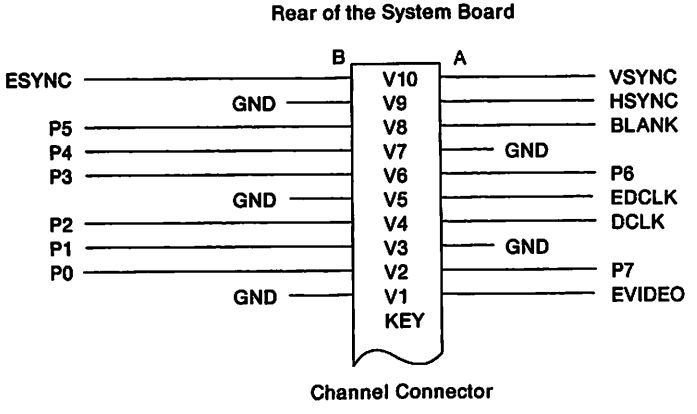
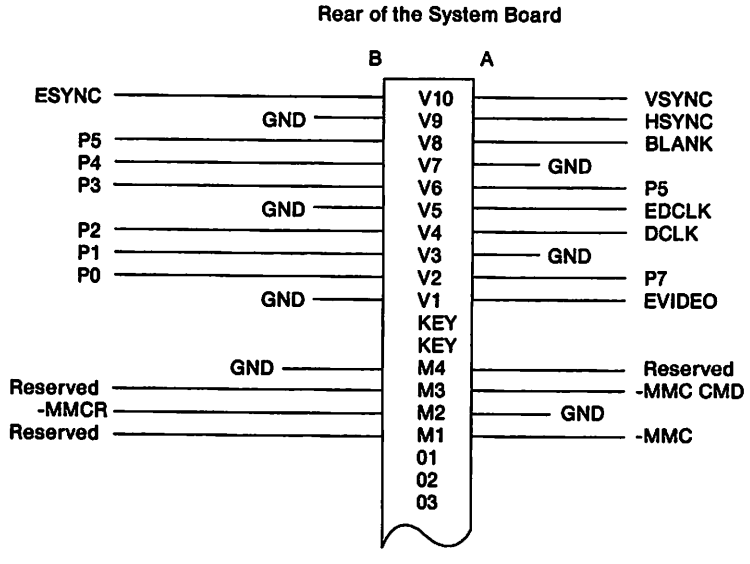


Figure 90. Auxiliary Video Extension

Base Video Extension

This extension is for adapters that provide the base video subsystem. Only systems without a base video subsystem on the system board have a connector with this extension.



Notes:

1. The KEY between M4 and V1 is a physical key.
2. Support of the matched-memory signals (M1 - M4) is optional.

Figure 91. Base Video Extension

Adapter Physical Specifications

The following are the physical specifications for Micro Channel adapters; these adapters can be Type 3, Type 3 half-card, or Type 5. The type number identifies the dimensions of the adapter. Systems specify the adapter type that they support.

Systems that specify support for the Type 3 adapter do not provide the space or the current capacity to accommodate Type 5 adapters (but can accommodate Type 3 half-card adapters). Systems that specify support for the Type 5 adapter support the Type 3, Type 3 half-card, and Type 5 adapters.

Adapter Size

The following are overall dimensions for Micro Channel adapters. Additional and more detailed information (plus tolerances) is available in Micro Channel Adapter Design.

Type 3 Adapter

- Card height - 3.475 in.
- Card plus retainer length - 12.3 in.

Type 3 Half-card Adapter

The Type 3 half card adapter does not require a retainer.

- Card height - 3.475 in.
- Card plus retainer length - 6.35 in.

Type 5 Adapter

The Type 5 adapter does not require a retainer.

- Card height - 4.825 in.
- Card length - 13.1 in.

Connector Spacing

- Minimum Spacing from Card Center to Card Center - 0.85 in. (Maximum spacing is system dependent.)
- Maximum Component Height (Measured from card surface.)
  A side (without electrical insulation) = 0.6 in
  A side (with electrical insulation) = 0.675 in
  B side = 0.105 in.

Electrical insulation is also required if the card bows in excess of 0.025 inch. Measure the bow at a point 0.1 inch from the top of the card and 5.75 inches from the connector end (rear) of the card.

Power

The following figures list the minimum available load current for each voltage present on each connector. Some systems can provide more current than the amount specified below. For more information, refer to the system-specific technical manual.

Supply Voltage	Minimum Available Current Per 16-Bit Connector	Minimum Available Current Per 32-Bit Connector
+ 5.0 Vdc	1.6 A	2.0 A
+12.0 Vdc	0.175 A	0.175 A
−12.0 Vdc	0.040 A	0.040 A

Figure 92. Minimum current available for the Type 3 adapter.

Supply Voltage	Minimum Available Current Per 16-Bit Connector	Minimum Available Current Per 32-Bit Connector
+ 5.0 Vdc	3.12 A	3.12 A
+12.0 Vdc	0.25 A	0.25 A
−12.0 Vdc	0.10 A	0.10 A

Figure 93. Minimum current available for the Type 5 adapter

The following formulas can be used to determine maximum *statistical* current values.

$$(I_{TC_1} + \cdots + I_{TC_n}) + \sqrt{((I_{MC_1} - I_{TC_1})^2 + \cdots + (I_{MC_n} - I_{TC_n})^2)}$$

Where:

- I_{MC} is the maximum current for each component on a given adapter.
- I_{TC} is the typical current for each component on a given adapter and the sum of all I_{TC} equals I_{TA}.

Note: If I_{MC} or I_{TC} is not available, estimate by using: $I_{TC} = 0.7 \times I_{MC}$

The total channel current is also determined in a statistical manner.
Total Channel Current =

$$(I_{TA_1} + \cdots + I_{TA_n}) + \sqrt{((I_{MA_1} - I_{TA_1})^2 + \cdots + (I_{MA_n} - I_{TA_n})^2)}$$

Where:

- I_{MA} is a maximum statistical current for a given adapter.
- I_{TA} is the typical current for a given adapter.

Voltage Regulation

The voltage regulation at the channel connector is shown in the following table.

Voltage	Pins	Tolerance
Ground	A3, B3, B5, B9, B13, B17, B21, B25, B29,	N/A
	B33, B37, B41, A43, B45, B50, B54, B58	N/A
	A61, B63, B67, B71, B75, B79, B83, B87, A89, BM4*, AM2*	N/A
+ 5.0 Vdc	A7, A11, A15, A31, A39, A48, A56, A69, A73, A81, A85	+5% −4.5%
+12.0 Vdc	A19, A35, A52, A65, A77,	+5% −4.5%
−12.0 Vdc	A23, A27	+9.5% −10% (−10.86 to −13.2 Vdc)
Ground (Auxiliary Video)	BV1, AV3, BV5, AV7, BV9	N/A

* These pins are in the Matched-Memory Extension.

Figure 94. Channel Voltage Regulation

The tolerance includes all power distribution losses in both power and ground planes, up to the pins of the channel connector. It does ***not*** include the drop due to the connector (30 milliohm maximum per contact), or the drop due to distribution within the adapter.

Index

A

adapter configuration data 103
adapter ID field 104
adapter I/O address select field 105
adapter ROM/RAM address select field 105
adapters, multiple card 102
address boundary alignment, streaming transfer 64
address bus signals 9
address bus translator 55
address decode latch signal 9
address management, streaming data 63
address parity 127
address parity enable field 106
address parity enable signal 9
address parity signals 10
address space, POS 103
addressing model 6
ADL 9
APAREN 9
APAR0 – APAR3 10
arbiter logic 31
arbiter mismatch 30
arbitration 27
arbitration block diagram 29
arbitration bus 7
arbitration cycle 35
arbitration level field 105
arbitration timing 35
arbitration/grant signal 8
ARB0 - ARB3 7
ARB/-GNT 8
architecture extensions 26
asynchronous -CHCK 137
asynchronous-extended cycle 50
AUDIO 19
AUDIO GND 19
audio ground 19
audio sum node signal 19
auto-increment enable field 105
auxiliary video extension 149
auxiliary video signal 23
A0 - A31 9

B

base architecture, 16-bit 26
base architecture, 32-bit 25
base video extension 149
basic transfer 40
BE signals 55, 56
BE0 - BE3 13
burst DMA transfer 90
burst mode 39
burst signal 8
burst transfer 8, 32, 86
burst transfer, preempted 33
bus master 3
bus, arbitration 7
byte enable 55
byte enable signals 13

C

card data size 16 signal 14
card data size 32 signal 14
card enable field 104
card selected feedback signal 10
card setup signal 19
CD CHRDY 14
CD DS 16 14
CD DS 32 14
CD SETUP 19
CD SFDBK 10

© Copyright IBM Corp. 1989, 1990

central arbitration control point 28
central data steering 56
central steering logic 56
central translator logic 56
channel check field 104
channel check signal 19
channel connector definition 142
channel connector diagram 143
channel connector voltages and signal assignments 144
channel connectors 142
channel definition 7
channel load current 152
channel ownership 27
channel ready return signal 14
channel ready signal 14
channel reset 141
channel reset signal 19
channel signal groups 20
channel support 55
channel time-out 34
channel timeout exception 130
CHCK 19
CHCK field 104
CHRDYRTN 14
CHRESET 19
CHRESET signal 141
CMD 14
command signal 14
conditionally-required fields 104
configuration data 103
configuration fields 103
configuration, system 101
connectors, channel 142
connector, auxiliary video extension 149
connector, base video extension 149
connector, matched-memory extension 148
connector, 16 bit 144
connector, 32 bit 144
controlling master 27
cycle, arbitration 35
cycle, asynchronous-extended 50
cycle, default 43
cycle, synchronous-extended 48

D

data and address parity enable field 119
data bus signals 13
data bus signals (0 - 31) 13
data bus steering 57
data bus steering control 54
data parity 120
data parity enable field 106
data parity enable signal 15
data parity exception handling 126
data parity read operations 120
data parity signals (0 - 3) 15
data parity write operations 122
data port 3
data replication, data bus 55
data size 16 return signal 16
data size 32 return signal 16
data steering 53
data transfer cycle 39
default cycle 43
default master 27
definition, channel 7
description, micro channel 1
device I/O address assignment field 106
direct memory access (DMA) procedure 83
DMA controller 3
DMA cycle termination 86
DMA slave 4
DMA use of -BURST 87
DPAREN 15
DPAR0 - DPAR3 15
DS 16 RTN 16
DS 32 RTN 16

D0 - D31 13

E

end of transfer 28, 86
EOT 28
error condition 118
exception condition reporting 118
exception conditions 118
exception handling 138
exception reporting 106, 131
exception reporting using external interrupt 138
exception, channel time-out 34
exiting inactive state 35
extensions, architecture 26

F

fairness enable field 108
fairness feature 34
first cycle after grant 37
formula, maximum statistical current 152
formula, total channel current 153

H

handling, exception condition 118

I

inactive state 34
inactive state, exiting 35
interrupt 98
interrupt signals 18
IRQ 14-15 18
IRQ 3-7 18
IRQ 9-12 18
I/O address decode 113
I/O slave 4

L

load current 152
local arbiters 28, 30

M

MADE 24 10
masters 2
master, controlling 27
master, default 27
matched memory enable field 107
matched-memory connector 148
matched-memory signal 23
maximum statistical current formula 152
memory address enable 24 signal 10
memory slave 4
memory/-input output signal 11
micro channel description 1
MSDR 16
multiple card adapters 102
multiple data transfers 28
multiplexed streaming data request signal 16
M/-IO 11

O

option select data fields 108
optional configuration fields 108
OSC 19
oscillator signal 19
ownership, channel 27

P

parity errors 119
POS I/O address selection 112
POS I/O address space 103
POS registers 103
power on reset 141

power specifications 152
PREEMPT 8, 33
preempt signal 8
preempted burst transfer 33
preemption 33
primary cards 102

R

REFRESH 19
refresh signal 19
regulation, voltage 153
reporting procedure, exception condition 118
required configuration fields 103
reserved signals 143
reset 141

S

SBHE 16
SD STROBE 16
SDR(0,1) 16
secondary cards 102
select feedback exception 140
selected feedback return exception enable field 107
selected feedback return signal 11
setup cycle timing 114
SFDBKRTN 11
signal descriptions 7
signal groups 20
signals
  -ADL 9
  -APAREN 9
  -BE0 - -BE3 13
  -BURST 8
  -CD DS 16 14
  -CD DS 32 14
  -CD SETUP 19
  -CD SFDBK 10
  -CHCK 19
  -CMD 14
  -DPAREN 15
signals *(continued)*
  -DS 16 RTN 16
  -DS 32 RTN 16
  -MSDR 16
  -PREEMPT 8
  -REFRESH 19
  -SD STROBE 16
  -SDR(0,1) 16
  -SFDBKRTN 11
  -S0, -S1 17
  -TC 18
  address bus 9
  APAR0 - APAR3 10
  ARB0 - ARB3 7
  ARB/-GNT 8
  AUDIO 19
  AUDIO GND 19
  auxiliary video 23
  CD CHRDY 14
  CHRDYRTN 14
  CHRESET 19
  data bus 13
  DPAR0 – DPAR3 15
  D0 - D31 13
  IRQ 14-15 18
  IRQ 3-7 18
  IRQ 9-12 18
  MADE 24 10
  matched-memory 23
  M/-IO 11
  OSC 19
  TR 32 18
signals, address bus 9
signals, arbitration 7
signals, interrupt 18
signals, reserved 143
single DMA transfer 88
single transfer 85
slaves 3
specifications, power 152
status field 106
status information field 111

status signals 17
steering control, data bus 54
streaming data address management 63
streaming data cycle termination 74
streaming data enable field 108
streaming data request signals 16
streaming data request 1 signal 17
streaming data strobe signal 16
streaming data timing specifications 77
streaming data transfer 58
streaming transfer, address boundary alignment 64
synchronous -CHCK 132
synchronous-extended cycle 48
system byte high enable signal 16
refid.-SBHE 16
system configuration 101
system configuration fields 103
system configuration procedure 112
system configuration timing 114
system master 3
S0, S1 17

T

TC 18
terminal count signal 18
termination, DMA cycle 86
termination, streaming data 74
timing diagrams, system configuration 114
timing, first cycle after grant 37
total channel current formula 153
TR 32 18
TR 32 signal 56
translate 32 signal 18
translator, address bus 55

V

voltage regulation 153

Numerics

16-bit base architecture 26
16-bit connectors 142
32-bit base architecture 25
32-bit connectors 142

Programmable Option Select

Description 1
Card Selected Feedback 3
System Board Setup 3
Adapter Setup 4
  Adapter Identification 6
  Required Fields 6
  Adapter Selection for Setup 7
Adapter POS Implementation 7
POS Implementation Procedure 9
System Configuration Utilities 10
  Automatic Configuration Utility 11
  Change Configuration Utility 12
  View Configuration Utility 13
  Backup and Restore Configuration Utilities 13
  Copy an Option Diskette Utility 13
Adapter Description Files 14
  Format 14
  Syntax 14
  Example 20

Index 23

Figures

1. POS I/O Address Space 3
2. POS I/O Address Decode 4
3. Channel-Check Active Indicator 5
4. Typical Adapter Implementation of POS 8
5. Subaddressing POS Extension Example 9
6. Syntax Symbol Key 14
7. Adapter Description File Syntax 15

Description

Programmable Option Select (POS) eliminates the need for switches from the system board and adapters by replacing their function with programmable registers.

The System Configuration utilities (described on page 10) automatically create configuration data for the system board and each adapter. This is achieved by reading a unique adapter ID number from each adapter, matching it with an adapter description file (ADF), and configuring the system accordingly. The resulting data and adapter ID numbers are stored in battery-backed CMOS RAM.

This data permits the power-on self-test (POST) to automatically configure the system whenever the system is powered on. The POST first verifies that the configuration has not changed by reading the adapter ID numbers and comparing them with the values stored in the battery-backed CMOS. If the configuration has changed, it is necessary to rerun the System Configuration utilities.

The adapters and the system board setup functions all share I/O addresses hex 0100 through 0107.

Warning:

- IBM recommends that programmable options be set only through the System Configuration utilities. Directly setting the POS registers or CMOS RAM POS parameters can result in multiple assignment of the same system resource, improper operation of the feature, loss of data, or possible damage to the hardware.
- Application programs should avoid using the adapter identification (ID) whenever possible. Software compatibility problems with systems and options may result.
- If an adapter and the system board are in setup mode at the same time, bus contention will occur, no useful programming can take place, and damage to the hardware can occur.
- After setup operations are complete, the Adapter Enable/Setup register (hex 0096) should be set to hex 00, and the System Board Enable/Setup register (hex 0094) should be set to hex FF.
- The channel reset bit (bit 7) in the Adapter Enable/Setup register must be 0 to program the adapters.

Setup functions respond to I/O addresses hex 0100 through 0107 only when their unique setup signal is active.

The system board does not support 16-bit I/O operations to 8-bit POS registers. Using 16-bit I/O instructions on 8-bit POS registers will cause erroneous data to be written to or read from the registers. Only 8-bit transfers are supported for setup operations.

The following figure shows the organization of the I/O address space used by the POS.

Address (Hex)	Function
0100	POS Register 0 - Adapter Identification Byte (Low byte)
0101	POS Register 1 - Adapter Identification Byte (High byte)
0102	POS Register 2 - Option Select Data Byte 1 Bit 0 is designated as Card Enable.
0103	POS Register 3 - Option Select Data Byte 2
0104	POS Register 4 - Option Select Data Byte 3
0105	POS Register 5 - Option Select Data Byte 4 Bit 7 is designated as channel check active. Bit 6 is designated as channel check status available.
0106	POS Register 6 - Subaddress Extension (Low byte)
0107	POS Register 7 - Subaddress Extension (High byte)

Figure 1. POS I/O Address Space

Bits 6 and 7 of address hex 0105 and bit 0 of address hex 0102 are fixed. All other fields within the address range of hex 0102 and 0105 are free form.

Card Selected Feedback

When the adapter is addressed, it responds by setting the Card Selected Feedback signal (-CD SFDBK) to 0. -CD SFDBK is derived by the adapter from the address decode, and driven by a totem pole driver. -CD SFDBK is latched by the system board and made available on subsequent cycles. -CD SFDBK may be used by automatic configuration or diagnostics to verify operation of an adapter at a given set of addresses. -CD SFDBK enables diagnostics to verify the operation of the adapter.

System Board Setup

The integrated I/O functions of the system board use POS information during the setup procedure. The bit assignments and functions may vary from system to system (refer to the system board setup information in the system-specific technical reference for the system you are dealing with).

Adapter Setup

The '-card setup' signal (-CD SETUP *(n)*) is unique for each channel connector. When -CD SETUP *(n)* is active, adapters recognize setup read and write operations. The adapter decodes -CD SETUP and all three low-order address bits (A0 through A2) to determine the POS register to be read from or written to. -CD SETUP is enabled by an I/O operation on the address range 0100 through 0107. The figures in this section show the complete address.

The setup routine (Automatic Configuration) obtains adapter information from ADFs and uses I/O addresses hex 0100 through 0107 to address the POS bytes of the adapter. The following figure shows the organization of the address space used by POS during adapter setup operations.

Address (Hex)	-CD SETUP	Address Bit A2	A1	A0	Function
0100 (POS Register 0)	0	0	0	0	Adapter Identification Byte (Least-significant byte)
0101 (POS Register 1)	0	0	0	1	Adapter Identification Byte (Most-significant byte)
0102 (POS Register 2)	0	0	1	0	Option Select Data (Byte 1)*
0103 (POS Register 3)	0	0	1	1	Option Select Data (Byte 2)
0104 (POS Register 4)	0	1	0	0	Option Select Data (Byte 3)
0105 (POS Register 5)	0	1	0	1	Option Select Data (Byte 4)*
0106 (POS Register 6)	0	1	1	0	Subaddress Extension (Least-significant byte)
0107 (POS Register 7)	0	1	1	1	Subaddress Extension (Most-significant byte)

* These bytes contain one or more bits with specific assignments.

Figure 2. POS I/O Address Decode

Bytes hex 0100 and 0101 are 8-bit read-only. Bytes hex 0102 through 0107 are 8-bit read-only and read-write.

All bits in bytes hex 0102 through 0105 are free-form, adapter-dependent, and implemented except for the following:

- **Hex 0102, Bit 0:** Card Enable (CDEN): When this bit is set to 0, the adapter is disabled, responding only to setup read and write operations, and 'channel reset'. It does not respond to I/O or

memory read or write operations, nor does it make any interrupt requests. When this bit is set to 1, the adapter is fully enabled.

- **Hex 0105, Bit 7:** Channel-Check Active Indicator (-CHCK): System memory and I/O functions that report a channel-check must set a channel-check active indicator to identify the source of the error. This indicator is bit 7 of address hex 0105 of each adapter POS address space. This bit can be interrogated by the nonmaskable interrupt (NMI) handler responding to 'channel check' for each adapter position until all reporting adapters have been identified. The following figure shows a typical implementation of the channel-check active indicator.

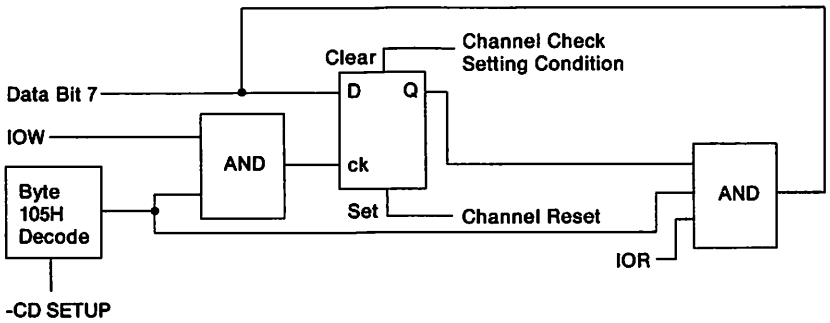


Figure 3. Channel-Check Active Indicator

The indicator is set to 0 on a channel-check condition or when bit 7 of POS Register 5 is set to 0. The indicator is set to 1 on a channel reset, or when bit 7 of hex 0105 is 1. This bit may be reset by any action occurring during the channel-check service routine. If the channel-check active indicator is used by an attachment, hex 0105 bit 6 must be used to indicate whether additional status is available through bytes hex 0106 and 0107.

- **Hex 0105, Bit 6:** Channel-Check Status Indicator: When set to 0, this bit indicates channel-check exception status is available from POS Registers 6 and 7. When set to 1, this bit indicates no status is available. Registers 6 and 7 may be the status, a pointer to status, or a command port to present the address elsewhere (for example, in a subaddress area).

  Bit 6 is required by all devices supporting the channel-check active indicator (bit 7). If a device does not use the channel-check active indicator, bit 6 may be defined to contain

device-unique information. If a device uses the channel-check active indicator, but does not report status, bit 6 must be set to 1.

Adapter Identification

Each adapter has a unique 2-byte adapter ID. This enables diagnostic programs, configuration utilities, and POST routines to initialize the adapter when the system is powered on or reset.

To minimize the need for hardware, only bits driven to 0 require drivers. Pull-up resistors on the system board provide a 1 for each remaining bit. See "Micro Channel Adapter Design" for more information.

Required Fields

Several fields are not assigned specific bit locations within the *free form* POS bytes. However, the following are required if the adapter supports the function:

- **Fairness Enable Bit:** All bursting devices using the arbitration mechanism must support the *fairness* feature through a programmable fairness-enable bit. The default state of this bit is 1, requiring all devices to honor the fairness feature. When fairness-enable is set to 0, the fairness feature is disabled.
- **Arbitration Level Field:** All devices (bursting and nonbursting) using the arbitration mechanism must support a programmable arbitration level through a 4-bit allocation. This field allows incorrectly prioritized devices to be reassigned by diagnostic or system programs to reduce impacts on performance. Only one device may be assigned to each arbitration level.
- **Device ROM Segment Address Field:** All I/O devices containing memory-mapped I/O ROM must support a programmable Device ROM Segment Address field. This field can be up to 4 bits and provides the ROM of a device a starting address at any one of sixteen 8KB segments.
- **I/O Device Address Field:** All I/O devices that can simultaneously reside in a system with a device of the same type must support programmable I/O device addresses. This field eliminates addressing conflicts.

Adapter Selection for Setup

Each channel position has a unique 'setup' line (-CD SETUP) associated with it. See the system-specific technical references for more information.

Adapter POS Implementation

The following figure shows how an adapter typically implements POS. All designs must latch the least-significant bit of the device-dependent option-select byte. Bit 7 of POS Register 5 is set to 1 unless -CHCK is active from the adapter. The remaining bits can be implemented as required.

Note: Any adapter that POS does not completely initialize should implement a second enable, which is activated by adapter ROM routines or loadable software. The card-disable function (POS Register 2, bit 0) must override a second enable.

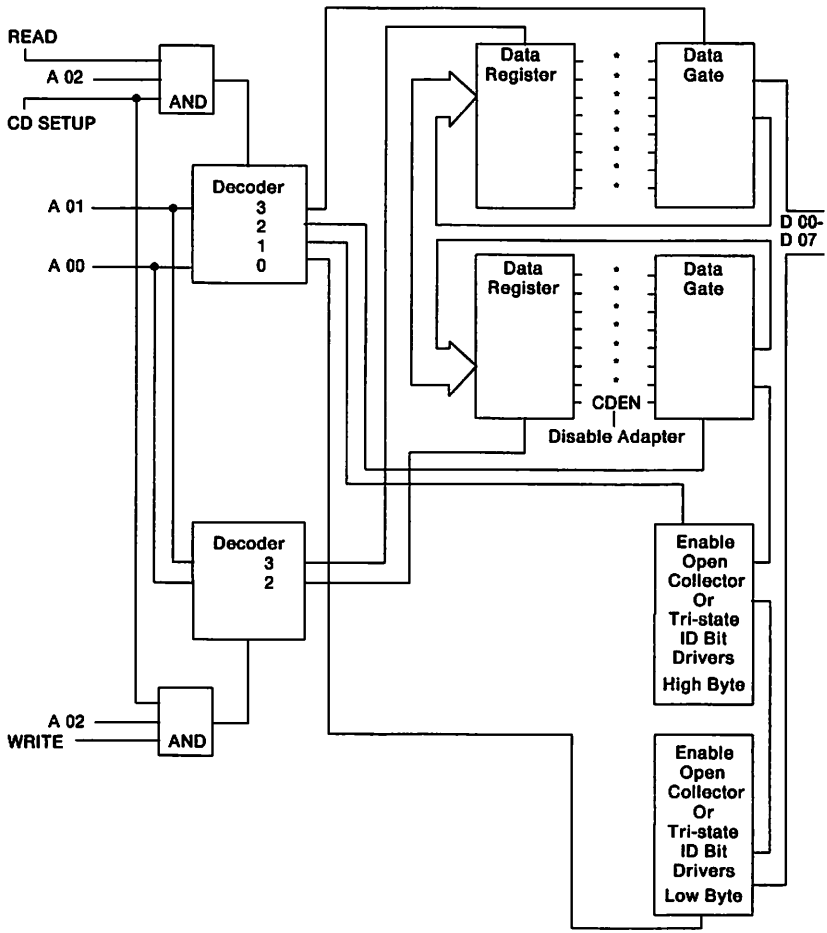


Figure 4. Typical Adapter Implementation of POS

The POS subaddressing extension allows the subaddressing of a block of initial program load (IPL) or setup information. Subaddressing bits (SAD00 through SAD15) are used to address RAM, a register stack, or other devices.

The following figure shows the subaddressing extension for memory. The counter registers increment after each least-significant byte of option-select information is written.

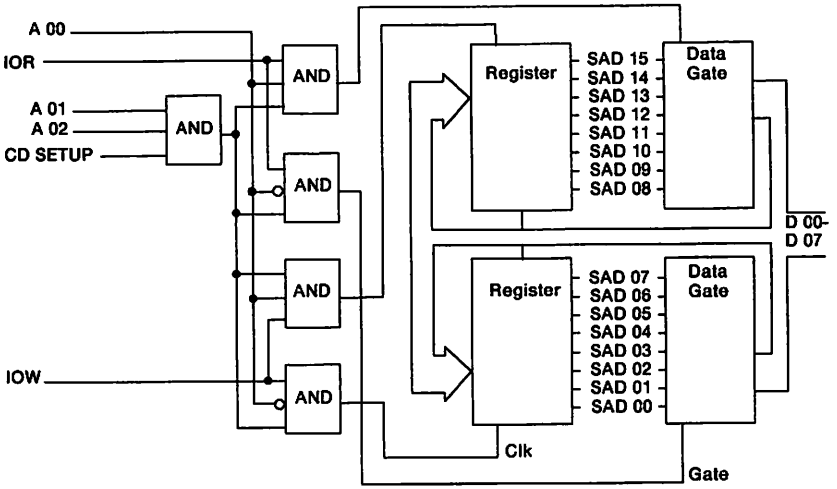


Figure 5. Subaddressing POS Extension Example

POS Implementation Procedure

Although the design of POS circuitry is the designer's choice, the following is an example of a typical POS implementation.

1. Disable interrupts.
2. Select the adapter for subsequent setup cycles. See the system-specific technical references for more information.
3. Read the adapter ID by an I/O read at hex 0100 and hex 0101.
4. Disable the adapter and place it in setup by performing an I/O write to hex 0102 with bit 0 off.
5. Write POS data to hex 0103, 0104, and 0105 in any order.
6. With bit 0 set to 1, write POS data to hex 0102.
7. Deselect the adapter. See the system-specific technical references for more information.
8. Enable interrupts.

The system microprocessor can communicate with the adapter, provided the adapter is enabled (bit 0 at hex 0102 set to 1). After the adapter has been set up, a subsequent I/O write does not affect these latches or permit the ID circuitry of the adapter to operate, unless the adapter is returned to setup.

System Configuration Utilities

Each system has a Reference Diskette containing the System Configuration utilities. These utilities identify the installed hardware and interpret the system resources (I/O ports, memory, interrupt levels, and arbitration levels) for each device. The System Configuration utilities are contained within the Set Configuration program.

The Reference Diskette enables the user to configure the system in one of two ways:

- Running the Automatic Configuration utility after a configuration error is displayed
- Selecting Set Configuration from the Main Menu.

The Set Configuration program uses information contained in adapter description files (ADFs) to track and allocate system resources. Each ADF describes the resources that can be allocated to a specific adapter and the POS setting used to indicate those resource assignments. When more than one device is configured to the same resource and that resource cannot be shared, only one of the conflicting devices is enabled.

ADF data for the system board and some adapters is contained on the Reference Diskette. Before new adapters are installed, their associated ADFs must be merged onto a backup copy of the Reference Diskette by selecting Copy an Option Diskette from the Main Menu and following the instructions on the screen.

Each adapter contains a 16-bit adapter ID and one to four 8-bit POS registers. Adapter IDs and the POS information are stored in CMOS RAM by the Set Configuration program. The CMOS RAM locations used to hold this information are not the same for all systems. The Set Configuration program determines the system type and handles differences between systems such as the CMOS RAM storage and the

number of available adapter slots. If the system is identified as having only a 64-byte CMOS RAM, adapter IDs and POS data are stored in the 64-byte CMOS RAM. Each adapter slot is allocated 2 bytes for an adapter ID and 4 bytes for POS data. If the system type is identified as having a 2KB CMOS RAM extension, adapter IDs and POS data are stored in the 2KB CMOS RAM.

Automatic Configuration Utility

Automatic Configuration can be run after a configuration error has occurred or by selecting Run Automatic Configuration from the Set Configuration menu. Each time the system is powered on, the POST compares the configuration of the system to the configuration indicated by CMOS RAM. If differences between the two are detected, an error is displayed and logged in system RAM. POST error message files contained on the Reference Diskette display text that provides further information about the POST error.

Note: The Reference Diskette must be installed when the system is powered on to receive POST error messages.

If a configuration error is caused by a battery failure or a bad cyclic-redundency check (CRC), Automatic Configuration is run immediately after the POST error is displayed. If the error is caused by a change to the configuration, the user is given a choice to either run Automatic Configuration or continue to the Main Menu. If the user continues with the Main Menu, the changed areas of the system are configured and CMOS RAM is updated when Set Configuration is selected from the Main Menu.

Depending on the source of the error, Automatic Configuration either reconfigures the entire system or configures only the areas of the system that have been changed since the last time a configuration was performed. The following POST errors cause the system to be completely reconfigured:

- 161 (battery failure)
- 162 (bad CMOS CRC).

The following POST errors cause only the areas of the system that have changed to be reconfigured:

- 162 (system configuration error not caused by a bad CMOS CRC)
- 164 (memory configuration)
- 165 (adapter configuration).

An adapter is considered previously configured when the POS data stored in CMOS RAM matches a POS setting in the appropriate ADF. If an ADF for an installed adapter cannot be found, the adapter is configured as an empty slot.

During Automatic Configuration, devices are configured to the first nonconflicting values as defined in the ADF. Adapters are configured in the order of the channel position in which they are installed. The system board is configured first, followed by each adapter slot starting with slot 1. If the interrupt level is the only resource defined for a specific adapter item, the choice of interrupt levels that are least used by other adapters are assigned.

Automatic Configuration does not backtrack to previously configured adapters to resolve resource conflicts. If conflicts can be resolved, they must be done by choosing a nonconflicting resource option through the Change Configuration utility. Any adapter having a resource conflict that cannot be resolved by the Set Configuration program is disabled; the program sets bit 0 of POS Register 2 (hex 0102) to 0 in CMOS RAM.

Change Configuration Utility

The Change Configuration utility allows the user to change the default configuration settings from those set by Automatic Configuration. This utility is used to resolve unusual conflicts or to set items for personal preference.

The user interface is through scrolling and paging screens. Changes are made by rotating field value names through a set of choices using the F5 (Previous) and F6 (Next) keys. Changes are not saved in CMOS until the F10 (Save) key is pressed. Help text is provided for each item by pressing the F1 (Help) key.

Resource conflicts are indicated by an asterisk (*) next to the conflicting items and also by the "* Conflicts" string in the upper right corner of the Change Configuration window. Conflicts with fixed resources have the asterisk (*) to the left of the slot number of the adapter. Adapters with conflicts are disabled; the program sets bit 0 of POS Register 2 (hex 0102) to 0 in CMOS RAM.

View Configuration Utility

The View Configuration utility is provided to view the configuration. This is the Change Configuration utility with the change capabilities disabled.

Backup and Restore Configuration Utilities

The Backup Configuration utility provides a means to back up the configuration data to a file on the Reference Diskette. If the battery fails or the battery is changed, the user can use the Restore utility to restore the configuration.

Note: A copy of the Reference Diskette that is not write-protected is needed for this backup and restore process.

Copy an Option Diskette Utility

The Copy an Option Diskette utility is a separate program from the Set Configuration program and is accessible through the Main Menu. This utility is used to merge the following files from an option diskette onto a backup copy of the Reference Diskette: *.adf, *.dgs, *.pep, COMMAND.COM, DIAGS.COM, CMD.COM, and SC.EXE. The files found on the option diskette are compared to the files on the backup copy of the Reference Diskette. If the file does not exist on the Reference Diskette, it is copied. If the file already exists on the Reference Diskette, the dates of the two files are compared. The file on the option diskette is copied only if it has a date later than the corresponding file on the Reference Diskette.

The option diskette must be a DOS formatted diskette.

Adapter Description Files

Adapter description files provide POS information and system resource information for Automatic Configuration. The adapter description files also provide text for System Configuration utilities, help screens, and prompts. This section provides guidelines for developing the adapter description files.

Format

- File names: @CARDID.adf (high byte of the adapter ID first).
- Type of file: ASCII text.
- Not case sensitive: Key words can be lowercase, uppercase, or mixed. The case is preserved within the user interface text strings.
- Blanks, tabs, new lines: These are considered as white space and ignored, except when in text strings for the Change Configuration user interface.
- Comments: Lines beginning with semicolons are comments and are ignored.

Syntax

The following figure shows the meaning of special symbols used in the adapter description file syntax.

Symbol	Meaning
{ }	an optional item
{ }*	0,1,2,... items allowed
{ }+	1,2,3,... items allowed
x | y	either x or y allowed
x{n}	n x's required
[0-9]+	one or more decimal digits

Figure 6. Syntax Symbol Key

```
adf_file => card_id card_name nbytes {fixed_resources} {named_item}*
```

This defines the contents of an ADF. The following definitions describe each portion of an ADF in detail.

```
card_id => AdapterId number
```

Each ADF must contain a card_id. The character string 'AdapterId' is a keyword and must be present in the ADF. The Configuration program looks for this ID, which must match the ID used in the filename.

Example: `AdapterId 0DEFFh`

```
card_name => AdapterName string
```

Each ADF must contain a card_name. The character string 'AdapterName' is a keyword and must be present in the ADF. The string following the 'AdapterName' keyword is displayed as the adapter name in the Change Configuration and View Configuration screens. The length of the AdapterName string is limited to (74 - (length of 'SlotX - ')) characters. (US English length = 66 characters)

Example: `AdapterName "IBM Multi-Protocol Communications Adapter"`

```
nbytes => NumBytes number
```

Each ADF must contain an nbytes. The character string 'NumBytes' is a keyword and must be present in the ADF. This is used to define the number of POS bytes used by the adapter. The number of POS bytes used should include all bytes from POS [0] to the last POS byte used. (If POS [3] is the only POS byte defined, NumBytes should be set to 4.)

Example: `NumBytes 4`

```
fixed_resources => FixedResources pos_setting resource_setting
```

A fixed_resources is not a requirement for ADFs. It is used to define resources required by an adapter and corresponding POS data. The character string 'FixedResources' is a keyword and must be present in the ADF only if the adapter needs to define resources that it requires.

Example: `FixedResources  POS[1]=XXXX01XXb int 3`

Figure 7 (Part 1 of 5). Adapter Description File Syntax

```
named_item => NamedItem prompt {named_choice}+  help
```

A named_item is not a requirement for ADFs. The named_item is used to define a field providing a choice of one or more resources. Each choice sets specified POS bits to a unique setting used to identify resources assigned to the adapter. The character string 'NamedItem' is a keyword and must be present in the ADF only if the adapter can be configured to use different resources. The adapter determines the resources it is configured to by how the POS bytes are set. When a 'NamedItem' is defined in an ADF it must be accompanied by a prompt (defined later), at least one named_choice (defined later), and help (defined later). Example:

```
NamedItem
Prompt "Communications Port"
choice "SDLC_1"   pos[0]=XXX1000Xb  io 0380h-038ch  int 3 4
choice "SDLC_2"   pos[0]=XXX1001Xb  io 03a0h-03ach  int 3 4
choice "BISYNC_1" pos[0]=XXX1100Xb  io 0380h-0389h  int 3 4
choice "BISYNC_2" pos[0]=XXX1101Xb  io 03a0h-03a9h  int 3 4
choice "SERIAL_1" pos[0]=XXX0000Xb  io 03f8h-03ffh  int 4
choice "SERIAL_2" pos[0]=XXX0001Xb  io 02f8h-02ffh  int 3
choice "SERIAL_3" pos[0]=XXX0010Xb  io 3220h-3227h  int 3
choice "SERIAL_4" pos[0]=XXX0011Xb  io 3228h-322fh  int 3
choice "SERIAL_5" pos[0]=XXX0100Xb  io 4220h-4227h  int 3
choice "SERIAL_6" pos[0]=XXX0101Xb  io 4228h-422fh  int 3
choice "SERIAL_7" pos[0]=XXX0110Xb  io 5220h-5227h  int 3
choice "SERIAL_8" pos[0]=XXX0111Xb  io 5228h-522fh  int 3
Help
"This port can be assigned as a:  primary (SDLC1) or
secondary (SDLC2) sdlc, primary (BISYNC1) or secondary
(BISYNC2) bisync, or as a serial port (Serial 1 through Serial
8).  Use the F5=Previous and the F6=Next keys to change this
assignment in the 'Change configuration' window.  Conflicting
assignments are marked with an asterisk and must be changed
to use the adapter."
```

```
prompt => Prompt string
```

The prompt is required when a NamedItem is defined. The prompt is used to define a title for a NamedItem field. The character string 'Prompt' is a keyword and must be present in the ADF when there is a NamedItem present. The string following the 'Prompt' keyword appears after the adapter name in the Change Configuration and View Configuration screens. Following the prompt string is a field that can be toggled in the Change Configuration screen if two or more named_items are defined. The length of the prompt string cannot exceed 38 characters. Example: (See the example for named_item).

Figure 7 (Part 2 of 5). Adapter Description File Syntax

```
named_choice => Choice choice_name pos_setting resource_setting
```

At least one named_choice is required when a NamedItem is defined. The character string 'Choice' is a keyword and must be present in the ADF when there is a NamedItem present. One or more named_choices must follow a prompt. Each named_choice must contain a string describing the current choice in the prompt field. Each named_choice must define a pos_setting, for at least one POS byte, which will uniquely identify the resource_setting defined in the named_choice. The length of the named_choice string cannot exceed 28 characters.
Example: (See the example for named_item).

```
help => Help string
```

The help is a string of text used to give the user assistance at a prompt. This text is displayed in the Change Configuration and View Configuration screen when the cursor is at the associated prompt and the F1 key is pressed. The character string 'Help' is a keyword and must be present in the ADF when there is a NamedItem present. The string following the keyword 'Help' is the text describing the prompt defined in the same NamedItem as the help. The length of the help string is limited to 1000 characters.
Example: (See the example for named_item).

```
pos_setting => {pos_byte_setting}+
```

The pos_setting must contain at least one pos_byte_setting. See the definition of pos_byte_setting for more information.

```
pos_byte_setting => pos[number]=pos_bit{8}b
```

This is the definition of the pos_byte_setting in the ADF.
The character string 'pos' is a keyword and must be present in a pos_byte_setting, followed by a number in brackets. The number in brackets refers to the following POS bytes:

- number = 0, POS byte at port 102h
- number = 1, POS byte at port 103h
- number = 2, POS byte at port 104h
- number = 3, POS byte at port 105h

The end bracket must be followed by an equal sign and then a bit definition of the POS byte (See pos_bit for information on the bit definition). The bit definition must define all 8 bits of the byte.
Bit 0 of pos[0] should always be defined as X in the ADF.
Example: `pos[0]=XXX1001Xb`

Figure 7 (Part 3 of 5). Adapter Description File Syntax

```
pos_bit => x | X | 0 | 1
```

A pos_bit can be defined as a mask bit (x or X), a clear bit (0), or a set bit (1).

Example: `pos[0]=XXX1001Xb`

```
resource_setting =>
    {ioblock_list} {interrupt_list} {arb_list} {memaddr_list}
```

The resource_setting defines a list of system resources. They may be fixed resources required by the adapter or they may be resources the adapter uses when configured to a specific choice in a named_item. The resources can consist of the following:

- Range of I/O addresses (limited to 16).
- List of interrupt levels (limited to 16).
- List of arbitration levels (limited to 16).
- Range of memory addresses (limited to 2).

Example: (See the following resource definitions).

```
ioblock_list => IO {range}+
```

The ioblock_list must be a list of one or more ranges of I/O addresses. The character string 'IO' is a keyword and must be present in the ioblock_list.

Example: `io 4220h-4227h`

```
Interrupt_list => INT {number}+
```

The interrupt_list must be a list of one or more interrupt levels. The character string 'INT' is a keyword and must be present in the interrupt_list.

Example: `INT 3 4`

```
arb_list => ARB {number}+
```

The arb_list must be a list of one or more arbitration levels. The character string 'ARB' is a keyword and must be present in the arb_list.

Example: `ARB 1`

```
memaddr_list => MEM {range}+
```

The memaddr_list must be a list of one or more ranges of RAM or ROM addresses. The character string 'MEM' is a keyword and must be present in the memaddr_list. This keyword is used to allocate memory address space in the hex 000C0000 through hex 000DFFFF range.

Example: `MEM 0CC000h - 0CDFFFh`

```
range => number - number
```

Figure 7 (Part 4 of 5). Adapter Description File Syntax

```
number => [0-9]+ {d} | [0-9a-f]+ h | [0-9A-F]+ H

string => " [ascii except for "]+ "
```

A string is a set of ASCII characters beginning with a double quote (") and ending with a double quote.
Example:

```
"This port can be assigned as a: primary (SDLC1) or
secondary (SDLC2) sdlc port, primary (BISYNC1) or secondary
(BISYNC2) bisync port, or as a serial port (Serial 1 through
Serial 8). Use the F5=Previous and the F6=Next keys to change
this assignment in the 'Change configuration' window.
Conflicting assignments are marked with an asterisk and must
be changed to use the adapter."
```

Figure 7 (Part 5 of 5). Adapter Description File Syntax

Example

The following is an example of an adapter description file for the IBM Personal System/2 Multiprotocol Communications Adapter/A. The name of the file for this adapter is @DEFF.adf. An explanation of each numbered item begins on page 21.

```
AdapterId 0DEFFh 1

AdapterName "IBM Multiprotocol Communications Adapter" 2

NumBytes 2 3

NamedItem 4

Prompt "Communications Port"

choice "SDLC_1"    pos[0]=XXX1000Xb  io 0380h-038ch  int 3 4
choice "SDLC_2"    pos[0]=XXX1001Xb  io 03a0h-03ach  int 3 4
choice "BISYNC_1" pos[0]=XXX1100Xb  io 0380h-0389h  int 3 4
choice "BISYNC_2" pos[0]=XXX1101Xb  io 03a0h-03a9h  int 3 4
choice "SERIAL_1" pos[0]=XXX0000Xb  io 03f8h-03ffh  int 4
choice "SERIAL_2" pos[0]=XXX0001Xb  io 02f8h-02ffh  int 3
choice "SERIAL_3" pos[0]=XXX0010Xb  io 3220h-3227h  int 3
choice "SERIAL_4" pos[0]=XXX0011Xb  io 3228h-322fh  int 3
choice "SERIAL_5" pos[0]=XXX0100Xb  io 4220h-4227h  int 3
choice "SERIAL_6" pos[0]=XXX0101Xb  io 4228h-422fh  int 3
choice "SERIAL_7" pos[0]=XXX0110Xb  io 5220h-5227h  int 3
choice "SERIAL_8" pos[0]=XXX0111Xb  io 5228h-522fh  int 3

Help

"This port can be assigned as a: primary (SDLC1) or
secondary (SDLC2) sdlc port, primary (BISYNC1) or secondary
(BISYNC2) bisync port, or as a serial port (Serial 1 through
Serial 8).  Use the F5=Previous and the F6=Next keys to
change this assignment.  Conflicting assignments are marked
with an asterisk and must be changed."
```

```
NamedItem 5

Prompt "Arbitration Level for SDLC"

choice "Level_1"   pos[1]=XXXX0001b  arb  1
choice "Level_0"   pos[1]=XXXX0000b  arb  0
choice "Level_2"   pos[1]=XXXX0010b  arb  2
choice "Level_3"   pos[1]=XXXX0011b  arb  3
choice "Level_4"   pos[1]=XXXX0100b  arb  4
choice "Level_5"   pos[1]=XXXX0101b  arb  5
choice "Level_6"   pos[1]=XXXX0110b  arb  6
choice "Level_7"   pos[1]=XXXX0111b  arb  7
choice "Level_8"   pos[1]=XXXX1000b  arb  8
choice "Level_9"   pos[1]=XXXX1001b  arb  9
choice "Level_10"  pos[1]=XXXX1010b  arb 10
choice "Level_11"  pos[1]=XXXX1011b  arb 11
choice "Level_12"  pos[1]=XXXX1100b  arb 12
choice "Level_13"  pos[1]=XXXX1101b  arb 13
choice "Level_14"  pos[1]=XXXX1110b  arb 14

Help

"This assignment need only be changed if it is in conflict
with another assignment.  Conflicting assignments are marked
with an asterisk.  Use the F5=Previous and the F6=Next keys
to change arbitration level assignments.  Using arbitration
levels, this adapter accesses memory directly without
burdening the computer's main microprocessor.  An
arbitration level of 0 has the highest priority, and
increasing levels have corresponding decreased priority"
```

1 The card_id for this adapter is hex 0DEFF. This is an ASCII representation of the ID generated by the adapter. The high byte is followed by the low byte. The card_id is required for all ADFs.

2 The card_name is "IBM Multiprotocol Communications Adapter." The card_name is required for all ADFs.

3 The nbytes (NumBytes 2) in this file indicates the adapter uses two POS bytes located at hex 0102 and 0103.

4 This is the first named_item for the adapter. The title of the field is "Communications Port." The user can toggle between the 12 named_choices. Each named_choice has a unique pos_setting assigned to it in bit locations 1 through 4 of POS byte hex 0102 (pos [0]). Also shown is a resource_setting that corresponds to the pos_setting of the named_choice. The resources allocated in this named_item are I/O addresses and interrupt levels. A help string for this named_item is provided below the last named_choice.

5 This is the second named_item for the adapter. The title of the field is "Arbitration Level for SDLC." The user can toggle between the 15 named_choices. Each named_choice has a unique pos_setting assigned to it in bit locations 0 through 3 of POS byte hex 0103 (pos [1]). Also shown is a resource_setting that corresponds to the pos_setting of the named_choice. The resources allocated in this named_item are arbitration levels. A help string for this named_item is provided below the last named_choice.

Index

A

adapter configuration, order of 12
adapter description files 1
adapter ID number 1
adapter identification 6, 10
adapter POS implementation 7
adapter setup 4, 7
address decode 3
address space, POS 3
arbitration level field 6
automatic configuration 1, 4, 11

B

backup configuration utility 13
battery failure 11
bus contention 2

C

card enable 4
card selected feedback register 3
CD SETUP 4
CD SFDBK 3
change configuration utility 12
channel-check active indicator 5
channel-check status indicator 5
CMOS RAM, card ID bytes 10
configuration error 11
configuration utilities 6
copy an option diskette utility 13
CRC error 11

D

device ROM segment address field 6

F

fairness enable bit 6
fairness feature 6

H

help text 12

I

I/O address decode 4
I/O device address field 6
initial program load 8

P

POS I/O address space 3, 4
POS overview 1
POS registers 3
POST error message files 11
power-on self-test 1

R

reference diskette 10
restore configuration utility 13

S

SAD00 - SAD15 8
set configuration program 10
setup procedures 2
setup, system board 3
subaddressing bits 8
subaddressing extension 8
system board setup 3
system configuration utilities 2, 10
system resources 10

V

view configuration 13

Micro Channel Adapter Design

General Guidelines 1
Dimensions 1
Power 19
Voltage Regulation 20
General Design Considerations 20
  Safety 20
  Thermal 21
  Electromagnetic Compatibility 21
  Diagnostics 22
Design Guidelines 22

Index 25

Figures

1. Adapter Dimensions (8- or 16-Bit) 2
2. Connector Dimensions (8- or 16-Bit) 3
3. Adapter Dimensions (8- or 16-Bit with Video Extension) . . . 4
4. Connector Dimensions (8- or 16-Bit with Video Extension) . . 5
5. Adapter Dimensions (32-Bit) 6
6. Connector Dimensions (32-Bit) 7
7. Adapter Dimensions (32-Bit with Matched Memory) 8
8. Connector Dimensions (32-Bit with Matched Memory) 9
9. Connector (Common Detail) 10
10. Typical Adapter Assembly 11
11. Adapter Holder 12
12. Adapter Retainer 13
13. Adapter Bracket 15
14. Channel Load Current 19
15. Channel Voltage Regulation 20
16. Vendor ID Assignments 23

General Guidelines

This section provides some basic guidelines to design adapters for the Micro Channel architecture 16- and 32-bit products. Topics include physical specifications, power requirements and limitations, and configuration program support.

The system board provides channel connectors to support the following types of adapters. Some systems do not support all types of adapters. See the system-specific technical references for more information.

- 16-bit adapter
- 16-bit adapter with video extension
- 32-bit adapter
- 32-bit adapter with matched-memory extension.

Connector contacts are not required for signals not used by an adapter. See "Micro Channel Architecture" for more information on channel connectors and signals.

Dimensions

The following figures show the dimensions of each type of adapter and the associated mounting hardware. The tolerances shown include all individual process tolerances and are not cumulative. The maximum height for components mounted on the adapter is 15 millimeters (0.6 inch) on the component (A) side. The maximum height for pins and components on the B side of the adapter is 2 millimeters (0.078 inch). Adapters using CMOS technology should have all plated connector contacts the same length to reduce the exposure of incorrect bias to modules.

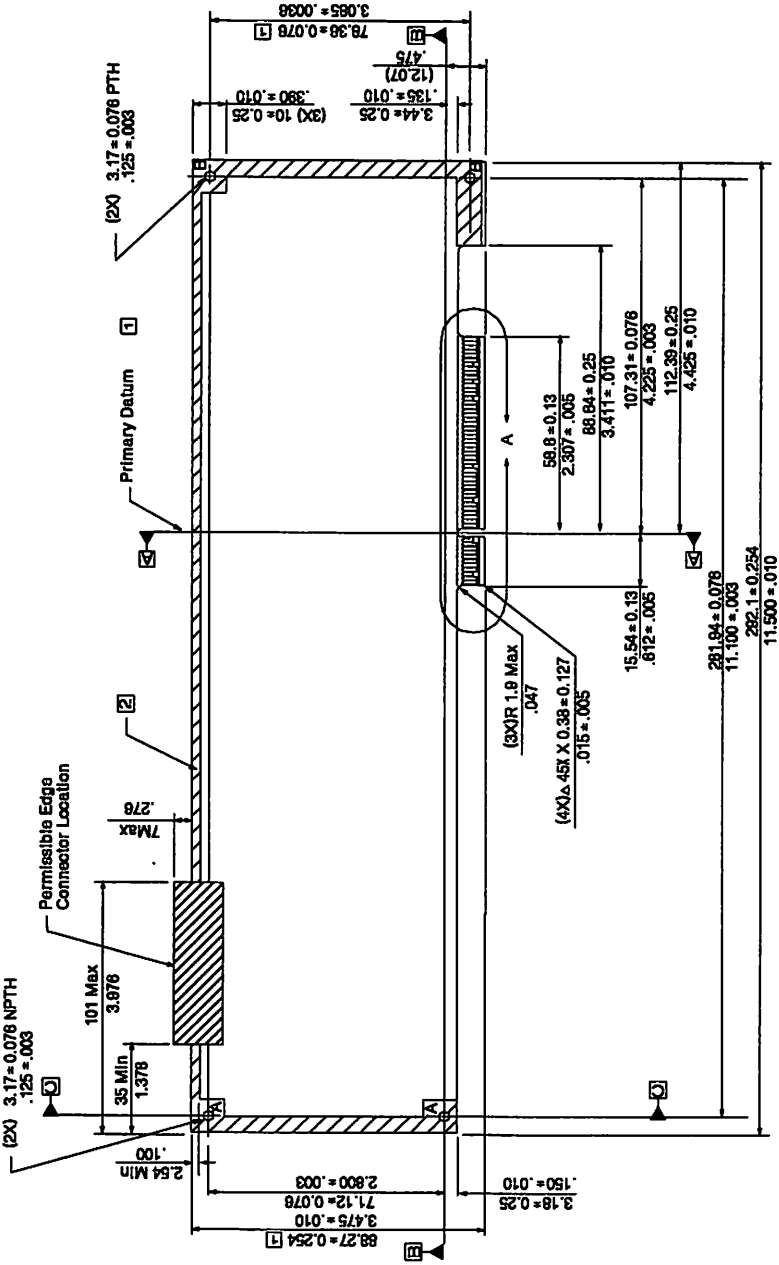


Figure 1. Adapter Dimensions (8- or 16-Bit)

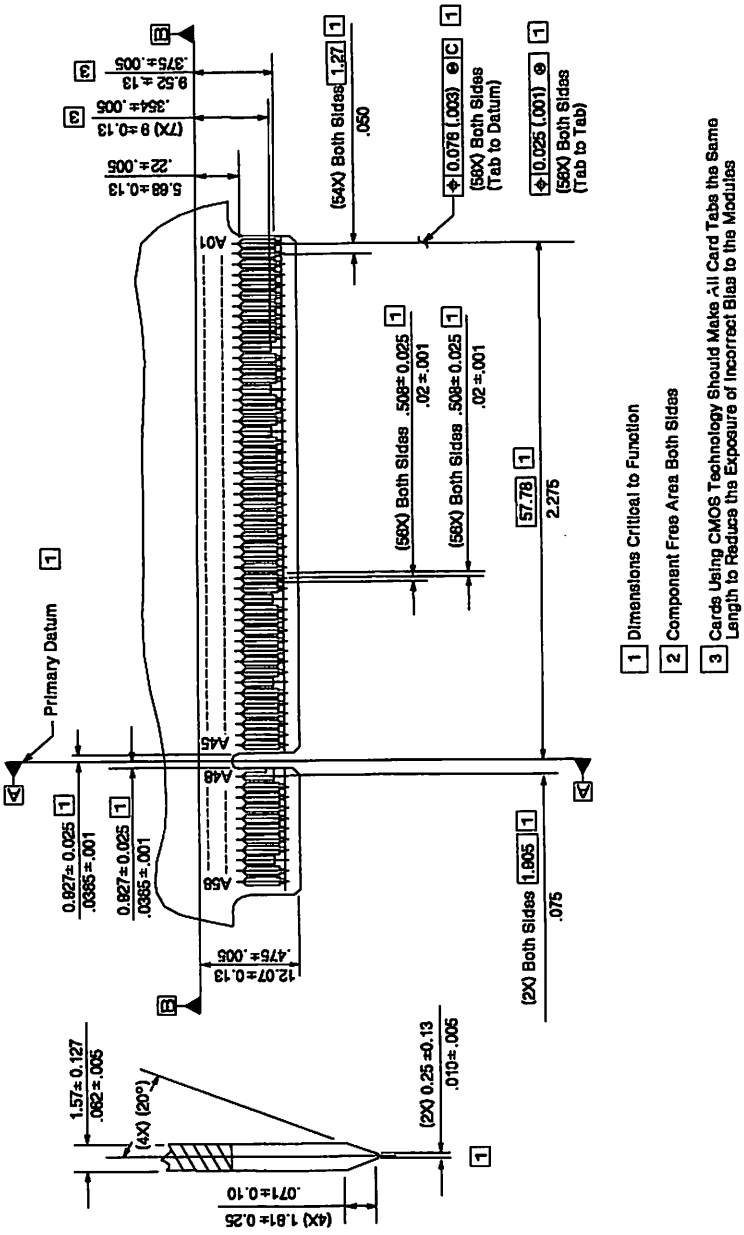


Figure 2. Connector Dimensions (8- or 16-Bit)

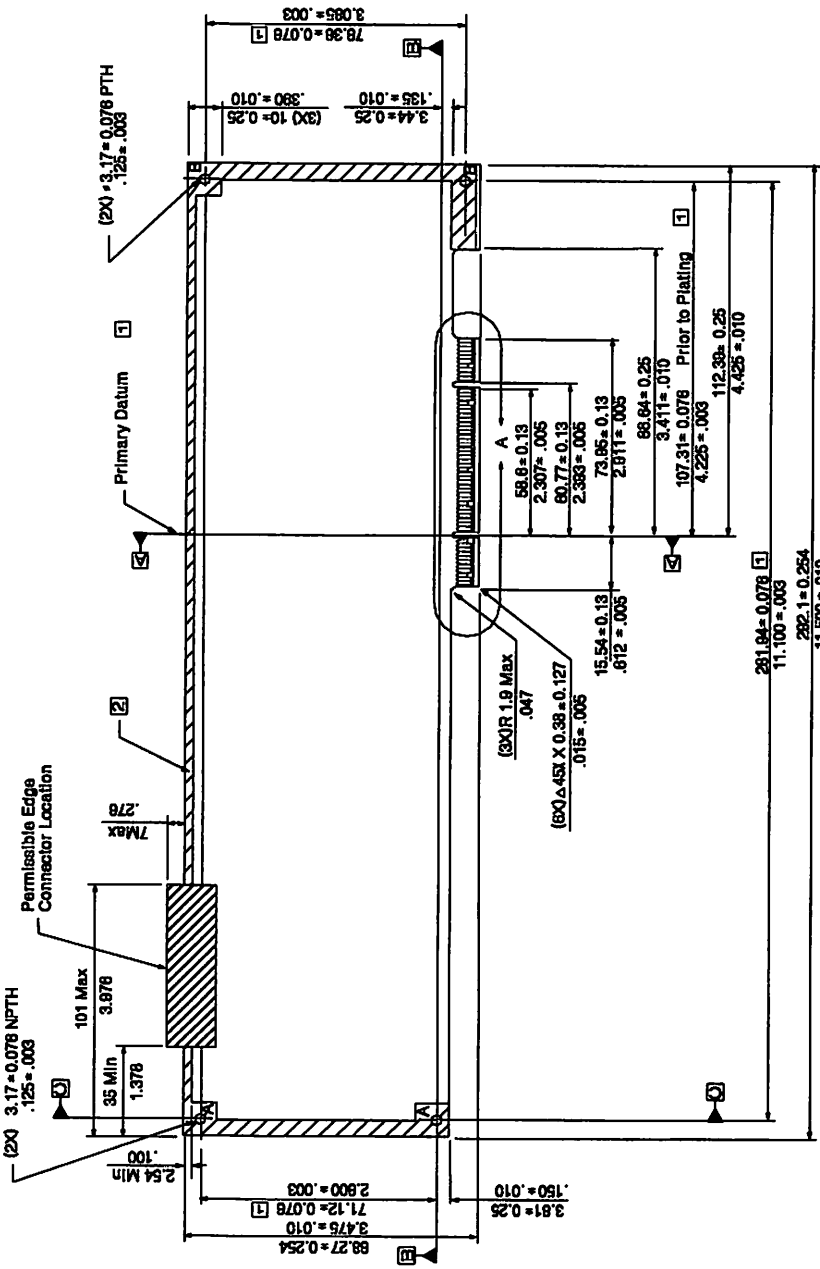


Figure 3. Adapter Dimensions (8- or 16-Bit with Video Extension)

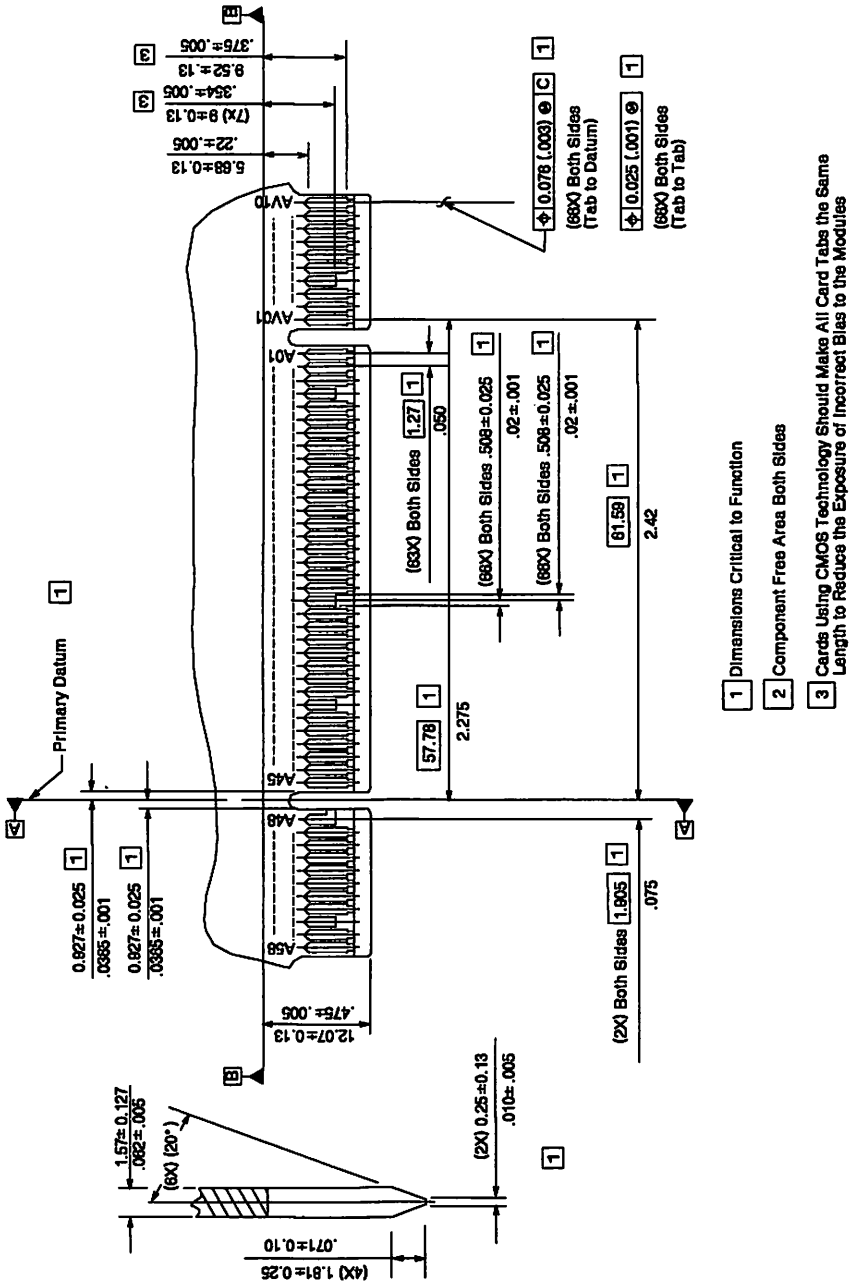


Figure 4. Connector Dimensions (8- or 16-Bit with Video Extension)

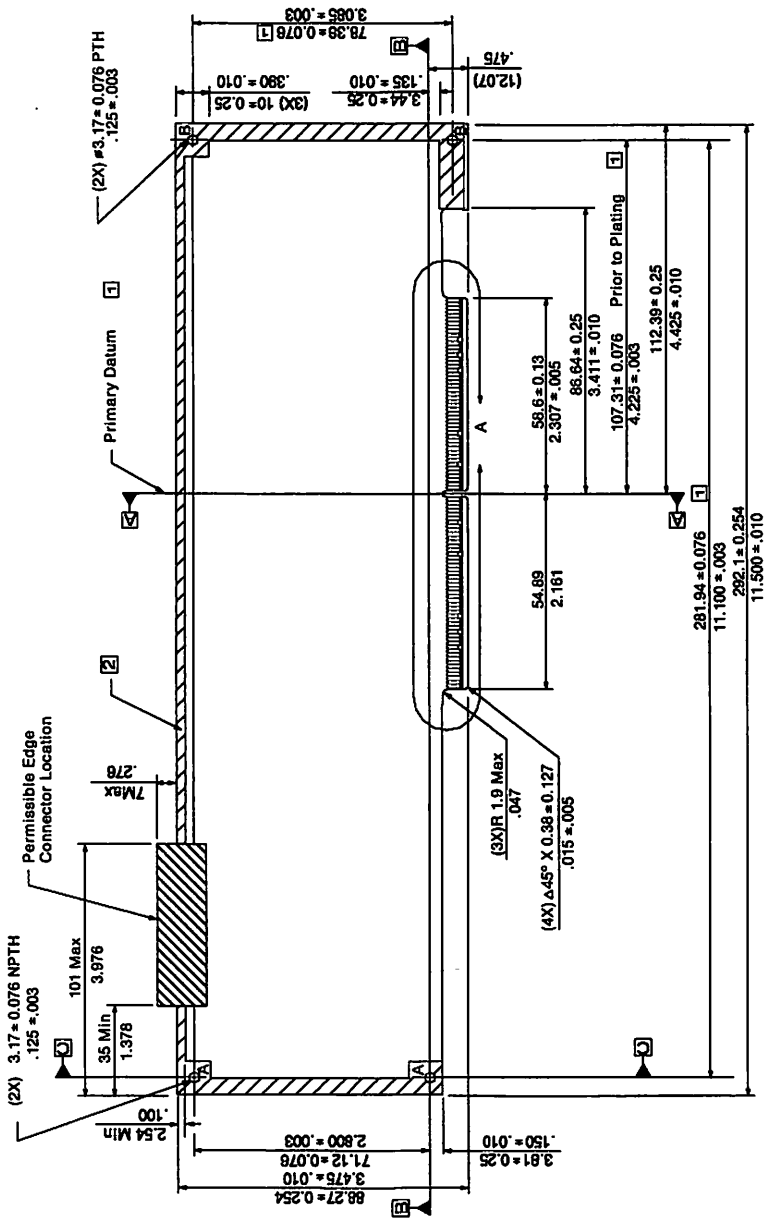


Figure 5. Adapter Dimensions (32-Bit)

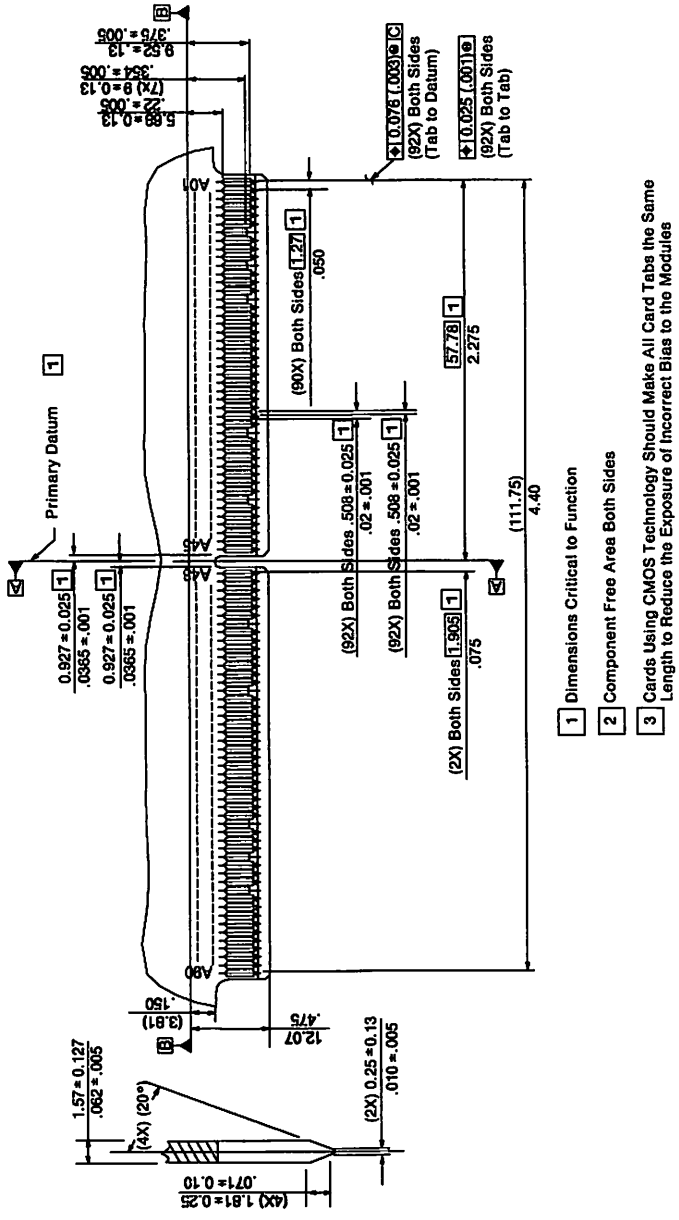


Figure 6. Connector Dimensions (32-Bit)

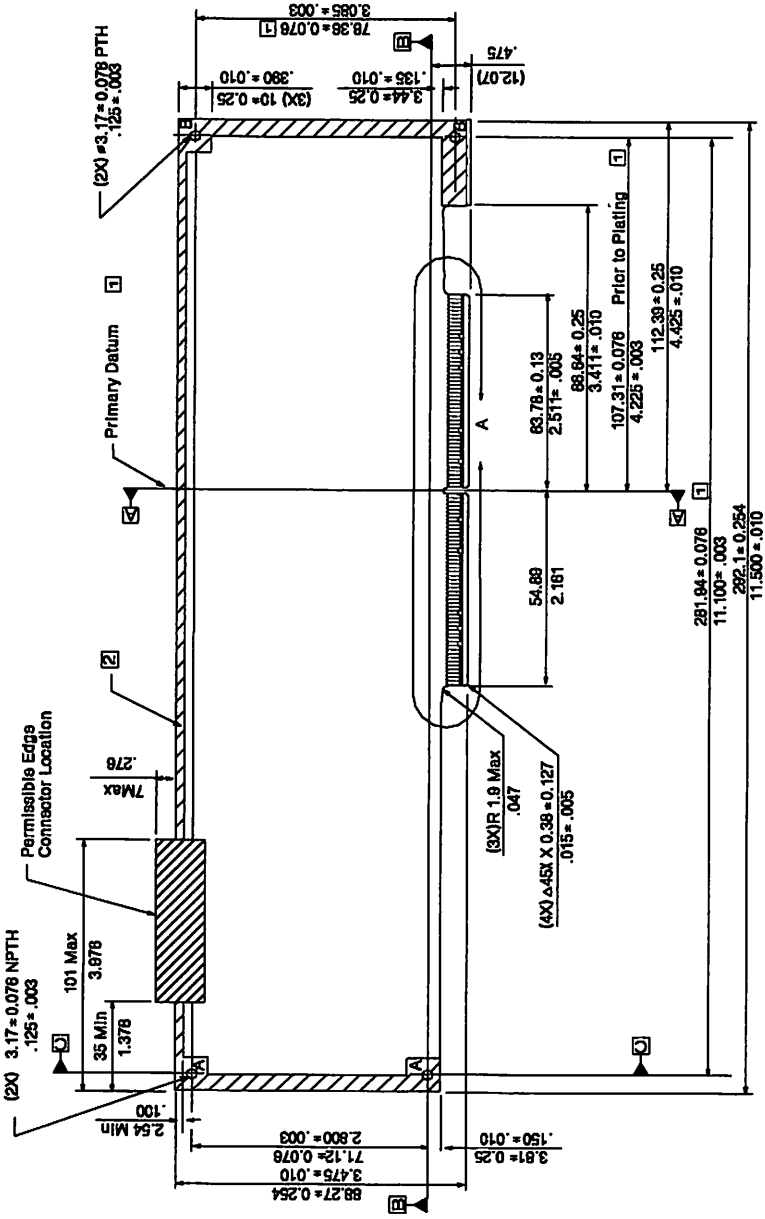


Figure 7. Adapter Dimensions (32-Bit with Matched Memory)

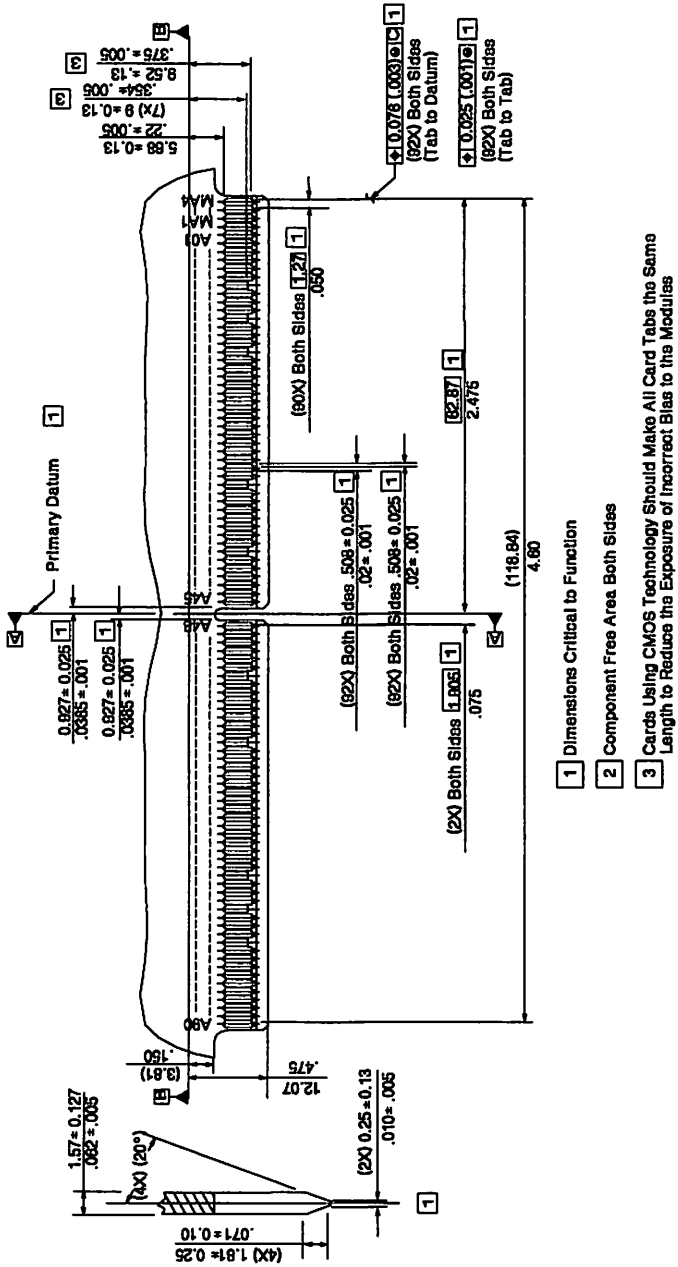


Figure 8. Connector Dimensions (32-Bit with Matched Memory)

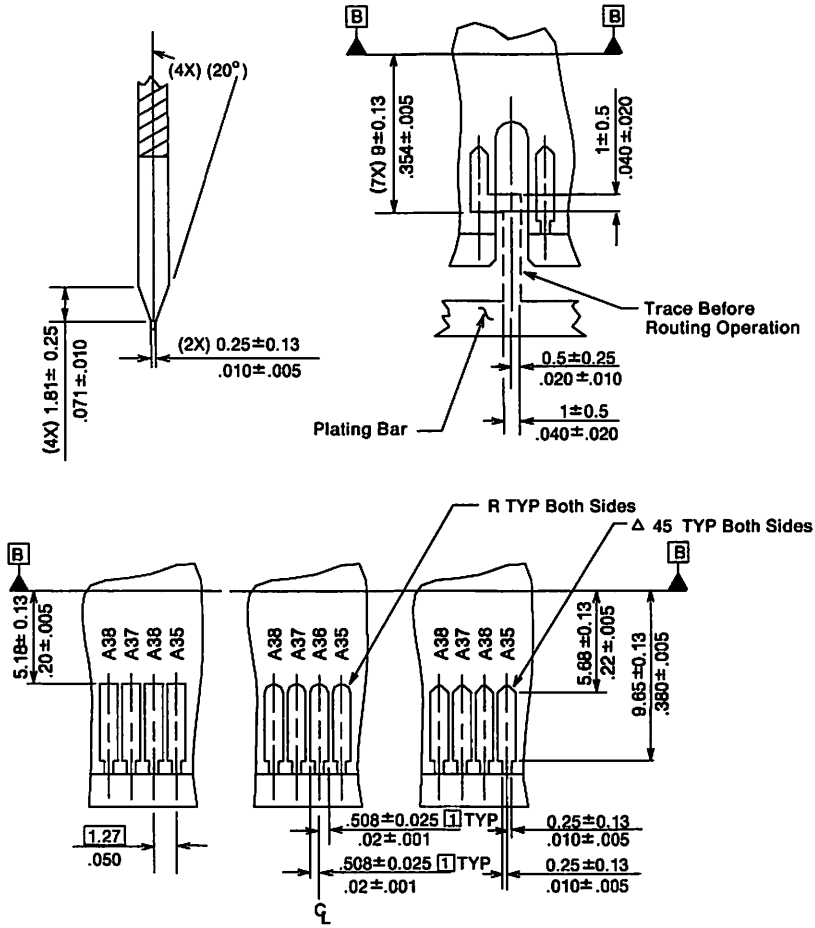
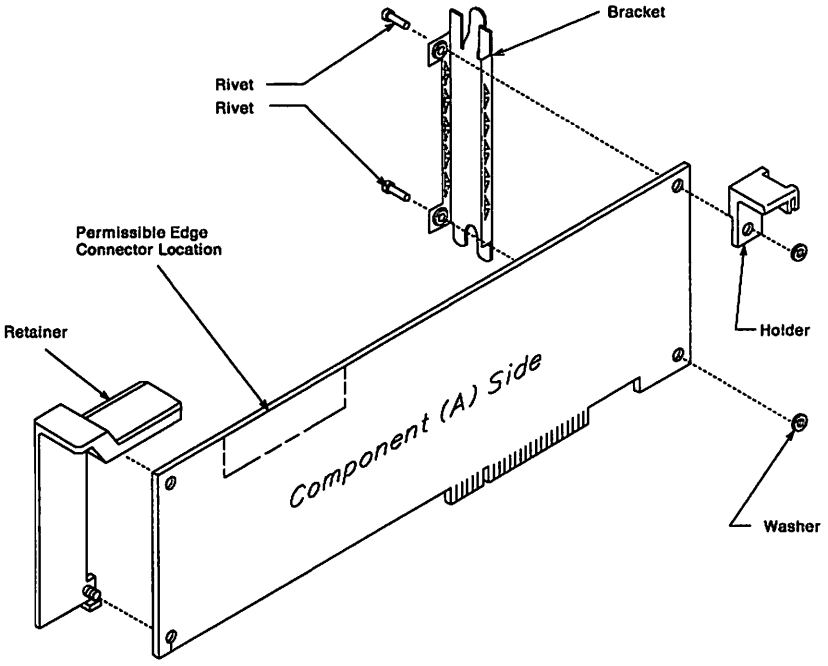


Figure 9. Connector (Common Detail)



Materials:
Holder and Retainer - Polycarbonate UL 94 V-0
Bracket - AISI Type 302 1/4 Hard Stainless Steel

Figure 10. Typical Adapter Assembly

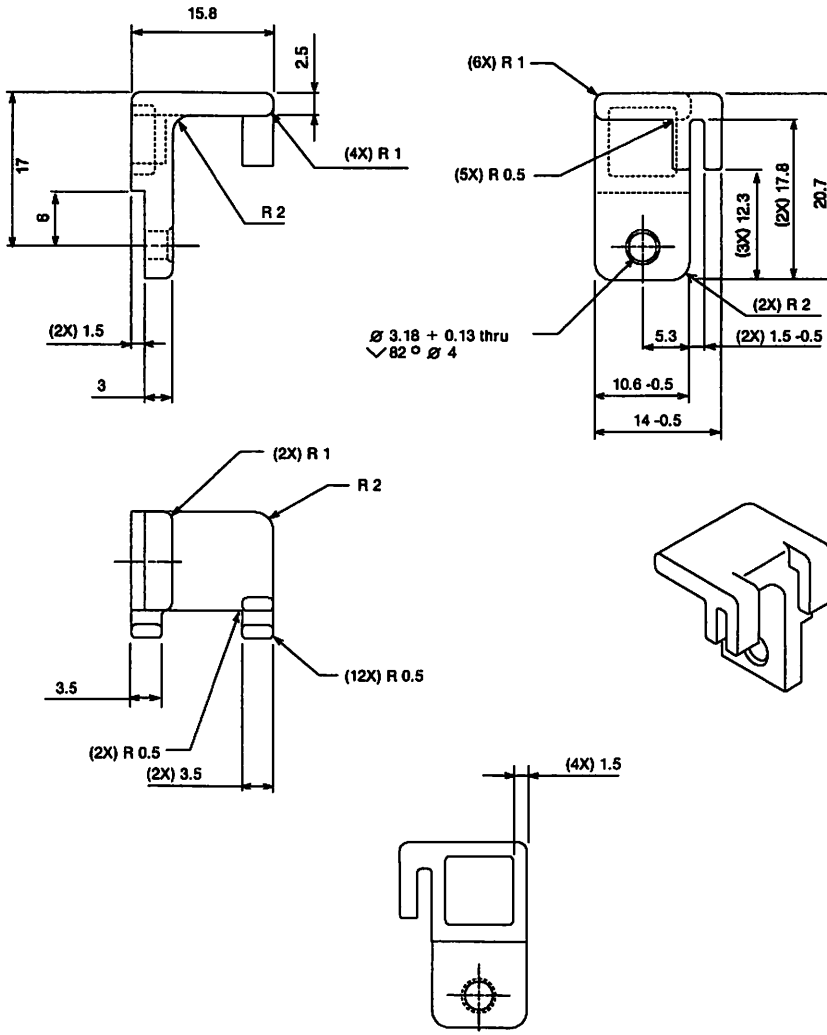


Figure 11. Adapter Holder

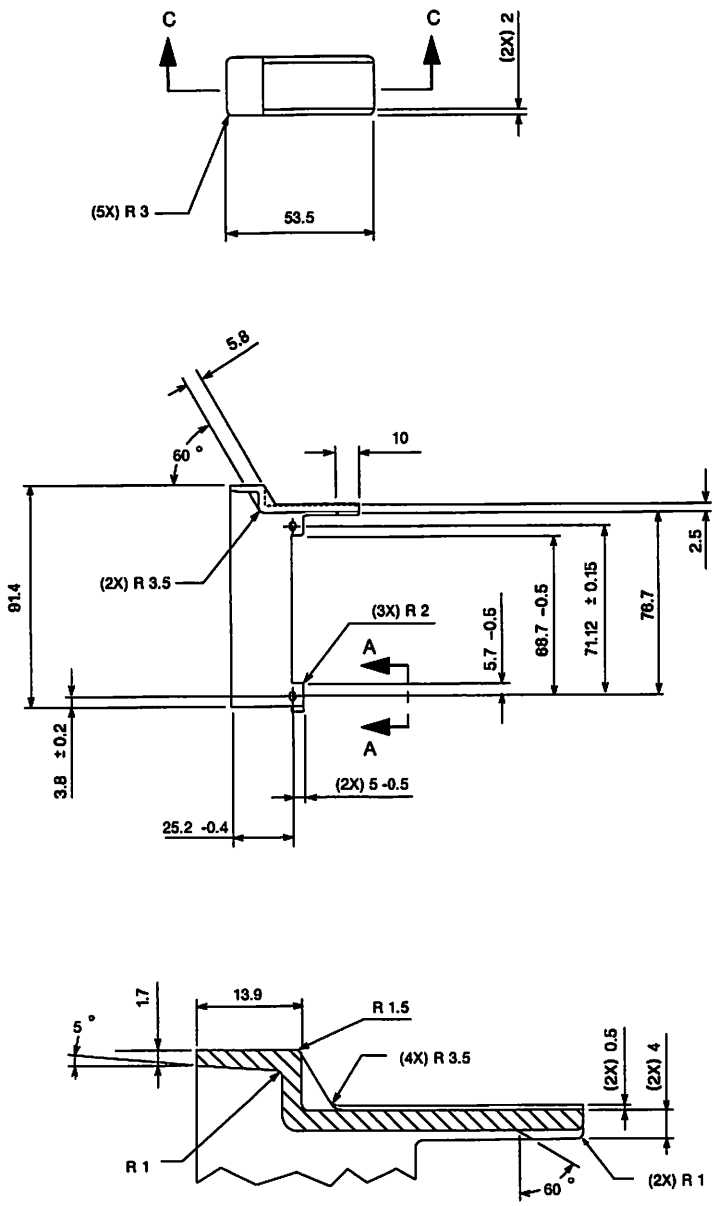
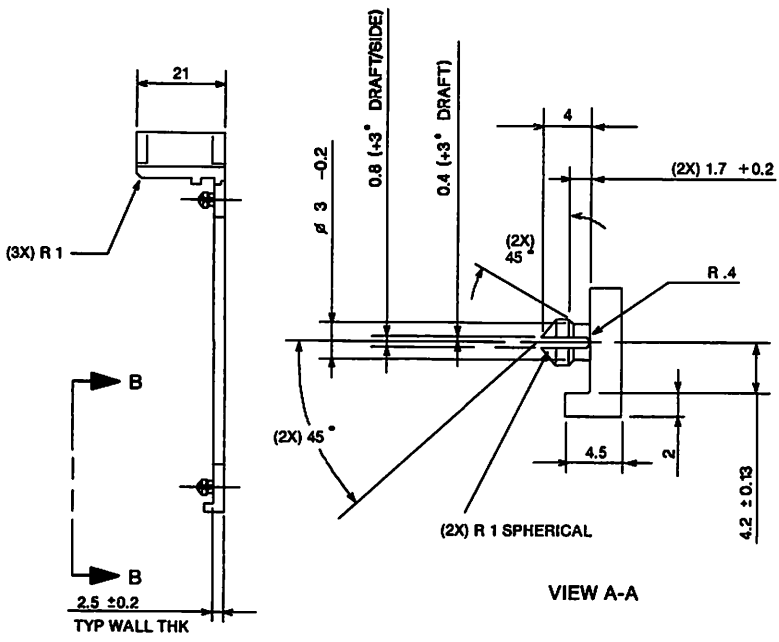


Figure 12 (Part 1 of 2). Adapter Retainer



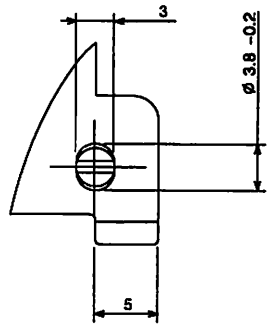
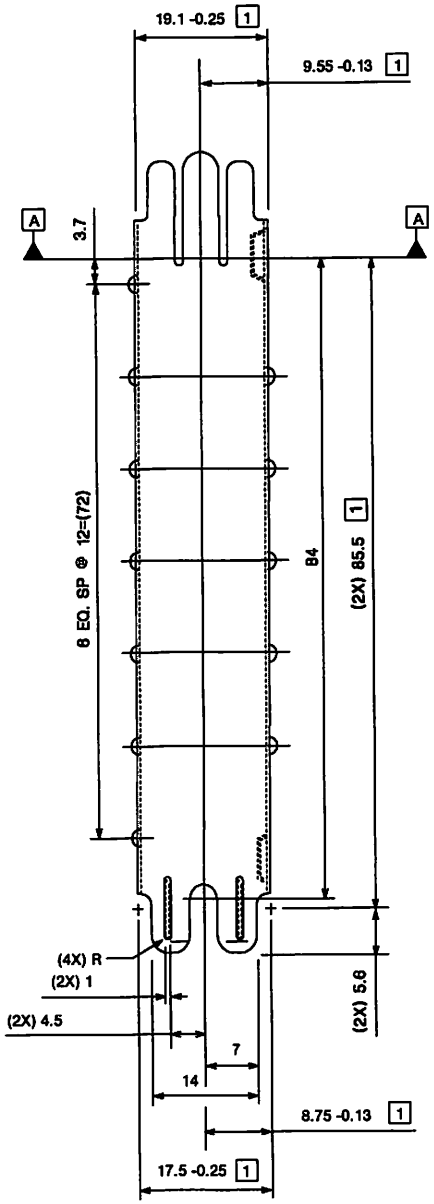


Figure 12 (Part 2 of 2). Adapter Retainer



1 DIMENSION IS CRITICAL TO FUNCTION.

Figure 13 (Part 1 of 4). Adapter Bracket

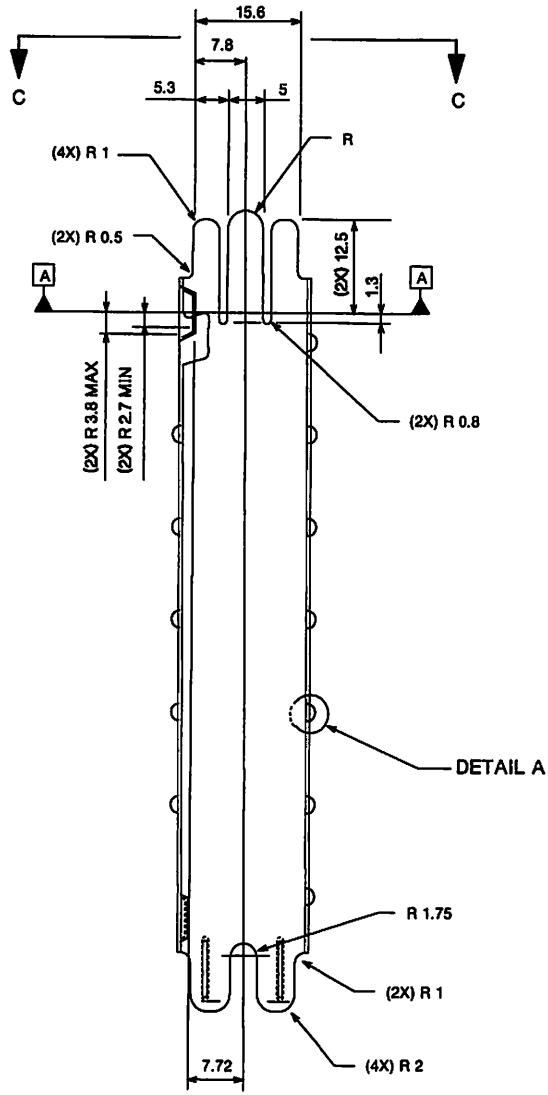


Figure 13 (Part 2 of 4). Adapter Bracket

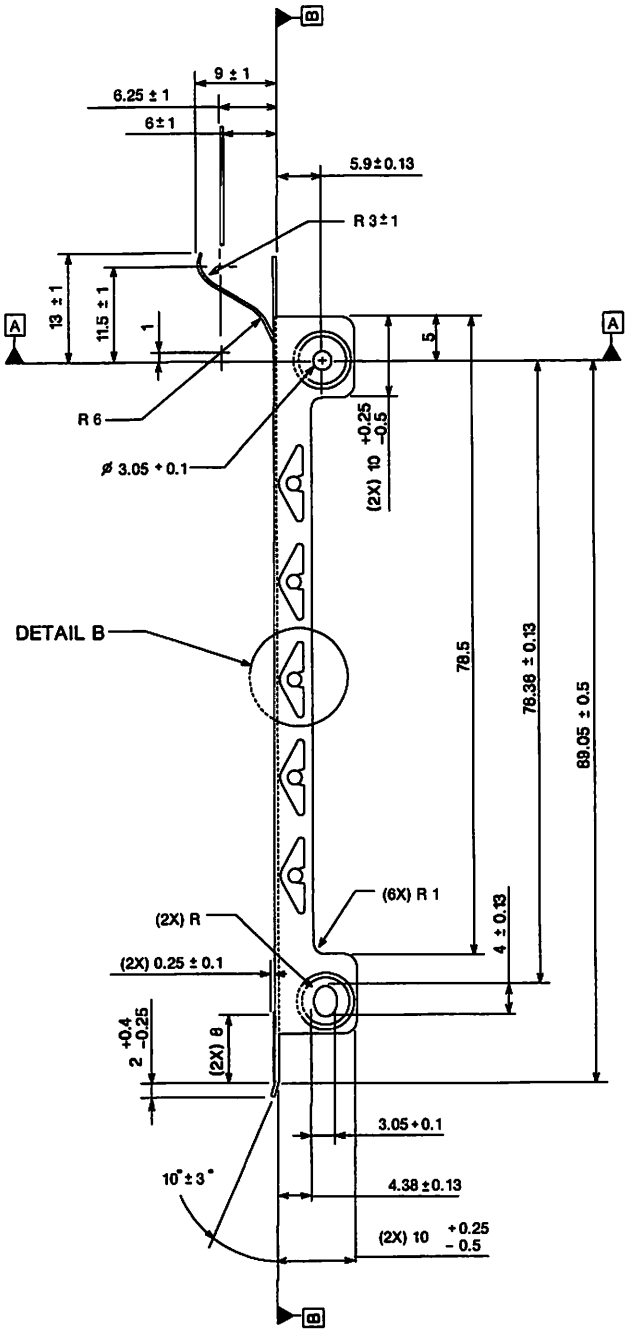
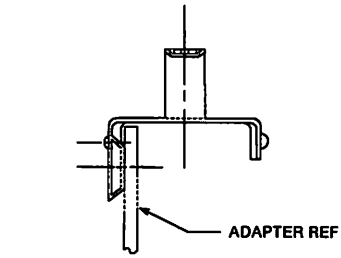
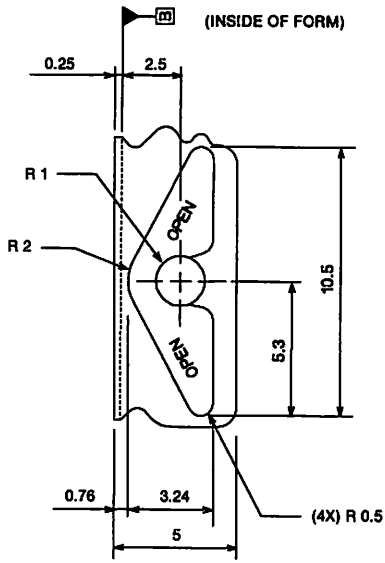


Figure 13 (Part 3 of 4). Adapter Bracket

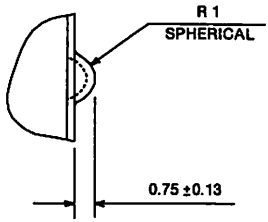


VIEW C-C



DETAIL B

SCALE 10/1
(12X)



DETAIL A

SCALE 10/1
(12X)

Figure 13 (Part 4 of 4). Adapter Bracket

Power

The allowable load current for each voltage present on each channel connector is as follows.

Supply Voltage	Maximum Current* Per 16-Bit Connector	Maximum Current Per 16/32-Bit Connector
+ 5.0 Vdc	1.6 A	2.0 A
+12.0 Vdc	0.175 A	0.175 A
−12.0 Vdc	0.040 A	0.040 A

* This maximum current may not apply to all systems.

Figure 14. Channel Load Current

The following formulas can be used to determine maximum statistical current values.

$$(I_{TC_1} + \cdots + I_{TC_n}) + \sqrt{((I_{MC_1} - I_{TC_1})^2 + \cdots + (I_{MC_n} - I_{TC_n})^2)}$$

Where:

- I_{MC} is the maximum current for each component on a given adapter.
- I_{TC} is the typical current for each component on a given adapter and the sum of all I_{TC} equals I_{TA}.

Note: If I_{MC} or I_{TC} is not available, estimate by using: $I_{TC} = 0.7 \times I_{MC}$

The total channel current is also determined in a statistical manner.
Total Channel Current =

$$(I_{TA_1} + \cdots + I_{TA_n}) + \sqrt{((I_{MA_1} - I_{TA_1})^2 + \cdots + (I_{MA_n} - I_{TA_n})^2)}$$

Where:

- I_{MA} is a maximum statistical current for a given adapter.
- I_{TA} is the typical current for a given adapter.

Voltage Regulation

The voltage regulation at the channel connector is shown in the following figure.

Voltage	Pins	Tolerance
Ground	A3, B3, B5, B9, B13, B17, B21, B25, B29,	N/A
	B33, B37, B41, A43, B45, B50, B54, B58	N/A
	A61, B63, B67, B71, B75, B79, B83, B87,	N/A
	A89, BM4*, AM2*	
+ 5.0 Vdc	A7, A11, A15, A31, A39, A48, A56, A69,	+5% −4.5%
	A73, A81, A85	
+12.0 Vdc	A19, A35, A52, A65, A77,	+5% −4.5%
−12.0 Vdc	A23, A27	+10% −9.5%
Ground (Auxiliary Video)	BV1, AV3, BV5, AV7, BV9	N/A

* These connectors are in the matched-memory portion of the connector.

Figure 15. Channel Voltage Regulation

The tolerance includes all power distribution losses in both power and ground planes, up to the pins of the channel connector. It does not include the drop due to the connector (30 milliohm maximum per contact), or the drop due to distribution within the adapter.

General Design Considerations

Each designer must take the precautions necessary to protect the safety of the end user, provide reliable operation of the device, and ensure the device does not interfere with the operation of the system or any other installed devices. The design considerations described in this section are not the **only** considerations, but rather those that might otherwise be overlooked.

Safety

Avoid exposed high-voltage or current points, sharp edges, and exposed components that operate at high temperatures. Devices must not channel dc power outside the system unit in any manner that violates Underwriters Laboratory and Canadian Standards Association guidelines.

Note: Canadian Standards Association C22.2, paragraph 4.11.3, number 154 requires protection of conductors of external interconnecting cords and cables connected to secondary circuits.

IBM does not support installing or removing adapters or components when the system power is on.

Thermal

The system unit is cooled internally by low-volume forced air. Adapter designs must allow for adequate air space between the adapters. Avoid using internal cables as a mechanism for signal communication inside the system unit, they can interfere with the air flow. If internal cables are required, they must be positioned to minimize the impact on airflow. The maximum height for components mounted on the adapter should not exceed the dimensions specified under "Dimensions" on page 1. The adapter design should avoid clustering of high-temperature components. No component should exceed its maximum thermal rating.

Electromagnetic Compatibility

Adhere to the following guidelines to reduce electromagnetic compatibility (EMC) problems.

- The adapter end brackets make a continuous 360° connection to the outside "skin" of the system unit cabinet. A similar 360° connection to the inside skin should also be provided. The adapter bracket must not be used as a dc voltage return path, a logic-ground connection, or an audio ground connection.
- The end bracket at the rear of the adapter is isolated from dc ground on the adapter. The bracket must be grounded through a screw connection to the system unit and designed as shown in Figure 13 on page 15.
- All connector ground pins must be connected to the interplane ground at the channel connector, and the +5 Vdc power must be immediately connected to the +5 Vdc power plane.
- All adapters must provide nonsegmented internal power and ground planes.

- Each surface-mount technology module position should provide a decoupling capacitor pad with minimal connection inductance. Pin-in-hole modules should be decoupled if they drive or contain edge-triggered logic. Capacitors can range between 0.01 and 0.10 microfarad and should be low-inductance ceramic or a layered design.
- Internal cables should be avoided as a mechanism for signal communication inside the system unit. The channel should not be extended outside the system unit, except by an adapter.
- Clocks should be properly imbedded and terminated. When clocks, strobes, and handshakes are generated or received, care should be taken to control the rise-and-fall times to minimize radiation.
- External cables should connect through 360° shielded D-shell or equivalent connectors. Avoid the use of "pigtail" shield connections. Shield terminations should be connected to the external shield of the cable connector. Do not bring the shield through the connector and connect it to either logic ground or the inside skin of the cabinet.
- High-current power within the system unit should provide adjacent return paths to allow the maximum cancelation of radiated magnetic fields by the mutual coupling between the supply and return lines.

Diagnostics

All writable registers typically are readable at the same address. External interfaces typically include 100% diagnostic wrap capability by electronic switching or an external wrap tool.

Design Guidelines

Adapters designed for the Micro Channel architecture must comply with the following design guidelines:

- Each I/O adapter must decode all 16 lines of the I/O address.
- Each memory adapter must decode all 24 lines of the memory address and MADE 24.

- Each 32-bit memory adapter must decode all 32 lines of the memory address if MADE 24 is inactive.
- Each adapter must replace the function of switches and jumpers with registers that incorporate POS logic.
- Each adapter must issue an *adapter ID* to the data bus when interrogated.

  To minimize the number of required drivers, only the logical 0 bits in the adapter ID need to be driven. This provides 39,202 combinations with 8 drivers or less.

  The following figure shows the recommended ID values for vendors. ID values 8100 to FFFE are assigned for IBM products only.

  Note: Programs should not make decisions based on the high nibble of the Adapter ID groupings.

ID	Definition
0000	Device Not Ready
0001 to 0FFF	Bus Master
5000 to 5FFF	Direct Memory Access Devices
6000 to 6FFF	Direct Program Control (Including Memory-Mapped I/O)
7000 to 7FFF	Storage*
8000 to 80FF	Video
FFFF	Device Not Attached

* Multiple-function adapters containing storage typically respond as storage.

Figure 16. Vendor ID Assignments

- Each enabled adapter must return a '-card selected feedback' signal (-CD SFDBK) to the system microprocessor when an access is made to the address space of the adapter, or when the adapter is selected by arbitration level. -CD SFDBK must not be generated when the '-card setup' (-CD SETUP) line is active.
- Each adapter design must be capable of degating all outputs to the system board (including -CD SFDBK, -CD DS 16/32, interrupts, and so on) if bit 0 of POS Register 2 (hex 0102) is set to 0.
- Each adapter design must implement an open-collector driver (or a tri-state driver gated negative-active) to drive the interrupt request line. The design must also implement a status register (readable at an I/O address bit position) that remains active when

the interrupt is set, and stays active until reset by the service routine. The adapter must hold the level-sensitive interrupt active until it is reset as a direct result of servicing the interrupt. The service routine must not reset the interrupt controller until after the interrupt bus signal has been reset by the adapter.

- Following a reset, each adapter must set bit 0 (Card Enable) of its POS Register 2 to 0.
- If applicable, the adapter can reside at an alternate address (corresponding to one selected by switches on a Personal Computer-type adapter).
- All adapters must provide adapter description files (.adf) on a 3.5-inch diskette for system configuration.
- To provide maximum portability, devices designed for arbitration level 0 or 1 should have limited bandwidth or short bursts so diskette overruns can be prevented or recovered by retry operations. The diskette drive controller on arbitration level 2 may be held inactive by devices on levels 0 and 1, by a refresh operation, and by the previous controlling master. The diskette drive controller should not be held inactive for more than 12 microseconds to prevent overrun.
- Adapter designs should not extend the card-edge connector beyond the basic 16- or 32-bit connector unless the signals provided by the extension are used by the adapter.

Index

A

adapter bracket 15
adapter design 1
adapter design considerations 20
adapter diagnostics 22
adapter dimensions 1, 2
adapter holder 12
adapter retainer 13
assembly, typical adapter 11
assignments, recommended 23

B

bracket, adapter 15

C

channel load current 19
circuit protection, secondary 21
CMOS technology 1
compatibility, electromagnetic 21
component height 1
conductor protection 21
connector description 10
connector dimensions 3
connector, common detail 10

D

design considerations 20
design guidelines 22
design, adapter 1
diagnostics 22
dimensions, adapter 1, 2
dimensions, connector 3
dimensions, video adapter 4
dimensions, video connector 5

E

electromagnetic compatibility 21

F

formula, maximum statistical current 19
formula, total channel current 19

G

guidelines, design 22

H

height of components 1
holder, adapter 12

L

load current 19

M

maximum statistical current formula 19
micro channel adapter design 1

P

power specifications 19
protection of conductors 21

R

recommended assignments 23
regulation, voltage 20
retainer, adapter 13

S

safety 20
secondary circuit protection 21
specifications, power 19

T

technology, CMOS 1
thermal 21
total channel current formula 19
typical adapter assembly 11

V

video adapter dimensions 4
video connector dimensions 5
voltage regulation 20

Numerics

16-bit adapter 1
16-bit adapter with video 1
32-bit adapter 1
32-bit adapter with
matched-memory extension 1

Microprocessors and Instruction Sets

80286 Microprocessor 1
Real-Address Mode 1
Protected Virtual Address Mode 2
80287 Math Coprocessor 2
Programming Interface 3
Hardware Interface 3
80386 Microprocessor 5
Real Address Mode 5
Protected Virtual Address Mode 6
Virtual 8086 Mode 7
80386 Paging Mechanism 8
80387 Math Coprocessor 10
Programming Interface 11
Hardware Interface 12
80286 Microprocessor Instruction Set 14
Data Transfer 14
Arithmetic 18
Logic 22
String Manipulation 24
Control Transfer 26
Processor Control 30
Protection Control 32
80287 Math Coprocessor Instruction Set 35
Data Transfer 35
Comparison 37
Constants 38
Arithmetic 39
Transcendental 40
Processor Control 41
Introduction to the 80386 Instruction Set 43
Code and Data Segment Descriptors 43
Prefixes 44
Instruction Format 45
Encoding 46
Address Mode 47
Operand Length (w) Field 50
Segment Register (sreg) Field 51
General Register (reg) Field 51
Operation Direction (d) Field 52

Sign-Extend (s) Field 52
Conditional Test (tttn) Field 52
Control, Debug, or Test Register (eee) Field 53
80386 Microprocessor Instruction Set 54
Data Transfer 54
Segment Control 57
Flag Control 58
Arithmetic 59
Logic 64
String Manipulation 68
Repeated String Manipulation 69
Bit Manipulation 71
Control Transfer 72
Conditional Jumps 73
Conditional Byte Set 78
Interrupt Instructions 80
Processor Control 81
Processor Extension 82
Prefix Bytes 82
Protection Control 83
Introduction to the 80387 Instruction Set 86
80387 Usage of the Scale-Index-Base Byte 86
Instruction and Data Pointers 87
New Instructions 89
80387 Math Coprocessor Instruction Set 90
Data Transfer 90
Comparison 91
Constants 93
Arithmetic 94
Transcendental 95
Processor Control 96

Index 99

Figures

1. 80287 Data Types 3
2. 80386 Addressing 7
3. Paging Mechanism 8
4. Data Type Classifications and Instructions 10
5. 80387 Data Types 12
6. 2-Bit Register Field 34
7. 3-Bit Register Field 34
8. 80287 Encoding Field Summary 35
9. 80386 Code and Data Segment Descriptor Format 43
10. Instruction Format 45
11. 80386 Instruction Set Encoding Field Summary 46
12. Effective Address (16-Bit and 32-Bit Address Modes) 47
13. Scale Factor (s-i-b Byte Present) 48
14. Index Registers (s-i-b Byte Present) 48
15. Base Registers (s-i-b Byte Present) 49
16. Effective Address (32-Bit Address Mode — s-i-b Byte Present) 50
17. Operand Length Field Encoding 51
18. Segment Register Field Encoding 51
19. General Register Field Encoding 51
20. Operand Direction Field Encoding 52
21. Sign-Extend Field Encoding 52
22. Conditional Test Field Encoding 53
23. Control, Debug, and Test Register Field Encoding 53
24. 80387 Encoding Field Summary 86
25. Instruction and Pointer Image (16-Bit Real Address Mode) 88
26. Instruction and Pointer Image (16-Bit Protected Mode) . . . 88
27. Instruction and Pointer Image (32-Bit Real Address Mode) 88
28. Instruction and Pointer Image (32-Bit Protected Mode) . . . 89

Notes:

80286 Microprocessor

The 80286 microprocessor subsystem has the following:

- 24-bit address
- 16-bit data interface
- Extensive instruction set, including string I/O
- Hardware fixed-point multiply and divide
- Two operational modes:
  - 8086-compatible Real Address
  - Protected Virtual Address.
- 16MB (MB equals 1,048,576 or 2^{20} bytes) of physical address space
- 1GB (GB equals 1,073,741,824 or 2^{30} bytes) of virtual address space.

Real-Address Mode

In the real-address mode, the address space of the system microprocessor is a contiguous array of up to 1MB. The system microprocessor generates 20-bit physical addresses to address memory.

The segment portion of the pointer is interpreted as the upper 16 bits of a 20-bit segment address; the lower 4 bits are always 0. Therefore, segment addresses begin on multiples of 16 bytes.

All segments in the real-address mode are 64KB (KB equals 1024 bytes) and can be read, written, or executed. An exception or interrupt can occur if data operands or instructions attempt to wrap around the end of a segment (for example, a word with its low-order byte at offset hex FFFF and its high-order byte at hex 0000). If, in the real-address mode, the information contained in the segment does not use the full 64KB, the unused end of the segment can be overlaid by another segment to reduce physical memory requirements.

Protected Virtual Address Mode

The protected virtual address mode (hereafter called protected mode) offers extended physical and virtual memory address space, memory protection mechanisms, and new operations to support operating systems and virtual memory.

The protected mode provides a virtual address space of 1GB for each task mapped into a 16MB physical address space. The virtual address space may be larger than the physical address space, because any use of an address that does not map to a physical memory location will cause a restartable exception.

Like the real-address mode, the protected mode uses 32-bit pointers, consisting of 16-bit selector and offset components. The selector specifies an index into a memory-resident table rather than the upper 16 bits of a real address. The 24-bit base address of the desired segment is obtained from a table in memory. The 16-bit offset is added to the segment base address to form the physical address. The system microprocessor automatically refers to the tables whenever a segment register is loaded with a selector. All instructions that load a segment register refer to the table without additional program support. Each entry in a table is 8-bytes.

80287 Math Coprocessor

The optional 80287 Math Coprocessor enables the system to perform high-speed arithmetic, logarithmic, and trigonometric operations. The coprocessor works in parallel with the microprocessor. The parallel operation decreases operating time by allowing the coprocessor to do mathematical calculations while the microprocessor continues to do other functions.

The coprocessor works with seven numeric data types, which are divided into the following three classes:

- Binary integers (three types)
- Decimal integers (one type)
- Real numbers (three types).

Programming Interface

The coprocessor offers extended data types, registers, and instructions to the microprocessor. The coprocessor has eight 80-bit registers, which provide the equivalent capacity of forty 16-bit registers. This register space allows constants and temporary results to be held in registers during calculations, thus reducing memory access, improving speed, and increasing bus availability. The register space can be used as a stack or as a fixed register set. When used as a stack, only the top two stack elements are operated on.

The following figure shows representations of large and small numbers in each data type.

Data Type	Bits	Significant Digits (Decimal)	Approximate Range (Decimal)
Word Integer	16	4	$-32{,}768 \leq x \leq +32{,}767$
Short Integer	32	9	$-2 \times 10^{9} \leq x \leq +2 \times 10^{9}$
Long Integer	64	19	$-9 \times 10^{18} \leq x \leq +9 \times 10^{18}$
Packed Decimal	80	18	$-9..99 \leq x \leq +9..99$ (18 digits)
Short Real *	32	6 - 7	$8.43 \times 10^{-37} \leq x \leq 3.37 \times 10^{38}$
Long Real *	64	15 - 16	$4.19 \times 10^{-307} \leq x \leq 1.67 \times 10^{308}$
Temporary Real **	80	19	$3.4 \times 10^{-4932} \leq x \leq 1.2 \times 10^{4932}$

* The Short Real and Long Real Data Types Correspond to the Single- and Double-precision Data Types.

** The Temporary Real Data Type Corresponds to the Extended-precision Data Type.

Figure 1. 80287 Data Types

Hardware Interface

The coprocessor uses the same clock generator as the microprocessor and operates in the asynchronous mode. The coprocessor is wired so that it functions as an I/O device through I/O port addresses hex 00F8, 00FA, and 00FC. The microprocessor sends opcodes and operands through these I/O ports. It also receives and stores results through the same I/O ports. The coprocessor 'busy' signal informs the microprocessor that it is executing; the

microprocessor Wait instruction forces the microprocessor to wait until the coprocessor is finished executing.

The coprocessor detects six different exception conditions that can occur during instruction execution:

- Invalid operation
- Denormal operand
- Zero-divide
- Overflow
- Underflow
- Precision.

If the appropriate exception mask bit within the coprocessor is not set, the coprocessor activates the 'error' signal. The 'error' signal generates a hardware interrupt (IRQ 13) causing the 'busy' signal to be held in the busy state. The 'busy' signal may be cleared by an 8-bit I/O Write command to address hex 00F0, with D7 through D0 equal to 0. This action also clears IRQ 13.

The power-on self-test code in the system ROM enables IRQ 13 and sets up its vector to point to a routine in ROM. The ROM routine clears the 'busy' signal latch and then transfers control to the address pointed to by the non-maskable interrupt (NMI) vector. This maintains code compatibility across the IBM Personal Computer and Personal System/2 product lines. The NMI handler reads the coprocessor status to determine if the coprocessor generated the NMI. If it wasn't generated by the coprocessor, control is passed to the original NMI handler.

The coprocessor has two operating modes: real-address mode and protected mode. They are similar to the two modes of the microprocessor. The coprocessor is in the real-address mode if reset by a power-on reset, system reset, or I/O write operation to port hex 00F1. This mode is compatible with the 8087 Math Coprocessor used in IBM Personal Computers. The coprocessor is placed in the protected mode by executing the SETPM ESC instruction. It is placed back in the real-address mode by an I/O write operation to port hex 00F1, with D7 through D0 equal to 0.

Detailed information for the internal functions of the 80287 Math Coprocessor is in the books listed in the Bibliography. Also see "Compatibility" for more information.

80386 Microprocessor

The 80386 microprocessor subsystem has the following:

- 32-bit address
- 32-bit data interface
- Extensive instruction set, including string I/O
- Hardware fixed-point multiply and divide
- Three operational modes:
  - Real Address
  - Protected Virtual Address
  - Virtual 8086.
- 4GB of physical address space
- 8 general-purpose 32-bit registers
- 64TB (TB equals 1,099,511,627,776 or 2^{40} bytes) of total virtual-address space.

Real Address Mode

In the real-address mode, the address space of the system microprocessor is a contiguous array of up to 1MB. The system microprocessor generates 20-bit physical addresses to address memory.

The segment portion of the pointer is interpreted as the upper 16 bits of a 20-bit segment address; the lower 4 bits are always 0. Therefore, segment addresses begin on multiples of 16 bytes.

All segments in the real-address mode are 64KB and can be read, written, or executed. An exception or interrupt can occur if data operands or instructions attempt to wrap around the end of a segment (for example, a word with its low-order byte at offset hex FFFF and its high-order byte at hex 0000). If, in the real-address mode, the information contained in the segment does not use the full 64KB, the unused end of the segment can be overlaid by another segment to reduce physical memory requirements.

Protected Virtual Address Mode

The protected virtual-address mode offers extended physical and virtual memory address space, memory protection mechanisms, and new operations to support operating systems and virtual memory.

The protected mode provides up to 64TB of virtual address space for each task mapped into a 4GB physical address space.

From a programmer's point of view, the main difference between the real-address mode and protected mode is the increased address space and the method of calculating the base address. The protected mode uses 32- or 48-bit pointers, consisting of 16-bit selector and 16- or 32-bit offset components. The selector specifies an index into one of two memory-resident tables, the global descriptor table (GDT) or the local descriptor table (LDT). These tables contain the 32-bit base address of a given segment. The 32-bit effective offset is added to the segment base address to form the physical address. The system microprocessor automatically refers to the tables whenever a segment register is loaded with a selector. All instructions that load a segment register refer to the memory-resident tables without additional program support. The memory-resident tables contain 8-byte values called descriptors.

The paging option provides an additional way of managing memory in the very large segments of the 80386. Paging operates in the protected mode only, beneath segmentation. The paging mechanism translates the protected linear address (which comes from the segmentation unit) into a physical address. When paging is not enabled, the physical address is the same as the linear address. The following figure shows the 80386 addressing mechanism.

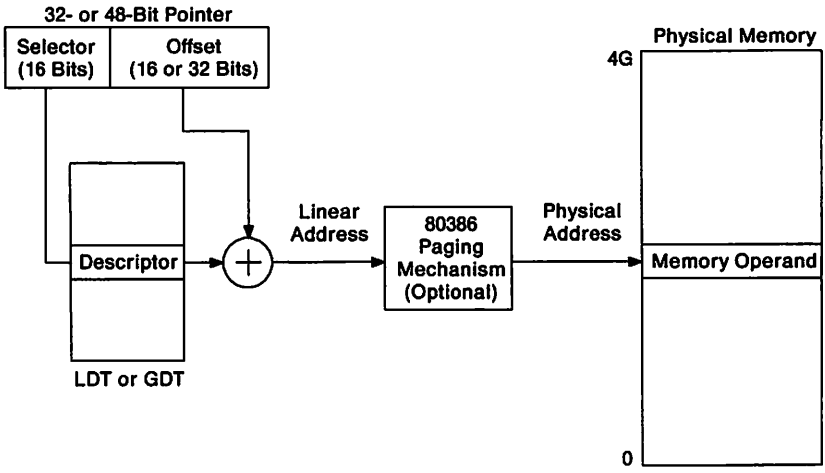


Figure 2. 80386 Addressing

Virtual 8086 Mode

The virtual-8086 mode ensures compatibility of programs written for 8086- and 8088-based systems by establishing a protected 8086 environment within the 80386 multitasking framework.

Since the address space of an 8086 is limited to 1MB, the logical addresses generated by the virtual-8086 mode lie within the first 1MB of the 80386 linear address space. To support multiple virtual-8086 tasks, paging can be used to give each virtual-8086 task a 1MB address space anywhere in the 80386 physical address space.

On a task-by-task basis, the value of the virtual-8086 flag (VM86 flag in the Flags register) determines whether the 80386 behaves as an 80386 or as an 8086. Some instructions, such as Clear Interrupt Flag, can disrupt all operations in a multitasking environment. The 80386 raises an exception when a virtual-8086 mode task attempts to execute an I/O instruction, interrupt-related instruction, or other sensitive instruction. Anytime an exception or interrupt occurs, the 80386 leaves the virtual 8086 mode, making the full resources of the 80386 available to an interrupt handler or exception handler. These handlers can determine if the source of the exception was a virtual-8086 mode task by inspecting the VM86 flag in the Flags image on the stack. If the source is a virtual-8086 mode task, the handler

calls on a routine in the operating system to simulate an 8086 instruction and return to the virtual-8086 mode.[1]

80386 Paging Mechanism

The 80386 uses two levels of tables to translate the linear address from the segmentation unit into a physical address. There are three components to the paging mechanism:

- Page directory
- Page tables
- Page frame (the page itself).

The following figure shows how the two-level paging mechanism works.

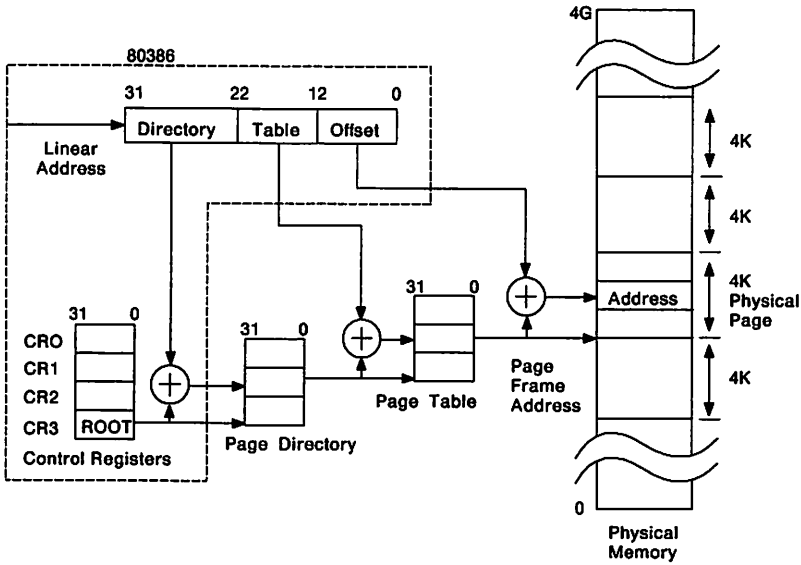


Figure 3. Paging Mechanism

CR2 is the Page-Fault Linear-Address register. It holds the 32-bit linear address that caused the last detected page fault.

---

[1] The routine in the operating system, called a *virtual machine monitor*, simulates a limited number of 8086 instructions.

CR3 is the Page Directory Physical Base Address register. It contains the physical starting address of the page directory.

The page directory is 4KB and allows up to 1024 page-directory entries. Each page-directory entry contains the address of the next level of tables, the page tables, and information about the page tables. The upper 10 bits of the linear address (A22 through A31) are used as an index to select the correct page-directory entry.

Each page table is 4KB and holds up to 1024 page-table entries. Page-table entries contain the starting address of the page frame and statistical information about the page. Address bits A12 through A21 are used as an index to select one of the 1024 page-table entries. The upper 20 bits of the page-frame address (from the page-table entry) are linked with the lower 12 bits of the linear address to form the physical address. The page-frame address bits become the most-significant bits; the linear-address bits become the least-significant bits.

80387 Math Coprocessor

The optional 80387 Math Coprocessor enables the system to perform high-speed arithmetic, logarithmic, and trigonometric operations. The 80387 effectively extends the 80386 register and instruction set for existing data types and also adds several new data types. The following figure shows the four data type classifications and the instructions associated with each.

Classification	Size	Instructions
Integer	16, 32, 64 Bits	Load, Store, Compare, Add, Subtract, Multiply, Divide
Packed BCD*	80 Bits	Load, Store
Real	32, 64 Bits	Load, Store, Compare, Add, Subtract, Multiply, Divide
Temporary Real	80 Bits	Add, Subtract, Multiply, Divide, Square Root, Scale, Remainder, Integer Part, Change, Sign, Absolute Value, Extract Exponent and Significand, Compare, Examine, Test, Exchange Tangent, Arctangent, $2^X - 1$, $Y*Log_2 (X+1)$, $Y*Log_2 (X)$, Load Constant (0.0, π, etc.), Sine, Cosine, Unordered Compare

* BCD = Binary-coded decimal

Figure 4. Data Type Classifications and Instructions

The 80386/80387 configuration fully conforms to the ANSI[2] and IEEE[3] floating-point standard and are upward, object-code compatible from 80286/80287- and 8086/8087-based systems.

[2] American National Standards Institute

[3] Institute of Electrical and Electronics Engineers

Programming Interface

The 80387 is not sensitive to the processing mode of the 80386. The 80387 functions the same whether the 80386 is executing in real-address mode, protected mode, or virtual-8086 mode. All memory access is handled by the 80386; the 80387 merely operates on instructions and values passed to it by the 80386.

All communication between the 80386 and 80387 is transparent to application programs. The 80386 automatically controls the 80387 whenever a numeric instruction is executed. All physical and virtual memory is available for storage of instructions and operands of programs that use the 80387. All memory address modes, including use of displacement, base register, index register, and scaling are available for addressing numeric operands.

The coprocessor has eight 80-bit registers. The total capacity of these eight registers is equivalent to twenty 32-bit registers. This register space allows constants and temporary results to be held in registers during calculations, thus reducing memory access, improving speed, and increasing bus availability. The register set can be used as a stack or as a fixed register set. When it is used as a stack, only the top two stack elements are operated on.

The following figure shows the seven data types supported by the 80387 Math Coprocessor.

Data Type	Range	Precision
Word Integer	10^4	16 Bits
Short Integer	10^9	32 Bits
Long Integer	10^{19}	64 Bits
Packed BCD	10^{18}	18 Digits (2 digits per byte)
Single Precision (Short Real)	$10^{\pm 38}$	24 Bits
Double Precision (Long Real)	$10^{\pm 308}$	53 Bits
Extended Precision (Temporary Real)	$10^{\pm 4932}$	64 Bits

Figure 5. 80387 Data Types

Hardware Interface

The 80387 Math Coprocessor uses the same clock generator as the 80386 system microprocessor. The coprocessor is wired so that it functions as an I/O device through I/O port addresses hex 00F8, 00FA, and 00FC. The system microprocessor sends opcodes and operands through these I/O ports. The coprocessor 'busy' signal informs the system microprocessor that it is executing; the system microprocessor Wait instruction forces the system microprocessor to wait until the coprocessor is finished executing.

The coprocessor detects six different exception conditions that can occur during instruction execution:

- Invalid operation
- Denormal operand
- Zero-divide
- Overflow
- Underflow
- Precision.

If the appropriate exception mask bit within the coprocessor is not set, the coprocessor activates the 'error' signal. The 'error' signal

generates a hardware interrupt (IRQ 13) causing the 'busy' signal to be held in the busy state. The 'busy' signal may be cleared by an 8-bit I/O Write command to address hex 00F0, with D7 through D0 equal to 0. This action also clears IRQ 13.

The power-on self-test code in the system ROM enables IRQ 13 and sets up its vector to point to a routine in ROM. The ROM routine clears the 'busy' signal latch and then transfers control to the address pointed to by the (NMI) vector. This maintains code compatibility across the IBM Personal Computer and Personal System/2 product lines. The NMI handler reads the status of the coprocessor to determine if the coprocessor generated the NMI. If it wasn't generated by the coprocessor, control is passed to the original NMI handler.

Detailed information about the internal functions of the 80387 Math Coprocessor is in the books listed in the Bibliography. Also see "Compatibility" for more information.

80286 Microprocessor Instruction Set

Data Transfer

MOV = Move

Register to Register/Memory

1000100w	mod reg r/m

Register/Memory to Register

1000101w	mod reg r/m

Immediate to Register/Memory

1100011w	mod 000 r/m	data	data if w = 1

Immediate to Register

1011wreg	data	data if w = 1

Memory to Accumulator

1010000w	addr-low	addr-high

Accumulator to Memory

1010001w	addr-low	addr-high

Register/Memory to Segment Register

10001110	mod 0 reg r/m

Segment Register to Register/Memory

10001100	mod 0 reg r/m

PUSH = Push

Memory

1 1 1 1 1 1 1 1	mod 1 1 0 r/w

Register

0 1 0 1 0 reg

Segment Register

0 0 0 reg 1 1 0

Immediate

0 1 1 0 1 0 s 0	data	data if s = 0

PUSHA = Push All

0 1 1 0 0 0 0 0

POP = Pop

Register/Memory

1 0 0 0 1 1 1 1	mod 0 0 0 r/m

Register

0 1 0 1 1 reg

Segment Register

0 0 0 reg 1 1 1	reg ≠ 0 1

POPA = Pop All

01100001

XCHG = Exchange

Register/Memory with Register

1000011w	mod reg r/m

Register with Accumulator

10010 reg

IN = Input From

Fixed Port

1110010w	port

Variable Port

1110110w

OUT = Output To

Fixed Port

1110011w	port

Variable Port

1110111w

XLAT = Translate Byte to AL

11010111

LEA = Load EA to Register

10001101	mod reg r/m

LDS = Load Pointer to DS

11000101	mod reg r/m mod ≠ 11

LES = Load Pointer to ES

11000100	mod reg r/m mod ≠ 11

LAHF = Load AH with Flags

10011111

SAHF = Store AH with Flags

10011110

PUSHF = Push Flags

10011100

POPF = Pop Flags

10011101

Arithmetic

ADD = Add

Register/Memory with Register to Either

0 0 0 0 0 0 dw	mod reg r/m

Immediate to Register/Memory

1 0 0 0 0 0 sw	mod 0 0 0 r/m	data	data if sw = 0 1

Immediate to Accumulator

0 0 0 0 0 1 0 w	data	data if w = 1

ADC = Add with Carry

Register/Memory with Register to Either

0 0 0 1 0 0 dw	mod reg r/m

Immediate to Register/Memory

1 0 0 0 0 0 sw	mod 0 1 0 r/m	data	data if sw = 0 1

Immediate to Accumulator

0 0 0 1 0 1 0 w	data	data if w = 1

INC = Increment

Register/Memory

1 1 1 1 1 1 1 w	mod 0 0 0 r/m

Register

0 1 0 0 0 reg

SUB = Subtract

Register/Memory with Register to Either

0 0 1 0 1 0 dw	mod reg r/m

Immediate from Register/Memory

1 0 0 0 0 0 sw	mod 1 0 1 r/m	data	data if sw = 0 1

Immediate from Accumulator

0 0 1 0 1 1 0 w	data	data if w = 1

SBB = Subtract with Borrow

Register/Memory with Register to Either

0 0 0 1 1 0 dw	mod reg r/m

Immediate from Register/Memory

1 0 0 0 0 0 sw	mod 0 1 1 r/m	data	data if sw = 0 1

Immediate from Accumulator

0 0 0 1 1 1 0 w	data	data if w = 1

DEC = Decrement

Register/Memory

1 1 1 1 1 1 1 w	mod 0 0 1 r/m

Register

0 1 0 0 1 reg

CMP = Compare

Register/Memory with Register

0 0 1 1 1 0 1 w	mod reg r/m

Register with Register/Memory

0 0 1 1 1 0 0 w	mod reg r/m

Immediate with Register/Memory

1 0 0 0 0 0 sw	mod 1 1 1 r/m	data	data if sw = 0 1

Immediate with Accumulator

0 0 1 1 1 1 0 w	data	data if w = 1

NEG = Change Sign

1 1 1 1 0 1 1 w	mod 0 1 1 r/m

AAA = ASCII Adjust for Add

0 0 1 1 0 1 1 1

DAA = Decimal Adjust for Add

0 0 1 0 0 1 1 1

AAS = ASCII Adjust for Subtract

0 0 1 1 1 1 1 1

DAS = Decimal Adjust for Subtract

0 0 1 0 1 1 1 1

MUL = Multiply (Unsigned)

1 1 1 1 0 1 1 w	mod 1 0 0 r/m

IMUL = Integer Multiply (Signed)

1 1 1 1 0 1 1 w	mod 1 0 1 r/m

IIMUL = Integer Immediate Multiply (Signed)

0 1 1 0 1 0 s 1	mod reg r/m	data	data if s = 0

DIV = Divide (Unsigned)

1 1 1 1 0 1 1 w	mod 1 1 0 r/m

IDIV = Integer Divide (Signed)

1 1 1 1 0 1 1 w	mod 1 1 1 r/m

AAM = ASCII Adjust for Multiply

1 1 0 1 0 1 0 0	0 0 0 0 1 0 1 0

AAD = ASCII Adjust for Divide

1 1 0 1 0 1 0 1	0 0 0 0 1 0 1 0

CBW = Convert Byte to Word

1 0 0 1 1 0 0 0

CWD = Convert Word to Doubleword

1 0 0 1 1 0 0 1

Logic

Shift/Rotate Instructions

Register/Memory by 1

1 1 0 1 0 0 0 w	mod T T T r/m

Register/Memory by CL

1 1 0 1 0 0 1 w	mod T T T r/m

Register/Memory by Count

1 1 0 0 0 0 0 w	mod T T T r/m	count

T T T	Instruction
0 0 0	ROL
0 0 1	ROR
0 1 0	RCL
0 1 1	RCR
1 0 0	SHL/SAL
1 0 1	SHR
1 1 1	SAR

AND = And

Register/Memory and Register to Either

0 0 1 0 0 0 dw	mod reg r/m

Immediate to Register/Memory

1 0 0 0 0 0 0 w	mod 100 r/m	data	data if w = 1

Immediate to Accumulator

0 0 1 0 0 1 0 w	data	data if w = 1

TEST = AND Function to Flags; No Result

Register/Memory and Register

1000010w	mod reg r/m

Immediate Data and Register/Memory

1111011w	mod 000 r/m	data	data if w = 1

Immediate Data and Accumulator

1010100w	data	data if w = 1

Or = Or

Register/Memory and Register to Either

000010dw	mod reg r/m

Immediate to Register/Memory

1000000w	mod 001 r/m	data	data if w = 1

Immediate to Accumulator

0000110w	data	data if w = 1

XOR = Exclusive OR

Register/Memory and Register to Either

001100dw	mod reg r/m

Immediate to Register/Memory

1000000w	mod 110 r/m	data	data if w = 1

Immediate to Accumulator

0011010w	data	data if w = 1

NOT = Invert Register/Memory

1 1 1 1 0 1 1 w	mod 0 1 0 r/m

String Manipulation

MOVS = Move Byte Word

1 0 1 0 0 1 0 w

CMPS B/W = Compare Byte/Word

1 0 1 0 0 1 1 w

SCAS = Scan Byte/Word

1 0 1 0 1 1 1 w

LODS = Load Byte/Word to AL/AX

1 0 1 0 1 1 0 w

STOS = Store Byte/Word from AL/AX

1 0 1 0 1 0 1 w

INS = Input Byte/Word from DX Port

0 1 1 0 1 1 0 w

OUTS = Output Byte/Word to DX Port

0 1 1 0 1 1 1 w

REP/REPNE, REPZ/REPNZ = Repeat String

Repeat Move String

11110011	1010010w

Repeat Compare String (z/Not z)

1111001z	1010011w

Repeat Scan String (z/Not z)

1111001z	1010111w

Repeat Load String

11110011	1010110w

Repeat Store String

11110011	1010101w

Repeat Input String

11110011	0110110w

Repeat Output String

11110011	0110111w

Control Transfer

CALL = Call

Direct within Segment

1 1 1 0 1 0 0 0	disp-low	disp-high

Register/Memory Indirect within Segment

1 1 1 1 1 1 1 1	mod 0 1 0 r/m

Direct Intersegment

1 0 0 1 1 0 1 0	Segment Offset	Segment Selector

Indirect Intersegment

1 1 1 1 1 1 1 1	mod 0 1 1 r/m (mod $\neq$ 11)

JMP = Unconditional Jump

Short/Long

1 1 1 0 1 0 1 1	disp-low

Direct within Segment

1 1 1 0 1 0 0 1	disp-low	disp-high

Register/Memory Indirect within Segment

1 1 1 1 1 1 1 1	mod 1 0 0 r/m

Direct Intersegment

1 1 1 0 1 0 1 0	Segment Offset	Segment Selector

Indirect Intersegment

11111111	mod 1 0 1 r/m (mod ≠ 11)

RET = Return from Call

Within Segment

11000011

Within Segment Adding Immediate to SP

11000010	data-low	data-high

Intersegment

11001011

Intersegment Adding Immediate to SP

11001010	data-low	data-high

JE/JZ = Jump on Equal/Zero

01110100	disp

JL/JNGE = Jump on Less/Not Greater, or Equal

01111100	disp

JLE/JNG = Jump on Less, or Equal/Not Greater

01111110	disp

JB/JNAE = Jump on Below/Not Above, or Equal

01110010	disp

JBE/JNA = Jump on Below, or Equal/Not Above

01110110	disp

JP/JPE = Jump on Parity/Parity Even

01111010	disp

JO = Jump on Overflow

01110000	disp

JS = Jump on Sign

01111000	disp

JNE/JNZ = Jump on Not Equal/Not Zero

01110101	disp

JNL/JGE = Jump on Not Less/Greater, or Equal

01111101	disp

JNLE/JG = Jump on Not Less, or Equal/Greater

01111111	disp

JNB/JAE = Jump on Not Below/Above, or Equal

01110011	disp

JNBE/JA = Jump on Not Below, or Equal/Above

01110111	disp

JNP/JPO = Jump on Not Parity/Parity Odd

01111011	disp

JNO = Jump on Not Overflow

01110001	disp

JNS = Jump on Not Sign

01111001	disp

LOOP = Loop CX Times

11100010	disp

LOOPZ/LOOPE = Loop while Zero/Equal

11100001	disp

LOOPNZ/LOOPNE = Loop while Not Zero/Not Equal

11100000	disp

JCXZ = Jump on CX Zero

11100011	disp

ENTER = Enter Procedure

11001000	data-low	data-high	L

LEAVE = Leave Procedure

11001001

INT = Interrupt

Type Specified

11001101	type

Type 3

11001100

INTO = Interrupt on Overflow

11001110

IRET = Interrupt Return

11001111

BOUND = Detect Value Out of Range

01100010	mod reg r/m

Processor Control

CLC = Clear Carry

11111000

CMC = Complement Carry

11110101

STC = Set Carry

11111001

CLD = Clear Direction

11111100

STD = Set Direction

11111101

CLI = Clear Interrupt

11111010

STI = Set Interrupt Enable Flag

11111011

HLT = Halt

11110100

WAIT = Wait

10011011

LOCK = Bus Lock Prefix

11110000

CTS = Clear Task Switched Flag

00001111	00000110

ESC = Processor Extension Escape

11011TTT	mod LLL r/m

Protection Control

LGDT = Load Global Descriptor Table Register

00001111	00000001	mod 0 1 0 r/m

SGDT = Store Global Descriptor Table Register

00001111	00000001	mod 0 0 0 r/m

LIDT = Load Interrupt Descriptor Table Register

00001111	00000001	mod 0 1 1 r/m

SIDT = Store Interrupt Descriptor Table Register

00001111	00000001	mod 0 0 1 r/m

LLDT = Load Local Descriptor Table Register from Register/Memory

00001111	00000000	mod 0 1 0 r/m

SLDT = Store Local Descriptor Table Register from Register/Memory

00001111	00000000	mod 0 0 0 r/m

LTR = Load Task Register from Register/Memory

00001111	00000000	mod 0 1 1 r/m

STR = Store Task Register to Register/Memory

00001111	00000000	mod 0 0 1 r/m

LMSW = Load Machine Status Word from Register/Memory

00001111	00000001	mod 1 1 0 r/m

SMSW = Store Machine Status Word

00001111	00000001	mod 100 r/m

LAR = Load Access Rights from Register/Memory

00001111	00000010	mod reg r/m

LSL = Load Segment Limit from Register/Memory

00001111	00000011	mod reg r/m

ARPL = Adjust Requested Privilege Level from Register/Memory

01100011	mod reg r/m

VERR = Verify Read Access; Register/Memory

00001111	00000000	mod 100 r/m

VERW = Verify Write Access

00001111	00000000	mod 101 r/m

The effective address (EA) of the memory operand is computed according to the mod and r/m fields:

If mod = 11, then r/m is treated as a reg field.
If mod = 00, then disp = 0, disp-low and disp-high are absent.
If mod = 01, then disp = disp-low sign-extended to 16 bits, disp-high is absent.
If mod = 10, then disp = disp-high:disp-low.

If r/m = 000, then EA = (BX) + (SI) + DISP
If r/m = 001, then EA = (BX) + (DI) + DISP
If r/m = 010, then EA = (BP) + (SI) + DISP
If r/m = 011, then EA = (BP) + (DI) + DISP
If r/m = 100, then EA = (SI) + DISP
If r/m = 101, then EA = (DI) + DISP
If r/m = 110, then EA = (BP) + DISP
If r/m = 111, then EA = (BX) + DISP

The disp field follows the second byte of the instruction (before data if required).

Note: An exception to the above statements occurs when mod = 00 and r/m = 110, in which case EA = disp-high; disp-low.

Segment Override Prefix

0 0 1 reg 1 1 0

The 2-bit and 3-bit reg fields are defined in the following figures.

Reg	Segment Register	Reg	Segment Register
00	ES	10	SS
01	CS	11	DS

Figure 6. 2-Bit Register Field

16-Bit (w = 1)	8-Bit (w = 0)
000 AX	000 AL
001 CX	001 CL
010 DX	010 DL
011 BX	011 BL
100 SP	100 AH
101 BP	101 CH
110 SI	110 DH
111 DI	111 BH

Figure 7. 3-Bit Register Field

The physical addresses of all operands addressed by the BP register are computed using the SS Segment register. The physical addresses of the destination operands of the string primitive operations (those addressed by the DI register) are computed using the ES segment, which may not be overridden.

80287 Math Coprocessor Instruction Set

The following is an instruction-set summary for the 80287 Math Coprocessor.

The following figure shows abbreviations used in the summary.

Field	Description	Bit Information
escape	80286 Extension Escape	Bit Pattern = 11011
MF	Memory Format	00 = 32-Bit Real 01 = 32-Bit Integer 10 = 64-Bit Real 11 = 16-Bit Integer
ST(0)	Current Stack Top	
ST(i)	ith Register Below the Stack Top	
d	Destination	0 = Destination is ST(0) 1 = Destination is ST(i)
P	Pop	0 = No pop 1 = Pop ST(0)
R	Reverse*	0 = Destination (op) source 1 = Source (op) destination

* When d = 1, reverse the sense of R.

Figure 8. 80287 Encoding Field Summary

Data Transfer

FLD = Load

Integer/Real Memory to ST(0)

escape MF 1	mod 0 0 0 r/m

Long Integer Memory to ST(0)

escape 1 1 1	mod 1 0 1 r/m

Temporary Real Memory to ST(0)

escape 0 1 1	mod 1 0 1 r/m

BCD Memory to ST(0)

escape 1 1 1	mod 1 0 0 r/m

ST(i) to ST(0)

escape 0 0 1	1 1 0 0 0 ST(i)

FST = Store

ST(0) to Integer/Real Memory

escape MF 1	mod 0 1 0 r/m

ST(0) to ST(i)

escape 1 0 1	1 1 0 1 0 ST(i)

FSTP = Store and Pop

ST(0) to Integer/Real Memory

escape MF 1	mod 0 1 1 r/m

ST(0) to Long Integer Memory

escape 1 1 1	mod 1 1 1 r/m

ST(0) to Temporary Real Memory

escape 0 1 1	mod 1 1 1 r/m

ST(0) to BCD Memory

escape 1 1 1	mod 1 1 0 r/m

ST(0) to ST(i)

escape 1 0 1	1 1 0 1 1 ST(i)

FXCH = Exchange ST(i) and ST(0)

escape 0 0 1	1 1 0 0 1 ST(i)

Comparison

FCOM = Compare

Integer/Real Memory to ST(0)

escape MF 0	mod 0 1 0 r/m

ST(i) to ST(0)

escape 0 0 0	1 1 0 1 0 ST(i)

FCOMP = Compare and Pop

Integer/Real Memory to ST(0)

escape MF 0	mod 0 1 1 r/m

ST(i) to ST(0)

escape 0 0 0	1 1 0 1 1 ST(i)

FCOMPP = Compare ST(1) to ST(0) and Pop Twice

escape 1 1 0	1 1 0 1 1 0 0 1

FTST = Test ST(0)

escape 0 0 1	1 1 1 0 0 1 0 0

FXAM = Examine ST(0)

escape 0 0 1	1 1 1 0 0 1 0 1

Constants

FLDZ = Load + 0.0 into ST(0)

escape 0 0 1	1 1 1 0 1 1 1 0

FLD1 = Load + 1.0 into ST(0)

escape 0 0 1	1 1 1 0 1 0 0 0

FLDPI = Load π into ST(0)

escape 0 0 1	1 1 1 0 1 0 1 1

FLDL2T = Load $\log_2 10$ into ST(0)

escape 0 0 1	1 1 1 0 1 0 0 1

FLDL2E = Load $\log_2 e$ into ST(0)

escape 0 0 1	1 1 1 0 1 0 1 0

FLDLG2 = Load $\log_{10} 2$ into ST(0)

escape 0 0 1	1 1 1 0 1 1 0 0

FLDLN2 = Load $\log_e 2$ into ST(0)

escape 0 0 1	1 1 1 0 1 1 0 1

Arithmetic

FADD = Addition

Integer/Real Memory with ST(0)

escape	MF	0	mod	0	0	0	r/m

ST(i) and ST(0)

escape	dP	0	1	1	0	0	0	ST(i)

FSUB = Subtraction

Integer/Real Memory with ST(0)

escape	MF	0	mod	1	0	R	r/m

ST(i) and ST(0)

escape	dP	0	1	1	1	0	R	r/m

FMUL = Multiplication

Integer/Real Memory with ST(0)

escape	MF	0	mod	0	0	1	r/m

ST(i) and ST(0)

escape	dP	0	1	1	0	0	1	r/m

FDIV = Division

Integer/Real Memory with ST(0)

escape	MF	0	mod	1	1	R	r/m

ST(i) and ST(0)

escape	dP	0	1	1	1	1	R	r/m

FSQRT = Square Root of ST(0)

escape 0 0 1	1 1 1 1 1 0 1 0

FSCALE = Scale ST(0) by ST(1)

escape 0 0 1	1 1 1 1 1 1 0 1

FPREM = Partial Remainder of ST(0) ÷ ST(1)

escape 0 0 1	1 1 1 1 1 0 0 0

FRNDINT = Round ST(0) to Integer

escape 0 0 1	1 1 1 1 1 1 0 0

FXTRACT = Extract Components of ST(0)

escape 0 0 1	1 1 1 1 0 1 0 0

FABS = Absolute Value of ST(0)

escape 0 0 1	1 1 1 0 0 0 0 1

FCHS = Change Sign of ST(0)

escape 0 0 1	1 1 1 0 0 0 0 0

Transcendental

FPTAN = Partial Tangent of ST(0)

escape 0 0 1	1 1 1 1 0 0 1 0

FPATAN = Partial Arctangent of ST(1) ÷ ST(0)

escape 0 0 1	1 1 1 1 0 0 1 1

F2XM1 = $2^{ST(0)}$ -1

escape 0 0 1	1 1 1 1 0 0 0 0

FYL2X = ST(1) x Log_2 [ST(0)]

escape 0 0 1	1 1 1 1 0 0 0 1

FYL2XP1 = ST(1) x Log_2 [ST(0) + 1]

escape 0 0 1	1 1 1 1 1 0 0 1

Processor Control

FINIT = Initialize NPX

escape 0 1 1	1 1 1 0 0 0 1 1

FSETPM = Enter Protected Mode

escape 0 1 1	1 1 1 0 0 1 0 0

FSTSW AX = Store Control Word

escape 1 1 1	1 1 1 0 0 0 0 0

FLDCW = Load Control Word

escape 0 0 1	mod 1 0 1 r/m

FSTCW = Store Control Word

escape 0 0 1	mod 1 1 1 r/m

FSTSW = Store Status Word

escape 1 0 1	mod 1 1 1 r/m

FCLEX = Clear Exceptions

escape 0 1 1	1 1 1 0 0 0 1 0

FSTENV = Store Environment

escape 0 0 1	mod 1 1 0 r/m

FLDENV = Load Environment

escape 0 0 1	mod 1 0 0 r/m

FSAVE = Save State

escape 1 0 1	mod 1 1 0 r/m

FRSTOR = Restore State

escape 1 0 1	mod 1 0 0 r/m

FINCSTP = Increment Stack Pointer

escape 0 0 1	1 1 1 1 0 1 1 1

FDECSTP = Decrement Stack Pointer

escape 0 0 1	1 1 1 1 0 1 1 0

FFREE = Free ST(I)

escape 1 0 1	1 1 0 0 0 ST(i)

FNOP = No Operation

escape 0 0 1	1 1 0 1 0 0 0 0

Introduction to the 80386 Instruction Set

The 80386 instruction set is an extended version of the 8086 and 80286 instruction sets. The instruction sets have been extended in two ways:

- The instructions have extensions that allow operations on 32-bit operands, registers, and memory.
- A 32-bit addressing mode allows flexible selection of registers for base and index as well as index scaling capabilities (x2, x4, x8) for computing a 32-bit effective address. The 32-bit effective address yields a 4GB address range.

  Note: The effective address size must be less than 64KB in the real-address or virtual-address modes to avoid an exception.

Code and Data Segment Descriptors

Although the 80386 supports all 80286 Code and Data segment descriptors, there are some differences in the format. The 80286 segment descriptors contain a 24-bit base address and a 16-bit limit field, while the 80386 segment descriptors have a 32-bit base address, a 20-bit limit field, a default bit, and a granularity bit.

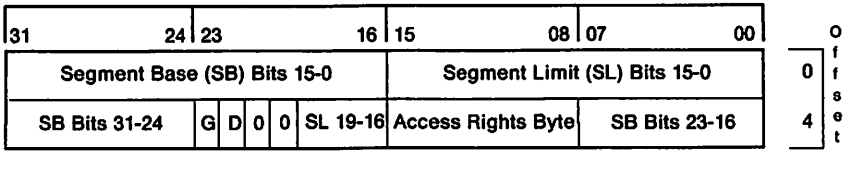


Figure 9. 80386 Code and Data Segment Descriptor Format

Note: Bits 31 through 16 shown at offset 4 are set to 0 for all 80286 segment descriptors.

The default (D) bit of the code segment register is used to determine whether the instruction is carried out as a 16-bit or 32-bit instruction. Code segment descriptors are not used in either the real-address mode or the virtual-8086 mode. When the system microprocessor is operating in either of these modes, a D-bit value of 0 is assumed and

operations default to a 16-bit length compatible with 8086 and 80286 programs.

The granularity (G) bit is used to determine the granularity of the segment length (1 = page granular, 0 = byte granular). If the value of the 20 segment-limit bits is defined as *N*, a G-bit value of 1 defines the segment size as follows:

Segment size = (*N* + 1) x 4KB

4KB represents the size of a page.

Prefixes

Two prefixes have been added to the instruction set. The Operand Size prefix overrides the default selection of the operand size; the Effective Address Size prefix overrides the effective address size. The presence of either prefix toggles the default setting to its opposite condition. For example:

- If the operand size defaults to 32-bit data operations, the presence of the Operand Size prefix sets it for 16-bit data operations.
- If the effective address size is 16-bits, the presence of the Effective Address Size prefix toggles the instruction to use 32-bit effective address computations.

The prefixes are available in all 80386 modes, including the real-address mode and the virtual-8086 mode. Since the default of these modes is always 16 bits, the prefixes are used to specify 32-bit operations. If needed, either or both of the prefixes may precede any opcode bytes and affect only the instruction they precede.

Instruction Format

The instructions are presented in this format:

Opcode	Mode Specifier	Address Displacement	Immediate Data

Term	Description
Opcode	The opcode may be one or two bytes in length. Within each byte, smaller encoding fields may be defined.
Mode Specifier	Consists of the "mod r/m" byte and the "scale-index-base" (s-i-b) byte. The mod r/m byte specifies the address mode to be used. Format: mod T T T r/m The "s-i-b" byte is optional and can be used only in 32-bit address modes. It follows the mod r/m byte to fully specify the manner in which the effective address is computed. Format: ss index base
Address Displacement	Follows the "mod r/m" byte or "s-i-b" byte. It may be 8, 16, or 32 bits.
Immediate Data	If specified, follows any displacement bytes and becomes the last field of the instruction. It may be 8, 16, or 32 bits. The term "8-bit data" indicates a fixed data length of 8 bits. The term "8-, 16-, or 32-bit data" indicates a variable data length. The length is determined by the w field and the current operand size. If w = 0, the data is always 8 bits. If w = 1, the size is determined by the operand size of the instruction.

Figure 10. Instruction Format

The instructions use a variety of fields to indicate register selection, the addressing mode, and so on. The following figure is a summary of the fields.

Field Name	Description	Bit Information
w	Specifies if data is byte or full size. (Full size is either 16 or 32 bits.)	1
d	Specifies the direction of data operation.	1
s	Specifies if an immediate data field must be sign-extended.	1
reg	General address specifier.	3
mod r/m	Address mode specifier (effective address can be a general register).	2 for mod; 3 for r/m
ss	Scale factor for scaled index address mode.	2
index	General register to be used as an index register.	3
base	General register to be used as base register.	3
sreg2	Segment register specifier for CS, SS, DS, and ES.	2
sreg3	Segment register specifier for CS, SS, DS, ES, FS, and GS.	3
tttn	For conditional instructions; specifies a condition asserted or a condition negated.	4

Figure 11. 80386 Instruction Set Encoding Field Summary

Encoding

This section defines the encoding of the fields used in the instruction sets.

Address Mode

The first addressing byte is the "mod r/m" byte. The effective address (EA) of the memory operand is computed according to the mod and r/m fields. The mod r/m byte can be interpreted as either a 16-bit or 32-bit addressing mode specifier. Interpretation of the byte depends on the address components used to calculate the EA. The following figure defines the encoding of 16-bit and 32-bit addressing modes with the mod r/m byte.

mod r/m	16-Bit Mode	32-Bit Mode (No s-i-b byte)
00 000	DS:[BX + SI]	DS:[EAX]
00 001	DS:[BX + DI]	DS:[ECX]
00 010	SS:[BP + SI]	DS:[EDX]
00 011	SS:[BP + DI]	DS:[EBX]
00 100	DS:[SI]	s-i-b present (see Figure 16 on page 50)
00 101	DS:[DI]	DS:d32
00 110	d16	DS:[ESI]
00 111	DS:[BX]	DS:[EDI]
01 000	DS:[BX + SI + d8]	DS:[EAX + d8]
01 001	DS:[BX + DI + d8]	DS:[ECX + d8]
01 010	SS:[BP + SI + d8]	DS:[EDX + d8]
01 011	SS:[BP + DI + d8]	DS:[EBX + d8]
01 100	DS:[SI + d8]	s-i-b present (see Figure 16 on page 50)
01 101	DS:[DI + d8]	SS:[EBP + d8]
01 110	SS:[BP + d8]	DS:[ESI + d8]
01 111	DS:[BX + d8]	DS:[EDI + d8]
10 000	DS:[BX + SI + d16]	DS:[EAX + d32]
10 001	DS:[BX + DI + d16]	DS:[ECX + d32]
10 010	SS:[BP + SI + d16]	SS:[EDX + d32]
10 011	SS:[BP + DI + d16]	DS:[EBX + d32]
10 100	DS:[SI + d16]	s-i-b present (see Figure 16 on page 50)
10 101	DS:[DI + d16]	SS:[EBP + d32]
10 110	SS:[BP + d16]	DS:[ESI + d32]
10 111	DS:[BX + d16]	DS:[EDI + d32]

Figure 12. Effective Address (16-Bit and 32-Bit Address Modes)

The displacement follows the second byte of the instruction (before data, if required).

The scale-index-base (s-i-b) byte can be specified as a second byte of addressing information. The s-i-b byte is specified when using a

32-bit addressing mode and the mod r/m byte has the following values:

- r/m = 100
- mod = 00, 01, or 10.

When the s-i-b byte is present, the 32-bit effective address is a function of the mod, ss, index, and base fields. The following figures show the scale factor, Index register selected, and base register selected when the s-i-b byte is present.

ss	Scale Factor
00	1
01	2
10	4
11	8

Figure 13. Scale Factor (s-i-b Byte Present)

Index	Index Register	
000	EAX	
001	ECX	
010	EDX	
011	EBX	
100	No Index Register	The ss field must equal 00 when the index field is 100; if not, the effective address is undefined.
101	EBP	
110	ESI	
111	EDI	

Figure 14. Index Registers (s-i-b Byte Present)

base	Base Register	
000	EAX	
001	ECX	
010	EDX	
011	EBX	
100	ESP	
101	EBP	If mod = 00, then EBP is not used to form the EA; immediate 32-bit address displacement follows the mode specifier byte.
110	ESI	
111	EDI	

Figure 15. Base Registers (s-i-b Byte Present)

The scaled-index information is determined by multiplying the contents of the Index register by the scale factor. The following example shows the use of the 32-bit addressing mode with scaling where:

- EAX is the base of ARRAY_A
- ECX is the index of the desired element
- 2 is the scale factor.

```
; ARRAY_A is an array of words
MOV EAX, offset ARRAY_A
MOV ECX, element_number
MOV BX, [EAX][ECX*2]
```

The following figure defines the encoding of the 32-bit addressing mode when the s-i-b byte is present.

Note: The mod field is from the mod r/m byte. The base field and scaled-index information are from the s-i-b byte.

Mod Base	32-Bit Address Mode
00 000	DS:[EAX + (scaled index)]
00 001	DS:[ECX + (scaled index)]
00 010	DS:[EDX + (scaled index)]
00 011	DS:[EBX + (scaled index)]
00 100	SS:[ESP + (scaled index)]
00 101	DS:[d32 + (scaled index)]
00 110	DS:[ESI + (scaled index)]
00 111	DS:[EDI + (scaled index)]
01 000	DS:[EAX + (scaled index) + d8]
01 001	DS:[ECX + (scaled index) + d8]
01 010	DS:[EDX + (scaled index) + d8]
01 011	DS:[EBX + (scaled index) + d8]
01 100	SS:[ESP + (scaled index) + d8]
01 101	SS:[EBP + (scaled index) + d8]
01 110	DS:[ESI + (scaled index) + d8]
01 111	DS:[EDI + (scaled index) + d8]
10 000	DS:[EAX + (scaled index) + d32]
10 001	DS:[ECX + (scaled index) + d32]
10 010	DS:[EDX + (scaled index) + d32]
10 011	DS:[EBX + (scaled index) + d32]
10 100	SS:[ESP + (scaled index) + d32]
10 101	SS:[EBP + (scaled index) + d32]
10 110	DS:[ESI + (scaled index) + d32]
10 111	DS:[EDI + (scaled index) + d32]

Figure 16. Effective Address (32-Bit Address Mode — s-i-b Byte Present)

Operand Length (w) Field

For an instruction performing a data operation, the instruction is executed as either a 32-bit or 16-bit operation. Within the constraints of the operation size, the w field encodes the operand size as either one byte or full operation.

w	16-Bit Data Operation	32-Bit Data Operation
0	8 Bits	8 Bits
1	16 Bits	32 Bits

Figure 17. Operand Length Field Encoding

Segment Register (sreg) Field

The 2-bit segment register field (sreg2) allows one of the four 80286 segment registers to be specified. The 3-bit segment register (sreg3) allows the 80386 FS and GS segment registers to be specified.

sreg2	sreg3	Segment Register
00	000	ES
01	001	CS
10	010	SS
11	011	DS
--	100	FS
--	101	GS
--	110	Reserved
--	111	Reserved

Figure 18. Segment Register Field Encoding

General Register (reg) Field

The General register is specified by the reg field, which may appear in the primary opcode bytes as the reg field of the mod reg r/m byte, or as the r/m field of the mod reg r/m byte when mod = 11.

reg	16-Bit (without w)	16-Bit (w = 0)	16-Bit (w = 1)	32-Bit (without w)	32-Bit (w = 0)	32-Bit (w = 1)
000	AX	AL	AX	EAX	AL	EAX
001	CX	CL	CX	ECX	CL	ECX
010	DX	DL	DX	EDX	DL	EDX
011	BX	BL	BX	EBX	BL	EBX
100	SP	AH	SP	ESP	AH	ESP
101	BP	CH	BP	EBP	CH	EBP
110	SI	DH	SI	ESI	DH	ESI
111	DI	BH	DI	EDI	BH	EDI

Figure 19. General Register Field Encoding

The physical addresses of all operands addressed by the BP register are computed using the SS Segment register. For string primitive

operations (those addressed by the DI register), addresses of the destination operands are computed using the ES segment, which may not be overridden.

Operation Direction (d) Field

The operation direction (d) field is used in many two-operand instructions to indicate which operand is the source and which is the destination.

d	Direction of Operation
0	Register/Memory <-- Register The "reg" field indicates the source operand; "mod r/m" or "mod ss index base" indicates the destination operand.
1	Register<-- Register/Memory The "reg" field indicates the destination operand; "mod r/m" or "mod ss index base" indicates the source operand.

Figure 20. Operand Direction Field Encoding

Sign-Extend (s) Field

The sign-extend (s) field appears primarily in instructions having immediate data fields. The s field affects only 8-bit immediate data being placed in a 16-bit or 32-bit destination.

s	8-Bit Immediate Data	16/32-Bit Immediate Data
0	No effect on data	No effect on data
1	Sign-extend 8-bit data to fill 16-bit or 32-bit destination	No effect on data

Figure 21. Sign-Extend Field Encoding

Conditional Test (tttn) Field

For conditional instructions (conditional jumps and set-on condition), the conditional test (tttn) field is encoded, with n indicating whether to use the condition (n = 0) or its negation (n = 1), and ttt defining the condition to test.

tttn	Condition	Mnemonic
0000	Overflow	O
0001	No Overflow	NO
0010	Below/Not Above or Equal	B/NAE
0011	Not Below/Above or Equal	NB/AE
0100	Equal/Zero	E/Z
0101	Not Equal/Not Zero	NE/NZ
0110	Below or Equal/Not Above	BE/NA
0111	Not Below or Equal/Above	NBE/A
1000	Sign	S
1001	Not Sign	NS
1010	Parity/Parity Even	P/PE
1011	Not Parity/Parity Odd	NP/PO
1100	Less Than/Not Greater or Equal	L/NGE
1101	Not Less Than/Greater or Equal	NL/GE
1110	Less Than or Equal/Not Greater Than	LE/NG
1111	Not Less or Equal/Greater Than	NLE/G

Figure 22. Conditional Test Field Encoding

Control, Debug, or Test Register (eee) Field

The following shows the encoding for loading and storing the Control, Debug, and Test registers (eee).

eee Code	Interpreted as Control Register	Interpreted as Debug Register	Interpreted as Test Register
000	CR0	DR0	---
001	---	DR1	---
010	CR2	DR2	---
011	CR3	DR3	---
100	---	---	---
101	---	---	---
110	---	DR6	TR6
111	---	DR7	TR7

Figure 23. Control, Debug, and Test Register Field Encoding

Data Transfer

MOV = Move

Register to Register/Memory

1000100w	mod reg r/m

Register/Memory to Register

1000101w	mod reg r/m

Immediate to Register/Memory

1100011w	mod 000 r/m	8-, 16-, or 32-bit data

Immediate to Register (Short Form)

1011w reg	8-, 16-, or 32-bit data

Memory to Accumulator (Short Form)

1010000w	full 16- or 32-bit displacement

Accumulator to Memory (Short Form)

1010001w	full 16- or 32-bit displacement

Register/Memory to Segment Register

10001110	mod sreg3 r/m

Segment Register to Register/Memory

10001100	mod sreg3 r/m

MOVSX = Move with Sign Extension

Register from Register/Memory

00001111	1011111w	mod reg r/m

MOVZX = Move with Zero Extension

Register from Register/Memory

00001111	1011011w	mod reg r/m

PUSH = Push

Register/Memory

11111111	mod 110 r/m

Register (Short Form)

01010 reg

Segment Register (ES, CS, SS, or DS) Short Form

000sreg2110

Segment Register (FS or GS)

00001111	10sreg3000

Immediate

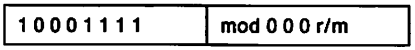
011010s0	8-, 16-, or 32-bit data

PUSHA = Push All

01100000

POP = Pop

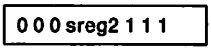
Register/Memory



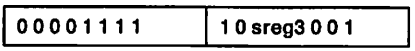
Register (Short Form)



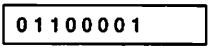
Segment Register (ES, SS, or DS) Short Form



Segment Register (FS or GS)

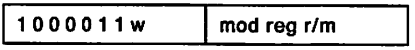


POPA = Pop All

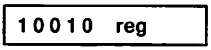


XCHG = Exchange

Register/Memory with Register

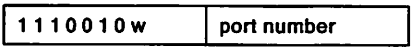


Register with Accumulator (Short Form)

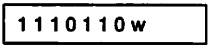


IN = Input From:

Fixed Port



Variable Port



OUT = Output To:

Fixed Port

1110011w	port number

Variable Port

1110111w

LEA = Load EA to Register

10001101	mod reg r/m

Segment Control

LDS = Load Pointer to DS

11000101	mod reg r/m

LES = Load Pointer to ES

11000100	mod reg r/m

LFS = Load Pointer to FS

00001111	10110100	mod reg r/m

LGS = Load Pointer to GS

00001111	10110101	mod reg r/m

LSS = Load Pointer to SS

00001111	10110010	mod reg r/m

Flag Control

CLC = Clear Carry Flag

11111000

CLD = Clear Direction Flag

11111100

CLI = Clear Interrupt Enable Flag

11111010

CLTS = Clear Task Switched Flag

00001111	00000110

CMC = Complement Carry Flag

11110101

LAHF = Load AH Into Flag

10011111

POPF = Pop Flags

10011101

PUSHF = Push Flags

10011100

SAHF = Store AH into Flags

10011110

STC = Set Carry Flag

11111001

STD = Set Direction Flag

11111101

STI = Set Interrupt Enable Flag

11111011

Arithmetic

ADD = Add

Register to Register

000000dw	mod reg r/m

Register to Memory

0000000w	mod reg r/m

Memory to Register

0000001w	mod reg r/m

Immediate to Register/Memory

100000sw	mod 000 r/m	8-, 16-, or 32-bit data

Immediate to Accumulator (Short Form)

0000010w	8-, 16-, or 32-bit data

ADC = Add with Carry

Register to Register

0 0 0 1 0 0 d w	mod reg r/m

Register to Memory

0 0 0 1 0 0 0 w	mod reg r/m

Memory to Register

0 0 0 1 0 0 1 w	mod reg r/m

Immediate to Register/Memory

1 0 0 0 0 0 s w	mod 0 1 0 r/m	8-, 16-, or 32-bit data

Immediate to Accumulator (Short Form)

0 0 0 1 0 1 0 w	8-, 16-, or 32-bit data

INC = Increment

Register/Memory

1 1 1 1 1 1 1 w	mod 0 0 0 r/m

Register (Short Form)

0 1 0 0 0 reg

SUB = Subtract

Register from Register

0 0 1 0 1 0 d w	mod reg r/m

Register from Memory

0 0 1 0 1 0 0 w	mod reg r/m

Memory from Register

0 0 1 0 1 0 1 w	mod reg r/m

Immediate from Register/Memory

1 0 0 0 0 0 s w	mod 1 0 1 r/m	8-, 16-, or 32-bit data

Immediate from Accumulator (Short Form)

0 0 1 0 1 1 0 w	8-, 16-, or 32-bit data

SBB = Subtract with Borrow

Register from Register

0 0 0 1 1 0 d w	mod reg r/m

Register from Memory

0 0 0 1 1 0 0 w	mod reg r/m

Memory from Register

0 0 0 1 1 0 1 w	mod reg r/m

Immediate from Register/Memory

1 0 0 0 0 0 s w	mod 0 1 1 r/m	8-, 16-, or 32-bit data

Immediate from Accumulator (Short Form)

0 0 0 1 1 1 0 w	8-, 16-, or 32-bit data

DEC = Decrement

Register/Memory

1 1 1 1 1 1 1 w	mod 0 0 1 r/m

Register (Short Form)

0 1 0 0 1 reg

CMP = Compare

Register with Register

0 0 1 1 1 0 d w	mod reg r/m

Memory with Register

0 0 1 1 1 0 0 w	mod reg r/m

Register with Memory

0 0 1 1 1 0 1 w	mod reg r/m

Immediate with Register/Memory

1 0 0 0 0 0 s w	mod 1 1 1 r/m	8-, 16-, or 32-bit data

Immediate with Accumulator (Short Form)

0 0 1 1 1 1 0 w	8-, 16-, or 32-bit data

NEG = Change Sign

1 1 1 1 0 1 1 w	mod 0 1 1 r/m

AAA = ASCII Adjust for Add

0 0 1 1 0 1 1 1

AAS = ASCII Adjust for Subtract

0 0 1 1 1 1 1 1

DAA = Decimal Adjust for Add

0 0 1 0 0 1 1 1

DAS = Decimal Adjust for Subtract

0 0 1 0 1 1 1 1

MUL = Multiply (Unsigned)

Accumulator with Register/Memory

1 1 1 1 0 1 1 w	mod 1 0 0 r/m

IMUL = Integer Multiply (Signed)

Accumulator with Register/Memory

1 1 1 1 0 1 1 w	mod 1 0 1 r/m

Register with Register/Memory

0 0 0 0 1 1 1 1	1 0 1 0 1 1 1 1	mod reg r/m

Register/Memory with Immediate to Register

0 1 1 0 1 0 s 1	mod reg r/m	8-, 16-, or 32-bit data

DIV = Divide (Unsigned)

Accumulator by Register/Memory

1111011w	mod 110 r/m

IDIV = Integer Divide (Signed)

Accumulator by Register/Memory

1111011w	mod 111 r/m

AAD = ASCII Adjust for Divide

11010101	00001010

AAM = ASCII Adjust for Multiply

11010100	00001010

CBW = Convert Byte to Word

10011000

CWD = Convert Word to Doubleword

10011001

Logic

Shift/Rotate Instructions
Not Through Carry (ROL, ROR, SAL, SAR, SHL, and SHR)

Register/Memory by 1

1101000w	mod TTT r/m

Register/Memory by CL

1101001w	mod TTT r/m

Register/Memory by Immediate Count

1 1 0 0 0 0 0 w	mod T T T r/m	8-bit data

Shift/Rotate Instructions Through Carry (RCL and RCR)

Register/Memory by 1

1 1 0 1 0 0 0 w	mod T T T r/m

Register/Memory by CL

1 1 0 1 0 0 1 w	mod T T T r/m

Register/Memory by Immediate Count

1 1 0 0 0 0 0 w	mod T T T r/m	8-bit data

T T T	Instruction
0 0 0	ROL
0 0 1	ROR
0 1 0	RCL
0 1 1	RCR
1 0 0	SHL/SAL
1 0 1	SHR
1 1 1	SAR

SHLD = Shift Left Double

Register/Memory by Immediate

0 0 0 0 1 1 1 1	1 0 1 0 0 1 0 0	mod reg r/m	8-bit data

Register/Memory by CL

0 0 0 0 1 1 1 1	1 0 1 0 0 1 0 1	mod reg r/m

SHRD = Shift Right Double

Register/Memory by Immediate

00001111	10101100	mod reg r/m	8-bit data

Register/Memory by CL

00001111	10101101	mod reg r/m

AND = And

Register to Register

001000d w	mod reg r/m

Register to Memory

0010000w	mod reg r/m

Memory to Register

0010001w	mod reg r/m

Immediate to Register/Memory

100000s w	mod 100 r/m	8-, 16-, or 32-bit data

Immediate to Accumulator (Short Form)

0010010w	8-, 16-, or 32-bit data

TEST = AND Function to Flags; No Result

Register/Memory and Register

1000010w	mod reg r/m

Immediate Data and Register/Memory

1111011w	mod 000 r/m	8-, 16-, or 32-bit data

Immediate Data and Accumulator (Short Form)

1010100w	8-, 16-, or 32-bit data

OR = Or

Register to Register

000010dw	mod reg r/m

Register to Memory

0000100w	mod reg r/m

Memory to Register

0000101w	mod reg r/m

Immediate to Register/Memory

100000sw	mod 001 r/m	8-, 16-, or 32-bit data

Immediate to Accumulator (Short Form)

0000110w	8-, 16-, or 32-bit data

XOR = Exclusive OR

Register to Register

0 0 1 1 0 0 d w	mod reg r/m

Register to Memory

0 0 1 1 0 0 0 w	mod reg r/m

Memory to Register

0 0 1 1 0 0 1 w	mod reg r/m

Immediate to Register/Memory

1 0 0 0 0 0 s w	mod 1 1 0 r/m	8-, 16-, or 32-bit data

Immediate to Accumulator (Short Form)

0 0 1 1 0 1 0 w	8-, 16-, or 32-bit data

NOT = Invert Register/Memory

1 1 1 1 0 1 1 w	mod 0 1 0 r/m

String Manipulation

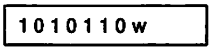
CMPS = Compare Byte Word

1 0 1 0 0 1 1 w

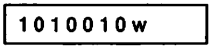
INS = Input Byte/Word from DX Port

0 1 1 0 1 1 0 w

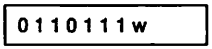
LODS = Load Byte/Word to AL/AX/EAX



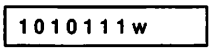
MOVS = Move Byte Word



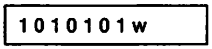
OUTS = Output Byte/Word to DX Port



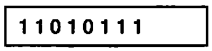
SCAS = Scan Byte Word



STOS = Store Byte/Word from AL/AX/EX



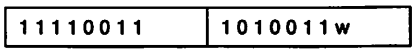
XLAT = Translate String



Repeated String Manipulation

Repeated by Count in CX or ECX

REPE CMPS = Compare String (Find Non-Match)



REPNE CMPS = Compare String (Find Match)

11110010	1010011w

REP INS = Input String

11110010	0110110w

REP LODS = Load String

11110010	1010110w

REP MOVS = Move String

11110010	1010010w

REP OUTS = Output String

11110010	0110111w

REPE SCAS = Scan String (Find Non-AL/AX/EAX)

11110011	1010111w

REPNE SCAS = Scan String (Find AL/AX/EAX)

11110010	1010111w

REP STOS = Store String

11110010	1010101w

Bit Manipulation

BSF = Scan Bit Forward

00001111	10111100	mod reg r/m

BSR = Scan Bit Reverse

00001111	10111101	mod reg r/m

BT = Test Bit

Register/Memory, Immediate

00001111	10111010	mod 100 r/m	8-bit data

Register/Memory, Register

00001111	10100011	mod reg r/m

BTC = Test Bit and Complement

Register/Memory, Immediate

00001111	10111010	mod 111 r/m	8-bit data

Register/Memory, Register

00001111	10111011	mod reg r/m

BTR = Test Bit and Reset

Register/Memory, Immediate

00001111	10111010	mod 110 r/m	8-bit data

Register/Memory, Register

00001111	10110011	mod reg r/m

BTS = Test Bit and Set

Register/Memory, Immediate

00001111	10111010	mod 101 r/m	8-bit data

Register/Memory, Register

00001111	10101011	mod reg r/m

Control Transfer

CALL = Call

Direct within Segment

11101000	full 16- or 32-bit displacement

Register/Memory Indirect within Segment

11111111	mod 010 r/m

Direct Intersegment

10011010	offset, selector

Indirect Intersegment

11111111	mod 011 r/m

JMP = Unconditional Jump

Short

11101011	8-bit disp.

Direct within Segment

11101001	full 16- or 32-bit displacement

Register/Memory Indirect within Segment

11111111	mod 100 r/m

Direct Intersegment

11101010	offset, selector

Indirect Intersegment

11111111	mod 101 r/m

RET = Return from Call

Within Segment

11000011

Within Segment Adding Immediate to SP

11000010	16-bit displacement

Intersegment

11001011

Intersegment Adding Immediate to SP

11001010	16-bit displacement

Conditional Jumps

JO = Jump on Overflow

8-Bit Displacement

01110000	8-bit disp.

Full Displacement

00001111	10000000	full 16- or 32-bit displacement

JNO = Jump on Not Overflow

8-Bit Displacement

01110001	8-bit disp.

Full Displacement

00001111	10000001	full 16- or 32-bit displacement

JB/JNAE = Jump on Below/Not Above or Equal

8-Bit Displacement

01110010	8-bit disp.

Full Displacement

00001111	10000010	full 16- or 32-bit displacement

JNB/JAE = Jump on Not Below/Above or Equal

8-Bit Displacement

01110011	8-bit disp.

Full Displacement

00001111	10000011	full 16- or 32-bit displacement

JE/JZ = Jump on Equal/Zero

8-Bit Displacement

01110100	8-bit disp.

Full Displacement

00001111	10000100	full 16- or 32-bit displacement

JNE/JNZ = Jump on Not Equal/Not Zero

8-Bit Displacement

01110101	8-bit disp.

Full Displacement

00001111	10000101	full 16- or 32-bit displacement

JBE/JNA = Jump on Below or Equal/Not Above

8-Bit Displacement

01110110	8-bit disp.

Full Displacement

00001111	10000110	full 16- or 32-bit displacement

JNBE/JA = Jump on Not Below or Equal/Above

8-Bit Displacement

01110111	8-bit disp.

Full Displacement

00001111	10000111	full 16- or 32-bit displacement

JS = Jump on Sign

8-Bit Displacement

01111000	8-bit disp.

Full Displacement

00001111	10001000	full 16- or 32-bit displacement

JNS = Jump on Not Sign

8-Bit Displacement

01111001	8-bit disp.

Full Displacement

00001111	10001001	full 16- or 32-bit displacement

JP/JPE = Jump on Parity/Parity Even

8-Bit Displacement

01111010	8-bit disp.

Full Displacement

00001111	10001010	full 16- or 32-bit displacement

JNP/JPO = Jump on Not Parity/Parity Odd

8-Bit Displacement

01111011	8-bit disp.

Full Displacement

00001111	10001011	full 16- or 32-bit displacement

JL/JNGE = Jump on Less/Not Greater or Equal

8-Bit Displacement

01111100	8-bit disp.

Full Displacement

00001111	10001100	full 16- or 32-bit displacement

JNL/JGE = Jump on Not Less/Greater or Equal

8-Bit Displacement

01111101	8-bit disp.

Full Displacement

00001111	10001101	full 16- or 32-bit displacement

JLE/JNG = Jump on Less or Equal/Not Greater

8-Bit Displacement

01111110	8-bit disp.

Full Displacement

00001111	10001110	full 16- or 32-bit displacement

JNLE/JG = Jump on Not Less or Equal/Greater

8-Bit Displacement

01111111	8-bit disp.

Full Displacement

00001111	10001111	full 16- or 32-bit displacement

JCXZ = Jump on CX Zero

11100011	8-bit disp.

JECXZ = Jump on ECX Zero

11100011	8-bit disp.

Note: The operand size prefix differentiates JCXZ from JECXZ.

LOOP = Loop CX Times

11100010	8-bit disp.

LOOPZ/LOOPE = Loop with Zero/Equal

11100001	8-bit disp.

LOOPNZ/LOOPNE = Loop while Not Zero

11100000	8-bit disp.

Conditional Byte Set

SETO = Set Byte on Overflow

To Register/Memory

00001111	10010000	mod 000 r/m

SETNO = Set Byte on Not Overflow

To Register/Memory

00001111	10010001	mod 000 r/m

SETB/SETNAE = Set Byte on Below/Not Above or Equal

To Register/Memory

00001111	10010010	mod 000 r/m

SETNB = Set Byte on Not Below/Above or Equal

To Register/Memory

00001111	10010011	mod 000 r/m

SETE/SETZ = Set Byte on Equal/Zero

To Register/Memory

00001111	10010100	mod 000 r/m

SETNE/SETNZ = Set Byte on Not Equal/Not Zero

To Register/Memory

00001111	10010101	mod 000 r/m

SETBE/SETNA = Set Byte on Below or Equal/Not Above

To Register/Memory

00001111	10010110	mod 000 r/m

SETNBE/SETA = Set Byte on Not Below or Equal/Above

To Register/Memory

00001111	10010111	mod 000 r/m

SETS = Set Byte on Sign

To Register/Memory

00001111	10011000	mod 000 r/m

SETNS = Set Byte on Not Sign

To Register/Memory

00001111	10011001	mod 000 r/m

SETP/SETPE = Set Byte on Parity/Parity Even

To Register/Memory

00001111	10011010	mod 000 r/m

SETNP/SETPO = Set Byte on Not Parity/Parity Odd

To Register/Memory

00001111	10011011	mod 000 r/m

SETL/SETNGE = Set Byte on Less/Not Greater or Equal

To Register/Memory

00001111	10011100	mod 000 r/m

SETNL/SETGE = Set Byte on Not Less/Greater or Equal

To Register/Memory

00001111	01111101	mod 000 r/m

SETLE/SETNG = Set Byte on Less or Equal/Not Greater

To Register/Memory

00001111	10011110	mod 000 r/m

SETNLE/SETG = Set Byte on Not Less or Equal/Greater

To Register/Memory

00001111	10011111	mod 000 r/m

ENTER = Enter Procedure

11001000	16-bit displacement	8-bit level

LEAVE = Leave Procedure

11001001

Interrupt Instructions

INT = Interrupt

Type Specified

11001101	type

Type 3

11001100

INTO = Interrupt 4 if Overflow Flag Set

11001110

BOUND = Interrupt 5 if Detect Value Out of Range

01100010	mod reg r/m

IRET = Interrupt Return

11001111

Processor Control

HLT = Halt

11110100

MOV = Move to and from Control/Debug/Test Registers

CR0/CR2/CR3 from Register

00001111	00100010	11eee reg

Register from CR0-3

00001111	00100000	11eee reg

DR0-3, DR6-7 from Register

00001111	00100011	11eee reg

Register from DR0-3, DR6-7

00001111	00100001	11eee reg

TR6-7 from Register

00001111	00100110	11eee reg

Register from TR6-7

00001111	00100100	11eee reg

NOP = No Operation

10010000

WAIT = Wait until BUSY Pin is Negated

10011011

Processor Extension

ESC = Processor Extension Escape

11011TTT	mod L L L r/m

Note: TTT and LLL bits are opcode information for the coprocessor.

Prefix Bytes

Address Size Prefix

01100111

Operand Size Prefix

01100110

LOCK = Bus Lock Prefix

11110000

Note: The use of LOCK is restricted to an exchange with memory, or bit test and reset type of instruction.

Segment Override Prefix

CS:

00101110

DS:

00111110

ES:

00100110

FS:

01100100

GS:

01100101

SS:

00110110

Protection Control

ARPL = Adjust Requested Privilege Level from Register/Memory

01100011	mod reg r/m

LAR = Load Access Rights from Register/Memory

00001111	00000010	mod reg r/m

LGDT = Load Global Descriptor Table Register

00001111	00000001	mod 010 r/m

LIDT = Load Interrupt Descriptor Table Register

00001111	00000001	mod 011 r/m

LLDT = Load Local Descriptor Table Register to Register/Memory

00001111	00000000	mod 0 1 0 r/m

LMSW = Load Machine Status Word from Register/Memory

00001111	00000001	mod 1 1 0 r/m

LSL = Load Segment Limit from Register/Memory

00001111	00000011	mod reg r/m

LTR = Load Task Register from Register/Memory

00001111	00000000	mod 0 0 1 r/m

SGDT = Store Global Descriptor Table Register

00001111	00000001	mod 0 0 0 r/m

SIDT = Store Interrupt Descriptor Table Register

00001111	00000001	mod 0 0 1 r/m

SLDT = Store Local Descriptor Table Register to Register/Memory

00001111	00000000	mod 0 0 0 r/m

SMSW = Store Machine Status Word

00001111	00000001	mod 1 0 0 r/m

STR = Store Task Register to Register/Memory

00001111	00000000	mod 0 0 1 r/m

VERR = Verify Read Access; Register/Memory

00001111	00000000	mod 100 r/m

VERW = Verify Write Access

00001111	00000000	mod 101 r/m

Introduction to the 80387 Instruction Set

The 80387 instructions use many of the same fields defined earlier in this section for the 80386 instructions. Additional fields used by the 80387 instructions are defined in the following figure.

Field	Description	Bit Information
escape	80386 Extension Escape	Bit Pattern = 11011
MF	Memory Format	00 = 32-bit Real 01 = 32-bit integer 10 = 64-bit Real 11 = 16-bit integer
ST(0)	Current Stack Top	
ST(i)	ith register below the stack top	
d	Destination	0 = Destination is ST(0) 1 = Destination is ST(i)
P	Pop	0 = No pop 1 = Pop ST(0)
R	Reverse*	0 = Destination (op) source 1 = Source (op) destination

* When d = 1, reverse the sense of R.

Figure 24. 80387 Encoding Field Summary

Within the 80387 Instruction Set:

- Temporary (Extended) Real is 80-bit Real.
- Long Integer is a 64-bit integer.

80387 Usage of the Scale-Index-Base Byte

The "mod r/m" byte of an 80387 instruction can be followed by a scale-index-base (s-i-b) byte having the same address mode definition as in the 80386 instruction. The mod field in the 80387 instruction is never equal to 11.

Instruction and Data Pointers

The parallel operation of the 80386 and 80387 may allow errors detected by the 80387 to be reported after the 80386 has executed the ESC instruction that caused the error. The 80386/80387 provides two pointer registers to identify the failing numeric instruction. The pointer registers supply the address of the failing numeric instruction and the address of its numeric memory operand when applicable.

Although the pointer registers are located in the 80386, they appear to be located in the 80387 because they are accessed by the ESC instructions FLDENV, FSTENV, FSAVE, and FRSTOR. Whenever the 80386 decodes a new ESC instruction, it saves the address of the instruction along with any prefix bytes that may be present, the address of the operand (if present), and the opcode.

The instruction and data pointers appear in one of four available formats:

- 16-bit Real Mode/Virtual 8086 Mode
- 32-bit Real Mode
- 16-bit Protected Mode
- 32-bit Protected Mode

The Real Mode formats are used whenever the 80386 is in the Real Mode or Virtual 8086 Mode. The Protected Mode formats are used when the 80386 is in the Protected Mode. The Operand Size Prefix can also be used with the 80387 instructions. The operand size of the 80387 instruction determines whether the 16-bit or 32-bit format is used.

Note: FSAVE and FRSTOR have an additional eight fields (10 bytes per field) that contain the current contents of ST(0) through ST(7). These fields follow the instruction and data pointer image shown in the following figures.

The following figures show the instruction and data pointer image format used in the various address modes. The ESC instructions FLDENV, FSTENV, FSAVE, and FRSTOR are used to transfer these values between the 80386/80387 registers and memory.

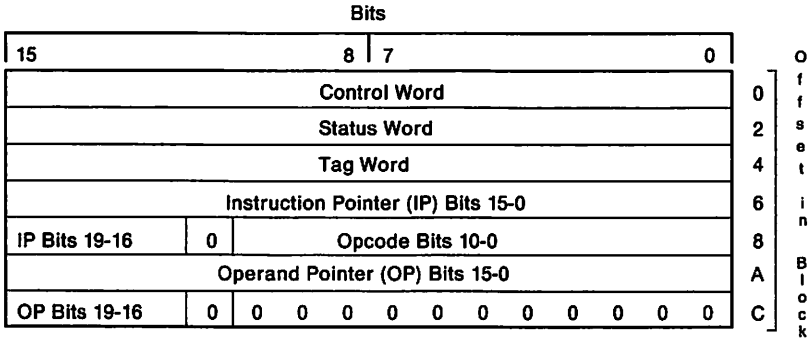


Figure 25. Instruction and Pointer Image (16-Bit Real Address Mode)

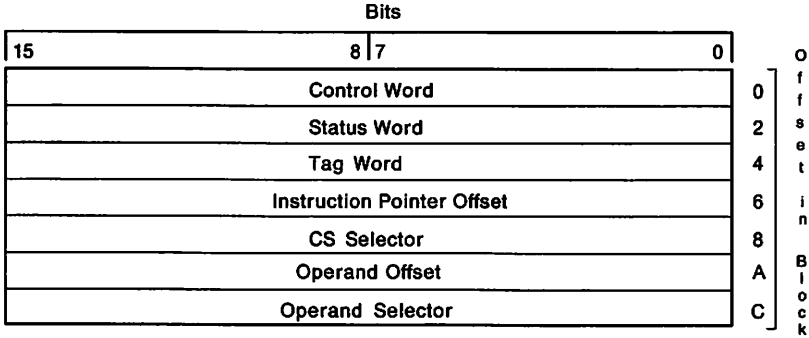


Figure 26. Instruction and Pointer Image (16-Bit Protected Mode)

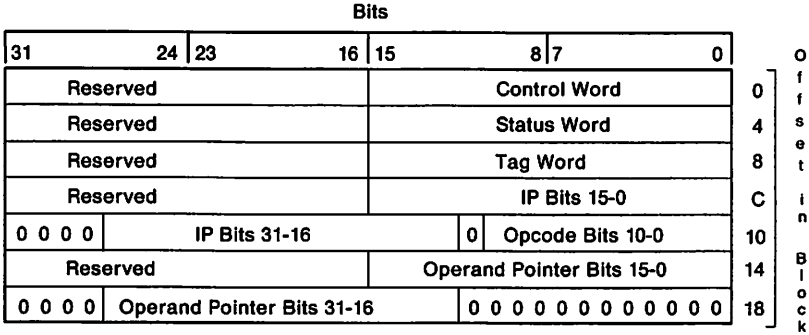


Figure 27. Instruction and Pointer Image (32-Bit Real Address Mode)

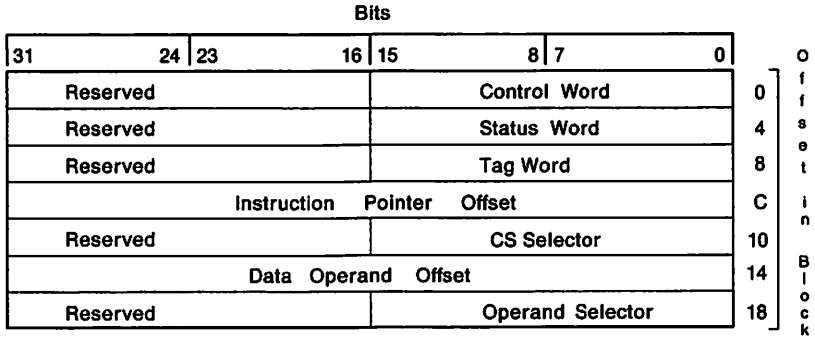


Figure 28. Instruction and Pointer Image (32-Bit Protected Mode)

New Instructions

Several new instructions are included in the 80387 instruction set that are not available to the 80287 or 8087 math coprocessors. The new instructions are:

FUCOM (Unordered Compare Real)
FUCOMP (Unordered Compare Real and Pop)
FUCOMPP (Unordered Compare Real and Pop Twice)
FPREM1 (IEEE Partial Remainder)
FSINE (Sine)
FCOS (Cosine)
FSINCOS (Sine and Cosine).

80387 Math Coprocessor Instruction Set

The following is an instruction set summary for the 80387 coprocessor. In the following, the bit pattern for escape is 11011.

Data Transfer

FLD = Load

Integer/Real Memory to ST(0)

escape MF 1	mod 0 0 0 r/m

Long Integer Memory to ST(0)

escape 1 1 1	mod 1 0 1 r/m

Temporary Real Memory to ST(0)

escape 0 1 1	mod 1 0 1 r/m

BCD Memory to ST(0)

escape 1 1 1	mod 1 0 0 r/m

ST(i) to ST(0)

escape 0 0 1	1 1 0 0 0 ST(i)

FST = Store

ST(0) to Integer/Real Memory

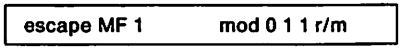
escape MF 1	mod 0 1 0 r/m

ST(0) to ST(i)

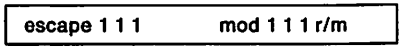
escape 1 0 1	1 1 0 1 0 ST(i)

FSTP = Store and Pop

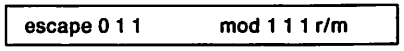
ST(0) to Integer/Real Memory



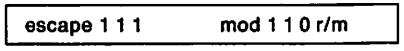
ST(0) to Long Integer Memory



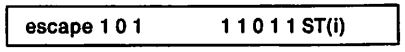
ST(0) to Temporary Real Memory



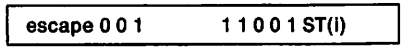
ST(0) to BCD Memory



ST(0) to ST(i)



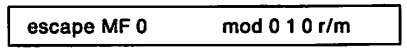
FXCH = Exchange ST(i) and ST(0)



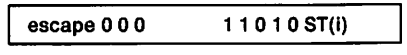
Comparison

FCOM = Compare

Integer/Real Memory to ST(0)



ST(i) to ST(0)



FCOMP = Compare and Pop

Integer/Real Memory to ST(0)

escape MF 0	mod 0 1 1 r/m

ST(i) to ST(0)

escape 0 0 0	1 1 0 1 1 ST(i)

FCOMPP = Compare ST(1) to ST(0) and Pop Twice

escape 1 1 0	1 1 0 1 1 0 0 1

FUCOM = Unordered Compare Real

escape 1 0 1	1 1 1 0 0 ST(i)

FUCOMP = Unordered Compare Real and Pop

escape 1 0 1	1 1 1 0 1 ST(i)

FUCOMPP = Unordered Compare Real and Pop Twice

escape 0 1 0	1 1 1 0 1 0 0 1

FTST = Test ST(0)

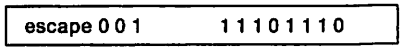
escape 0 0 1	1 1 1 0 0 1 0 0

FXAM = Examine ST(0)

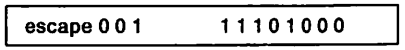
escape 0 0 1	1 1 1 0 0 1 0 1

Constants

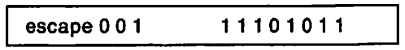
FLDZ = Load +0.0 into ST(0)



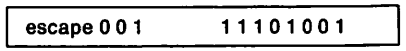
FLD1 = Load +1.0 into ST(0)



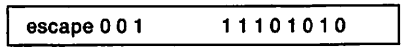
FLDPI = Load π into ST(0)



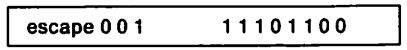
FLDL2T = Load $\log_2 10$ into ST(0)



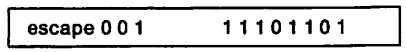
FLDL2E = Load $\log_2 e$ into ST(0)



FLDLG2 = Load $\log_{10} 2$ into ST(0)



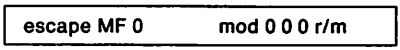
FLDLN2 = Load $\log_e 2$ into ST(0)



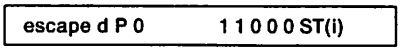
Arithmetic

FADD = Addition

Integer/Real Memory with ST(0)

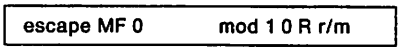


ST(i) and ST(0)

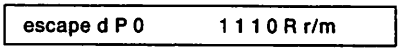


FSUB = Subtraction

Integer/Real Memory with ST(0)

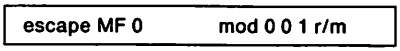


ST(i) and ST(0)

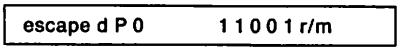


FMUL = Multiplication

Integer/Real Memory with ST(0)

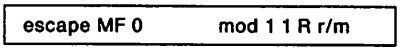


ST(i) and ST(0)

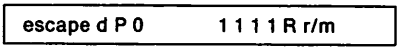


FDIV = Division

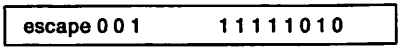
Integer/Real Memory with ST(0)



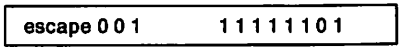
ST(i) and ST(0)



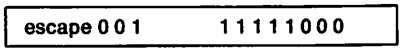
FSQRT = Square Root of ST(0)



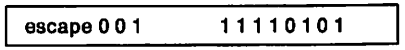
FSCALE = Scale ST(0) by ST(1)



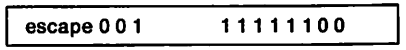
FPREM = Partial Remainder of ST(0) ÷ ST(1)



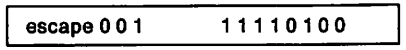
FPREM1 = IEEE Partial Remainder



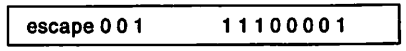
FRNDINT = Round ST(0) to Integer



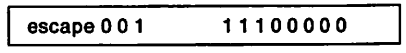
FXTRACT = Extract Components of ST(0)



FABS = Absolute Value of ST(0)

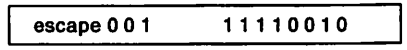


FCHS = Change Sign of ST(0)

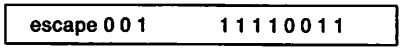


Transcendental

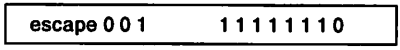
FPTAN = Partial Tangent of ST(0)



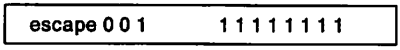
FPATAN = Partial Arctangent of ST(1) ÷ ST(0)



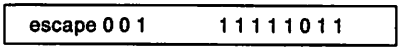
FSIN = Sine



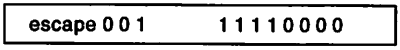
FCOS = Cosine



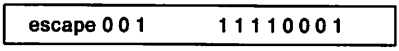
FSINCOS = Sine and Cosine



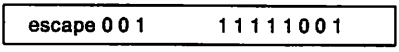
F2XM1 = $2^{ST(0)}$ -1



FYL2X = ST(1) x Log_2 [ST(0)]

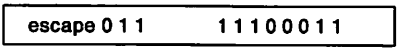


FYL2XP1 = ST(1) x Log_2 [ST(0) + 1]

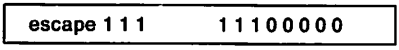


Processor Control

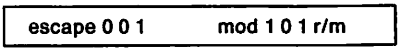
FINIT = Initialize NPX



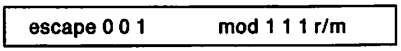
FSTSW AX = Store Control Word



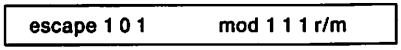
FLDCW = Load Control Word



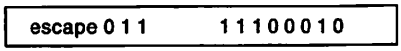
FSTCW = Store Control Word



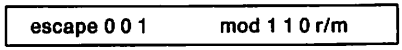
FSTSW = Store Status Word



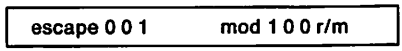
FCLEX = Clear Exceptions



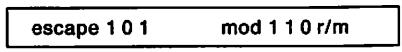
FSTENV = Store Environment



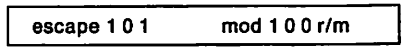
FLDENV = Load Environment



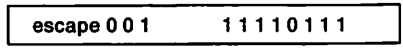
FSAVE = Save State



FRSTOR = Restore State



FINCSTP = Increment Stack Pointer



FDECSTP = Decrement Stack Pointer

escape 0 0 1	1 1 1 1 0 1 1 0

FFREE = Free ST(I)

escape 1 0 1	1 1 0 0 0 ST(i)

FNOP = No Operation

escape 0 0 1	1 1 0 1 0 0 0 0

Index

A

AAA instruction 20, 63
AAD instruction 21, 64
AAM instruction 21, 64
AAS instruction 20, 63
ADC instruction 18, 60
ADD instruction 18, 59
address mode 47
address size prefix 82
address space 1, 2, 5, 6
AND instruction 22, 66
arithmetic instructions 18, 39, 59, 94
ARPL instruction 33, 83

B

bit manipulation instructions 71
BOUND instruction 30, 81
BSF instruction 71
BT instruction 71
BTC instruction 71
BTR instruction 71
BTS instruction 72

C

CALL instruction 26, 72
CBW instruction 21, 64
CLC instruction 30, 58
CLD instruction 31, 58
CLI instruction 31, 58
clock generator 3
CLTS instruction 58
CMC instruction 30
CMP instruction 20, 62
CMPS instruction 24, 68
comparison instructions 37, 91
conditional byte set 78
conditional jumps 73
conditional test field 52
constants 38, 93
control register field 53
control transfer instructions 26, 72
CTS instruction 31
CWD instruction 21, 64

D

DAA instruction 20, 63
DAS instruction 20, 63
data pointers 87
data transfer 35
data transfer instructions 14, 35, 54, 90
data types 2, 3, 12
data types, math coprocessor 3
debug register field 53
DEC instruction 19, 62
descriptors 6, 43
DIV instruction 21, 64

E

effective address 47
effective address size prefix 44
encoding field summary 46, 86
encoding, instructions 46
ENTER instruction 29, 80
ESC instruction 31, 82
exception 1, 5
exception conditions 4

F

FABS instruction 40, 95
FADD instruction 39, 94
FCHS instruction 40, 95
FCLEX instruction 42, 97
FCOM instruction 37, 91
FCOMP instruction 37, 92
FCOMPP instruction 37, 92
FCOS instruction 96
FDECSTP instruction 42, 98
FDIV instruction 39, 94
FFREE instruction 42, 98
FINCSTP instruction 42, 97
FINIT instruction 96
FINT instruction 41
flag control instructions 58
FLD instruction 35, 90
FLDCW instruction 41, 97
FLDENV instruction 42, 97
FLDLG2 instruction 38, 93
FLDLN2 instruction 38, 93
FLDL2E instruction 38, 93
FLDL2T instruction 38, 93
FLDPI instruction 38, 93
FLDZ instruction 38, 93
FLD1 instruction 38, 93
FMUL instruction 39, 94
FNOP instruction 42, 98
FPATAN instruction 40, 96
FPREM instruction 40, 95
FPREM1 instruction 95
FPTAN instruction 40, 95
FRNDINT instruction 40, 95
FRSTOR instruction 42, 97
FSAVE instruction 42, 97
FSCALE instruction 40, 95
FSETPM instruction 41
FSIN instruction 96
FSINCOS instruction 96
FSQRT instruction 40, 95
FST instruction 36, 90
FSTCW instruction 41, 97
FSTENV instruction 42, 97
FSTP instruction 36, 91
FSTSW AX instruction 41, 96
FSTSW instruction 41, 97
FSUB instruction 39, 94
FTST instruction 38, 92
FUCOM instruction 92
FUCOMP instruction 92
FUCOMPP instruction 92
FXAM instruction 38, 92
FXCH instruction 37, 91
FXTRACT instruction 40, 95
FYL2X instruction 41, 96
FYL2XP1 instruction 41, 96
F2XM1 instruction 41, 96

G

general register field 51
global descriptor table 6

H

hardware interface 3, 12
HLT instruction 31, 81

I

I/O ports 3
IDIV instruction 21, 64
IIMUL instruction 21
IMUL instruction 21, 63
IN instruction 16, 56
INC instruction 18, 60
INS instruction 24, 68
instruction pointers 87
instruction set, math coprocessor 86
INT instruction 30, 80
interrupt 1, 5
interrupt instructions 80
interrupts 4
INTO instruction 30, 80

IRET instruction 30, 81

J

JB/JNAE instruction 27, 74
JBE/JNA instruction 28, 75
JCXZ instruction 29, 77
JE/JZ instruction 27, 74
JECXZ instruction 77
JL/JNGE instruction 27, 76
JLE/JNG instruction 27, 77
JMP instruction 26, 72
JNB/JAE instruction 28, 74
JNBE/JA instruction 28, 75
JNE/JNZ instruction 28, 75
JNL/JGE instruction 28, 77
JNLE/JG instruction 28, 77
JNO instruction 29, 74
JNP/JPO instruction 29, 76
JNS instruction 29, 76
JO instruction 28, 73
JP/JPE instruction 28, 76
JS instruction 28, 75

L

LAHF instruction 17, 58
LAR instruction 33, 83
LDS instruction 17, 57
LEA instruction 17, 57
LEAVE instruction 29, 80
LES instruction 17, 57
LFS instruction 57
LGDT instruction 32, 83
LGS instruction 57
LIDT instruction 32, 83
LLDT instruction 32, 84
LMSW instruction 32, 84
local descriptor table 6
LOCK instruction 31, 82
LODS instruction 24, 69
logic instructions 22, 64
LOOP instruction 29, 78
LOOPNZ/LOOPNE instruction 29, 78
LOOPZ/LOOPE instruction 29, 78
LSL instruction 33, 84
LSS instruction 57
LTR instruction 32, 84

M

memory, virtual 2
mode, protected 2
mode, real address 5
mode, real-address 1
MOV instruction 14, 54, 81
MOVS instruction 24, 69
MOVSX instruction 55
MOVZX instruction 55
MUL instruction 21, 63

N

NEG instruction 20, 63
NOP instruction 82
NOT instruction 24, 68

O

operand length field 50
operand size 44
operand size prefix 44, 82
operation direction field 52
OR instruction 23, 67
OUT instruction 16, 57
OUTS instruction 24, 69

P

page directory 8
page directory physical base address register 9
page fault linear address register 8
page frame 8
page tables 8

paging 6
paging mechanism, 80386 8
pointer 1, 2, 5
pointer image 87
pointer registers 87
POP instruction 15, 56
POPA instruction 16, 56
POPF instruction 17, 58
prefix bytes 82
processor control instructions 30, 81, 96
processor extension 82
programming interface 3
protected mode 2, 6, 11
protected virtual address mode 2, 6
protection control instructions 32, 83
PUSH instruction 15, 55
PUSHA instruction 15, 55
PUSHF instruction 17, 58

R

real address mode 5, 11
real-address mode 1
registers 3
REP INS instruction 70
REP LODS instruction 70
REP MOVS instruction 70
REP OUTS instruction 70
REP STOS instruction 70
REP/REPNE, REPZ/REPNZ instructions 25
REPE CMPS instruction 69
REPE SCAS instruction 70
repeated string manipulation instructions 69
REPNE CMPS instruction 70
REPNE SCAS instruction 70
RET instruction 27, 73
rotate instructions 22, 64

S

SAHF instruction 17, 59
SBB instruction 19, 61
scale-index-base byte 47
SCAS instruction 24, 69
segment control instructions 57
segment descriptors 43
segment override prefix 82
segment register field 51
segments 1, 5
SETB/SETNAE instruction 78
SETBE/SETNA instruction 79
SETE/SETZ instruction 78
SETL/SETNGE instruction 79
SETLE/SETNG instruction 80
SETNB instruction 78
SETNBE/SETA instruction 79
SETNE/SETNZ instruction 79
SETNL/SETGE instruction 80
SETNLE/SETG instruction 80
SETNO instruction 78
SETNP/SETPO instruction 79
SETNS instruction 79
SETO instruction 78
SETP/SETPE instruction 79
SETS instruction 79
SGDT instruction 32, 84
shift instructions 22, 64
SHLD instruction 65
SHRD instruction 66
SIDT instruction 32, 84
sign extend field 52
SLDT instruction 32, 84
SMSW instruction 33, 84
STC instruction 30, 59
STD instruction 31, 59
STI instruction 31, 59
STOS instruction 24, 69
STR instruction 32, 84
string manipulation instructions 24, 68
SUB instruction 19, 61

T

TEST instruction 23, 67
test register field 53
transcendental instructions 95
two-level paging 8

V

VERR instruction 33, 85
VERW instruction 33, 85
virtual memory 2, 6
virtual 8086 flag 7
virtual 8086 mode 7, 11

W

WAIT instruction 31, 82

X

XCHG instruction 16, 56
XLAT instruction 16
XOR instruction 23, 68

Numerics

80286 microprocessor 1
80287 math coprocessor 2
80386 microprocessor 5
80386 paging mechanism 8
80387 clock generator 12
80387 coprocessor 86
80387 exception conditions 12
80387 I/O ports 12
80387 math coprocessor 10

Notes:

Direct Memory Access Controller (Type 1)

Description 1
DMA Controller Operations 2
  Data Transfers between Memory and I/O Devices 2
  Read Verifications 2
DMA I/O Address Map 3
Byte Pointer 4
DMA Registers 4
  Memory Address Register 4
  I/O Address Register 5
  Transfer Count Register 5
  Temporary Holding Register 5
  Mask Register 6
  Mode Register 7
  Extended Mode Register 7
  Status Register 8
  DMA Extended Function Register (Hex 0018) 8
  Arbus Register 9
DMA Extended Operations 9
  Extended Commands 10

© Copyright IBM Corp. 1990

Figures

1. DMA I/O Address Map 3
2. DMA Registers 4
3. Set/Clear Single Mask Bit Using 8237 Compatible Mode . . . 6
4. DMA Mask Register Write Using 8237 Compatible Mode . . . 6
5. 8237 Compatible Mode Register 7
6. Extended Mode Register 8
7. Status Register 8
8. DMA Extended Function Register (Hex 0018) 8
9. Arbus Register 9
10. DMA Extended Address Decode 9
11. DMA Extended Commands 10
12. DMA Channel 2 Programming Example, Extended Commands 11

© Copyright IBM Corp. 1990

Description

The Direct Memory Access (DMA) controller allows I/O devices to transfer data directly to and from memory. This frees the system microprocessor of I/O tasks, resulting in a higher throughput.

The DMA controller is software programmable. The system microprocessor can address the DMA controller and read or modify the internal registers to define the various DMA modes, transfer addresses, transfer counts, channel masks, and page registers.

The functions of the DMA controller can be grouped into two categories: program mode and DMA transfer mode.

In the program mode, the system microprocessor accesses the DMA controller within the specific address range. These addresses are identified in Figure 1 on page 3. In this mode, the DMA registers can be read from or written to.

In the DMA mode, the DMA controller performs the data transfer. The transfer is initiated when a DMA slave has won the arbitration bus and the DMA controller has been programmed to service the winning request in process. Data transfers can be a single-byte transfer, or multiple-byte transfers (burst).

Deactivation of CD CHRDY by a device can extend accesses for slower I/O or memory devices.

The DMA controller supports the following:

- Register and program compatibility with the IBM Personal Computer AT® DMA channels (8237 compatible mode)
- 16MB (MB equals 1,048,576 bytes) 24-bit address capability for memory and 64KB (KB equals 1024 bytes) 16-bit address capability for I/O
- Eight independent DMA channels capable of transferring data between memory and I/O devices
- DMA operation with a separate read and write cycle for each transfer operation
- Channel programmable for byte or word transfer
- Extended operations:

Personal Computer AT is a registered trademark of the International Business Machines Corporation.

  - Extended program control
  - Extended Mode register
- 8- and 16-bit DMA slaves only
- Programmable arbitration levels for two channels.

DMA Controller Operations

The DMA controller does two types of operations:

- Data transfers between memory and I/O devices
- Read verifications.

Data Transfers between Memory and I/O Devices

The DMA controller performs serial transfers for all read and write operations. These transfers can be between memory and I/O on any channel. Data is read from a device and latched in the DMA controller before it is written back to a second device. The memory address needs to be specified only for a DMA data transfer. A programmable 16-bit I/O address can be provided during the I/O portion of the transfer as a programmable option. If the programmable 16-bit I/O address is not selected, the I/O address is forced to hex 0000 during the I/O transfer.

Read Verifications

The DMA controller can do a memory-read operation without a transfer. The address and the count are updated, and the terminal count is provided.

DMA I/O Address Map

Address (Hex)	Description	Bit Description	Byte Pointer
0000	Channel 0, Memory Address Register	00-15	Used
0001	Channel 0, Transfer Count Register	00-15	Used
0002	Channel 1, Memory Address Register	00-15	Used
0003	Channel 1, Transfer Count Register	00-15	Used
0004	Channel 2, Memory Address Register	00-15	Used
0005	Channel 2, Transfer Count Register	00-15	Used
0006	Channel 3, Memory Address Register	00-15	Used
0007	Channel 3, Transfer Count Register	00-15	Used
0008	Channel 0-3, Status Register	00-07	
000A	Channel 0-3, Mask Register (Set/Reset)	00-02	
000B	Channel 0-3, Mode Register (Write)	00-07	
000C	Clear Byte Pointer (Write)	N/A	
000D	DMA Controller Reset (Write)	N/A	
000E	Channel 0-3, Clear Mask Register (Write)	N/A	
000F	Channel 0-3, Write Mask Register	00-03	
0018	Extended Function Register (Write)	00-07	
001A	Extended Function Execute	00-07	Used *
0081	Channel 2, Page Table Address Register **	00-07	
0082	Channel 3, Page Table Address Register **	00-07	
0083	Channel 1, Page Table Address Register **	00-07	
0087	Channel 0, Page Table Address Register **	00-07	
0089	Channel 6, Page Table Address Register **	00-07	
008A	Channel 7, Page Table Address Register **	00-07	
008B	Channel 5, Page Table Address Register **	00-07	
008F	Channel 4, Page Table Address Register **	00-07	
00C0	Channel 4, Memory Address Register	00-15	Used
00C2	Channel 4, Transfer Count Register	00-15	Used
00C4	Channel 5, Memory Address Register	00-15	Used
00C6	Channel 5, Transfer Count Register	00-15	Used
00C8	Channel 6, Memory Address Register	00-15	Used
00CA	Channel 6, Transfer Count Register	00-15	Used
00CC	Channel 7, Memory Address Register	00-15	Used
00CE	Channel 7, Transfer Count Register	00-15	Used
00D0	Channel 4-7, Status Register	00-07	
00D4	Channel 4-7, Mask Register (Set/Reset)	00-02	
00D6	Channel 4-7, Mode Register (Write)	00-07	
00D8	Clear Byte Pointer (Write)	N/A	
00DA	DMA Controller Reset (Write)	N/A	
00DC	Channel 4-7, Clear Mask Register (Write)	N/A	
00DE	Channel 4-7, Write Mask Register	00-03	

Note: * Used only during extended functions, see "Extended Commands" on page 10. ** Upper byte of memory address register.

Figure 1. DMA I/O Address Map

Byte Pointer

A byte pointer gives 8-bit ports access to consecutive bytes of registers greater than 8 bits. For program I/O, the registers that use it are the Memory Address registers (3 bytes), the Transfer Count registers (2 bytes), and the I/O Address registers (2 bytes). Interrupts should be masked off while programming DMA controller operations.

DMA Registers

All system microprocessor access to the DMA controller must be 8-bit I/O instructions. The following figure lists the names and sizes of the DMA registers.

Register	Size (Bits)	Quantity of Registers	Allocation
Memory Address	24	8	1 per Channel
I/O Address	16	8	1 per Channel
Transfer Count	16	8	1 per Channel
Temporary Holding	16	1	All Channels
Mask	4	2	1 for Channels 7 - 4 1 for Channels 3 - 0
Arbus	4	2	1 for Channel 4 1 for Channel 0
Mode	8	8	1 per Channel
Status	8	2	1 for Channels 7 - 4 1 for Channels 3 - 0
Function	8	1	All Channels
Refresh	9	1	Independent of DMA

Figure 2. DMA Registers

Memory Address Register

Each channel has a 24-bit Memory Address register, which is loaded by the system microprocessor. The Mode register determines whether the address is incremented or decremented. The Mode register can be read by the system microprocessor in successive I/O byte operations. To read this register, the microprocessor must use the extended DMA commands.

I/O Address Register

Each channel has a 16-bit I/O Address register, which is loaded by the system microprocessor. The bits in this register do not change during DMA transfers. This register can be read by the system microprocessor in successive I/O byte operations. To read this register, the microprocessor must use the extended DMA commands.

Typically, a DMA slave is selected for DMA transfers by a decode of the arbitration level, status (-S0 exclusively ORed with -S1), and M/-IO. In this case, the respective I/O address register must have a value of 0.

A DMA slave can be selected based on a decode of the address rather than the arbitration level. In this case, the respective I/O address register must have the proper I/O address value.

Transfer Count Register

Each channel has a 16-bit Transfer Count register, which is loaded by the system microprocessor. The transfer count determines how many transfers the DMA channel will execute before reaching the terminal count. The number of transfers is always 1 more than the count specifies. If the count is 0, the DMA controller does one transfer. This register can be read by the system microprocessor in successive I/O byte operations. To read this register, the system microprocessor can use only the extended DMA commands.

Temporary Holding Register

This 16-bit register holds the intermediate value for the serial DMA transfer taking place. A DMA operation requires the data to be held in the register before it is written back. This register is not accessible by the system microprocessor.

Mask Register

Bit	Function
7 - 3	Reserved = 0
2	0 Clear Mask Bit 1 Set Mask Bit
1, 0	00 Select Channel 0 or 4 01 Select Channel 1 or 5 10 Select Channel 2 or 6 11 Select Channel 3 or 7

Figure 3. Set/Clear Single Mask Bit Using 8237 Compatible Mode

Bit	Function
7 - 4	Reserved = 0
3	0 Clear Channel 3 or 7 Mask Bit 1 Set Channel 3 or 7 Mask Bit
2	0 Clear Channel 2 or 6 Mask Bit 1 Set Channel 2 or 6 Mask Bit
1	0 Clear Channel 1 or 5 Mask Bit 1 Set Channel 1 or 5 Mask Bit
0	0 Clear Channel 0 or 4 Mask Bit 1 Set Channel 0 or 4 Mask Bit

Figure 4. DMA Mask Register Write Using 8237 Compatible Mode

Each channel has a corresponding mask bit that, when set, disables the DMA from servicing the requesting device. Each mask bit can be set to 0 or 1 by the system microprocessor. A system reset or DMA Controller Reset command sets all mask bits to 1. A Clear Mask Register command sets mask bits 0 - 3 or mask bits 4 - 7 to 0.

When a device requesting DMA cycles wins the arbitration cycle, and the mask bit is set to 1 on the corresponding channel, the DMA controller does not execute any cycles in its behalf and allows external devices to provide the transfer. If no device responds, the bus times out and causes a nonmaskable interrupt (NMI). This register can be programmed using the 8237 compatible mode commands (used by the IBM Personal Computer AT) or the extended DMA commands.

Mode Register

The Mode register for each channel identifies the type of operation that takes place when that channel transfers data.

Bit	Function
7, 6	Reserved = 0
5, 4	Reserved = 0
3, 2	00 Verify Operation 01 Write Operation 10 Read Operation 11 Reserved
1, 0	Channel Accessed 00 Select Channel 0 or 4 01 Select Channel 1 or 5 10 Select Channel 2 or 6 11 Select Channel 3 or 7

Figure 5. 8237 Compatible Mode Register

The Mode register is programmed by the system microprocessor, and its contents are reformatted and stored internally in the DMA controller. In the 8237 compatible mode, this register can only be written.

Extended Mode Register

Besides the 8237 compatible mode, all channels support an 8-bit Extended Mode register. The Extended Mode register can be programmed and read by the system microprocessor.

The DMA controller supports an Extended Mode register for each channel that can be programmed and read by the system microprocessor. This register is used whenever a DMA channel requests a DMA data transfer.

The DMA channel must be programmed to match the transfer size of the DMA slave on the channel. Bit 6 of this register is used to program the size of the DMA transfer.

Bit	Function
7	Reserved = 0
6	0 = 8-Bit Transfer 1 = 16-Bit Transfer
5	Reserved = 0
4	Reserved = 0
3	0 = Read Memory Transfer 1 = Write Memory Transfer
2	0 = Read Verifications Operation 1 = Data Transfer Operation
1	Reserved = 0
0	0 = I/O Address equals 0000H 1 = Use programmed I/O Address

Figure 6. Extended Mode Register

Status Register

The Status register, which can be read by the system microprocessor, contains information about the status of the devices. This information tells which channels have reached the terminal count and which channels have requested the bus since the last time the register was read.

Bit	Function
7	Channel 3 or 7 Request
6	Channel 2 or 6 Request
5	Channel 1 or 5 Request
4	Channel 0 or 4 Request
3	TC on Channel 3 or 7
2	TC on Channel 2 or 6
1	TC on Channel 1 or 5
0	TC on Channel 0 or 4

Figure 7. Status Register

Bits 3 through 0 in each Status register are set every time a terminal count is reached by a corresponding channel. Bits 7 through 4 are set when a corresponding arbitration level has controlled the bus. All bits are cleared by a system reset or following a system microprocessor Status Read command. This register can be read using the 8237 commands or extended DMA commands.

DMA Extended Function Register (Hex 0018)

This 8-bit register minimizes I/O address requirements and provides the extended program functions. The system microprocessor loads this register using I/O write operations. See "Extended Commands" on page 10 for more information.

Bit	Function
7 - 4	Program Command (DMA Extended Commands)
3	Reserved = 0
2 - 0	Channel Number (0 through 7)

Figure 8. DMA Extended Function Register (Hex 0018)

Arbus Register

This register is used for virtual DMA operations.

Bit	Function
7 - 4	Reserved
3 - 0	Arbitration Level

Figure 9. Arbus Register

Virtual DMA channel operation permits programming of the arbitration level assignment for channels 0 and 4 using the two 4-bit Arbus registers. These registers enable the system microprocessor to dynamically reassign the arbitration ID value by which the DMA controller responds to bus arbitration for DMA requests. This allows channels 0 and 4 to service devices at any arbitration level. The value of arbitration level hex F is reserved.

DMA Extended Operations

The function register supports an extended set of commands for the DMA channels. The extended command hex 8 programs the Arbus registers; the upper 4 bits of the Extended Function register are set to a value of 8 to select the Arbus register, and the lower 4 bits are set to the channel number (0 or 4). If channel 0 = 1 - 3 or 5 - 7 then channel 0 is active; if channel 0 = 4 then channel 0 is inactive. The system microprocessor uses the following addresses to gain control of the internal DMA registers.

I/O Address (Hex)	Command
0018	Write Extended Function Register
0019	Reserved
001A	Execute Extended Function Register
001B	Reserved

Figure 10. DMA Extended Address Decode

The system microprocessor uses the following steps to write to or read from any of the DMA internal registers:

1. Write to the Extended Function register by executing an I/O Write instruction to address hex 0018, with the proper data to indicate the function and the channel number. The internal byte pointer is always reset to 0 when an I/O write to address hex 0018 is detected.
2. Execute the Extended Function command by doing an I/O Read or I/O Write instruction to address hex 001A. The byte pointer automatically increments and points to the next byte each time port address hex 001A is used. This step is not required for Direct commands because they are executed when the Out command to address hex 0018 is detected.

Extended Commands

The following figure shows the available extended command set contained in the Extended Function register.

Registers/Bits Accessed	Bits	Extended Command (Hex) (7-4 *)	Byte Pointer
I/O Address Register	00-15	0	Used
Reserved		1	
Memory Address Register Write	00-23	2	Used
Memory Address Register Read	00-23	3	Used
Transfer Count Register Write	00-15	4	Used
Transfer Count Register Read	00-15	5	Used
Status Register Read	00-07	6	
Mode Register	00-07	7	
Arbus Register	00-07	8	
Mask Register Set Single Bit **		9	
Mask Register Reset Single Bit **		A	
Reserved		B	
Reserved		C	
Master Clear **		D	
Reserved		E	
Reserved		F	

Note: * Bits 7-4 of the Extended Function Register. ** Direct commands to the Extended Function register

Figure 11. DMA Extended Commands

The following is an example showing the programming of DMA channel 2 using the 8237 compatible mode and the extended mode. In this example, to perform each step, write the data indicated to the corresponding addresses.

Program Step	8237 Compatible Mode Address/Data	Extended Mode Address/Data
Set Channel Mask Bit	(000AH) x6H	(0018H) 92H
Clear Byte Pointer	(000CH) xxH	(0018H) 22H
Write Memory Address	(0004H) xxH	(001AH) xxH
Write Page Table Address	(0081H) xxH	(001AH) xxH
Clear Byte Pointer	(000CH) xxH	(0018H) 42H
Write Register Count	(0005H) xxH	(001AH) xxH
Write Register Count	(0005H) xxH	(001AH) xxH
Write Mode Register	(000BH) xxH	(0018H) 72H (001AH) xxH
Clear Channel 2 Mask Bit	(000AH) x2H	(0018H) A2H

Note: x's represent data.

Figure 12. DMA Channel 2 Programming Example, Extended Commands

Interrupt Controller (Type 1)

Description 1
Interrupt Assignments 1
Interrupt Sharing 2
Interrupt Controller Registers 2
Interrupt Request Register and In-Service Register 3
Interrupt Mask Register 3
Initialization Command Registers and Operation Command Registers 3
Modes of Operation 4
- Fully-Nested Mode 4
- Special Fully-Nested Mode 5
- Automatic Rotation Mode 6
- Specific Rotation Mode 6
- Special Mask Mode 7
- Poll Mode 7
- Level-Sensitive Mode 8

Programming the Interrupt Controller 8
- Initialization Command Byte 1 9
- Initialization Command Byte 2 9
- Initialization Command Byte 3 10
- Initialization Command Byte 4 10
- Operation Command Byte 1 11
- Operation Command Byte 2 11
- Operation Command Byte 3 12

Figures

1. Interrupt Level Assignments by Priority 1
2. Automatic Rotation Mode 6
3. Specific Rotation Mode when IRQ5 Has the Lowest Priority . 7
4. Poll Mode Status Byte 8
5. Initialization Command Byte 1 9
6. Initialization Command Byte 2 9
7. Initialization Command Byte 3 10
8. Initialization Command Byte 4 10
9. Operation Command Byte 1 11
10. Operation Command Byte 2 11
11. Operation Command Byte 2 (Bits 7 - 5) 12
12. Operation Command Byte 2 (Bits 2 - 0) 12
13. Operation Command Byte 3 12
14. Operation Command Byte 3 (Bits 6 and 5) 13
15. Operation Command Byte 3 (Bits 1 and 0) 13

Description

The system provides 16 levels of hardware interrupts. Any interrupt can be masked, including the nonmaskable interrupt. The interrupt controller must be initialized to the level-sensitive mode; the edge-triggered mode is not supported. Attempts to set the controller to the edge-triggered mode will result in level-sensitive operation. For more information on nonmaskable interrupt, see the system-specific technical references.

Interrupt Assignments

The following figure shows the interrupt assignments, interrupt levels, and their functions. The interrupt levels are listed by order of priority, from highest (NMI) to lowest (IRQ 7). See system-specific technical references for masking interrupts.

Level	Master Function	Level	Slave Function
NMI	Channel Check *		
IRQ 0	Timer		
IRQ 1	Keyboard		
IRQ 2	Cascade Interrupt Control —	IRQ 8	Real Time Clock
		IRQ 9	Redirect Cascade
		IRQ 10	Reserved
		IRQ 11	Reserved
		IRQ 12	Auxiliary Device
		IRQ 13	Math Coprocessor Exception
		IRQ 14	Fixed Disk
		IRQ 15	Reserved
IRQ 3	Serial Alternate		
IRQ 4	Serial Primary		
IRQ 5	Reserved		
IRQ 6	Diskette		
IRQ 7	Parallel Port		

IRQ 8 through 15 are cascaded through IRQ 2
* For Channel Check and other System Specific Functions, Refer to the System-Specific Technical References.

Figure 1. Interrupt Level Assignments by Priority

Interrupt Sharing

Hardware interrupt IRQ9 is defined as the replacement interrupt level for the cascade level IRQ2. Program interrupt sharing should be implemented on IRQ2, interrupt hex 0A. The following processing occurs to maintain compatibility with the IRQ2 used by IBM Personal Computer products:

1. A device drives the interrupt request active on IRQ2 of the channel.
2. This interrupt request is mapped in hardware to IRQ9 input on the slave interrupt controller.
3. When the interrupt occurs, the system microprocessor passes control to the IRQ9 (interrupt hex 71) interrupt handler.
4. The interrupt handler performs an end of interrupt (EOI) to the slave interrupt controller and passes control to the IRQ2 (interrupt hex 0A) interrupt handler.
5. The IRQ2 interrupt handler, when handling the interrupt, causes the device to reset the interrupt request prior to performing an EOI to the master interrupt controller that finishes servicing the IRQ2 request.

Note: Prior to the programming of the interrupt controllers, interrupts should be disabled with a CLI instruction. This includes the Mask register, EOIs, initialization command bytes, and operation command bytes.

Interrupt Controller Registers

The interrupt controller contains the following registers:

- Interrupt Request register
- In-Service register
- Interrupt Mask register
- Initialization Command registers
- Operation Command registers.

Interrupt Request Register and In-Service Register

These registers handle incoming interrupt requests. The Interrupt Request register stores all interrupt levels requesting service. The In-Service register stores all interrupt levels currently being serviced. A priority resolver prioritizes the bits in the Interrupt Request register and strobes the bit with the highest priority into the corresponding bit of the In-Service register.

Both registers can be read by issuing a Read Register command through Operation Command Byte 3, and then reading port hex 0020 or 00A0. The controller keeps track of the last register selected; therefore, subsequent reads of the same register do not require another Operation Command Byte 3 to be written. See "Operation Command Byte 3" on page 12 for more information.

Note: After initialization, the controller is set to read the Interrupt Request register.

Interrupt Mask Register

The Interrupt Mask register contains bits that mask each of the interrupt request lines of the Interrupt Request register. Lower priority levels are not affected when a higher priority level is masked.

The contents of this register are placed on the output data bus when -READ is active and port hex 0021 or 00A1 is accessed.

Initialization Command Registers and Operation Command Registers

These registers store commands from the system microprocessor that define initialization parameters and operating modes. See "Programming the Interrupt Controller" on page 8 for more information.

Modes of Operation

The interrupt controller can be programmed to operate in a variety of modes through the Initialization Command bytes and the Operation Command bytes.

Fully-Nested Mode

In the fully-nested mode, interrupts are prioritized from 0 (highest priority) to 7 (lowest priority). This mode is automatically entered after initialization unless another mode has been defined.

Note: The priorities can be changed by rotating the priorities through Operation Command Byte 2.

A typical interrupt request occurs in the following manner:

1. One or more 'interrupt request' lines are set active causing the corresponding bits in the Interrupt Request register to be set to 1.
2. The interrupt controller evaluates the requests and sends an interrupt to the system microprocessor, if appropriate.
3. The system microprocessor responds with an 'interrupt acknowledge' pulse to the interrupt controller.
4. The controller prioritizes the unmasked bits in the Interrupt Request register and strobes the bit with the highest priority into the corresponding bit of the In-Service register. No data is sent to the system microprocessor.

   Note: If an interrupt request is not present (for example, the duration of the request was too short), the interrupt controller issues an interrupt 7.
5. The system microprocessor sends a second 'interrupt acknowledge' pulse to the interrupt controller.
6. The interrupt controller responds by releasing the interrupt vector on the data bus, where it is read by the system microprocessor.
7. The highest priority in-service bit remains set to 1 until the proper End of Interrupt command is issued by the interrupt subroutine. If the source of the interrupt request is the slave interrupt controller, the End of Interrupt command must be issued twice, once for the master and once for the slave. When the in-service

bit is set to 1, all other interrupts with the same or lower priority are inhibited; interrupts with a higher priority cause an interrupt, but the interrupt is acknowledged only if the system-microprocessor interrupt input has been re-enabled by software.

The End of Interrupt command has two forms, specific and nonspecific. The controller responds to a nonspecific End of Interrupt command by resetting the highest in-service bit of those set. In a mode that uses a fully-nested interrupt structure, the highest in-service bit set is the level that was just acknowledged and serviced. In a mode that can use other than the fully-nested interrupt structure, a specific End of Interrupt command is required to define which in-service bit to reset.

Note: An in-service bit masked by an Interrupt Mask register bit cannot be reset by a nonspecific End of Interrupt command when in the special mask mode. See "Special Mask Mode" on page 7 for more information.

Special Fully-Nested Mode

The special fully-nested mode is used when the priority in the slave interrupt controller must be preserved. This mode is similar to the normal nested mode with the following exceptions:

- When the slave's interrupt request is in service, the slave can still generate additional interrupt requests of a higher priority that are recognized by the master, and initiate interrupts to the system microprocessor.
- Upon completion of the interrupt service routine, software must send a nonspecific End of Interrupt command to the slave and read the slave's In-Service register to ensure that the interrupt just serviced was the only one generated by the slave. If the register is not empty, additional interrupts are pending, and an End of Interrupt command must not be sent to the master. If the register is empty, a nonspecific End of Interrupt command can be sent to the master.

The special fully-nested mode is selected through Initialization Command Byte 4. See "Initialization Command Byte 4" on page 10 for more information.

Automatic Rotation Mode

The automatic rotation mode accommodates multiple devices having the same interrupt priority. After a device is serviced, it is assigned the lowest priority and must wait until all other devices requesting an interrupt are serviced once before the first device is serviced again.

The following example shows the status and priorities of the In-Service register bits before and after bit 4 of the Interrupt-Request register is serviced by a Rotation on Nonspecific End of Interrupt command.

In-Service Register Bits	Status before Service	Priority before Rotate	Status after Service	Priority after Rotate
7	0	7 (Lowest)	0	2
6	1 (Pending)	6	1 (Pending)	1
5	0	5	0	0 (Highest)
4	1 (Pending)	4	0 (Serviced)	7 (Lowest)
3	0	3	0	6
2	0	2	0	5
1	0	1	0	4
0	0	0 (Highest)	0	3

Figure 2. Automatic Rotation Mode

The automatic rotation mode is selected by issuing a Rotation on Nonspecific End of Interrupt command through Operation Command Byte 2. See "Operation Command Byte 2" on page 11 for more information.

Specific Rotation Mode

The specific rotation mode allows the application programs to change the priority levels by assigning the lowest priority to a specific interrupt level. Once the lowest-level priority is selected, all other priority levels change. The following example compares the normal-nested mode to the specific rotation mode with bit 5 of the Interrupt Request register set to the lowest priority.

Interrupt Request Register Bits	Nested Mode Priority Level	Specific Rotation Mode Priority Level
7	7 (Lowest)	1
6	6	0 (Highest)
5	5	7 (Lowest)
4	4	6
3	3	5
2	2	4
1	1	3
0	0 (Highest)	2

Figure 3. Specific Rotation Mode when IRQ5 Has the Lowest Priority

The specific rotation mode is selected by issuing a Rotate on Specific End of Interrupt command or a Set Priority command through Operation Command Byte 2. See "Operation Command Byte 2" on page 11 for more information.

Special Mask Mode

The special mask mode allows application programs to selectively enable and disable any interrupt or combination of interrupts at any time during its execution. The special mask mode is selected through Operation Command Byte 3. Once the controller is in the special mask mode, setting a bit in Operation Command Byte 1 sets a corresponding bit in the Interrupt Mask register. Each bit set in the Interrupt Mask register masks the corresponding interrupt channel. Interrupt channels above and below a masked channel are not affected. See "Operation Command Byte 1" on page 11 and "Operation Command Byte 3" on page 12 for more information.

Poll Mode

Before the poll mode can be used, a CLI instruction must be issued to disable the system-microprocessor interrupt input. Devices are serviced by software issuing a Poll command through Operation Command Byte 3. The first 'read' pulse following a Poll command is interpreted by the controller as an 'interrupt acknowledge' pulse; the controller sets the appropriate in-service bit and reads the priority level. The byte placed on the data bus during a 'read' pulse is shown in the following figure.

Bit	Function
7	Interrupt Present
6-3	Undefined
2-0	Highest Priority Level

Figure 4. Poll Mode Status Byte

Bit 7 This bit is set to 1 if an interrupt is present.

Bits 6 - 3 These bits are not used and may be set to either 0 or 1.

Bits 2 - 0 These bits contain the binary code of the highest priority level requesting service.

Level-Sensitive Mode

The interrupt controller cannot be placed in the edge-triggered mode. In the level-sensitive mode, interrupt requests are recognized by a high level on the interrupt-request input. Interrupt requests must be removed before the End of Interrupt command is issued to prevent a second interrupt from occurring.

Programming the Interrupt Controller

Before the system can be used, the interrupt controller must be programmed with four sequential initialization commands. When a command is issued to a master at port hex 0020 (or a slave at port hex 00A0) with bit 4 set to 1, the command is recognized as Initialization Command Byte 1. Initialization Command Byte 1 is the first of four initialization commands required to program the interrupt controller. The following events occur during the initialization sequence:

1. The level sense circuit is set to the level-sensitive mode. (Following the initialization procedure, interrupts are generated by a high level on the interrupt-request input.)
2. The Interrupt Mask register is cleared.
3. IRQ7 is assigned priority 7.
4. The slave mode address is set to 7.

5. The special mask mode is cleared and the controller is set to read the Interrupt Request register.

Once the interrupt controller is programmed by the initialization command bytes, the controller can be programmed by the operation command bytes to operate in other modes.

Note: The master interrupt controller must be initialized before the slave interrupt controller. Failure to do so will cause unexpected results.

Initialization Command Byte 1

This is the first byte of the 4-byte initialization command sequence. This byte is issued to either the master (port hex 0020) or the slave (port hex 00A0).

Bit	Function
7 - 5	Reserved — Must be set to 0.
4	Initialization Command Byte 1 Identifier — Must be set to 1.
3	Level-Sensitive Mode — Must be set to 1.
2	Call Address Interval of 8 — Must be set to 0.
1	Cascade Mode — Must be set to 0.
0	4-byte Initialization Command Sequence — Must be set to 1.

Figure 5. Initialization Command Byte 1

Initialization Command Byte 2

This byte defines the address of the interrupt vector. Bits 7 through 3 define the five high-order bits of the interrupt vector address. Bits 2 through 0 are initialized to 0 and replaced by the hardware interrupt level when an interrupt occurs. This byte is issued to either the master (port hex 0021) or the slave (port hex 00A1).

Bit	Function
7 - 3	Bits 7 - 3 of the Interrupt Vector Address
2 - 0	Initialized to 0

Figure 6. Initialization Command Byte 2

Initialization Command Byte 3

This byte loads a value into an 8-bit slave register.

- In the master device mode, this byte has a value of hex 04 to identify interrupt 02 as a slave providing the input request.
- In the slave device mode, this byte has a value of hex 02 to tell the slave that it is using hardware interrupt 02 to communicate with the master. The slave compares this value to the cascade input; if they are equal, the slave releases the Interrupt Vector Address on the data bus.

This byte is issued to either the master (port hex 0021) or the slave (port hex 00A1).

Bit	Slave Function	Master Function
7 - 3	0	0
2	1	0
1	0	1
0	0	0

Figure 7. Initialization Command Byte 3

Initialization Command Byte 4

This byte is issued to either the master (port hex 0021) or the slave (port hex 00A1).

Bit	Function
7 - 5	Reserved – Must be set to 0.
4	Special Fully-Nested Mode
3, 2	Reserved – Must be set to 0.
1	Normal End of Interrupt – Must be set to 0.
0	80286/80386 Microprocessor Mode – Must be set to 1.

Figure 8. Initialization Command Byte 4

Operation Command Byte 1

This byte controls the individual bits in the Interrupt Mask register.

This byte is issued to either the master (port hex 0021) or the slave (port hex 00A1).

Bit	Function
7 - 0	Interrupt Mask Bits 7 - 0

Figure 9. Operation Command Byte 1

Bits 7 - 0 When set to 1, these bits inhibit their respective interrupt request input signals.

Operation Command Byte 2

This byte controls the interrupt priority and End of Interrupt command.

This byte is issued to either a master (port hex 0020) or a slave (port hex 00A0).

Bit	Function
7	Rotate Mode
6	Set Interrupt Level
5	End of Interrupt Mode
4, 3	Reserved – Must be set to 0
2 - 0	Interrupt Level (When bit 6 = 1)

Figure 10. Operation Command Byte 2

Bits 7 - 5 These bits define the rotate mode, end of interrupt mode, or a combination of the two, as shown in Figure 11 on page 12.

Bit 7	Bit 6	Bit 5	Function
0	0	0	Reserved
0	0	1	Nonspecific End of Interrupt Command
0	1	0	No Operation
0	1	1	Specific End of Interrupt Command*
1	0	0	Reserved
1	0	1	Rotate on Nonspecific End of Interrupt Command
1	1	0	Set Priority Command**
1	1	1	Rotate on Specific End of Interrupt Command**

* Bits 0, 1, and 2 are the binary level of the in-service bit to be reset.

** Bits 0, 1, and 2 are the binary level of the lowest priority device.

Figure 11. Operation Command Byte 2 (Bits 7 - 5)

Bits 4, 3 These bits are reserved and must be set to 0.

Bits 2 - 0 These bits define the hardware interrupt level to be acted upon when bit 6 is set to 1.

Bit 2	Bit 1	Bit 0	Function
0	0	0	Interrupt Level 0
0	0	1	Interrupt Level 1
0	1	0	Interrupt Level 2
0	1	1	Interrupt Level 3
1	0	0	Interrupt Level 4
1	0	1	Interrupt Level 5
1	1	0	Interrupt Level 6
1	1	1	Interrupt Level 7

Figure 12. Operation Command Byte 2 (Bits 2 - 0)

Operation Command Byte 3

This byte is issued to either a master (port hex 0020) or a slave (port hex 00A0).

Bit	Function
7	Reserved – Must be set to 0.
6, 5	Special Mask Mode Bits
4	Reserved – Must be set to 0.

Figure 13 (Part 1 of 2). Operation Command Byte 3

Bit	Function
3	Reserved – Must be set to 1.
2	Poll Command
1, 0	Read Register Command

Figure 13 (Part 2 of 2). Operation Command Byte 3

Bit 7 This bit is reserved and must be set to 0.

Bits 6, 5 These bits enable the special mask mode, as shown in the following figure.

Bit 6	Bit 5	Function
0	0	No Action
0	1	No Action
1	0	Normal Mask Mode
1	1	Special Mask Mode

Figure 14. Operation Command Byte 3 (Bits 6 and 5)

Bit 4 This bit is reserved and must be set to 0.

Bit 3 This bit is reserved and must be set to 1.

Bit 2 When set to 1, this bit sets the Poll command.

Bits 1, 0 These bits determine the register to be read on the next 'read' pulse, as shown in the following figure.

Bit 1	Bit 0	Function
0	0	No Action
0	1	No Action
1	0	Read Interrupt Request Register
1	1	Read In-Service Register

Figure 15. Operation Command Byte 3 (Bits 1 and 0)

Notes:

System Timers (Type 1)

Description 1
Channel 0 - System Timer 2
Channel 2 - Tone Generation for Speaker 2
Channel 3 - Watchdog Timer 4
Counters 0, 2, and 3 5
Programming the System Timers 5
Counter Write Operations 5
Counter Read Operations 6
Registers 6
Count Register - Channel 0 (Hex 0040) 6
Count Register - Channel 2 (Hex 0042) 6
Control Byte Register - Channel 0 or 2 (Hex 0043) 7
Count Register - Channel 3 (Hex 0044) 9
Control Byte Register - Channel 3 (Hex 0047) 9
Counter Latch Command 10
System Timer Modes 11
Mode 0 - Interrupt on Terminal Count 11
Mode 1 - Hardware Retriggerable One-Shot 12
Mode 2 - Rate Generator 13
Mode 3 - Square Wave 13
Mode 4 - Software Retriggerable Strobe 15
Mode 5 - Hardware Retriggerable Strobe 16
Operations Common to All Modes 17

Figures

1. Counters 1
2. Audio Subsystem Block Diagram 3
3. System Timer/Counter Registers 6
4. Select Counter Bits, Port Hex 0043 7
5. Read/Write Counter Bits, Port Hex 0043 7
6. Counter Mode Bits, Port Hex 0043 8
7. Select Counter, Port Hex 0047 9
8. Read/Write Counter, Port Hex 0047 9
9. Counter Latch Command 10
10. Minimum and Maximum Initial Counts, Counters 0, 2 17

Description

The system has three programmable timers/counters: Channel 0 is the System Timer, Channel 2 is the Tone Generator for the speaker, and Channel 3 is the Watchdog Timer. Channel 0 and Channel 2 are similar to Channel 0 and Channel 2 of the IBM Personal Computer, IBM Personal Computer XT™, and the IBM Personal Computer AT®. Channel 3 does not have a counterpart in earlier IBM personal computer systems. The following is a block diagram of the counters.

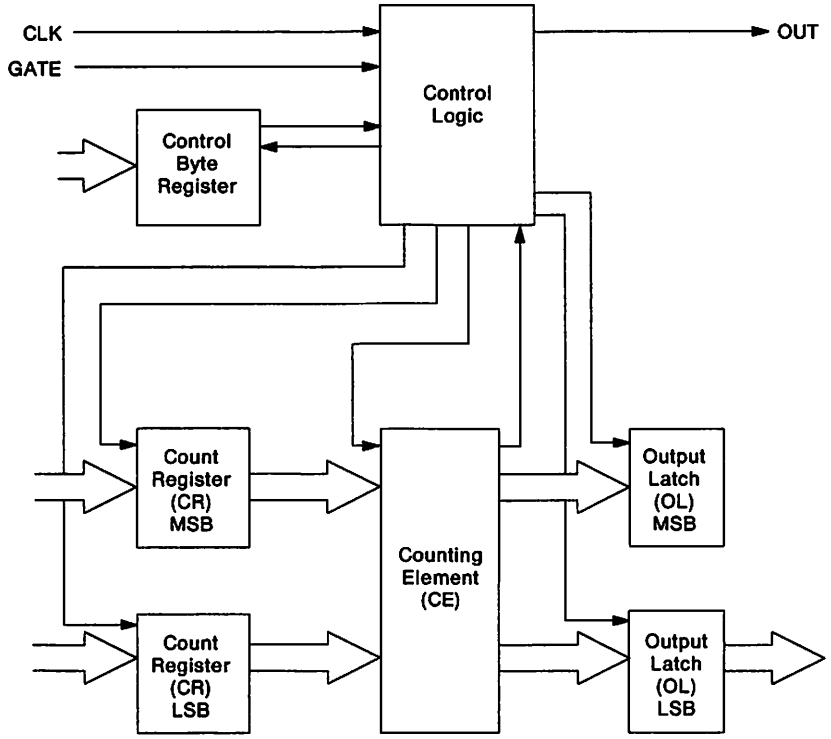


Figure 1. Counters

Personal Computer XT is a trademark of the International Business Machines Corporation.

Personal Computer AT is a registered trademark of the International Business Machines Corporation.

Channel 0 - System Timer

- GATE 0 is always enabled.
- CLK IN 0 is driven by a 1.193 MHz signal.
- CLK OUT 0 indirectly drives 'interrupt request 0' signal (IRQ 0).

  IRQ 0 is driven by a latch that is set by the rising edge of the 'clock out 0' signal (CLK OUT 0). The latch may be cleared by a system reset, an interrupt acknowledge cycle with a vector of hex 08, or an I/O write to System Control Port B (hex 0061) setting bit 7 to 1.

  Signals derived from CLK OUT 0 are used to gate and clock Channel 3.

Channel 2 - Tone Generation for Speaker

- GATE 2 is controlled by bit 0 of port hex 0061.
- CLK IN 2 is driven by a 1.193 MHz signal.
- CLK OUT 2 has two connections. One is to input port hex 0061, bit 5. CLK OUT 2 is also logically ANDed with port hex 0061, bit 1 (speaker data enable). The output of the AND gate drives the 'audio sum node' signal.

The audio subsystem is a speaker driven by a linear amplifier. The linear amplifier input node can be driven from the following sources:

- System-timer Channel 2 can be enabled to drive the speaker using bit 1 of I/O port hex 0061 set to 1. For information about system timer Channel 2 see "Description" on page 1.
- The channel using the 'audio sum node' signal.

The following block diagram shows the audio subsystem.

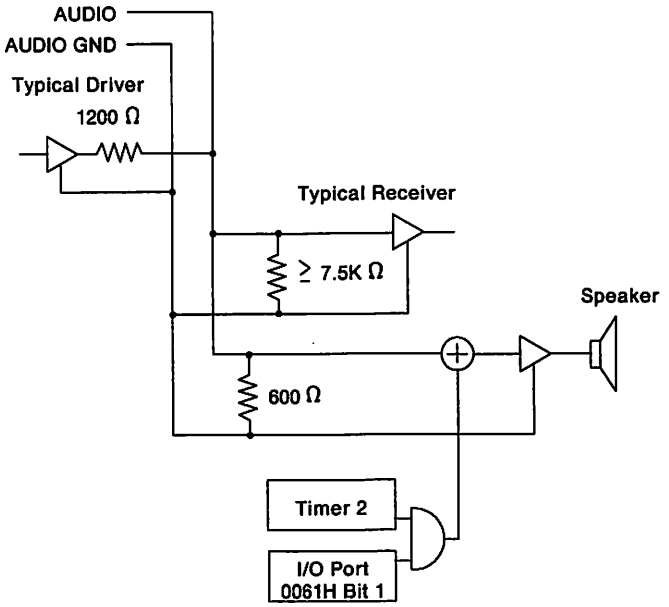


Figure 2. Audio Subsystem Block Diagram

Each audio driver must have a 1200 ohm source impedance, and a 7.5 kilohm or greater impedance is required for each audio receiver. Volume control is provided by the driver. Output level is a function of the number of drivers and receivers that share the AUDIO line.

Logic ground is connected to AUDIO GND at the amplifier.

Channel 3 - Watchdog Timer

This channel operates only in Mode 0, and counts in 8-bit binary.

- GATE 3 is tied to IRQ 0.
- CLK IN 3 is tied to CLK OUT 0 inverted.
- CLK OUT 3, when high, drives the NMI active.

The Watchdog Timer detects when IRQ 0 is active for more than one period of CLK OUT 0. If IRQ 0 is active when a rising edge of CLK OUT 0 occurs, the count is decremented. When the count is decremented to 0, an NMI is generated. Thus, the Watchdog Timer can be used to detect when IRQ 0 is not being serviced. This is useful for detecting error conditions.

BIOS interfaces are provided to enable and disable the Watchdog Timer. When the Watchdog Timer times out, it causes an NMI and sets System Control Port A (hex 0092), bit 4 to 1. This bit may be set to 0 by using the BIOS interface to disable the Watchdog Timer.

Note: The NMI stops all arbitration on the bus until bit 6 of the Arbitration register (I/O address hex 0090) is set to 0. This can result in lost data or an overrun error on some I/O devices.

If the Watchdog Timer is used to detect "tight looping" software tasks that inhibit interrupts, some I/O devices may be overrun (not serviced in time). The operating system may be required to restart these devices.

When the Watchdog Timer is enabled, the 'inhibit' signal (INHIBIT) is active only when IRQ 0 is pending for longer than one period of CLK OUT 0. When INHIBIT is active, any data written to Channel 0 or Channel 3 is ignored. INHIBIT is never active if the Watchdog Timer is disabled.

The Watchdog Timer operation is defined only when Channel 0 is programmed in Mode 2 or Mode 3. The operation of the Watchdog Timer is undefined when Channel 0 is programmed in any other mode.

Counters 0, 2, and 3

Each counter is independent. Counters 0 and 2 are 16-bit down counters that can be preset. They can count in binary or binary coded decimal (BCD). Counter 3 is an 8-bit down counter that can be preset. It counts in binary only.

Programming the System Timers

The system treats the programmable interval timer as an arrangement of five external I/O ports. Three ports are treated as count registers and two are control registers for mode programming. Counters are programmed by writing a control word and then an initial count. All control words are written into the Control Word registers, which are located at address hex 0043 for counters 0 and 2, and address hex 0047 for counter 3. Initial counts are written into the Count registers, not the Control Byte registers. The format of the initial count is determined by the control word used.

After the initial count is written to the Count register, it is transferred to the counting element, according to the mode definition. When the count is read, the data is presented by the output latch.

Counter Write Operations

The control word must be written before the initial count, and the count must follow the count format specified in the control word.

A new initial count may be written to the counters at any time without affecting the counter's programmed mode. Counting is affected as described in the mode definitions. The new count must follow the programmed count format.

Counter Read Operations

The counters can be read using the Counter Latch command (see "Counter Latch Command" on page 10).

If the counter is programmed for two-byte counts, two bytes must be read. The two bytes need not be read consecutively; read, write, or programming operations of other counters may be inserted between them.

Note: If the counters are programmed to read or write two-byte counts, the program must not transfer control between writing the first and second byte to another routine that also reads or writes into the same counter. This will cause an incorrect count.

Registers

I/O Address (Hex)	Register
0040	Count Register - Channel 0 (Read/Write)
0042	Count Register - Channel 2 (Read/Write)
0043	Control Byte Register - Channel 0 or 3 (Write)
0044	Count Register - Channel 3 (Read/Write)
0047	Control Byte Register - Channel 3 (Write)

Figure 3. System Timer/Counter Registers

Count Register - Channel 0 (Hex 0040)

The control byte is written to port hex 0043 indicating the format of the count (least-significant byte only, most-significant byte only, or least-significant byte followed by most-significant byte). This must be done before writing the count to port hex 0040.

Count Register - Channel 2 (Hex 0042)

The control byte is written to port hex 0043 indicating the format of the count (least-significant byte only, most-significant byte only, or least-significant byte followed by most-significant byte). This must be done before writing the count to port hex 0042.

Control Byte Register - Channel 0 or 2 (Hex 0043)

This is a write-only register. The following gives the format for the control byte (port hex 0043) for counters 0 and 2.

Bits 7, 6 These bits select Counter 0 or 2.

Bits 7 6	Function
0 0	Select Counter 0
0 1	Reserved
1 0	Select Counter 2
1 1	Reserved

Figure 4. Select Counter Bits, Port Hex 0043

Bits 5, 4 These bits distinguish a counter latch command from a control byte. If a control byte is selected, these bits also determine the method in which each byte is read or written.

Bits 5 4	Function
0 0	Counter Latch Command
0 1	Read/Write Counter bits 0 - 7 only
1 0	Read/Write Counter bits 8 - 15 only
1 1	Read/Write Counter bits 0 - 7 first, then bits 8 - 15

Figure 5. Read/Write Counter Bits, Port Hex 0043

Bits 3 - 1 These bits select the mode.

Bits 3 2 1	Function
0 0 0	Mode 0 - Interrupt on Terminal Count
0 0 1	Mode 1 - Hardware Retriggerable One Shot
X 1 0	Mode 2 - Rate Generator
X 1 1	Mode 3 - Square Wave
1 0 0	Mode 4 - Software Retriggerable Strobe
1 0 1	Mode 5 - Hardware Retriggerable Strobe

Don't care bits (X) should be set to 0.

Figure 6. Counter Mode Bits, Port Hex 0043

Bit 0 When set to 1, this bit selects the binary coded decimal method of counting. When set to 0, it selects the 16-bit binary method.

Count Register - Channel 3 (Hex 0044)

The control byte is written to port hex 0047 indicating the format of the count (least-significant byte only). This must be done before writing the count to port hex 0044.

Control Byte Register - Channel 3 (Hex 0047)

This is a write-only register. The following gives the format for the control byte (port hex 0047) for counter 3.

Bits 7, 6 These bits select Counter 3.

Bits 7 6	Function
0 0	Select Counter 3
0 1	Reserved
1 0	Reserved
1 1	Reserved

Figure 7. Select Counter, Port Hex 0047

Bits 5, 4 These bits distinguish a counter latch command from a control byte.

Bits 5 4	Function
0 0	Counter Latch Command Select Counter 0
0 1	R/W Counter Bits 0 - 7 Only
1 0	Reserved
1 1	Reserved

Figure 8. Read/Write Counter, Port Hex 0047

Bits 3 - 0 These bits are reserved and must be written as 0.

Counter Latch Command

The Counter Latch command is written to the Control Byte register. Bits 7 and 6 select the counter, and bits 5 and 4 distinguish this command from a control byte. The following figure shows the format of the Counter Latch command.

Bit	Function
7, 6	Specifies the counter to be latched
5, 4	00 Specifies the Counter Latch command
3 - 0	Reserved = 0

Figure 9. Counter Latch Command

The count is latched into the selected counter's output latch when the Counter Latch command is received. This count is held in the latch until it is read by the system microprocessor (or until the counter is reprogrammed). After the count is read by the system microprocessor, it is automatically unlatched, and the output latch returns to following the counting element. Counter Latch commands do not affect the programmed mode of the counter in any way. All subsequent latch commands issued to a given counter before the count is read, are ignored. A read cycle to the counter latch returns the value latched by the first Counter Latch command.

System Timer Modes

The following definitions are used when describing the timer modes.

CLK pulse A rising edge, then a falling edge on the counter CLK input.

Trigger A rising edge on a counter's input GATE.

Counter Load The transfer of a count from the Counter register to the counting element.

Mode 0 - Interrupt on Terminal Count

Event counting can be done using Mode 0. Counting is enabled when GATE is equal to 1, and disabled when GATE is equal to 0. If GATE is equal to 1 when the control byte and initial count are written to the counter, the sequence is as follows:

1. The control byte is written to the counter, and OUT goes low.
2. The initial count is written.
3. Initial count is loaded on the next CLK pulse. The count is not decremented for this CLK pulse.

   The count is decremented until the counter reaches 0. For an initial count of N, the counter reaches 0 after N+1 CLK pulses.
4. OUT goes high.

OUT remains high until a new count or new Mode 0 control byte is written into the counter.

If GATE equals 0 when an initial count is written to the counter, it is loaded on the next CLK pulse even though counting is not enabled. After GATE enables counting, OUT goes high N CLK pulses later.

If a new count is written to a counter while counting, it is loaded on the next CLK pulse. Counting then continues from the new count. If a 2-byte count is written to the counter the following occurs:

1. The first byte written to the counter disables the counting. OUT goes low immediately, and there is no delay for the CLK pulse.
2. When the second byte is written to the counter, the new count is loaded on the next CLK pulse. OUT goes high when the counter reaches 0.

Mode 1 - Hardware Retriggerable One-Shot

The sequence for Mode 1 is as follows:

1. OUT is high.
2. On the CLK pulse following a trigger, OUT goes low and begins the one-shot pulse.
3. When the counter reaches zero, OUT goes high.

OUT remains high until the CLK pulse following the next trigger.

The counter is armed by writing the control byte and initial count to the counter. When a trigger occurs, the counter is loaded. OUT goes low on the next CLK pulse, starting the one-shot pulse. For an initial count of N, a one-shot pulse is N CLK pulses long. The one-shot pulse repeats the count of N for the next triggers. OUT remains low N CLK pulses following any trigger. GATE does not affect OUT. The current one-shot pulse is not affected by a new count written to the counter, unless the counter is retriggered. If the counter is retriggered, the new count is loaded and the one-shot pulse continues.

Note: Mode 1 is valid only for Counter 2.

Mode 2 - Rate Generator

This mode causes the counter to perform a divide-by-N function. Counting is enabled when GATE equals 1, and disabled when GATE equals 0.

The sequence for Mode 2 is as follows:

1. OUT is high.
2. The initial count decrements to 1.
3. OUT goes low for one CLK pulse.
4. OUT goes high.
5. The counter reloads the initial count.
6. The process is repeated.

If GATE goes low during the OUT pulse, OUT goes high. On the next CLK pulse a trigger reloads the counter with the initial count. OUT goes low N CLK pulses after the trigger. This allows the GATE input to be used to synchronize the counter.

The counter is loaded on the CLK pulse after a control byte and initial count are written to the counter. OUT goes low N CLK pulses after writing the initial count. This allows software to synchronize the counter.

The current counting sequence is not affected by a new count being written to the counter. If the counter receives a trigger after a new count is written and before the end of the current count, the new count is loaded on the next CLK pulse, and counting continues from the new count. If the trigger is not received by the counter, the new count is loaded following the current counting cycle.

Mode 3 - Square Wave

Mode 3 is similar to Mode 2 except for the duty cycle of OUT. Counting is enabled when GATE is equal to 1, and disabled when GATE is equal to 0. An initial count of N results in a square wave on OUT. The period of the square wave is N CLK pulses. If OUT is low and GATE goes low, OUT goes high. On the next CLK pulse, a trigger reloads the counter with the initial count.

The counter is loaded on the CLK pulse following the writing of a control byte and the initial count.

The current counting sequence is not affected by a new count being written to the counter. If the counter receives a trigger after a new count is written, and before the end of the current count's half-cycle of the square wave, the new count is loaded on the next CLK pulse, and counting continues from the new count. If the trigger is not received by the counter, the new count is loaded following the current half-cycle.

The way Mode 3 is implemented depends on whether the count written is odd or even. If the count is even, OUT begins high and the following applies:

1. The initial count is loaded on the first CLK pulse.
2. The count is decremented by 2 on succeeding CLK pulses.
3. The count decrements to 0.
4. OUT changes state.
5. The counter is reloaded with the initial count.
6. The process repeats indefinitely.

If the count is odd, the following applies:

1. OUT is high.
2. The initial count minus 1 is loaded on the first CLK pulse.
3. The count is decremented by 2 on succeeding CLK pulses.
4. The count decrements to 0.
5. One CLK pulse after the count reaches 0, OUT goes low.
6. The counter is reloaded with the initial count minus 1.
7. Succeeding CLK pulses decrements the count by 2.
8. The count decrements to 0.
9. OUT goes high.
10. The counter is reloaded with the initial count minus 1.
11. The process repeats indefinitely.

Mode 3, using an odd count, causes OUT to go high for a count of (N+1)/2 and low for a count of (N-1)/2.

Mode 3 may operate such that OUT is initially set low when the control byte is written. For this condition, Mode 3 operates as follows:

1. OUT is low.
2. The count decrements to half of the initial count.
3. OUT goes high.
4. The count decrements to 0.
5. OUT goes low.
6. The process repeats indefinitely.

This process results in a square wave with a period of N CLK pulses.

Note: If OUT needs to be high after the control byte is written, the control byte must be written twice. This applies only to Mode 3.

Mode 4 - Software Retriggerable Strobe

Counting is enabled when GATE equals 1, and disabled when GATE equals 0. Counting begins when an initial count is written.

The sequence for Mode 4 is as follows:

1. OUT is high.
2. The control byte and initial count are written to the counter.
3. The initial count is loaded on the next CLK pulse. The count is not decremented for this clock pulse.
4. The count is decremented to 0. For an initial count of N, the counter reaches 0 after N+1 CLK pulses.
5. OUT goes low for one CLK pulse.
6. OUT goes high.

GATE should not go low one-half CLK pulse before or after OUT goes low. If this occurs, OUT remains low until GATE goes high.

If a new count is written to a counter while counting, it is loaded on the next CLK pulse. Counting then continues from the new count. If a 2-byte count is written, the following occurs:

1. Writing the first byte does not affect counting.
2. The new count is loaded on the CLK pulse following the writing of the second byte.

The Mode 4 sequence can be retriggered by software. The period from when the new count of N is written to when OUT strobes low is (N+1) pulses.

Mode 5 - Hardware Retriggerable Strobe

The sequence for Mode 5 is as follows:

1. OUT is high.
2. The control byte and initial count are written to the counter.
3. Counting is triggered by a rising edge of GATE.
4. The counter is loaded on the CLK pulse following the trigger. This CLK pulse does not decrement the count.
5. The count decrement to 0.
6. OUT goes low for one CLK pulse. This occurs (N+1) CLK pulses after the trigger.
7. OUT goes high.

The counting sequence can be retriggered. OUT strobes low (N+1) pulses after the trigger. GATE does not affect OUT.

The current counting sequence is not affected by a new count being written to the counter. If the counter receives a trigger after a new count is written and before the end of the current count, the new count is loaded on the next CLK pulse, and counting continues from the new count.

Note: Mode 5 is valid only on Counter 2.

Operations Common to All Modes

Control bytes written to a counter cause all control logic to reset. OUT goes to a known state. This does not take a CLK pulse.

The falling edge of the CLK pulse occurs when new counts are loaded and counters are decremented.

Counters do not stop when they reach zero. In Modes 0, 1, 4, and 5 the counter wraps to the highest count, and continues counting. Modes 2 and 3 are periodic; the counter reloads itself with the initial count and continues from there.

The GATE is sampled on the rising edge of the CLK pulse.

The following shows the minimum and maximum initial counts for the counters.

Mode	Minimum Count	Maximum Count
0	1	$0 = 2^{16}$ (Binary Counting) or 10^4 (BCD Counting)
1	1	$0 = 2^{16}$ (Binary Counting) or 10^4 (BCD Counting)
2	2	$0 = 2^{16}$ (Binary Counting) or 10^4 (BCD Counting)
3	2	$0 = 2^{16}$ (Binary Counting) or 10^4 (BCD Counting)
4	1	$0 = 2^{16}$ (Binary Counting) or 10^4 (BCD Counting)
5	1	$0 = 2^{16}$ (Binary Counting) or 10^4 (BCD Counting)

Figure 10. Minimum and Maximum Initial Counts, Counters 0, 2

Counter 3 can use only Mode 0, Interrupt on Terminal Count. The minimum initial count is 1 and the maximum is hex FF.

Notes:

Diskette Drive Controller

| Description 1
FIFO Mode 3
| Diskette Drive Controller Registers 3
Status Register A (Hex 03F0) 4
| Status Register B (Hex 03F1) 4
Drive Control Register (Hex 03F2) 5
Drive Status Register (Hex 03F3 Read) 5
Diskette Drive Controller Status Register (Hex 03F4 Read) 7
Precompensation Select Register (Hex 03F4 Write) 8
Diskette Drive Controller Command/Data Register (Hex 03F5) . 9
Reserved Register (Hex 03F6) 9
Data Rate Status Register (Hex 03F7 - Read) 9
Data Rate Control Register (Hex 03F7 - Write) 10
Diskette Drive Controller Programming Considerations 10
Controller Commands 11
Configure Command 15
Dumpreg Command 16
Format Track Command 16
| Lock Command 17
| Perpendicular Mode Command 17
Read Data Command 18
Read Deleted Data Command 18
Read ID Command 19
Read Track Command 20
Recalibrate Command 20
Relative Seek Command 21
Scan Equal Command 21
Scan High or Equal Command 22
Scan Low or Equal Command 22
Seek Command 23
Sense Drive Status Command 23
Sense Interrupt Status Command 24
Specify Command 24
Verify Command 25
Version Command 25
Write Data Command 26
Write Deleted Data Command 26
Invalid Command Status 27
Command Status Registers 28
Status Register 0 28
Status Register 1 29
Status Register 2 30

© Copyright IBM Corp. 1991

Status Register 3 31
Interface Signal Descriptions 32
Output Signals 32
Input Signals 34
Connector 35

Index 37

Figures

1. Media Format Table 2
2. Status Register A (Hex 03F0) 4
3. Status Register B (Hex 03F1) 4
4. Drive Control Register (Hex 03F2) 5
5. Drive Status Register (Hex 03F3 Read) 5
6. Media Type 6
7. Drive Type 6
8. Start-Up Drive 6
9. Diskette Drive Controller Status Register (Hex 03F4) 7
10. Precompensation Select Register (Hex 03F4 Write) 8
11. Precompensation Values 8
12. Default Precompensation Values 9
13. Data Rate Status Register (Hex 03F7 - Read) 9
14. Data Rate Control Register (Hex 03F7 - Write) 10
15. Data Rate Selection 10
16. Command Symbols 11
17. HLT Parameter Definitions 14
18. HUT Parameter Definitions 14
19. SRT Parameter Definitions 15
20. Configure Command 15
21. Dumpreg Command 16
22. Dumpreg Result 16
23. Format Track Command 16
24. Format Track Result 17
25. Lock Command 17
26. Lock Result 17
27. Perpendicular Mode Command 17
28. Read Data Command 18
29. Read Data Result 18
30. Read Deleted Data Command 18
31. Read Deleted Data Result 19
32. Read ID Command 19
33. Read ID Result 19
34. Read Track Command 20
35. Read Track Result 20
36. Recalibrate Command 20
37. Relative Seek Command 21
38. Scan Equal Command 21
39. Scan Equal Result 21
40. Scan High or Equal Command 22
41. Scan High or Equal Result 22
42. Scan Low or Equal Command 22

© Copyright IBM Corp. 1991

43. Scan Low or Equal Result 23
44. Seek Command 23
45. Sense Drive Status Command 23
46. Sense Drive Status Result 23
47. Sense Interrupt Status Command 24
48. Sense Interrupt Status Result 24
49. Specify Command 24
50. Verify Command 25
51. Verify Result 25
52. Version Command 25
53. Version Result 25
54. Write Data Command 26
55. Write Data Result 26
56. Write Deleted Data Command 26
57. Write Deleted Data Result 27
58. Invalid Command Result 27
59. Status Register 0 28
60. Status Register 0 (Bits 7, 6) 28
61. Status Register 0 (Bits 1, 0) 28
62. Status Register 1 29
63. Status Register 2 30
64. Status Register 3 31
65. Diskette Drive Connector Signal Assignments For the 40-pin Card Edge Interface 35
66. Diskette Drive Connector Signal Assignments For the 34-pin Header Interface 35
67. Diskette Drive Connector Signal Assignments For the Enhanced 34-Pin Header Interface 36

Description

The diskette drive controller and interface connector reside on the system board. The controller can be a Type 1 or a Type 2. The controller type is dependent on the system model and can be determined by issuing the Unlock command. The Type 1 returns a hex 80 in the Result byte; the Type 2 returns a hex 00.

The Type 1 controller supports:

- Two data transfer rates:
  - 250,000 bits per second (bps)
  - 500,000 bits per second.
- 125 nanoseconds of precompensation on all tracks
- The following IBM diskette drives:
  - 3.5-inch 1.44MB
  - 5.25-inch 360KB.

The Type 2 controller supports:

- Four data transfer rates:
  - 250,000 bits per second
  - 300,000 bits per second
  - 500,000 bits per second
  - 1,000,000 bits per second.
- Programmable precompensation
- 16 bytes of data buffering
- The following IBM diskette drives:
  - 3.5-inch 1.44MB
  - 3.5-inch 2.88MB
  - 5.25-inch 360KB
  - 5.25-inch 1.2MB.

The media formats supported by IBM drives are shown below.

Media Size	Capacity		Sectors/ Track	No. of Tracks	Data Rate (kbps)
	Unformatted	Formatted			
3.5 in.	1.0MB	720KB	9	80	250
3.5 in.	2.0MB	1.44MB	18	80	500
3.5 in.	4.0MB	2.88MB	36	80	1000*
5.25 in.	0.5MB	360KB	9	40	300* /250
5.25 in.	1.6MB	1.20MB*	15	80	500
* Type 2 controller only					

Figure 1. Media Format Table

Warning: The controller does not check to see that the media supports the selected capacity. Attempting to format media to an unsupported capacity may cause loss of data.

- 0.5MB (double sided, double density 5.25-inch) media can be reliably formatted only to the 360KB capacity.
- 1.0MB media can be reliably formatted only to the 720KB capacity.
- 1.6MB (high capacity 5.25-inch) media can be reliably formatted only to the 1.2MB capacity.
- 2.0MB media can be reliably formatted only to the 1.44MB capacity.
- 4.0MB media can be reliably formatted only to the 2.88MB capacity.

When the Type 1 diskette drive controller is switched from one data rate to another, the controller clock rate automatically changes to:

- 8 MHz for 2.0MB mode
- 4 MHz for 1.0MB mode.

The step rate time (SRT), head load time (HLT), and the head unload time (HUT) parameters must then be changed to maintain the desired timings at the diskette interface.

For Type 2 controllers, the controller clock rate remains at 24 MHz, but the SRT, HLT, and HUT parameters must still be updated using the same values as the Type 1 controller.

Note: With Type 1 controllers, 32-bit operations to the video subsystem can cause a Direct Memory Access (DMA) overrun. If the BIOS returns an error code indicating that an overrun has occurred, the operation should be repeated.

FIFO Mode

A 16-byte FIFO is provided on the Type 2 controller, which allows direct-memory-access (DMA) data transfers to be delayed for longer periods of time without causing DMA overrun errors. To maintain compatibility, the FIFO defaults to a Type 1 compatibility mode after system reset. All functions are then similar to the Type 1 controller.

The Configure command is used to enable FIFO operations. After the FIFO operations are enabled, the controller temporarily enters a byte mode during the command and result phases of the diskette controller operation. While in this mode, operations to, or from, the disk controller are FIFO compatible.

The FIFO is enabled only during the data transfer phase of operation. All command and status information is transferred in Type 1 compatibility mode. When the Type 2 controller first enters the data transfer phase, the FIFO is cleared of any residual data from previous operations.

Compatibility problems may occur when the FIFO mode is used with software that monitors the progress of a data transfer during the execution phase. It is recommended that the FIFO mode be disabled when software of this type is used.

Diskette Drive Controller Registers

The diskette drive controller can be a Type 1 and a Type 2 controller. Certain functions are provided by the Type 2 controller only.

Note: Some registers contain bits that are labeled as reserved. These bits must be set to the value specified when writing to these registers.

Status Register A (Hex 03F0)

This read-only register shows the status of signals on the diskette drive interface.

Bit	Function
7	Interrupt Pending
6	-2nd Drive Installed
5	Step
4	-Track 0
3	Head 1 Select
2	-Index
1	-Write Protect
0	Direction In

Figure 2. Status Register A (Hex 03F0)

Status Register B (Hex 03F1)

This read-only register shows the status of signals on the diskette drive interface.

The write-data and read-data bits change state for each positive transition of the '-write data' or '-read data' signals.

Bit	Function
7, 6	Reserved
5	Drive Select 0
4	Write Data
3	Read Data
2	Write Enable
1	Motor Enable 1
0	Motor Enable 0

Figure 3. Status Register B (Hex 03F1)

Drive Control Register (Hex 03F2)

This read and write register controls drive motors, drive selection, and feature enable. All bits are set to 0 by a reset.

Bit	Function
7	Motor Enable 3*
6	Motor Enable 2*
5	Motor Enable 1
4	Motor Enable 0
3	Reserved = 1
2	-Controller Reset
1	Drive Select 1*
0	Drive Select 0

* Type 2 Controller Only.

Figure 4. Drive Control Register (Hex 03F2)

Note: The controller reset bit must be set to 0 for a minimum of 3.5 microseconds to ensure the controller is properly reset.

Drive Status Register (Hex 03F3 Read)

This register contains information about the diskette drive type, and start-up drive location. This register is for the Type 2 controller only.

Bit	Function
7,6	Media Type 1,0
5,4	Drive Type 1,0
3,2	Start-Up Drive 1,0
1,0	Reserved

Figure 5. Drive Status Register (Hex 03F3 Read)

The following tables show the media type, drive type and start-up drive indicated.

Bits 7 6	Media Type
0 0	Reserved
0 1	4MB
1 0	2MB
1 1	1MB

Note: These bits indicate the state of the 'media type' signals. These signals are inputs from the drive.

Figure 6. Media Type

Bits 5 4	Drive Type
0 0	3.5-inch, 1.44MB
0 1	3.5-inch, 2.88MB
1 0	5.25 inch, 1.2MB
1 1	Reserved

Figure 7. Drive Type

Bits 3 2	Start-Up Drive
0 0	First Drive
0 1	Second Drive
1 0	Third Drive
1 1	Reserved

Figure 8. Start-Up Drive

Note: The media-type and drive-type bits are valid only when the drive is selected. The media-type bits are valid only for 3.5-inch media.

Diskette Drive Controller Status Register (Hex 03F4 Read)

This read-only register facilitates the transfer of data between the system microprocessor and the controller.

Bit	Function
7	Request for Master
6	Data Input/Output
5	Non-DMA Mode
4	Diskette Controller Busy
3	Drive 3 Busy
2	Drive 2 Busy
1	Drive 1 Busy
0	Drive 0 Busy

Figure 9. Diskette Drive Controller Status Register (Hex 03F4)

Bit 7 When this bit is 1, the Data register is ready to transfer data with the system microprocessor.

Bit 6 This bit indicates the direction of data transfer between the diskette drive controller and the system microprocessor. When this bit is 1, the transfer is to the system microprocessor; when the bit is 0, the transfer is to the controller.

Bit 5 When this bit is 1, the controller is in the non-DMA mode.

Bit 4 When this bit is 1, command is being processed.

Bit 3 When this bit is 1, diskette drive 3 is in the seek mode.

Bit 2 When this bit is 1, diskette drive 2 is in the seek mode.

Bit 1 When this bit is 1, diskette drive 1 is in the seek mode.

Bit 0 When this bit is 1, diskette drive 0 is in the seek mode.

Precompensation Select Register (Hex 03F4 Write)

This write-only register is used to select the data rate and precompensation value for each data rate. This register is supported by the Type 2 controllers only.

Bit	Function
7-5	Reserved = 0
4	Precomp 2
3	Precomp 1
2	Precomp 0
1	Data Rate Select 1
0	Data Rate Select 0

Figure 10. Precompensation Select Register (Hex 03F4 Write)

The following table shows the precompensation values selected.

Bits 4 3 2	Precompensation Delay
0 0 0	Default
0 0 1	41.7 ns
0 1 0	83.3 ns
0 1 1	125 ns
1 0 0	167 ns
1 0 1	208 ns
1 1 0	250 ns
1 1 1	0.0 ns

Figure 11. Precompensation Values

| The following table shows the transfer rate selected and the resulting
| default precompensation values.

Bits 1 0	Transfer Rate	Default Precomp
0 0	500 kbps	125 ns
0 1	300 kbps*	125 ns
1 0	250 kbps	125 ns
1 1	1000 kbps*	41.7 ns

* Type 2 Controller Only

Figure 12. Default Precompensation Values

Diskette Drive Controller Command/Data Register (Hex 03F5)

This read and write register passes data, commands and parameters to the diskette drive controller and provides diskette-drive status information.

Reserved Register (Hex 03F6)

This register is reserved.

Data Rate Status Register (Hex 03F7 - Read)

This read-only register identifies the data rate selected on the interface, and senses the state of the 'diskette change' and '-high density select' signals.

Bit	Function
7	Diskette Change
6-3	Reserved
2	Data Rate Select 1
1	Data Rate Select 0
0	-High Density Select

Figure 13. Data Rate Status Register (Hex 03F7 - Read)

Data Rate Control Register (Hex 03F7 - Write)

This write-only register sets the transfer rate.

Bit	Function
7-2	Reserved
1	Data Rate Select 1
0	Data Rate Select 0

Figure 14. Data Rate Control Register (Hex 03F7 - Write)

Bits 7 - 2 Reserved as 0.

Bits 1, 0 These bits select the data rate, as shown in the following figure.

Bits 1 0	Data Rate
0 0	500,000 bits per second
0 1	300,000 bits per second*
1 0	250,000 bits per second
1 1	1,000,000 bits per second*

* Type 2 Controller Only

Figure 15. Data Rate Selection

Diskette Drive Controller Programming Considerations

Each command is initiated by a multibyte transfer from the system microprocessor; the result can be a multibyte transfer back to the system microprocessor. Most transfers consist of three phases:

- *Command Phase:* The system microprocessor writes a series of command bytes to the controller directing it to perform a specific operation.
- *Execution Phase:* The controller performs the specified operation.
- *Result Phase:* After the operation is complete, status information is available to the system microprocessor through a sequence of read commands.

Note: The Seek, Relative Seek, and Recalibrate commands have no result phase. After issuing these commands, the Sense Interrupt Status command must be issued for proper

termination and verification of the head position (Present Cylinder Number parameter or PCN).

Controller Commands

The following are supported commands for diskette drive controller:

- Configure#
- Dumpreg#
- Format Track
- Read Data
- Read Deleted Data
- Read ID
- Read Track
- Recalibrate
- Relative Seek#
- Scan Equal
- Scan High or Equal
- Scan Low or Equal
- Seek
- Sense Drive Status
- Sense Interrupt Status
- Specify
- Verify#
- Version#
- Write Data
- Write Deleted Data.

Notes:

1. Commands marked with a pound sign are not supported by the Type 1 controller.
2. Diskette BIOS does not support all commands listed. To ensure software compatibility across system models, the diskette hardware should be accessed only through the diskette BIOS.

Symbol	Name	Description
D0 - D3	Drive Perpendicular	These bits represent the state of drive 0 through drive 3. When the bit is 1, the associated drive is a perpendicular drive. GAP and WGATE must not be set to 1 for these bits to be valid.
DIR	Direction Control	When set to 0, this bit causes the head to move away from the spindle during an implied seek. When set to 1, the head moves toward the spindle.
EC	Enable Count	When this bit is set to 1, the data-length byte indicates the number of sectors/track.
EFF	Enable FIFO	When set to 1, this bit enables the FIFO data buffer in the Type 2 controller.
EIS	Enable Implied Seek	When set to 1, this bit causes an implied seek to be performed before executing a command that requires the cylinder parameter in the command phase in the Type 2 controller.
GAP	GAP 2 Size	This parameter is used with WGATE to set the length of the GAP 2 written during write or format operations. GAP should be set to 0 for normal operation.
GPL	GAP 3 Length	This parameter determines the length of GAP 3 (in bytes) during a Format operation and determines the VCO synchronization timing during Read operations.
HD	Head Select	This bit selects the head used. When set to 0, head 0 is selected; when set to 1, head 1 is selected.
HLT	Head Load Time	This parameter specifies the head-load time (see Figure 17 on page 14).
HUT	Head Unload Time	This parameter specifies the head-unload time (see Figure 18 on page 14).
MFM	FM or MFM	When set to 0, this bit selects FM mode; when set to 1, it selects MFM mode.
MT	Multitrack	When set to 1, this bit selects multitrack operations. (Side 0 and Side 1 are automatically read from, written to, or verified).

Symbol	Name	Description
ND	Non-DMA Mode	When set to 1, this bit causes diskette operations to be performed in the non-DMA mode.
OW	Overwrite	When set to 1, this bit allows the bits specified by D0-D3 to be overwritten. **Warning:** To avoid the loss of data, this bit should not be set to 1.
PD	Polling Disable	When set to 1, this bit disables polling; when set to 0, it enables polling.
PTN	Precomp Track Number	This parameter selects the track number where write precompensation begins. It is programmable from 0 to hex FF.
RCN	Relative Cylinder Number	This parameter specifies the relative cylinder to go to from the present cylinder, as used by the Relative Seek command.
SK	Skip Flag	When this bit is set to 1, the sectors containing a deleted-data address mark are automatically skipped during a Read Data command. For a Read Deleted Data command, only sectors with a deleted-data address mark are accessed. When this bit is set to 0, the sector is read or written the same as the Read Data command.
SRT	Step Rate Time	This parameter specifies the stepping-rate time (see Figure 19 on page 15).
THR	Threshold	This parameter is the FIFO threshold for Type 2 controllers, where 0 = 1 byte, F = 16 bytes.
US	Unit Select	This parameter indicates the drive number selected. 00 selects drive 0, 01 selects drive 1, 10 selects drive 2 and 11 selects Drive 3.
WGATE	Write Gate	This parameter alters the timing of the 'write enable' signal for the perpendicular-recording mode. WGATE should be set 0 for normal operation.

Figure 16 (Part 2 of 2). Command Symbols

The following figure shows the head-load time selected as determined by the HLT parameter and the transfer rate. The transfer rates are shown in 1000 of bytes per second.

HLT Parameter	Head Load Time/Transfer Rate			
	1000 kbps	500 kbps	300 kbps	250 kbps
00	128	256	426	512
01	1	2	3.3	4
02	2	4	6.7	8
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
7E	126	252	420	504
7F	127	254	423	508
Note: Delay time in milliseconds				

Figure 17. HLT Parameter Definitions

The following figure shows the head-unload-time selected as determined by the HUT parameter and the transfer rate.

HUT Parameter	Head Unload Time/Transfer Rate			
	1000 kbps	500 kbps	300 kbps	250 kbps
0	128	256	426	512
1	8	16	26.7	32
2	16	32	53.4	64
3	24	48	80.1	96
4	32	64	107	128
5	40	80	134	160
6	48	96	160	192
7	56	112	187	224
8	64	128	214	256
9	72	144	240	288
A	80	160	267	320
B	88	176	294	352
C	96	192	320	384
D	104	208	347	416
E	112	224	373	448
F	120	240	400	480
Note: Delay time in milliseconds				

Figure 18. HUT Parameter Definitions

The following figure shows the stepping rate selected as determined by the SRT parameter and the transfer rate.

SRT Parameter	Steeping Rate/Transfer Rate			
	1000 kbps	500 kbps	300 kbps	250 kbps
0	8.0	16.0	26.7	32.0
1	7.5	15.0	25.0	30.0
2	7.0	14.0	23.3	28.0
3	6.5	13.0	21.7	26.0
4	6.0	12.0	20.0	24.0
5	5.5	11.0	18.3	22.0
6	5.0	10.0	16.7	20.0
7	4.5	9.0	15.0	18.0
8	4.0	8.0	13.3	16.0
9	3.5	7.0	11.7	14.0
A	3.0	6.0	10.0	12.0
B	2.5	5.0	8.33	10.0
C	2.0	4.0	6.67	8.0
D	1.5	3.0	5.00	6.0
E	1.0	2.0	3.33	4.0
F	0.5	1.0	1.67	2.0

Note: Delay time in milliseconds

Figure 19. SRT Parameter Definitions

The following commands are issued to the controller by the system.

Configure Command

Command Phase

	7	6	5	4	3	2	1	0
Byte 0	0	0	0	1	0	0	1	1
Byte 1	0	0	0	0	0	0	0	0
Byte 2	0	EIS	EFF	PD	Threshold (THR)			
Byte 3	Precompensation Track Number (PTN)							

Figure 20. Configure Command

Result Phase: This command has no result phase.

Dumpreg Command

Command Phase

	7	6	5	4	3	2	1	0
Byte 0	0	0	0	0	1	1	1	0

Figure 21. Dumpreg Command

Result Phase

	7	6	5	4	3	2	1	0
Byte 0	Present Track Number – Drive 0							
Byte 1	Present Track Number – Drive 1							
Byte 2	Present Track Number – Drive 2							
Byte 3	Present Track Number – Drive 3							
Byte 4	Stepping Rate Time				Head Unload Time			
Byte 5	Head Load Time							ND
Byte 6	Number of Sectors per Track/End of Track							
Byte 7	X	0	Reserved			X	GAP	WGATE
Byte 8	0	EIS	EFF	PD	Threshold			
Byte 9	Precompensation Track Number							

Figure 22. Dumpreg Result

Format Track Command

Command Phase

	7	6	5	4	3	2	1	0
Byte 0	0	MFM	0	0	1	1	0	1
Byte 1	0	0	0	0	0	HD	US	US
Byte 2	Number of Data Bytes in Sector							
Byte 3	Sectors per Track							
Byte 4	Gap Length							
Byte 5	Fill Byte							

Figure 23. Format Track Command

Result Phase

Byte 0	Status Register 0
Byte 1	Status Register 1
Byte 2	Status Register 2
Byte 3	Reserved
Byte 4	Reserved
Byte 5	Reserved
Byte 6	Reserved

Figure 24. Format Track Result

Lock Command

Command Phase

	7	6	5	4	3	2	1	0
Byte 0	Lock	0	0	1	0	1	0	0

Figure 25. Lock Command

Result Phase

Byte 0	0	0	0	Lock	0	0	0	0

Figure 26. Lock Result

Perpendicular Mode Command

Command Phase

	7	6	5	4	3	2	1	0
Byte 0	0	0	1	0	0	0	1	0
Byte 1	OW	0	D3	D2	D1	D0	GAP	WGATE

Figure 27. Perpendicular Mode Command

Read Data Command

Command Phase

	7	6	5	4	3	2	1	0
Byte 0	MT	MFM	SK	0	0	1	1	0
Byte 1	0	0	0	0	0	HD	US	US
Byte 2	Track Number							
Byte 3	Head Address							
Byte 4	Sector Number							
Byte 5	Number of Data Bytes in Sector							
Byte 6	End-of-Track							
Byte 7	Gap Length							
Byte 8	Data Length							

Figure 28. Read Data Command

Result Phase

Byte 0	Status Register 0
Byte 1	Status Register 1
Byte 2	Status Register 2
Byte 3	Track Number
Byte 4	Head Address
Byte 5	Sector Number
Byte 6	Number of Data Bytes in Sector

Figure 29. Read Data Result

Read Deleted Data Command

Command Phase

	7	6	5	4	3	2	1	0
Byte 0	MT	MFM	SK	0	1	1	0	0
Byte 1	0	0	0	0	0	HD	US	US
Byte 2	Track Number							
Byte 3	Head Address							
Byte 4	Sector Number							
Byte 5	Number of Data Bytes in Sector							
Byte 6	End-of-Track							
Byte 7	Gap Length							
Byte 8	Data Length							

Figure 30. Read Deleted Data Command

Result Phase

Byte 0	Status Register 0
Byte 1	Status Register 1
Byte 2	Status Register 2
Byte 3	Track Number
Byte 4	Head Address
Byte 5	Sector Number
Byte 6	Number of Data Bytes In Sector

Figure 31. Read Deleted Data Result

Read ID Command

Command Phase

	7	6	5	4	3	2	1	0
Byte 0	0	MFM	0	0	1	0	1	0
Byte 1	0	0	0	0	0	HD	US	US

Figure 32. Read ID Command

Result Phase

Byte 0	Status Register 0
Byte 1	Status Register 1
Byte 2	Status Register 2
Byte 3	Track Number
Byte 4	Head Address
Byte 5	Sector Number
Byte 6	Number of Data Bytes In Sector

Figure 33. Read ID Result

Read Track Command

Command Phase

	7	6	5	4	3	2	1	0
Byte 0	0	MFM	0	0	0	0	1	0
Byte 1	0	0	0	0	0	HD	US	US
Byte 2	Track Number							
Byte 3	Head Address							
Byte 4	Sector Number							
Byte 5	Number of Data Bytes in Sector							
Byte 6	End-of-Track							
Byte 7	Gap Length							
Byte 8	Data Length							

Figure 34. Read Track Command

Result Phase

Byte 0	Status Register 0
Byte 1	Status Register 1
Byte 2	Status Register 2
Byte 3	Track Number
Byte 4	Head Address
Byte 5	Sector Number
Byte 6	Number of Data Bytes in Sector

Figure 35. Read Track Result

Recalibrate Command

Command Phase

	7	6	5	4	3	2	1	0
Byte 0	0	0	0	0	0	1	1	1
Byte 1	0	0	0	0	0	0	US	US

Figure 36. Recalibrate Command

Result Phase: This command has no result phase.

Relative Seek Command

Command Phase

	7	6	5	4	3	2	1	0
Byte 0	1	DIR	0	0	1	1	1	1
Byte 1	0	0	0	0	0	HD	US	US
Byte 2	Relative Cylinder Number							

Figure 37. Relative Seek Command

Result Phase: This command has no result phase.

Scan Equal Command

Command Phase

	7	6	5	4	3	2	1	0
Byte 0	MT	MFM	SK	1	0	0	0	1
Byte 1	0	0	0	0	0	HD	US	US
Byte 2	Track Number							
Byte 3	Head Address							
Byte 4	Sector Number							
Byte 5	Number of Data Bytes in Sector							
Byte 6	End-of-Track							
Byte 7	Gap Length							
Byte 8	Scan Test							

Figure 38. Scan Equal Command

Result Phase

Byte 0	Status Register 0
Byte 1	Status Register 1
Byte 2	Status Register 2
Byte 3	Track Number
Byte 4	Head Address
Byte 5	Sector Number
Byte 6	Number of Data Bytes in Sector

Figure 39. Scan Equal Result

Scan High or Equal Command

Command Phase

	7	6	5	4	3	2	1	0
Byte 0	MT	MFM	SK	1	1	1	0	1
Byte 1	0	0	0	0	0	HD	US	US
Byte 2	Track Number							
Byte 3	Head Address							
Byte 4	Sector Number							
Byte 5	Number of Data Bytes in Sector							
Byte 6	End-of-Track							
Byte 7	Gap Length							
Byte 8	Scan Test							

Figure 40. Scan High or Equal Command

Result Phase

Byte 0	Status Register 0
Byte 1	Status Register 1
Byte 2	Status Register 2
Byte 3	Track Number
Byte 4	Head Address
Byte 5	Sector Number
Byte 6	Number of Data Bytes in Sector

Figure 41. Scan High or Equal Result

Scan Low or Equal Command

Command Phase

	7	6	5	4	3	2	1	0
Byte 0	MT	MFM	SK	1	1	0	0	1
Byte 1	0	0	0	0	0	HD	US	US
Byte 2	Track Number							
Byte 3	Head Address							
Byte 4	Sector Number							
Byte 5	Number of Data Bytes in Sector							
Byte 6	End-of-Track							
Byte 7	Gap Length							
Byte 8	Scan Test							

Figure 42. Scan Low or Equal Command

Result Phase

Byte 0	Status Register 0
Byte 1	Status Register 1
Byte 2	Status Register 2
Byte 3	Track Number
Byte 4	Head Address
Byte 5	Sector Number
Byte 6	Number of Data Bytes in Sector

Figure 43. Scan Low or Equal Result

Seek Command

Command Phase

	7	6	5	4	3	2	1	0
Byte 0	0	0	0	0	1	1	1	1
Byte 1	0	0	0	0	0	HD	US	US
Byte 2	New Track Number after Seek							

Figure 44. Seek Command

Result Phase: This command has no result phase.

Sense Drive Status Command

Command Phase

	7	6	5	4	3	2	1	0
Byte 0	0	0	0	0	0	1	0	0
Byte 1	0	0	0	0	0	HD	US	US

Figure 45. Sense Drive Status Command

Result Phase

Byte 0	Status Register 3

Figure 46. Sense Drive Status Result

Sense Interrupt Status Command

Command Phase

	7	6	5	4	3	2	1	0
Byte 0	0	0	0	0	1	0	0	0

Figure 47. Sense Interrupt Status Command

Result Phase

Byte 0	Status Register 0
Byte 1	Present Track Number

Figure 48. Sense Interrupt Status Result

Specify Command

Command Phase

	7	6	5	4	3	2	1	0
Byte 0	0	0	0	0	0	0	1	1
Byte 1	Stepping Rate Time				Head Unload Time			
Byte 2	Head Load Time							ND

Figure 49. Specify Command

Result Phase: This command has no result phase.

Verify Command

Command Phase

	7	6	5	4	3	2	1	0
Byte 0	MT	MFM	SK	1	0	1	1	0
Byte 1	EC	0	0	0	0	HD	US	US
Byte 2	Cylinder Address							
Byte 3	Head Address							
Byte 4	Sector Address							
Byte 5	Sector size							
Byte 6	End-of-Track							
Byte 7	Gap Length							
Byte 8	Byte Transfer Control							

Figure 50. Verify Command

Result Phase

Byte 0	Status Register 0
Byte 1	Status Register 1
Byte 2	Status Register 2
Byte 3	Cylinder Address
Byte 4	Head Address
Byte 5	Sector Address
Byte 6	Sector size

Figure 51. Verify Result

Version Command

Command Phase

	7	6	5	4	3	2	1	0
Byte 0	0	0	0	1	0	0	0	0

Figure 52. Version Command

Result Phase

Byte 0	1	0	0	X	0	0	0	0

Note – x can be a 1 or a 0.

Figure 53. Version Result

Write Data Command

Command Phase

	7	6	5	4	3	2	1	0
Byte 0	MT	MFM	0	0	0	1	0	1
Byte 1	0	0	0	0	0	HD	US	US
Byte 2	Track Number							
Byte 3	Head Address							
Byte 4	Sector Number							
Byte 5	Number of Data Bytes in Sector							
Byte 6	End-of-Track							
Byte 7	Gap Length							
Byte 8	Data Length							

Figure 54. Write Data Command

Result Phase

Byte 0	Status Register 0
Byte 1	Status Register 1
Byte 2	Status Register 2
Byte 3	Track Number
Byte 4	Head Address
Byte 5	Sector Number
Byte 6	Number of Data Bytes in Sector

Figure 55. Write Data Result

Write Deleted Data Command

Command Phase

	7	6	5	4	3	2	1	0
Byte 0	MT	MFM	0	0	1	0	0	1
Byte 1	0	0	0	0	0	HD	US	US
Byte 2	Track Number							
Byte 3	Head Address							
Byte 4	Sector Number							
Byte 5	Number of Data Bytes in Sector							
Byte 6	End-of-Track							
Byte 7	Gap Length							
Byte 8	Data Length							

Figure 56. Write Deleted Data Command

Result Phase

Byte 0	Status Register 0
Byte 1	Status Register 1
Byte 2	Status Register 2
Byte 3	Track Number
Byte 4	Head Address
Byte 5	Sector Number
Byte 6	Number of Data Bytes in Sector

Figure 57. Write Deleted Data Result

Invalid Command Status

Result Phase: The following status byte is returned to the system microprocessor when an invalid command has been received.

Byte 0	Status Register 0 = hex 80

Figure 58. Invalid Command Result

Note: Bits 6 and 7 in Status Register 0 are used to indicate command status. When an invalid command is processed, this information is returned to the system microprocessor in the Invalid Command Status byte.

Command Status Registers

This section provides definitions for status registers 0 through 3.

Status Register 0

Bit	Function
7, 6	Interrupt Code
5	Seek End
4	Equipment Check
3	Reserved
2	Head Address
1, 0	Drive Select

Figure 59. Status Register 0

Bits 7, 6 These bits indicate the command interrupt status.

Bits 7 6	Function
0 0	Normal Termination of Command
0 1	Abnormal Termination of Command
1 0	Invalid Command Issued
1 1	Reserved

Figure 60. Status Register 0 (Bits 7, 6)

Bit 5 This bit is set to 1 when the diskette drive completes the Seek or Recalibrate command, or a read or write operation with an implied Seek command.

Bit 4 This bit is set to 1 if the '-track 0' signal fails to occur after the Recalibrate command is issued or Relative Seek command to step outward beyond track 0.

Bit 3 Reserved. This bit is always set to 0.

Bit 2 This bit indicates the state of the '-head select' signal after the command was performed. When set to 1, head 1 was selected; when set to 0, head 0 was selected.

Bit 1, 0 These bits indicate the drive that was selected upon command completion.

Bits 1 0	Function
0 0	Drive 0
0 1	Drive 1
1 0	Drive 2
1 1	Drive 3

Figure 61. Status Register 0 (Bits 1, 0)

Status Register 1

Bit	Function
7	End-of-Track
6	Reserved
5	Cyclic Redundancy Check (CRC) Error
4	Overrun/Underrun Error
3	Reserved
2	No Data
1	Not Writable
0	Missing Address Mark

Figure 62. Status Register 1

Bit 7 This bit is set to 1 when the controller tries to gain access to a sector beyond the final sector of a track.

Bit 6 Reserved. This bit is always set to 0.

Bit 5 This bit is set to 1 when a CRC error is detected in the ID or data field.

Bit 4 This bit is set to 1 if the system does not service the diskette drive controller within an adequate period of time during data transfers.

Bit 3 Reserved. This bit is always set to 0.

Bit 2 This bit is set to 1 when:

- The controller cannot find the sector specified in the ID register during the execution of a Read Data, Read Deleted Data, or Read ID or Read Track command.
- The controller cannot read the ID field without an error during the execution of a Read ID command
- The starting sector cannot be found during the execution of a Read Track command.

Bit 1 This bit is set to 1 when the '-write-protect' signal is active during a Write Data, Write Deleted Data, or Format Track command.

Bit 0 This bit is set to 1 if the controller cannot detect an address mark. When this occurs, bit 0 of Status Register 2 indicates whether the missing address mark is an ID-address mark or a data-address mark.

Status Register 2

Bit	Function
7	Reserved
6	Control Mark
5	CRC Error in Data Field
4	Wrong Track
3	Scan Equal
2	Scan Not Satisfied
1	Bad Track
0	Missing Address Mark in Data Field

Figure 63. Status Register 2

Bit 7 Reserved. This bit is always set to 0.

Bit 6 This bit is set to 1 when the controller encounters a sector that has a deleted data-address mark during a Read Data or a Read Deleted Data encounters a data address mark.

Bit 5 This bit is set to 1 if the controller detects an error in the data.

Bit 4 This bit is set to 1 when the track number on the media is different from the track number issued by the command. When this occurs, bit 2 of Status Register 1 is also set to 1.

Bit 3 This bit is set to 1 during the Scan command when the conditions for "Equal" are satisfied.

Bit 2 This bit is set to 1 during the Scan Command when the scan conditions are not satisfied.

Bit 1 This bit is set to 1 when the track number on the media is hex FF and the track number value stored in the ID register is not hex FF. When this occurs, bit 2 of Status Register 1 is also set to 1.

Bit 0 This bit is set to 1 when the controller cannot find a data-address mark. This bit is set to 0 when an ID-address mark cannot be found. Bit 0 in Status Register 0 is also set if either address mark cannot be found.

Status Register 3

Bit	Function
7	Reserved
6	Write Protect
5	Reserved
4	Track 0
3	Reserved
2	Head Address
1, 0	Drive Select

Figure 64. Status Register 3

Bit 7 Reserved. This bit is always set to 0.

Bit 6 This bit indicates the status of the '-write-protect' signal from the diskette drive. When this bit is set to 1, the '-write-protect' signal is active.

Bit 5 Reserved. This bit is always set to 1.

Bit 4 This bit indicates the status of the '-track 0' signal from the diskette drive. When this bit is set to 1, the '-track 0' signal is active.

Bit 3 Reserved. This bit is always set to 1.

Bit 2 This bit indicates the status of the '-head 1 select' signal from the diskette drive. When this bit is set to 1, the '-head 1 select' signal is active.

Bits 1, 0 These bits indicate the current selected drive.

Interface Signal Descriptions

The following section describes the interface signals to the diskette drive.

Output Signals

All output signals to the diskette drive operate between 5 V dc and ground, with the following definitions:

- The inactive level is 2.0 V dc minimum.
- The active level is 0.8 V dc maximum.

-HIGH DENSITY SELECT: When this signal is active, the 2MB mode is selected. Diskettes are formatted with 18 sectors per track and a capacity of 1.44MB. When this signal is inactive, the 1MB mode is selected. Diskettes are formatted with 9 sectors per track and a capacity of 720KB.

DATA RATE SELECT 0 - 1: These signals are driven by the system to select the data rate of devices on the diskette drive interface. These signals are controlled by the data-rate-select bits (see Figure 12 on page 9).

-DRIVE SELECT: The drive-select signal enables or disables all drive interface signals except -MOTOR ENABLE. When the drive select signal is active, the drive is enabled. When it is inactive, all controlled inputs are ignored and all drive outputs are disabled.

-MOTOR ENABLE: When this signal is made active, the spindle starts to turn. There must be a 500-millisecond minimum delay after -MOTOR ENABLE becomes active before a read, write, or seek operation is initiated (750 milliseconds for 5.25-inch drive types). When inactive, this signal causes the spindle motor to decelerate and stop.

-DIRECTION IN: When this signal is active, -STEP moves the heads toward the drive spindle. When this signal is inactive, -STEP moves the heads away from the drive spindle. This signal is stable for at least 1 microsecond before and after the trailing edge of the -STEP pulse.

Note: After a direction change, a 15-millisecond minimum delay is required before the next '-step' pulse is issued.

-STEP: A 1-microsecond active pulse of this signal causes the read/write heads to move one track. The state of -DIRECTION at the trailing edge of -STEP determines the direction of motion.

Note: A 15-millisecond seek settle time must be provided after the last step pulse occurs before a read, write, or seek operation initiated.

-WRITE DATA: A 125-nanosecond minimum pulse of this signal causes a flux reversal to occur on the media if -WRITE ENABLE is active.

-WRITE ENABLE: When active, this signal enables the write current circuits and -WRITE DATA controls the writing of information. Motor-start and head-settle times must be observed before this signal becomes active.

-HEAD 1 SELECT: When active, this signal selects the upper head; when inactive, the lower head is selected.

Input Signals

All input signals from the drive can sink 4.0 milliamperes at the active level.

- The inactive level is 3.7 V dc minimum.
- The active level is 0.4 V dc maximum.

DRIVE TYPE ID 0 - 1: These signals provide encoded drive-type information to the system when the drive is selected.

MEDIA TYPE ID 0 - 1: These signals provide encoded media-type information to the system when the drive is selected.

-INDEX: When the drive senses the index, it generates an active pulse of at least 1 millisecond on this line.

-TRACK 0: This signal is active when the read/write head is on track 0. Track 0 is determined by a sensor, not a track counter.

The drive can seek to track 0, under control of the system even if no diskette is installed. This allows system software to determine how many drives there are attached to the system. Software selects each drive and attempts to recalibrate that drive to track 0. The track 0 indication determines whether or not each drive is installed in the system.

-WRITE PROTECT: When this signal is active the drive cannot write data to the diskette.

-READ DATA: Each flux reversal detected on the media provides a 125-nanosecond minimum active pulse on this line.

-DISKETTE CHANGE: This signal is active at power-on and latched inactive when a diskette is present, the drive is selected and a step pulse occurs. This signal goes active when the diskette is removed from the drive. The presence of a diskette is determined by a sensor.

Connector

Signals and voltages are transferred between the system board and the diskette drives by a cable or printed-circuit board. The printed-circuit board provides a 2- by 20-pin card edge connector for each diskette drive, with a locator key between pins 34 and 36. The cable interface provides a 2 x 17 pin header connector to each diskette drive, with a locator key below pin 17.

The following figures show the signals and DC voltages for each diskette drive connector type:

Pin	Signal	Pin	Signal
1	Ground	2	-High Density Select
3	Reserved	4	Reserved
5	Ground	6	Reserved
7	Ground	8	-Index
9	Ground	10	Reserved
11	Ground	12	-Drive Select
13	Ground	14	Reserved
15	Ground	16	-Motor Enable
17	Ground	18	-Direction In
19	Ground	20	-Step
21	Ground	22	-Write Data
23	Ground	24	-Write Enable
25	Ground	26	-Track 0
27	Ground	28	-Write Protect
29	Ground	30	-Read Data
31	Ground	32	-Head 1 Select
33	Ground	34	-Diskette Change
35	Ground	36	Ground
37	Ground	38	+5 V dc
39	Ground	40	+12 V dc

Figure 65. Diskette Drive Connector Signal Assignments For the 40-pin Card Edge Interface

Pin	Signal	Pin	Signal
1	Signal Return	2	-High Density Select
3	+5 V dc	4	Drive Type ID 1
5	Signal Return	6	+12 V dc
7	Signal Return	8	-Index
9	Signal Return	10	Reserved
11	Signal Return	12	-Drive Select
13	Signal Return	14	Reserved
15	Signal Return	16	-Motor Enable
17	Signal Return	18	-Direction In
19	Signal Return	20	-Step
21	Signal Return	22	-Write Data
23	Signal Return	24	-Write Enable
25	Signal Return	26	-Track 0
27	Signal Return	28	-Write Protect
29	Signal Return	30	-Read Data
31	Signal Return	32	-Head 1 Select
33	Signal Return	34	-Diskette Change

Figure 66. Diskette Drive Connector Signal Assignments For the 34-pin Header Interface

Pin	Signal	Pin	Signal
1	Signal Return	2	Data Rate Select 1
3	+5 V dc	4	Drive Type ID 1
5	Signal Return	6	+12 V dc
7	Signal Return	8	-Index
9	Drive Type ID 0	10	Reserved
11	Signal Return	12	-Drive Select
13	Signal Return	14	Reserved
15	Signal Return	16	-Motor Enable
17	Media Type ID 1	18	-Direction In
19	Signal Return	20	-Step
21	Signal Return	22	-Write Data
23	Signal Return	24	-Write Enable
25	Signal Return	26	-Track 0
27	Media Type ID 0	28	-Write Protect
29	Signal Return	30	-Read Data
31	Signal Return	32	-Head 1 Select
33	Data Rate Select 0	34	-Diskette Change

Figure 67. Diskette Drive Connector Signal Assignments For the Enhanced 34-Pin Header Interface

Index

C

clock rate 2
command phase 10
command status registers 28
command summary 10, 11
command symbols 11
commands 10
  format track 16
  invalid command status 27
  lock 17
  perpendicular mode 17
  read data 18
  read deleted data 18
  read id 19
  read track 20
  recalibrate 20
  scan equal 21
  scan high or equal 22
  scan low or equal 22
  sense drive status 23
  sense interrupt status 24
  specify 24
  write data 26
  write deleted data 26
configure command 15
connector 35
connector voltages, diskette drive 35
controller registers 3

D

data rate control register 10
data register 9
data width conversions 2
digital input register 9
digital output 5
digital output register 5
diskette command symbols 11
diskette drive connector 35
diskette drive controller 10
diskette drive controller status register 7
drive status register 5
dumpreg command 16

E

execution phase 10

F

FIFO compatibility 3
format track command 16

I

input signals 34
interface signal descriptions 32
invalid command status 27

L

lock command 17

M

multibyte transfer 10

O

output signals 32
output, digital 5

P

perpendicular mode command 17
precompensation select register 8
programming considerations 10

R

read data command 18
read deleted data command 18

© Copyright IBM Corp. 1991

read id command 19
read track command 20
recalibrate command 20
registers
  data 9
  data rate control 10
  digital input 9
  digital output 5
  diskette drive controller
    status 7
  status register A 4
  status register B 4
  status register 0 28
  status register 1 29
  status register 2 30
  status register 3 31
registers, command status 28
registers, controller 3
relative seek command 21
reserved register 9
result phase 10

S

scan equal command 21
scan high or equal command 22
scan low or equal command 22
seek 23
seek command 23
sense drive status command 23
sense interrupt status
  command 24
specify command 24
status register 7
status register A 4
status register B 4
status register 0 28
status register 1 29
status register 2 30
status register 3 31
switching density 2

V

verify command 25

W

write data command 25, 26
write deleted data command 26

Keyboard ans Auxiliary Device Controller

Description 1
Keyboard Password 2
Controller Status Register 4
  Input and Output Buffers 5
  Input and Output Ports 6
  Controller Commands 7
Keyboard and Auxiliary Device Programming Considerations . . 12
Auxiliary Device and System Timings 13
  System Receiving Data 13
  System Sending Data 14
Signals 16
Connector 16

© Copyright IBM Corp. 1990

Figures

1. Controller Status Register, Read Port Hex 0064 4
2. Input Port Definitions 6
3. Output Port Definitions 7
4. Controller Command Byte 8
5. Bit Definitions of Auxiliary-Device Data Stream 13
6. Receiving Data Timings 14
7. Sending Data Timings 15
8. Keyboard and Auxiliary Device Signals 16
9. Keyboard and Auxiliary Device Connector Information . . . 16

© Copyright IBM Corp. 1990

Description

Input to the keyboard and auxiliary device controllers is through two connectors at the rear of the system unit. One connector is dedicated to the keyboard, the other is available for an auxiliary device. An auxiliary device can be any type of serial input device compatible with the controller interface. The device types include:

- Mouse
- Touchpad
- Trackball.

Both Type 1 and Type 2 controllers receive the serial data, check the parity, and present the data to the system as a byte of data at data port hex 0060. The Type 2 controller provides only 7 of the 32 internal addresses available on the Type 1, and two commands are not available on the Type 2. Also, the Type 1 controller can translate keyboard scan codes, and the Type 2 cannot. (To determine which controller is present, see the description for bit 6 of the Controller Command byte on page 8.)

Note: Because the Type 1 controller supports translation and the Type 2 does not, the keyboard must provide both scan set 1 and scan set 2.

The controller interrupts the system when data is available or waits for polling from the system microprocessor.

Address hex 0064 is the command/status port. When the system reads port hex 0064, it receives status information from the controller. When the system writes to the port, the controller interprets the byte as a command.

Secondary circuit protection is provided on the system board for the +5 Vdc line to the keyboard and auxiliary device.

Keyboard Password

Legal password characters are restricted to the 128 ASCII character set. The password can be up to seven bytes long but must be installed using the keyboard scan codes less than hex 80 (the controller does not compare keyboard scan codes greater than hex 7F). Scan code set 1 is recommended for all password operations.

While the password is enabled, the controller compares the incoming keyboard scan code against the installed password, and it discards all data from the keyboard and auxiliary device that does not match the password. After a match occurs, the controller is restored to normal operation and data is again passed to the system microprocessor.

The controller provides three commands for keyboard password operation; it does not provide a command to verify the installed password.

A4 Test Password Installed
A5 Load Password
A6 Enable Password.

The Test Password Installed command determines if a keyboard password is currently installed. The controlling program can use this command to determine if a keyboard password is loaded before enabling the password.

The Load Password command allows the system microprocessor to set a keyboard password in the controller at any time. The existing password is lost, and the new password becomes active. The keyboard password can be changed at any time and must be installed in scan-code format.

The Enable Password command places the controller in the secure mode. While in the secure mode, the controller intercepts the keyboard data stream and continuously compares it to the installed password pattern. The controller does not pass any information to the system microprocessor or accept any commands while the keyboard password is enabled. (To enable the 'address 20' signal while the password is enabled, use System Control Port A at address hex 0092.)

The controller provides four internal RAM locations to support the keyboard password: addresses hex 13, 14, 16, and 17. See "Controller Commands" on page 7 for the commands on reading and writing to these locations.

Hex Address	Description
13	Security on: If this byte is nonzero when the password is enabled, the controller loads this byte into the output buffer and generates a system interrupt.
14	Security off: If this byte is nonzero when the password is matched, the controller loads this byte into the output buffer and generates a system interrupt.
16 and 17	Make 1 and 2: While the password is enabled, the controller first compares the incoming scan code against these two bytes. If the incoming scan code matches one of these two bytes, the scan code is discarded before it is compared to the password. This allows the controller to ignore certain keystrokes such as that for the right and left shift keys.

Controller Status Register

The following table shows the Controller Status register.

Bit	Function
7	Parity Error
6	General Time-Out
5	Auxiliary Device Output Buffer Full
4	Inhibit Switch
3	Command/Data
2	System Flag
1	Input Buffer Full
0	Output Buffer Full

Figure 1. Controller Status Register, Read Port Hex 0064

Note: On the Type 1 controller, commands C1 and C2 place data in bits 7 through 4 of the Controller Status register. See commands C1 and C2 on page 10 for more information.

Bit 7 When set to 0, this bit indicates that the last byte of data received from the keyboard had odd parity. When a parity error occurs, this bit is set to 1 and hex FF is placed in the output buffer (the keyboard and auxiliary device use odd parity).

Bit 6 When set to 1, this bit indicates that a transmission was started by the keyboard but did not finish within the receive time-out delay, or that a transmission was started by the controller but the byte transmitted was not clocked out within the specified time limit.

When a receive time-out occurs, the controller places a hex FF in the output buffer. When a transmit time-out occurs, the controller places a hex FE in the output buffer.

The Type 1 controller also indicates a transmit time-out if a transmission was started and:

- The byte was clocked out, but a response was not received within the time limit (only this bit is set to 1).
- The byte was clocked out, but a response indicates a parity error (this bit and bit 7 are both set to 1).

Bit 5 This bit works in conjunction with bit 0. When this bit and bit 0 are set to 1, auxiliary device data is in the output buffer. When this bit is set to 0 and bit 0 is set to 1, keyboard or command controller response data is in the output buffer.

Bit 4 When set to 0, this bit indicates the password state is active and the keyboard is inhibited. When set to 1, this bit indicates the password state is inactive and the keyboard is not inhibited. See "Keyboard Password" on page 2 for more information.

Bit 3 The keyboard controller input buffer can be addressed as either address hex 0060 or 0064. Address hex 0060 is defined as the data port, and address hex 0064 is defined as the command/status port. Writing to address hex 0064 sets this bit to 1. The controller uses this bit to determine if the byte in its input buffer should be interpreted as a command byte or a data byte.

Bit 2 This bit is set to 0 or 1 by writing to the system flag bit (bit 2) in the Controller Command byte. This bit is set to 0 after a power-on reset.

Bit 1 When set to 1, this bit indicates that data has been written into the buffer, but the controller has not read the data. When the controller reads the input buffer, this bit returns to 0, indicating that the input buffer is empty.

On the Type 2 controller, this bit is also set to 1 while transmitting to the keyboard or auxiliary device. After the last bit is sent, this bit is reset to 0.

Bit 0 When set to 1, this bit indicates the controller has placed data into its output buffer but the system microprocessor has not yet read the data. When the system microprocessor reads the output buffer (address hex 0060), this bit returns to 0.

Input and Output Buffers

The output buffer is an 8-bit, read-only register at address hex 0060. When the output buffer is read, the controller sends information to the system microprocessor. The information can be keyboard scan codes, auxiliary device data, or data bytes from a controller command.

The input buffer is an 8-bit, write-only register at address hex 0060 or address hex 0064. When the input buffer is written to, the

input-buffer-full bit (bit 1) in the Controller Status Byte is set to 1. Data written to the input buffer through address hex 0064 is interpreted as a controller command. Data written to address hex 0060 is sent to the keyboard, unless the controller expects a data byte following a controller command. Bit 3 of the Controller Status register indicates whether the contents of the input buffer is a command or a data byte.

Note: Data should be written to the controller input buffer only if the input-buffer-full bit (bit 1) in the Controller Status register (address hex 0064) is 0.

Input and Output Ports

The input port consists of two signals driven to the controller by the keyboard and auxiliary device. The output port consists of eight signals driven by the controller to the keyboard, auxiliary device, or system interface. The following tables show the input port and the output port bytes.

Bit	Function
7 - 2	Reserved
1	Auxiliary Data In
0	Keyboard Data In

Figure 2. Input Port Definitions

Bit 7 - 2 Reserved.

Bit 1 This bit reflects the state of the 'data' line driven by the auxiliary device. For more information on the auxiliary device 'data' line, see "Auxiliary Device and System Timings" on page 13.

Bit 0 This bit reflects the state of the 'data' line driven by the keyboard.

Bit	Function
7	Keyboard Data Out
6	Keyboard Clock Out
5	IRQ12
4	IRQ01
3	Auxiliary Clock Out
2	Auxiliary Data Out
1	Gate Address Line 20
0	Reset Microprocessor

Figure 3. Output Port Definitions

Bit 7 This bit reflects the state of the 'data' line driven by the controller to the keyboard.

Bit 6 This bit reflects the state of the 'clock' line driven by the controller to the keyboard.

Bit 5 When set to 1, this bit indicates an interrupt has been generated by data from the auxiliary device in the output buffer. When the system reads the data from address hex 0060, this bit will be set to 0.

Bit 4 When set to 1, this bit indicates an interrupt has been generated by data from the keyboard or a command in the output buffer. When the system reads the data from address hex 0060, this bit will be set to 0.

Bit 3 This bit reflects the state of the 'clock' line driven by the controller to the auxiliary device.

Bit 2 This bit reflects the state of the 'data' line driven by the controller to the auxiliary device.

Bit 1 When this bit and bit 1 in port hex 0092 are set to 0, the system address line A20 is disabled and set to 0. This bit is set to 1 at power-on. (See "System Control Port A" in the system-specific technical reference.)

Bit 0 When set to 0, this bit resets the system microprocessor and holds it reset until the bit is set to 1.

Controller Commands

A command is a data byte written to the controller through address hex 0064. The commands are listed in order of their hex values; commands not listed are reserved.

20 - 3F Read Controller RAM: This command causes the controller to return the data contained in the internal address specified by bits 5 through 0 of this command. Internal address hex 00 is assigned as the Controller Command byte. The Type 2 controller supports only the internal addresses hex 00, 13, 14, 16, 17, 1D, and 1F.

Command hex 20 requests a read operation of the Controller Command byte. The controller returns the data to port hex 0060.

Bit	Function
7	Reserved
6	Keyboard Translate
5	Disable Auxiliary Device
4	Disable Keyboard
3	Reserved
2	System Flag
1	Enable Auxiliary Interrupt
0	Enable Keyboard Interrupt

Figure 4. Controller Command Byte

Bit 7 This bit is reserved.

Bit 6 When this bit is set to 1, the Type 1 controller translates the incoming keyboard scan codes to scan set 1. When this bit is set to 0, the controller passes the incoming scan codes without translation. Following power-on or a keyboard reset, the keyboard transmits using scan code set 2.

On the Type 2 controller, this bit cannot be set to 1; therefore, it can be used to determine the type of controller. Writing this bit as a 1 and reading it as a 0 indicates a Type 2 controller.

For keyboard operations that are compatible with IBM Personal Computers, the Type 1 controller is placed in the translate mode. To perform the same operations with the Type 2 controller, the keyboard is set up to transmit in scan code set 1 by using the Select Alternate Scan Codes command (see the Keyboard section for more information).

Note: For Type 1 controllers, this bit must be set to 0 while requesting the keyboard for its scan set. This prevents the controller from translating the keyboard response.

Bit 5 Setting this bit to 1 disables the auxiliary device interface by driving the 'clock' line low. Data is not received while the interface is disabled.

Bit 4 Setting this bit to 1 disables the keyboard interface by driving the 'clock' line low. Data is not received while the interface is disabled.

Bit 3 This bit is reserved.

Bit 2 The value written to this bit is placed in the system flag bit of the Controller Status register.

Bit 1 Setting this bit to 1 causes the controller to generate an interrupt (IRQ 12) when it places auxiliary device data into its output buffer.

Bit 0 Setting this bit to 1 causes the controller to generate an interrupt (IRQ 1) when it places keyboard or command controller response data into its output buffer.

60 - 7F Write to Controller RAM: Bits 5 through 0 of the command specify the address. Internal address hex 00 is assigned as the Controller Command byte. The Type 2 controller supports only the internal addresses hex 00, 13, 14, 16, 17, 1D, and 1F.

Command hex 0060 writes the Controller Command byte. The next byte of data written to address hex 0060 is placed in the Controller Command byte.

Warning: On the Type 2 controller, writing to unsupported internal addresses can cause the data to be transmitted to the keyboard.

A4 Test Password Installed: This command checks for a password currently installed in the controller. The test result is placed in the output buffer (address hex 0060 and IRQ 01). Hex FA means the password is installed; hex F1 means it is not installed.

A5 Load Password: This command initiates the Password Load procedure. Following this command, the controller takes input from the data port until a null (0) is detected. The null terminates password entry. (See "Keyboard Password" on page 2.)

A6 Enable Password: This command enables the controller password feature. This command is valid only when a password pattern is currently loaded in the controller. (See "Keyboard Password" on page 2.)

A7 Disable Auxiliary Device Interface: This command sets bit 5 of the Controller Command byte to 1. This disables the auxiliary device interface by driving the 'clock' line low. Data is not received while the interface is disabled.

A8 Enable Auxiliary Device Interface: This command sets bit 5 of the Controller Command byte to 0, releasing the auxiliary device interface.

AD Disable Keyboard Interface: This command sets bit 4 of the Controller Command byte to 1. This disables the keyboard interface by driving the 'clock' line low. Data is not received while the interface is disabled.

AE Enable Keyboard Interface: This command clears bit 4 of the Controller Command byte to 0, releasing the keyboard interface.

C0 Read Input Port: This command causes the controller to read its input port and place the data in its output buffer.

C1 Poll Input Port Low: This command causes the Type 1 controller to read its input port bits 3 through 0 and place them in bits 7 through 4 of the Controller Status register. This command is not supported by the Type 2 controller.

C2 Poll Input Port High: This command causes the Type 1 controller to read its input port bits 7 through 4 and place them in bits 7 through 4 of the Controller Status register. This command is not supported by the Type 2 controller.

D0 Read Output Port: This command causes the controller to read its output port and place the data in its output buffer. This command should be used only if the output buffer is empty.

D1 Write Output Port: The next byte of data written to address hex 0060 is placed in the controller output port. For this command, the Type 2 controller supports writing to only bit 1 (gate A20).

Note: For Type 1 controllers, bit 0 of the output port is connected to the 'reset' signal of the system microprocessor. Pulsing this bit resets the system.

D2 Write Keyboard Output Buffer: The next byte written to address hex 0060 input buffer is written to address hex 0060 output buffer as if initiated by the keyboard. An interrupt occurs if the interrupt is enabled.

D3 Write Auxiliary Device Output Buffer: The next byte written to address hex 0060 input buffer is written to address hex 0060 output buffer as if initiated by an auxiliary device. An interrupt occurs if the interrupt is enabled.

D4 Write to Auxiliary Device: The next byte written to address hex 0060 input buffer is transmitted to the auxiliary device.

E0 Read Test Inputs: This command causes the controller to read its test inputs and place the results in the output buffer. Test 0 (T0) is connected to the keyboard 'clock' line, and test 1 (T1) is connected to the auxiliary device 'clock' line. Data bit 0 represents T0, and data bit 1 represents T1.

F0 - FF Pulse Output Port: This command pulses selected bits in the controller output port for approximately 6 microseconds. Bits 3 through 0 select the respective bits in the controller output port. For example, when bit 0 of this command is set to 0, bit 0 of the output port is pulsed and the system microprocessor is reset.

Note: The only command supported for the Type 2 controller is hex FE, pulse bit 0.

Keyboard and Auxiliary Device Programming Considerations

The following are some programming considerations for the keyboard and auxiliary device controller:

- Address hex 0064 (Controller Status register) can be read at any time.
- The auxiliary-device-output-buffer-full bit in the Controller Status register indicates that the data in address hex 0060 came from the auxiliary device.
- Address hex 0060 and address hex 0064 should be written to only when the input-buffer-full bit and output-buffer-full bit in the Controller Status register are set to 0.
- To ensure that the buffer data is valid, disable the keyboard and auxiliary devices before initiating a command that causes the controller to generate output at port 60 (such as commands D1 and D3).
- When polling the Type 1 controller for the output-buffer-full condition, wait 7 microseconds from the buffer-full indication in the Controller Status register before reading the output buffer.

Auxiliary Device and System Timings

Data transmissions to and from the auxiliary device connector consist of an 11-bit data stream sent serially over the 'data' line. The following table shows the function of each bit.

Bit	Function
11	Stop bit (always 1)
10	Parity Bit (odd parity)
9	Data Bit 7 (most-significant)
8	Data Bit 6
7	Data Bit 5
6	Data Bit 4
5	Data Bit 3
4	Data Bit 2
3	Data Bit 1
2	Data Bit 0 (least-significant)
1	Start Bit (always 0)

Figure 5. Bit Definitions of Auxiliary-Device Data Stream

The parity bit is either 1 or 0, and the 8 data bits, plus the parity bit, always have an odd number of 1's.

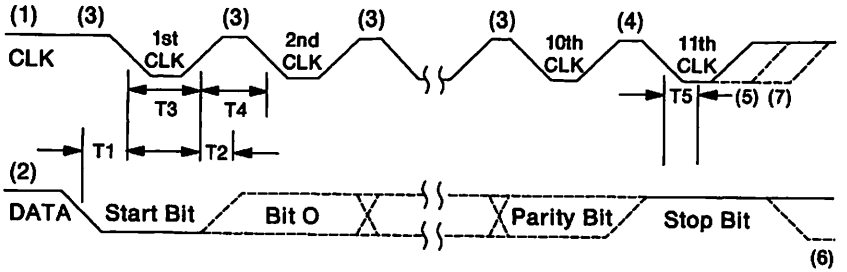
System Receiving Data

The following describes the typical sequence of events when the system is receiving data from the auxiliary device.

1. The auxiliary device checks the 'clock' line. If the line is inactive, output from the device is not allowed.
2. The auxiliary device checks the 'data' line. If the line is inactive, the controller receives data from the system.
3. The auxiliary device checks the 'clock' line during the transmission at intervals not exceeding 100 microseconds. If the device finds the system holding the 'clock' line inactive, the transmission is terminated. The system can terminate transmission anytime during the first 10 clock cycles.
4. A final check for terminated transmission is performed at least 5 microseconds after the 10th clock.
5. The system can hold the 'clock' signal inactive to inhibit the next transmission.
6. The system can set the 'data' line inactive if it has a byte to transmit to the device. The 'data' line is set inactive when the start bit (always 0) is placed on the 'data' line.

7. The system raises the 'clock' line to allow the next transmission.

The timing relationship is shown in the following figure.



	Timing Parameter	Min/Max
T1	Time from DATA transition to falling edge of CLK	5/25 μs
T2	Time from rising edge of CLK to DATA transition	5/T4 - 5 μs
T3	Duration of CLK inactive	30/50 μs
T4	Duration of CLK active	30/50 μs
T5	Time to auxiliary device inhibit after clock 11 to ensure the auxiliary device does not start another transmission	>0/50 μs

Figure 6. Receiving Data Timings

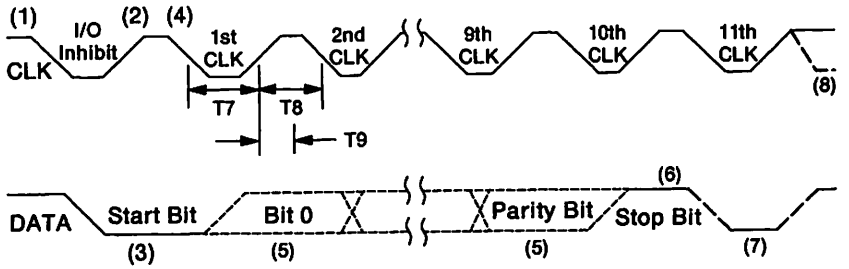
System Sending Data

The following describes the typical sequence of events when the system is sending data to the auxiliary device.

1. The system checks for an auxiliary device transmission in process. If a transmission is in process and beyond the 10th clock, the system must receive the data.
2. The auxiliary device checks the 'clock' line. If the line is inactive, an I/O operation is not allowed.
3. The auxiliary device checks the 'data' line. If the line is inactive, the system has data to transmit. The 'data' line is set inactive when the start bit (always 0) is placed on the 'data' line.
4. The auxiliary device sets the 'clock' line inactive. The system then places the first bit on the 'data' line. Each time the auxiliary device sets the 'clock' line inactive, the system places the next bit on the 'data' line until all bits are transmitted.
5. The auxiliary device samples the 'data' line for each bit while the 'clock' line is active. Data must be stable within 1 microsecond after the rising edge of the 'clock' line.

6. The auxiliary device checks for a positive-level stop bit after the 10th clock. If the 'data' line is inactive, the auxiliary device continues to clock until the 'data' line becomes active. Then it clocks the line-control bit and, at the next opportunity, sends a Resend command to the system.
7. The auxiliary device pulls the 'data' line inactive, producing the line-control bit.
8. The system can pull the 'clock' line inactive, inhibiting the auxiliary device.

The timing relationship is shown in the following figure.



	Timing Parameter	Min/Max
T7	Duration of CLK inactive	30/50 μs
T8	Duration of CLK active	30/50 μs
T9	Time from inactive to active CLK transition, used to time when the auxiliary device samples DATA	5/25 μs

Figure 7. Sending Data Timings

Signals

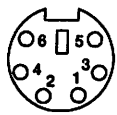
The keyboard and auxiliary device signals are driven by open-collector drivers pulled to 5 Vdc through a pull-up resistor. The following lists the characteristics of the signals.

Sink Current	20 mA	Maximum
High-Level Output Voltage	5.0 Vdc minus pullup	Minimum
Low-Level Output Voltage	0.5 Vdc	Maximum
High-Level Input Voltage	2.0 Vdc	Minimum
Low-Level Input Voltage	0.8 Vdc	Maximum

Figure 8. Keyboard and Auxiliary Device Signals

Connector

The keyboard and auxiliary device connectors use 6-pin miniature DIN connectors. The signals and voltages are the same for both connectors and are assigned as shown in the following table.



Pin	I/O	Signal Name
1	I/O	Data
2	NA	Reserved
3	NA	Ground
4	NA	+5 Vdc
5	I/O	Clock
6	NA	Reserved

Figure 9. Keyboard and Auxiliary Device Connector Information

Serial Port Controller

| Description 1
| Communications Application 3
| Programmable Baud-Rate Generator 4
| Modem Status Interrupts 4
| FIFO Modes of Operation 5
| Interrupt Mode 5
| Polled Mode 6
| DMA Modes of Operation 7
| Receive Mode 8
| Transmit Mode 8
| Received Data Status Register 8
| Transmit Commands 9
| Modem Pacing 10
| Character Orientated Pacing 10
| Receive Character Count Register 10
| Byte Pacing 10
| Enhanced Interrupts 11
| Serial Port Controller Programming Considerations 11
| Registers 11
| Compatible Registers 12
| Transmitter Holding Register (Base c+0) 13
| Receiver Buffer Register (Base c+0) 14
| Divisor Latch Register (Base c+1) 14
| Divisor Latch Register (Base c+0) 15
| Interrupt Enable Register (Base c+1) 17
| FIFO Control Register (Base c+2) 17
| Interrupt Identification Register (Base c+2) 19
| Line Control Register (Base c+3) 21
| Modem Control Register (Base c+4) 23
| Line Status Register (Base c+5) 24
| Modem Status Register (Base c+6) 26
| Scratch Register (Base c+7) 27
| Enhanced Registers 28
| Enhanced Command Register (Base e+0) 28
| Reserved Register (Base e+1) 29
| Enhanced Interrupt ID Register (Base e+2) 30
| Enhanced Function Register 1 (Base e+3) 31
| Enhanced Function Register 2 (Base e+4) 33
| Enhanced Function Register 3 (Base e+5) 35
| Character Compare Data Register (Base e+6) 36
| Receive Character Count Register (Base e+7) 37
| Signal Descriptions 38

© Copyright IBM Corp. 1990

| Modem-Control Input Signals 38
| -Clear to Send (-CTS) 38
| -Data Set Ready (-DSR) 38
| -Ring Indicator (-RI) 38
| -Data Carrier Detect (-DCD) 38
| Modem-Control Output Signals 38
| -Data Terminal Ready (-DTR) 38
| -Request to Send (-RTS) 39
| Voltage Interchange Information 39
| Extended Performance Requirements 39
| Connectors 41

Index 43

Figures

| 1. Serial Port Controller Block Diagram 2
| 2. Serial Port Register - Base Addresses 3
| 3. Serial Port Data Format 3
| 4. Received Data Status Register 8
| 5. Serial Port Register - Base Addresses 12
| 6. Serial Port Compatible Register Address Offsets 12
| 7. Serial Port Enhanced Register Address Offsets 12
| 8. Transmitter Holding Register (Base c+0) 13
| 9. Receiver Buffer Register (Base c+0) 14
| 10. Divisor Latch Register, Low Byte (Base c+1) 14
| 11. Divisor Latch Register, Low Byte (Base c+0) 15
| 12. Divisor Latch Register, High Byte (Base c+1) 15
| 13. Baud Rates at 1.8432 MHz (Low Frequency Mode) 16
| 14. Baud Rates at 11.0592 MHz (High Frequency Mode) 16
| 15. Interrupt Enable Register (Base c+1) 17
| 16. FIFO Control Register (Base c+2) 18
| 17. Trigger Level 18
| 18. Interrupt Identification Register (Base c+2) 19
| 19. Type controllers for Bits 7 and 6 19
| 20. Interrupt Control Functions 20
| 21. Line Control Register (Base c+3) 21
| 22. Stop Bits and Word Length 22
| 23. Modem Control Register (Base c+4) 23
| 24. Line Status Register (Base c+5) 24
| 25. Modem Status Register (Base c+6) 26
| 26. Enhanced Command Register 28
| 27. Command Decode 28
| 28. Enhanced Interrupt ID Register 30
| 29. Enhanced Interrupt Control Functions 30
| 30. Enhanced Function Register 1 (Base e+3) 31
| 31. Enhanced Function Register 2 (Base e+4) 33
| 32. Enhanced Function Register 3 (Base e+3) 35
| 33. Access to CCRs and CCFRs 36
| 34. Character Compare Register (Base e+6) 36
| 35. Voltage Levels 39
| 36. Serial Port Connector Signal and Pin Assignments (25-Pin) 41
| 37. Serial Port Connector Signal and Pin Assignments (9-Pin) . 42

© Copyright IBM Corp. 1990

Description

The serial port controller is programmable and supports asynchronous communications. The controller automatically adds and removes start, stop, and parity bits. A programmable baud-rate generator allows operation from 50 baud to 345.6KB. The controller supports 5-, 6-, 7-, and 8-bit characters with 1, 1.5, or 2 stop bits. A prioritized interrupt system controls transmit, receive, error, line status, and data-set interrupts.

The serial port controller provides the following functions:

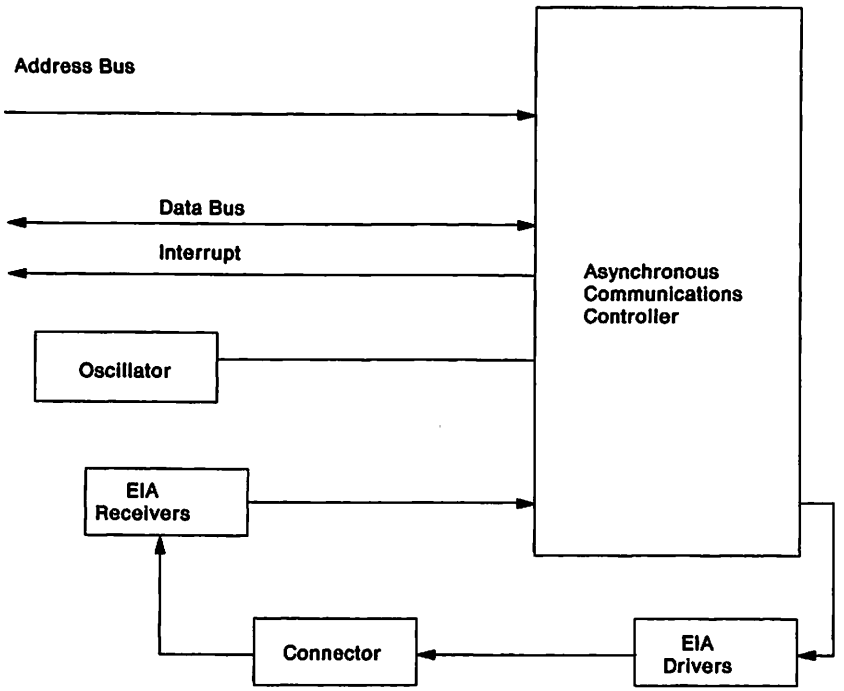
- Full double buffering in the character mode, eliminating the need for precise synchronization
- False-start bit detection
- Line-break generation and detection
- Modem control functions:
  - Clear to send (CTS)
  - Request to send (RTS)
  - Data set ready (DSR)
  - Data terminal ready (DTR)
  - Ring indicator (RI)
  - Data carrier detect (DCD).

Three types of serial port controllers have been used on the system boards. To programs, the Type 1 controller appears to be identical to the serial portion of the IBM Personal Computer AT IBM Personal Computer Serial/Parallel Adapter. The Type 2 controller incorporates all functions of the Type 1 and also provides support of the first-in-first-out (FIFO) mode. The Type 3 controller incorporates all functions of the Type 2 controller and provides the Direct Memory Access (DMA) mode.

Note: Some systems using the Type 2 controller do not support the FIFO mode. For information about individual systems refer to the system-specific technical reference manuals.

Support for the Type 1 controller is restricted to the functions that are identical to the NS16450. Using the Type 1 controller in the FIFO mode can result in nondetectable data errors. See "Registers" on page 11 for detailed FIFO information.

| The following figure is a block diagram of the serial port controller.



| *Figure 1. Serial Port Controller Block Diagram*

Communications Application

For Type 1 and Type 2 controllers the serial port can be addressed as either serial port 1 (Serial 1) or serial port 2 (Serial 2). For Type 3 controllers, the serial port can be addressed as any one of eight serial ports (Serial 1 through Serial 8). The following table illustrates the base addresses and their corresponding serial ports:

Serial Port	Compatible (Hex)	Enhanced (Hex)
Serial 1	03F8	83F8
Serial 2	02F8	82F8
Serial 3	3220	B220
Serial 4	3228	B228
Serial 5	4220	C220
Serial 6	4228	C228
Serial 7	5220	D220
Serial 8	5228	D228

Figure 2. Serial Port Register - Base Addresses

Type 1 and Type 2 controllers have only compatible registers. Type 3 controllers have both compatible and enhanced registers. In this section, serial port register addresses will contain a ***base c*** or a ***base e*** to signify the compatible or enhanced register base addresses, followed by an offset to be added to the base address to get the effective address of the register. The register assignments are controlled by Programmable Option Select (POS) and are made during system board setup.

Two interrupt lines are provided to the system. For Type 1 and Type 2 controllers, interrupt level 4 (IRQ4) is for Serial 1 and interrupt level 3 (IRQ3) is for Serial 2. For Type 3 controllers, there is no restriction. Either of the interrupt levels (IRQ3 and IRQ4) can be assigned to any of the eight serial ports. For the serial port controller to send interrupts to the interrupt controller, bit 3 of the Modem Control register must be set to 1.

The data format is shown in the following figure.

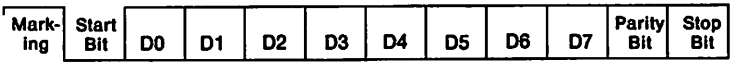


Figure 3. Serial Port Data Format

Data bit 0 (D0)is the first bit to be sent or received. The controller automatically inserts the start bit, the correct parity bit (if

| programmed to do so), and the stop bits (1, 1.5, or 2 depending on the
| command in the Line Control register).

Programmable Baud-Rate Generator

| The controller has a programmable baud-rate generator that can
| divide the clock input (11.0592 MHz) by any divisor from 1 to 65,535.
| In compatibility mode, a 1.8432 MHz clock is used. The output
| frequency of the baud-rate generator is the baud rate multiplied by
| 16. Two 8-bit latches store the divisor in a 16-bit binary format. The
| divisor latches are loaded during setup to ensure desired operation of
| the baud-rate generator. When either of the divisor latches is loaded,
| a 16-bit baud counter is immediately loaded. This prevents long
| counts on the first load.

Modem Status Interrupts

| The modem status interrupts occur as soon as the corresponding
| input signals change state, whether the Received Data Status register
| is enabled or not. The current modem status is immediately
| available in the Modem Status register. However, the change in the
| modem status is reflected in the received-data-status character that
| follows the actual change.

| Whenever the overrun error occurs, the interrupt is generated and the
| corresponding bit in the Line Status register is set. In character
| mode, when an overrun error occurs, the character in the Receiver
| Buffer register is overwritten. In the FIFO or DMA mode, when an
| overrun error occurs, the data in the FIFO mode is preserved, and the
| character in the Receive Shift register is overwritten. When the
| Received Data Status register is enabled, the overrun error is
| indicated in the received-status-byte of the first character received
| after the last error. The indicator stays on for only one character
| regardless of the number of characters lost, unless another error
| occurs.

| The interrupts for parity error and frame error occur when the error
| character is the next one to be read from the FIFO mode in DMA,
| whether the Received Data Status register is enabled or not.
| Although these interrupts are reset by reading the Line Status
| register, the bits can cause an interrupt, but they do not identify which
| character had the error in DMA mode. The Received Data Status
| register must be enabled to determine which character had the error.

FIFO Modes of Operation

The serial port contains two register stacks of 16 bytes each. These register stacks are called FIFOs. One is the Receive FIFO and the other is the Transmit FIFO.

In the FIFO mode, the controller can operate in the interrupt mode or the polled mode. To enable the FIFO mode, set bit 0 in the FIFO Control register to 1.

Interrupt Mode

When the receiver interrupts are enabled in the Interrupt Enable register, they occur as follows:

- A received-data-available interrupt is issued to the system when the FIFO register has reached the programmed trigger level.
- The Interrupt Identification register's received-data-available condition is set when the trigger level is reached and, like the interrupt, is cleared when the register drops below the trigger level.
- The receiver-line-status interrupt has a higher priority than the received-data-available interrupt.
- Bit 0 in the Line Status register is set to 1 to indicate that a character is transferred from the Shift register to the FIFO register. It is set to 0 when the Receiver FIFO register is empty.

When the Receiver FIFO register and receiver interrupts are enabled, the following occurs:

- A FIFO time-out interrupt occurs if the following conditions exist:
  - At least 1 character is in the Receiver FIFO register.
  - The last character was received more than four continuous-character times ago (if 2 stop bits are programmed, the second one is included in this time delay).
  - The most recent system microprocessor read off the Receiver FIFO register was longer than four continuous-character times ago.

  This causes a maximum character-received to interrupt-issued delay of 160 milliseconds at 300 baud, with a 12-bit character.

- Character times are calculated by using the 'receiver clock' input for a clock signal (this makes the delay proportional to the baud rate).
- When a time-out interrupt has occurred, it is cleared, and the timer is reset when the system microprocessor reads one character from the Receiver FIFO register.
- When a time-out interrupt has not occurred, the time-out timer is reset after a new character is received, or after the system microprocessor reads the Receiver FIFO register.

When the Transmitter FIFO register and transmitter interrupts are enabled (FIFO Control register bit 0 and Interrupt Enable register bit 1 are set to 1), the following occurs:

- The transmitter-holding-register-empty interrupt (02) occurs when the Transmitter FIFO register is empty. It is cleared when the Transmitter Holding register is written to (1 to 16 characters can be written to the Transmitter FIFO register while this interrupt is being serviced), or the Interrupt Identification register is read.
- The transmitter-FIFO-register-empty indications are delayed one character time minus the last stop-bit time whenever both of the following occur:
  - Bit 5 (transmitter-holding-register-empty) of the Line Status register is set to 1.
  - There have not been at least two bytes in the Transmitter FIFO register at the same time since the last time bit 5 of the Line Status register was set to 1.

  The first transmitter interrupt after changing bit 0 in the FIFO Control register is immediate, if enabled.

Character time-out and Receiver FIFO register trigger-level interrupts have the same priority as the current received-data-available interrupt. The transmitter-FIFO-register-empty interrupt has the same priority as the current transmitter-holding-register-empty interrupt.

Polled Mode

To put the controller in the FIFO polled mode, disable the interrupts through the Interrupt Enable register and enable the FIFO mode. The Receiver and Transmitter FIFO registers are controlled separately. Either or both registers can be in the polled mode of operation.

In the FIFO-polled mode of operation, the system reads the status of the Receiver and Transmitter FIFO register through the Line Status register.

- The data-ready bit indicates whether or not the Receiver FIFO register contains data.
- The error bits indicate the type of error. Character error status is handled the same way as when in the interrupt mode. The Interrupt Identification register is not affected because bit 2 of the Interrupt Enable register is set to 0.
- Line Status register bit 5 indicates when the Transmitter FIFO register is empty.
- Line Status register bit 6 indicates that both the Transmitter FIFO register and Transmitter Shift register are empty.
- Line Status register bit 7 indicates any errors in the Receiver FIFO register.

There is no trigger level reached or time-out condition indicated in the FIFO polled mode; however, the Receiver and Transmitter FIFO registers are still fully capable of holding characters.

DMA Modes of Operation

In addition to the character mode and the FIFO mode of the Type 2 controller, the Type 3 controller supports the use of DMA for receiving and transmitting. The Type 3 controller also provides new functions and new interrupts, many of which are available in the FIFO mode.

The presence of a Type 3 controller can be detected by enabling the DMA transmit mode (bit 6 in Enhanced Function register 1), reading bits 6 and 7 of the Interrupt ID register, and then disabling the DMA transmit mode. If bit 6 is a 1 and bit 7 is a 0, then a Type 3 controller is installed. If the FIFO mode is enabled, but not the DMA transmit mode, then bits 6 and 7 in the Interrupt ID register read as 1. If character mode is enabled, then bits 6 and 7 are 0.

The DMA mode uses separate DMA channels for transmitting and receiving. In addition, the transmit and receive modes may be set independently. (Operating the receiver in DMA mode and the transmitter in FIFO mode will conserve DMA channels). This allows a high performance receiving function and requires only one interrupt per 16 characters transmitted.

Receive Mode

While in receive mode, the Type 3 controller signals the needs to transfer data when the Receive FIFO register has reached the receiver trigger level or when a timeout occurs. The data is then transferred from the controller until the FIFO register is empty or the DMA Terminal Count (TC) is reached. A timeout occurs if there is at least one character in the FIFO register and a character has not been read in the last four character times. An interrupt on the transmit terminal count and the receive terminal count occur independently.

Transmit Mode

While in the transmit mode, data is transferred until the FIFO register is full or the end of the data is reached.

Two separate terminal count interrupts are available: one for transmit and the other for receive.

Received Data Status Register

When received, a status byte can be placed in the FIFO register with each data byte. This option is available in the DMA receive mode and in the FIFO mode. The status byte is defined as the Received Data Status register and is stored after the corresponding data byte. The bit definitions of Received Data Status register are as follows:

Bit	Description
7	Data Carrier Detect
6	Clear to Send
5	Data Set Ready
4	Break
3	Framing
2	Parity Error
1	Overrun Error
0	Error/Break

Figure 4. Received Data Status Register

Bit 7 This bit indicates the state of the '-data carrier detect' signal (-DCD).

Bit 6 This bit indicates the state of the '-clear to send' signal (-CTS).

Bit 5 This bit indicates the state of the '-data send ready' signal (-DSR).

| **Bit 4** This bit determines Break (BI)

| **Bit 3** This bit determines Framing (FE)

| **Bit 2** This bit determines Parity Error (PE)

| **Bit 1** This bit determines Overrun Error (OE)

| **Bit 0** This bit determines the Error/Break. Bit 0 is set and reset.

| Bit 0 is set to 1 when an error or break occurs and remains set for subsequent characters until it is reset to 0 by a software command (write hex 03 to the Enhanced Command register). This allows scanning of the received data (in memory or as it exits from the FIFO mode) to find the character that was received with an error. While the error/break bit indicates that an error has occurred, it is not possible to distinguish multiple errors from single errors without checking the status of each character that has this bit set.

| Bit 0 is set to 1 and reset to 0 before the Received Data status register is placed in the FIFO mode. This means that up to eight characters have bit 0 set after the Reset Error/Break Indicator command is given and there are no additional errors.

| When the Received Data Status register is enabled the Receive FIFO register can hold eight data bytes and eight status bytes.

| Transmit Commands

| New transmit commands provide better software control of the transmitter and allow the insertion of special control characters such as XON and XOFF into the the transmitted data stream. The new transmit commands are available in both the DMA transmit mode and in the FIFO mode. The new transmit commands are:

- Start Sending - starts or continues transmit
- Stop Sending - stops transmit.

| To insert a character in the transmitted data stream, stop the transmitter by issuing the Stop Sending command, write a character to the Transmitter Holding register, and then start the transmitter by issuing the Start Sending command.

Modem Pacing

Modem pacing is handled by several new functions without software involvement, which are available in the DMA mode and the FIFO mode. These new functions also prevent the async port from receiving invalid data.

The transmitter and receiver are controlled by the following signals:

- The '-clear-to-send' signal
  - When this signal is equal to 0, the transmitter is turned off.
- The '-data-carrier-detect' signal
  - When this signal is equal to 0, the transmitter is turned off.
- The '-data-set-ready' signal
  - When this signal is equal to 0, the transmitter and the receiver are turned off.

Character Orientated Pacing

Three software-programmable registers are provided to handle character oriented pacing. The contents of these registers are compared to each received character. If there is a match, a preprogrammed action takes place. The possible actions on a match are interrupt, delete character, stop transmitter, and start transmitter. If an error or break occurs in a received character, it is not compared.

Receive Character Count Register

This register enables the user to keep track of the number of characters sent to the central processing unit or the DMA. The Receive Character Count Interrupt is asserted when the counter is decremented to 0, provided that bit 0 of the Enhanced Function register 1 is set.

Byte Pacing

The Byte Pacing function is useful when the Type 3 controller is communicating with a slow processor. This function enables the Type 3 controller to transmit every byte at 16 times the Receive Character Count value or 256 times the RCCR value. The result of this product is called RCLK time.

Enhanced Interrupts

Several new interrupts are available to support the DMA mode, the Receive Character Count register, and the Character Compare registers. The new interrupts are:

- Interrupt on Transmitter FIFO and transmitter-shift-register-empty
- Interrupt on Terminal Count in the DMA transmit mode
- Interrupt on Terminal Count in the DMA receive mode
- Interrupt on Receiver Character Count equals 0
- Interrupt on Character Compare Register Match.

Serial Port Controller Programming Considerations

The serial port uses either the Type 1, Type 2, or Type 3 serial communications controller. The following should be considered when programming the serial controller:

- The Type 1 serial controller does not support the FIFO mode.
- Some systems using the Type 2 controller do not support the FIFO mode. For more information, refer to the system-specific technical reference manuals.
- The Type 1 or Type 2 serial port on the system board can be configured to either Serial 1 or Serial 2 using the system configuration utilities programs. The Type 3 can be configured as Serial 1 through 8.
- Before changing the Line Control register, make sure the Transmitter Holding register is empty.

Registers

The controller has several accessible registers. These control the operations of the controller and transmit and receive data. The system programmer can gain access to or control any of the controller registers through the system microprocessor.

Compatible Registers

Type 1, Type 2 and Type 3 controllers have certain registers that are common to them all. These registers will be referred to as Compatible registers. Type 3 controllers have additional registers that Type 1 and Type 2 controllers do not have. These registers will be referred to as Enhanced registers.

Serial Port	Compatible (Hex)	Enhanced (Hex)
Serial1	03F8	83F8
Serial 2	02F8	82F8
Serial 3	3220	B220
Serial 4	3228	B228
Serial 5	4220	C220
Serial 6	4228	C228
Serial 7	5220	D220
Serial 8	5228	D228

Figure 5. Serial Port Register - Base Addresses

The bit definitions of the Interrupt Enable register, Interrupt Identification register, and Line Status register have been modified from the Type 1 controller registers. A FIFO Control register has been added to support the FIFO mode.

Note: Using the Type 1 controller in the FIFO mode can result in nondetectable data errors.

Specific registers are selected according to the figure below and the figure on the following page.

Address Offsets	R/W	Register
+ 0 *	W	Transmitter Holding Register
+ 0 *	R	Receiver Buffer Register
+ 0 *	R/W	Divisor Latch, Low Byte
+ 1 *	R/W	Divisor Latch, High Byte
+ 1 *	R/W	Interrupt Enable Register
+ 2	R	Interrupt Identification Register
+ 2	W	FIFO Control Register
+ 3	R/W	Line Control Register
+ 4	R/W	Modem Control Register
+ 5	R	Line Status Register
+ 6	R/W	Modem Status Register
+ 7	R/W	Scratch Register

Note: *The DLAB state is controlled by bit 7 of the Line Control register.

Figure 6. Serial Port Compatible Register Address Offsets

Port Address Offsets	EFR3 Bits 2 1 0	R/W	Register
+ 0	x x x	W	Enhanced Command
+ 1	x x x	R	Reserved
+ 2	x x x	R	Enhanced Interrupt Identification
+ 3	x x x	R/W	Enhanced Function 1
+ 4	x x x	R/W	Enhanced Function 2
+ 5	x x x	R/W	Enhanced Function 3
+ 5 *	0 0 0	R/W	Char Compare Function 0
+ 5 *	0 0 1	R/W	Char Compare Reg 0
+ 5 *	0 1 0	R/W	Char Compare Function 1
+ 5 *	0 1 1	R/W	Char Compare Reg 1
+ 5 *	1 0 0	R/W	Char Compare Function 2
+ 5 *	1 0 1	R/W	Char Compare Reg 2
+ 6 *	x x x	R/W	Char Compare Data Reg
+ 7	x x x	R./W	Receive Character Count

*The Char Compare Function Register (CCFR) and the Char Compare Register (CCR) are selected by writing the address to the Enhanced Function Register 3 and the data is read or written by reading or writing to the Char Compare Data Register (CCDR).

Figure 7. Serial Port Enhanced Register Address Offsets

Transmitter Holding Register (Base c + 0)

The Transmitter Holding register contains the character to be sent when the (DLAB) divisor latch access bit 1 equals 0. Bit 0 is the least-significant bit and the first bit sent serially, as shown below.

Bit	Description
7	Data Bit 7
6	Data Bit 6
5	Data Bit 5
4	Data Bit 4
3	Data Bit 3
2	Data Bit 2
1	Data Bit 1
0	Data Bit 0

Figure 8. Transmitter Holding Register (Base c+0)

Receiver Buffer Register (Base c+0)

The Receiver Buffer register contains the received character and can be accesses when the divisor-latch-access bit (DLAB) equals 0. Bit 0 is the least-significant bit and the first bit received serially, as shown in the following figure.

Bit	Description
7	Data Bit 7
6	Data Bit 6
5	Data Bit 5
4	Data Bit 4
3	Data Bit 3
2	Data Bit 2
1	Data Bit 1
0	Data Bit 0

Figure 9. Receiver Buffer Register (Base c+0)

Divisor Latch Register (Base c+1)

The Divisor Latch register is used to program the baud-rate generator. The value in this register forms the divisor of the clock input (1.8432 MHz or 11,0592MHz), which establishes the desired baud-rate (DLAB=1).

Bit	Description
7	Bit 7
6	Bit 6
5	Bit 5
4	Bit 4
3	Bit 3
2	Bit 2
1	Bit 1
0	Bit 0

Figure 10. Divisor Latch Register, Low Byte (Base c+1)

Note: If bit 6 of the Enhanced Function register 2 is set to 0, then the input clock of the baud-rate generator is 1.8432 MHz. Otherwise, the input clock is 11.0592 MHz.

Divisor Latch Register (Base c + 0)

The Divisor Latch register is used to program the baud-rate generator. The value in this register forms the divisor of the clock input (1.8432 MHz or 11,0592MHz), which establishes the desired baud-rate (DLAB = 0).

Bit	Description
7	Bit 7
6	Bit 6
5	Bit 5
4	Bit 4
3	Bit 3
2	Bit 2
1	Bit 1
0	Bit 0

Figure 11. Divisor Latch Register, Low Byte (Base c+0)

Bit	Description
7	Bit 15
6	Bit 14
5	Bit 13
4	Bit 12
3	Bit 11
2	Bit 10
1	Bit 9
0	Bit 8

Figure 12. Divisor Latch Register, High Byte (Base c+1)

Figure 13 on the following page illustrates the use of baud-rate generator with a frequency of 1.8432 MHz.

Note: Data speed should not exceed 19,200 baud (For Type 1 and Type 2).

Desired Baud Rate Rate	Divisor Used to Generate 16x Clock (Decimal)	(Hex)	Percent of Error Difference between Desired and Actual
50	2304	0900	--
75	1536	0600	--
110	1047	0417	0.026
134.5	857	0359	0.058
150	768	0300	--
300	384	0180	--
600	192	00C0	--
1200	96	0060	--
1800	64	0040	--
2000	58	003A	0.69
2400	48	0030	--
3600	32	0020	--
4800	24	0018	--
7200	16	0010	--
9600	12	000C	--
19200	6	0006	--

Figure 13. Baud Rates at 1.8432 MHz (Low Frequency Mode)

The following figure illustrates the use of the baud rate generator with a frequency of 11.0592 MHz.

Desired Baud Rate	Divisor Used to Generate 16x Clock (Decimal)	(Hex)	Percent of Error Difference between Desired and Actual
50	13824	3600	--
75	9216	2400	--
110	6284	188C	0.006
134.5	5139	1413	0.001
150	4608	1200	--
300	2304	900	--
600	1152	480	--
1200	576	240	--
1800	384	180	--
2000	346	15A	0.116
2400	288	120	--
3600	192	C0	--
4800	144	90	--
7200	96	60	--
9600	72	48	--
19200	36	24	--
31250	22	16	0.538
38400	18	12	--
57600	12	C	--
115200	6	6	--
172800	4	4	--
345600	2	2	--

Note: Divisor of 1 not supported. Data speed must not exceed 345.6Kbaud.

Figure 14. Baud Rates at 11.0592 MHz (High Frequency Mode)

Interrupt Enable Register (Base c + 1)

This 8-bit register allows the four types of controller interrupts to separately activate the 'chip interrupt output' signal. The interrupt system can be completely disabled by setting bits 0 through 3 of the Interrupt Enable register to 0. Similarly, by setting the appropriate bits of this register to 1, selected interrupts can be enabled. Disabling prevents the controller from generating the external interrupt to the system. All other system functions operate normally, including the setting of the Line Status and Modem Status registers (DLAB = 0).

Bit	Description
7 - 4	Reserved = 0
3	Modem-Status Interrupt
2	Receiver-Line-Status Interrupt
1	Transmitter-Holding-Register-Empty Interrupt
0	Received-Data-Available Interrupt (Character and FIFO Mode) and Time-Out Interrupts (FIFO Mode Only)

Figure 15. Interrupt Enable Register (Base c + 1)

Bits 7 - 4 These bits are reserved and always set to 0.

Bit 3 When set to 1, this bit enables the modem-status interrupt.

Bit 2 When set to 1, this bit enables the receiver-line-status interrupt.

Bit 1 When set to 1, this bit enables the transmitter-holding-register-empty interrupt.

Bit 0 When set to 1, this bit enables the received-data-available interrupt. In the FIFO mode, this bit also enables the time-out interrupts.

FIFO Control Register (Base c + 2)

The FIFO Control register is a write-only register at the same location as the read-only Interrupt Identification register. The FIFO Control register enables the FIFO registers, clears the FIFO registers, and sets the Receiver FIFO register trigger level.

Note: The Transmitter and Receiver FIFO registers are not accessible serial controller registers.

The contents of the FIFO Control register are shown in the following figure.

Bit	Description
7, 6	Receiver FIFO Register Trigger
5 - 3	Reserved = 0
2	Transmitter FIFO Register Reset
1	Receiver FIFO Register Reset
0	FIFO Mode Enable

Figure 16. FIFO Control Register (Base c+2)

Bits 7, 6 These bits select the trigger level for the receiver-register interrupt, as shown in the following figure.

Bits 7 6	Receiver Trigger Level
0 0	01 Byte
0 1	04 Bytes
1 0	08 Bytes
1 1	14 Bytes

Figure 17. Trigger Level

Bits 5 - 3 These bits are reserved and always set to 0.

Bit 2 When this bit is set to 1, all bytes in the Transmitter FIFO register are cleared and its counter logic is reset to 0. The Transmitter Shift register is not cleared. This bit is self-clearing.

Bit 1 When this bit is set to 1, all bytes in the Receiver FIFO register are cleared and its counter logic is reset to 0. The Transmitter Shift register is not cleared. This bit is self-clearing.

Bit 0 When this bit is set to 1, the FIFO mode is enabled. When this bit is changed, the transmit and receive (XMIT and RCV) FIFOs are cleared. When writing to any other FIFO Control register bits, this bit must be a 1.

Interrupt Identification Register (Base c + 2)

To minimize programming overhead during character mode transfers, the controller prioritizes interrupts into four levels:

- Priority 1 - Receiver-line-status
- Priority 2 - Received-data-available
- Priority 2 - Time-out (FIFO mode)
- Priority 3 - Transmitter-holding-register-empty
- Priority 4 - Modem status.

Information about a pending interrupt is stored in the Interrupt Identification register. When this register is addressed, the pending interrupt with the highest priority is held, and no other interrupts are acknowledged until the system microprocessor services that interrupt.

Bit	Description
7, 6	FIFO Registers Enabled
5, 4	Interrupt ID, Bit 4
3	Interrupt ID, Bit 2
2	Interrupt ID, Bit 1
1	Interrupt ID, Bit 0
0	Interrupt Not Pending

Figure 18. Interrupt Identification Register (Base c + 2)

Bits 7,6 Programs can determine whether a Type 1, Type 2, or Type 3 controller is present by reading these two bits when bit 0 of the FIFO Control register is set to 1. If bits 7 and 6 are set to 1, the Type 2 controller is present and FIFO support is provided. If bit 6 is set to 0, the controller is a Type 1 and the FIFO mode should not be used.

Bit 7	Bit 6	Version
0	0	Type 1 controller
1	0	N/A
0	1	Type 3 controller
1	1	Type 2 controller

Figure 19. Type controllers for Bits 7 and 6

Note: Some systems using the Type 2 controller do not support the FIFO mode. For information about individual systems refer to the system-specific technical reference manuals.

Bits 5, 4 These bits are always set to 0.

Bit 3 In the FIFO mode, this bit is set to 1, along with bit 2, to indicate that a time-out interrupt is pending. In the character mode, this bit is always set to 0.

Bits 2, 1 These two bits identify the pending interrupt with the highest priority.

Bit 0 When this bit is set to 1, no interrupt is pending and polling (if used) continues. When this bit is set to 0, an interrupt is pending, and the contents of this register can be used as a pointer to the appropriate interrupt service routine. This bit can be used in hard-wired, prioritized, or polled conditions to indicate if an interrupt is pending.

Figure 20 on page 21 illustrates the Interrupt Control functions, beginning with the highest priority and ending with the lowest priority.

Bits 5 4 3 2 1 0	Type	Cause	Interrupt Reset Control
0 0 0 1 1 0	Receiver Line Status	Overrun, Parity, or Framing Error or Break Interrupt	Read the Line Status Register
0 0 0 1 0 0	Received Data Available	Data in the Receiver Buffer or the Trigger Level Has Been Reached.	Read the Receiver Buffer Register or FIFO Register Drops Below the Trigger Level.
0 0 1 1 0 0*	Character Time-Out Indication	No Characters Have Been Removed From or Put Into the Receiver FIFO Register During the Last Four Character Times, and at Least 1 Character is in It at This Time.	Read the Receiver Buffer Register
0 0 0 0 1 0	Transmitter Holding Register Empty	Transmitter Holding Register is Empty	Read the Interrupt Identification Register or Write to Transmitter Holding Register
0 0 0 0 0 0	Modem Status	Change in Signal Status From Modem	Read the Modem Status Register

* FIFO Mode Only

Figure 20. Interrupt Control Functions

Line Control Register (Base c+3)

The format of asynchronous communications is programmed through the Line Control register.

Bit	Description
7	Divisor Latch Access Bit
6	Set Break
5	Stick Parity
4	Even Parity Select
3	Parity Enable
2	Number of Stop Bits
1	Word Length Select, Bit 1
0	Word Length Select, Bit 0

Figure 21. Line Control Register (Base c+3)

Bit 7 This bit is set to 1 to gain access to the divisor latches of the baud-rate generator. It is set to 0 to gain access to the Receiver Buffer, Transmitter Holding, or Interrupt Enable registers.

Bit 6 When this bit is set to 1, set break is enabled. The serial output is forced to the spacing state and remains there regardless of other transmitter activity. When this bit is set to 0, set break is disabled.

Bit 5 When bits 5, 4, and 3 are set to 1, the parity bit is sent and checked as a logical 0. When bits 5 and 3 are set to 1, and bit 4 is set to 0, the parity bit is sent and checked as a logical 1. If bit 5 is set to 0, stick parity is disabled.

Bit 4 When this bit and bit 3 are set to 1, an even number of logical 1s are transmitted and checked in the data word bits and parity bit. When this bit is set to 0, and bit 3 is set to 1, an odd number of logical 1s are transmitted and checked in the data word bits and parity bit.

Bit 3 When set to 1, a parity bit is generated (transmit data) or checked (receive data) between the last data-word bit and stop bit of the serial data. (The parity bit produces an even or odd number of 1s when the data-word bits and the parity bit are summed).

Bit 2 This bit, with bits 0 and 1, specifies the number of stop bits in each serial character sent or received, as shown in the following figure.

Bits 2 1 0	Number of Stop Bits	Word Length
0 0 0	1	5 Bits
0 0 1	1	6 Bits
0 1 0	1	7 Bits
0 1 1	1	8 Bits
1 0 0	1.5	5 Bits
1 0 1	2	6 Bits
1 1 0	2	7 Bits
1 1 1	2	8 Bits

Word length is specified by bits 1 and 0 in this register.

Figure 22. Stop Bits and Word Length

Bits 1, 0 These bits specify the number of bits in each serial character that is sent or received.

Modem Control Register (Base c+4)

This 8-bit register controls the data exchange with the modem, data set, or peripheral device emulating a modem.

Bit	Description
7 - 5	Reserved = 0
4	Loop Mode
3	Out 2 (IRQ Output Control)
2	Out 1
1	Request-to-Send
0	Data-Terminal-Ready

Figure 23. Modem Control Register (Base c+4)

Bits 7 - 5 These bits are reserved and always set to 0.

Bit 4 This bit provides a loopback feature for diagnostic testing of the serial port. When bit 4 is set to 1:

- Transmitter-serial-output is set to the marking state.
- Receiver-serial-input is disconnected.
- Output of the Transmitter Shift register is "looped back" to the Receiver Shift register input.

  Note: The Transmitter and Receiver Shift registers are not accessible NS16550 registers.
- The modem control inputs (CTS, DSR, DCD, AND RI) are disconnected.
- The modem control outputs (DTR, RTS, OUT 1, AND OUT 2) are internally connected to the four modem control inputs.
- The modem control output pins are forced inactive.

When the serial port is in the diagnostic mode, transmitted data is immediately received. This feature allows the system microprocessor to verify the transmit-data and receive-data paths of the serial port.

When the serial port is in the diagnostic mode, the receiver and transmitter interrupts are fully operational. The modem control interrupts are also operational, but their sources are the lower four bits of the Modem Control

register instead of the four modem control input signals. The interrupts are still controlled by the Interrupt Enable register.

Bit 3 When this bit is set to 0, the IRQ signal is disabled, the IRQ signal is always disabled.

Bit 2 This bit is not used in a normal mode. In loop mode, its status is reported to bit 6 (RI) of the Modem Status register.

Bit 1 This bit controls the '-request to send' signal (-RTS) modem control output. When this bit is set to 1, -RTS is active. When this bit is set to 0, -RTS is inactive.

Bit 0 This bit controls the '-data terminal ready' signal (-DTR) modem control output. When this bit is set to 1, -DTR is active. When this bit is set to 0, -DTR is inactive.

Line Status Register (Base c+5)

This 8-bit read-only register provides the system microprocessor with status information about the data transfer.

Note: Writing to this register can produce unpredictable results.

Bit	Description
7	Error in Receiver FIFO Register
6	Transmitter Shift Register Empty
5	Transmitter Holding Register Empty
4	Break Interrupt
3	Framing Error
2	Parity Error
1	Overrun Error
0	Data Ready

Figure 24. Line Status Register (Base c+5)

Bit 7 In FIFO mode, this bit indicates that a parity error, framing error, or break occurred. This bit is cleared when the Line Status register is read in the FIFO mode. It is set to 0 in the Character mode.

Bit 6 This bit is set to 1 to indicate the Transmitter Holding register and the Transmitter Shift register are both empty. This bit is set to 0 when either register contains a data character.

In the FIFO or DMA mode, this bit is set to 1 when the Transmitter FIFO register and the Transmitter Shift register are both empty.

Bit 5 This bit indicates that the controller is ready to accept the next character for transmission. This bit is set to 1 to indicate that a character was transferred from the Transmitter Holding register to the Transmitter Shift register. This bit is set to 0 when a character is written to the Transmitter Holding register.

This bit also causes the controller to issue an interrupt if the interrupt is enabled.

In the FIFO or DMA register, this bit is set to 1 when the Transmitter FIFO register is empty. It is set to 0 when at least one byte is written to the Transmitter FIFO register.

Bit 4 This bit is set to 1 to indicate the received data input is held in the spacing state for longer than a fullword transmission time (the total time of start bit + data bits + parity + stop bits). This bit is reset to 0 when the Line Status register is read.

When a break interrupt occurs, only one zero character is loaded into the Receiver FIFO register. The next character is loaded after the receiver serial input changes to the marking state and receives the next valid start bit.

Note: Bits 1 through 4 are the error conditions that produce a receiver-line-status interrupt whenever any of the corresponding conditions are detected and the interrupt is enabled.

Bit 3 This bit is set to 1 when the stop bit, following the last data bit or parity bit, is at a spacing level. This indicates that the received character did not have a valid stop bit (framing error). This bit is reset to 0 when the Line Status register is read.

Note: In the FIFO or DMA mode, the framing error (or parity error for bit 2) is associated with the particular character in the Receiver FIFO register that it applies to. The error is indicated to the system microprocessor when its associated character is at the top of the Receiver FIFO register.

Bit 2 This bit is set to 1 to indicate a parity error (the received character does not have the correct even or odd parity, as selected by the even-parity-select bit). This bit is reset to 0 when the Line Status register is read.

Bit 1 When set to 1, this bit indicates that data in the Receiver Buffer register was not read before the next character was transferred into the Receiver Buffer register, destroying the previous character. This bit is reset to 0 when the Line Status register is read.

If the FIFO or DMA mode data continues to fill the Receiver FIFO register beyond the trigger level, an overrun error occurs. The overrun occurs only after the Receiver FIFO register is full and the next character is completely received in the Receiver Shift register. An overrun error is indicated to the system microprocessor when it happens. The character in the Receiver Shift register is overwritten, but it is not transferred to the Receiver FIFO register.

Bit 0 This bit is the receiver data-ready indicator. It is set to 1 when a complete incoming character has been received and transferred into the Receiver Buffer register or the Receiver FIFO register. This bit is reset to 0 by reading the Receiver Buffer register or by reading all of the data in the Receiver FIFO register.

Modem Status Register (Base c + 6)

This 8-bit register is used to monitor the current state of the control lines from the modem (or external device). Also, bits 3 through 0 indicate change information.

Bit	Description
7	Data-Carrier-Detect
6	Ring Indicator
5	Data-Set-Ready
4	Clear-to-Send
3	Delta-Data-Carrier-Detect
2	Trailing Edge Ring Indicator
1	Delta-Data-Set-Ready
0	Delta-Clear-to-Send

Figure 25. Modem Status Register (Base c + 6)

Bit 7 This bit is the inverted '-data carrier detect' signal (-DCD) modem control input. If bit 4 of the Modem Control register is set to 1, this bit is equivalent to bit 3 in the Modem Control register.

Bit 6 This bit is the inverted '-ring indicator' signal (-RI) modem control input. If bit 4 of the Modem Control register is set to 1, this bit is equivalent to bit 2 in the Modem Control register.

Bit 5 This bit is the inverted '-data set ready' signal (-DSR) modem control input. If bit 4 of the Modem Control register is set to 1, this bit is equivalent to bit 0 in the Modem Control register.

Bit 4 This bit is the inverted '-clear to send' signal (-CTS) modem control input. If bit 4 of the Modem Control register is set to 1, this bit is equivalent to bit 1 in the Modem Control register.

Bit 3 When set to 1, this bit indicates that the '-data carrier detect' signal (-DCD) modem control input has changed state since the last time it was read by the system microprocessor.

Note: Whenever bit 0, 1, 2, or 3 is set to 1, a modem status interrupt is generated.

Bit 2 When set to 1, this bit indicates that the '-ring indicator' signal (-RI) modem control input has changed from an active condition to an inactive condition.

Bit 1 When set to 1, this bit indicates that the '-data set ready' signal (-DSR) modem control input has changed state since the last time it was read by the system microprocessor.

Bit 0 When set to 1, this bit indicates that the '-clear to send' signal (-CTS) modem control input has changed state since the last time it was read by the system microprocessor.

Scratch Register (Base c+7)

This register can be used by the system microprocessor as a temporary buffer or work area.

Enhanced Registers

The registers in this section are only available with the Type 3 controller. These registers are:

- Enhanced Command Register
- Enhanced Interrupt ID Register
- Enhanced Function Register 1
- Enhanced Function Register 2
- Enhanced Function Register 3
- Character Compare Data Register
- Receive Character Count Register.

Enhanced Command Register (Base e + 0)

This write-only register is used to issue the new commands: Stop Sending, Start Sending, and Reset Error/Break Indicator. A write to the register causes the command to be executed.

Bits	Description
7 - 2	Reserved
1, 0	Command Bits

Figure 26. Enhanced Command Register

Bits 7-2 These bits are reserved and always written as 0 to allow future expansion of the command bits.

Bits 1,0 These command bits (CB1-CB0) enable Stop Sending and Start Sending.

CB1	CB0	Command
0	0	Reserved
0	1	Start Sending
1	0	Stop Sending
1	1	Reset Error/Break Indicator

Figure 27. Command Decode

Start Sending - This command starts or continues transmitting in the DMA or the FIFO mode. Characters in the FIFO mode are transmitted first. This command is used in the FIFO mode to restart the

| transmitter if it has been stopped by a Stop Sending command or by a Stop on Match function. In DMA mode, the transmit DMA request (TX DMA REQ) will not be active until the Start Sending Command is issued.

Stop Sending - This command empties the Transmitter Shift register and stops transmitting, regardless of the FIFO mode. After the transmitter is stopped, a character is written to the Transmitter Holding register. The character is then loaded into the Shift register and transmitted. After the character is written to the transmitter-holding-register-empty, the interrupt can be enabled. When the interrupt occurs, it signals that the send-single-character operation is complete. If multiple characters are written to the Transmitter Holding register, with the transmitter-holding register-empty interrupt enabled, multiple interrupts occur. To ensure that all characters have been transmitted, wait for the appropriate number of transmitter-holding register-empty interrupts before issuing the next Start Sending command to resume transmitting the FIFO register. There is no interrupt associated with this command. The shift register operation is not affected by this command.

Reset Error/Break - This command resets the Error/Break bit in the received-data-status byte, which is optionally stored with received data. The Reset command affects the next status byte, which is stored in the FIFO register.

The logic for controlling the Error/Break bit is at the input of the Receive FIFO. If there is an error for the current character being received, the corresponding status byte will have the Error/Break bit set. When the command to reset the Error/Break indicator is issued, the indicator is 0 in the next received-data-status byte placed in the FIFO register, unless that byte also has an error.

Check each byte that has the Error/Break bit set and issue a Reset command every time an error is found. This procedure should minimize the overhead associated with error detection.

Reserved Register (Base e+1)

This is a reserved register.

Enhanced Interrupt ID Register (Base e + 2)

The pending interrupt is determined by decoding bits 1 through 5. For interrupt reset purposes, reading this register is equivalent to reading the Interrupt ID register.

Bits	Description
7 - 6	Reserved
5 - 1	Interrupt ID
0	Interrupt Pending

Figure 28. Enhanced Interrupt ID Register

Bits 7 - 6 These bits are reserved and are always set to 0.

Bits 5 - 1 These bits are the encoded IDs of the pending interrupt.

Bit 0 This bit indicates if an interrupt is pending. When it is a 0, an interrupt is pending, and bits 1 - 5 identify the interrupt. When it is a 1, no interrupt is pending.

The new interrupts have higher priority than the existing Type 2 controller interrupts. After a new interrupt has been enabled, interrupt priority exists. The following figure illustrates the Enhanced Interrupt Control functions, beginning with the highest priority and ending with the lowest priority.

Bits 5 4 3 2 1 0	Type	Source	Interrupt Reset Control
1 0 0 0 0 0	Receive TC	TC on DMA Receive	Read the Enhanced Interrupt Register
1 0 0 0 1 0	Transmit TC	TC on DMA Transmit	Read the Enhanced Interrupt Register
1 1 0 0 0 0	CC0 Match	Match on CC0	Read the Enhanced Interrupt ID Register
1 1 0 0 1 0	CC1 Match	Match on CC1	Read the Enhanced Interrupt ID Register
1 1 0 1 0 0	CC2 Match	Match on CC2	Read the Enhanced Interrupt ID Register
1 0 0 1 0 0	RCCR = 0	Receive Character Count	Read the Enhanced Interrupt ID Register
1 0 0 1 1 0	Transmitter Empty	THR and TSR Empty	Read the Enhanced Interrupt ID Register
0 0 0 1 1 0	Receiver Line Status	Overrun, Parity, or Framing Error or Break Interrupt	Read the Line Status Register
0 0 0 1 0 0*	Received Data Available	Data in the Receiver Buffer or the Trigger Level Has Been Reached.	Read the Receiver Buffer Register or FIFO Register Drops Below the Trigger Level.
0 0 1 1 0 0**	Character Time-Out Indication	No Characters Have Been Removed From or Put Into the Receiver FIFO Register During the Last Four Character Times	Read the Receiver Buffer Register in FIFO Mode. In DMA Mode Read EIIR to Reset.
0 0 0 0 1 0	Transmitter Holding Register Empty	Transmitter Holding Register is Empty	Read the Interrupt Identification Register or Write to Transmitter Holding Register
0 0 0 0 0 0	Modem Status	Change in Signal Status from Modem	Read the Modem Status Register

Note: *No trigger level interrupt in DMA Mode. ** FIFO and DMA Mode

Figure 29. Enhanced Interrupt Control Functions

Enhanced Function Register 1 (Base e + 3)

This register is a read and write register that enables new interrupts and DMA modes. A logical 1 enables the function and a logical 0 disables the function.

Bits	Description
7	DMA Receive
6	DMA Transmit
5	Enable Receive Data Status
4	Terminal Count Receive
3	Terminal Count Transmit
2	Stop Transmitter Line Error
1	Transmitter Empty
0	Receive Character Count Register

Figure 30. Enhanced Function Register 1 (Base e+3)

Bit 7 This bit enables the DMA receive mode. The FIFO mode must be enabled (bit 0 in the FIFO Control register) before the DMA receive mode is enabled. The FIFO trigger level can be programmed, as appropriate, by writing to bits 6 and 7 in the FIFO Control register.

Bit 6 This bit enables the DMA transmit mode. The FIFO mode must be enabled (bit 0 in the FIFO Control Register) before the DMA transmit mode is enabled. Initially, a Start Sending command must be issued to start actual transmission. TX REQ cannot be generated until a Start Sending Command is issued.

Bit 5 This bit enables alternate bytes of data followed by Received Data status, to be stored in the Receive FIFO register. When this bit is set to 1, the Receive FIFO register has a capacity of eight data bytes plus eight status bytes. If a status byte is at the bottom of the FIFO register, the Line Status register indicates a good byte regardless of the status of the associated data byte. The enhanced-received-data-status bit should be set as part of the async port initialization. Toggling this bit while receiving serial data produces undefined results.

Note: No data is received during the time the FIFO register is cleared and the received-data status bit is toggled. Resetting this bit disables the storing of received data status.

Bit 4 This bit enables an interrupt when the terminal count is reached on a DMA receive operation. This signals that the last character of the last DMA buffer has been filled and that the FIFO register can fill and overrun if new DMA buffers are not allocated.

Bit 3 This bit enables an interrupt when the terminal count is reached on a DMA transmit operation. This signals that the last character in the last DMA buffer has been read into the FIFO register.

Bit 2 When set, this bit stops the transmitter after the Shift register empties on any received line error (OE, PE, FE, BI). The received line error is detected before the character is placed in the FIFO or DMA mode. The receiver continues to function normally. The transmitter can be restarted with the Start Sending command. If an interrupt is desired, the enable-line-status interrupt bit must be set in the Interrupt Enable register.

Bit 1 This bit enables an interrupt when the Transmit Hold register and Transmit Send register are empty in Character mode, or when the FIFO register and Transmit Send register are empty in the FIFO or DMA mode. The transmitter empty bit in the Line Status register is changed from 0 to 1.

Bit 0 This bit enables an interrupt when the Receive Character Count register is decremented to zero.

Enhanced Function Register 2 (Base e + 4)

This read and write register enables the transmitter controls, the modem pacing, and the baud-rate functions. A logical 1 enables the function and a logical 0 disables the function.

Bits	Description
7	Byte Pacing
6	Set High Frequency Rate
5	Set Slow Transmit Rate
4	Set Slow Receiver Rate
3	Receive the Receiver via DSR
2	Control the Transmitter via DCD
1	Control the Transmitter via DSR
0	Control the Transmitter via CTS

Figure 31. Enhanced Function Register 2 (Base e+4)

Bit 7 This bit affects byte pacing. This function supports a slow processor, from which a Type 3 controller transmits data. When this bit is high, every byte of data is transmitted at a pacing rate of 256 times the Receive Character Count register value 'receiver clocks'. Interrupt is generated when the Receive Character Count register reaches 0 and the Enhanced Function Register One equals 1. When this bit is

low, the Receive Character Count register is used for counting receiving characters if the Receive Character Count register is loaded with a value greater than 0.

Bit 6 This bit sets the high-frequency rate. When set to 1 this bit selects a a 11.0592 frequency.

Note: Applications that select the 11.0592 MHz rate should reset this bit to 0 when exiting.

This allows higher bit rates to be selected, up to a maximum rate of 345,600 bits per second on transmit and receive. If this bit is set, then the divisor latches must be set to 2 or greater.

Bit 5 This bit sets the slow transmit rate to 1/16 of the rate programmed in the baud-rate generator.

Bit 4 This bit sets the slow receiver rate. If this bit is set, the receiver rate is set to 1/16 of the rate programmed in the baud-rate generator.

Note: Bits 4, 5, and 6 should be set as part of the async port initialization. Toggling these bits during transmit and receive can produce undefined results.

Bit 3 This bit resets the receiver by issuing the '-data set ready' signal (-DSR, bit 5 of the Modem Status register). If this function is enabled, the receiver is turned off when -DSR equals 0 (bit 5 of the Modem Status register equals 0) and is turned on when -DSR equals 1 (bit 5 of the Modem Status register equals 1). If -DSR becomes inactive, the character currently being received is discarded.

Bit 2 This bit controls the transmitter by issuing the '-data carrier detect' signal (-DCD, bit 3 of the Modem Status register). If this function is enabled, the transmitter is turned off when the -DCD equals 0 (bit 3 of the Modem Status register), and is turned on if -DCD equals 1. However, the transmitter must be initially turned on by a Start Sending command (After a Start Sending command has been issued, the transmitter is turned on and off based on the state of the '-data carrier detect' signal).

Note: If the transmitter is stopped, the Transmitter Shift register is emptied but no additional characters are loaded from the Transmitter FIFO or Transmitter Hold register.

Bit 1 This bit controls the transmitter via the '-data set ready' signal (-DSR). If this function is enabled, the transmitter is turned off when -DSR equals 0, and is turned on when -DSR equals 1. However, the transmitter must be initially turned on by a Start Sending command (After a Start Sending command has been issued the transmitter is turned on and off based on the state of the '-data set ready' signal).

Bit 0 This bit controls the transmitter via the '-clear-to-send' signal. (-CTS, bit 4 of the Modem Status register). If this function is enabled, the transmitter is turned off when -CTS equals 0, and is turned on when -CTS equals 1. However, the transmitter must be initially turned on by a Start Sending command (After a Start Sending command has been issued the transmitter is turned on and off based on the state of the 'clear-to-send' signal).

Enhanced Function Register 3 (Base e+5)

This read and write register is used to control the Character Compare registers. There are three 8-bit Character Compare registers and three 4-bit Character Compare Function registers. The Character Compare registers contain match characters and the Character Compare Function registers contain match functions. A Character Compare register and its Character Compare Function register can be programmed independently of the other Character Compare registers and character compare function registers.

To read from or write to the Character Compare registers, the address for the register must be written to Enhanced Function Register 3. Then the registers can be read from or written to, using the Character Compare register.

Bits	Description
7 - 3	Reserved
2 - 1	Character Compare Address Lines
0	Select Character Compare Register

Figure 32. Enhanced Function Register 3 (Base e+3)

Bits 7 - 3 These bits are reserved and are always set to 0.

Bits 2 - 1 These bits select the character compare address lines, the address of the character compare register, or the address of the Character Compare Function register.

Bit 0 This bit enables the Select Character Compare register. A logical 1 specifies the address in bits 1 and 2 for a Character Compare register. A logical 0 specifies the address for a Character Compare Function register.

The following table shows access to the Character Compare registers and the Character Compare Function registers for bits 2, 1 and 0.

Bits 2 1 0	Function of Character Compare Data Register	Result
0 0 0	Match Functions	R/W Character Compare Function Register 0
0 1 0	Match Functions	R/W Character Compare Function Register 1
1 0 0	Match Functions	R/W Character Compare Function Register 2
0 0 1	Match Character	R/W Character Compare Register 0
0 1 1	Match Character	R/W Character Compare Register 1
1 0 1	Match Character	R/W Character Compare Register 2

Figure 33. Access to CCRs and CCFRs

Character Compare Data Register (Base e+6)

This register is used to read and write the match character for a Character Compare register, or the match function for a Character Compare Function register. The register is specified by first writing to Enhanced Function Register 3.

On a match, the Type 3 controller can be programmed to start the transmitter, stop the transmitter, delete the matched character from the incoming data stream, or to interrupt. Multiple match functions per Character Compare Function register are supported. If START and STOP are both set, then STOP takes precedence because they are mutually exclusive. Each character compare register is compared to the received data character, and if there is a match, then the programmed action takes place.

If the Type 3 controller is programmed to interrupt on a character match, then the interrupt occurs as soon as the match is detected. To disable a character compare register, clear the corresponding Character Compare Function register. The format of the Character Compare Data register for the Character Compare Function register is shown on the following page.

Bits	Description
3	Start Transmitter on Character Match
2	Stop Transmitter on Character Match
1	Delete Character on Character Match
0	Interrupt on Character Match

Figure 34. Character Compare Register (Base e+6)

Bit 3 A1 starts the transmitter when a match occurs. The transmitter does not start unless a Start Transmitter command has been issued previously.

Bit 2 A1 stops the transmitter when a match occurs.

Bit 1 A1 causes the character to be deleted when a match occurs.

Bit 0 A1 enables an interrupt when a match occurs. This interrupt occurs as soon as there is a match with a received data character.

The format of the Character Compare Data register for a character compare register is an 8-bit match character. If the word length is less than eight bits, the match character should be right justified and any unused bits should be set to 0 when written to the Character Compare Data Register.

Receive Character Count Register (Base e+7)

This 8-bit register is used in byte pacing and receive character count functions. The Receive Character Compare register is decremented when a character is read from the Receive FIFO register during a receive character count operation, or at every 256 receiver clocks during Byte Pacing operation.

If the interrupt on Receiver Character Count is set (Enhanced Function Register 1, Bit 0), then an interrupt is generated when the Receive Character Count register is decremented to 0, regardless of any operation.

Because this register is a countdown counter and does not wrap around when reaching zero, the user must load a value to the Receive Character Count register before using it. When read, this register contains the current count (the count cannot be exact since the receiver or 'receiver clocks' cannot be stopped to read this register).

Signal Descriptions

Modem-Control Input Signals

The following are input signals from the modem or external device to the controller. Bits 7 through 4 in the Modem Status register indicate the condition of these signals. Bits 3 through 0 monitor these signals to indicate when the modem changes state.

-Clear to Send (-CTS)

When active, this signal indicates that the modem is ready for the serial port to transmit data.

-Data Set Ready (-DSR)

When active, this signal indicates that the modem or data set is ready to establish the communications link and transfer data with the controller.

-Ring Indicator (-RI)

When active, this signal indicates that the modem or data set detected a telephone ringing signal.

-Data Carrier Detect (-DCD)

When active, this signal indicates that the modem or data set detected a data carrier.

Modem-Control Output Signals

The following are controller output signals. All are set inactive by a master reset operation. These signals are controlled by bits 3 through 0 in the Modem Control register.

-Data Terminal Ready (-DTR)

When active, this signal informs the modem or data set that the controller is ready to communicate.

-Request to Send (-RTS)

When active, this signal informs the modem or data set that the controller is ready to send data.

Voltage Interchange Information

The signal is considered in the *marking* condition when the voltage on the interchange circuit, measured at the interface point, is more negative than -3 Vdc, with respect to signal ground. The signal is considered in the *spacing* condition when the voltage is more positive than +3 Vdc with respect to signal ground. The region between +3 Vdc and -3 Vdc is defined as the transition region and is considered an invalid level. Voltage that is more negative than -15 Vdc or more positive than +15 Vdc also is considered an invalid level.

Interchange Voltage	Binary State	Signal Condition	Interface Control Function
Positive Voltage	Binary 0	Spacing	On
Negative Voltage	Binary 1	Marking	Off

Figure 35. Voltage Levels

Extended Performance Requirements

Extended performance applies to Type 3 serial port controllers only. Although the serial port is compatible with EIA-232-D at speeds up to 20,000 bits per second, there are additional requirements for operating the port at speeds up to 345,600 bits per second. These requirements fall into two areas, the interconnection cable and the attached equipment.

The cable should not be longer than 20 feet when operating at extended performance speeds. Each signal wire has an individual shield and should have capacitance between 100 and 450 picofarads to the shield. All individual shields are connected to pin 7 at each end. These requirements can be met by using one IBM Personal Computer Communications Adapter Cable for 10 feet, or two connected in series for 20 feet.

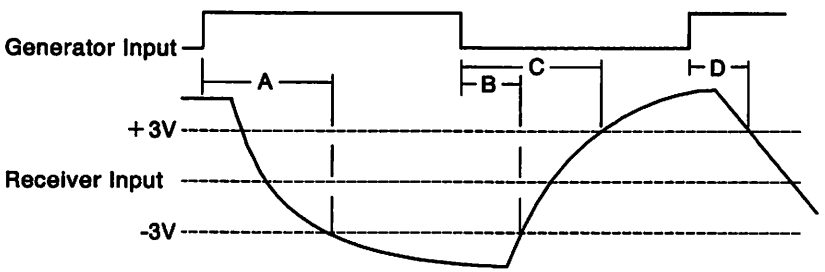
The attached equipment should meet the following requirements:

- The capacitance for an input or output signal should be less than or equal to 120 picofarads.
- The timing oscillator accuracy is ±.01%.
- The maximum generator skew equals 970 nanoseconds with the cable attached and a capacitive load (including cable) between 100 and 690 picofarads.
- The maximum receiver skew equals 160 nanoseconds.
- The receiver deserializer should decode the data stream by sampling a minimum of 16 times per unit interval such as is done in the NS8250 and similar Universal Asynchronous Receive Transmitters (UARTs).

In order to specify the maximum distortion in the signals, the concept of skew is introduced. The receiver skew is defined as the worst case difference in delay between the rising and falling edges from a 12V peak-to-peak (P-P) square wave connected to the input of the receiver to when it reaches the deserializer. For example, if the rising edge delay is 200 nanoseconds and the falling edge delay is 120 nanoseconds then the skew is 200 nanoseconds minus 120nanoseconds, which equals 80 nanoseconds. The 80 nanoseconds is well within the 160nS requirement.

The generator skew is measured using the sending serializer to generate a square wave and measuring the delays at the input of an attached receiver with the cable and all capacitive loadings (690 picofarad maximum) included.

The generator waveforms graphic shown below illustrates where the measurements are taken. The generator skew is A-B or C-D, whichever is larger.

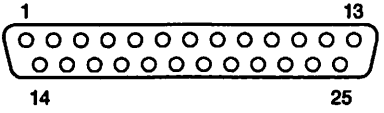


Generator Waveforms

Connectors

The hardware interface uses the standard 25-pin and 9-pin male D-shell connectors with pin assignments defined for EIA-232-D. The voltage levels are EIA-232-D only. Current loop interface is not supported.

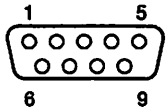
Figures 36 and 37 show the pin configurations and signal assignments for the serial port.



Pin No.	Signal Name	Pin No.	Signal Name
1	Not Connected	14	Not Connected
2	Transmit Data	15	Not Connected
3	Receive Data	16	Not Connected
4	Request to Send	17	Not Connected
5	Clear to Send	18	Not Connected
6	Data Set Ready	19	Not Connected
7	Signal Ground	20	Data Terminal Ready
8	Data Carrier Detect	21	Not Connected
9	Not Connected	22	Ring Indicator
10	Not Connected	23	Not Connected
11	Not Connected	24	Not Connected
12	Reserved	25	Not Connected
13	Not Connected		

Figure 36. Serial Port Connector Signal and Pin Assignments (25-Pin)

Figures 38 below shows the pin configuration and signal assignments for the serial port.



Pin No.	Signal Name
1	Data Carrier Detect
2	Receive Data
3	Transmit Data
4	Data Terminal Read
5	Signal Ground
6	Data Set Ready
7	Request To Send
8	Clear To Send
9	Ring Indicator

Figure 37. Serial Port Connector Signal and Pin Assignments (9-Pin)

Index

A
applications 3
asynchronous communications 1

B
baud-rate generator 4, 16
block diagram 2
break interrupt 25
byte pacing 10

C
character oriented pacing 10
communications applications 3
compatible registers 12
connector 41
considerations, programming 11

D
data format 3
data speed 16
descriptions, signal 37
divisor latch access bit (DLAB) 22
divisor latch registers 14
DLAB (divisor latch access bit) 12, 22
DMA modes 7
  receive 8
  transmit 8

E
enhanced registers 28
  command register 28
  compare data register 36
  function register 1 31
  function register 2 33
  function register 3 35
  interrupt ID register 30
  receive character count 37

F
FIFO control register 17
FIFO modes 5
  interrupt mode 5
  polled mode 6
framing error 25

I
interface 41
interrupt enable register 17
interrupt identification register 19
interrupts 17

L
line control register 21
line status register 24

M
marking condition, signal 39
modem control register 23
modem pacing 10
modem status interrupts 4
modem status register 26

O
output port 1 (Serial 1) 3
output port 2 (Serial 2) 3
overrun error 26

P
parity error 26
pin assignments 41
programming considerations 11

© Copyright IBM Corp. 1990

R

receiver buffer register 14

S

scratch register 27
signal descriptions 37
  input signals 38
  output signals 38
spacing condition, signal 39
support 1

T

transition region, signal 39
transmit commands 9
transmitter holding register 13

V

voltage interchange information 39
voltages 41

Parallel Port Controller (Type 1)

Description 1
Parallel Port Programmable Option Select 2
  Parallel Port Extended Mode 2
Parallel Port Controller Programming Considerations 2
  Data Port 3
  Status Port 4
  Parallel Control Port 5
Parallel Port Timing 6
Signal Descriptions 7
Connector 8

Figures

1. Parallel Port Controller 1
2. Parallel Port Address Assignments 2
3. Data Port 3
4. Status Port 4
5. Parallel Control Port 5
6. Parallel-Port Timing Sequence 6
7. Data and Interrupt Signals 7
8. Control Signals 7
9. Parallel Port Connector Signal and Pin Assignments 8

Description

The parallel port allows the attachment of devices that transfer 8 bits of parallel data at standard transistor-transistor levels. It has a 25-pin, D-shell connector. The primary function of the parallel port is to attach a printer with a parallel interface to the system.

This port may be addressed as parallel port 1, 2, or 3, and is compatible with IBM Personal Computer parallel port implementations. The parallel port has an extended mode that supports bidirectional input and output. The port also supports level-sensitive interrupts and a readable interrupt-pending status.

The following figure is a block diagram of the parallel port controller.

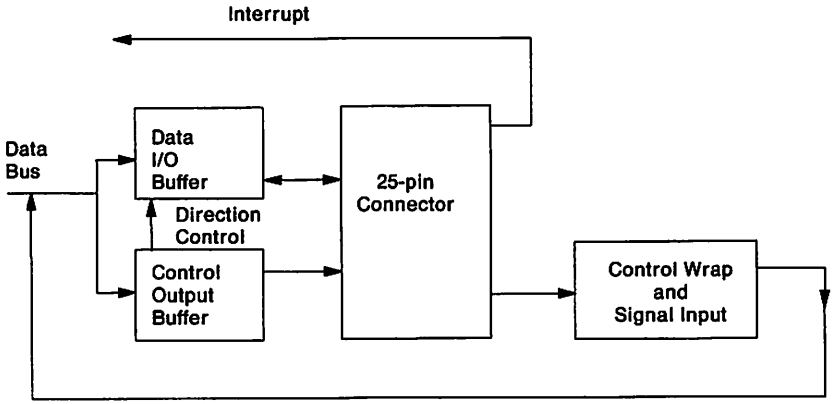


Figure 1. Parallel Port Controller

Parallel Port Programmable Option Select

The parallel port can be configured to the same three address spaces used by IBM Personal Computer products. These addresses are selected through Programmable Option Select (POS) registers during system board setup.

The address assignments for each configuration are shown in the following figure.

	Data Port (Hex Address)	Status Port (Hex Address)	Control Port (Hex Address)	Reserved Address	IRQ
Parallel 1	03BC	03BD	03BE	03BF	7
Parallel 2	0378	0379	037A	037B	7
Parallel 3	0278	0279	027A	027B	7

Figure 2. Parallel Port Address Assignments

Parallel Port Extended Mode

The extended mode option is selected through the POS function during system board setup. The extended mode makes the parallel port an 8-bit parallel bidirectional interface. Direction is determined by bit 5 of the Parallel Control port. See "Parallel Control Port" on page 5.

Parallel Port Controller Programming Considerations

The following are some considerations for programming the parallel port controller.

The interface responds to five I/O instructions: two output and three input. In the compatible mode, the output instructions transfer data into two latches whose output is presented on the pins of the D-shell connector. In the extended mode, the 8-bit data latch output to the D-shell connector is controlled by bit 5 in the Parallel Control port.

In the compatible mode, two of the three input instructions allow the system microprocessor to read back the contents of the two latches.

In the extended mode, the read-back of the 8-bit data in the Data Address is controlled by bit 5 in the Parallel Control port. The third input instruction allows the system microprocessor to read the real-time status of a group of pins on the connector.

The extended mode can be used by externally attached equipment.

During the power-on self test (POST), the parallel port is configured as an output port. POST status information is written to this port during the power-on initialization or the initialization caused by a reset from the keyboard (Ctrl, Alt, Del).

The following is a detailed description of each interface-port instruction.

Data Port

The Data port is the 8-bit data port for both the compatible and extended modes. In the compatible mode, a write operation to this port immediately presents data to the connector pins. In the compatible mode, a read operation from this port produces the last data written to it.

In the extended mode, a write operation to this port latches the data, but the data is only presented to the connector pins if the direction bit was set to 0 (Write) in the Parallel Control port. A read operation in the extended mode produces either:

- The data previously written if the direction bit in the Parallel Control port is set to 0 (Write).
- The data on the connector pins from another device if the direction bit in the Parallel Control port is set to 1 (Read).

Bit	Port Data
7 - 0	Data

Figure 3. Data Port

Bits 7 - 0 These bits represent the 'data' (D7 - D0) signal lines.

Status Port

The Status port is a read-only port in both modes. A read operation to this port presents the system microprocessor with the interrupt-pending status of the interface, and the real-time status of the connector pins, as shown in the following figure. An interrupt is pending when bit 2 (-IRQ STATUS) is set to 0.

Bit	Port Data
7	-BUSY
6	-ACK
5	PE
4	SLCT
3	-ERROR
2	-IRQ STATUS
1, 0	Reserved

Figure 4. Status Port

Bit 7 This bit represents the state of the '-busy' signal (-BUSY). When this signal is active, the printer is busy and cannot accept data.

Bit 6 This bit represents the current state of the printer '-acknowledge' signal (-ACK) . When this bit is set to 0, the printer has received a character and is ready to accept another.

Bit 5 This bit represents the current state of the printer 'paper end' signal (PE). When this bit is set to 1, the printer has detected the end of the paper.

Bit 4 This bit represents the current state of the 'select' signal (SLCT). When this bit is set to 1, the printer has been selected.

Bit 3 This bit represents the current state of the printer '-error' signal (-ERROR). When this bit is set to 0, the printer has encountered an error condition.

Bit 2 When this bit is set to 0, the printer has acknowledged the previous transfer using the '-acknowledge' signal. An interrupt is pending when bit 2 (-IRQ STATUS) is set to 0.

Bits 1, 0 These bits are reserved.

Parallel Control Port

The Parallel Control port is a read/write port. A write operation to this port latches bits 0 through 5 of the bus. Bit 5 is the direction control bit used in the extended mode only. A read operation to the Parallel Control port presents the system microprocessor the data that was last written to it, except for the write-only direction bit.

Bit	Port Data
7, 6	Reserved = 0
5	Direction
4	IRQ EN
3	Pin 17 (SLCT IN)
2	Pin 16 (-INIT)
1	Pin 14 (AUTO FD XT)
0	Pin 1 (STROBE)

Figure 5. Parallel Control Port

Bits 7, 6 These bits are reserved and must be set to 0.

Bit 5 This write-only bit controls the direction of the data port. When this bit is set to 0, the data port is written to. When this bit is set to 1, the data port is read from.

Bit 4 This bit enables the parallel port interrupt. When this bit is set to 1, an interrupt occurs when the '-acknowledge' signal changes from active to inactive.

Bit 3 This bit controls the 'select in' signal (SLCT IN). When this bit is set to 1, the printer is selected.

Bit 2 This bit controls the 'initialize printer' signal (-INIT). When this bit is set to 0, the printer starts.

Bit 1 This bit controls the 'automatic feed XT' signal (AUTO FD XT). When this bit is set to 1, the printer automatically spaces the paper up one line for every line return.

Bit 0 This bit controls the 'strobe' signal (STROBE) to the printer. When this bit is set to 1, data is clocked into the printer.

Parallel Port Timing

Timing for the parallel port depends on the devices connected to the port. The following figure shows the sequence for typical parallel-port signal timing.

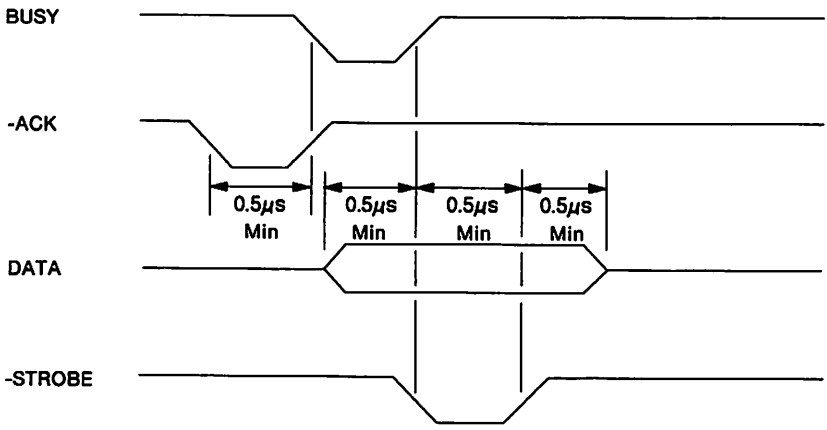


Figure 6. Parallel-Port Timing Sequence

For specific signal timing parameters, refer to the specifications for the equipment connected to the parallel port connector.

Signal Descriptions

The following figures list characteristics of the signals.

Sink Current	24 mA	Maximum
Source Current	15 mA	Maximum
High-Level Output Voltage	2.4 Vdc	Minimum
Low-Level Output Voltage	0.5 Vdc	Maximum

Figure 7. Data and Interrupt Signals

Pins 1, 14, 16, and 17 are driven by open collector drivers pulled to 5 Vdc through 4.7 kilohm resistors.

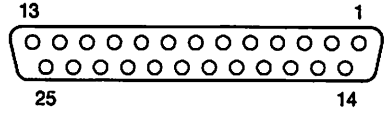
Sink Current	20 mA	Maximum
Source Current	0.55 mA	Maximum
High-Level Output Voltage	5.0 Vdc minus pullup	Minimum
Low-Level Output Voltage	0.5 Vdc	Maximum

Figure 8. Control Signals

Connector

The parallel port connector is a standard 25-pin, D-shell connector. The data lines on the connector are driven by drivers capable of sourcing 15 milliamps and sinking 24 milliamps.

The following figure shows the signal and pin assignments for the parallel port controller connector.



Pin No.	I/O	Signal Name	Pin No.	I/O	Signal Name
1	I/O	-STROBE	14	O	-AUTO FD XT
2	I/O	Data 0	15	I	-ERROR
3	I/O	Data 1	16	O	-INIT
4	I/O	Data 2	17	O	-SLCT IN
5	I/O	Data 3	18	NA	Ground
6	I/O	Data 4	19	NA	Ground
7	I/O	Data 5	20	NA	Ground
8	I/O	Data 6	21	NA	Ground
9	I/O	Data 7	22	NA	Ground
10	I	-ACK	23	NA	Ground
11	I	BUSY	24	NA	Ground
12	I	PE	25	NA	Ground
13	I	SLCT			

Figure 9. Parallel Port Connector Signal and Pin Assignments

Video Subsystem

Video Subsystem 1
Type 1 1
Type 2 1

VGA Function 2
Major Components 4
ROM BIOS 4
Support Logic 4
VGA Components 4
CRT Controller 4
Sequencer 5
Graphics Controller 5
Attribute Controller 7
Hardware Considerations 8
Modes of Operation 9
Display Support 11
Programmable Option Select 11
Alphanumeric Modes 12
Graphics Modes 14
320 x 200 Four-Color Graphics (Modes Hex 4 and 5) 15
640 x 200 Two-Color Graphics (Mode Hex 6) 17
640 x 350 Graphics (Mode Hex F) 18
640 x 480 Two-Color Graphics (Mode Hex 11) 19
16-Color Graphics Modes (Mode Hex 10, D, E, and 12) . . . 19
256-Color Graphics Mode (Mode Hex 13) 19
Video Memory Organization 21
Memory Modes 21
Modes Hex 0, 1 22
Modes Hex 2, 3 23
Modes Hex 4, 5 24
Mode Hex 6 25
Mode Hex 7 26
Mode Hex D 27
Mode Hex E 29
Mode Hex F 31
Mode Hex 10 32
Mode Hex 11 33
Mode Hex 12 34
Mode Hex 13 35
Memory Operations 36
Write Operations 36
Read Operations 37

© Copyright IBM Corp. 1989, 1990

Registers 38
General Registers 39
Miscellaneous Output Register 39
Input Status Register 0 40
Input Status Register 1 41
Feature Control Register 41
Video Subsystem Enable Register 42
Sequencer Registers 42
Sequencer Address Register 42
Reset Register 43
Clocking Mode Register 43
Map Mask Register 45
Character Map Select Register 46
Memory Mode Register 48
CRT Controller Registers 49
Address Register 50
Horizontal Total Register 50
Horizontal Display-Enable End Register 51
Start Horizontal Blanking Register 51
End Horizontal Blanking Register 51
Start Horizontal Retrace Pulse Register 52
End Horizontal Retrace Register 53
Vertical Total Register 54
Overflow Register 54
Preset Row Scan Register 55
Maximum Scan Line Register 56
Cursor Start Register 57
Cursor End Register 57
Start Address High Register 58
Start Address Low Register 58
Cursor Location High Register 58
Cursor Location Low Register 59
Vertical Retrace Start Register 59
Vertical Retrace End Register 59
Vertical Display Enable End Register 60
Offset Register 61
Underline Location Register 61
Start Vertical Blanking Register 62
End Vertical Blanking Register 62
CRT Mode Control Register 63
Line Compare Register 66
Graphics Controller Registers 67
Address Register 67
Set/Reset Register 68
Enable Set/Reset Register 68
Color Compare Register 69

Data Rotate Register 69
Read Map Select Register 70
Graphics Mode Register 70
Miscellaneous Register 72
Color Don't Care Register 73
Bit Mask Register 73
Attribute Controller Registers 74
Address Register 74
Internal Palette Registers 0 through F 75
Attribute Mode Control Register 76
Overscan Color Register 77
Color Plane Enable Register 77
Horizontal PEL Panning Register 78
Color Select Register 79
VGA Programming Considerations 80
Programming the Registers 83
RAM Loadable Character Generator 84
Creating a Split Screen 85
Video Digital-to-Analog Converter 87
Device Operation 87
Video DAC to System Interface 87
Programming Considerations 88
VGA Video Extensions 89
Auxiliary Video Extension 91
Base Video Extension 92
Auxiliary Signal Descriptions 93
Video Extension Signal Timing 95

XGA Function 96
VGA Function 96
132-Column Text Mode 96
Extended Graphics Function 96
Major Components 98
System Bus Interface 100
Memory and CRT Controller 100
Coprocessor 100
Video Memory 101
Attribute Controller 101
Sprite Controller 101
A/N Font and Sprite Buffer 101
The Serializer, Palette and DAC. 102
Video Memory Requirements 102
Compatibility 103
8514/A 103
Drivers using ROM Address Space 104

Display Connector 105
Signal Timing 106

Index 114

Figures

1. Diagram of the VGA Function 3
2. Graphics Controller 6
3. Attribute Controller 7
4. BIOS Video Modes 9
5. Double Scanning and Border Support 10
6. Direct-Drive Analog Displays 11
7. Character/Attribute Format 13
8. Attribute Byte Definitions 13
9. BIOS Color Set 14
10. Video Memory Format 15
11. PEL Format, Modes Hex 4 and 5 16
12. Color Selections, Modes Hex 4 and 5 16
13. PEL Format, Mode Hex 6 17
14. Bit Definitions C2,C0 18
15. Compatible Color Coding 20
16. 256KB Video Memory Map 21
17. Data Flow for Write Operations 36
18. Color Compare Operations 37
19. Video Subsystem Register Overview 38
20. General Registers 39
21. Miscellaneous Output Register, Hex 03CC/03C2 39
22. Display Vertical Size 39
23. Clock Select Definitions 40
24. Input Status Register 0, Hex 03C2 40
25. Input Status Register 1, Hex 03DA/03BA 41
26. Feature Control Register, Hex 03DA/03BA and 03CA 41
27. Video Subsystem Enable Register, Hex 03C3 42
28. Sequencer Registers 42
29. Sequencer Address Register 42
30. Reset Register, Index Hex 00 43
31. Clocking Mode Register, Index Hex 01 43
32. Map Mask Register, Index Hex 02 45
33. Character Map Select Register, Index Hex 03 46
34. Character Map Select A 47
35. Character Map Select B 47
36. Memory Mode Register, Index Hex 04 48
37. Map Selection, Chain 4 48
38. CRT Controller Registers 49
39. CRT Controller Address Register, Hex 03B4/03D4 50
40. Horizontal Total Register, Index Hex 00 50
41. Horizontal Display Enable-End Register, Index Hex 01 . . . 51
42. Start Horizontal Blanking Register, Index Hex 02 51

© Copyright IBM Corp. 1989, 1990

43. End Horizontal Blanking Register, Index Hex 03 51
44. Display Enable Skew 52
45. Start Horizontal Retrace Pulse Register, Index Hex 04 ... 52
46. End Horizontal Retrace Register, Index Hex 05 53
47. Vertical Total Register, Index Hex 06 54
48. CRT Overflow Register, Index Hex 07 54
49. Preset Row Scan Register, Index Hex 08 55
50. Maximum Scan Line Register, Index Hex 09 56
51. Cursor Start Register, Index Hex 0A 57
52. Cursor End Register, Index Hex 0B 57
53. Start Address High Register, Index Hex 0C 58
54. Start Address Low Register, Index Hex 0D 58
55. Cursor Location High Register, Index Hex 0E 58
56. Cursor Location Low Register, Index Hex 0F 59
57. Vertical Retrace Start Register, Index Hex 10 59
58. Vertical Retrace End Register, Index Hex 11 59
59. Vertical Display Enable End Register, Index Hex 12 60
60. Offset Register, Index Hex 13 61
61. Underline Location Register, Index Hex 14 61
62. Start Vertical Blanking Register, Index Hex 15 62
63. End Vertical Blanking Register, Index Hex 16 62
64. CRT Mode Control Register, Index Hex 17 63
65. CRT Memory Address Mapping 64
66. Line Compare Register, Index Hex 18 66
67. Graphics Controller Register Overview 67
68. Graphics Controller Address Register, Hex 03CE 67
69. Set/Reset Register, Index Hex 00 68
70. Enable Set/Reset Register, Index Hex 01 68
71. Color Compare Register, Index Hex 02 69
72. Data Rotate Register, Index Hex 03 69
73. Operation Select Bit Definitions 70
74. Read Map Select Register, Index Hex 04 70
75. Graphics Mode Register, Index Hex 05 70
76. Write Mode Definitions 71
77. Miscellaneous Register, Index Hex 06 72
78. Video Memory Assignments 72
79. Color Don't Care Register, Index Hex 07 73
80. Bit Mask Register, Index Hex 08 73
81. Attribute Controller Register Addresses 74
82. Address Register, Hex 03C0 74
83. Internal Palette Registers, Index Hex 00 - 0F 75
84. Attribute Mode Control Register, Index Hex 10 76
85. Overscan Color Register, Index Hex 11 77
86. Color Plane Enable Register, Index Hex 12 77
87. Horizontal PEL Panning Register, Index Hex 13 78
88. Image Shifting 78

89. Color Select Register, Index Hex 14 79
90. Character Table Structure 84
91. Character Pattern Example 85
92. Split Screen Definition 85
93. Screen Mapping within the Display Buffer Address Space . 86
94. Video DAC Register 87
| 95. Auxiliary Video Connector Interface 90
96. Auxiliary Video Extension 91
97. Base Video Extension 92
| 98. Auxiliary Video Extension Signal Timing (DAC Signals) . . 95
| 99. XGA Function Block Diagram 99
| 100. Video Memory Requirements 102
101. Display Connector 105
| 102. Display Connector Signals 105
103. Vertical Size 106
104. Vertical Timing, 350 Lines 107
105. Vertical Timing, 400 Lines 108
106. Vertical Timing, 480 Lines 109
| 107. Vertical Timing, 768 Lines 110
108. Horizontal Timing, 80 Column with Border 111
109. Horizontal Timing, 40/80 Column, without Border 112
| 110. Horizontal Timing, 1024 PELs 113

Video Subsystem

| The system video can be generated by a Type 1 or Type 2 video
| subsystem:

| • Type 1 video—Video Graphics Array (VGA)
| • Type 2 video—Extended Graphics Array (XGA).

| Type 1

| The Type 1 video provides the VGA function. The capabilities and
| operation of the VGA function are described in "VGA Function" on
| page 2.

| Only one Type 1 video subsystem is allowed in a system.

| Type 2

| The Type 2 video subsystem provides the XGA function, which
| consists of the VGA function, 132-column text mode, and the extended
| graphics function. The capabilities and operation of the XGA function
| are described in "XGA Function" on page 96.

© Copyright IBM Corp. 1989, 1990

VGA Function

The basic system video is generated by the Type 1 or Type 2 video subsystem. The circuitry that provides the VGA function includes a video buffer, a video digital-to-analog converter (DAC), and test circuitry. Video memory is mapped as four planes of 64Kb by 8 bits (maps 0 through 3). The video DAC drives the analog output to the display connector. The test circuitry is used to test for the type of display attached, color or monochrome.

The video subsystem controls the access to video memory from the system and the cathode-ray tube (CRT) controller. It also controls the system addresses assigned to video memory. Up to three different starting addresses can be programmed for compatibility with previous video adapters.

In the graphics modes, the mode determines the way video information is formatted into memory, and the way memory is organized.

In alphanumeric modes, the system writes the ASCII character code and attribute data to video memory maps 0 and 1, respectively. Memory map 2 contains the character font loaded by BIOS during an alphanumeric mode set. The font is used by the character generator to create the character image on the display.

Three fonts are contained in ROM: an 8-by-8 font, an 8-by-14 font, and an 8-by-16 font. Up to eight 256-character fonts can be loaded into the video memory map 2; two of these fonts can be active at any one time, allowing a 512-character font.

The video subsystem formats the information in video memory and sends the output to the video DAC. For color displays, the video DAC sends three analog color signals (red, green, and blue) to the display connector. For monochrome displays, BIOS translates the color information in the DAC, and the DAC drives the summed signal onto the green output.

The auxiliary video connector allows video data to be passed between the video subsystem and an adapter plugged into the channel connector.

© Copyright IBM Corp. 1989, 1990

| When it is disabled, the video subsystem will not respond to video
| memory or I/O reads or writes; however, the video image continues
| to be displayed.

Note: Compatibility with other hardware is best achieved by using the BIOS interface or operating system interface whenever possible.

| The following is a diagram of the VGA function.

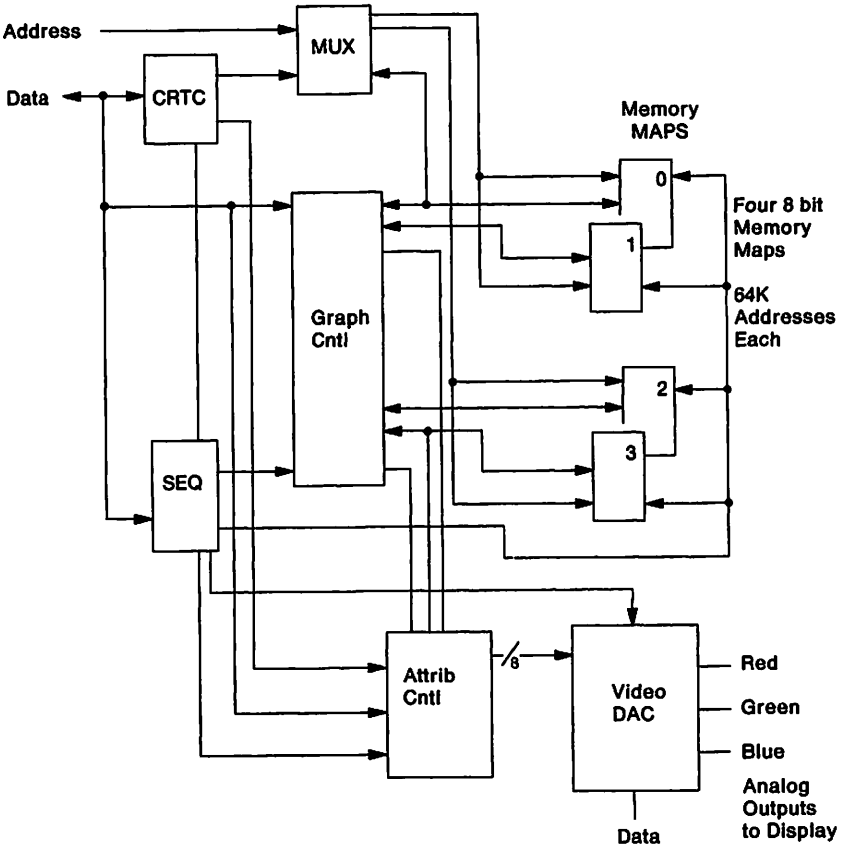


Figure 1. Diagram of the VGA Function

Major Components

The video subsystem contains all circuits necessary to generate the timing for the video memory, and generates the video information going to the video DAC. The major components are: ROM BIOS, the support logic, and the Video Graphics Array interface.

ROM BIOS

BIOS provides software support and contains the character fonts and the system interface to run the video subsystem.

Support Logic

The support logic consists of the video memory, the clocks, and the video DAC. The video memory consists of at least 256KB; its use and mapping depend on the mode selected.

Two clock sources provide the dot rate. The clock source is selected in the Miscellaneous Output register.

The video DAC contains the color palette that is used to convert the video data into the video signal sent to the display. Three analog signals (red, green, blue) are output from the DAC.

The maximum number of colors displayed is 256 out of 256K, and the maximum number of gray shades is 64 out of 64.

VGA Components

The VGA function has four major functional areas: the CRT controller, the sequencer, the graphics controller, and the attribute controller.

CRT Controller

The CRT controller generates horizontal and vertical synchronization signal timings, addressing for the regenerative buffer, cursor and underline timings, and refresh addressing for the video memory.

Sequencer

The sequencer generates basic memory timings for the video memory and the character clock for controlling regenerative buffer fetches. It allows the system to access memory during active display intervals by periodically inserting dedicated system microprocessor memory cycles between the display memory cycles. Map mask registers in the sequencer are available to protect entire memory maps from being changed.

Graphics Controller

The graphics controller is the interface between the video memory and the attribute controller during active display times, and between video memory and the system microprocessor during memory accesses.

During active display times, memory data is latched and sent to the attribute controller. In graphics modes, the memory data is converted from parallel to serial bit-plane data before being sent; in alphanumeric modes, the parallel attribute data is sent.

During system accesses of video memory, the graphics controller can perform logical operations on the memory data before it reaches video memory or the system data bus. These logical operations are composed of four logical write modes and two logical read modes. The logical operators allow enhanced operations, such as a color compare in the read mode, individual bit masking during write modes, internal 32-bit writes in a single memory cycle, and writing to the display buffer on nonbyte boundaries.

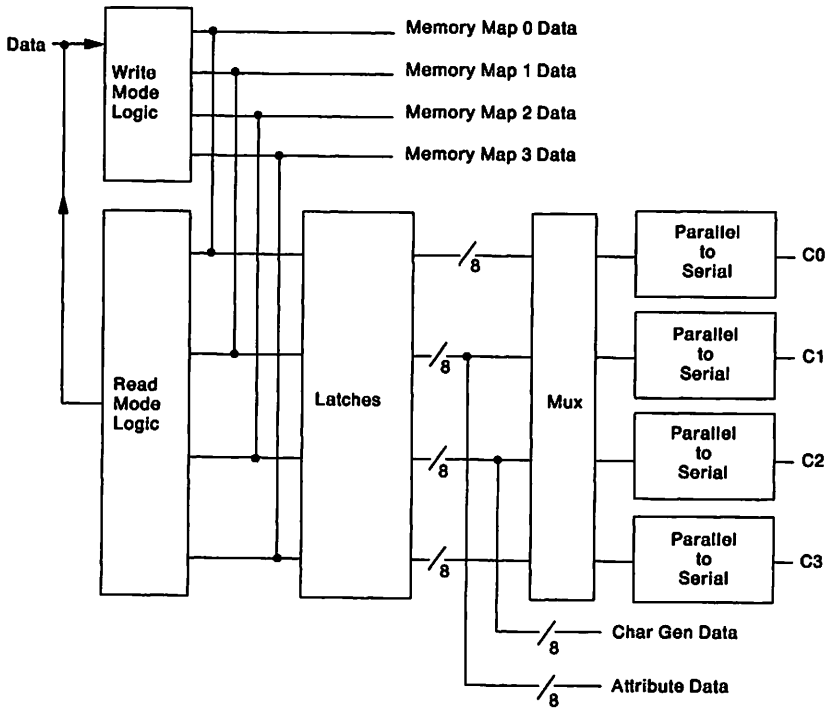


Figure 2. Graphics Controller

Attribute Controller

The attribute controller takes in data from video memory through the graphics controller and formats it for display. Attribute data in alphanumeric mode and serialized bit-plane data in graphics mode are converted to an 8-bit color value.

Each color value is selected from an internal color palette of 64 possible colors (except in 256-color mode). The color value is used as a pointer into the video DAC where it is converted to the analog signals that drive the display.

Blinking, underlining, cursor insertion, and PEL panning are also controlled in the attribute controller.

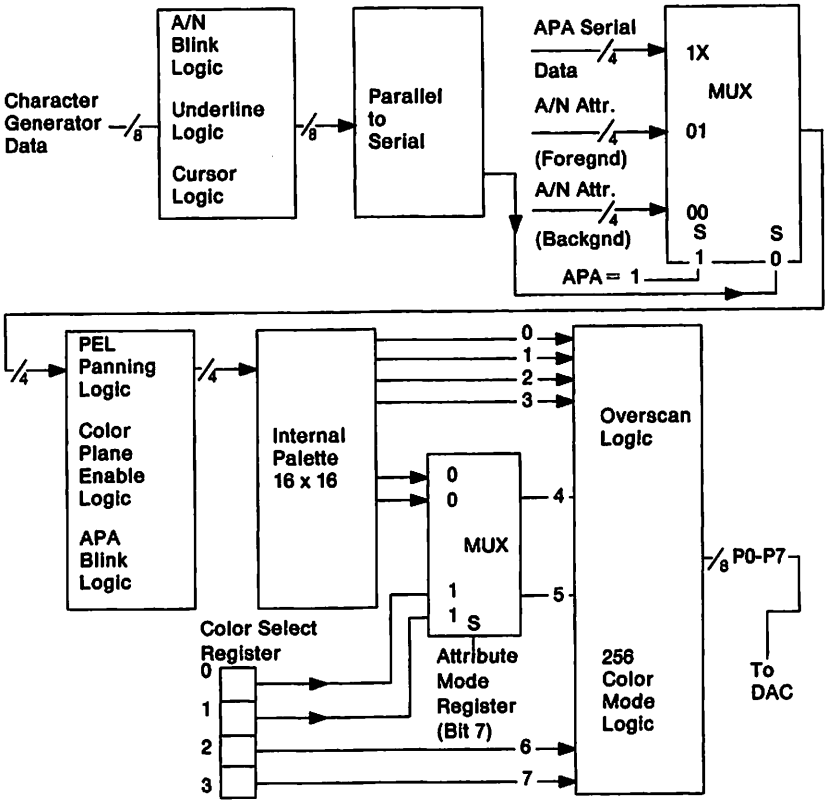


Figure 3. Attribute Controller

Hardware Considerations

The following are hardware characteristics of the Type 2 video subsystem that must be considered to ensure program compatibility with the Type 1 video subsystem.

Performance: The Type 2 video runs generally faster than the Type 1 video; programs that depend on execution time of the video subsystem will operate differently.

Video Buffer Compatibility: For each of the given video modes, the Type 2 video subsystem maintains a memory mapping that is the same as the Type 1. To maintain this compatibility, the internal addresses to video memory are manipulated so that video memory looks the same. When switching video modes, video data may not be at the same address in video memory.

BIOS calls to set and change modes make allowances for changes in addresses, and should be used for all mode switches.

Character Generator: Differences in the character generator for the Type 2 video increase the time that it takes to load a new font. Because of the additional load time, there is a chance of briefly observing spurious data on the display. BIOS compensates for this during video mode sets.

Register Differences: The following bits for the Type 2 video subsystem differ from the Type 1:

- Bits 2 and 4 in the Clocking Mode register
- Bits 5 and 6 in the End Horizontal Blanking register
- Bits 2 and 4 in the Preset Row Scan register
- Bit 5 in the Address register of the attribute controller.

Modes of Operation

Certain modes on previous IBM display adapters distinguished between monochrome and color displays. For example, mode 0 was the same as mode 1 with the color burst turned off. Because color burst is not supported by the PS/2 video, the mode pairs are exactly the same. The support logic for the VGA function recognizes the type of display, and adjusts the output accordingly. When a monochrome display is attached, the colors for the color modes appear as shades of gray.

Mode 3+ is the default mode with a color display attached and mode 7+ is the default mode with a monochrome display attached.

The following figure describes the alphanumeric (A/N) and all points addressable (APA) graphics modes supported by BIOS. Each color is selected from 256K possibilities, and gray shades from 64 possibilities. The variations within the basic BIOS modes are selected through BIOS calls that set the number of scan lines. The scan line count is set before the mode call is made.

Mode (Hex)	Type	Colors	Alpha Format	Buffer Start	Box Size	Max. Pgs.	Freq.	Vert. PELs
0, 1	A/N	16	40 x 25	B8000	8 x 8	8	70 Hz	320 x 200
0*, 1*	A/N	16	40 x 25	B8000	8 x 14	8	70 Hz	320 x 350
0+, 1+	A/N	16	40 x 25	B8000	9 x 16	8	70 Hz	360 x 400
2, 3	A/N	16	80 x 25	B8000	8 x 8	8	70 Hz	640 x 200
2*, 3*	A/N	16	80 x 25	B8000	8 x 14	8	70 Hz	640 x 350
2+, 3+	A/N	16	80 x 25	B8000	9 x 16	8	70 Hz	720 x 400
4, 5	APA	4	40 x 25	B8000	8 x 8	1	70 Hz	320 x 200
6	APA	2	80 x 25	B8000	8 x 8	1	70 Hz	640 x 200
7	A/N	-	80 x 25	B0000	9 x 14	8	70 Hz	720 x 350
7+	A/N	-	80 x 25	B0000	9 x 16	8	70 Hz	720 x 400
D	APA	16	40 x 25	A0000	8 x 8	8	70 Hz	320 x 200
E	APA	16	80 x 25	A0000	8 x 8	4	70 Hz	640 x 200
F	APA	-	80 x 25	A0000	8 x 14	2	70 Hz	640 x 350
10	APA	16	80 x 25	A0000	8 x 14	2	70 Hz	640 x 350
11	APA	2	80 x 30	A0000	8 x 16	1	60 Hz	640 x 480
12	APA	16	80 x 30	A0000	8 x 16	1	60 Hz	640 x 480
13	APA	256	40 x 25	A0000	8 x 8	1	70 Hz	320 x 200

* or + Enhanced modes

Figure 4. BIOS Video Modes

In the 200-scan-line modes, the data for each scan line is scanned twice. This double scanning allows the 200-scan-line image to be displayed in 400 scan lines.

Border support and double scanning depend on the mode selected. The following shows which modes use double scanning and which support a border.

Mode (Hex)	Double Scan	Border Support
0, 1	Yes	No
0*, 1*	No	No
0+, 1+	No	No
2, 3	Yes	Yes
2*, 3*	No	Yes
2+, 3+	No	Yes
4, 5	Yes	No
6	Yes	Yes
7	No	Yes
7+	No	Yes
D	Yes	No
E	Yes	Yes
F	No	Yes
10	No	Yes
11	No	Yes
12	No	Yes
13	Yes	Yes

Figure 5. Double Scanning and Border Support

Display Support

The video subsystem supports direct-drive analog displays. The displays must have a horizontal sweep frequency of 31.5 kHz, and a vertical sweep frequency capability of 50 to 70 Hz. Displays that use a digital input, such as the IBM Color Display, are *not* supported. The following figure summarizes the minimum display characteristics required to support VGA mode operation.

Parameter	Color	Monochrome
Horizontal Scan Rate	31.5 kHz	31.5 kHz
Vertical Scan Rate	50 to 70 Hz	50 to 70 Hz
Video Bandwidth	28 MHz	28 MHz
Maximum Horizontal Resolution	720 PELs	720 PELs
Maximum Vertical Resolution	480 PELs	480 PELs

Figure 6. Direct-Drive Analog Displays

Since color and monochrome displays run at the same sweep rate, all modes work on both displays. The vertical gain of the display is controlled by the polarity of the vertical and horizontal synchronization pulses. This is done so 350, 400, or 480 lines can be displayed without adjusting the display. See "Signal Timing" on page 106 for more information.

Programmable Option Select

The video subsystem supports programmable option select (POS). The video subsystem is placed in the setup mode through the System Board Setup/Enable register (hex 0094). (For more information, see the system-specific sections.) For information on BIOS calls to enable or disable the video, see the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference.*

Alphanumeric Modes

The alphanumeric modes are modes 0 through 3 and 7. The mode chart lists the variations of these modes. The data format for alphanumeric modes is the same as the data format on the IBM Color/Graphics Monitor Adapter, the IBM Monochrome Display Adapter, and the IBM Enhanced Graphics Adapter.

BIOS initializes the video subsystem according to the selected mode and loads the color values into the video DAC. These color values can be changed to give a different color set to select from. Bit 3 of the attribute byte may be redefined by the Character Map Select register to act as a switch between character sets, giving the programmer access to 512 characters at one time.

When an alphanumeric mode is selected, the BIOS transfers character font patterns from the ROM to map 2. The system stores the character data in map 0, and the attribute data in map 1. In the alphanumeric modes, the programmer views maps 0 and 1 as a single buffer. The CRT controller generates sequential addresses, and fetches one character code byte and one attribute byte at a time. The character code and row scan count are combined to make up the address into map 2, which contains the character font. The appropriate dot patterns are then sent to the attribute controller, where color is assigned according to the attribute data.

Every display-character position in the alphanumeric mode is defined by two bytes in the display buffer. Both the color/graphics and the monochrome emulation modes use the following 2-byte character/attribute format.

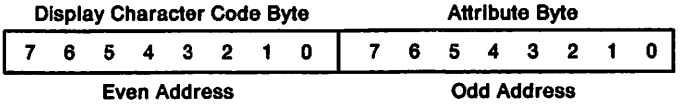


Figure 7. Character/Attribute Format

See "Characters and Keystrokes" for characters loaded during a BIOS mode set.

The functions of the attribute byte are defined in the following table. Bit 7 can be redefined in the Attribute Mode Control register to give 16 possible background colors; its default is to control character blinking. Bit 3 can be redefined in the Character Map Select register to select between two character fonts; its default is to control foreground color selection.

Bit	Color	Function
7	B/I	Blinking or Background Intensity
6	R	Background Color
5	G	Background Color
4	B	Background Color
3	I/CS	Foreground Intensity or Character Font Select
2	R	Foreground Color
1	G	Foreground Color
0	B	Foreground Color

Figure 8. Attribute Byte Definitions

For more information about the attribute byte, see "Character Map Select Register" on page 46 and "Attribute Mode Control Register" on page 76.

The following are the color values loaded by BIOS for the 16-color modes.

I	R	G	B	Color
0	0	0	0	Black
0	0	0	1	Blue
0	0	1	0	Green
0	0	1	1	Cyan
0	1	0	0	Red
0	1	0	1	Magenta
0	1	1	0	Brown
0	1	1	1	White
1	0	0	0	Gray
1	0	0	1	Light Blue
1	0	1	0	Light Green
1	0	1	1	Light Cyan
1	1	0	0	Light Red
1	1	0	1	Light Magenta
1	1	1	0	Yellow
1	1	1	1	White (High Intensity)

Figure 9. BIOS Color Set

Both 40-column and 80-column alphanumeric modes are supported. The features of the 40-column alphanumeric modes (all variations of modes hex 0 and 1) are:

- 25 rows of 40 characters
- 2,000 bytes of video memory per page
- One character byte and one attribute byte per character.

The features of the 80-column alphanumeric modes (all variations of modes hex 2, 3, and 7) are:

- 25 rows of 80 characters
- 4,000 bytes of video memory per page
- One character byte and one attribute byte per character.

Graphics Modes

This section describes the graphics modes supported in BIOS. The colors in this section are generated when the BIOS is used to set the mode. BIOS initializes the video subsystem and the DAC palette to generate these colors. If the DAC palette is changed, different colors are generated.

320 x 200 Four-Color Graphics (Modes Hex 4 and 5)

Addressing, mapping, and data format are the same as the 320 x 200 PEL mode of the IBM Color/Graphics Monitor Adapter. The display buffer is configured at hex B8000. Bit image data is stored in memory maps 0 and 1. The two bit planes (C0 and C1) are each formed from bits from both memory maps.

Features of this mode are:

- A maximum of 200 rows of 320 PELs
- Double-scanned to display as 400 rows
- Memory-mapped graphics
- Four colors for each PEL
- Four PELs per byte
- 16,000 bytes of read/write memory.

The video memory is organized into two banks of 8,000 bytes each using the following format. Address hex B8000 contains the PEL information for the upper-left corner of the display area.

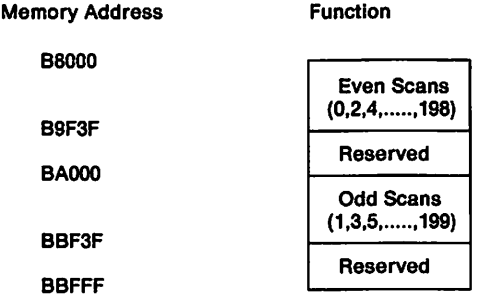


Figure 10. Video Memory Format

The following figure shows the format for each byte.

Bit	Function
7	C1 - First Display PEL
6	C0 - First Display PEL
5	C1 - Second Display PEL
4	C0 - Second Display PEL
3	C1 - Third Display PEL
2	C0 - Third Display PEL
1	C1 - Fourth Display PEL
0	C0 - Fourth Display PEL

Figure 11. PEL Format, Modes Hex 4 and 5

The color selected depends on the color set that is used. Color set 1 is the default. For information on changing the color set, see the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference.*

Bits C1 C0	Color Selected	
	Color Set 1	Color Set 0
0 0	Black	Black
0 1	Light Cyan	Green
1 0	Light Magenta	Red
1 1	Intensified White	Brown

Figure 12. Color Selections, Modes Hex 4 and 5

640 x 200 Two-Color Graphics (Mode Hex 6)

Addressing, scan-line mapping, and data format are the same as the 640 x 200 PEL black and white mode of the IBM Color/Graphics Monitor Adapter. The display buffer is configured at hex B8000. Bit image data is stored in memory map 0 and comprises a single bit plane (C0). Features of this mode are:

- A maximum of 200 rows of 640 PELs
- Double scanned to display as 400 rows
- Same addressing and scan-line mapping as 320 x 200 graphics
- Two colors for each PEL
- Eight PELs per byte
- 16,000 bytes of read/write memory.

The following shows the format for each byte.

Bit	Function
7	First Display PEL
6	Second Display PEL
5	Third Display PEL
4	Fourth Display PEL
3	Fifth Display PEL
2	Sixth Display PEL
1	Seventh Display PEL
0	Eighth Display PEL

Figure 13. PEL Format, Mode Hex 6

The bit definition for each PEL is 0 equals black and 1 equals intensified white.

640 x 350 Graphics (Mode Hex F)

This mode emulates the EGA graphics with the monochrome display and the following attributes: black, video, blinking video, and intensified video. A resolution of 640 x 350 uses 56,000 bytes of video memory to support the four attributes. This mode uses maps 0 and 2; map 0 is the video bit plane (C0), and map 2 is the intensity bit plane (C2). Both planes reside at address hex A0000.

The two bits, one from each bit plane, define one PEL. The bit definitions are given in the following table.

C2	C0	PEL Color
0	0	Black
0	1	White
1	0	Blinking White
1	1	Intensified White

Figure 14. Bit Definitions C2,C0

Memory is organized with successive bytes defining successive PELs. The first eight PELs displayed are defined by the byte at hex A0000, the second eight PELs at hex A0001, and so on. The most-significant bit in each byte defines the first PEL for that byte.

Since both bit planes reside at address hex A0000, the user must select the plane to update through the Map Mask register of the sequence controller (see "Video Memory Organization" on page 21).

640 x 480 Two-Color Graphics (Mode Hex 11)

This mode provides two-color graphics with the same data format as mode 6. Addressing and mapping are shown under "Video Memory Organization" on page 21.

The bit image data is stored in map 0 and comprises a single bit plane (C0). The video buffer starts at hex A0000. The first byte contains the first eight PELs; the second byte at hex A0001 contains the second eight PELs, and so on. The bit definition for each PEL is 0 equals black and 1 equals intensified white.

16-Color Graphics Modes (Mode Hex 10, D, E, and 12)

These modes support 16 colors. For all modes, the bit image data is stored in all four memory maps. Each memory map contains the data for one bit plane. The bit planes are C0 through C3 and represent the following colors:

C0 = Blue
C1 = Green
C2 = Red
C3 = Intensified

The four bits define each PEL on the screen by acting as an address (pointer) into the internal palette in the Type 2 video.

The display buffer resides at address hex A0000. The Map Mask register selects any or all of the maps to be updated when the system writes to the display buffer.

256-Color Graphics Mode (Mode Hex 13)

This mode provides graphics with the capability of displaying 256 colors at one time.

The display buffer is sequential, starts at address hex A0000, and is 64,000 bytes long. The first byte contains the color information for the upper-left PEL. The second byte contains the second PEL, and so on, for 64,000 PELs (320 x 200). The bit image data is stored in all four memory maps and comprises four bit planes. The four bit planes are sampled twice to produce eight bit-plane values that address the video DAC.

In this mode, the internal palette of the video subsystem is loaded by BIOS and should not be changed. The first 16 locations in the external palette, which is in the video DAC, contain the colors

compatible with the alphanumeric modes. The second 16 locations contain 16 evenly spaced gray shades. The next 216 locations contain values based on a hue-saturation-intensity model tuned to provide a usable, generic color set that covers a wide range of color values.

The following figure shows the color information that is compatible with the colors in other modes.

PEL Bits 76543210	Color Output
00000000	Black
00000001	Blue
00000010	Green
00000011	Cyan
00000100	Red
00000101	Magenta
00000110	Brown
00000111	White
00001000	Dark Gray
00001001	Light Blue
00001010	Light Green
00001011	Light Cyan
00001100	Light Red
00001101	Light Magenta
00001110	Yellow
00001111	Intensified White

Figure 15. Compatible Color Coding

Each color in the palette can be programmed to one of 256K different colors.

The features of this mode are:

- A maximum of 200 rows with 320 PELs
- Double scanned to display as 400 rows
- Memory-mapped graphics
- 256 of 256K colors for each PEL
- One byte per PEL
- 64,000 bytes of video memory.

Video Memory Organization

The display buffer consists of 256KB of dynamic read/write memory configured as four 64KB memory maps.

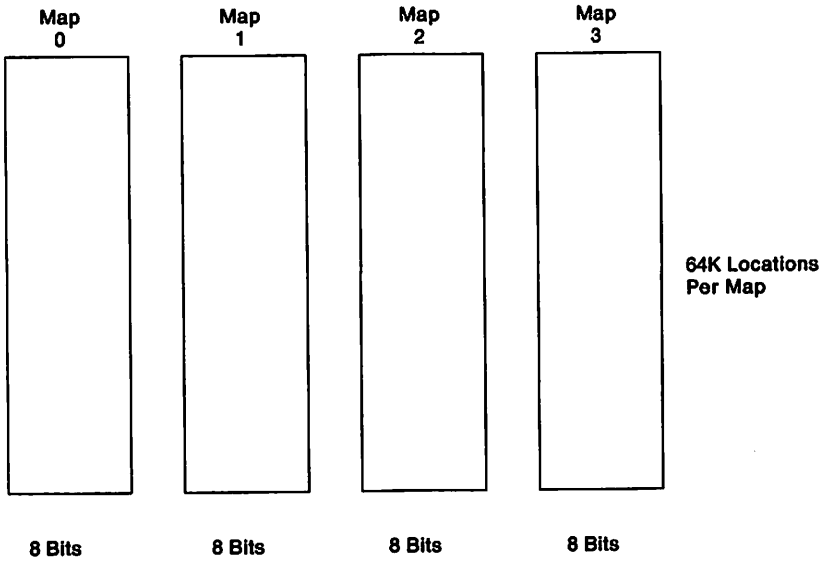


Figure 16. 256KB Video Memory Map

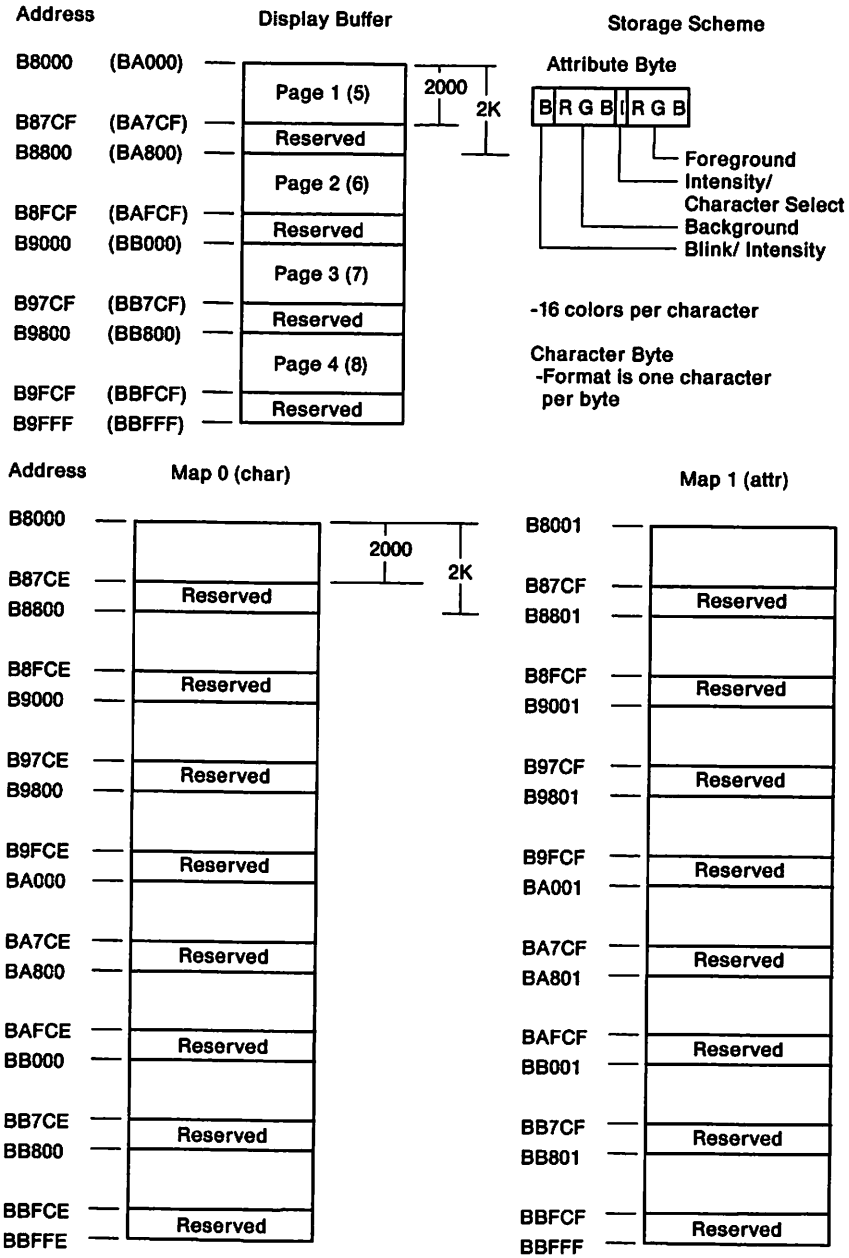
The starting address and size of the display buffer can be changed to maintain compatibility with other display adapters and application software. There are three configurations used by other adapters:

Address hex A0000 for a length of 64KB
Address hex B0000 for a length of 32KB
Address hex B8000 for a length of 32KB.

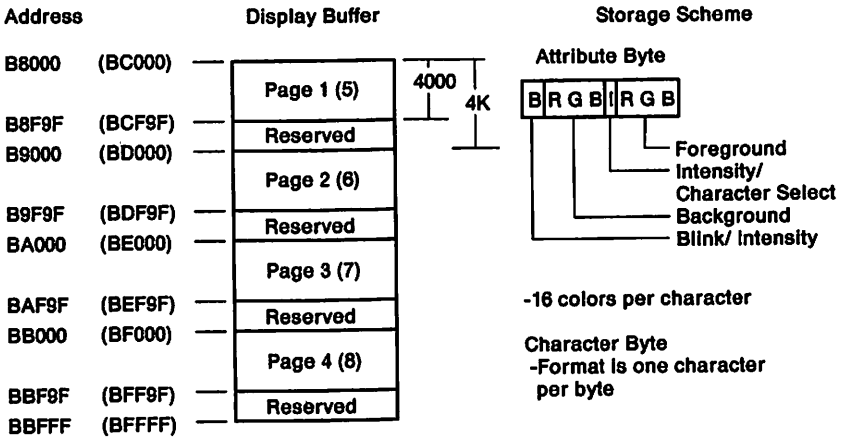
Memory Modes

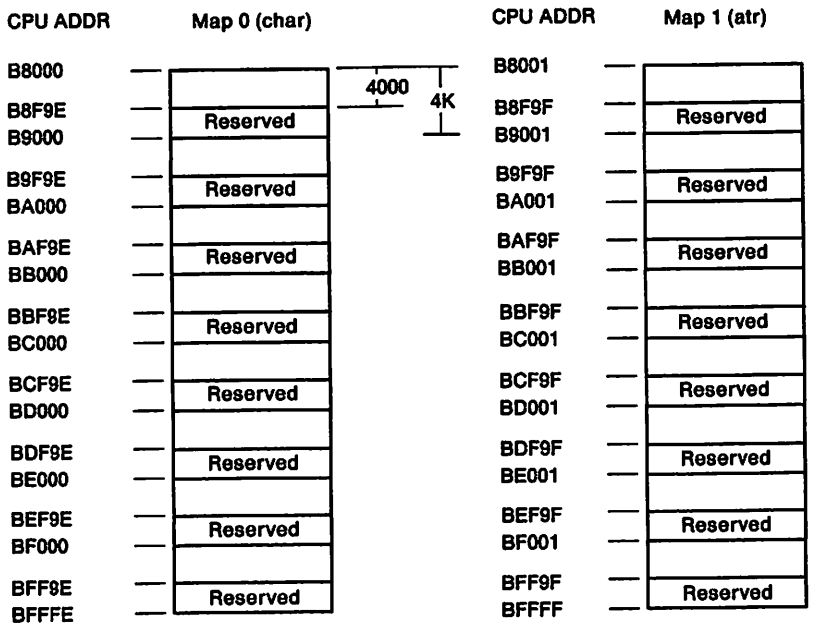
The following pages show the memory organization for each of the BIOS modes.

Modes Hex 0, 1

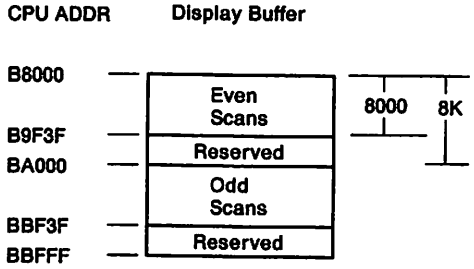

Address
Display Buffer
Storage Scheme
B8000 (BA000)
Page 1 (5)
2000
2K
B87CF (BA7CF)
Reserved
B8800 (BA800)
Page 2 (6)
B8FCF (BAFCF)
Reserved
B9000 (BB000)
Page 3 (7)
B97CF (BB7CF)
Reserved
B9800 (BB800)
Page 4 (8)
B9FCF (BBFCF)
Reserved
B9FFF (BBFFF)
Attribute Byte
B R G B I R G B
Foreground
Intensity/
Character Select
Background
Blink/ Intensity
-16 colors per character
Character Byte
-Format is one character
per byte
Address
Map 0 (char)
Map 1 (attr)
B8000
B8001
2000
2K
B87CE
B87CF
Reserved
B8800
B8801
B8FCE
B8FCF
Reserved
B9000
B9001
B97CE
B97CF
Reserved
B9800
B9801
B9FCE
B9FCF
Reserved
BA000
BA001
BA7CE
BA7CF
Reserved
BA800
BA801
BAFCE
BAFCF
Reserved
BB000
BB001
BB7CE
BB7CF
Reserved
BB800
BB801
BBFCE
BBFCF
Reserved
BBFFE
BBFFF

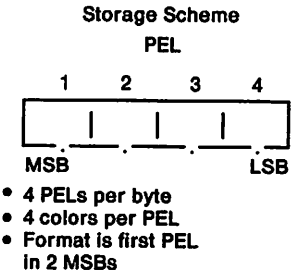
Modes Hex 2, 3

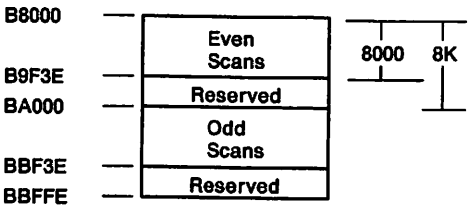

Address
Display Buffer
Storage Scheme
B8000 (BC000)
B8F9F (BCF9F)
B9000 (BD000)
B9F9F (BDF9F)
BA000 (BE000)
BAF9F (BEF9F)
BB000 (BF000)
BBF8F (BFF9F)
BBFFF (BFFFF)
Page 1 (5)
Reserved
Page 2 (6)
Reserved
Page 3 (7)
Reserved
Page 4 (8)
Reserved
4000
4K
Attribute Byte
B R G B I R G B
Foreground
Intensity/
Character Select
Background
Blink/ Intensity
-16 colors per character
Character Byte
-Format is one character
per byte

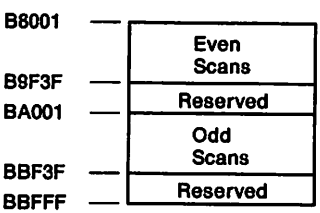

CPU ADDR
Map 0 (char)
B8000
B8F9E
B9000
B9F9E
BA000
BAF8E
BB000
BBF8E
BC000
BCF9E
BD000
BDF9E
BE000
BEF9E
BF000
BFF8E
BFFFE
Reserved
4000
4K
CPU ADDR
Map 1 (atr)
B8001
B8F9F
B9001
B9F9F
BA001
BAF9F
BB001
BBF9F
BC001
BCF9F
BD001
BDF9F
BE001
BEF9F
BF001
BFF9F
BFFFF
Reserved

Modes Hex 4, 5

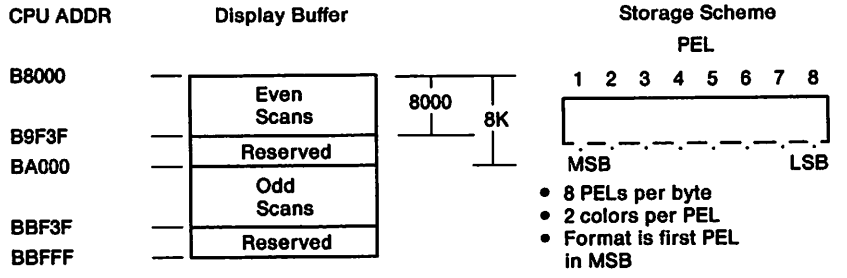

CPU ADDR
Display Buffer
B8000
B9F3F
BA000
BBF3F
BBFFF
Even Scans
Reserved
Odd Scans
Reserved
8000
8K

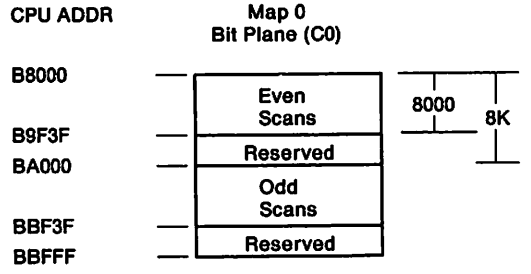

Storage Scheme
PEL
1
2
3
4
MSB
LSB
• 4 PELs per byte
• 4 colors per PEL
• Format is first PEL in 2 MSBs


CPU ADDR
Map 0
B8000
B9F3E
BA000
BBF3E
BBFFE
Even Scans
Reserved
Odd Scans
Reserved
8000
8K

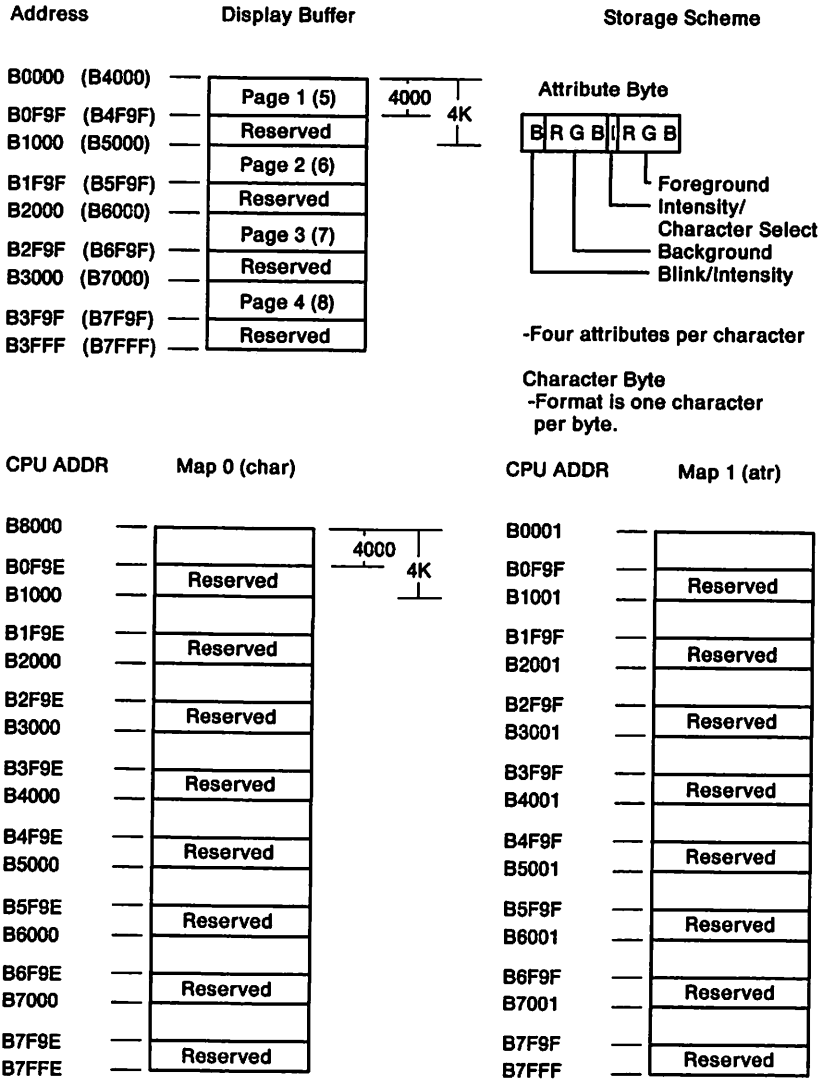

CPU ADDR
Map 1
B8001
B9F3F
BA001
BBF3F
BBFFF
Even Scans
Reserved
Odd Scans
Reserved

Mode Hex 6

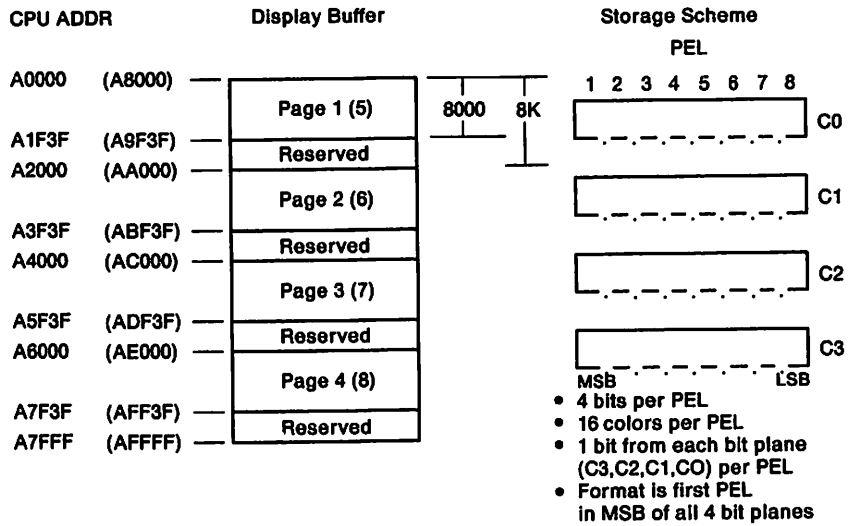

CPU ADDR
Display Buffer
Storage Scheme
PEL
1 2 3 4 5 6 7 8
B8000
B9F3F
BA000
BBF3F
BBFFF
Even Scans
Reserved
Odd Scans
Reserved
8000
8K
MSB
LSB
• 8 PELs per byte
• 2 colors per PEL
• Format is first PEL in MSB

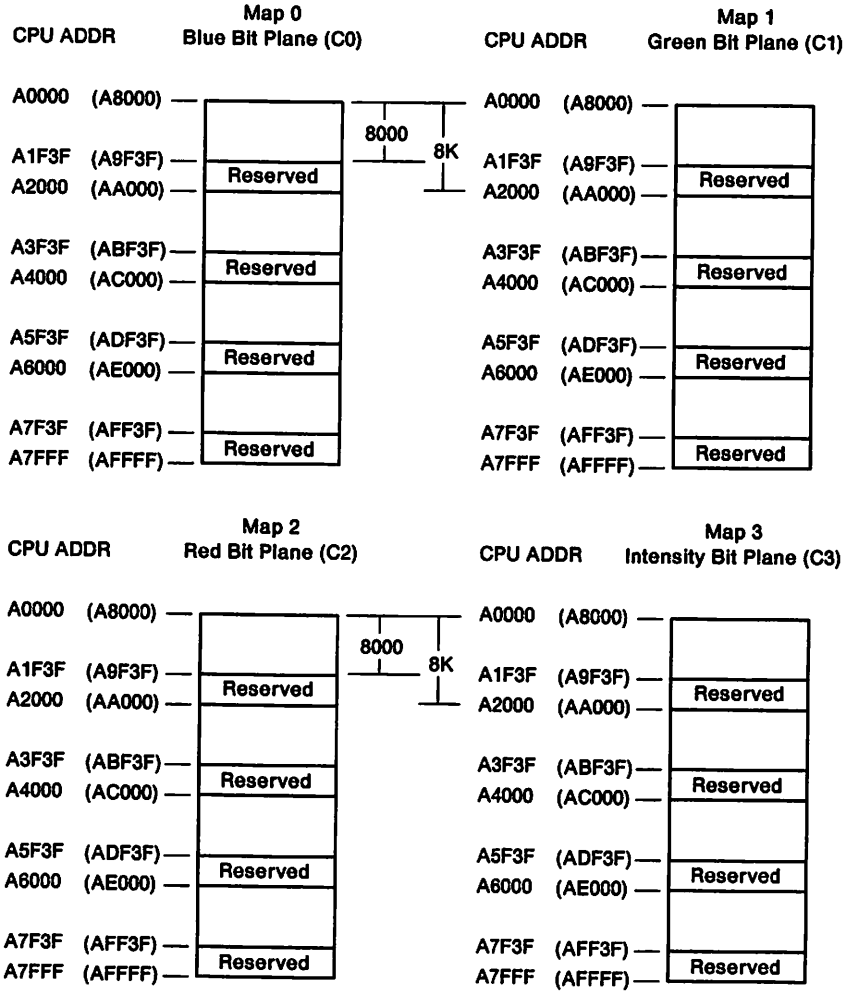

CPU ADDR
Map 0
Bit Plane (C0)
B8000
B9F3F
BA000
BBF3F
BBFFF
Even Scans
Reserved
Odd Scans
Reserved
8000
8K

Mode Hex 7

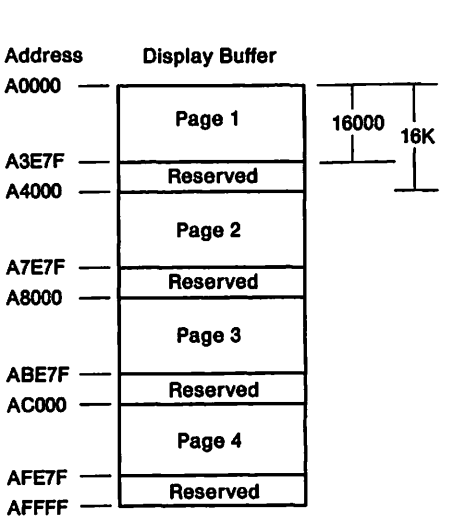

Address
Display Buffer
Storage Scheme
B0000 (B4000)
B0F9F (B4F9F)
B1000 (B5000)
B1F9F (B5F9F)
B2000 (B6000)
B2F9F (B6F9F)
B3000 (B7000)
B3F9F (B7F9F)
B3FFF (B7FFF)
Page 1 (5)
Reserved
Page 2 (6)
Reserved
Page 3 (7)
Reserved
Page 4 (8)
Reserved
4000
4K
Attribute Byte
B R G B I R G B
Foreground
Intensity/
Character Select
Background
Blink/Intensity
-Four attributes per character
Character Byte
-Format is one character
per byte.
CPU ADDR
Map 0 (char)
B8000
B0F9E
B1000
B1F9E
B2000
B2F9E
B3000
B3F9E
B4000
B4F9E
B5000
B5F9E
B6000
B6F9E
B7000
B7F9E
B7FFE
Reserved
4000
4K
CPU ADDR
Map 1 (atr)
B0001
B0F9F
B1001
B1F9F
B2001
B2F9F
B3001
B3F9F
B4001
B4F9F
B5001
B5F9F
B6001
B6F9F
B7001
B7F9F
B7FFF
Reserved

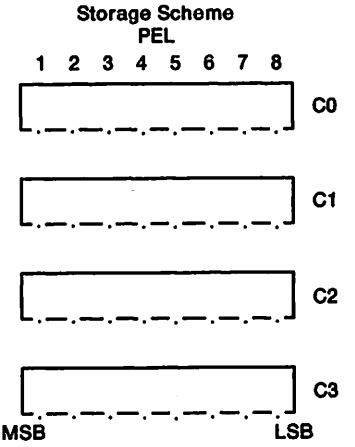
Mode Hex D


CPU ADDR
Display Buffer
Storage Scheme
PEL
A0000 (A8000)
A1F3F (A9F3F)
A2000 (AA000)
A3F3F (ABF3F)
A4000 (AC000)
A5F3F (ADF3F)
A6000 (AE000)
A7F3F (AFF3F)
A7FFF (AFFFF)
Page 1 (5)
Reserved
Page 2 (6)
Reserved
Page 3 (7)
Reserved
Page 4 (8)
Reserved
8000
8K
1 2 3 4 5 6 7 8
C0
C1
C2
C3
MSB
LSB
• 4 bits per PEL
• 16 colors per PEL
• 1 bit from each bit plane (C3,C2,C1,C0) per PEL
• Format is first PEL in MSB of all 4 bit planes

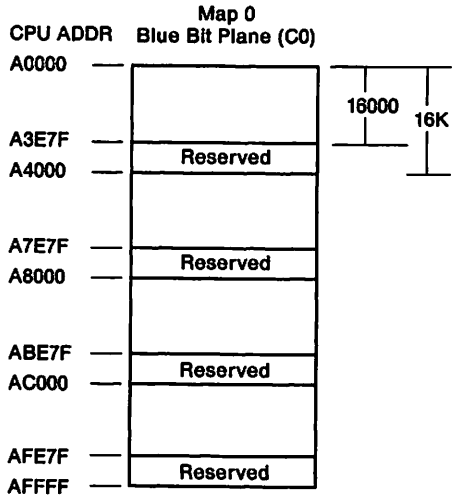

Map 0
CPU ADDR
Blue Bit Plane (C0)
A0000 (A8000)
A1F3F (A9F3F)
A2000 (AA000)
A3F3F (ABF3F)
A4000 (AC000)
A5F3F (ADF3F)
A6000 (AE000)
A7F3F (AFF3F)
A7FFF (AFFFF)
Reserved
Reserved
Reserved
Reserved
8000
8K
Map 1
CPU ADDR
Green Bit Plane (C1)
A0000 (A8000)
A1F3F (A9F3F)
A2000 (AA000)
A3F3F (ABF3F)
A4000 (AC000)
A5F3F (ADF3F)
A6000 (AE000)
A7F3F (AFF3F)
A7FFF (AFFFF)
Reserved
Reserved
Reserved
Reserved
Map 2
CPU ADDR
Red Bit Plane (C2)
A0000 (A8000)
A1F3F (A9F3F)
A2000 (AA000)
A3F3F (ABF3F)
A4000 (AC000)
A5F3F (ADF3F)
A6000 (AE000)
A7F3F (AFF3F)
A7FFF (AFFFF)
Reserved
Reserved
Reserved
Reserved
8000
8K
Map 3
CPU ADDR
Intensity Bit Plane (C3)
A0000 (A8000)
A1F3F (A9F3F)
A2000 (AA000)
A3F3F (ABF3F)
A4000 (AC000)
A5F3F (ADF3F)
A6000 (AE000)
A7F3F (AFF3F)
A7FFF (AFFFF)
Reserved
Reserved
Reserved
Reserved

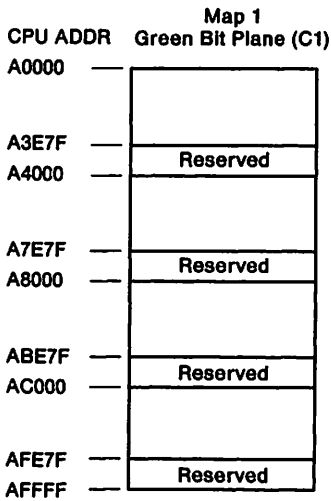
Mode Hex E

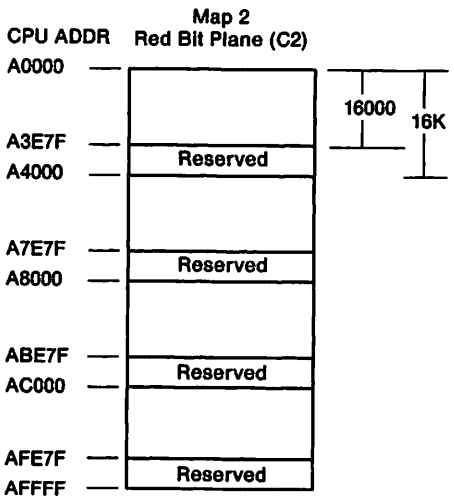


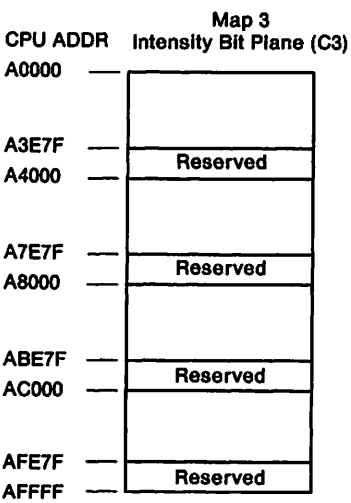


- 4 bits per PEL
- 16 colors per PEL
- 1 bit from each bit plane (C3,C2,C1,C0) per PEL
- First PEL is MSB of all 4 bit planes

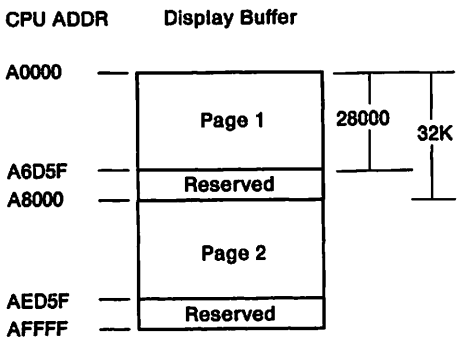

Map 0
CPU ADDR
Blue Bit Plane (C0)
A0000
A3E7F
A4000
A7E7F
A8000
ABE7F
AC000
AFE7F
AFFFF
Reserved
Reserved
Reserved
Reserved
16000
16K

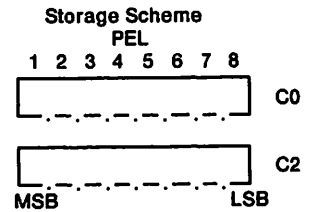

Map 1
CPU ADDR
Green Bit Plane (C1)
A0000
A3E7F
A4000
A7E7F
A8000
ABE7F
AC000
AFE7F
AFFFF
Reserved
Reserved
Reserved
Reserved


Map 2
CPU ADDR
Red Bit Plane (C2)
A0000
A3E7F
A4000
A7E7F
A8000
ABE7F
AC000
AFE7F
AFFFF
Reserved
Reserved
Reserved
Reserved
16000
16K

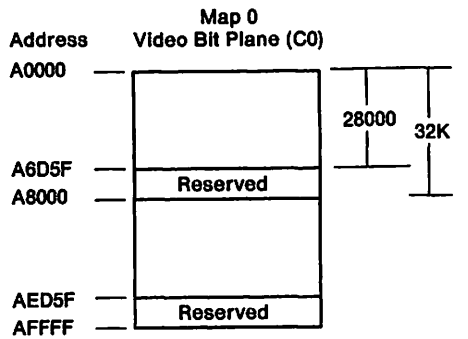

Map 3
CPU ADDR
Intensity Bit Plane (C3)
A0000
A3E7F
A4000
A7E7F
A8000
ABE7F
AC000
AFE7F
AFFFF
Reserved
Reserved
Reserved
Reserved

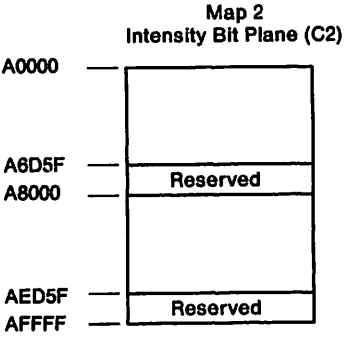
Mode Hex F



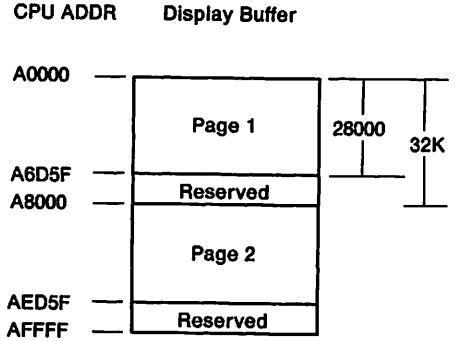


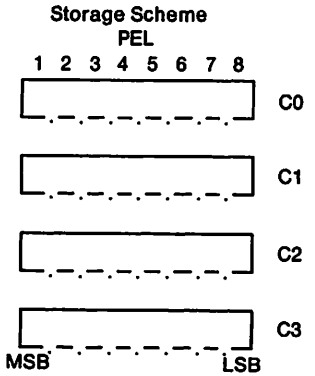
- 2 bits per PEL
- 4 attributes per PEL
- 1 bit from each bit plane (C2,C0)
- Format is first PEL is MSB of video and intensity bit planes



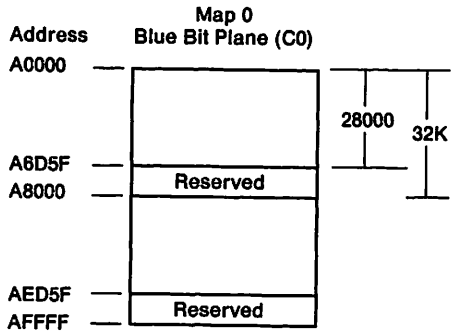


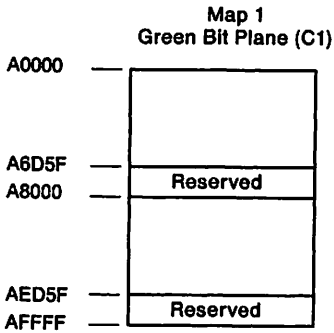
Mode Hex 10

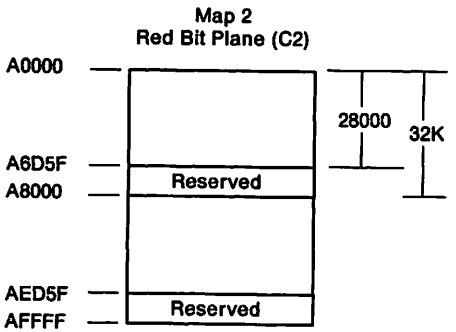


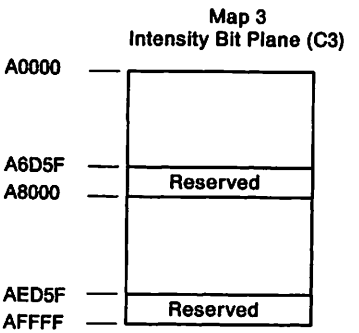


- 4 bits per PEL
- 16 colors per PEL
- 1 bit from each bit plane (C3,C2,C1,C0) per PEL
- Format is first PEL in MSB of all 4 bit planes

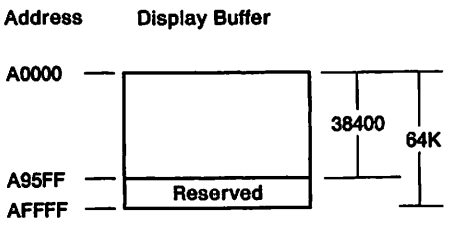


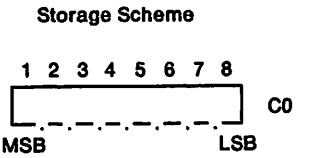




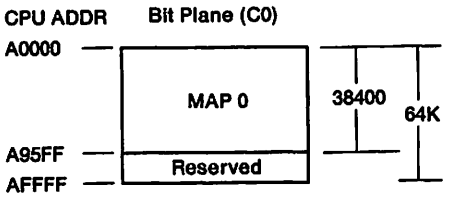


Mode Hex 11

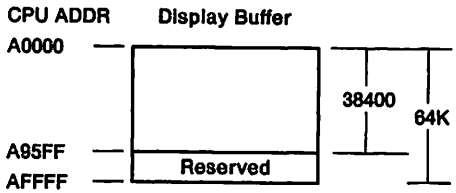


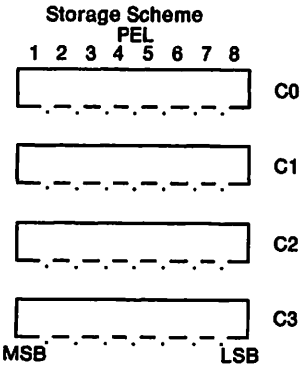


- One bit per PEL
- Two attributes per PEL
- First PEL is MSB

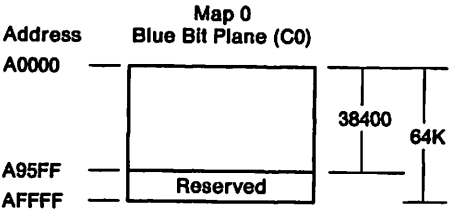


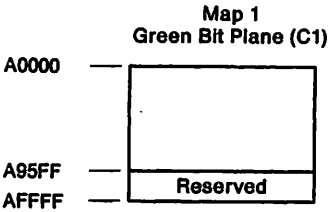
Mode Hex 12

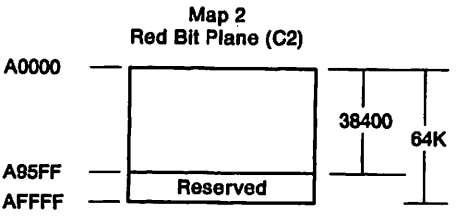


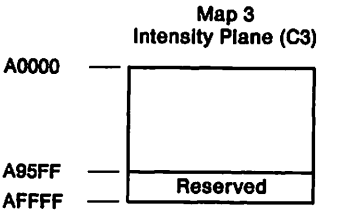


- 4 bits per PEL
- 16 colors per PEL
- 1 bit from each bit plane (C3,C2,C1,C0) per PEL
- Format is first PEL in MSB of all 4 bit planes

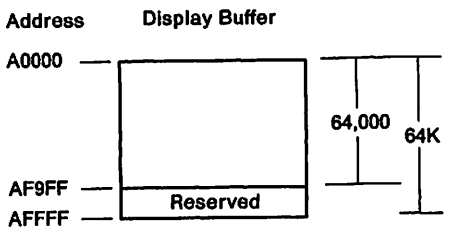




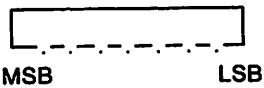




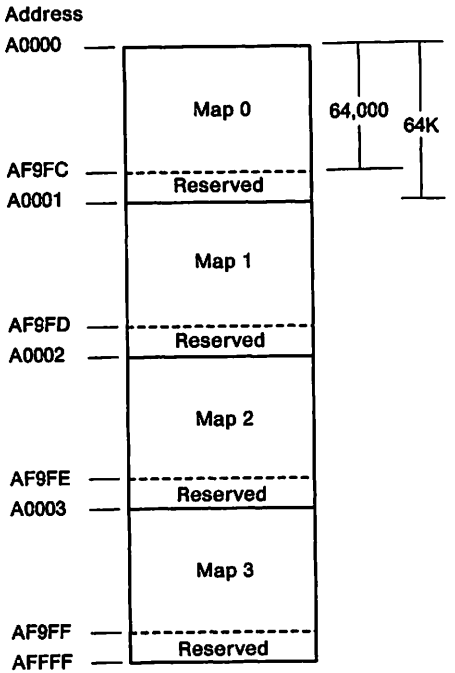
Mode Hex 13



Storage Scheme



- 8 Bits per PEL
- 256 Colors per PEL
- 1 PEL per Byte
- First PEL is at Address A0000



Memory Operations

Write Operations

When the system is writing to the display buffer, the maps are enabled by the logical decode of the memory address and the Map Mask register. The addresses used for video memory depend on the mode selected. The data flow for a system Write operation is illustrated in the following figure.

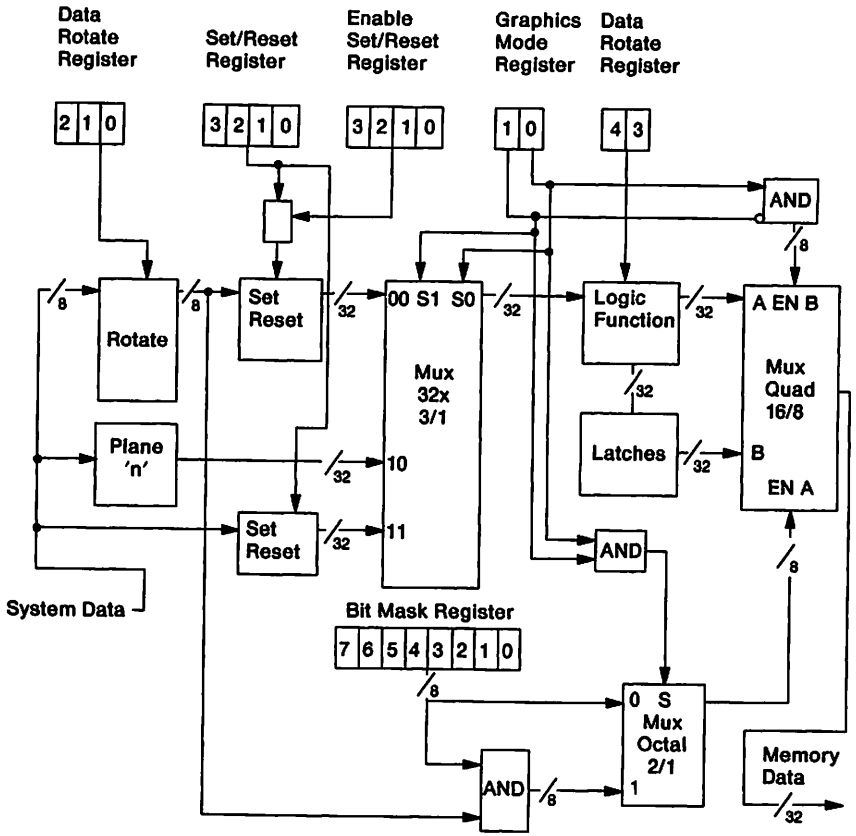


Figure 17. Data Flow for Write Operations

Read Operations

The two ways to read the video buffer are selected through the Graphics Mode register in the graphics controller. The Mode 0 Read operation returns the 8-bit value determined by the logical decode of the memory address and, if applicable, the Read Map Select register. The Mode 1 Read operation returns the 8-bit value resulting from the Color Compare operation controlled by the Color Compare and Color Don't Care registers. The data flow for the Color Compare operation is shown in the following figure.

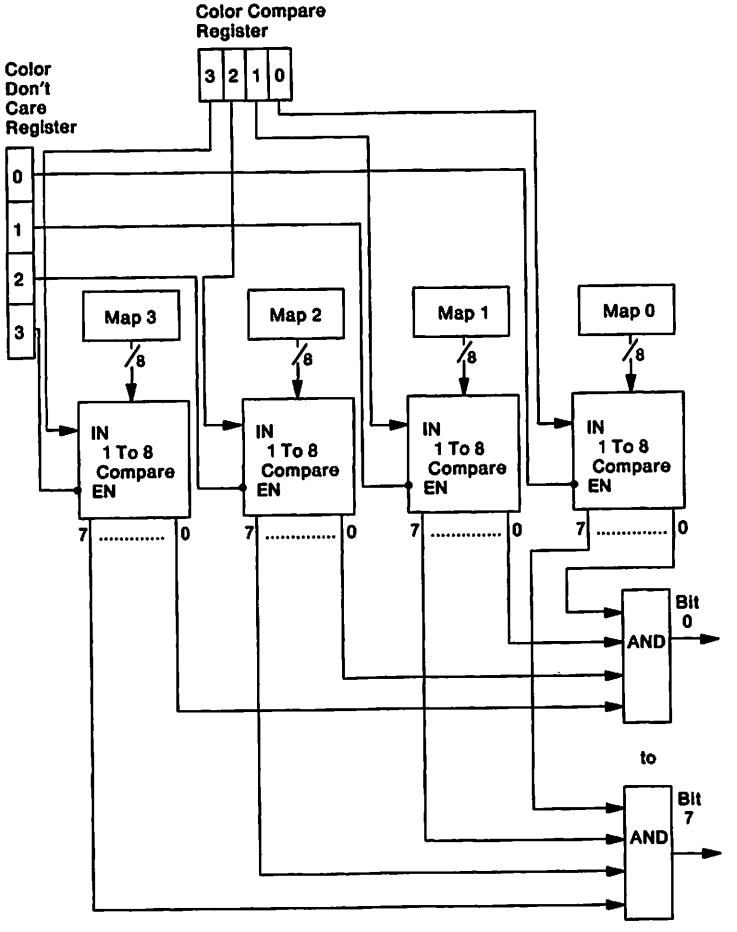


Figure 18. Color Compare Operations

Registers

There are six groups of registers in the video subsystem. All video registers are readable except the system data latches and the attribute address flip-flop. The following figure lists the register groups, their I/O addresses with the type of access (read or write), and page reference numbers.

The question mark in the address can be a hex B or D depending on the setting of the I/O address bit in the Miscellaneous Output register, described in "General Registers" on page 39.

Note: All registers in the video subsystem are read/write. The value of reserved bits in these registers must be preserved. Read the register first and change only the bits required.

Registers	R/W	Port Address	Page Reference
General Registers			39
Sequencer Registers			42
Address Register	R/W	03C4	
Data Registers	R/W	03C5	
CRT Controller Registers			49
Address Register	R/W	03?4	
Data Registers	R/W	03?5	
Graphics Controller Registers			67
Address Register	R/W	03CE	
Data Registers	R/W	03CF	
Attribute Controller Registers			74
Address Register	R/W	03C0	
Data Registers	W	03C0	
	R	03C1	
Video DAC Palette Registers			87
Write Address	R/W	03C8	
Read Address	W	03C7	
Data	R/W	03C9	
PEL Mask	R/W	03C6	

Figure 19. Video Subsystem Register Overview

General Registers

Register	Read Address	Write Address
Miscellaneous Output Register	03CC	03C2
Input Status Register 0	03C2	-
Input Status Register 1	03?A	-
Feature Control Register	03CA	03?A
Video Subsystem Enable Register	03C3	03C3

Figure 20. General Registers

Miscellaneous Output Register

The read address for this register is hex 03CC and its write address is hex 03C2.

Bit	Function
7	Vertical Sync Polarity
6	Horizontal Sync Polarity
5, 4	Reserved
3, 2	Clock Select
1	Enable RAM
0	I/O Address Select

Figure 21. Miscellaneous Output Register, Hex 03CC/03C2

Bit 7 When set to 0, this bit selects a positive 'vertical retrace' signal. This bit works with bit 6 to determine the vertical size.

Bit 6 When set to 0, this bit selects a positive 'horizontal retrace' signal. Bits 7 and 6 select the vertical size as shown in the following figure.

Bits 7 6	Vertical Size
0 0	Reserved
0 1	400 lines
1 0	350 lines
1 1	480 lines

Figure 22. Display Vertical Size

Bits 5, 4 Reserved.

Bits 3, 2 These two bits select the clock source according to the following figure. The external clock is driven through the auxiliary video extension. The input clock should be kept between 14.3 MHz and 28.4 MHz.

Bits 3 2	Function
0 0	Selects 25.175 MHz clock for 640/320 Horizontal PELs
0 1	Selects 28.322 MHz clock for 720/360 Horizontal PELs
1 0	Selects External Clock
1 1	Reserved

Figure 23. Clock Select Definitions

Bit 1 When set to 0, this bit disables address decode for the display buffer from the system.

Bit 0 This bit selects the CRT controller addresses. When set to 0, this bit sets the CRT controller addresses to hex 03Bx and the address for the Input Status Register 1 to hex 03BA for compatibility with the monochrome adapter. When set to 1, this bit sets CRT controller addresses to hex 03Dx and the Input Status Register 1 address to hex 03DA for compatibility with the color/graphics adapter. The Write addresses to the Feature Control register are affected in the same manner.

Input Status Register 0

The address for this read-only register is address hex 03C2.

Bit	Function
7	CRT Interrupt
6, 5	Reserved
4	Switch Sense Bit
3 - 0	Reserved

Figure 24. Input Status Register 0, Hex 03C2

Bit 7 When set to 1, this bit indicates a vertical retrace interrupt is pending.

Bits 6, 5 Reserved.

Bit 4 This bit is used by BIOS in determining the type of display attached.

Bits 3 - 0 Reserved.

Input Status Register 1

The address for this read-only register is address hex 03DA or 03BA.

Bit	Function
7 - 4	Reserved
3	Vertical Retrace
2, 1	Reserved
0	Display Enable

Figure 25. Input Status Register 1, Hex 03DA/03BA

Bits 7 - 4 Reserved.

Bit 3 When set to 1, this bit indicates a vertical retrace interval. This bit can be programmed, through the Vertical Retrace End register, to generate an interrupt at the start of the vertical retrace.

Bits 2, 1 Reserved.

Bit 0 When set to 1, this bit indicates a horizontal or vertical retrace interval. This bit is the real-time status of the inverted 'display enable' signal. Programs have used this status bit to restrict screen updates to the inactive display intervals in order to reduce screen flicker. The video subsystem is designed to eliminate this software requirement; screen updates may be made at any time without screen degradation.

Feature Control Register

The write address of this register is hex 03DA or 03BA; its read address is hex 03CA. All bits are reserved.

Bit	Function
7 - 0	Reserved

Figure 26. Feature Control Register, Hex 03DA/03BA and 03CA

Video Subsystem Enable Register

This register (hex 03C3) is reserved. To disable address decoding by the video subsystem, use BIOS INT 10 call, AH = hex 12, BL = hex 32.

Bit	Function
7 - 0	Reserved

Figure 27. Video Subsystem Enable Register, Hex 03C3

Sequencer Registers

The Address register is at address hex 03C4 and the data registers are at address hex 03C5. All registers within the sequencer are read/write.

Register	Index (Hex)
Sequencer Address	-
Reset	00
Clocking Mode	01
Map Mask	02
Character Map Select	03
Memory Mode	04

Figure 28. Sequencer Registers

Sequencer Address Register

The Address register is at address hex 03C4. This register is loaded with an index value that points to the desired sequencer data register.

Bit	Function
7 - 3	Reserved
2 - 0	Sequencer Address

Figure 29. Sequencer Address Register

Bits 7 - 3 Reserved.

Bits 2 - 0 These bits contain the index value that points to the data register to be accessed.

Reset Register

This read/write register has an index of hex 00; its address is hex 03C5.

Bit	Function
7 - 2	Reserved
1	Synchronous Reset
0	Asynchronous Reset

Figure 30. Reset Register, Index Hex 00

Bits 7 - 2 Reserved.

Bit 1 When set to 0, this bit commands the sequencer to synchronously clear and halt. Bits 1 and 0 must be 1 to allow the sequencer to operate. To prevent the loss of data, bit 1 must be set to 0 during the active display interval before changing the clock selection. The clock is changed through the Clocking Mode register or the Miscellaneous Output register.

Bit 0 When set to 0, this bit commands the sequencer to asynchronously clear and halt. Resetting the sequencer with this bit can cause loss of video data.

Clocking Mode Register

This read/write register has an index of hex 01; its address is hex 03C5.

Bit	Function
7, 6	Reserved
5	Screen Off
4	Shift 4
3	Dot Clock
2	Shift Load
1	Reserved
0	8/9 Dot Clocks

Figure 31. Clocking Mode Register, Index Hex 01

Bits 7, 6 Reserved.

Bit 5 When set to 1, this bit turns off the display and assigns maximum memory bandwidth to the system. Although the display is blanked, the synchronization pulses are maintained. This bit can be used for rapid full-screen updates.

Bit 4 When this bit and bit 2 are set to 0, the video serializers are loaded every character clock. When this bit is set to 1, the serializers are loaded every fourth character clock, which is useful when 32 bits are fetched per cycle and chained together in the shift registers.

The Type 2 video behaves as if this bit is set to 0; therefore, programs should set it to 0.

Bit 3 When set to 0, this bit selects the normal dot clocks derived from the sequencer master clock input. When this bit is set to 1, the master clock will be divided by 2 to generate the dot clock. All other timings are affected because they are derived from the dot clock. The dot clock divided by 2 is used for 320 and 360 horizontal PEL modes.

Bit 2 When this bit and bit 4 are set to 0, the video serializers are loaded every character clock. When this bit is set to 1, the video serializers are loaded every other character clock, which is useful when 16 bits are fetched per cycle and chained together in the shift registers.

The Type 2 video behaves as if this bit is set to 0; therefore, programs should set it to 0.

Bit 1 Reserved.

Bit 0 When set to 0, this bit directs the sequencer to generate character clocks 9 dots wide; when set to 1, it directs the sequencer to generate character clocks 8 dots wide. The 9 dot mode is for alphanumeric modes 0+, 1+, 2+, 3+, 7, and 7+ only; the 9th dot equals the 8th dot for ASCII codes hex C0 through DF. All other modes must use 8 dots per character clock. See the line graphics character bit in the Attribute Mode Control register on page 76.

Map Mask Register

This read/write register has an index of hex 02; its address is hex 03C5.

Bit	Function
7 - 4	Reserved
3	Map 3 Enable
2	Map 2 Enable
1	Map 1 Enable
0	Map 0 Enable

Figure 32. Map Mask Register, Index Hex 02

Bits 7 - 4 Reserved.

Bits 3 - 0 When set to 1, these bits enable system access to the corresponding map. If all maps are enabled, the system can write its 8-bit value to all four maps in a single memory cycle. This substantially reduces the system overhead during display updates in graphics modes.

Data scrolling operations can be enhanced by enabling all maps and writing the display buffer address with the data stored in the system data latches. This is a Read-Modify-Write operation.

When odd/even modes are selected, maps 0 and 1 and maps 2 and 3 should have the same map mask value.

When chain 4 mode is selected, all maps should be enabled.

Character Map Select Register

This register's index is hex 03; its address is hex 03C5. In alphanumeric modes, bit 3 of the attribute byte normally defines the foreground intensity. This bit can be redefined as a switch between character sets allowing 512 displayable characters. To enable this feature:

1. Set the extended memory bit in the Memory Mode register (hex 04) to 1.
2. Select different values for character map A and character map B.

This function is supported by BIOS and is a function call within the character generator routines.

Bit	Function
7, 6	Reserved
5	Character Map A Select (MSB)
4	Character Map B Select (MSB)
3, 2	Character Map A Select
1, 0	Character Map B Select

Figure 33. Character Map Select Register, Index Hex 03

Bits 7, 6 Reserved.

Bit 5 This bit is the most-significant bit for selecting the location of character map A.

Bit 4 This bit is the most-significant bit for selecting the location of character map B.

Bits 3, 2 These bits and bit 5 select the location of character map A. Map A is the area of map 2 containing the character font table used to generate characters when attribute bit 3 is set to 1. The selection is shown in the following figure.

Bits 5 3 2	Map Selected	Table Location
0 0 0	0	1st 8KB of Map 2
0 0 1	1	3rd 8KB of Map 2
0 1 0	2	5th 8KB of Map 2
0 1 1	3	7th 8KB of Map 2
1 0 0	4	2nd 8KB of Map 2
1 0 1	5	4th 8KB of Map 2
1 1 0	6	6th 8KB of Map 2
1 1 1	7	8th 8KB of Map 2

Figure 34. Character Map Select A

Bits 1, 0 These bits and bit 4 select the location of character map B. Map B is the area of map 2 containing the character font table used to generate characters when attribute bit 3 is set to 0. The selection is shown in the following figure.

Bits 4 1 0	Map Selected	Table Location
0 0 0	0	1st 8KB of Map 2
0 0 1	1	3rd 8KB of Map 2
0 1 0	2	5th 8KB of Map 2
0 1 1	3	7th 8KB of Map 2
1 0 0	4	2nd 8KB of Map 2
1 0 1	5	4th 8KB of Map 2
1 1 0	6	6th 8KB of Map 2
1 1 1	7	8th 8KB of Map 2

Figure 35. Character Map Select B

Memory Mode Register

This register's index is hex 04; its address is hex 03C5.

Bit	Function
7 - 4	Reserved
3	Chain 4
2	Odd/Even
1	Extended Memory
0	Reserved

Figure 36. Memory Mode Register, Index Hex 04

Bits 7 - 4 Reserved.

Bit 3 This bit controls the map selected during system Read operations. When set to 0, this bit enables system addresses to sequentially access data within a bit map by using the Map Mask register. When set to 1, this bit causes the two low-order bits to select the map accessed as shown in the following figure.

Address Bits A1 A0	Map Selected
0 0	0
0 1	1
1 0	2
1 1	3

Figure 37. Map Selection, Chain 4

Bit 2 When this bit is set to 0, even system addresses access maps 0 and 2, while odd system addresses access maps 1 and 3. When this bit is set to 1, system addresses sequentially access data within a bit map, and the maps are accessed according to the value in the Map Mask register (hex 02).

Bit 1 When set to 1, this bit enables the video memory from 64KB to 256KB. This bit must be set to 1 to enable the character map selection described for the previous register.

Bit 0 Reserved.

CRT Controller Registers

A data register is accessed by writing its index to the Address register at address hex 03D4 or 03B4, and then writing the data to the access port at address hex 03D5 or 03B5. The I/O address used depends on the setting of the I/O address select bit (bit 0) in the Miscellaneous Output register, which is described in "General Registers" on page 39. The following figure shows the variable part of the address as a question mark.

Note: When modifying a register, the setting of reserved bits must be preserved. Read the register first and change only the bits required.

Register	Address (Hex)	Index (Hex)
Address	03?4	-
Horizontal Total	03?5	00
Horizontal Display Enable End	03?5	01
Start Horizontal Blanking	03?5	02
End Horizontal Blanking	03?5	03
Start Horizontal Retrace Pulse	03?5	04
End Horizontal Retrace	03?5	05
Vertical Total	03?5	06
Overflow	03?5	07
Preset Row Scan	03?5	08
Maximum Scan Line	03?5	09
Cursor Start	03?5	0A
Cursor End	03?5	0B
Start Address High	03?5	0C
Start Address Low	03?5	0D
Cursor Location High	03?5	0E
Cursor Location Low	03?5	0F
Vertical Retrace Start	03?5	10
Vertical Retrace End	03?5	11
Vertical Display Enable End	03?5	12
Offset	03?5	13
Underline Location	03?5	14
Start Vertical Blanking	03?5	15
End Vertical Blanking	03?5	16
CRT Mode Control	03?5	17
Line Compare	03?5	18

Index values not listed are reserved.

Figure 38. CRT Controller Registers

Address Register

This register is at address hex 03B4 or 03D4, and is loaded with an index value that points to the data registers within the CRT controller.

Bit	Function
7 - 5	Reserved
4 - 0	Index 4 - 0

Figure 39. CRT Controller Address Register, Hex 03B4/03D4

Bits 7 - 5 Reserved.

Bits 4 - 0 These bits are the index that points to the data register accessed through address hex 03D5 or 03B5.

Horizontal Total Register

This register's index is hex 00; its address is hex 03D5 or 03B5. It defines the total number of characters in the horizontal scan interval including the retrace time. The value directly controls the period of the 'horizontal retrace' signal. A horizontal character counter in the CRT controller counts the character clock inputs; comparators are used to compare the register value with the character's horizontal width to provide horizontal timings. All horizontal and vertical timings are based on this register.

Bit	Function
7 - 0	Horizontal Total

Figure 40. Horizontal Total Register, Index Hex 00

Bits 7 - 0 The value of these bits is the total number of characters minus 5.

Horizontal Display-Enable End Register

This register's index is hex 01; its address is hex 03D5 or 03B5.

Bit	Function
7 - 0	Horizontal Display Enable End

Figure 41. Horizontal Display Enable-End Register, Index Hex 01

Bits 7 - 0 These bits define the length of the 'horizontal display-enable' signal, and determine the number of character positions per horizontal line. The value of these bits is the total number of displayed characters minus 1.

Start Horizontal Blanking Register

This register's index is hex 02; its address is hex 03D5 or 03B5.

Bit	Function
7 - 0	Start Horizontal Blanking

Figure 42. Start Horizontal Blanking Register, Index Hex 02

Bits 7 - 0 This value is the horizontal character count where the 'horizontal blanking' signal goes active.

End Horizontal Blanking Register

This register's index is hex 03; its address is hex 03D5 or 03B5. It determines when the 'horizontal blanking' signal will go active.

Bit	Function
7	Reserved
6, 5	Display Enable Skew Control
4 - 0	End Blanking

Figure 43. End Horizontal Blanking Register, Index Hex 03

Bit 7 Reserved.

Bits 6, 5 These two bits determine the amount of skew of the 'display enable' signal. This skew control is needed to provide sufficient time for the CRT controller to read a character and attribute code from the video buffer, to gain access to the character generator, and go through the Horizontal PEL Panning register in the attribute controller. Each access requires the 'display enable' signal to be

skewed one character clock so that the video output is synchronized with the horizontal and vertical retrace signals. The skew values are shown in the following figure.

Bits 6 5	Amount of Skew
0 0	No character clock skew
0 1	One character clock skew
1 0	Two character clock skew
1 1	Three character clock skew

Figure 44. Display Enable Skew

Note: Character skew is not adjustable on the Type 2 video and the bits are ignored; however, programs should set these bits for the appropriate skew to maintain compatibility.

Bits 4 - 0 These bits are the five low-order bits of a 6-bit value that is compared with the value in the Start Horizontal Blanking register to determine when the 'horizontal blanking' signal will go inactive. The most-significant bit is bit 7 in the End Horizontal Retrace register (index hex 05).

To program these bits for a signal width of W, the following algorithm is used: the width W, in character clock units, is added to the value from the Start Horizontal Blanking register. The six low-order bits of the result are the 6-bit value programmed.

Start Horizontal Retrace Pulse Register

This register's index is hex 04; its address is hex 03D5 or 03B5.

Bit	Function
7 - 0	Start Horizontal Retrace Pulse

Figure 45. Start Horizontal Retrace Pulse Register, Index Hex 04

Bits 7 - 0 These bits are used to center the screen horizontally by specifying the character position where the 'horizontal retrace' signal goes active.

End Horizontal Retrace Register

This register's index is hex 05; its address is hex 03D5 or 03B5.

Bit	Function
7	End Horizontal Blanking, Bit 5
6, 5	Horizontal Retrace Delay
4 - 0	End Horizontal Retrace

Figure 46. End Horizontal Retrace Register, Index Hex 05

Bit 7 This bit is the most-significant bit of the end horizontal blanking value in the End Horizontal Blanking register (index hex 03).

Bits 6, 5 These bits control the skew of the 'horizontal retrace' signal. The value of these bits is the amount of skew provided (from 0 to 3 character clock units). For certain modes, the 'horizontal retrace' signal takes up the entire blanking interval. Some internal timings are generated by the falling edge of the 'horizontal retrace' signal. To ensure that the signals are latched properly, the 'retrace' signal is started before the end of the 'display enable' signal and then skewed several character clock times to provide the proper screen centering.

Bits 4 - 0 These bits are compared with the Start Horizontal Retrace register to give a horizontal character count where the 'horizontal retrace' signal goes inactive.

To program these bits with a signal width of W, the following algorithm is used: the width W, in character clock units, is added to the value in the Start Retrace register. The five low-order bits of the result are the 5-bit value programmed.

Vertical Total Register

This register's index is hex 06; its address is hex 03D5 or 03B5.

Bit	Function
7 - 0	Vertical Total

Figure 47. Vertical Total Register, Index Hex 06

Bits 7 - 0 These are the eight low-order bits of a 10-bit vertical total. The value for the vertical total is the number of horizontal raster scans on the display, including vertical retrace, minus 2. This value determines the period of the 'vertical retrace' signal.

Bits 8 and 9 are in the Overflow register (index hex 07).

Overflow Register

This register's index is hex 07; its address is hex 03D5 or 03B5.

Bit	Function
7	Vertical Retrace Start, Bit 9
6	Vertical Display Enable End, Bit 9
5	Vertical Total, Bit 9
4	Line Compare, Bit 8
3	Vertical Blanking Start, Bit 8
2	Vertical Retrace Start, Bit 8
1	Vertical Display Enable End, Bit 8
0	Vertical Total, Bit 8

Figure 48. CRT Overflow Register, Index Hex 07

Bit 7 Bit 9 of the Vertical Retrace Start register (index hex 10).

Bit 6 Bit 9 of the Vertical Display Enable End register (index hex 12).

Bit 5 Bit 9 of the Vertical Total register (index hex 06).

Bit 4 Bit 8 of the Line Compare register (index hex 18).

Bit 3 Bit 8 of the Start Vertical Blanking register (index hex 15).

Bit 2 Bit 8 of the Vertical Retrace Start register (index hex 10).

Bit 1 Bit 8 of the Vertical Display Enable End register (index hex 12).

Bit 0 Bit 8 of the Vertical Total register (index hex 06).

Preset Row Scan Register

This register's index is hex 08; its address is hex 03D5 or 03B5.

Bit	Function
7	Reserved
6	Byte Panning 1
5	Byte Panning 0
4 - 0	Starting Row Scan Count

Figure 49. Preset Row Scan Register, Index Hex 08

Bit 7 Reserved.

Bits 6, 5 These two bits control byte panning in multiple shift modes. (Current BIOS modes do not use multiple shift operation.) These bits are used in PEL-panning operations, and should normally be set to 0.

The Type 2 video behaves as if these bits are set to 0; therefore, programs should set it to 0.

Bits 4 - 0 These bits specify the row scan count for the row starting after a vertical retrace. The row scan counter is incremented every horizontal retrace time until the maximum row scan occurs. When the maximum row scan is reached, the row scan counter is cleared (not preset).

Note: The CRT controller latches the start address at the start of the vertical retrace. These register values should be loaded during the active display time.

Maximum Scan Line Register

This register's index is hex 09; its address is hex 03D5 or 03B5.

Bit	Function
7	200 to 400 Line Conversion
6	Line Compare, Bit 9
5	Start Vertical Blanking, Bit 9
4 - 0	Maximum Scan Line

Figure 50. Maximum Scan Line Register, Index Hex 09

Bit 7 When this bit is set to 1, 200-scan-line video data is converted to 400-scan-line output. To do this, the clock in the row scan counter is divided by 2, which allows the 200-line modes to be displayed as 400 lines on the display (this is called double scanning; each line is displayed twice). When this bit is set to 0, the clock to the row scan counter is equal to the horizontal scan rate.

Bit 6 Bit 9 of the Line Compare register (index hex 18).

Bit 5 Bit 9 of the Start Vertical Blanking register (index hex 15).

Bits 4 - 0 These bits specify the number of scan lines per character row. The value of these bits is the maximum row scan number minus 1.

Cursor Start Register

This register's index is hex 0A; its address is hex 03D5 or 03B5.

Bit	Function
7, 6	Reserved
5	Cursor Off
4 - 0	Row Scan Cursor Begins

Figure 51. Cursor Start Register, Index Hex 0A

Bits 7, 6 Reserved.

Bit 5 When set to 1, this bit disables the cursor.

Bits 4 - 0 These bits specify the row within the character box where the cursor begins. The value of these bits is the first line of the cursor minus 1. When this value is greater than that in the Cursor End register, no cursor is displayed.

Cursor End Register

This register's index is hex 0B; its address is hex 03D5 or 03B5.

Bit	Function
7	Reserved
6, 5	Cursor Skew Control
4 - 0	Row Scan Cursor Ends

Figure 52. Cursor End Register, Index Hex 0B

Bit 7 Reserved.

Bits 6, 5 These bits control the skew of the cursor. The skew value delays the cursor by the selected number of character clocks from 0 to 3. For example, a skew of 1 moves the cursor right one position on the screen.

Bits 4 - 0 These bits specify the row within the character box where the cursor ends. If this value is less that the cursor start value, no cursor is displayed.

Start Address High Register

This register's index is hex 0C; its address is hex 03D5 or 03B5.

Bit	Function
7 - 0	High Byte of the Start Address

Figure 53. Start Address High Register, Index Hex 0C

Bits 7 - 0 These are the eight high-order bits of a 16-bit value that specifies the starting address for the regenerative buffer. The start address points to the first address after the vertical retrace on each screen refresh.

Note: The CRT controller latches the start address at the start of the vertical retrace. These register values should be loaded during the active display time.

Start Address Low Register

This register's index is hex 0D; its address is hex 03D5 or 03B5.

Bit	Function
7 - 0	Low Byte of the Start Address

Figure 54. Start Address Low Register, Index Hex 0D

Bits 7 - 0 These are the eight low-order bits of the starting address for the regenerative buffer.

Cursor Location High Register

This register's index is hex 0E; its address is hex 03D5 or 03B5.

Bit	Function
7 - 0	High Byte of the Cursor Location

Figure 55. Cursor Location High Register, Index Hex 0E

Bits 7 - 0 These are the eight high-order bits of the 16-bit cursor location.

Cursor Location Low Register

This register's index is hex 0F; its address is hex 03D5 or 03B5.

Bit	Function
7 - 0	Low Byte of the Cursor Location

Figure 56. Cursor Location Low Register, Index Hex 0F

Bits 7 - 0 These are the eight low-order bits of the cursor location.

Vertical Retrace Start Register

This register's index is hex 10; its address is hex 03D5 or 03B5.

Bit	Function
7 - 0	Vertical Retrace Pulse

Figure 57. Vertical Retrace Start Register, Index Hex 10

Bits 7 - 0 These are the eight low-order bits of the 9-bit start position for the 'vertical retrace' pulse; it is programmed in horizontal scan lines. Bit 8 is in the Overflow register (index hex 07).

Vertical Retrace End Register

This register's index is hex 11; its address is hex 03D5 or 03B5.

Bit	Function
7	Protect Registers 0-7
6	Select 5 Refresh Cycles
5	-Enable Vertical Interrupt
4	-Clear Vertical Interrupt
3 - 0	Vertical Retrace End

Figure 58. Vertical Retrace End Register, Index Hex 11

Bit 7 When set to 1, this bit disables write access to the CRT controller registers at index 00 through 07. The line compare bit in the Overflow register (index hex 07) is not protected.

Bit 6 When set to 1, this bit generates five memory refresh cycles per horizontal line. When set to 0, this bit selects three refresh cycles. Selecting five refresh cycles allows use of the VGA chip with 15.75 kHz displays. This bit

should be set to 0 for supported operations. It is set to 0 by a BIOS mode set, a reset, or a power on.

Bit 5 When set to 0, this bit enables a vertical retrace interrupt. The vertical retrace interrupt is IRQ2. This interrupt level can be shared; therefore, to determine whether the video generated the interrupt, check the CRT interrupt bit in Input Status Register 0.

Bit 4 When set to 0, this bit clears a vertical retrace interrupt. At the end of the active vertical display time, a flip-flop is set to indicate an interrupt. An interrupt handler resets this flip-flop by first setting this bit to 0, then resetting it to 1.

Bits 3 - 0 The Vertical Retrace Start register is compared with these four bits to determine where the 'vertical retrace' signal goes inactive. It is programmed in units of horizontal scan lines. To program these bits with a signal width of W, the following algorithm is used: the width W, in horizontal scan units, is added to the value in the Start Vertical Retrace register. The four low-order bits of the result are the 4-bit value programmed.

Vertical Display Enable End Register

This register's index is hex 12; its address is hex 03D5 or 03B5.

Bit	Function
7 - 0	Vertical Display Enable End

Figure 59. Vertical Display Enable End Register, Index Hex 12

Bits 7 - 0 These are the eight low-order bits of a 10-bit value that defines the vertical-display-enable end position. The two high-order bits are contained in the Overflow register (index hex 07). The 10-bit value is equal to the total number of scan lines minus 1.

Offset Register

This register's index is hex 13; its address is hex 03D5 or 03B5.

Bit	Function
7 - 0	Logical Line Width of the Screen

Figure 60. Offset Register, Index Hex 13

Bits 7 - 0 These bits specify the logical line width of the screen. The starting memory address for the next character row is larger than the current character row by 2 or 4 times the value of these bits. Depending on the method of clocking the CRT controller, this address is either a word or doubleword address.

Underline Location Register

This register's index is hex 14; its address is hex 03D5 or 03B5.

Bit	Function
7	Reserved
6	Doubleword Mode
5	Count By 4
4 - 0	Start Underline

Figure 61. Underline Location Register, Index Hex 14

Bit 7 Reserved.

Bit 6 When this bit is set to 1, memory addresses are doubleword addresses. See the description of the word/byte mode bit (bit 6) in the CRT Mode Control register on page 63.

Bit 5 When this bit is set to 1, the memory-address counter is clocked with the character clock divided by 4, which is used when doubleword addresses are used.

Bits 4 - 0 These bits specify the horizontal scan line of a character row on which an underline occurs. The value programmed is the scan line desired minus 1.

Start Vertical Blanking Register

This register's index is hex 15; its address is hex 03D5 or 03B5.

Bit	Function
7 - 0	Start Vertical Blanking

Figure 62. Start Vertical Blanking Register, Index Hex 15

Bits 7 - 0 These are the eight low-order bits of a 10-bit value that specifies the starting location for the 'vertical blanking' signal. Bit 8 is in the Overflow register (index hex 07) and bit 9 is in the Maximum Scan Line register (index hex 09). The 10-bit value is the horizontal scan line count where the 'vertical blanking' signal becomes active minus 1.

End Vertical Blanking Register

This register's index is hex 16; its address is hex 03D5 or 03B5.

Bit	Function
7 - 0	End Vertical Blanking

Figure 63. End Vertical Blanking Register, Index Hex 16

Bits 7 - 0 This register specifies the horizontal scan count where the 'vertical blanking' signal becomes inactive. The register is programmed in units of the horizontal scan line.

To program these bits with a 'vertical blanking' signal of width W, the following algorithm is used: the width W, in horizontal scan line units, is added to the value in the Start Vertical Blanking register minus 1. The eight low-order bits of the result are the 8-bit value programmed.

CRT Mode Control Register

This register's index is hex 17; its address is hex 03D5 or 03B5.

Bit	Function
7	Hardware Reset
6	Word/Byte Mode
5	Address Wrap
4	Reserved
3	Count By Two
2	Horizontal Retrace Select
1	Select Row Scan Counter
0	CMS 0

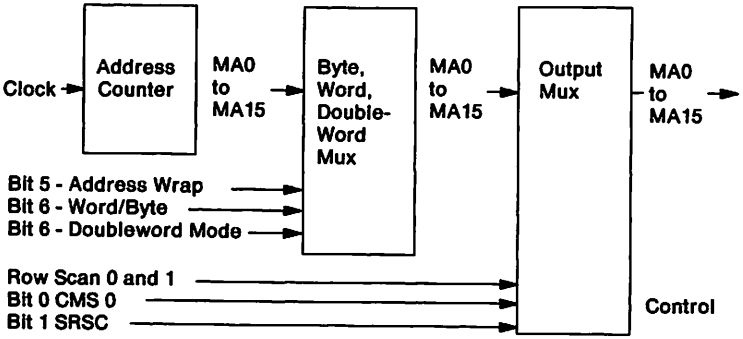
Figure 64. CRT Mode Control Register, Index Hex 17

Bit 7 When set to 0, this bit disables the horizontal and vertical retrace signals and forces them to an inactive level. When set to 1, this bit enables the horizontal and vertical retrace signals. This bit does not reset any other registers or signal outputs.

Bit 6 When this bit is set to 0, the word mode is selected. The word mode shifts the memory-address counter bits down one bit; the most-significant bit of the counter appears on the least-significant bit of the memory address outputs.

The doubleword bit in the Underline Location register (hex 14) also controls the addressing. When the doubleword bit is 0, the word/byte bit selects the mode. When the doubleword bit is set to 1, the addressing is shifted by two bits.

When set to 1, bit 6 selects the byte address mode. See the following figures for address output details.



Memory Address Outputs	Modes of Addressing		
	Byte	Word	Doubleword
MA 0/RFA 0	MA 0	MA 15 or 13	MA 12
MA 1/RFA 1	MA 1	MA 0	MA 13
MA 2/RFA 2	MA 2	MA 1	MA 0
MA 3/RFA 3	MA 3	MA 2	MA 1
MA 4/RFA 4	MA 4	MA 3	MA 2
MA 5/RFA 5	MA 5	MA 4	MA 3
MA 6/RFA 6	MA 6	MA 5	MA 4
MA 7/RFA 7	MA 7	MA 6	MA 5
MA 8/RFA 8	MA 8	MA 7	MA 6
MA 9	MA 9	MA 8	MA 7
MA 10	MA 10	MA 9	MA 8
MA 11	MA 11	MA 10	MA 9
MA 12	MA 12	MA 11	MA 10
MA 13	MA 13	MA 12	MA 11
MA 14	MA 14	MA 13	MA 12
MA 15	MA 15	MA 14	MA 13

Figure 65. CRT Memory Address Mapping

Bit 5 This bit selects the memory-address bit, bit MA 13 or MA 15, that appears on the output pin MA 0, in the word address mode. If the VGA is not in the word address mode, bit 0 from the address counter appears on the output pin, MA 0.

When set to 1, this bit selects MA 15. In odd/even mode, this bit should be set to 1 because 256KB of video memory is installed on the system board. (Bit MA 13 is selected in applications where only 64KB is present. This function

maintains compatibility with the IBM Color/Graphics Monitor Adapter.)

Bit 4 Reserved.

Bit 3 When this bit is set to 0, the address counter uses the character clock. When this bit is set to 1, the address counter uses the character clock input divided by 2. This bit is used to create either a byte or word refresh address for the display buffer.

Bit 2 This bit selects the clock that controls the vertical timing counter. The clocking is either the horizontal retrace clock or horizontal retrace clock divided by 2. When this bit is set to 1, the horizontal retrace clock is divided by 2.

Dividing the clock effectively doubles the vertical resolution of the CRT controller. The vertical counter has a maximum resolution of 1024 scan lines because the vertical total value is 10-bits wide. If the vertical counter is clocked with the horizontal retrace divided by 2, the vertical resolution is doubled to 2048 scan lines.

Bit 1 This bit selects the source of bit 14 of the output multiplexer. When this bit is set to 0, bit 1 of the row scan counter is the source. When this bit is set to 1, the bit 14 of the address counter is the source.

Bit 0 This bit selects the source of bit 13 of the output multiplexer. When this bit is set to 0, bit 0 of the row scan counter is the source, and when this bit is set to 1, bit 13 of the address counter is the source.

The CRT controller used on the IBM Color/Graphics Adapter was capable of using 128 horizontal scan-line addresses. For the VGA to obtain 640-by-200 graphics resolution, the CRT controller is programmed for 100 horizontal scan lines with two scan-line addresses per character row. Row scan address bit 0 becomes the most-significant address bit to the display buffer. Successive scan lines of the display image are displaced in 8KB of memory. This bit allows compatibility with the graphics modes of earlier adapters.

Line Compare Register

This register's index is hex 18; its address is hex 03D5 or 03B5.

Bit	Function
7 - 0	Line Compare Target

Figure 66. Line Compare Register, Index Hex 18

Bits 7 - 0 These bits are the eight low-order bits of the 10-bit compare target. When the vertical counter reaches the target value, the internal start address of the line counter is cleared. This creates a split screen where the lower screen is immune to scrolling. Bit 8 is in the Overflow register (index hex 07), and bit 9 is in the Maximum Scan Line register (index hex 09).

Graphics Controller Registers

The Address register for the graphics controller is at address hex 03CE. The data registers are at address hex 03CF. All registers are read/write.

Register Name	Address (Hex)	Index (Hex)
Address	03CE	
Set/Reset	03CF	00
Enable Set/Reset	03CF	01
Color Compare	03CF	02
Data Rotate	03CF	03
Read Map Select	03CF	04
Graphics Mode	03CF	05
Miscellaneous	03CF	06
Color Don't Care	03CF	07
Bit Mask	03CF	08

Figure 67. Graphics Controller Register Overview

Address Register

The Address register is at address hex 03CE. This register is loaded with the index value that points to the desired data register within the graphics controller.

Bit	Function
7 - 4	Reserved
3 - 0	Register Index

Figure 68. Graphics Controller Address Register, Hex 03CE

Bits 7 - 4 Reserved.

Bits 3 - 0 These bits contain the index value that points to the data registers.

Set/Reset Register

This register's index is hex 00; its address is hex 03CF.

Bit	Function
7 - 4	Reserved
3	Set/Reset Map 3
2	Set/Reset Map 2
1	Set/Reset Map 1
0	Set/Reset Map 0

Figure 69. Set/Reset Register, Index Hex 00

Bits 7 - 4 Reserved.

Bits 3 - 0 When write mode 0 is selected, the system writes the value of each set/reset bit to its respective memory map. For each Write operation, the set/reset bit, if enabled, is written to all eight bits within that map. Set/reset operation can be enabled on a map-by-map basis through the Enable Set/Reset register.

Enable Set/Reset Register

The index for this register is hex 01; its address is hex 03CF.

Bit	Function
7 - 4	Reserved
3	Enable Set/Reset Map 3
2	Enable Set/Reset Map 2
1	Enable Set/Reset Map 1
0	Enable Set/Reset Map 0

Figure 70. Enable Set/Reset Register, Index Hex 01

Bits 7 - 4 Reserved.

Bits 3 - 0 These bits enable the set/reset function used when write mode 0 is selected in the Graphics Mode register (index hex 05). When the bit is set to 1, the respective memory map receives the value specified in the Set/Reset register. When Set/Reset is not enabled for a map, that map receives the value sent by the system.

Color Compare Register

This register's index is hex 02; its address is hex 03CF.

Bit	Function
7 - 4	Reserved
3	Color Compare Map 3
2	Color Compare Map 2
1	Color Compare Map 1
0	Color Compare Map 0

Figure 71. Color Compare Register, Index Hex 02

Bits 7 - 4 Reserved.

Bits 3 - 0 These bits are the 4-bit color value to be compared when the read mode bit in the Graphics Mode register is set to 1. When the system does a memory read, the data returned from the memory cycle will be a 1 in each bit position where the four maps equal the Color Compare register. If the read mode bit is 0, the data is returned without comparison.

The color compare bit is the value that all bits of the corresponding map's byte are compared with. Each of the eight bit positions in the selected byte are compared across the four maps, and a 1 is returned in each position where the bits of all four maps equal their respective color compare values.

Data Rotate Register

This register's index is hex 03; its address is hex 03CF.

Bit	Function
7 - 5	Reserved
4, 3	Function Select
2 - 0	Rotate Count

Figure 72. Data Rotate Register, Index Hex 03

Bits 7 - 5 Reserved.

Bits 4, 3 Data written to the video buffer can be operated on logically by data already in the system latches.

Data can be any of the choices selected by the write mode bits except system latches, which cannot be modified. If rotated data is selected also, the rotate is performed

before the logical operation. The logical operations selected are shown in the following table.

Bits 4 3	Function
0 0	Data Unmodified
0 1	Data ANDed with Latched Data
1 0	Data ORed with Latched Data
1 1	Data XORed with Latched Data

Figure 73. Operation Select Bit Definitions

Bits 2 - 0 In write mode 0, these bits select the number of positions the system data is rotated to the right during a system Memory Write operation. To write data that is not rotated in mode 0, all bits are set to 0.

Read Map Select Register

This register's index is hex 04; its address is hex 03CF.

Bit	Function
7 - 2	Reserved
1, 0	Map Select

Figure 74. Read Map Select Register, Index Hex 04

Bits 7 - 2 Reserved.

Bits 1, 0 These bits select the memory map for system Read operations. This register has no effect on the color compare read mode. In odd/even modes, the value can be a binary 00 or 01 to select the chained maps 0, 1 and the value can be a binary 10 or 11 to select the chained maps 2, 3.

Graphics Mode Register

This register's index is hex 05; its address is hex 03CF.

Bit	Function
7	Reserved
6	256-Color Mode
5	Shift Register Mode
4	Odd/Even
3	Read Mode
2	Reserved
1, 0	Write Mode

Figure 75. Graphics Mode Register, Index Hex 05

Bit 7 Reserved.

Bit 6 When set to 0, this bit allows bit 5 to control the loading of the shift registers. When set to 1, this bit causes the shift registers to be loaded in a manner that supports the 256-color mode.

Bit 5 When set to 1, this bit directs the shift registers in the graphics controller to format the serial data stream with even-numbered bits from both maps on even-numbered maps, and odd-numbered bits from both maps on the odd-numbered maps. This bit is used for modes 4 and 5.

Bit 4 When set to 1, this bit selects the odd/even addressing mode used by the IBM Color/Graphics Monitor Adapter. Normally, the value here follows the value of Memory Mode register bit 2 in the sequencer.

Bit 3 When this bit is set to 1, the system reads the results of the comparison of the four memory maps and the Color Compare register.

When this bit is set to 0, the system reads data from the memory map selected by the Read Map Select register, or by the two low-order bits of the memory address (this selection depends on the chain-4 bit in the Memory Mode register of the sequencer).

Bit 2 Reserved.

Bits 1, 0 The write mode selected and its operation are defined in the following figure. The logic operation specified by the function select bits is performed on system data for modes 0, 2, and 3.

Bits 1 0	Mode Description
0 0	Each memory map is written with the system data rotated by the count in the Data Rotate register. If the set/reset function is enabled for a specific map, that map receives the 8-bit value contained in the Set/Reset register.
0 1	Each memory map is written with the contents of the system latches. These latches are loaded by a system Read operation.
1 0	Memory map *n* (0 through 3) is filled with eight bits of the value of data bit *n*.
1 1	Each memory map is written with the 8-bit value contained in the Set/Reset register for that map (the Enable Set/Reset register has no effect). Rotated system data is ANDed with the Bit Mask register to form an 8-bit value that performs the same function as the Bit Mask register in write modes 0 and 2 (see also Bit Mask register on page 73).

Figure 76. Write Mode Definitions

Miscellaneous Register

This register's index is hex 06; its address is hex 03CF.

Bit	Function
7 - 4	Reserved
3	Memory Map 1
2	Memory Map 0
1	Odd/Even
0	Graphics Mode

Figure 77. Miscellaneous Register, Index Hex 06

Bits 7 - 4 Reserved.

Bits 3, 2 These bits control the mapping of the regenerative buffer into the system address space. The bit functions are defined in the following figure.

Bits 3 2	Addressing Assignment
0 0	A0000 for 128KB
0 1	A0000 for 64KB
1 0	B0000 for 32KB
1 1	B8000 for 32KB

Figure 78. Video Memory Assignments

Bit 1 When set to 1, this bit directs the system address bit, A0, to be replaced by a higher-order bit. The odd map is then selected when A0 is 1, and the even map when A0 is 0.

Bit 0 This bit controls alphanumeric mode addressing. When set to 1, this bit selects graphics modes, which also disables the character generator latches.

Color Don't Care Register

This register's index is hex 07; its address is hex 03CF.

Bit	Function
7 - 4	Reserved
3	Map 3 is Don't Care
2	Map 2 is Don't Care
1	Map 1 is Don't Care
0	Map 0 is Don't Care

Figure 79. Color Don't Care Register, Index Hex 07

Bits 7 - 4 Reserved.

Bits 3 - 0 These bits select whether a map is going to participate in the color compare cycle. When the bit is set to 1, the bits in that map are compared.

Bit Mask Register

This register's index is hex 08; its address is hex 03CF.

Bit	Function
7 - 0	-Bit Mask 7 - 0

Figure 80. Bit Mask Register, Index Hex 08

Bits 7 - 0 When the bit is set to 1, the corresponding bit position in each map can be changed. When the bit set to 0, the bit position in the map is masked to prevent change, provided that the location being written was the last location read by the system microprocessor.

The bit mask applies to write modes 0 and 2. To preserve bits using the bit mask, data must be latched internally by reading the location. When data is written to preserve the bits, the most current data in the latches is written in those positions. The bit mask applies to all maps simultaneously.

Attribute Controller Registers

Each register for the attribute controller has two addresses. Address hex 03C0 is the write address and hex 03C1 is the read address. The individual data registers are selected by writing their index to the Address register (hex 03C0).

Register Name	Write Address	Read Address	Index
Address	03C0	03C0	-
Internal Palette	03C0	03C1	00 - 0F
Attribute Mode Control			10
Overscan Color			11
Color Plane Enable			12
Horizontal PEL Panning			13
Color Select			14

Figure 81. Attribute Controller Register Addresses

Address Register

This read/write register is at address hex 03C0.

The attribute controller registers do not have an input bit to control selection of the address and data registers. An internal address flip-flop controls this selection. Reading Input Status Register 1 clears the flip-flop and selects the Address register.

After the Address register has been loaded with the index, the next Write operation to 03C0 loads the data register. The flip-flop toggles for each Write operation to address hex 03C0. It does not toggle for Read operations to 03C0 or 03C1. (Also see "VGA Programming Considerations" on page 80.)

Bit	Function
7, 6	Reserved
5	Internal Palette Address Source
4 - 0	Register Index

Figure 82. Address Register, Hex 03C0

Bits 7, 6 Reserved.

Bit 5 This bit is set to 0 to load color values to the registers in the internal palette. It is set to 1 for normal operation of the attribute controller.

Note: Do not access the internal palette while this bit is set to 1. While this bit is 1, the Type 1 video

subsystem disables accesses to the palette; however, the Type 2 does not, and the actual color value addressed cannot be ensured.

Bits 4 - 0 These bits contain the index to the data registers in the attribute controller.

Internal Palette Registers 0 through F

These registers are at indexes hex 00 through 0F. Their write address is hex 03C0; their read address is hex 03C1.

Bit	Function
7, 6	Reserved
5	P5
4	P4
3	P3
2	P2
1	P1
0	P0

Figure 83. Internal Palette Registers, Index Hex 00 - 0F

Bits 7, 6 Reserved.

Bits 5 - 0 These 6-bit registers allow a dynamic mapping between the text attribute or graphic color input value and the display color on the CRT screen. When set to 1, this bit selects the appropriate color. The Internal Palette registers should be modified only during the vertical retrace interval to avoid problems with the displayed image. These internal palette values are sent off-chip to the video DAC, where they serve as addresses into the DAC registers. (Also see the attribute controller block diagram on page 7.)

Note: These registers can be accessed only when bit 5 in the Address register is set to 0. When the bit is 1, writes are "don't care" and reads return undefined data.

Attribute Mode Control Register

This read/write register is at index hex 10. Its write address is hex 03C0; its read address is hex 03C1.

Bit	Function
7	P5, P4 Select
6	PEL Width
5	PEL Panning Compatibility
4	Reserved
3	Enable Blink/-Select Background Intensity
2	Enable Line Graphics Character Code
1	Mono Emulation
0	Graphics/-Alphanumeric Mode

Figure 84. Attribute Mode Control Register, Index Hex 10

Bit 7 This bit selects the source for the P5 and P4 video bits that act as inputs to the video DAC. When this bit is set to 0, P5 and P4 are the outputs of the Internal Palette registers. When this bit is set to 1, P5 and P4 are bits 1 and 0 of the Color Select register. For more information, refer to "VGA Programming Considerations" on page 80.

Bit 6 When this bit is set to 1, the video data is sampled so that eight bits are available to select a color in the 256-color mode (hex 13). This bit is set to 0 in all other modes.

Bit 5 When this bit is set to 1, a successful line-compare in the CRT controller forces the output of the PEL Panning register to 0 until a vertical synchronization occurs, at which time the output returns to its programmed value. This bit allows a selected portion of a screen to be panned.

When this bit is set to 0, line compare has no effect on the output of the PEL Panning register.

Bit 4 Reserved.

Bit 3 When this bit is set to 0, the most-significant bit of the attribute selects the background intensity (allows 16 colors for background). When set to 1, this bit enables blinking.

Bit 2 When this bit is set to 0, the ninth dot will be the same as the background. When set to 1, this bit enables the special line-graphics character codes for the monochrome emulation mode. This emulation mode forces the ninth dot of a line graphic character to be identical to the eighth

dot of the character. The line-graphics character codes for the monochrome emulation mode are hex C0 through hex DF.

For character fonts that do not utilize these line-graphics character codes, bit 2 should be set to 0 to prevent unwanted video information from displaying on the CRT screen.

BIOS will set this bit, the correct dot clock, and other registers when the 9-dot alphanumeric mode is selected.

Bit 1 When this bit is set to 1, monochrome emulation mode is selected. When this bit is set to 0, color emulation mode is selected.

Bit 0 When set to 1, this bit selects the graphics mode of operation.

Overscan Color Register

This read/write register is at index hex 11. Its write address is hex 03C0; its read address is hex 03C1. This register determines the border (overscan) color.

Bit	Function
7 - 0	P7 - P0

Figure 85. Overscan Color Register, Index Hex 11

Bits 7 - 0 These bits select the border color used in the 80-column alphanumeric modes and in the graphics modes other than modes 4, 5, and D.

Color Plane Enable Register

This read/write register is at index hex 12. Its write address is hex 03C0; its read address is hex 03C1.

Bit	Function
7 - 4	Reserved
3 - 0	Enable Color Plane

Figure 86. Color Plane Enable Register, Index Hex 12

Bits 7 - 4 Reserved.

Bits 3 - 0 Setting a bit to 1, enables the corresponding display-memory color plane.

Horizontal PEL Panning Register

This read/write register is at index hex 13. Its write address is hex 03C0; its read address is hex 03C1.

Bit	Function
7 - 4	Reserved
3 - 0	Horizontal PEL Panning

Figure 87. Horizontal PEL Panning Register, Index Hex 13

Bits 7 - 4 Reserved.

Bits 3 - 0 These bits select the number of PELs that the video data is shifted to the left. PEL panning is available in both alphanumeric and graphics modes. The following figure shows the number of bits shifted for each mode.

	Number of PELs Shifted to the Left		
Register Value	Mode 13	A/N Modes *	All Other Modes
0	0	1	0
1	-	2	1
2	1	3	2
3	-	4	3
4	2	5	4
5	-	6	5
6	3	7	6
7	-	8	7
8	-	0	-
* Only mode 7 and the A/N modes with 400 scan lines.			

Figure 88. Image Shifting

Color Select Register

This read/write register is at index hex 14. Its write address is hex 03C0; its read address is hex 03C1.

Bit	Function
7 - 4	Reserved
3	S_color 7
2	S_color 6
1	S_color 5
0	S_color 4

Figure 89. Color Select Register, Index Hex 14

Bits 7 - 4 Reserved.

Bits 3, 2 In modes other than mode hex 13, these are the two most-significant bits of the 8-bit digital color value to the video DAC. In mode hex 13, the 8-bit attribute is the digital color value to the video DAC. These bits are used to rapidly switch between sets of colors in the video DAC. (For more information, refer to "VGA Programming Considerations" on page 80.)

Bits 1, 0 These bits can be used in place of the P4 and P5 bits from the Internal Palette registers to form the 8-bit digital color value to the video DAC. Selecting these bits is done in the Attribute Mode Control register (index hex 10). These bits are used to rapidly switch between colors sets within the video DAC.

VGA Programming Considerations

The following are some programming considerations for the VGA:

- The following rules must be followed to guarantee the critical timings necessary to ensure the proper operation of the CRT controller:
  - The value in the Horizontal Total register must be at least hex 19.
  - The minimum positive pulse width of the 'horizontal synchronization' signal must be four character clock units.
  - The End Horizontal Retrace register must be programmed such that the 'horizontal synchronization' signal goes to 0 at least one character clock time before the 'horizontal display enable' signal goes active.
  - The End Vertical Blanking register must be set to a minimum of one horizontal scan line greater than the line-compare value.
- When PEL panning compatibility is enabled in the Attribute Mode Control register, a successful line compare in the CRT controller forces the output of the Horizontal PEL Panning register to 0's until a vertical synchronization occurs. When the vertical synchronization occurs, the output returns to the programmed value. This allows the portion of the screen indicated by the Line Compare register to be operated on by the Horizontal PEL Panning register.
- A write to the Character Map Select register becomes valid on the next whole character line. This will prevent deformed character images when changing character generators in the middle of a character scan line.
- For mode hex 13, the attribute controller is configured so that the 8-bit attribute in video memory becomes the 8-bit address (P0 - P7) into the video DAC. The user should not modify the contents of the Internal Palette registers when using this mode.

- The following is the sequence for accessing the attribute data registers:
  1. Disable interrupts.
  2. Reset the flip-flop for the Attribute Address register.
  3. Write the index.
  4. Access the data register.
  5. Enable interrupts.
- The Color Select register in the attribute controller section allows the programmer to rapidly switch color sets in the video DAC. Bit 7 of the Attribute Mode Control register controls the number of bits in the Color Select register used to address the color information in the video DAC (either two or four bits are used). By changing the value in the Color Select register, an application can switch color sets in graphics and alphanumeric modes (mode hex 13 does not use this feature).

  Note: For multiple color sets, the user must load the color values.
- An application that saves the video state must store the four bytes of information contained in the system microprocessor latches in the graphics controller subsection. These latches are loaded with 32 bits from video memory (8 bits per map) each time the system reads from video memory. The application needs to:
  1. Use write mode 1 to write the values in the latches to a location in video memory that is not part of the display buffer, such as the last location in the address range.
  2. Save the values of the latches by reading them back from video memory.

     Note: If memory addressing is in the chain-4 or odd/even mode, reconfigure the memory as four sequential maps prior to performing the sequence above.

BIOS provides support for completely saving and restoring the video state. Refer to the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference* for more information.

- The Horizontal PEL Panning register allows programs to control the starting position of the display area on the screen. The display area can be shifted to the left up to eight PEL positions. In single-byte shift modes, to pan to the PEL position above 8, the CRT controller start address is incremented and the PEL Panning register is reset to 0.

  In multiple shift modes, the byte-panning bits (in the Reset Row Scan register) are used as extensions to the Horizontal PEL Panning register. This allows panning across the width of the video output. For example, in the 32-bit shift mode, the byte pan and PEL-panning bits provide panning up to 31 bits. To pan from position 31 to 32, the CRT controller start address is incremented and the panning bits, both PEL and byte, are reset to 0.

  Further panning can be accomplished by changing the start-address value in the CRT controller registers, Start Address High and Start Address Low. The sequence is:

  1. Use the Horizontal PEL Panning register to shift the maximum number of bits to the left.
  2. Increment the start address.
  3. Set the Horizontal PEL Panning register so that no bits are shifted.

     The screen will now be shifted one PEL to the left of the position it was in at the end of Step 1. Step 1 through Step 3 are repeated as often as necessary.

- When using a split-screen application that scrolls a second screen on top of the first screen and operating in a mode with 200 scan lines, the Line Compare register (CRT Controller register hex 19) must contain an even value. This is a requirement of the double scanning logic in the CRT controller.
- If the value in the Cursor Start register (CRT Controller register hex 0A) is greater than that in the Cursor End register (CRT Controller register hex 0B), the cursor is not displayed.
- In 8-dot character modes, the underline attribute produces a solid line across adjacent characters. In 9-dot character modes, the underline across adjacent characters is dashed. In 9-dot modes with the line-graphics characters (C0 - DF character codes), the underline is solid.

Programming the Registers

Each of the video components has an address register and a number of data registers. The data registers have addresses common to all registers for that component. The individual registers are selected by a pointer (index) in its Address register. To write to a data register, the Address register is loaded with the index of the desired data register, then the data register is loaded by writing to the common I/O address.

The general registers do not share a common address; they each have their own I/O address.

See "Video DAC to System Interface" on page 87 for details on programming the video DAC.

For compatibility with the IBM Enhanced Graphics Adapter (EGA), the internal video subsystem palette is programmed the same as the EGA. Using BIOS to program the palette will produce a color compatible to that produced by the EGA. Mode hex 13 (256 colors) is programmed so that the first 16 locations in the DAC produce compatible colors.

When BIOS is used to load the color palette for a color mode and a monochrome display is attached, the color palette is changed. The colors are summed to produce shades of gray that allow color applications to produce a readable screen.

Modifying the following bits must be done while the sequencer is held in a synchronous reset through its Reset register. The bits are:

- Bits 3 and 0 of the Clocking Mode register.
- Bits 3 and 2 of the Miscellaneous Output register.

RAM Loadable Character Generator

The character generator is RAM loadable and can support characters up to 32 scan lines high. Three character fonts are stored in BIOS, and one is automatically loaded when an alphanumeric mode is selected. The Character Map Select register can be programmed to redefine the function of bit 3 of the attribute byte to be a character-font switch. This allows the user to select between any two character sets residing in map 2, and effectively gives the user access to 512 characters instead of 256. Character fonts can be loaded off-line, and up to eight fonts can be loaded at any one time.

The structure of the character fonts is described in the following figure. The character generator is in map 2 and must be protected using the map mask function.

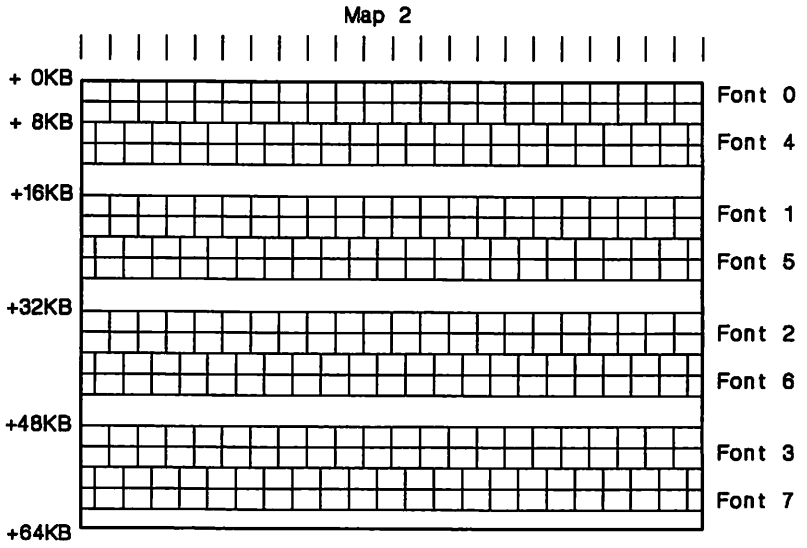


Figure 90. Character Table Structure

The following figure illustrates the structure of each character pattern. If the CRT controller is programmed to generate 16 row scans, then 16 bytes must be filled in for each character in the font. The example below assumes eight row scans per character.

Address	Byte Image								Data
CC * 32 + 0				X	X				18H
1			X	X	X	X			3EH
2		X	X			X	X		66H
3		X	X			X	X		66H
4		X	X	X	X	X	X		7EH
5		X	X			X	X		66H
6		X	X			X	X		66H
7		X	X			X	X		66H

Figure 91. Character Pattern Example

CC equals the value of the character code. For example, hex 41 equals an ASCII "A."

Creating a Split Screen

The VGA hardware supports a split screen. The top portion of the screen is designated as screen A, and the bottom portion is designated as screen B, as in the following figure.

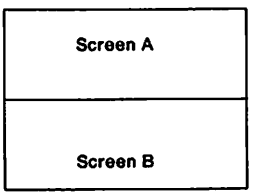


Figure 92. Split Screen Definition

The following figure shows the screen mapping for a system containing a 32KB alphanumeric storage buffer, such as the VGA. Information displayed on screen A is defined by the Start Address

High and Low registers of the CRT controller. Information displayed on screen B always begins at video address hex 0000.

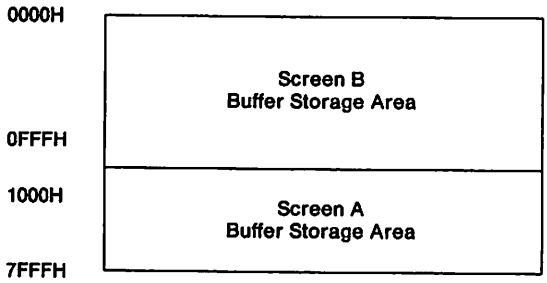


Figure 93. Screen Mapping within the Display Buffer Address Space

The Line Compare register of the CRT controller performs the split screen function. The CRT controller has an internal horizontal scan line counter and logic that compares the counter value to the value in the Line Compare register and clears the memory address generator when a comparison occurs. The linear address generator then sequentially addresses the display buffer starting at location 0. Each subsequent row address is determined by the 16-bit addition of the start-of-line latch and the Offset register.

Screen B can be smoothly scrolled onto the display by updating the Line Compare register in synchronization with the 'vertical retrace' signal. Screen B information is not affected by scrolling operations that use the Start Address registers to scroll through the screen A information.

When PEL-panning compatibility is enabled (Attribute Mode Control register), a successful line comparison forces the output of the Horizontal PEL Panning register to 0's until vertical synchronization occurs. This feature allows the information on screen B to remain unaffected by PEL-panning operations on screen A.

Video Digital-to-Analog Converter

The video digital-to-analog converter (DAC) integrates the function of a color palette with three internal DACs for driving an analog display.

The DAC has 256 registers containing 18 bits each to allow the display of up to 256 colors from a possible 256K colors. Each output signal is driven by a 6-bit DAC.

Register Name	R/W	Address (In Hex)
Palette Address (Write Mode)	R/W	03C8
Palette Address (Read Mode)	W	03C7
DAC State	R	03C7
Palette Data	R/W	03C9
PEL Mask	R	03C6

Figure 94. Video DAC Register

Device Operation

The palette address (P7 - P0) and the blanking input are sampled on the rising edge of the PEL clock. After three more PEL clock cycles, the video reflects the state of these inputs.

During normal operation the palette address is used as a pointer to one of the 256 data registers in the palette. The value in each data register is converted to an analog signal for each of the three outputs (red, green, blue). The blanking input is used to force the video output to 0 volts. The blanking operation is independent of the palette operation.

Each data register is 18 bits wide: 6 bits each for red, green, and blue. The data registers are accessible through the system interface.

Video DAC to System Interface

The Palette Address register holds an 8-bit value that is used to address a location within the video DAC. The Palette Address register responds to two addresses; the address depends on the type of palette access, Read or Write. Once the address is loaded, successive accesses to the data register automatically increment the address register.

For palette Write operations, the address for the Palette Address register is hex 03C8. A write cycle consists of writing three successive bytes to the Data register at address hex 03C9. The six least-significant bits of each byte are concatenated to form the 18-bit palette data. The order is red value first, then green, then blue.

For palette Read operations, the address for the Palette Address register is hex 03C7 (in the read mode, the Palette Address register is write only). A read cycle consists of reading three successive bytes from the Data register at address hex 03C9. The six least-significant bits of each byte contain the corresponding color value. The order is red value first, then green, then blue.

If the Palette Address register is written to during a Read or Write cycle, a new cycle is initialized and the unfinished cycle is terminated. The effects of writing to the Data register during a Read cycle or reading from the Data register during a Write cycle are undefined and can change the palette contents.

The DAC State register is a read-only register at address hex 03C7. Bits 1 and 0 return the last active operation to the DAC. If the last operation was a Read operation, both bits are set to 1. If the last operation was a Write, both bits are set to 0.

Reading the Read Palette Address register at hex 03C8 or the DAC State register at hex 03C7 does not interfere with read or write cycles.

Programming Considerations

- As explained in "Video DAC to System Interface" on page 87, the effects of writing to the Data register during a read cycle or reading from the Data register during a write cycle are undefined and can change the palette contents. Therefore, the following sequence must be followed to ensure the integrity of the color palette during accesses to it:

  1. Disable interrupts.
  2. Write the address to PEL Address register.
  3. Write or read three bytes of data.
  4. Go to Step 2, repeat for the desired number of locations.
  5. Enable interrupts.

  Note: All accesses to the DAC registers are byte-wide I/O operations.

- To prevent "snow" on the screen, an application reading data from or writing data to the DAC registers should ensure that the blank input to the DAC is asserted. This can be accomplished either by restricting data transfers to retrace intervals (use Input Status register 1 to determine when retrace is occurring) or by using the Screen Off bit located in the Clocking Mode register in the sequencer.

  Note: BIOS provides read and write interfaces to the Video DAC.

- Do not write to the PEL Mask register (hex 03C6). Palette information can be changed as a result. This register is correctly initialized to hex FF during a mode set.

VGA Video Extensions

The video extensions provide a means of transferring video information between two video subsystems: the base video subsystem and an auxiliary video adapter.

The video extensions consist of:

- The auxiliary video extension
- The base video extension
- The auxiliary video signals.

The base video is provided by the video subsystem integrated onto the system board, or, when not provided on the system board, it can be provided by a suitable video adapter. Such an adapter can provide a Micro Channel connector with the base video extension. Video adapters supporting the base video extension must provide the VGA function as the default. (See the description of the Micro Channel Adapter Design for detailed connector dimensions.)

The video buffers for the base video can be turned off, and video output from the auxiliary video can be sent through the base video DAC to the display. The video extension can be driven in only one direction at a time.

Note: The video extension supports only the VGA function (see Figure 98 on page 95).

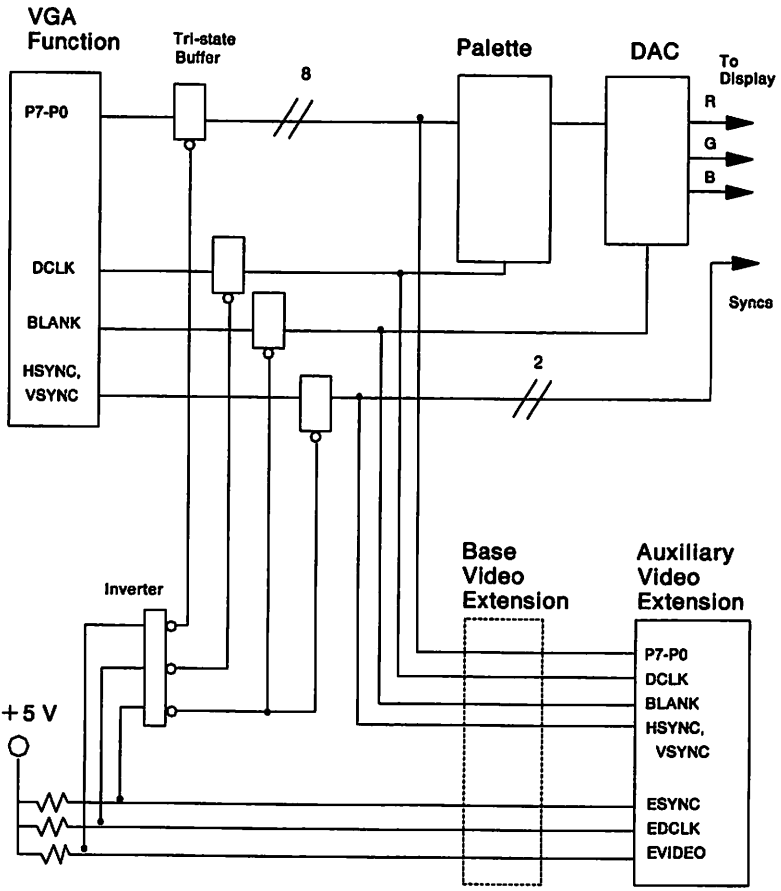


Figure 95. Auxiliary Video Connector Interface

Auxiliary Video Extension

This extension provides a video adapter with access to resources of the base video subsystem and allows it to extend or supplement the functions provided.

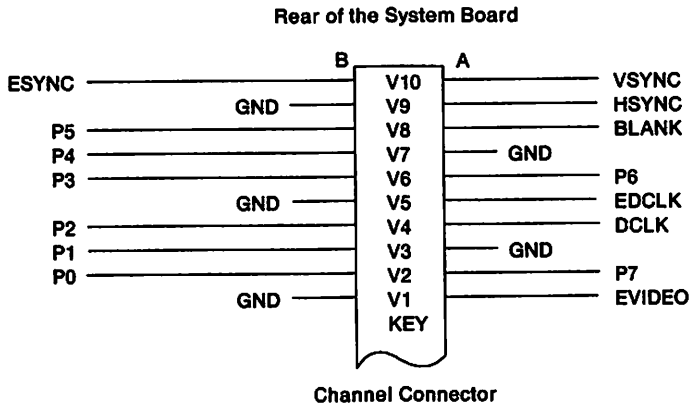


Figure 96. Auxiliary Video Extension

Base Video Extension

This extension is for adapters that provide the base video subsystem. Only systems without a base video subsystem on the system board have a connector with this extension.

Video adapters supporting the base video extension must provide the VGA function as the default.

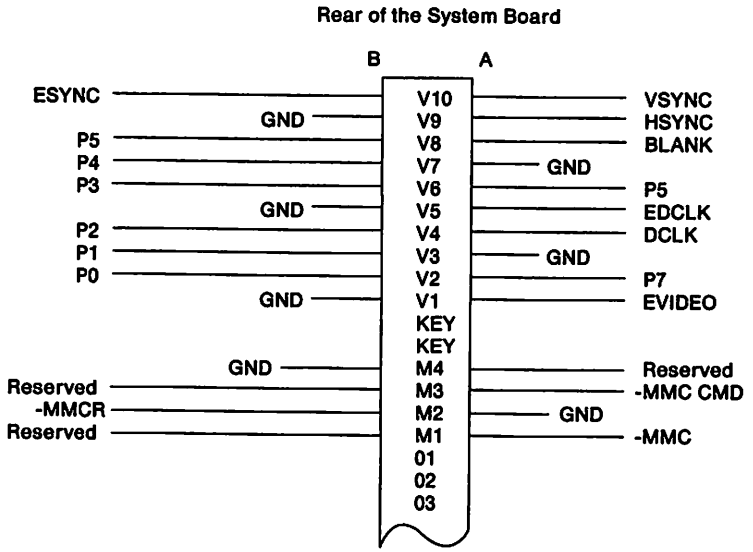


Figure 97. Base Video Extension

Auxiliary Signal Descriptions

The following are signal descriptions for the Auxiliary Video extension of the channel connector.

VSYNC: Vertical Synchronization: This signal is the vertical synchronization signal to the display. Also see the ESYNC description.

HSYNC: Horizontal Synchronization: This signal is the horizontal synchronization signal to the display. Also see the ESYNC description.

BLANK: Blanking Signal: This signal is connected to the BLANK input of the video DAC. When active (0 Vdc), this signal tells the DAC to drive its analog color outputs to 0 Vdc. Also see the ESYNC description.

P7 - P0: Palette Bits: These eight signals contain video information and comprise the PEL address inputs to the video DAC. See also the EVIDEO description.

DCLK: Dot Clock: This signal is the PEL clock used by the DAC to latch the digital video signals, P7 through P0. The signals are latched into the DAC on the rising edge of DCLK.

This signal is driven through the EXTCLK input to the VGA when DCLK is driven by the adapter. If an adapter is providing the clock, it must also provide the video data to the DAC. Also see the EDCLK description.

ESYNC: External Synchronization: This signal is the output-enable signal for the buffer that drives BLANK, VSYNC, and HSYNC. ESYNC is tied to +5 Vdc through a pull-up resistor. When ESYNC is high, the VGA drives BLANK, VSYNC, andHSYNC. When ESYNC is pulled low, the adapter drives BLANK, VSYNC, andHSYNC.

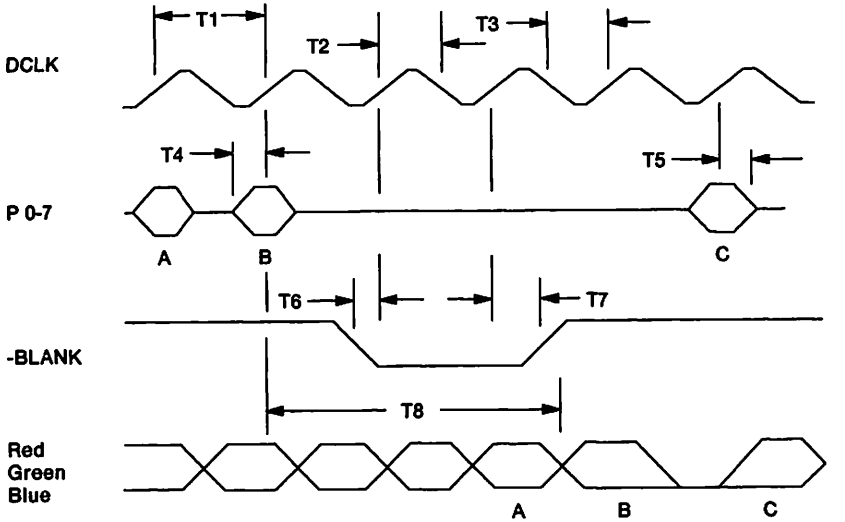
EVIDEO: External Video: This signal is the output-enable signal for the buffer that drives P7 through P0. EVIDEO is tied to +5 Vdc through a pull-up resistor. When EVIDEO is high, the VGA drives P7 through P0. When it is pulled low, the adapter drives P7 through P0.

EDCLK: External Dot Clock: This signal is the output-enable signal for the buffer that drives DCLK. EDCLK is tied to +5 Vdc through a pull-up resistor.

When EDCLK is high, the VGA is the source of DCLK to the DAC and the adapter. The Miscellaneous Output register should not select clock source 2 (010 binary) when EDCLK is high.

When EDCLK is pulled low, the adapter drives DCLK. If the adapter is driving the clock, it must also provide the video data to the DAC, and the Miscellaneous Output register must select clock source 2 (010 binary).

Video Extension Signal Timing



Symbol	Description	Min.(ns)	Max.(ns)
T1	PEL Clock Period	28	10,000
T2	Clock Pulse Width High	7	10,000
T3	Clock Pulse Width Low	9	10,000
T4	PEL Set-Up Time	4	-
T5	PEL Hold Time	4	-
T6	Blank Set-Up Time	4	-
T7	Blank Hold Time	4	-
T8	Analog Output Delay	3(T1) + 5	3(T1) + 30

Figure 98. Auxiliary Video Extension Signal Timing (DAC Signals)

XGA Function

The XGA provides:

- VGA function
- 132-column text mode
- Extended graphics function.

VGA Function

When in VGA mode, the XGA subsystem is VGA register compatible. At power on, the XGA is set to provide the VGA function as the default. The VGA function is described in "VGA Function" on page 2.

132-Column Text Mode

The XGA function can be operated in the 132-column text mode. In this mode, text is displayed in 132 vertical columns using 200, 350, or 400 scan lines. Each character is eight PELs wide. The 132-column text mode cannot be used with the auxiliary video extension.

Extended Graphics Function

The extended graphics function provides resolutions up to 1024 rows by 768 lines with 256 colors. It cannot be used with the auxiliary video extension.

The extended graphics function provides the following software and hardware support.

8514/A Adapter Compatibility
Compatibility is provided through the XGA Adapter Interface device driver, supplied with the subsystem as programming support for applications operating in the DOS environment. (Refer to the *XGA Adapter Interface Technical Reference.*)

Direct Color Mode
In this mode, 16-bit PELs in video memory directly specify the color output of the DAC, rather than going through the palette. The direct-color mode provides 64,000 colors with a maximum resolution of 640 x 480.

Packed Pixel Format

In the packed-format, reads and writes to the video memory can access all of the data that defines a PEL (pixel) in a single operation, rather than a single bit from one map.

Sprite

The sprite is a programmable image 64-by-64 PELs that is controlled by the hardware. When enabled, the sprite overlays the displayed image and can be positioned anywhere on the screen without affecting the contents of video memory.

Display Identification

The display has four monitor ID signals that identify the type of display attached. Applications can read the ID bits to determine the maximum resolution and color characteristics of the display.

Coprocessor

A coprocessor provides hardware drawing-assist functions throughout real or virtual memory. These functions can be used with the XGA Adapter Interface.

Major Components

The subsystem components providing XGA function are:

- System-bus interface
- Memory and CRT controller
- Coprocessor
- Video memory
- Attribute controller
- Sprite controller
- A/N font and sprite buffer
- Serializer
- Palette
- Digital-to-analog convertor (DAC).

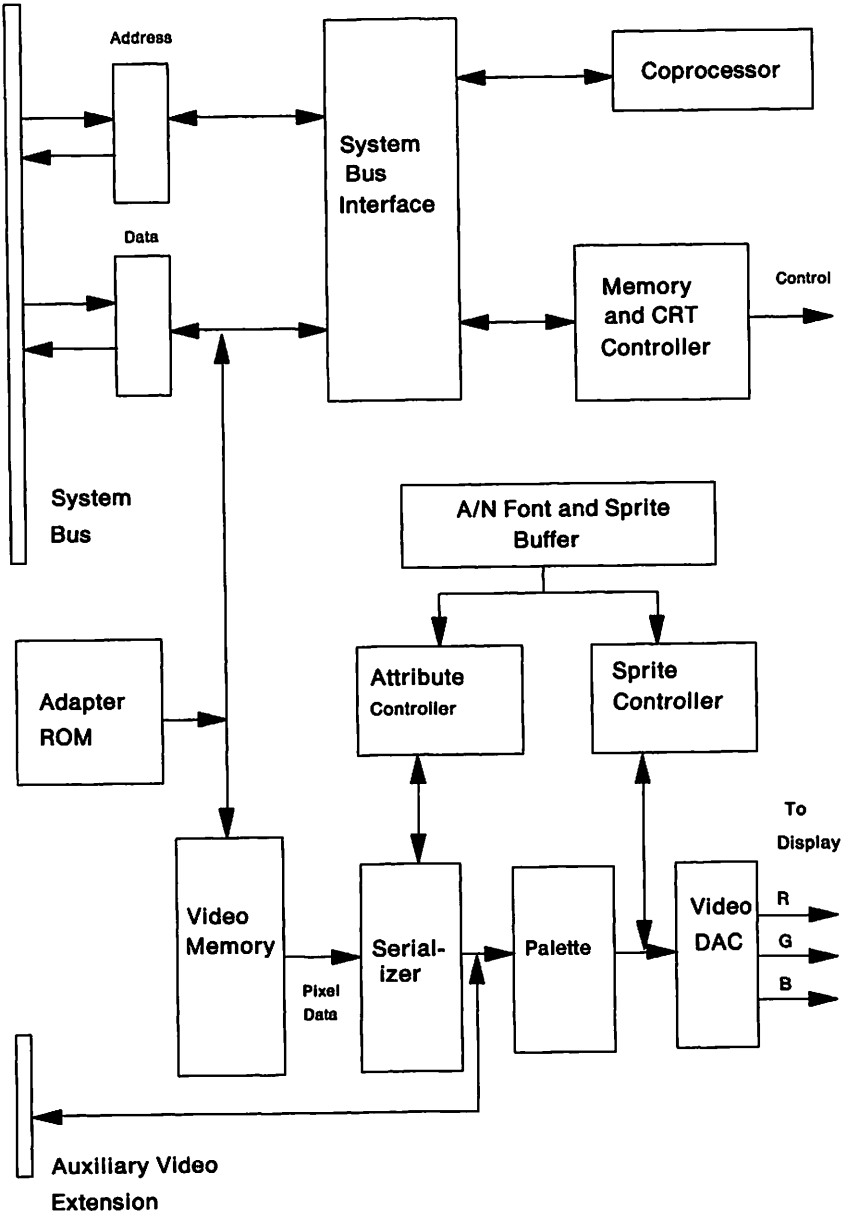


Figure 99. XGA Function Block Diagram

System Bus Interface

The system bus interface provides control of the interface between the video subsystem and the system channel. It decodes the I/O addresses to the I/O registers and the memory addresses to the coprocessor registers and video memory.

It also provides the bus-master capability and identifies the width of the system data bus, 16- or 32-bits.

Memory and CRT Controller

This controller controls accesses to video memory by the system microprocessor, displays contents of video memory on the display, and provides the additional support for the 132-column text mode.

Coprocessor

The coprocessor provides hardware drawing-assist functions used in the extended graphics mode. These functions can be performed on graphics data in both video and system memory. The coprocessor updates this memory independent of the system microprocessor.

The coprocessor is controlled through instructions written to a set of registers. These registers are mapped into the top 1KB of the 8KB ROM space assigned to the video subsystem. After the instructions are written, the coprocessor performs the drawing function without further support from the system.

The hardware-assist functions include the following:

Pixel-Block or Bit-Block Transfers
This function transfers an entire bit map, or part of a bit map, from one location to another. Transfers can be:

- Within video memory
- Within system memory
- Between video and system memory.

Line Drawing
This function draws lines, with a programmable style and thickness, into a bit map in video memory or system memory.

Area Filling
This function fills an outlined area with a programmable pattern that is loaded into a bit map. The bit map can be in video or system memory.

Logical and Arithmetic Mixing
These functions provide logical and arithmetic operations that can be used with data in video and system memory.

Map Masking
This function provides update control of each PEL over all drawing functions.

Scissoring
This function provides a rectangular-mask function, which can be used instead of the mask map.

X,Y Axis Addressing
This function allows a PEL to be specified by its X and Y coordinates within a pixel map, instead of its linear address in memory.

Video Memory

The video subsystem uses dual-ported video memory to store on-screen data. Dual-ported memory allows data to be updated simultaneously with screen refresh.

Attribute Controller

The attribute controller works together with the memory and CRT controller to control the color selection and character generation in text modes for the VGA function and in the 132-column text mode.

Sprite Controller

The sprite controller is used to control the position and image of the sprite, which can be used as a cursor. The sprite is not available in the VGA-function modes or the 132-column text mode.

A/N Font and Sprite Buffer

The buffer for the sprite controller stores the sprite image while in graphics modes and holds the character fonts in the text modes.

The Serializer, Palette and DAC.

The serializer takes data from the serial port of video memory in 16- or 32-bit widths (depending on the amount of video memory installed) and converts this data to a serial stream of PEL data. The PEL data is used to address a palette location, which contains the color value. The color value is then passed to the DAC, which converts the digital information into analog red, green, and blue signals to the display.

Video Memory Requirements

While in extended graphics mode, the amount of video memory installed determines the screen resolution available and the number of colors supported.

Video Memory Installed	Maximum Resolution	Maximum Colors
512KB	640 x 480	256
	1024 x 768	16
1MB	640 x 480	65,536
	1024 x 768	256

Figure 100. Video Memory Requirements

Compatibility

8514/A

The extended graphics function is not compatible with the 8514/A adapter at the register level. However, it does provide function level compatibility with the 8514/A Adapter Interface. The XGA Adapter Interface is used by applications operating in the DOS environment and is supplied with the video subsystem.

Applications written to the 8514/A Adapter Interface operate with the XGA Adapter Interface with the following considerations:

Protected Mode
The XGA Adapter Interface does not operate in protected mode.

640x480 4-by-4 Mode with 512KB Video Memory
Although this is not an extended graphics mode, applications that use this mode and abide by the rules described in the 8514/A technical reference will operate correctly.

Dual Display Buffer
In the 8514/A environment, applications could operate in both the advanced function mode and the VGA function mode with a single display. They could operate with one display because the two functions were provided by two video subsystems; each instance of video had its own video memory.

In the XGA environment, these application will not operate correctly if only one video subsystem is present because the same video memory is shared between the two functions. These applications will operate correctly if a second video subsystem is present.

Non-Display Memory
The mapping of non-display (or off-screen) memory differs for the extended graphics function and 8514/A adapter. Applications that use areas of non-display memory for their own storage will not operate correctly in the extended graphics modes.

EMS Drivers
If an application written for the 8514/A adapter locates resources (such as bit maps and font definitions) in EMS memory and passes the addresses of these resources to the Adapter Interface, the application must use an EMS driver that provides the physical-address services at BIOS interrupt hex 4B. See the *IBM Personal System/2 and Personal*

Computer BIOS Interface Technical Reference for information on physical address services.

Time Dependent Applications
Certain extended graphic and 8514/A functions run at different speeds. Applications that rely on specific performance will operate differently.

Adapter Interface Enhancements
The XGA Adapter Interface provides additional functions to that of the 8514/A Adapter Interface. Applications for the 8514/A adapter that use reserved codes in the parameter blocks could trigger some of the additional functions of the XGA Adapter Interface.

Naming Differences
The module name and path of the XGA Adapter Interface (\XGAPCDOS\XGAAIDOS.SYS) are different from that of the 8514/A (\HDIPCODS\HDILOAD.EXE).

Adapter Interface Code Size
The code size of the XGA Adapter Interface is larger than that for the 8514/A adapter, thereby reducing the amount of system memory available to applications.

Adapter Interface Code Type
The XGA Adapter Interface is implemented as a device driver (.SYS); the 8514/A Adapter Interface is a terminate-and-stay-resident (TSR) executable program (.EXE).

Drivers using ROM Address Space

The coprocessor registers are mapped into memory address space used for adapter ROM; this memory is between hex 0C0000 and 0DFFFF. The registers are in the top 1KB of the 8KB block assigned to the video subsystem.

When the extended graphics function is on an adapter, the first 7KB of this block access the ROM code on the adapter. When the function is on the system board, the 8KB block is still assigned to the video subsystem, but the first 7KB is not used.

Note: Applications or device drivers that scan these ROM addresses looking for unused space can incorrectly assume that the entire 8KB is available. Conflicts are resolved using the Reference Diskette.

Display Connector

The synchronization and monitor ID signals are TTL levels. The video signals are analog signals ranging from 0 to 0.7 volts.

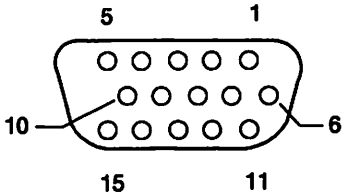


Figure 101. Display Connector

Pin	Signal Description	Display Pins	
		Monochrome	Color
1	Red	N/C	Red
2	Green	Mono	Green
3	Blue	N/C	Blue
4	Monitor ID 2		
5	Ground	Self Test	Self Test
6	Red Ground	N/C	Red Ground
7	Green Ground	Mono Ground	Green Ground
8	Blue Ground	N/C	Blue Ground
9	Plug	No Pin	No Pin
10	Ground	Ground	Ground
11	Monitor ID 0		
12	Monitor ID 1		
13	Hsync	Hsync	Hsync
14	Vsync	Vsync	Vsync
15	Monitor ID 3		

Figure 102. Display Connector Signals

© Copyright IBM Corp. 1989, 1990

Signal Timing

In the VGA modes, BIOS sets the video subsystem according to the video mode selected. All VGA modes use a 70-Hz, vertical retrace except for modes 11 and 12. These two modes use 60 Hz.

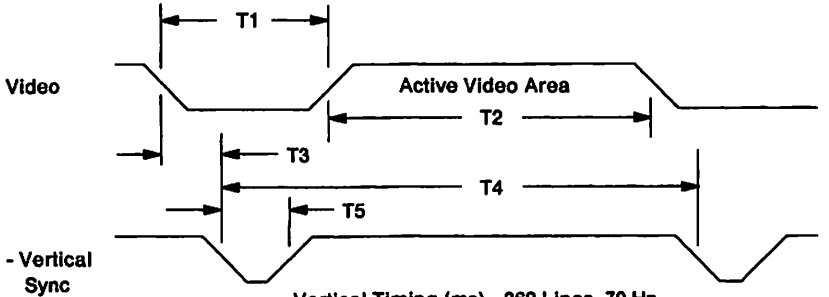
For 1024 x 768 modes, the vertical retrace rate is 43 Hz (this mode is not set by BIOS).

The video subsystem generates the signal timings required by the displays according to the mode selected. The following timing diagrams represent only the vertical frequencies.

Note: The vertical size of the display is encoded using the polarity of the synchronization signals, as shown in the following figure.

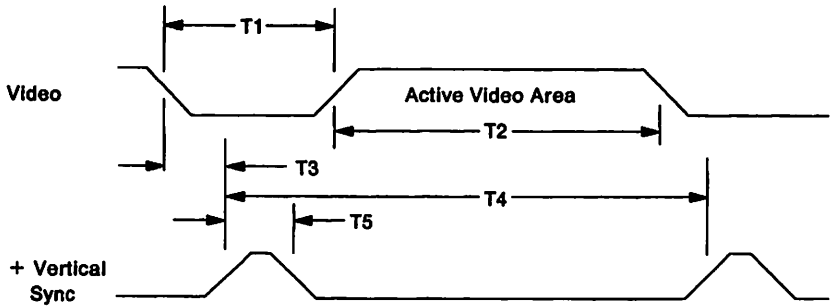
VSYNC Polarity	HSYNC Polarity	Vertical Size
+	+	768 lines (1024 x 768 displays only)
+	−	400 lines
−	+	350 lines
−	−	480 lines

Figure 103. Vertical Size



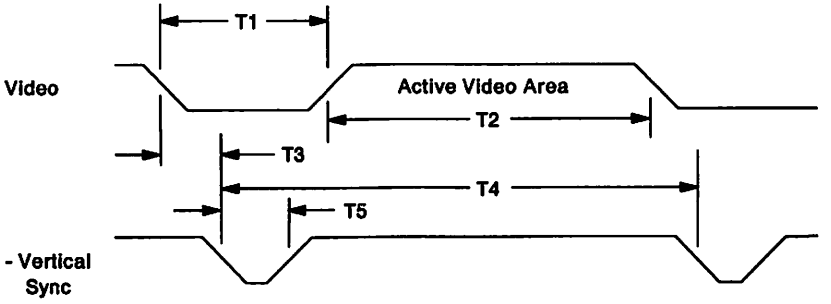
Symbol	Signal Time
T1	2.765 ms
T2	11.504 ms
T3	0.985 ms
T4	14.268 ms
T5	0.064 ms

Figure 104. Vertical Timing, 350 Lines



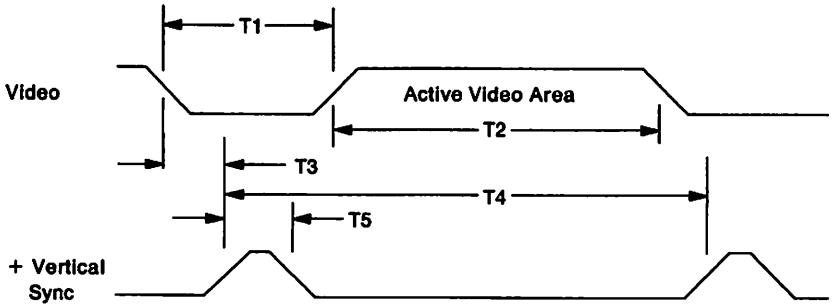
Symbol	Signal Time
T1	1.112 ms
T2	13.156 ms
T3	0.159 ms
T4	14.268 ms
T5	0.064 ms

Figure 105. Vertical Timing, 400 Lines



Symbol	Signal Time
T1	0.922 ms
T2	15.762 ms
T3	0.064 ms
T4	16.683 ms
T5	0.064 ms

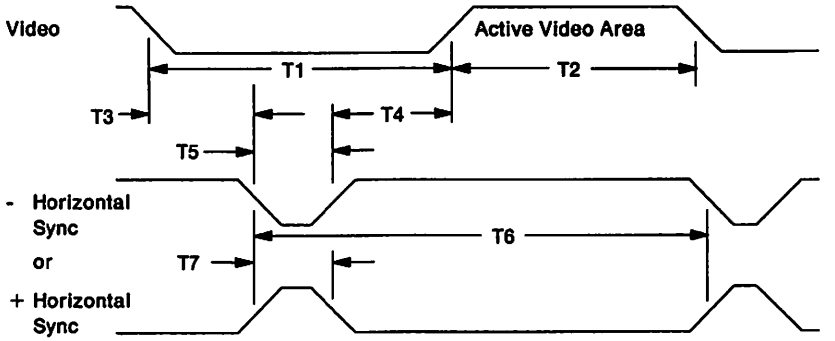
Figure 106. Vertical Timing, 480 Lines



Symbol	Signal Times	
T1	0.676 / 0.704 ms	(Odd/Even Lines)
T2	21.620 ms	
T3	0.010 / 0.140 ms (Odd/Even Lines)	
T4	23.000 ms	
T5	0.113 ms	

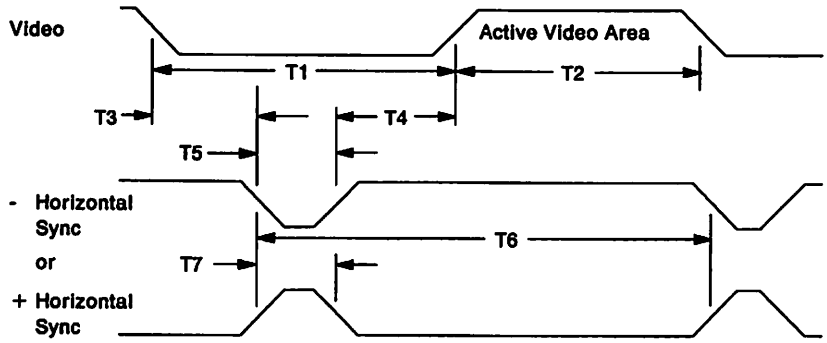
Because this is an interlaced mode, the timing for the odd and even lines is different.

Figure 107. Vertical Timing, 768 Lines. Type 1 video does not support these timings.



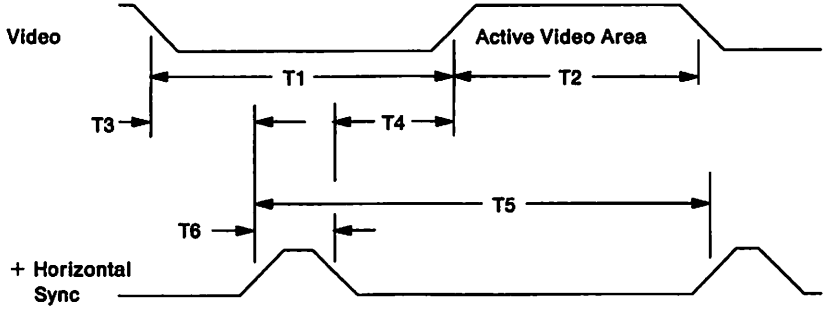
Symbol	Signal Time
T1	5.720 µs
T2	26.058 µs
T3	0.318 µs
T4	1.589 µs
T5	3.813 µs
T6	31.778 µs
T7	3.813 µs

Figure 108. Horizontal Timing, 80 Column with Border



Symbol	Signal Time
T1	6.356 µs
T2	25.422 µs
T3	0.636 µs
T4	1.907 µs
T5	3.813 µs
T6	31.778 µs
T7	3.813 µs

Figure 109. Horizontal Timing, 40/80 Column, without Border



Symbol	Signal Times
T1	5.35 μs
T2	22.80 μs
T3	0.18 μs
T4	1.25 μs
T5	28.15 μs
T6	3.92 μs

Figure 110. Horizontal Timing, 1024 PELs. Type 1 video does not support these timings.

Index

A

address registers
  attribute controller 74
  palette 87
  sequencer 42
address select 72
all points addressable (APA)
 modes 14
  mds.graphics 14
alphanumeric memory mapping 23
alphanumeric modes 12
AND data select 70
asynchronous reset 43
attribute controller 7
  blink enable 76
  block diagram 7
  registers 74
attribute controller registers 74
  address 74
  color plane enable 77
  color select 79
  horiz PEL panning 78
  internal palette 0-F 75
  mode control 76
  overscan color 77
  reserved bits 38
attribute definition 13
auxiliary video extension 89, 91
  connector timing 95
  signals 93

B

base video extension 92
BIOS 4
BIOS video modes 9
bit mask register 73
blanking signal 93
blink enable 76
block diagram
  address mapping 64
  attribute controller 7
block diagram *(continued)*
  graphics controller 6
  memory read 37
  memory write 36
  subsystem 3
border color select 77
buffer address 58
buffer address select 72

C

chain 4 bit 48
character generator 46, 77
character generator, RAM
 loadable 84
character map select register 46
clocking mode register 43
color compare operations 37
color compare register 69
color don't care register 73
color graphics modes 640 x 480, 2
 color 19
color modes 9
color plane enable register 77
color select register 79
color/graphics modes 14
compare, line 66
compatibility 83
components, subsystem 4
connector timing 95
connector, auxiliary video
 extension 91
connector, base video
 extension 92
connector, display 105
considerations, programming 80
control cursor 57
control cursor skew 57
controller
  attribute 7
  CRT 4
  graphics 5

© Copyright IBM Corp. 1989, 1990

count by two 63
count by 4 61
creating a split screen 85
CRT controller 4
  address mapping 64
  horiz skew 51
  registers 49
CRT controller compatibility 65
CRT controller registers 49
  address 50
  cursor location 58
  cursor start and end 57
  end horiz blanking 51
  end horiz retrace 53
  end vert blank 62
  horizontal end 51
  horizontal total 50
  line compare 66
  max scan line 56
  mode control 63
  offset 61
  overflow 54
  preset row scan 55
  reserved bits 38
  start address 58
  start horiz blanking 51
  start horiz retrace 52
  start vert blank 62
  start vert total 54
  underline location 61
  vert retrace end 59
  vert retrace start 59
  vertical end 60
CRT mode control register 63
cursor location register 58
cursor off bit 57
cursor skew 57
cursor start/end register 57

D

DAC palette 87
  operation 87
  programming considerations 88
  register 87
  state register 88
DAC state register 88
data rotate register 69
DCLK signal 93
description 2
device operation, DAC 87
digital-to-analog converter (DAC) 87
display connector 105
display connector timing 106
display support 11
display synchronization signals 106
display, vertical gain 39
display, vertical size 106
don't care, color 73
double scanning 56
doubleword mode 61

E

EDCLK signal 94
enable bit, video subsystem 11
enable blink 76
enable line graphics 76
enable set/reset register 68
end horizontal blanking register 51
end horizontal retrace register 53
end vertical blanking register 62
ESYNC signal 93
EVIDEO signal 94

F

feature control 41
font, RAM loadable 84
format, video
  mode F 18
  mode 11 19
  mode 13 20
  mode 4,5 15
  mode 6 17
  16 color modes 19
full-screen updates 43

G

general description 2
general registers 39
  input status 40, 41
  miscellaneous output 39
  reserved bits 38
  video enable 42
graphics controller 5
  block diagram 6
  register 67
graphics controller registers 67
  bit mask 73
  color compare 69
  color don't care 73
  data rotate 69
  enable set/reset 68
  miscellaneous 72
  mode 70
  read map select 70
  reserved bits 38
  set/reset 68
graphics mode register 70
graphics modes 14
graphics modes 640 x 200, 2-color 17
graphics modes 640 x 350 18
graphics modes, 16-color 19
graphics modes, 320 x 200, four-color 15
graphics mode, 256-color 19
guidelines, programming 80

H

horizontal display enable end register 51
horizontal PEL panning register 78
horizontal retrace select 63
horizontal skew 51
horizontal total register 50
HSYNC signal 93

I

image shift 78
input status register 40
interface, system to DAC 87
internal palette select 74
interrupt enable 60

L

line compare register 66
line graphics 76
line graphics character 76
logical operator select 70
logic, support 4

M

major components 4
map mask register 45
map select register 46
map select, read 70
mapping, memory 64
mapping, split screen 86
maps, memory 22
maximum scan line register 56
memory address select 72
memory maps 22, 64
memory mode register 48
memory read/writes 36
methods for programming 83
miscellaneous output register 39
miscellaneous register 72
mode control register (CRT) 63
mode control register, attribute 76
mode hex F, 640 x 350 graphics 18
mode hex 11, 640 x 480 two color graphics 19
mode hex 13, 256-color 19
mode register, graphics 70
mode 16-color graphics 19
mode 6, 640 x 200 two-color 17
modes
  alphanumeric 12
  BIOS 9
  count by two 63
  count by 4 61
  doubleword 61
  memory maps 21
  odd/even 71
  read and write 71

modes *(continued)*
word/byte 63
512 char font 84
modes 4 and 5, 320 x 200
four-color 15
modes, memory 21
monochrome modes 9

O

odd/even address bit 48
odd/even mode 71
offset register 61
OR data select 70
organization, video memory 21, 22
overflow register, CRT
controller 54
overscan color register 77

P

palette address select 74
palette operation 87
palette registers 75
palette, DAC 87
PEL panning 78
polarity, sync 39
POS information 11
preset row scan register 55
programming considerations 80
programming registers 83
protect registers 60
P4,P5 select 76
P7 - P0 signal 93

R

RAM loadable character
generator 84
read map select register 70
read modes 71
read/writes, memory 36
refresh select 60
regen buffer address 58
registers 38
attribute controller 74
CRT controller 49
registers *(continued)*
DAC 87
feature control 41
general 39
input status 40
miscellaneous output 39
programming 83
reserved bits 38
video enable 42
write protect 60
reset bit 63
reset register 43
retrace polarity 39
ROM BIOS 4
rotate data 69
row scan counter 63

S

scanning, double 56
screen off bit 43
select address 72
select horizontal retrace 63
select operator 70
select P4,P5 76
select row scan 63
sequencer 5
registers 42
sequencer registers 42
address 42
clocking mode 43
map mask 45
map select 46
memory mode 48
reserved bits 38
reset 43
setting modes 9
setup mode 11
set/reset register 68
shift image 78
signal, auxiliary video 93
skew control, cursor 57
skew control, horizontal 51
source select 74
specifications, display 11
split screen, creating 85

start address high register 58
start address low register 58
start address register 58
start horizontal blanking register 51
start horizontal retrace pulse register 52
start vertical blanking register 62
support logic 4
support, display 11
sync polarity 39, 106
synchronization signals 106
synchronous reset 43
system interface, DAC to 87

T

timing, display 106
timing, video 95

U

underline location register 61
updates, full-screen 43

V

vertical display enable end register 60
vertical gain 39
vertical interrupt 59
vertical retrace end register 59
vertical retrace interrupt 60
vertical retrace start register 59
vertical size 106
vertical total register 54
VGA programming considerations 80
video connector 105
video DAC programming considerations 88
video enable bit 11
video graphics array 2, 4
video modes 9
video registers 83
video subsystem enable 42
VSYNC signal 93

W

word/byte mode 63
write modes 71
write operations 36

Numerics

16 background colors 76
512 character fonts 84

Keyboards (101- and 102-Key)

Description 1
Keyboard Layouts 1
- 101-Key Keyboard 2
- 102-Key Keyboard 3
- Belgian Keyboard 4
- Canadian French Keyboard 5
- Danish Keyboard 6
- Dutch Keyboard 7
- French Keyboard 8
- German Keyboard 9
- Italian Keyboard 10
- Latin American Keyboard 11
- Norwegian Keyboard 12
- Portuguese Keyboard 13
- Spanish Keyboard 14
- Swedish Keyboard 15
- Swiss Keyboard 16
- U.K. English Keyboard 17
- U.S. English Keyboard 18

Sequential Key-Code Scanning 19
- Buffer 19
- Keys 19

Power-On Routine 20
- Power-On Reset (POR) 20
- Basic Assurance Test 20

Commands from the System 21
Commands to the System 26
Scan Codes 27
- Set 1 Scan-Code Tables 27
- Set 2 Scan-Code Tables 30
- Set 3 Scan Code Tables 34

Clock and Data Signals 37
- Data Stream 37
- Data Output 38
- Data Input 39
- Encode and Usage 40

Extended Functions 43
Shift States 45
Special Handling 47
- System Reset 47
- Break 47
- Pause 47

© Copyright IBM Corp. 1990

Print Screen 47
System Request 48
Other Characteristics 48
Cables and Connectors 49
Specifications 50

Figures

1. Keyboard Commands from the System 21
2. Set All Keys Commands 23
3. Set Key Type Commands 23
4. Set/Reset Status Indicators 24
5. Typematic Rate 25
6. Keyboard Commands to the System 26
7. Keyboard Scan Codes, Set 1 28
8. Keyboard Scan Codes, Set 1 29
9. Keyboard Scan Codes, Set 1 30
10. Keyboard Scan Codes, Set 1 30
11. Keyboard Scan Codes, Set 1 30
12. Keyboard Scan Codes, Set 2 31
13. Keyboard Scan Codes, Set 2 32
14. Keyboard Scan Codes, Set 2 33
15. Keyboard Scan Codes, Set 2 33
16. Keyboard Scan Codes, Set 2 33
17. Keyboard Scan Codes, Set 3 34
18. Keyboard Data Stream Bit Definitions 38
19. Character Codes 41
20. Special Character Codes 43
21. Keyboard Extended Functions 44
22. Keyboard Connectors Signal and Voltage Assignments . . . 49

© Copyright IBM Corp. 1990

Notes:

Description

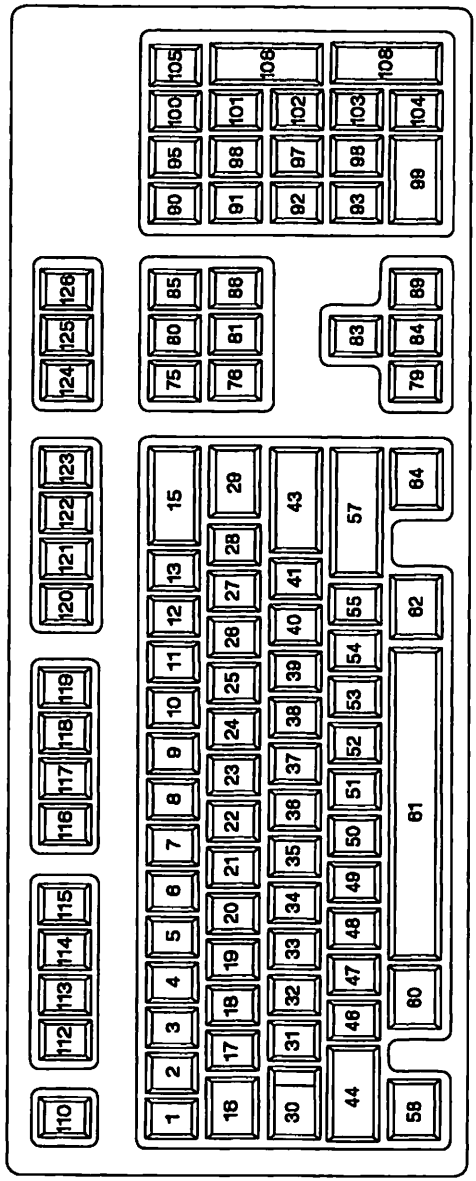
The keyboard has 102 keys (101 in the U.S.). At system power-on, the keyboard monitors the signals on the 'clock' and 'data' lines and establishes its line protocol. A bidirectional serial interface in the keyboard converts the 'clock' and 'data' signals and sends this information to and from the keyboard through the keyboard cable.

Keyboard Layouts

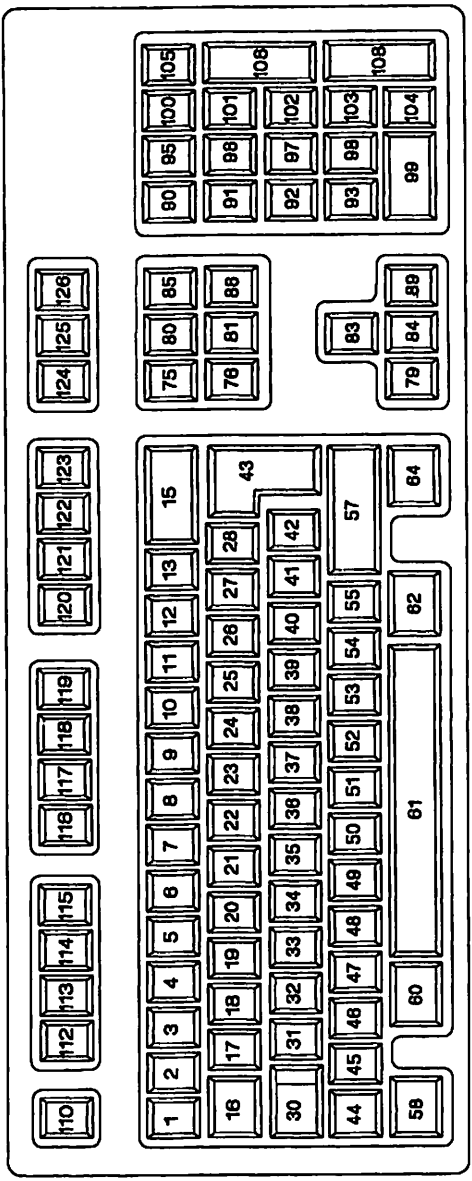
Keyboard layouts are in alphabetic order on the following pages. Nomenclature is on both the top and front face of the keybuttons.

- Belgian
- Canadian French
- Danish
- Dutch
- French
- German
- Italian
- Latin American
- Norwegian
- Portuguese
- Spanish
- Swedish
- Swiss
- U.K. English
- U.S. English.

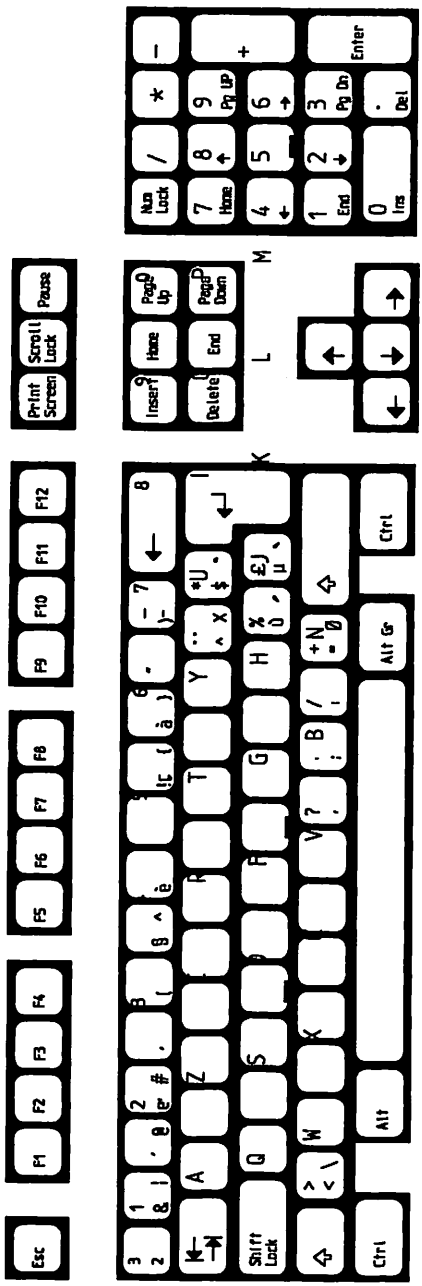
101-Key Keyboard



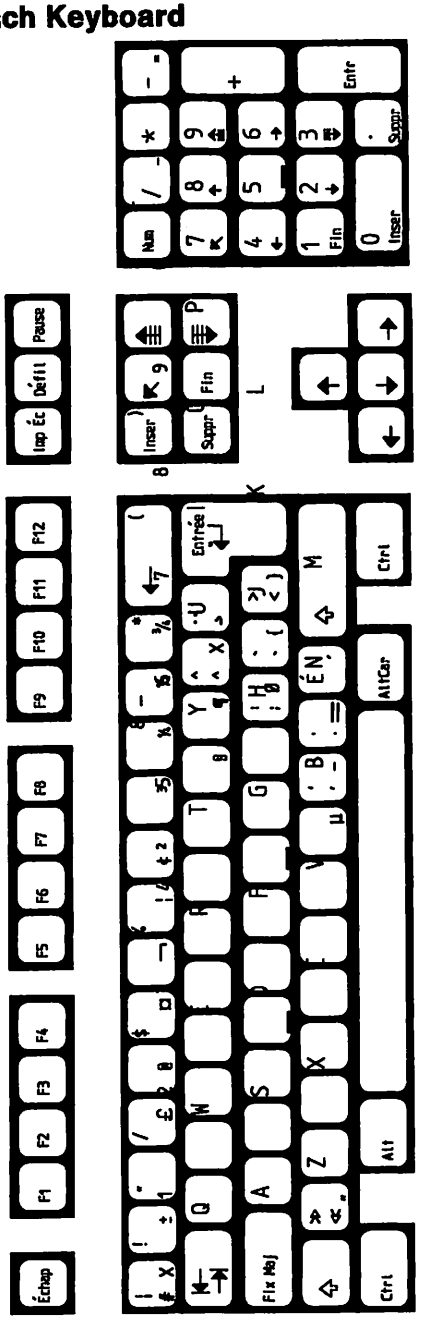
102-Key Keyboard



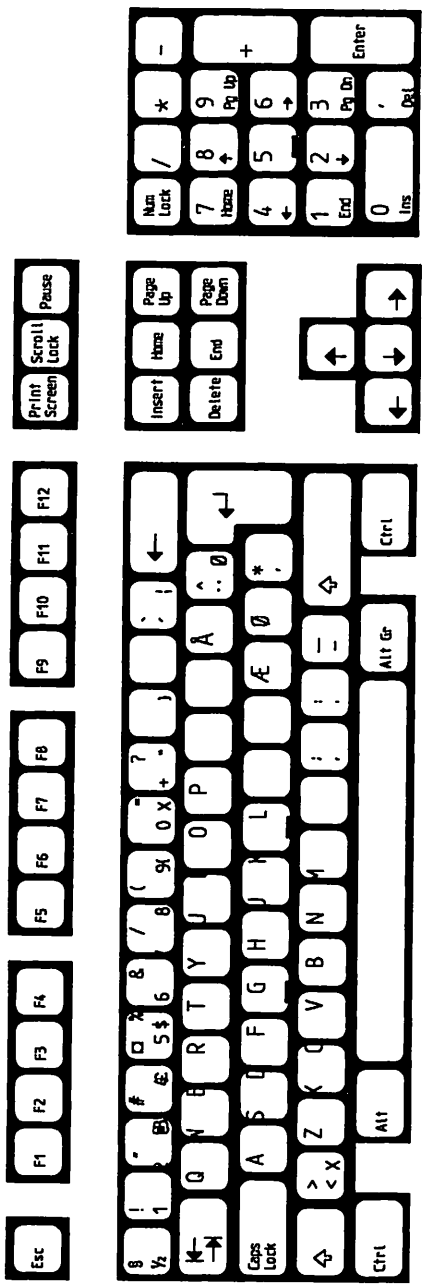
Belgian Keyboard



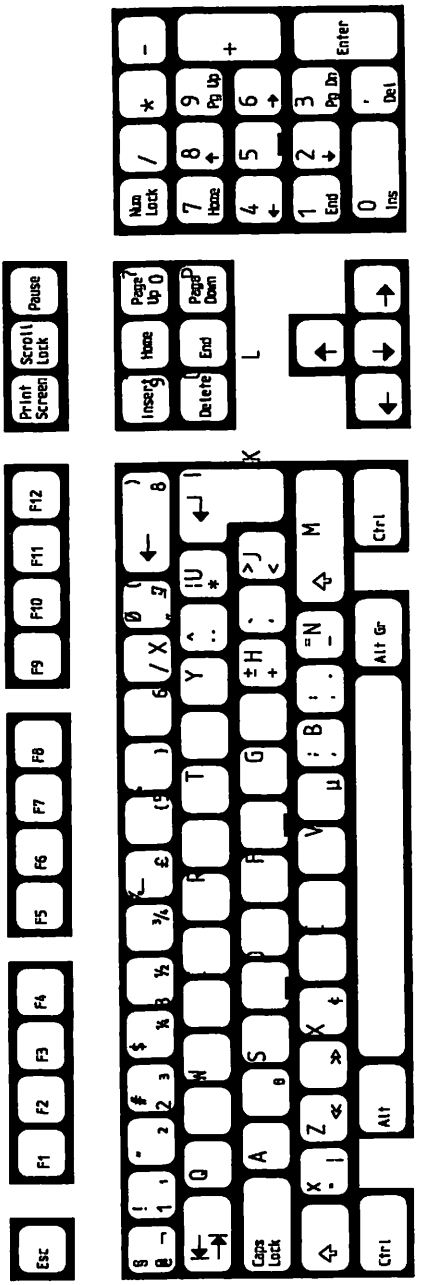
Canadian French Keyboard



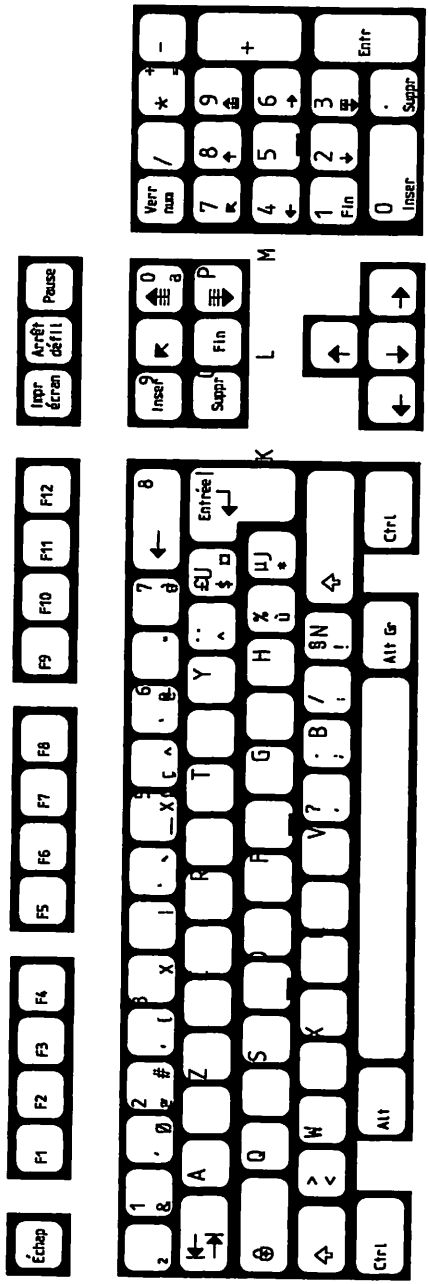
Danish Keyboard



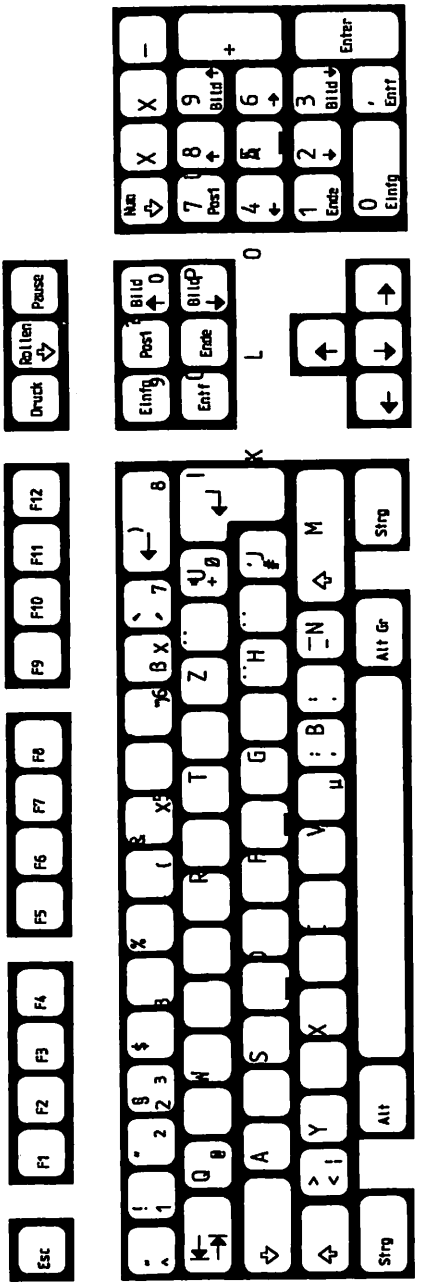
Dutch Keyboard



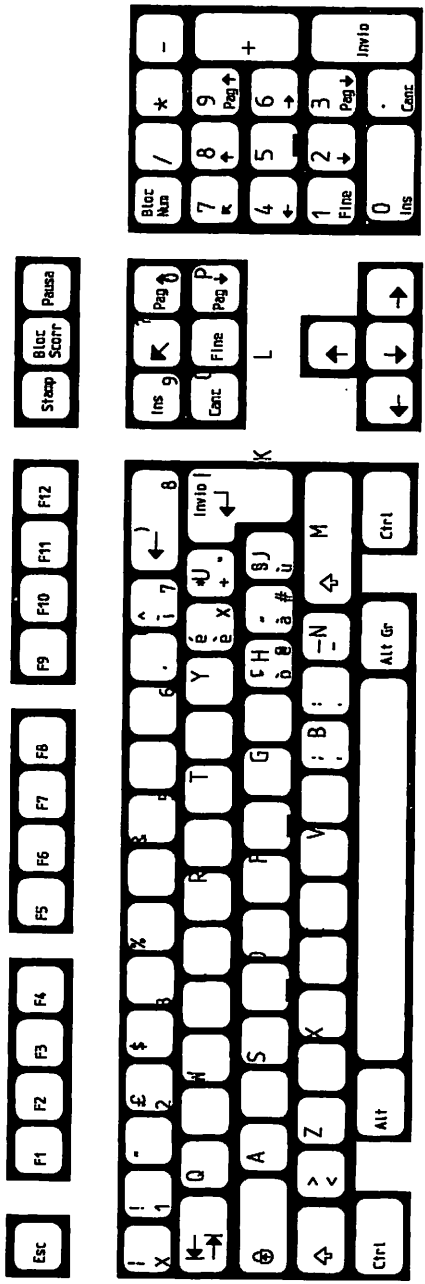
French Keyboard



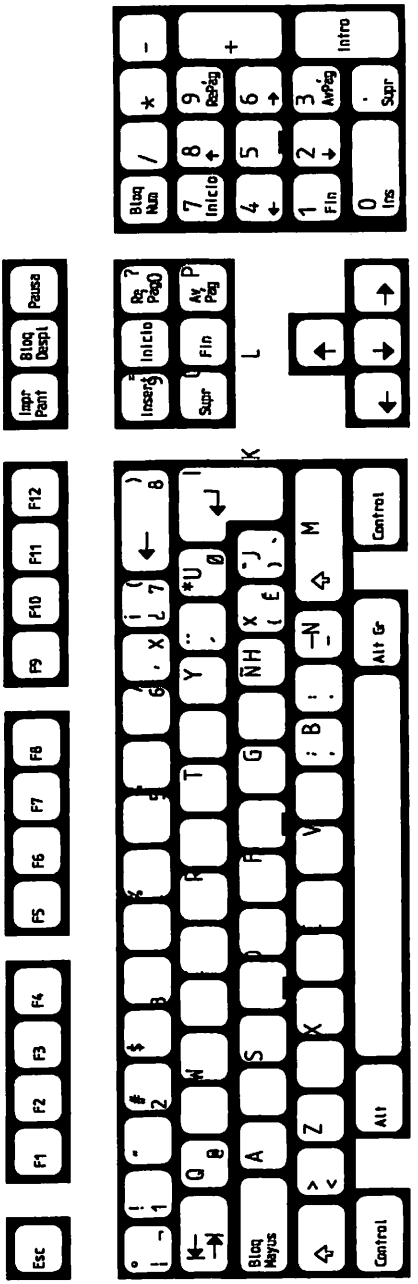
German Keyboard



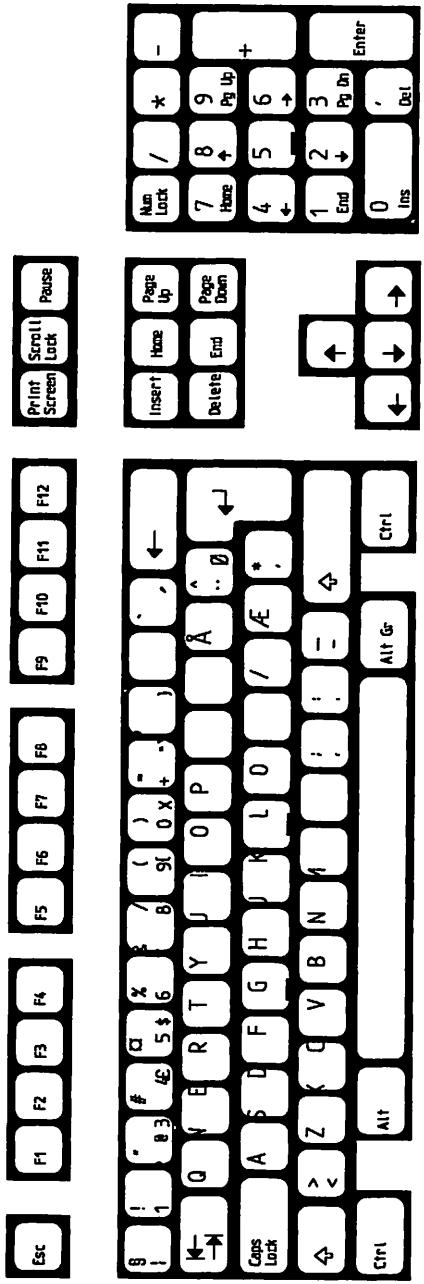
Italian Keyboard



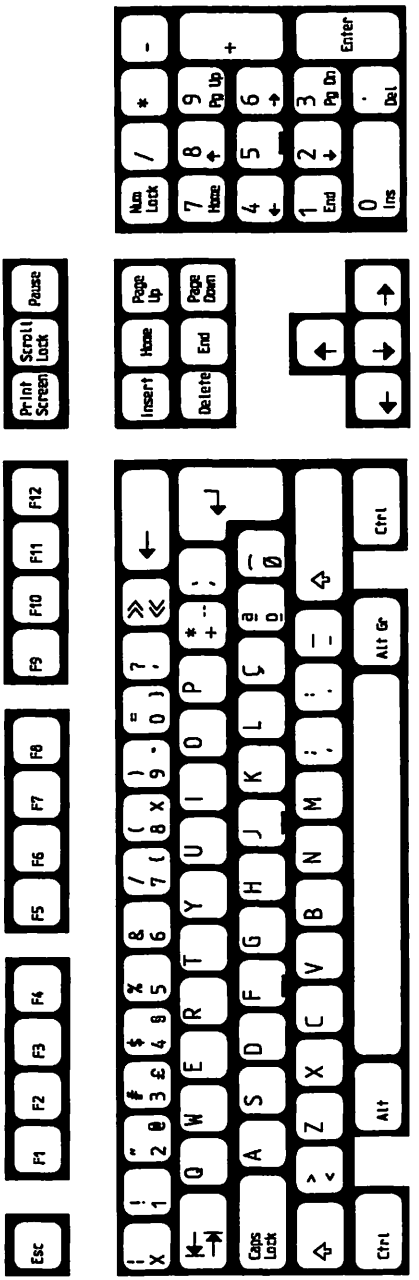
Latin American Keyboard



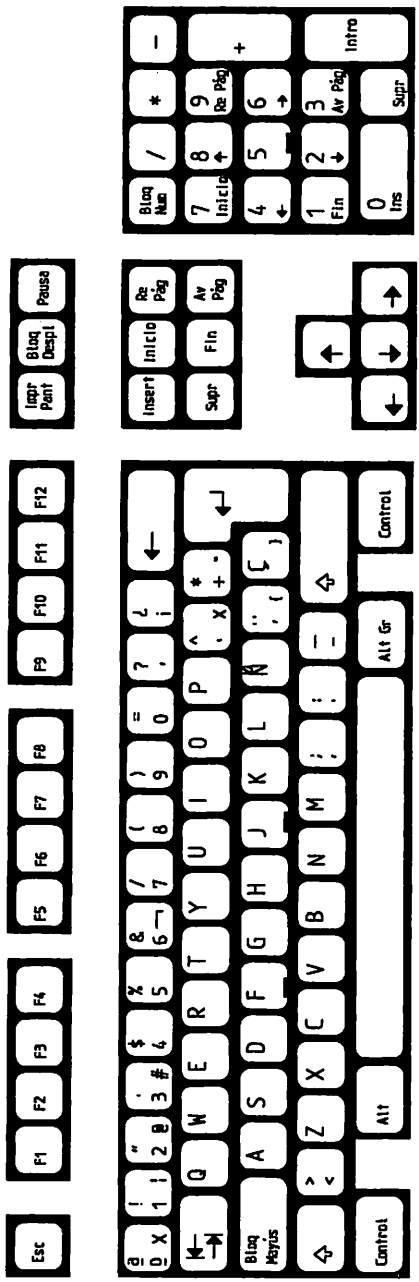
Norwegian Keyboard



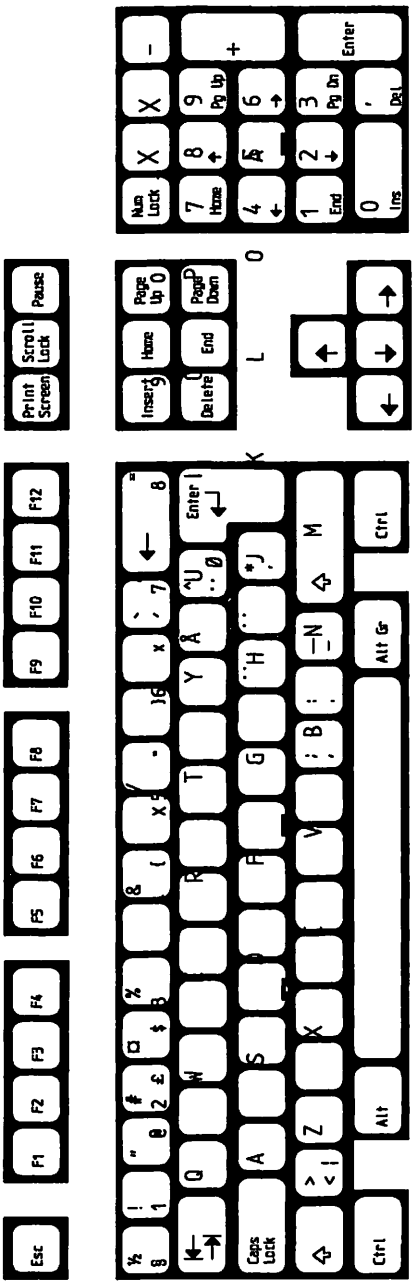
Portuguese Keyboard



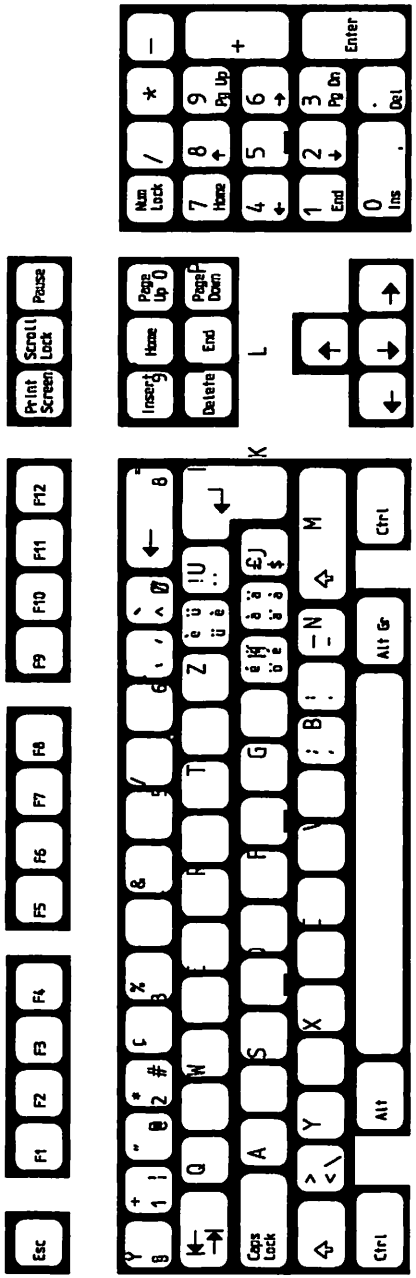
Spanish Keyboard



Swedish Keyboard



Swiss Keyboard



U.K. English Keyboard

U.S. English Keyboard

Sequential Key-Code Scanning

The keyboard detects all keys pressed and sends each scan code in the correct sequence. When not being serviced by the system, the keyboard stores the scan codes in its buffer.

Buffer

A 16-byte first-in-first-out (FIFO) buffer in the keyboard stores the scan codes until the system is ready to receive them. A buffer-overrun condition occurs when more than 16 bytes are placed in the keyboard buffer. An overrun code replaces the 17th byte. If more keys are pressed before the system allows keyboard output, the additional data is lost.

When the keyboard is allowed to send data, the bytes in the buffer are sent as in normal operation, and new data entered is detected and sent. Response codes do not occupy a buffer position.

If keystrokes generate a multiple-byte sequence, the entire sequence must fit into the available buffer space, or the keystroke is discarded and a buffer-overrun condition occurs.

Keys

Except for the Pause key, all keys are *make/break*. The make scan code of a key is sent to the keyboard controller when the key is pressed. When the key is released, its break scan code is sent.

Also, except for the Pause key, all keys are *typematic*. When a key is pressed and held down, the keyboard sends the make code for that key, delays 500 milliseconds ±20%, and begins sending a make code for that key at a rate of 10.9 characters per second ±20%. The typematic rate and delay can be modified (see "Set Typematic Rate/Delay (Hex F3)" on page 24).

If two or more keys are held down, only the last key pressed repeats at the typematic rate. Typematic operation stops when the last key pressed is released, even if other keys are still held down. If a key is pressed and held down while keyboard transmission is inhibited, only the first make code is stored in the buffer. This prevents buffer overflow caused by typematic action.

Note: Scan-code set 3 allows key types to be changed by the system. See "Set 3 Scan Code Tables" on page 34 for the default settings.

Power-On Routine

The following activities take place when power is first applied to the keyboard:

Power-On Reset (POR)

The keyboard logic generates a 'power-on reset' signal when power is first applied to the keyboard. A POR takes a minimum of 150 milliseconds and a maximum of 2.0 seconds from the time power is first applied to the keyboard.

Basic Assurance Test

The basic assurance test (BAT) consists of a keyboard processor test, a checksum of the read-only memory (ROM), and a random-access memory (RAM) test. During the BAT, activity on the 'clock' and 'data' lines is ignored. The LEDs are turned on at the beginning and off at the end of the BAT. The BAT takes a minimum of 300 milliseconds and a maximum of 500 milliseconds. This is in addition to the time required by the POR.

On satisfactory completion of the BAT, a completion code (hex AA) is sent to the system, and keyboard scanning begins. If a BAT failure occurs, the keyboard sends an error code to the system. The keyboard is then disabled pending command input. Completion codes are sent between 450 milliseconds and 2.5 seconds after the POR, and between 300 and 500 milliseconds after a Reset command is acknowledged.

Immediately following a POR, the keyboard monitors the signals on the keyboard 'clock' and 'data' lines and sets the line protocol.

Commands from the System

The following figure shows the commands that the system may send and their hexadecimal values.

Command	Hex Value
Set/Reset Status Indicators	ED
Echo	EE
Invalid Command	EF
Select Alternate Scan Codes	F0
Invalid Command	F1
Read ID	F2
Set Typematic Rate/Delay	F3
Enable	F4
Default Disable	F5
Set Default	F6
Set All Keys - Typematic	F7
- Make/Break	F8
- Make	F9
- Typematic/Make/Break	FA
Set Key Type - Typematic	FB
- Make/Break	FC
- Make	FD
Resend	FE
Reset	FF

Figure 1. Keyboard Commands from the System

These commands can be sent to the keyboard at any time. The keyboard responds within 20 milliseconds, except when performing the BAT or executing a Reset command.

Note: Mode 1 accepts only the Reset command.

The following commands are in alphabetic order. They have different meanings when issued by the keyboard (see "Commands to the System" on page 26).

Default Disable (Hex F5): The Default Disable command resets all conditions to the power-on default state. The keyboard responds with ACK, clears its output buffer, sets the default key types (scan-code set 3 operation only) and typematic rate/delay, and clears the last typematic key. The keyboard stops scanning and awaits further instructions.

Echo (Hex EE): Echo is a diagnostic aid. When the keyboard receives this command, it issues a hex EE response. If previously enabled, it continues scanning.

Enable (Hex F4): On receipt of this command, the keyboard responds with ACK, clears its output buffer, clears the last typematic key, and starts scanning.

Invalid Command (Hex EF and F1): Hex EF and hex F1 are invalid commands and are not supported. If one of these is sent, the keyboard does not acknowledge the command but returns a Resend command and continues in its prior scanning state. No other activities occur.

Read ID (Hex F2): This command requests identification information from the keyboard. The keyboard responds with ACK, stops scanning, and sends the two keyboard ID bytes. The second byte must follow completion of the first by no more than 500 microseconds. After the output of the second ID byte, the keyboard resumes scanning.

Resend (Hex FE): The system sends this command when it detects an error in any transmission from the keyboard. It is sent only after a keyboard transmission and before the system allows the next keyboard output. When a Resend command is received, the keyboard sends the previous output again (unless the previous output was the Resend command, in which case the keyboard sends the last byte before the Resend command).

Reset (Hex FF): The system issues a Reset command to start a program reset and a keyboard internal self-test. The keyboard acknowledges the command with an ACK and ensures the system accepts ACK before executing the command. The system signals acceptance of ACK by raising the 'clock' and 'data' lines for a minimum of 500 microseconds. The keyboard is disabled from the time it receives the Reset command until ACK is accepted, or until another command is sent that overrides the previous command.

Following acceptance of ACK, the keyboard is reinitialized and performs the BAT. After returning the completion code, the keyboard defaults to scan-code set 2.

Select Alternate Scan Codes (Hex F0): This command instructs the keyboard to select one of three sets of scan codes. The keyboard acknowledges receipt of this command with ACK and clears both the output buffer and the typematic key (if one is active). The system then sends the option byte and the keyboard responds with another ACK. An option byte value of hex 01 selects scan code set 1, hex 02 selects scan code set 2, and hex 03 selects scan code set 3.

An option byte value of hex 00 causes the keyboard to acknowledge with an ACK and send a byte telling the system which scan code set is currently in use. To prevent the controller from translating this byte, disable the keyboard-controller translate mode.

After establishing the new scan code set, the keyboard returns to the scanning state it was in before receiving the Select Alternate Scan Codes command.

Set All Keys (Hex F7, F8, F9, FA)

These commands instruct the keyboard to set all keys to a condition listed in the following figure.

Hex Value	Command
F7	Set All Keys - Typematic
F8	Set All Keys - Make/Break
F9	Set All Keys - Make
FA	Set All Keys - Typematic/Make/Break

Figure 2. Set All Keys Commands

The keyboard responds with ACK, clears its output buffer, sets all keys to the condition indicated by the command, and continues scanning (if it was previously enabled). Although these commands can be sent using any scan-code set, they affect only the operation of scan-code set 3.

Set Default (Hex F6): The Set Default command resets all conditions to the power-on default state. The keyboard responds with ACK, clears its output buffer, sets the default key types (scan-code set 3 operation only) and typematic rate/delay, clears the last typematic key, and continues scanning.

Set Key Type (Hex FB, FC, FD): These commands instruct the keyboard to set individual keys to a condition listed in the following figure.

Hex Value	Command
FB	Set Key Type - Typematic
FC	Set Key Type - Make/Break
FD	Set Key Type - Make

Figure 3. Set Key Type Commands

The keyboard responds with ACK, clears its output buffer, and prepares to receive key identification. The system identifies each key

by its scan-code value, as defined in scan-code set 3. Only scan code set 3 values are valid for key identification. The type of each identified key is set to the value indicated by the command.

These commands can be sent using any scan code set, but affect only the operation of scan code set 3.

Set/Reset Status Indicators (Hex ED): Three status indicators on the keyboard—Num Lock, Caps Lock, and Scroll Lock—are accessible by the system. The keyboard activates or deactivates these indicators when it receives a valid command-code sequence from the system. The command sequence begins with the command byte (hex ED). The keyboard responds with ACK, stops scanning, and waits for the option byte from the system. The bit assignments for this option byte are as follows.

Bit	Function
7 - 3	Reserved (must be 0's)
2	Caps Lock Indicator
1	Num Lock Indicator
0	Scroll Lock Indicator

Figure 4. Set/Reset Status Indicators

If a bit for an indicator is set to 1, the indicator is turned on. If a bit is set to 0, the indicator is turned off.

The keyboard responds to the option byte with ACK, sets the indicators and, if the keyboard was previously enabled, continues scanning. The state of the indicators reflects the bits in the option byte and can be activated or deactivated in any combination. If another command is received in place of the option byte, execution of the Set/Reset Mode Indicators command is stopped, with no change to the indicator states, and the new command is processed.

Immediately after power-on, the lights default to the Off state. If the Set Default and Default Disable commands are received, the lamps remain in the state they were in before the command was received.

Set Typematic Rate/Delay (Hex F3): The system issues this command to change the typematic rate and delay. The keyboard responds to the command with ACK, stops scanning, and waits for the system to issue the rate/delay value byte. The keyboard responds to the rate/delay value byte with another ACK, sets the rate and delay to the values indicated, and continues scanning (if it was previously enabled). Bits 6 and 5 indicate the delay, and bits 4, 3, 2, 1, and 0 (the

least-significant bit) indicate the rate. Bit 7, the most-significant bit, is always 0. The delay is determined by the following equation:

Delay = (1 + A) × 250 milliseconds ±20%.
where:
A = binary value of bits 5 and 6

The period (interval from one typematic output to the next) is determined by the following equation:

Period = (8 + A) × (2^B) × 0.00417 seconds ±20%
where:
A = binary value of bits 2, 1, and 0
B = binary value of bits 4 and 3

The typematic rate (make codes per second) is 1 for each period.

Bit	Typematic Rate ± 20%	Bit	Typematic Rate ± 20%
00000	30.0	10000	7.5
00001	26.7	10001	6.7
00010	24.0	10010	6.0
00011	21.8	10011	5.5
00100	20.0	10100	5.0
00101	18.5	10101	4.6
00110	17.1	10110	4.3
00111	16.0	10111	4.0
01000	15.0	11000	3.7
01001	13.3	11001	3.3
01010	12.0	11010	3.0
01011	10.9	11011	2.7
01100	10.0	11100	2.5
01101	9.2	11101	2.3
01110	8.6	11110	2.1
01111	8.0	11111	2.0

Figure 5. Typematic Rate

The default values for the system keyboard are as follows:

Typematic rate = 10.9 characters per second ± 20%

Delay = 500 milliseconds ± 20%.

The execution of this command stops without change to the existing rate if another command is received instead of the rate/delay value byte.

Commands to the System

The following figure shows the commands that the keyboard can send to the system, and their hexadecimal values.

Command	Hex Value
Key Detection Error/Overrun	00 (Code Sets 2 and 3)
Keyboard ID	83AB
BAT Completion Code	AA
BAT Failure Code	FC
Echo	EE
Acknowledge (ACK)	FA
Resend	FE
Key Detection Error/Overrun	FF (Code Set 1)

Figure 6. Keyboard Commands to the System

The commands the keyboard sends to the system are described in alphabetic order. They have different meanings when issued by the system.

Acknowledge (Hex FA): The keyboard issues ACK to any valid input other than an Echo, or Resend command. If the keyboard is interrupted while sending ACK, it discards ACK and accepts and responds to the new command.

BAT Completion Code (Hex AA): Following satisfactory completion of the BAT, the keyboard sends hex AA. Any other code indicates a failure of the keyboard.

BAT Failure Code (Hex FC): If a BAT failure occurs, the keyboard sends this code, stops scanning, and waits for a system response or reset.

Echo (Hex EE): The keyboard sends this code in response to an Echo command.

Keyboard ID (Hex 83AB): The keyboard ID consists of two bytes, hex 83AB. The keyboard responds to the Read ID command with ACK, stops scanning, and sends the two ID bytes. The low byte is sent first followed by the high byte. Following the output of the keyboard ID, the keyboard begins scanning. Because of keyboard controller translation, the keyboard ID might not be returned to the system as hex 83AB.

Key Detection Error (Hex 00 or FF): The keyboard sends a key detection error character if conditions in the keyboard make it impossible to identify a switch closure. If the keyboard is using scan-code set 1, the code is hex FF. For sets 2 and 3, the code is hex 00.

Overrun (Hex 00 or FF): An overrun character is placed in the keyboard buffer and replaces the last code when the buffer capacity has been exceeded. The code is sent to the system when it reaches the top of the buffer queue. If the keyboard is using scan code set 1, the code is hex FF. For sets 2 and 3, the code is hex 00.

Resend (Hex FE): The keyboard issues a Resend command following receipt of an invalid input, or any input with incorrect parity. If the system sends nothing to the keyboard, no response is required.

Scan Codes

The following figures list the key numbers of the three scan code sets and their hexadecimal values. The system defaults to scan set 2, but can be switched to set 1 or set 3 (see "Select Alternate Scan Codes (Hex F0)" on page 22).

Set 1 Scan-Code Tables

In scan-code set 1, each key is assigned a base scan code and, sometimes, extra codes to generate artificial shift states in the system. The typematic scan codes are identical to the base scan code for each key.

The following figure shows the codes sent for the keys, regardless of any shift states in the keyboard or system. Refer to "Keyboard Layouts" beginning on page 1 to determine the character associated with each key number.

Key Number	Make Code	Break Code	Key Number	Make Code	Break Code
1	29	A9	47	2D	AD
2	02	82	48	2E	AE
3	03	83	49	2F	AF
4	04	84	50	30	B0
5	05	85	51	31	B1
6	06	86	52	32	B2
7	07	87	53	33	B3
8	08	88	54	34	B4
9	09	89	55	35	B5
10	0A	8A	57	36	B6
11	0B	8B	58	1D	9D
12	0C	8C	60	38	B8
13	0D	8D	61	39	B9
15	0E	8E	62	E0 38	E0 B8
16	0F	8F	64	E0 1D	E0 9D
17	10	90	90	45	C5
18	11	91	91	47	C7
19	12	92	92	4B	CB
20	13	93	93	4F	CF
21	14	94	96	48	C8
22	15	95	97	4C	CC
23	16	96	98	50	D0
24	17	97	99	52	D2
25	18	98	100	37	B7
26	19	99	101	49	C9
27	1A	9A	102	4D	CD
28	1B	9B	103	51	D1
*29	2B	AB	104	53	D3
30	3A	BA	105	4A	CA
31	1E	9E	106	4E	CE
32	1F	9F	108	E01C	E0 9C
33	20	A0	110	01	81
34	21	A1	112	3B	BB
35	22	A2	113	3C	BC
36	23	A3	114	3D	BD
37	24	A4	115	3E	BE
38	25	A5	116	3F	BF
39	26	A6	117	40	C0
40	27	A7	118	41	C1
41	28	A8	119	42	C2
** 42	2B	AB	120	43	C3
43	1C	9C	121	44	C4
44	2A	AA	122	57	D7
** 45	56	D6	123	58	D8
46	2C	AC	125	46	C6

Note: * 101-key keyboard only, ** 102-key keyboard only.

Figure 7. Keyboard Scan Codes, Set 1

The remaining keys send a series of codes that are dependent on the states of the various shift keys (Ctrl, Alt, and Shift), and the state of Num Lock (On or Off). Because the base scan code is identical to another key, an extra code (hex E0) has been added to the base code to make it unique.

Key No.	Base Case, or Shift + Num Lock Make/Break	Shift Case Make/Break *	Num Lock on Make/Break
75	E0 52	E0 AA E0 52	E0 2A E0 52
	/E0 D2	/E0 D2 E0 2A	/E0 D2 E0 AA
76	E0 53	E0 AA E0 53	E0 2A E0 53
	/E0 D3	/E0 D3 E0 2A	/E0 D3 E0 AA
79	E0 4B	E0 AA E0 4B	E0 2A E0 4B
	/E0 CB	/E0 CB E0 2A	/E0 CB E0 AA
80	E0 47	E0 AA E0 47	E0 2A E0 47
	/E0 C7	/E0 C7 E0 2A	/E0 C7 E0 AA
81	E0 4F	E0 AA E0 4F	E0 2A E0 4F
	/E0 CF	/E0 CF E0 2A	/E0 CF E0 AA
83	E0 48	E0 AA E0 48	E0 2A E0 48
	/E0 C8	/E0 C8 E0 2A	/E0 C8 E0 AA
84	E0 50	E0 AA E0 50	E0 2A E0 50
	/E0 D0	/E0 D0 E0 2A	/E0 D0 E0 AA
85	E0 49	E0 AA E0 49	E0 2A E0 49
	/E0 C9	/E0 C9 E0 2A	/E0 C9 E0 AA
86	E0 51	E0 AA E0 51	E0 2A E0 51
	/E0 D1	/E0 D1 E0 2A	/E0 D1 E0 AA
89	E0 4D	E0 AA E0 4D	E0 2A E0 4D
	/E0 CD	/E0 CD E0 2A	/E0 CD E0 AA

Note: * If the left Shift key is held down, the AA/2A shift make and break are sent with the other scan codes. If the right Shift key is held down, B6/36 is sent. If both Shift keys are down, both sets of codes are sent with the other scan code.

Figure 8. Keyboard Scan Codes, Set 1

Key No.	Scan Code Make/Break	Shift Case Make/Break *
95	E0 35/E0 B5	E0 AA E0 35/E0 B5 E0 2A

Note: * If the left Shift key is held down, the AA/2A shift make and break are sent with the other scan codes. If the right Shift key is held down, B6/36 is sent. If both Shift keys are down, both sets of codes are sent with the other scan code.

Figure 9. Keyboard Scan Codes, Set 1

Key No.	Scan Code Make/Break	Ctrl Case, Shift Case Make/Break	Alt Case Make/Break
124	E0 2A E0 37 /E0 B7 E0 AA	E0 37/E0 B7	54/D4

Figure 10. Keyboard Scan Codes, Set 1

Key No.	Make Code	Ctrl Key Pressed
126 *	E1 1D 45 E1 9D C5	E0 46 E0 C6

Note: * This key is not typematic. All associated scan codes occur on the make of the key.

Figure 11. Keyboard Scan Codes, Set 1

Set 2 Scan-Code Tables

In scan-code set 2, each key is assigned a unique 8-bit make scan code, which is sent when the key is pressed. Each key also sends a break code when the key is released. The break code consists of 2 bytes, the first of which is the break code prefix, hex F0; the second byte is the same as the make scan code for that key. The typematic scan code for a key is the same as the key's make code.

The following figure shows the codes sent for the keys, regardless of any shift states in the keyboard or system. Refer to "Keyboard Layouts" beginning on page 1 to determine the character associated with each key number.

Key Number	Make Code	Break Code	Key Number	Make Code	Break Code
1	0E	F0 0E	47	22	F0 22
2	16	F0 16	48	21	F0 21
3	1E	F0 1E	49	2A	F0 2A
4	26	F0 26	50	32	F0 32
5	25	F0 25	51	31	F0 31
6	2E	F0 2E	52	3A	F0 3A
7	36	F0 36	53	41	F0 41
8	3D	F0 3D	54	49	F0 49
9	3E	F0 3E	55	4A	F0 4A
10	46	F0 46	57	59	F0 59
11	45	F0 45	58	14	F0 14
12	4E	F0 4E	60	11	F0 11
13	55	F0 55	61	29	F0 29
15	66	F0 66	62	E0 11	E0 F0 11
16	0D	F0 0D	64	E0 14	E0 F0 14
17	15	F0 15	90	77	F0 77
18	1D	F0 1D	91	6C	F0 6C
19	24	F0 24	92	6B	F0 6B
20	2D	F0 2D	93	69	F0 69
21	2C	F0 2C	96	75	F0 75
22	35	F0 35	97	73	F0 73
23	3C	F0 3C	98	72	F0 72
24	43	F0 43	99	70	F0 70
25	44	F0 44	100	7C	F0 7C
26	4D	F0 4D	101	7D	F0 7D
27	54	F0 54	102	74	F0 74
28	5B	F0 5B	103	7A	F0 7A
* 29	5D	F0 5D	104	71	F0 71
30	58	F0 58	105	7B	F0 7B
31	1C	F0 1C	106	79	F0 79
32	1B	F0 1B	108	E0 5A	E0 F0 5A
33	23	F0 23	110	76	F0 76
34	2B	F0 2B	112	05	F0 05
35	34	F0 34	113	06	F0 06
36	33	F0 33	114	04	F0 04
37	3B	F0 3B	115	0C	F0 0C
38	42	F0 42	116	03	F0 03
39	4B	F0 4B	117	0B	F0 0B
40	4C	F0 4C	118	83	F0 83
41	52	F0 52	119	0A	F0 0A
** 42	5D	F0 5D	120	01	F0 01
43	5A	F0 5A	121	09	F0 09
44	12	F0 12	122	78	F0 78
** 45	61	F0 61	123	07	F0 07
46	1A	F0 1A	125	7E	F0 7E

Note: * 101-key keyboard only, ** 102-key keyboard only.

Figure 12. Keyboard Scan Codes, Set 2

The remaining keys send a series of codes that are dependent on the state of the shift keys (Ctrl, Alt, and Shift), and the state of Num Lock (On or Off). Because the base scan code is identical to another key, an extra code (hex E0) is added to the base code to make it unique.

Key No.	Base Case, or Shift + Num Lock Make/Break	Shift Case Make/Break *	Num Lock on Make/Break
75	E0 70	E0 F0 12 E0 70	E0 12 E0 70
	/E0 F0 70	/E0 F0 70 E0 12	/E0 F0 70 E0 F0 12
76	E0 71	E0 F0 12 E0 71	E0 12 E0 71
	/E0 F0 71	/E0 F0 71 E0 12	/E0 F0 71 E0 F0 12
79	E0 6B	E0 F0 12 E0 6B	E0 12 E0 6B
	/E0 F0 6B	/E0 F0 6B E0 12	/E0 F0 6B E0 F0 12
80	E0 6C	E0 F0 12 E0 6C	E0 12 E0 6C
	/E0 F0 6C	/E0 F0 6C E0 12	/E0 F0 6C E0 F0 12
81	E0 69	E0 F0 12 E0 69	E0 12 E0 69
	/E0 F0 69	/E0 F0 69 E0 12	/EO F0 69 E0 F0 12
83	E0 75	E0 F0 12 E0 75	E0 12 E0 75
	/E0 F0 75	/E0 F0 75 E0 12	/E0 F0 75 E0 F0 12
84	E0 72	E0 F0 12 E0 72	E0 12 E0 72
	/E0 F0 72	/E0 F0 72 E0 12	/E0 F0 72 E0 F0 12
85	E0 7D	E0 FO 12 E0 7D	E0 12 E0 7D
	/E0 F0 7D	/E0 F0 7D E0 12	/E0 F0 7D E0 F0 12
86	E0 7A	E0 F0 12 E0 7A	E0 12 E0 7A
	/E0 F0 7A	/E0 F0 7A E0 12	/E0 F0 7A E0 F0 12
89	E0 74	E0 F0 12 E0 74	E0 12 E0 74
	/E0 F0 74	/E0 F0 74 E0 12	/E0 F0 74 E0 F0 12

Note: * If the left Shift key is held down, the F0 12/12 shift make and break are sent with the other scan codes. If the right Shift key is held down, F0/59/59 is sent. If both Shift keys are down, both sets of codes are sent with the other scan code.

Figure 13. Keyboard Scan Codes, Set 2

Key No.	Scan Code Make/Break	Shift Case Make/Break *
95	E0 4A/E0 F0 4A	E0 F0 12 E0 4A/E0 F0 4A E0 12

Note: * If the left Shift key is held down, the F0 12/12 shift make and break are sent with the other scan codes. If the right Shift key is held down, F0 59/59 is sent. If both Shift keys are down, both sets of codes are sent with the other scan code.

Figure 14. Keyboard Scan Codes, Set 2

Key No.	Scan Code Make/Break	Ctrl Case, Shift Case Make/Break	Alt Case Make/Break
124	E0 12 E0 7C /E0 F0 7C E0 F0 12	E0 7C/E0 F0 7C	84/F0 84

Figure 15. Keyboard Scan Codes, Set 2

Key No.	Make Code	Ctrl Key Pressed
126 *	E1 14 77 E1 F0 14 F0 77	E0 7E E0 F0 7E

Note: * This key is not typematic. All associated scan codes occur on the make of the key.

Figure 16. Keyboard Scan Codes, Set 2

Set 3 Scan Code Tables

In scan-code set 3, each key is assigned a unique 8-bit make scan code, which is sent when the key is pressed. Each key also sends a break code when the key is released. The break code consists of two bytes, the first of which is the break-code prefix, hex F0; the second byte is the same as the make scan code for that key. The typematic scan code for a key is the same as the key's make code. With this scan-code set, each key sends only one scan code, and no keys are affected by the state of any other keys.

The following figure shows the codes sent for the keys, regardless of any shift states in the keyboard or system. Refer to "Keyboard Layouts" beginning on page 1 to determine the character associated with each key number.

Key Number	Make Code	Break Code	Default Key State
1	0E	F0 0E	Typematic
2	16	F0 16	Typematic
3	1E	F0 1E	Typematic
4	26	F0 26	Typematic
5	25	F0 25	Typematic
6	2E	F0 2E	Typematic
7	36	F0 36	Typematic
8	3D	F0 3D	Typematic
9	3E	F0 3E	Typematic
10	46	F0 46	Typematic
11	45	F0 45	Typematic
12	4E	F0 4E	Typematic
13	55	F0 55	Typematic
15	66	F0 66	Typematic
16	0D	F0 0D	Typematic
17	15	F0 15	Typematic
18	1D	F0 1D	Typematic
19	24	F0 24	Typematic
20	2D	F0 2D	Typematic
21	2C	F0 2C	Typematic
22	35	F0 35	Typematic
23	3C	F0 3C	Typematic
24	43	F0 43	Typematic
25	44	F0 44	Typematic
26	4D	F0 4D	Typematic
27	54	F0 54	Typematic
28	5B	F0 5B	Typematic
* 29	5C	F0 5C	Typematic
30	14	F0 14	Make/Break

Note: * 101-key keyboard only and ** 102-key keyboard only.

Figure 17 (Part 1 of 3). Keyboard Scan Codes, Set 3

Key Number	Make Code	Break Code	Default Key State
31	1C	F0 1C	Typematic
32	1B	F0 1B	Typematic
33	23	F0 23	Typematic
34	2B	F0 2B	Typematic
35	34	F0 34	Typematic
36	33	F0 33	Typematic
37	3B	F0 3B	Typematic
38	42	F0 42	Typematic
39	4B	F0 4B	Typematic
40	4C	F0 4C	Typematic
41	52	F0 52	Typematic
** 42	53	F0 53	Typematic
43	5A	F0 5A	Typematic
44	12	F0 12	Make/Break
** 45	13	F0 13	Typematic
46	1A	F0 1A	Typematic
47	22	F0 22	Typematic
48	21	F0 21	Typematic
49	2A	F0 2A	Typematic
50	32	F0 32	Typematic
51	31	F0 31	Typematic
52	3A	F0 3A	Typematic
53	41	F0 41	Typematic
54	49	F0 49	Typematic
55	4A	F0 4A	Typematic
57	59	F0 59	Make/Break
58	11	F0 11	Make/Break
60	19	F0 19	Make/Break
61	29	F0 29	Typematic
62	39	F0 39	Make only
64	58	F0 58	Make only
75	67	F0 67	Make only
76	64	F0 64	Typematic
79	61	F0 61	Typematic
80	6E	F0 6E	Make only
81	65	F0 65	Make only
83	63	F0 63	Typematic
84	60	F0 60	Typematic
85	6F	F0 6F	Make only
86	6D	F0 6D	Make only
89	6A	F0 6A	Typematic
90	76	F0 76	Make only
91	6C	F0 6C	Make only
92	6B	F0 6B	Make only
93	69	F0 69	Make only
95	77	F0 77	Make only
96	75	F0 75	Make only
97	73	F0 73	Make only
98	72	F0 72	Make only
99	70	F0 70	Make only
100	7E	F0 7E	Make only

Note: * 101-key keyboard only and ** 102-key keyboard only.

Figure 17 (Part 2 of 3). Keyboard Scan Codes, Set 3

Key Number	Make Code	Break Code	Default Key State
101	7D	F0 7D	Make only
102	74	F0 74	Make only
103	7A	F0 7A	Make only
104	71	F0 71	Make only
105	84	F0 84	Make only
106	7C	F0 7C	Typematic
108	79	F0 79	Make only
110	08	F0 08	Make only
112	07	F0 07	Make only
113	0F	F0 0F	Make only
114	17	F0 17	Make only
115	1F	F0 1F	Make only
116	27	F0 27	Make only
117	2F	F0 2F	Make only
118	37	F0 37	Make only
119	3F	F0 3F	Make only
120	47	F0 47	Make only
121	4F	F0 4F	Make only
122	56	F0 56	Make only
123	5E	F0 5E	Make only
124	57	F0 57	Make only
125	5F	F0 5F	Make only
126	62	F0 62	Make only

Note: * 101-key keyboard only and ** 102-key keyboard only.

Figure 17 (Part 3 of 3). Keyboard Scan Codes, Set 3

Clock and Data Signals

The keyboard and system communicate over the 'clock' and 'data' lines. The source of each of these lines is an open-collector device on the keyboard that allows either the keyboard or system to force a line to an inactive (low) level. When no communication is occurring, the 'clock' line is at an active (high) level. The state of the 'data' line is held active (high) by the keyboard.

When the system sends data to the keyboard, it forces the 'data' line to an inactive level and allows the 'clock' line to go to an active level.

An inactive signal has a value of at least 0 volts, but not more than +0.7 volts. A signal at the inactive level is a logical 0. An active signal has a value of at least +2.4 volts, but not more than +5.5 volts. A signal at the active level is a logical 1. Voltages are measured between a signal source and the dc network ground.

When the keyboard sends data to, or receives data from the system, it generates the 'clock' signal to time the data. The system can prevent the keyboard from sending data by forcing the 'clock' line to an inactive level; the 'data' line can be active or inactive during this time.

During the BAT, the keyboard allows the 'clock' and 'data' lines to go to an active level.

Data Stream

Data transmissions to and from the keyboard consist of an 11-bit data stream (Mode 2) sent serially over the 'data' line. The following figure shows the functions of the bits.

Bit	Function
11	Stop bit (always 1)
10	Parity bit (odd parity)
9	Data bit 7 (most-significant)
8	Data bit 6
7	Data bit 5
6	Data bit 4
5	Data bit 3
4	Data bit 2
3	Data bit 1
2	Data bit 0 (least-significant)
1	Start bit (always 0)

Figure 18. Keyboard Data Stream Bit Definitions

The parity bit is either 1 or 0, and the 8 data bits, plus the parity bit, always have an odd number of 1's.

Note: Mode 1 is a 9-bit data stream that does not have a parity bit or stop bit, and the start bit is always 1.

Data Output

When the keyboard is ready to send data, it first checks for a keyboard-inhibit or system request-to-send status on the 'clock' and 'data' lines. If the 'clock' line is inactive (low), data is stored in the keyboard buffer. If the 'clock' line is active (high) and the 'data' line is inactive (request-to-send), data is stored in the keyboard buffer, and the keyboard receives system data.

If the 'clock' and 'data' lines are both active, the keyboard sends the 0 start bit, 8 data bits, the parity bit, and the stop bit. Data is valid before the trailing edge and beyond the leading edge of the clock pulse. During transmission, the keyboard checks the 'clock' line for an active level at least every 60 milliseconds. If the system lowers the 'clock' line from an active level after the keyboard starts sending data, a condition known as *line contention* occurs, and the keyboard stops sending data. If line contention occurs before the leading edge of the 10th clock signal (parity bit), the keyboard buffer returns the 'clock' and 'data' lines to an active level. If contention does not occur by the 10th clock signal, the keyboard completes the transmission. Following line contention, the system may or may not request the keyboard to resend the data.

Following a transmission, the system can inhibit the keyboard until the system processes the input, or until it requests that a response be sent.

Data Input

When the system is ready to send data to the keyboard, it first checks to see if the keyboard is sending data. If the keyboard is sending, but has not reached the 10th 'clock' signal, the system can override the keyboard output by forcing the keyboard 'clock' line to an inactive (low) level. If the keyboard transmission is beyond the 10th 'clock' signal, the system must receive the transmission.

If the keyboard is not sending, or if the system elects to override the keyboard's output, the system forces the keyboard 'clock' line to an inactive level for more than 60 microseconds while preparing to send data. When the system is ready to send the start bit (the 'data' line will be inactive), it allows the 'clock' line to go to an active (high) level.

The keyboard checks the state of the 'clock' line at intervals of no more than 10 milliseconds. If a system request-to-send signal (RTS) is detected, the keyboard counts 11 bits. After the 10th bit, the keyboard checks for an active level on the 'data' line, and if the line is active, forces it inactive and counts one more bit. This action signals the system that the keyboard has received its data. On receipt of this signal, the system returns to a ready state, in which it can accept keyboard output, or goes to the inhibited state until it is ready.

If the keyboard 'data' line is found at an inactive level following the 10th bit, a framing error has occurred, and the keyboard continues to count until the 'data' line becomes active. The keyboard then makes the 'data' line inactive and sends a Resend command.

Each system command or data transmission to the keyboard requires a response from the keyboard before the system can send its next output. The keyboard will respond within 20 milliseconds unless the system prevents keyboard output. If the keyboard response is invalid or has a parity error, the system sends the command or data again. However, two-byte commands require special handling. If hex F3 (Set Typematic Rate/Delay), hex F0 (Select Alternate Scan Codes), or hex ED (Set/Reset Mode Indicators) have been sent and acknowledged, and the value byte has been sent but the response is invalid or has a parity error, the system resends both the command and the value byte.

Encode and Usage

The keyboard routine, provided in the ROM BIOS, is responsible for converting the keyboard scan codes into what is called *Extended ASCII*. The extended ASCII codes returned by the ROM routine are mapped to the U.S. English keyboard layout. Some operating systems might make provisions for alternate keyboard layouts by providing an interrupt replacement routine, which resides in the read/write memory. This section discusses only the ROM routine.

Extended ASCII encompasses one-byte character codes with possible values of 0 to 255, an extended code for certain extended keyboard functions, and functions handled within the keyboard routine or through interrupts.

The character codes are passed through the BIOS keyboard routine to the system or application program. In the following figure "-1" means the combination is suppressed in the keyboard routine. The codes are returned in the AL register.

Key	Base Case	Uppercase	Ctrl	Alt
1	`	~	-1	(*)
2	1	!	-1	(*)
3	2	@	Null(000) (*)	(*)
4	3	#	-1	(*)
5	4	$	-1	(*)
6	5	%	-1	(*)
7	6	^	RS(030)	(*)
8	7	&	-1	(*)
9	8	*	-1	(*)
10	9	(	-1	(*)
11	0	)	-1	(*)
12	-	_	US(031)	(*)
13	=	+	-1	(*)
15	Backspace (008)	Backspace (008)	Del(127)	(*)
16	→| (009)	|← (*)	(*)	(*)
17	q	Q	DC1(017)	(*)
18	w	W	ETB(023)	(*)
19	e	E	ENQ(005)	(*)
20	r	R	DC2(018)	(*)
21	t	T	DC4(020)	(*)
22	y	Y	EM(025)	(*)
23	u	U	NAK(021)	(*)
24	i	I	HT(009)	(*)
25	o	O	SI(015)	(*)
26	p	P	DLE(016)	(*)
27	[	{	Esc(027)	(*)
28	]	}	GS(029)	(*)
29	\	|	FS(028)	(*)
30 Caps Lock	-1	-1	-1	-1
31	a	A	SOH(001)	(*)
32	s	S	DC3(019)	(*)
33	d	D	EOT(004)	(*)
34	f	F	ACK(006)	(*)
35	g	G	BEL(007)	(*)
36	h	H	BS(008)	(*)
37	j	J	LF(010)	(*)
38	k	K	VT(011)	(*)
39	l	L	FF(012)	(*)
40	;	:	-1	(*)

Note: (*) Refer to "Extended Functions" on page 43. (**) Refer to "Special Handling" on page 47.

Figure 19 (Part 1 of 2). Character Codes

Key	Base Case	Uppercase	Ctrl	Alt
41	'	"	-1	(*)
43	CR(013)	CR(013)	LF(010)	(*)
44 Shift (Left)	-1	-1	-1	-1
46	z	Z	SUB(026)	(*)
47	x	X	CAN(024)	(*)
48	c	C	ETX(003)	(*)
49	v	V	SYN(022)	(*)
50	b	B	STX(002)	(*)
51	n	N	SO(014)	(*)
52	m	M	CR(013)	(*)
53	,	<	-1	(*)
54	.	>	-1	(*)
55	/	?	-1	(*)
57 Shift (Right)	-1	-1	-1	-1
58 Ctrl (Left)	-1	-1	-1	-1
60 Alt (Left)	-1	-1	-1	-1
61	Space	Space	Space	Space
62 Alt (Right)	-1	-1	-1	-1
64 Ctrl (Right)	-1	-1	-1	-1
90 Num Lock	-1	-1	-1	-1
95	/	/	(*)	(*)
100	*	*	(*)	(*)
105	-	-	(*)	(*)
106	+	+	(*)	(*)
108	Enter	Enter	LF(010)	(*)
110	Esc	Esc	Esc	(*)
112	Null (*)	Null (*)	Null (*)	Null(*)
113	Null (*)	Null (*)	Null (*)	Null(*)
114	Null (*)	Null (*)	Null (*)	Null(*)
115	Null (*)	Null (*)	Null (*)	Null(*)
116	Null (*)	Null (*)	Null (*)	Null(*)
117	Null (*)	Null (*)	Null (*)	Null(*)
118	Null (*)	Null (*)	Null (*)	Null(*)
119	Null (*)	Null (*)	Null (*)	Null(*)
120	Null (*)	Null (*)	Null (*)	Null(*)
121	Null (*)	Null (*)	Null (*)	Null(*)
122	Null (*)	Null (*)	Null (*)	Null(*)
123	Null (*)	Null (*)	Null (*)	Null(*)
125 Scroll Lock	-1	-1	-1	-1
126	Pause(**)	Pause(**)	Break(**)	Pause(**)

Note: (*) Refer to "Extended Functions" on page 43. (**) Refer to "Special Handling" on page 47.

Figure 19 (Part 2 of 2). Character Codes

The following figure is a list of keys that have meaning only in Num Lock, Shift, or Ctrl states.

The Shift key temporarily reverses the current Num Lock state.

Key	Num Lock	Base Case	Alt	Ctrl
91	7	Home (*)	-1	Clear Screen
92	4	← (*)	-1	Reverse Word(*)
93	1	End (*)	-1	Erase to EOL(*)
96	8	↑ (*)	-1	(*)
97	5	(*)	-1	(*)
98	2	↓ (*)	-1	(*)
99	0	Ins	-1	(*)
101	9	Page Up (*)	-1	Top of Text and Home
102	6	→ (*)	-1	Advance Word (*)
103	3	Page Down (*)	-1	Erase to EOS (*)
104	.	Delete (*,**)	(**)	(**)
105	-	Sys Request	-1	-1
106	+	+ (*)	-1	-1

Note: (*) Refer to "Extended Functions." (**) Refer to "Special Handling" on page 47.

Figure 20. Special Character Codes

Extended Functions

For certain functions that cannot be represented by a standard ASCII code, an extended code is used. A character code of 000 (null) is returned in AL. This indicates that the system or application program should examine a second code, which indicates the actual function. Usually, but not always, this second code is the scan code of the primary key that was pressed. This code is returned in AH.

The following figure is a list of the extended codes and their functions.

Second Code	Function
1	Alt Esc
3	Null Character
14	Alt Backspace
15	⎸← (Back-tab)
16-25	Alt Q, W, E, R, T, Y, U, I, O, P
26-28	Alt [ ] ↵
30-38	Alt A, S, D, F, G, H, J, K, L
39-41	Alt ; ' `
43	Alt \
44-50	Alt Z, X, C, V, B, N, M
51-53	Alt , . /
55	Alt Keypad *
59-68	F1 to F10 Function Keys (Base Case)
71	Home
72	↑ (Cursor Up)
73	Page Up
74	Alt Keypad -
75	← (Cursor Left)
76	Center Cursor
77	→ (Cursor Right)
78	Alt Keypad +
79	End
80	↓ (Cursor Down)
81	Page Down
82	Ins (Insert)
83	Del (Delete)
84-93	Shift F1 to F10
94-103	Ctrl F1 to F10
104-113	Alt F1 to F10
114	Ctrl PrtSc (Start/Stop Echo to Printer)
115	Ctrl ← (Reverse Word)
116	Ctrl → (Advance Word)
117	Ctrl End (Erase to End of Line-EOL)
118	Ctrl PgDn (Erase to End of Screen-EOS)
119	Ctrl Home (Clear Screen and Home)
120-131	Alt 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, -, = keys 2-13
132	Ctrl PgUp (Top 25 Lines of Text and Cursor Home)
133-134	F11, F12
135-136	Shift F11, F12
137-138	Ctrl F11, F12
139-140	Alt F11, F12

Figure 21 (Part 1 of 2). Keyboard Extended Functions

Second Code	Function
141	Ctrl Up/8
142	Ctrl Keypad -
143	Ctrl Keypad 5
144	Ctrl Keypad +
145	Ctrl Down/2
146	Ctrl Ins/0
147	Ctrl Del/.
148	Ctrl Tab
149	Ctrl Keypad /
150	Ctrl Keypad *
151	Alt Home
152	Alt Up
153	Alt Page Up
155	Alt Left
157	Alt Right
159	Alt End
160	Alt Down
161	Alt Page Down
162	Alt Insert
163	Alt Delete
164	Alt Keypad /
165	Alt Tab
166	Alt Enter

Figure 21 (Part 2 of 2). Keyboard Extended Functions

Shift States

Most shift states are handled within the keyboard routine and are not apparent to the system or application program. In any case, the current status of active shift states is available by calling an entry point in the BIOS keyboard routine. The following keys result in altered shift states:

Shift: This key temporarily shifts keys 1 through 13, 16 through 29, 31 through 41, and 46 through 55 to uppercase (base case if in Caps Lock state). Also, the Shift key temporarily reverses the Num Lock or non-Num Lock state of keys 91 through 93, 96, 98, 99, and 101 through 104.

Ctrl: This key temporarily shifts keys 3, 7, 12, 15 through 29, 31 through 39, 43, 46 through 52, 75 through 89, 91 through 93, 95 through 108, 112 through 124, and 126 to the Ctrl state. The Ctrl key is also used with the Alt and Del keys to initiate the system-reset function, with the Scroll Lock key to initiate the break function, and with the Num Lock key to initiate the pause function. The system-reset, break, and pause functions are described under "Special Handling" on page 47.

Alt: This key temporarily shifts keys 1 through 29, 31 through 43, 46 through 55, 75 through 89, 95, 100, and 105 through 124 to the Alt state. The Alt key is also used with the Ctrl and Del keys to cause a system reset.

The Alt key also allows the user to enter any character code from 1 to 255. The user holds down the Alt key and types the decimal value of the characters desired on the numeric keypad (keys 91 through 93, 96 through 99, and 101 through 103). The Alt key is then released. If the number is greater than 255, a modulo-256 value is used. This value is interpreted as a character code and is sent through the keyboard routine to the system or application program. Alt is handled internally in the keyboard routine.

Caps Lock: This key shifts keys 17 through 26, 31 through 39, and 46 through 52 to uppercase. When Caps Lock is pressed again, it reverses the action. Caps Lock is handled internally in the keyboard routine. When Caps Lock is pressed, it changes the Caps Lock mode indicator. If the indicator was on, it goes off; if it was off, it goes on.

Scroll Lock: When interpreted by appropriate application programs, this key indicates that the cursor-control keys will cause windowing over the text rather than moving the cursor. When the Scroll Lock key is pressed again, it reverses the action. The keyboard routine simply records the current shift state of the Scroll Lock key. It is the responsibility of the application program to perform the function. When Scroll Lock is pressed, it changes the Scroll Lock mode indicator. If the indicator was on, it goes off; if it was off, it goes on.

Num Lock: This key shifts keys 91 through 93, 96 through 99, and 101 through 104 to uppercase. When Num Lock is pressed again, it reverses the action. Num Lock is handled internally in the keyboard routine. When Num Lock is pressed, it changes the Num Lock mode indicator. If the indicator was on, it goes off; if it was off, it goes on.

Shift Key Priorities and Combinations: If combinations of the Alt, Ctrl, and Shift keys are pressed and only one is valid, the priority is: Alt key first, Ctrl key second, and Shift key third. The only valid combination is Alt and Ctrl, which is used in the system-reset function.

Special Handling

System Reset

The combination of Alt, Ctrl, and Del keys results in the keyboard routine that starts a system reset or restart. System reset is handled by system BIOS.

Break

The combination of the Ctrl and Pause/Break keys results in the keyboard buffer being cleared. The keyboard routine then signals interrupt 1B, and the extended characters AL = hex 00, and AH = hex 00 are stored in the buffer.

Pause

The Pause key causes the keyboard interrupt routine to loop, waiting for any character or function key to be pressed. This provides a method of temporarily suspending an operation, such as listing or printing, and then resuming the operation. The method is not apparent to either the system or the application program. The key stroke used to resume operation is discarded. Pause is handled internally in the keyboard routine.

Print Screen

The Print Screen key results in an interrupt invoking the print-screen routine. This routine works in the alphanumeric or graphics mode, with unrecognizable characters causing blanks.

System Request

When the System Request (Alt and Print Screen) key is pressed, a hex 8500 is placed in AX, and an interrupt hex 15 is executed. When the System Request key is released, a hex 8501 is placed in AX, and another interrupt hex 15 is executed. If an application is to use System Request, the following rules must be observed:

Save the previous address.

Overlay interrupt vector hex 15.

Check AH for a value of hex 85:

If yes, process may begin.
If no, go to previous address.

The application program must preserve the value in all registers, except AX, on return. System Request is handled internally in the keyboard routine.

Other Characteristics

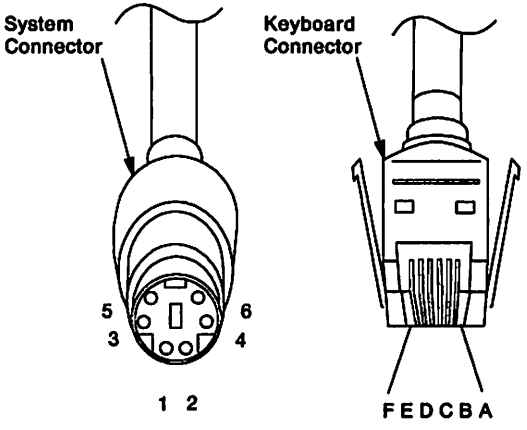
The keyboard routine does its own buffering, and the keyboard buffer is large enough to support entries by a fast typist. However, if a key is pressed when the buffer is full, the key is ignored and the alarm sounds.

The keyboard routine also suppresses the typematic action of the following keys: Ctrl, Shift, Alt, Num Lock, Scroll Lock, Caps Lock, and Ins.

During each interrupt hex 09 from the keyboard, an interrupt hex 15, function (AH) = hex 4F is generated by the BIOS after the scan code is read from the keyboard adapter. The scan code is passed in the AL register with the carry flag set. This allows an operating system to intercept each scan code before it is handled by the interrupt hex 09 routine and change or act on the scan code. If the carry flag is changed to 0 on return from interrupt hex 15, the scan code is ignored by the interrupt handler.

Cables and Connectors

The keyboard cable connects to the system with a 6-pin miniature DIN connector and to the keyboard with a 6-position connector. The following figures show the pin configuration and signal assignments.



DIN Connector Pins	Signal Name	Keyboard Connector Pins
1	+KBD DATA	B
2	Reserved	F
3	Ground	C
4	+5.0 Vdc	E
5	+KBD CLK	D
6	Reserved	A
Shield	Frame Ground	Shield

Figure 22. Keyboard Connectors Signal and Voltage Assignments

Specifications

Specifications for the keyboard are as follows:

Power Requirements

- +5 Vdc ± 10%
- 275 mA.

Size

- Length: 492 millimeters (19.4 inches)
- Depth: 210 millimeters (8.3 inches)
- Height: 58 millimeters (2.3 inches), legs extended.

Weight

- 2.25 kilograms (5.0 pounds).

Characters and Keystrokes

Character Codes 1
Quick Reference 8

Notes:

Character Codes

The following figures show the decimal values, hexadecimal values, and keystrokes for each character. The notes referred to in the figures are on page 7.

Value		As Characters		
Hex	Dec	Symbol	Keystrokes	Notes
00	0	Blank (Null)	Ctrl 2	
01	1	☺	Ctrl A	
02	2	☻	Ctrl B	
03	3	♥	Ctrl C	
04	4	♦	Ctrl D	
05	5	♣	Ctrl E	
06	6	♠	Ctrl F	
07	7	•	Ctrl G	
08	8	◘	Ctrl H, Backspace, Shift Backspace	
09	9	○	Ctrl I	
0A	10	◙	Ctrl J, Ctrl ↵	
0B	11	♂	Ctrl K	
0C	12	♀	Ctrl L	
0D	13	♪	Ctrl M, ↵, Shift ↵	
0E	14	♫	Ctrl N	
0F	15	☼	Ctrl O	
10	16	►	Ctrl P	
11	17	◄	Ctrl Q	
12	18	↕	Ctrl R	
13	19	‼	Ctrl S	
14	20	¶	Ctrl T	
15	21	§	Ctrl U	
16	22	▬	Ctrl V	
17	23	↨	Ctrl W	

Value		As Characters		
Hex	Dec	Symbol	Keystrokes	Notes
18	24	↑	Ctrl X	
19	25	↓	Ctrl Y	
1A	26	→	Ctrl Z	
1B	27	←	Ctrl [Esc, Shift Esc, Crtl Esc	
1C	28	∟	Ctrl \	
1D	29	↔	Ctrl]	
1E	30	▲	Ctrl 6	
1F	31	▼	Ctrl -	
20	32	Blank Space	Space Bar, Shift, Space, Ctrl Space, Alt Space	
21	33	!	!	Shift
22	34	"	"	Shift
23	35	#	#	Shift
24	36	$	$	Shift
25	37	%	%	Shift
26	38	&	&	Shift
27	39	'	'	Shift
28	40	(	(	Shift
29	41	)	)	
2A	42	*	*	Note 1
2B	43	+	+	Shift
2C	44	,	,	
2D	45	-	-	
2E	46	.	.	Note 2

Value		As Characters		
Hex	Dec	Symbol	Keystrokes	Notes
2F	47	/	/	
30	48	0	0	Note 3
31	49	1	1	Note 3
32	50	2	2	Note 3
33	51	3	3	Note 3
34	52	4	4	Note 3
35	53	5	5	Note 3
36	54	6	6	Note 3
37	55	7	7	Note 3
38	56	8	8	Note 3
39	57	9	9	Note 3
3A	58	:	:	Shift
3B	59	;	;	
3C	60	<	<	Shift
3D	61	=	=	
3E	62	>	>	Shift
3F	63	?	?	Shift
40	64	@	@	Shift
41	65	A	A	Note 4
42	66	B	B	Note 4
43	67	C	C	Note 4
44	68	D	D	Note 4
45	69	E	E	Note 4
46	70	F	F	Note 4
47	71	G	G	Note 4
48	72	H	H	Note 4
49	73	I	I	Note 4
4A	74	J	J	Note 4

Value		As Characters		
Hex	Dec	Symbol	Keystrokes	Notes
4B	75	K	K	Note 4
4C	76	L	L	Note 4
4D	77	M	M	Note 4
4E	78	N	N	
4F	79	O	O	Note 4
50	80	P	P	Note 4
51	81	Q	Q	Note 4
52	82	R	R	Note 4
53	83	S	S	Note 4
54	84	T	T	Note 4
55	85	U	U	Note 4
56	86	V	V	Note 4
57	87	W	W	Note 4
58	88	X	X	Note 4
59	89	Y	Y	Note 4
5A	90	Z	Z	Note 4
5B	91	[	[	
5C	92	\	\	Note 4
5D	93	]	]	
5E	94	^	^	Shift
5F	95	_	_	Shift
60	96	`	`	
61	97	a	a	Note 5
62	98	b	b	Note 5
63	99	c	c	Note 5
64	100	d	d	Note 5
65	101	e	e	Note 5
66	102	f	f	Note 5

Value		As Characters		
Hex	Dec	Symbol	Keystrokes	Notes
67	103	g	g	Note 5
68	104	h	h	Note 5
69	105	i	i	Note 5
6A	106	j	j	Note 5
6B	107	k	k	Note 5
6C	108	l	l	Note 5
6D	109	m	m	Note 5
6E	110	n	n	Note 5
6F	111	o	o	Note 5
70	112	p	p	Note 5
71	113	q	q	Note 5
72	114	r	r	Note 5
73	115	s	s	Note 5
74	116	t	t	Note 5
75	117	u	u	Note 5
76	118	v	v	Note 5
77	119	w	w	Note 5
78	120	x	x	Note 5
79	121	y	y	Note 5
7A	122	z	z	Note 5
7B	123	{	{	Shift
7C	124	¦	¦	Shift
7D	125	}	}	Shift
7E	126	~	~	Shift
7F	127	△	Ctrl-	

Value		As Characters		
Hex	Dec	Symbol	Keystrokes	Notes
80	128	Ç	Alt 128	Note 6
81	129	ü	Alt 129	Note 6
82	130	é	Alt 130	Note 6
83	131	â	Alt 131	Note 6
84	132	ä	Alt 132	Note 6
85	133	à	Alt 133	Note 6
86	134	å	Alt 134	Note 6
87	135	ç	Alt 135	Note 6
88	136	ê	Alt 136	Note 6
89	137	ë	Alt 137	Note 6
8A	138	è	Alt 138	Note 6
8B	139	ï	Alt 139	Note 6
8C	140	î	Alt 140	Note 6
8D	141	ì	Alt 141	Note 6
8E	142	Ä	Alt 142	Note 6
8F	143	Å	Alt 143	Note 6
90	144	É	Alt 144	Note 6
91	145	æ	Alt 145	Note 6
92	146	Æ	Alt 146	Note 6
93	147	ô	Alt 147	Note 6
94	148	ö	Alt 148	Note 6
95	149	ò	Alt 149	Note 6
96	150	û	Alt 150	Note 6
97	151	ù	Alt 151	Note 6
98	152	ÿ	Alt 152	Note 6
99	153	Ö	Alt 153	Note 6
9A	154	Ü	Alt 154	Note 6

Value		As Characters		
Hex	Dec	Symbol	Keystrokes	Notes
9B	155	¢	Alt 155	Note 6
9C	156	£	Alt 156	Note 6
9D	157	¥	Alt 157	Note 6
9E	158	Pt	Alt 158	Note 6
9F	159	ƒ	Alt 159	Note 6
A0	160	á	Alt 160	Note 6
A1	161	í	Alt 161	Note 6
A2	162	ó	Alt 162	Note 6
A3	163	ú	Alt 163	Note 6
A4	164	ñ	Alt 164	Note 6
A5	165	Ñ	Alt 165	Note 6
A6	166	ª	Alt 166	Note 6
A7	167	º	Alt 167	Note 6
A8	168	¿	Alt 168	Note 6
A9	169	⌐	Alt 169	Note 6
AA	170	¬	Alt 170	Note 6
AB	171	½	Alt 171	Note 6
AC	172	¼	Alt 172	Note 6
AD	173	¡	Alt 173	Note 6
AE	174	<<	Alt 174	Note 6
AF	175	>>	Alt 175	Note 6
B0	176	⁞	Alt 176	Note 6
B1	177	▒	Alt 177	Note 6
B2	178	▓	Alt 178	Note 6
B3	179	│	Alt 179	Note 6
B4	180	┤	Alt 180	Note 6
B5	181	╡	Alt 181	Note 6
B6	182	╢	Alt 182	Note 6

Value		As Characters		
Hex	Dec	Symbol	Keystrokes	Notes
B7	183	╖	Alt 183	Note 6
B8	184	╕	Alt 184	Note 6
B9	185	╣	Alt 185	Note 6
BA	186	║	Alt 186	Note 6
BB	187	╗	Alt 187	Note 6
BC	188	╝	Alt 188	Note 6
BD	189	╜	Alt 189	Note 6
BE	190	╛	Alt 190	Note 6
BF	191	┐	Alt 191	Note 6
C0	192	└	Alt 192	Note 6
C1	193	┴	Alt 193	Note 6
C2	194	┬	Alt 194	Note 6
C3	195	├	Alt 195	Note 6
C4	196	─	Alt 196	Note 6
C5	197	┼	Alt 197	Note 6
C6	198	╞	Alt 198	Note 6
C7	199	╟	Alt 199	Note 6
C8	200	╚	Alt 200	Note 6
C9	201	╔	Alt 201	Note 6
CA	202	╩	Alt 202	Note 6
CB	203	╦	Alt 203	Note 6
CC	204	╠	Alt 204	Note 6
CD	205	═	Alt 205	Note 6
CE	206	╬	Alt 206	Note 6
CF	207	╧	Alt 207	Note 6
D0	208	╨	Alt 208	Note 6

Value		As Characters		
Hex	Dec	Symbol	Keystrokes	Notes
D1	209		Alt 209	Note 6
D2	210		Alt 210	Note 6
D3	211		Alt 211	Note 6
D4	212		Alt 212	Note 6
D5	213		Alt 213	Note 6
D6	214		Alt 214	Note 6
D7	215		Alt 215	Note 6
D8	216		Alt 216	Note 6
D9	217		Alt 217	Note 6
DA	218		Alt 218	Note 6
DB	219		Alt 219	Note 6
DC	220		Alt 220	Note 6
DD	221		Alt 221	Note 6
DE	222		Alt 222	Note 6
DF	223		Alt 223	Note 6
E0	224	α	Alt 224	Note 6
E1	225	β	Alt 225	Note 6
E2	226	Γ	Alt 226	Note 6
E3	227	π	Alt 227	Note 6
E4	228	Σ	Alt 228	Note 6
E5	229	σ	Alt 229	Note 6
E6	230	μ	Alt 230	Note 6
E7	231	τ	Alt 231	Note 6
E8	232	Φ	Alt 232	Note 6
E9	233	θ	Alt 233	Note 6
EA	234	Ω	Alt 234	Note 6
EB	235	δ	Alt 235	Note 6

Value		As Characters		
Hex	Dec	Symbol	Keystrokes	Notes
EC	236	∞	Alt 236	Note 6
ED	237	φ	Alt 237	Note 6
EE	238	ε	Alt 238	Note 6
EF	239	∩	Alt 239	Note 6
F0	240	≡	Alt 240	Note 6
F1	241	±	Alt 241	Note 6
F2	242	≥	Alt 242	Note 6
F3	243	≤	Alt 243	Note 6
F4	244		Alt 244	Note 6
F5	245		Alt 245	Note 6
F6	246	÷	Alt 246	Note 6
F7	247	≈	Alt 247	Note 6
F8	248	°	Alt 248	Note 6
F9	249	∙	Alt 249	Note 6
FA	250	·	Alt 250	Note 6
FB	251	√	Alt 251	Note 6
FC	252	n	Alt 252	Note 6
FD	253	2	Alt 253	Note 6
FE	254	■	Alt 254	Note 6
FF	255	BLANK	Alt 255	Note 6

Notes:

1. Asterisk (*) can be typed by pressing the * key or, in the shift state, pressing the 8 key.
2. Period (.) can be typed by pressing the . key or, in the shift or Num Lock state, pressing the Del key.
3. Numeric characters 0-9 can be typed by pressing the numeric keys on the keyboard or, in the shift or Num Lock state, pressing the numeric keys in the keypad portion of the keyboard.
4. Uppercase alphabetic characters (A-Z) can be typed by pressing the character key in the shift state or the Caps Lock state.
5. Lowercase alphabetic characters (a-z) can be typed by pressing the character key in the normal state or in Caps Lock and shift state combined.
6. The three digits are typed on the numeric keypad while pressing the Alt key. Character codes 001-255 may be entered in this fashion (with Caps Lock activated, character codes 97-122 display in uppercase).

Quick Reference

DECIMAL VALUE	➡	0	16	32	48	64	80	96	112
⬇	HEXA-DECIMAL VALUE	0	1	2	3	4	5	6	7
0	0	BLANK (NULL)	►	BLANK (SPACE)	0	@	P	‘	p
1	1	☺	◄	!	1	A	Q	a	q
2	2	☻	↕	"	2	B	R	b	r
3	3	♥	!!	#	3	C	S	c	s
4	4	♦	¶	$	4	D	T	d	t
5	5	♣	§	%	5	E	U	e	u
6	6	♠	▬	&	6	F	V	f	v
7	7	•	↨	′	7	G	W	g	w
8	8	◘	↑	(	8	H	X	h	x
9	9	○	↓	)	9	I	Y	i	y
10	A	◙	→	*	:	J	Z	j	z
11	B	♂	←	+	;	K	[	k	{
12	C	♀	∟	,	<	L	\	l	¦
13	D	♪	↔	—	=	M	]	m	}
14	E	♫	▲	.	>	N	^	n	~
15	F	☼	▼	/	?	O	_	o	△

DECIMAL VALUE	➡	128	144	160	176	192	208	224	240
⬇	HEXA-DECIMAL VALUE	8	9	A	B	C	D	E	F
0	0	Ç	É	á	▒	└	╨	∝	≡
1	1	ü	æ	í	▒	┴	╤	β	±
2	2	é	Æ	ó	▒	┬	╘	Γ	≥
3	3	â	Ô	ú	│	├	╙	π	≤
4	4	ä	ö	ñ	┤	─	╒	Σ	⌠
5	5	à	ò	Ñ	╡	┼	╓	σ	⌡
6	6	å	û	ª	╢	╞	╫	μ	÷
7	7	ç	ù	º	╖	╟	╪	Υ	≈
8	8	ê	ÿ	¿	╕	╚	┘	Φ	°
9	9	ë	Ö	⌐	╣	╔	┌	Θ	•
10	A	è	Ü	¬	║	╩	╥	Ω	•
11	B	ï	¢	½	╗	╦	╙	δ	√
12	C	î	£	¼	╝	╠	╘	∞	n
13	D	ì	¥	¡	╜	═	╒	φ	2
14	E	Ä	₧	«	╛	╬	╓	∈	■
15	F	Å	ƒ	»	┐	╧	╫	∩	BLANK 'FF'

Notes:

Power Supply (Types 1 and 2)

Description 1
Input Protection 1
  Ground Leakage Current 1
Outputs 2
  Output Protection 2
  Voltage Sequencing 2
  No-Load Operation 2
Power Supply Connector (Type 1) 3
Power Supply Connectors (Type 2) 4

Figures

1. Power Supply Connector (Type 1) 3
2. Power Supply Connector Voltages and Signal Assignments . 3
3. System Board Power Supply Cable Connector (Type 2) 4
4. System Board Power Supply Connector Voltage and Signal Assignments 4
5. Fixed Disk Drive Power Supply Cable Connectors (Type 2) . 5
6. Fixed Disk Drive Power Supply Connectors Voltage and Signal Assignments 5

Description

The power supply converts the ac input voltage to three dc outputs. A description of the input voltage ranges to the power supply is in the system-specific technical references. The power supply provides power for the following:

- System board
- Channel adapters
- Diskette drives
- Fixed disk drives
- Auxiliary device
- Keyboard.

The power switch and two light-emitting diodes (LEDs) are on the front of the system unit. The green LED indicates that the power supply is operating. The yellow LED indicates fixed disk drive activity.

Input Protection

The input power line is protected against an overcurrent condition by an internal fuse.

Ground Leakage Current

The system unit ground leakage current does not exceed 500 microamperes at any nominal input voltage.

Outputs

The power supply provides three voltages: +5, +12, and −12 Vdc.

Output Protection

A short circuit placed on any dc output (between outputs or between an output and dc return) latches all dc outputs into a shutdown state with no damage to the power supply.

If an overvoltage fault occurs (internal to the power supply), the power supply latches all dc outputs into a shutdown state before any output exceeds 130% of its nominal value.

If either of these shutdown states is actuated, the power supply returns to normal operation only after the fault has been removed and the power switch has been turned off for at least ten seconds.

Voltage Sequencing

At power-on time, the output voltages track within 50 milliseconds of each other when measured at the 50% points.

No-Load Operation

The power supply is capable of operation with "no load" on the outputs. The output regulation is ± 25% and the 'power-good' signal can be either active or inactive.

Power Supply Connector (Type 1)

The Type 1 power supply uses a 50-pin card-edge connector mounted on the side of the power supply. The system board plugs into the card-edge connector, eliminating the need for separate cabling.

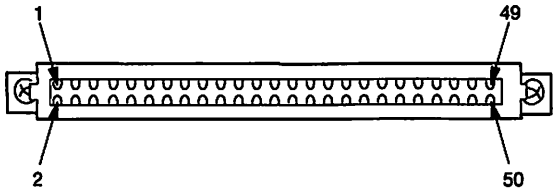


Figure 1. Power Supply Connector (Type 1)

Pin	I/O	Signal	Pin	I/O	Signal
1	N/A	−12 Vdc	2	N/A	Signal Ground
3	N/A	+12 Vdc	4	N/A	Signal Ground
5	N/A	+12 Vdc	6	N/A	Signal Ground
7	N/A	+12 Vdc	8	N/A	Signal Ground
9	N/A	+12 Vdc	10	N/A	Signal Ground
11	N/A	+12 Vdc	12	N/A	Signal Ground
13	N/A	+12 Vdc	14	N/A	Signal Ground
15	N/A	+5 Vdc	16	N/A	Signal Ground
17	N/A	+5 Vdc	18	N/A	Signal Ground
19	N/A	+5 Vdc	20	N/A	Signal Ground
21	N/A	+5 Vdc	22	N/A	Signal Ground
23	N/A	+5 Vdc	24	N/A	Signal Ground
25	N/A	+5 Vdc	26	N/A	Signal Ground
27	N/A	+5 Vdc	28	N/A	Signal Ground
29	N/A	+5 Vdc	30	N/A	Signal Ground
31	N/A	+5 Vdc	32	N/A	Signal Ground
33	N/A	+5 Vdc	34	N/A	Signal Ground
35	N/A	+5 Vdc	36	N/A	Signal Ground
37	N/A	+5 Vdc	38	N/A	Signal Ground
39	N/A	+5 Vdc	40	N/A	Signal Ground
41	N/A	+5 Vdc	42	N/A	Signal Ground
43	N/A	+5 Vdc	44	N/A	Signal Ground
45	N/A	+5 Vdc	46	N/A	Signal Ground
47	N/A	+5 Vdc	48	N/A	Signal Ground
49	I	System Status	50	O	Power Good

Figure 2. Power Supply Connector Voltages and Signal Assignments

Power Supply Connectors (Type 2)

The Type 2 power supply uses a 15-pin connector for the system board and two 4-pin connectors for the two fixed-disk-drive locations.

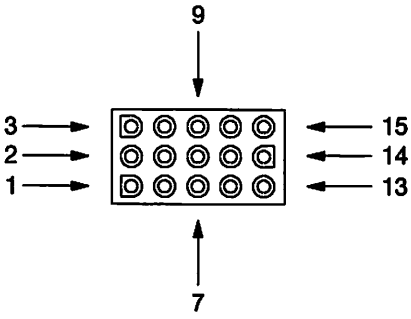


Figure 3. System Board Power Supply Cable Connector (Type 2)

Pin	I/O	Signal	Pin	I/O	Signal
1	N/A	+5 Vdc	2	N/A	Signal Ground
3	N/A	+12 Vdc	4	N/A	+5 Vdc
5	N/A	Signal Ground	6	N/A	Signal Ground
7	N/A	+5 Vdc	8	N/A	Signal Ground
9	N/A	−12 Vdc	10	N/A	+5 Vdc
11	N/A	Signal Ground	12	O	Power Good
13	N/A	+5 Vdc	14	N/A	Signal Ground
15	I	System Status			

Figure 4. System Board Power Supply Connector Voltage and Signal Assignments



Figure 5. Fixed Disk Drive Power Supply Cable Connectors (Type 2)

Pin	I/O	Signal	Pin	I/O	Signal
1	N/A	+12 Vdc	2	N/A	Signal Ground
3	N/A	Signal Ground	4	N/A	+5 Vdc

Figure 6. Fixed Disk Drive Power Supply Connectors Voltage and Signal Assignments

Notes:

Compatibility

Description 1
Application Guidelines 1
Hardware Interrupts 1
Software Interrupts 3
High-Level Language Considerations 3
Assembler Language Programming Considerations 3
  Opcodes 4
  80286 Anomalies 8
  80386 Anomalies 8
  ROM BIOS and Operating System Function Calls 15
Software Compatibility 18
Multitasking Provisions 19
  Interfaces 19
  Classes 20
  Time-Outs 21
Machine-Sensitive Programs 22
Math Coprocessor Compatibility 22
  80287 to 80387 Compatibility 23
  Exceptions 24
  8087 to 80287 Compatibility 25
Diskette Drives and Controller 27
Fixed Disk Drives and Controller 28

Index 29

Figures

1. String Instruction/Register Size Mismatch 11
2. Write and Format Head Settle Time 17
3. Functional Code Assignments 21
4. Math Coprocessor Software Compatibility 23
5. Diskette Drive Read, Write, and Format Capabilities 27

Description

The differences in system microprocessors, math coprocessors, general system architecture, and diskette drive capabilities must be taken into consideration when designing application programs exclusively for a specific model or programs compatible across the IBM Personal Computer and Personal System/2 product lines. This section discusses these major differences and provides some suggestions to aid you in developing your program.

Application Guidelines

Use the following information to develop application programs for the IBM Personal Computer and Personal System/2 products. Whenever possible, BIOS should be used as an interface to hardware in order to provide maximum compatibility and portability of applications across systems.

Hardware Interrupts

Hardware interrupts are level-sensitive for systems using the Micro Channel architecture while systems using the Personal Computer type I/O channel design have edge-triggered hardware interrupts. The interrupt controller clears its in-service register bit when the interrupt routine sends an End-of-Interrupt (EOI) command to the controller. The EOI command is sent whether the incoming interrupt request to the controller is active or inactive.

In level-sensitive systems, the interrupt-in-progress latch is readable at an I/O address bit position. This latch is read during the interrupt service routine and may be reset by the read operation or may require an explicit reset.

Note: Designers may want to limit the number of devices sharing an interrupt level for performance and latency considerations.

The interrupt controller on level-sensitive systems requires the interrupt request to be inactive at the time the EOI command is sent; otherwise, a "new" interrupt request will be detected and another microprocessor interrupt caused.

To avoid this problem, a level-sensitive interrupt handler must clear the interrupt condition (usually by a Read or Write to an I/O port on the device causing the interrupt). After clearing the interrupt condition, a JMP $+2 should be executed prior to sending the EOI command to the interrupt controller. This ensures that the interrupt request is removed prior to re-enabling the interrupt controller. Another JMP $+2 should be executed after sending the EOI command, but prior to enabling the interrupt through the Set Interrupt Enable Flag (STI) command.

In the level-sensitive systems, hardware prevents the interrupt controllers from being set to the edge-triggered mode.

Hardware interrupt IRQ9 is defined as the replacement interrupt level for the cascade level IRQ2. Program interrupt sharing should be implemented on IRQ2, interrupt hex 0A. The following processing occurs to maintain compatibility with the IRQ2 used by IBM Personal Computer products:

1. A device drives the interrupt request active on IRQ2 of the channel.
2. This interrupt request is mapped in hardware to IRQ9 input on the second interrupt controller.
3. When the interrupt occurs, the system microprocessor passes control to the IRQ9 (interrupt hex 71) interrupt handler.
4. This interrupt handler performs an EOI command to the second interrupt controller and passes control to IRQ2 (interrupt hex 0A) interrupt handler.
5. This IRQ2 interrupt handler, when handling the interrupt, causes the device to reset the interrupt request prior to performing an EOI command to the master interrupt controller that finishes servicing the IRQ2 request.

Software Interrupts

With the advent of software interrupt sharing, software interrupt routines must daisy chain interrupts. Each routine must check the function value and if it is not in the range of function calls for that routine, it must transfer control to the next routine in the chain. Because software interrupts are initially pointed to 0:0 before daisy chaining, it is necessary to check for this case. If the next routine is pointed to 0:0 and the function call is out of range, the appropriate action is to set the carry flag and do a RET 2 to indicate an error condition.

High-Level Language Considerations

The IBM-supported languages of IBM C, BASIC, FORTRAN, COBOL, and Pascal are the best choices for writing compatible programs.

If a program uses specific features of the hardware, that program may not be compatible with all IBM Personal Computer and Personal System/2 products. Specifically, the use of assembler language subroutines or hardware-specific commands (for example In, Out, Peek, and Poke) must follow the assembler language rules. See "Assembler Language Programming Considerations."

Any program that requires precise timing information should obtain it through an operating system or language interface; for example, TIME$ in BASIC. If greater precision is required, the assembler techniques in "Assembler Language Programming Considerations" are available. The use of programming loops may prevent a program from being compatible with other IBM Personal Computer products, IBM Personal System/2 products, and software.

Assembler Language Programming Considerations

This section describes fundamental differences between the systems in the Personal Computer and Personal System/2 product lines that may affect program development.

Opcodes

The following opcodes work differently on systems using either the 80286 or 80386 microprocessor than they do on systems using the 8088 or 8086 microprocessor.

- PUSH SP

  The 80286 and 80386 microprocessors push the current stack pointer; the 8088 and 8086 microprocessors push the new stack pointer, that is, the value of the stack pointer after the PUSH SP instruction is completed.

- Single step interrupt (when TF = 1) on the interrupt instruction (Opcode hex CC, CD):

  The 80286 and 80386 microprocessors do not perform a single-step interrupt on the INT instruction; the 8088 and 8086 microprocessors do perform a single-step interrupt on the INT instruction.

- The divide error exception (interrupt 0):

  The 80286 and 80386 microprocessors push the CS:IP of the instruction that caused the exception; the 8088 and 8086 microprocessors push the CS:IP of the instruction following the instruction that caused the exception.

- Shift counts for the 80286 and 80386 microprocessors:

  Shift counts are masked to 5 bits. Shift counts greater than 31 are treated mod 32. For example, a shift count of 36 shifts the operand four places.

- LOCK prefix:

  When the LOCK prefix is used with an instruction, the system microprocessor executes the entire instruction before allowing interrupts. If a Repeat String Move instruction is locked, interrupts may be disabled for a long duration.

  The 8088, 8086, and 80286 microprocessors allow the LOCK prefix to be used with most instructions. However, the 80386 microprocessor restricts the use of LOCK to the following instructions:

  - Bit Test and Set Memory, Register/Immediate
  - Bit Test and Reset Memory, Register/Immediate

- Bit Test and Complement Memory, Register/Immediate
- XCHG Register, Memory
- XCHG Memory, Register
- ADD, OR, ADC, SBB, Memory, Register/Immediate
- AND, SUB, XOR Memory, Register/Immediate
- NOT, NEG, INC, DEC Memory.

An undefined opcode trap (INT 6) is generated if the LOCK prefix is used in the 80386 environment with an instruction not listed.

When the 80286 is operating in the virtual memory mode, the LOCK prefix is IOPL-sensitive. Since the 80386 restricts the use of the LOCK prefix to a specific set of instructions, the LOCK prefix is not IOPL-sensitive in the 80386 environment.

- Multiple lockout instructions:

  There are several microprocessor instructions that, when executed, lock out external bus signals. DMA requests are not honored during the execution of these instructions. Consecutive instructions of this type prevent DMA activity from the start of the first instruction to the end of the last instruction. To allow for necessary DMA cycles, as required by the diskette controller in a multitasking system, multiple lock-out instructions must be separated by a JMP SHORT $+2.

- Back-to-back I/O commands:

  Back-to-back I/O commands to the same I/O ports do not permit enough recovery time for some I/O adapters. To ensure enough time, a JMP SHORT $+2 must be inserted between IN/OUT instructions to the same I/O adapters.

  Note: MOV AL,AH type instruction does not allow enough recovery time. An example of the correct procedure follows:

```
OUT  IO_ADD,AL
JMP  SHORT $+2
MOV  AL,AH
OUT  IO_ADD,AL
```

- I/O commands followed by an STI instruction:

  I/O commands followed immediately by an STI instruction do not permit enough recovery time for some system board and channel

operations. To ensure enough time, a JMP SHORT $+2 must be inserted between the I/O command and the STI instruction.

Note: MOV AL,AH type instruction does not allow enough recovery time. An example of the correct procedure follows:

```
OUT  IO_ADD,AL
JMP  SHORT $+2
MOV  AL,AH
STI
```

- NT bit and IOPL bits:

  When the 80286 is operating in the Real Address mode, the NT and IOPL bits in the flag register cannot be changed; the bits are zero.

  The 80386 allows the NT bit and the IOPL bits to be modified by POP stack into flags, and other instructions, while operating in the Real Address mode. This has no effect on the Real Address mode operation. However, upon entering Protected Mode operation, the NT bit should be cleared to prevent erroneous execution of the IRET instruction. If NT is set, the IRET attempts to perform a task switch to the previous task.

- Overlap of OUT and following instructions:

  The 80386 has a delayed write to memory and delayed output-to-I/O capability. It is possible for the actual output cycle to I/O devices to occur after the completion of instructions following the Out instruction. Under certain conditions, this may cause some programs to behave in an undesirable manner. For example, an interrupt handler routine may output an EOI command to the interrupt controller to drop the interrupt request. If the interrupt handler has an STI instruction following the output instruction, the 80386 may re-enable interrupts before the interrupt controller drops the interrupt request. This could cause the interrupt routine to be reentered.

  To avoid this problem, either of the following procedures may be used:

  – Place a JMP SHORT $+2 instruction between the OUT instruction and the STI instruction, or

- Read back the status from the interrupt controller before executing the STI instruction.

- Math coprocessor instructions:

  In 80386-based systems, the mode of the microprocessor and math coprocessor are tightly coupled. This is not the case for 80286-based systems. The 80286-based systems require the math coprocessor FSETPM instruction to be executed to enable the 80287 to operate in the Protected mode. The 80287 remains in the Protected mode until it is reset.

  The mode of the 80287 determines the format in which the math coprocessor state information is saved by the FSTENV and FSAVE instructions. In the Protected mode, the instruction and data operand pointers are saved as selector/offset pairs; in the Real Address mode, the physical address and opcode are saved.

  If the FSETPM instruction is encountered in the 80386 environment, it is ignored. The formatting is performed by the 80386, which internally maintains the instruction and data operand pointers. The Real Address mode format image is saved when the 80386 is operating in the Real Address mode or Virtual 8086 mode. The Protected mode format is used otherwise.

  See also "Math Coprocessor Compatibility" on page 22 for more information.

- Use of 32-bit registers and the 32-bit addressing mode:

  It is possible to use the 32-bit registers and 32-bit addressing mode in all operating modes of the 80386 through the use of the operand-size prefix or address-size prefix.

  In a multitasking environment, extreme care must be taken to avoid conflicts with other tasks that use extended registers. If the operating system saves the extended 32-bit registers and new segment registers in the task context save area, conflicts will be avoided; if the operating system does not provide this function, another method must be implemented.

  One possible method is to disable the interrupts while using the extended registers. The extended registers should be saved before use and restored immediately after use while the interrupts are still disabled. The time that interrupts are disabled should be kept as short as possible.

- Operand Alignment:

  When multiple bus cycles are required to transfer a multibyte logical operand (for example, a word operand beginning at an address not evenly divisible by 2), the 80386 transfers the highest order bytes first.

  This characteristic may affect adapters with memory-mapped I/O that require or assume that sequential memory accesses are made to the memory I/O ports.

  This problem may be avoided by using a REP MOVB(yte) instead of a REP MOVSW(ord).

80286 Anomalies

In the Protected mode, when any of the null selector values (hex 0000, 0001, 0002, and 0003) are loaded into the DS or ES registers with a MOV or POP instruction or a task switch, the 80286 always loads the null selector hex 0000 into the corresponding register.

If a coprocessor (80287) operand is read from an "executable and readable" and conforming (ERC) code segment, and the coprocessor operand is sufficiently near the segment limit that the second or subsequent byte lies outside the limit, an interrupt 9 will not be generated.

The following describes the operation of all 80286 parts:

- Instructions longer than 10 bytes (instructions using multiple redundant prefixes) generate an interrupt 13 (General Purpose Exception) in both the Real Address mode and Protected mode.
- If the second operand of an ARPL instruction is a null selector, the instruction generates an interrupt 13.

80386 Anomalies

The following describes anomalies that apply to the B-1 stepping level of the 80386 microprocessor.

80386 Real Address Mode Operation

- FSAVE/FSTENV opcode field incorrect:

The opcode of some numeric instructions is saved incorrectly in the FSAVE/FSTENV format image when the 80386 is operating in the Real Address mode or Virtual 8086 mode.

The power-on self-test (POST) code in the system ROM enables hardware interrupt 13 and sets up its vector (INT hex 75) to point to a math coprocessor exception routine in ROM. Any time this routine is executed as a result of an exception, it repairs the opcode field by performing the following sequence:

1. Clears the 'busy' signal latch
2. Executes FNSTENV (save image on stack)
3. Extracts instruction pointer from FSTENV memory image
4. Skips over prefix bytes until opcode is found
5. Inserts correct opcode information in the memory image
6. Executes FLDENV to restore the corrected opcode field
7. Writes the EOI command to the interrupt controller
8. Transfers control to the address pointed to by the NMI handler.

Any math coprocessor application containing an NMI handler should require its NMI handler to read the status of the coprocessor to determine if the NMI was caused by the coprocessor. If the interrupt was not generated by the coprocessor, control should be passed to the original NMI handler.

Applications do not require any modification for this errata because the BIOS exception routine repairs the opcode field after exceptions. However, if a debugger is used to display the math coprocessor state information, the opcode field will contain an incorrect value for some math coprocessor instructions.

- Single stepping repeated MOVS:

  If a repeated MOVS instruction is executed when single-stepping is turned on (TF = 1 in the EFLAGS register), a single-step interrupt is taken after two move steps on the 80386 microprocessor. The 8088, 8086, and 80286 microprocessors take a single-step interrupt after every iteration step. However, for the 80386, if a data breakpoint is encountered on the first iteration of a repeated MOVS, the data break is not taken until after the second iteration. Data

breakpoints encountered on the second and subsequent iterations stop immediately after the step causing a break.

- Wrong register size for string instructions:

  One of the (E)CX, (E)SI, or (E)DI registers will not be updated properly if certain string and loop instructions are followed by instructions that either:

  - Use a different address size (that is, either the string instruction or the following instruction uses an address size prefix), or
  - Reference the stack (such as PUSH/POP/CALL/RET) and the "B" bit in the SS descriptor is different from the address size used by the instructions.

  The size of the register (16 bits or 32 bits) is taken from the instruction following the string instruction rather than from the string instruction itself. This could result in one of the following conditions:

  - Only the lower 16 bits of a 32-bit instruction updated (if the 32-bit string instruction was followed by an instruction using a memory operand addressed with a 16-bit address).
  - All 32 bits of a register updated rather than just the lower 16 bits.

  The following is a list of the instructions and the affected registers:

Instruction	Register
REP MOVS	(E)SI
MOVS	(E)DI
STOS	(E)DI
INS	(E)DI
REP INS	(E)CX

Notes:

1. A 32-bit effective address size specified with a string instruction indicates that the 32-bit ESI and EDI registers should be used for forming addresses, and the 32-bit ECX register should be used as the count register.
2. A 32-bit operand size on a repeated string move (MOVS) should be used only if the compiler or programmer can guarantee that the strings do not overlap destructively. An 8-bit or 16-bit MOVS has a predictable effect when the strings overlap destructively.

Figure 1. String Instruction/Register Size Mismatch

The problem only occurs if instructions with different address sizes are mixed, or if a code segment of one size is used with a stack segment of the other size.

To avoid this problem, add a NOP instruction after each of the instructions listed in Figure 1 on page 10 and ensure that the NOP instruction has the same address size as the string/loop instruction. If necessary, an address size prefix hex 67 may precede the NOP instruction.

- Wrong ECX update with REP INS:

  ECX (or CX in a 16-bit address size) is not updated correctly in the case of a REP INS[1] followed by an early start instruction[2]. After executing any repeat-prefixed instruction, the contents of ECX is supposed to be 0, but in the case of an REP INS instruction, ECX is not updated correctly and its contents become hex FFFFFFFF for 32-bit address size operations and hex 0FFFF for 16-bit address size operations. INS is still executed the correct number of times and EDI is updated properly.

  To avoid this problem, one of the following procedures may be used:

  - Insert an explicit MOV ECX,0 (or MOV CX,0) instruction after any REP INS instruction. This ensures that the contents of ECX or CX are 0.
  - Do not rely on the count in ECX (or CX) after a REP INS instruction but instead, move a new count into ECX (or CX) before using it again.

- Test register access fails:

  Accessing the Translate Lookaside Buffer (TLB) test registers, TR6 and TR7, may not function properly.

  Avoid using test registers TR6 and TR7 to test the TLB.

---

1 REP INS refers to any input string instruction with a repeat prefix.

2 An early start instruction refers to PUSH, POP, or memory reference instructions.

- Math coprocessor Save/Restore environment operands:

  If either of the last two bytes of an FSAVE/FRSTOR or FSTENV/FLDENV is not accessible, the instruction cannot be restarted. An FNINIT instruction must be issued to the math coprocessor before any other math coprocessor instruction can be executed. This problem arises only in demand-paged systems, or demand-segmented systems that increase the segment size on demand.

- Wrap-around math coprocessor operands:

  The 80386 architecture does not permit a math coprocessor operand, or any other operand, to wrap around the end of a segment. If such an instruction is issued in a protected segmented system, and the operand starts and ends in valid parts of a segment, but passes through an inaccessible region of the segment, the math coprocessor may be put in an indeterminate state. Under these conditions, an FCLEX or FINIT must be sent before any other math coprocessor instruction is issued.

- Load Segment Limit instruction cannot precede PUSH/POP:

  If the instruction executed immediately after a Load Segment Limit (LSL) instruction does a stack operation, the value of (E)SP may be incorrect after the operation.

  Note: Stack operations resulting from noninstruction sources, such as exceptions or interrupts following the LSL, do not corrupt (E)SP.

  To avoid this problem, make sure that the instruction following an LSL instruction is never one that does a push to or pop off the stack. This includes PUSH, POP, RET, CALL, ENTER, and other such instructions. This can be achieved by always following an LSL instruction with a NOP instruction. Even if a forbidden instruction is used, (E)SP may be updated correctly since the problem is data-dependent and only occurs if the LSL operation succeeds (that is, sets the ZF flag).

- LSL/LAR/VERR/VERW malfunction with a NULL selector:

  An LSL, LAR, VERR, or VERW executed with a NULL selector (that is, bits 15 through 2 of the selector set to 0) operates on the descriptor at entry 0 of the Global Descriptor Table (GDT) instead of unconditionally clearing the ZF flag.

  This problem can be avoided by filling in the "NULL descriptor" (that is, the descriptor at entry 0 of the GDT) with all zeroes, which is an invalid descriptor type.

  The access to the "NULL descriptor" is made but fails since the descriptor has an invalid type. The failure is reported with ZF cleared, which is the desired behavior.

80386 Extended Protected Mode Operation

The following problems exist for operation in the Virtual 8086 mode.

- Task switch to Virtual 8086 mode does not set prefetch limit:

  The 80386 prefetch unit limit is not updated when doing a task switch to the Virtual 8086 mode. This may cause an incorrect segment limit violation to be reported if the microprocessor instruction fetches the segment limit that existed before the task switch.

  This problem can be avoided by using an IRET with the appropriate items on the stack to start the Virtual 8086 task in place of the task switch method.

- FAR jump near page boundary in Virtual 8086 mode:

  When paging is enabled in the Virtual 8086 mode, and a direct FAR jump (opcode EA) instruction is located at the end of a page (or within 16 bytes of the end), and the next page is not cached in the TLB internal to the 80386, the FAR jump instruction leaves the prefetcher limit at the "end" of the old code segment instead of setting it at the "end" of the new code segment. This can allow execution off the end of the new segment to trigger a segment limit violation, or cause a spurious GP fault if the old and new segments overlap and a prefetch crosses the old segment limit.

  There is no way to detect code "walking off" the end of a code segment. However, the spurious GP fault can be avoided by simply performing an IRET back to the instruction

causing the fault. The IRET will set the prefetch limit correctly, provided the exception handler has the ability to determine a spurious GP fault from a "real" GP fault.

The following problems exist for operations with paging enabled.

- Coprocessor operands:

  To avoid having a nonstartable instruction involving math coprocessor operands in demand-paged systems, ensure the operands do not cross page boundaries. This can be accomplished by aligning math coprocessor operands in 128-byte boundaries within a segment, and aligning the start of segments on 128-byte physical boundaries.

- Page fault error code on stack is not reliable:

  When a page fault (exception 14) occurs, the three defined bits in the error code can be unreliable if a certain sequence of prefetches occurred at the same time.

  Although the page-fault error code pushed onto the page-fault handler stack is sometimes unreliable, the page-fault linear address stored in register CR2 is always correct. The page-fault handler should refer to the page-fault linear address in register CR2 to access the corresponding page table entry and thereby determine whether the page fault was due to a page-not-present condition or a usage violation.

- I/O relocated in paged systems:

  When paging is enabled (PG = 1 in CR0), accessing I/O addresses in the range hex 00001000 to hex 0000FFFF, or accessing a 80387 math coprocessor using ESC instructions (I/O addresses hex 800000F8 to hex 800000FF) can generate incorrect I/O addresses on A12 through A31 if the I/O address is the same as a memory linear address that is mapped by the TLB.

  The physical address corresponding to the memory linear address mapped by the TLB is ANDed with the I/O address, causing the I/O address to be incorrect in most cases.

  A suggested method for handling normal I/O addresses between hex 00000 and hex 0FFFF is as follows: The operating system is required to map the lowest (first) 64KB of linear address space to 16 pages, which are defined such that bits 12 through 15 of the linear and physical addresses are

equal. This requires that the pages be aligned on a 64KB physical boundary (the physical address associated with the first page has address bits 15 through 0 equal to 0).

A suggested method for handling the math coprocessor I/O addresses requires that the memory page at linear address hex 80000000 always be marked "not present" so it cannot be cached in the TLB. This may be accomplished in one of the following ways:

- Require the operating system to handle a 4KB "hole" in the linear address space at the 2GB boundary.
- Restrict the linear address space to a 2GB maximum instead of 4GB. No segments will have a linear address above the 2GB boundary.

- Spurious page level protection fault:

  This problem only occurs when the page table and the directory entries that map the stacks for the inner levels of a task are marked as supervisor access only, and an external bus HOLD comes during the cycle that pops (E)SP off the stack during an inter-level RET or IRET.

  This problem can be avoided by marking the pages that map the inner level stacks (level 0, 1, and 2) to permit the user read access. The segmentation protection mechanism can be used to prevent user access to the linear addresses containing these stacks, if required.

ROM BIOS and Operating System Function Calls

For maximum portability, programs should perform all I/O operations through operating system function calls. In environments where the operating system does not provide the necessary programming interfaces, programs should access the hardware through ROM BIOS function calls, if permissible.

- In some environments, program interrupts are used for access to these functions. This practice removes the absolute addressing from the program. Only the interrupt number is required.
- The coprocessor detects six different exception conditions that can occur during instruction execution. If the appropriate exception mask within the coprocessor is not set, the coprocessor

sets the 'error' signal. This 'error' signal generates a hardware interrupt 13 (IRQ 13) causing the 'busy' signal to be held in the busy state. The 'busy' signal can be cleared by an 8-bit I/O Write command to address hex 00F0 with bits D0 through D7 equal to 0.

The power-on self-test code in the system ROM enables hardware IRQ 13 and sets up its vector to point to a routine in ROM. The ROM routine clears the 'busy' signal latch and then transfers control to the address pointed to by the NMI vector. This maintains code compatibility across the IBM Personal Computer and Personal System/2 product lines. The NMI handler reads the status of the coprocessor to determine if the NMI was caused by the coprocessor. If the interrupt was not caused by the coprocessor, control is passed to the original NMI handler.

- In systems using the 80286 or 80386 microprocessor, IRQ 9 is redirected to INT hex 0A (hardware IRQ 2). This ensures that hardware designed to use IRQ 2 will operate in these systems. See "Hardware Interrupts" on page 1 for more information.

- The system can mask hardware sensitivity. New devices can change the ROM BIOS to accept the same programming interface on the new device.

- In cases where BIOS provides parameter tables, such as for video or diskette, a program can substitute new parameter values by building a new copy of the table and changing the vector to point to that table. However, the program should copy the current table, using the current vector, and then modify those locations in the table that need to be changed. In this way, the program does not inadvertently change any values that should be left the same.

- The Diskette Parameters table pointed to by INT hex 1E consists of 11 parameters required for diskette operation. It is recommended that the values supplied in ROM be used. If it becomes necessary to modify any of the parameters, build another parameter block and modify the address at INT hex 1E (0:78) to point to the new block.

  The parameters were established to allow:

  - Some models of the IBM Personal Computer to operate both the 5.25-inch high capacity diskette drive (96 tracks per inch) and the 5.25-inch double-sided diskette drive (48 tracks per inch).

- Some models of the Personal System/2 to operate both the 3.5-inch 1.44M diskette drive and the 3.5-inch 720KB diskette drive.

The Gap Length Parameter is not always retrieved from the parameter block. The gap length used during diskette read, write, and verify operations is derived from within diskette BIOS. The gap length for format operations is still obtained from the parameter block.

Note: Special considerations are required for format operations. Refer to the diskette section of the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference* for the required details.

If a parameter block contains a head settle time parameter value of 0 milliseconds, and a write or format operation is being performed, the following minimum head settle times are enforced.

Drive Type	Head Settle Time
5.25-Inch Diskette Drives:	
Double Sided (48 TPI)	20 ms
High Capacity (96 TPI)	15 ms
3.5-Inch Diskette Drives:	
720K	20 ms
1.44M	15 ms

Figure 2. Write and Format Head Settle Time

Read and verify operations use the head settle time provided by the parameter block.

If a parameter block contains a motor-start wait parameter of less than 500 milliseconds (1 second for a Personal Computer product) for a write or format operation, diskette BIOS enforces a minimum time of 500 milliseconds (1 second for a Personal Computer product). Verify and write operations use the motor-start time provided by the parameter block.

- Programs may be designed to reside on both 5.25-inch or 3.5-inch diskettes. Since not all programs are operating-system dependent, the following procedure can be used to determine the type of media inserted into a diskette drive:

1. Verify Track 0, Head 0, Sector 1 (1 sector): This allows diskette BIOS to determine if the format of the media is a recognizable type.

   If the verify operation fails, issue the reset function (AH=0) to diskette BIOS and try the operation again. If another failure occurs, the media needs to be formatted or is defective.

2. Verify Track 0, Head 0, Sector 16 (1 sector).

   If the verify operation fails, either a 5.25-inch (48 TPI) or 3.5-inch 720KB diskette is installed. The type can be determined by verifying Track 78, Head 1, Sector 1 (1 sector). A successful verification of Track 78 indicates a 3.5-inch 720KB diskette is installed; a verification failure indicates a 5.25-inch (48 TPI) diskette is installed.

   Note: Refer to the *DOS Technical Reference* for the File Allocation Table parameters for single-sided and double-sided diskettes.

3. Read the diskette controller status in BIOS starting with address 40:42. The fifth byte defines the head that the operation ended with. If the operation ended with head 1, the diskette is a 5.25-inch high capacity (96 TPI) diskette; if the operation ended with head 0, the diskette is a 3.5-inch 1.44M diskette.

Software Compatibility

To maintain software compatibility, the interrupt polling mechanism used by IBM personal computer products is retained. Software that interfaces with the reset port for the IBM personal computer positive-edge interrupt sharing[3] does not create interference. Level-sensitive interrupt hardware allows several devices to simultaneously set a common interrupt line active (low) without interference.

Application code that deals directly with the interrupt controller may try to reset the controller to the positive edge-sensitive mode when

[3] Hex address 02FX or 06FX, where X is the interrupt level.

relinquishing control. The interrupt control circuitry of the system board prevents setting the controller to the edge-sensitive mode by blocking positive edge-sensitive commands to the interrupt controllers.

Multitasking Provisions

The BIOS contains a feature to assist multitasking implementation. "Hooks" are provided for a multitasking dispatcher. Whenever a busy (wait) loop occurs in the BIOS, a hook is provided for the program to break out of the loop. Also, whenever BIOS services an interrupt, a corresponding wait loop is exited, and another hook is provided. Thus a program can be written that employs the bulk of the device driver code. The following is valid only in the Real Address mode and must be taken by the code to allow this support.

- The program is responsible for the serialization of access to the device driver. The BIOS code is not reentrant.
- The program is responsible for matching corresponding Wait and Post calls.

Warning: 32-bit operations to the video subsystem can cause a diskette overrun in the 1.44M mode because data width conversions may require more than 12 microseconds. If an overrun occurs, BIOS returns an error code and the operation should be retried.

Interfaces

There are four interfaces to be used by the multitasking dispatcher:

Startup: First, the startup code hooks interrupt hex 15. The dispatcher is responsible to check for function codes of AH= hex 90 or 91. The "Wait" and "Post" sections describe these codes. The dispatcher must pass all other functions to the previous user of interrupt hex 15. This can be done by a JMP or a CALL. If the function code is hex 90 or 91, the dispatcher should do the appropriate processing and return by the IRET instruction.

Serialization: It is up to the multitasking system to ensure that the device driver code is used serially. Multiple entries into the code can result in serious errors.

Wait: Whenever the BIOS is about to enter a busy loop, it first issues an interrupt hex 15 with a function code of hex 90 in AH. This signals a wait condition. At this point, the dispatcher should save the task status and dispatch another task. This allows overlapped execution of tasks when the hardware is busy. The following is an outline of the code that has been added to the BIOS to perform this function.

```
MOV AX, 90XXH          ; wait code in AH and
                       ; type code in AL
INT 15H                ; issue call
JC  TIMEOUT            ; optional: for time-out or
                       ; if carry is set, time-out
                       ; occurred
NORMAL TIMEOUT LOGIC ; normal time-out
```

Post: Whenever the BIOS has set an interrupt flag for a corresponding busy loop, an interrupt hex 15 occurs with a function code of hex 91 in AH. This signals a Post condition. At this point, the dispatcher should set the task status to "ready to run" and return to the interrupt routine. The following is an outline of the code added to BIOS that performs this function.

```
MOV AX, 91XXH          ; post code AH and
                       ; type code AL
INT 15H                ; issue call
```

Classes

The following types of wait loops are supported:

- The class for hex 0 to 7F is serially reusable. This means that for the devices that use these codes, access to the BIOS must be restricted to only one task at a time.
- The class for hex 80 to BF is reentrant. There is no restriction on the number of tasks that can access the device.
- The class for hex C0 to FF is noninterrupt. There is no corresponding interrupt for the wait loop. Therefore, it is the responsibility of the dispatcher to determine what satisfies this condition to exit from the loop.

Function Code Classes

Type Code (AL)	Description
00H->7FH	Serially reusable devices; the operating system must serialize access.
80H->0BFH	Reentrant devices; ES:BX is used to distinguish different calls (multiple I/O calls are allowed simultaneously).
0C0H->0FFH	Wait-only calls; there is no complementary Post for these waits--these are time-out only. Times are function-number dependent.

Function Code Assignments: The following are specific assignments for the Personal System/2 BIOS. Times are approximate. They are grouped according to the classes described under "Function Code Classes."

Type Code (AL)	Time Out	Description
00H	Yes (12 seconds)	Fixed Disk
01H	Yes (2 seconds)	Diskette
02H	No	Keyboard
0FCH	Yes	Fixed Disk Reset
0FDH	Yes (500-ms Read/Write)	Diskette Motor Start
0FEH	Yes (20 seconds)	Printer

Figure 3. Functional Code Assignments

The asynchronous support has been omitted. The serial and parallel controllers generate interrupts, but BIOS does not support them in the interrupt mode. Therefore, the support should be included in the multitasking system code if that device is to be supported.

Time-Outs

To support time-outs properly, the multitasking dispatcher must be aware of time. If a device enters a busy loop, it generally should remain there for a specific amount of time before indicating an error. The dispatcher should return to the BIOS wait loop with the carry bit set if a time-out occurs.

Machine-Sensitive Programs

Programs can select machine-specific features, but they must first identify the machine and model type. IBM has defined methods for uniquely determining the specific machine type. The location of the machine model bytes can be found through interrupt 15 function code (AH) = hex C0. See the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference* for a listing of model bytes for IBM Personal Computer and Personal System/2 products.

Math Coprocessor Compatibility

IBM systems use three math coprocessors: the 8088- and 8086-based systems use the 8087, the 80286-based systems use the 80287, and the 80386-based systems use the 80387. In the Real Address mode and Virtual 8086 mode, the 80386/80387 is upward object-code compatible with software for the 8086/8087 and 80286/80287 Real-Address mode systems; in the Protected mode, the 80386/80387 is upward object-code compatible with software for the 80286/80287 Protected-mode systems. However, if a math coprocessor instruction other than FINIT, FSTSW, or FSTCW is executed by an 80386-based system without an 80387 present, the 80386 waits indefinitely for a response from the 80387. This causes the system to stop processing without providing an error indication. To prevent this problem, software should check for the presence of the 80387 before executing math coprocessor instructions. The BIOS equipment function should be used when possible as the method for detecting the presence of the math coprocessor.

The only other differences of operation that may appear when 8086/8087 programs are ported to a Protected-mode 80386/80387 system (*not* using the Virtual 8086 mode), are in the format of operands for the administration instructions FLDENV, FSTEN, FRSTOR, and FSAVE. These instructions are normally used only by exception handlers and operating systems, not by application programs.

Operating Modes	Software Written for:		
	8087 Real	80287 Real	80287 Protected
8087 Real Mode	Yes	Yes*	No
80287 Real Mode	Yes*	Yes	No
80387 Real Mode	Yes***	Yes**	No
80387 8086 Virtual Mode	Yes***	Yes**	No
80287 Protected Mode	No	Yes**	Yes
80387 Protected Mode	No	No	Yes**

* See "8087 to 80287 Compatibility."
** See "80287 to 80387 Compatibility."
***See "8087 to 80287 Compatibility" and "80287 to 80387 Compatibility."

Figure 4. Math Coprocessor Software Compatibility

Many changes have been designed into the 80387 to directly support the IEEE standards in hardware. These changes result in increased performance by eliminating the need for software that supports the IEEE standard.

80287 to 80387 Compatibility

The following summarizes the differences between the 80387 and 80287 Math Coprocessors, and provides details showing how 80287 software can be ported to the 80387 Math Coprocessor:

Note: Any migration from 8087 directly to the 80387 must also take into account the differences between the 8087 and the 80287. This information is provided on page 25.

- The 80387 supports only affine closure for infinity arithmetic, not projective closure.
- Operands for FSCALE and FPATAN are no longer restricted in range (except $\pm\infty$); F2XM1 and FPTAN accept a wider range of operands.
- Rounding control is in effect for FLD *constant*.
- Software cannot change entries of the tag word to values (other than empty) that differ from actual register contents.
- In conformance with the IEEE standard, the 80387 does not support special data formats pseudozero, pseudo-NaN, pseudoinfinity, and unnormal.

Exceptions

When the overflow or underflow exception is masked, the only difference from the 80287 is in rounding when overflow or underflow occurs. The 80387 produces results that are consistent with the rounding mode.

For exceptions that are not masked, a number of differences exist due to the IEEE standard and to functional improvements to the architecture of the 80387:

- There are fewer invalid-operation exceptions due to denormal operands, because the instructions FSQRT, FDIV, FPREM, and conversions to BCD or to integer normalize denormal operands before proceeding.
- The FSQRT, FBSTP, and FPREM instructions may cause underflow, because they support denormal operands.
- The denormal exception can occur during the transcendental instructions and the FXTRACT instruction.
- The denormal exception no longer takes precedence over all other exceptions.
- When the operand is zero, the FXTRACT instruction reports a zero-divide exception and leaves $-\infty$ in ST(1).
- The status word has a new bit (SF) that signals when invalid-operation exceptions are due to stack underflow or overflow.
- FLD *extended precision* no longer reports denormal exceptions, because the instruction is not numeric.
- FLD *single/double precision* when the operand is denormal converts the number to extended precision and signals the denormalized operand exception. When loading a signaling NaN, FLD *single/double precision* signals an invalid-operation exception.
- The 80387 only generates quiet NaNs (as on the 80287); however, the 80387 distinguishes between quiet NaNs and signaling NaNs. Signaling NaNs trigger exceptions when they are used as operands; quiet NaNs do not (except for FCOM, FIST, and FBSTP, which also raise IE for quiet NaNs).

- Most 80387 numeric instructions are automatically synchronized by the 80386. No explicit Wait instructions are required for these instructions. To maintain compatibility with systems using the 8087, an explicit Wait is required before each numeric instruction.
- The FLDENV and FRSTOR instructions should be followed by an explicit Wait when used in the 80387 environment. An explicit Wait is not required after these instructions in the 80287 environment.
- The 80287 FSETPM (set Protected mode) instruction performs no useful purpose in the 80387 environment; if encountered, it is ignored.
- The format of the FSAVE and FSTENV instructions is determined by the current mode of the 80386; the Real Address mode format is used when the 80386 is in the Real Address mode, and the Protected mode format is used when the 80386 is in the Protected mode.
- The following applies only to the B1 stepping level 80386: An interrupt 9 does not occur for an operand outside a segment size; an interrupt 13 occurs.

8087 to 80287 Compatibility

The 80287 operating in the Real Address mode can execute 8087 software without major modifications. However, because of differences in the handling of numeric exceptions by the 80287 and the 8087, exception-handling routines *may* need to be changed.

The following summarizes the differences between the 80287 and 8087 Math Coprocessors, and provides details showing how 8087 software can be ported to the 80287 Math Coprocessor.

- The 8087 instructions FENI/FNENI and FDISI/FNDISI perform no useful function in the 80287 environment. If the 80287 encounters one of these opcodes in its instruction stream, the instruction is effectively ignored; none of the 80287 internal states are updated. While 8086 code containing these instructions may be executed on an 80287, it is unlikely that the exception-handling routines containing these instructions will be completely portable to the 80287.

- The ESC instruction address saved in the 80287 includes any leading prefixes before the ESC opcode. The corresponding address saved in the 8087 does not include leading prefixes.
- In the Protected mode, the format of the 80287 saved instruction and address pointers is different from the format of the 8087. The instruction opcode is not saved in the Protected mode; exception handlers have to retrieve the opcode from memory if needed.
- Interrupt 7 occurs in the 80286 when executing ESC instructions with either TS (task switched) or EM (emulation) of the 80286 MSW set (TS = 1 or EM = 1). If TS is set, then a Wait instruction also causes interrupt 7. An exception handler should be included in 80286 code to handle these exceptions.
- Interrupt 9 occurs if the second or subsequent words of a floating-point operand fall outside a segment size. Interrupt 13 occurs if the starting address of a numeric operand falls outside a segment size. An exception handler should be included in the 80286 code to report these programming errors.
- Most 80287 numeric instructions are automatically synchronized by the 80286. The 80286 automatically tests the 'busy' signal from the 80287 to ensure that the 80287 has completed its previous instruction before executing the next ESC instruction. Explicit Wait instructions are not required to ensure this synchronization. An 8087 used with 8086 and 8088 system microprocessors requires explicit Waits before each numeric instruction to ensure synchronization. Although 8086 software having explicit Wait instructions executes perfectly on the 80286 without reassembly, these Wait instructions are unnecessary.

  The processor control instructions for the 80287 may be coded using either a WAIT or No-WAIT form of the mnemonic. The WAIT forms of these instructions cause the assembler to precede the ESC instruction with a microprocessor Wait instruction.

Diskette Drives and Controller

The following figure shows the read, write, and format capabilities for each type of diskette drive.

Diskette Drive Type	160/180KB Mode	320/360KB Mode	1.44MB Mode	720KB Mode
5.25-Inch Diskette Drive:				
Single Sided (48 TPI)	R W F	---	---	---
Double Sided (48 TPI)	R W F	R W F	---	---
3.5-Inch Diskette Drive:				
720KB Drive	---	---	---	R W F
1.44MB Drive	---	---	R W F	R W F
R-Read W-Write F-Format				

Figure 5. Diskette Drive Read, Write, and Format Capabilities

Notes:

1. 5.25-inch diskettes designed for the 1.2M mode cannot be used in either a 160/180KB or a 320/360KB diskette drive.
2. 3.5-inch diskettes designed for the 1.44M mode cannot be used in a 720KB diskette drive.

Warning: 32-bit operations to the video subsystem can cause a diskette overrun in the 1.44M mode because data width conversions may require more than 12 microseconds. If an overrun occurs, BIOS returns an error code and the operation should be retried.

Copy Protection

The following methods of copy protection may not work on systems using the 3.5-inch 1.44M diskette drive.

- Bypassing BIOS Routines:
  - Data Transfer Rate: BIOS selects the proper data transfer rate for the media being used.
  - Diskette Parameters Table: Copy protection, which creates its own Diskette Parameters table, may not work on these drives.

- Diskette Drive Controls:
  - Rotational Speed: The time between two events on a diskette is a function of the controller.
  - Access Time: Diskette BIOS routines must set the track-to-track access time for the different types of media used in the drives.
  - Diskette Change Signal: Copy protection may not be able to reset this signal.
- Write Current Control—Copy protection that uses write current control will not work because the controller selects the proper write current for the media being used.

Detailed information about specific diskette drives is available in separate technical references.

Fixed Disk Drives and Controller

Reading from and writing to the fixed disk drive is initiated in the same way as with IBM Personal Computer products; however, new functions are supported. Detailed information about specific fixed disk drives and fixed disk adapters is available in system-specific technical references.

Index

A

application guidelines 1
assembler language programming considerations 3

B

back-to-back I/O commands 5
BASIC 3
busy loop 20
bypassing BIOS 27

C

classes, wait loop 20
COBOL 3
condition, wait 20
copy protection 27

D

diskette change signal 28
diskette compatibility 27
diskette write current 28
divide error exception 4

E

edge-triggered interrupts 1
exception, divide error 4

F

fixed disk controller 28
FORTRAN 3
function calls, operating system 15
function calls, ROM BIOS 15
function codes, multitasking 21

H

hardware interrupts 1
high-level language considerations 3

I

I/O commands and STI instructions 5
IBM C 3
interfaces, multitasking 19
interrupt, single step 4
IRQ 2 16
IRQ 9 16

L

level-sensitive interrupts 1
loop, busy 20

M

machine model bytes 22
machine-sensitive programs 22
math coprocessor compatibility 22
model bytes 22
multiple lockout instructions 5

O

opcodes 4
operating system 3
operating system function calls 15

P

paging 13
parameter tables, BIOS 16
Pascal 3
polling mechanism 18
post 20
provisions, multitasking 19
PUSH SP 4

S

serialization, multitasking 19
shift counts 4
single step interrupt 4
software interrupts 3
startup, multitasking 19
STI instruction 5
system compatibility 1

T

time-outs 21

W

wait condition 20
wait loop classes 20
write current, diskette 28

Numerics

80286 anomalies 8
80386 anomalies 8

Bibliography

Microprocessor and Peripheral Handbook. INTEL Corporation, *210844.001*

Introduction to the iAPX 286. INTEL Corporation, *210308.001*

iAPX 286 Operating Systems Writer's Guide. INTEL Corporation, *121960.001*

iAPX 286 Programmer's Reference Manual. INTEL Corporation, *210498.001*

iAPX 286 Hardware Reference Manual. INTEL Corporation, *210760.001*

Numeric Processor Extension Data Sheet. INTEL Corporation, *210920*

80287 Support Library Reference Manual. INTEL Corporation, *122129*

80386 Hardware Reference Manual. INTEL Corporation, *231732.001*

Introduction to the 80386. INTEL Corporation, *231252.001*

80386 Programmer's Reference Manual. INTEL Corporation, *230985.001*

80386 System Software Writer's Guide. INTEL Corporation, *231499*

National Semiconductor Corporation, NS16550

Motorola Microprocessor's Data Manual. Motorola Inc. *Series B*

Notes:

Personal System/2 Model 50 Technical Reference

First Edition (May, 1988)

The following paragraph does not apply to the United Kingdom or any country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.

It is possible that this publication may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

THE PUBLICATION OF THE INFORMATION CONTAINED HEREIN IS NOT INTENDED TO AND DOES NOT CONVEY ANY RIGHTS OR LICENSES, EXPRESS OR IMPLIED, UNDER ANY IBM PATENTS, COPYRIGHTS, TRADEMARKS, MASK WORKS OR ANY OTHER INTELLECTUAL PROPERTY RIGHTS.

Requests for copies of this publication and for technical information about IBM products should be made to your IBM Authorized Dealer or your IBM Marketing Representative.

Personal System/2 is a registered trademark of the International Business Machines Corporation.

© Copyright International Business Machines Corporation 1988. All rights reserved. No part of this work may be reproduced or distributed in any form or by any means without prior permission in writing from the IBM Corporation.

Preface

The Technical Reference library is intended for those who develop hardware and software products for IBM Personal Computers and IBM Personal System/2. Users should understand computer architecture and programming concepts.

This technical reference provides hardware and software interface information for the IBM Personal System/2 Model 50 and should be used with the following publications:

IBM Personal System/2 Hardware Interface Technical Reference

IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference

This manual consists of the following sections:

Section 1, "System Overview," describes the system, features and specifications.

Section 2, "Programmable Option Select," describes registers used for configuration.

Section 3, "System Board," describes the system specific hardware implementations.

Warning: The term "Reserved" describes certain signals, bits, and registers that should not be changed. Use of reserved areas can cause compatibility problems, loss of data, or permanent damage to the hardware. When the contents of a register are changed, the state of the reserved bits must be preserved. When possible, read the register first and change only the bits that must be changed.

For information about components or devices not described in this manual, refer to the *Hardware Interface Technical Reference.* Information about diskette drives, fixed disk drives, adapters, and external options are in separate option technical references.

Notes:







Contents

Section 1. System Overview 1-1
Description 1-3
System Board Features 1-3
System Board Block Diagram 1-5
System Board I/O Address Map 1-6
Specifications 1-7

Section 2. Programmable Option Select 2-1
Description 2-3
POS Address Map 2-5
Card Selected Feedback 2-6
System Board Setup 2-6
System Board Enable/Setup Register (Hex 0094) 2-7
System Board POS Register 2 (Hex 0102) 2-8
System Board POS Register 3 (Hex 0103) 2-9
Adapter Enable/Setup Register (Hex 0096) 2-10

Section 3. System Board 3-1
Description 3-3
Micro Channel 3-3
Adapter Identification 3-3
Central Arbiter 3-3
Memory 3-7
Read-Only Memory Subsystem 3-7
Random Access Memory Subsystem 3-7
Real-Time Clock/Complementary Metal-Oxide Semiconductor RAM 3-12
Miscellaneous System Functions 3-24
Nonmaskable Interrupt 3-24
System Control Port B (Hex 0061) 3-24
System Control Port A (Hex 0092) 3-26
Power-On Password 3-28
Type 2 Serial Port Controller 3-29
Hardware Compatibility 3-29

Index X-1

Notes:

Figures

1-1. Model and Submodel Bytes 1-3
1-2. System Board Devices and Features 1-4
1-3. System Board 1-5
1-4. System Board I/O Address Map 1-6
1-5. Performance Specifications – Type 1 System Board . . 1-7
1-6. Performance Specifications – Type 2 System Board . . 1-8
1-7. Physical Specifications 1-9
2-1. POS I/O Address Map 2-5
2-2. Card Selected Feedback Register (Hex 0091) 2-6
2-3. System Board Enable/Setup Register (Hex 0094) 2-7
2-4. System Board POS Register 2 (Hex 0102) 2-8
2-5. Parallel Port Select Bits 2-8
2-6. System Board POS Register (Hex 0103) 2-9
2-7. Adapter Enable/Setup Register (Hex 0096) 2-10
3-1. Arbitration Bus Priority Assignments 3-4
3-2. Arbitration Register, Write to Hex 0090 3-5
3-3. Arbitration Register, Read Hex 0090 3-5
3-4. System Memory Map 3-8
3-5. System Board Memory Connector, Type 1 3-9
3-6. System Board Memory Connector, Type 2 3-10
3-7. Presence Detect Encoding 3-11
3-8. RT/CMOS RAM Address Map 3-12
3-9. RT/CMOS Address Register and NMI Mask (Hex 0070) 3-13
3-10. RT/CMOS Data Register (Hex 0071) 3-13
3-11. Real-Time Clock Bytes 3-15
3-12. Status Register A 3-16
3-13. Status Register B 3-16
3-14. Status Register C 3-18
3-15. Status Register D 3-18
3-16. Diagnostic Status Byte 3-19
3-17. Diskette Drive Type Byte 3-20
3-18. Diskette Drive Type Byte (Bits 7 - 4) 3-21
3-19. Diskette Drive Type Byte (Bits 3 - 0) 3-21
3-20. Equipment Byte 3-22
3-21. Equipment Byte (Bits 7, 6) 3-22
3-22. Equipment Byte (Bits 5, 4) 3-22
3-23. System Control Port B (Write) 3-25
3-24. System Control Port B (Read) 3-25
3-25. System Control Port A (Hex 0092) 3-26

Notes:

Section 1. System Overview

Description 1-3
System Board Features 1-3
System Board Block Diagram 1-5
System Board I/O Address Map 1-6
Specifications 1-7

Notes:

Description

The IBM Personal System/2 Model 50 is a self-contained, desktop computer system with a keyboard. It can support two diskette drives and one internal fixed disk drive.

A system can have either a Type 1 or Type 2 system board. The major differences between the system boards are component layout and memory. Programs identify the type by reading the model and submodel bytes and BIOS revision code. Interrupt hex 15, function code (AH) = hex C0, returns the model and submodel bytes and BIOS revision code. The following figure shows these bytes and the system board types.

Model Byte	Submodel Byte	Revision Code	System Board
FC	04	00	Type 1
FC	04	03	Type 2

Figure 1-1. Model and Submodel Bytes

Refer to the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference* for a listing of other systems, and check the supplements section for updates to that listing.

System Board Features

The following figure lists the system board devices and features. The *Hardware Interface Technical Reference* describes devices common to PS/2 products by type number.

Device	Type	Features
Microprocessor	---	80286 24-bit address and 16-bit data interface
System Timers	1	Channel 0 – System timer Channel 2 – Tone generation for speaker Channel 3 – Watchdog timer
ROM Subsystem	---	128KB (KB = 1024 bytes)
RAM Subsystem	---	1MB - Type 1 and 2 System Boards * (MB = 1,048,576 Bytes) Expandable on the channel Type 2 can be expanded on the system board to 2MB
CMOS RAM Subsystem	---	64-byte CMOS RAM with real-time clock/calendar Battery backup
Video Subsystem	1	Auxiliary connector on the channel Analog output 256KB video memory
Audio Subsystem	1	Driven by: - System-timer channel 2 - The 'audio sum node' signal.
DMA Controller	1	Eight independent DMA channels Single or burst transfers and read verification
Interrupt Controller	1	16 levels of system interrupts Interrupts are level-sensitive
Keyboard/Auxiliary Device Controller	1	Keyboard connector Auxiliary device connector Password security
Diskette Drive Controller	1	Supports: - 720KB formatted diskette density - 1.44MB formatted diskette density.
Serial Controller	1, 2**	RS-232C interface Programmable as serial port 1 or 2
Parallel Controller	1	Programmable as parallel port 1, 2, or 3 Supports bidirectional input and output
Micro Channel	---	Four channel connectors: - 8- or 16-bit device support - One with an auxiliary video extension - One occupied by the fixed disk drive adapter
Math Coprocessor Socket	---	Supports 80287 math coprocessor Same clock speed as the system microprocessor
Power Supply	1	110 and 220 Vac support

* Some models support 2MB on the Type 2 system board.
** The Model 50 does not support the FIFO mode.

Figure 1-2. System Board Devices and Features

System Board Block Diagram

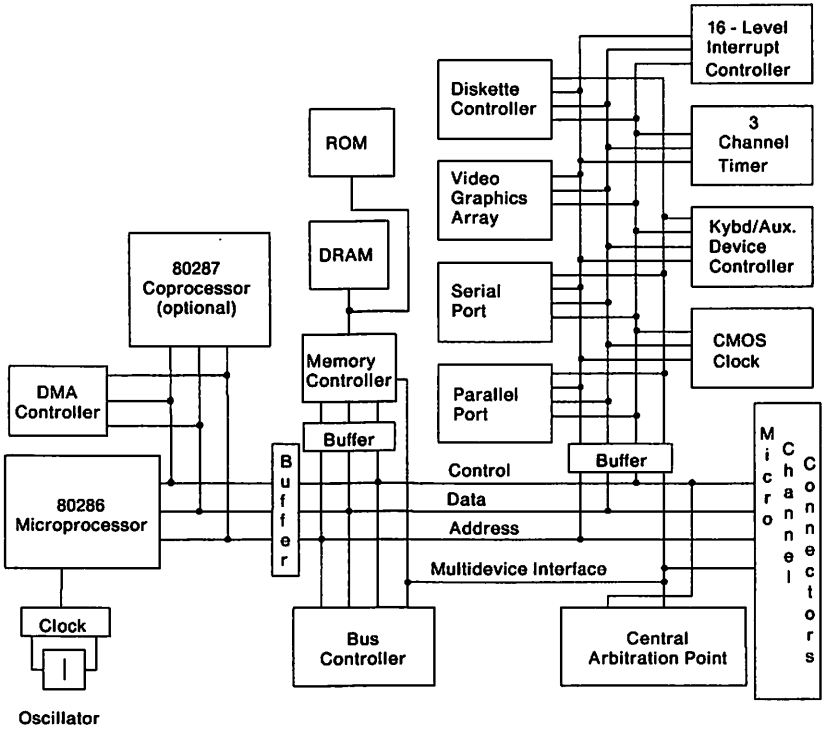


Figure 1-3. System Board

System Board I/O Address Map

Hex Addresses	Device
0000 - 001F	DMA Controller (0 – 3)
0020, 0021	Interrupt Controller (Master)
0040, 0042 - 0044, 0047	System Timers
0060	Keyboard, Auxiliary Device
0061	System Control Port B
0064	Keyboard, Auxiliary Device
0070, 0071	RT/CMOS and NMI Mask
0081 - 0083, 0087	DMA Page Registers (0 – 3)
0089, 008A, 008B, 008F	DMA Page Registers (4 – 7)
0090	Central Arbitration Control Point
0091	Card Selected Feedback Register
0092	System Control Port A
0094	System Board Enable/Setup Register
0096	Adapter Enable/Setup Register
00A0 - 00A1	Interrupt Controller (Slave)
00C0 - 00DF	DMA Controller (4 – 7)
00F0 - 00FF	Math Coprocessor
0100 - 0107	Programmable Option Select
01F0 - 01F8	Fixed Disk Drive Controller
0278 - 027B	Parallel Port 3
02F8 - 02FF	Serial Port 2 (RS-232C)
0378 - 037B	Parallel Port 2
03B4, 03B5, 03BA	Video Subsystem
03BC - 03BF	Parallel Port 1
03C0 - 03C5	Video Subsystem
03C6 - 03C9	Video DAC
03CA, 03CC, 03CE, 03CF	Video Subsystem
03D4, 03D5, 03DA	Video Subsystem
03F0 - 03F7	Diskette Drive Controller
03F8 - 03FF	Serial Port 1 (RS-232C)

Figure 1-4. System Board I/O Address Map

Specifications

Device	Number of Waits	Cycle Time (ns)
Microprocessor (10 MHz – 100 ns Clock):		
Access to System Board RAM	1	300
Access to System Board ROM	1	300
Access to Channel:		
Default Transfer Cycle:		
I/O Access	1	300
Memory Access	0	200
Synchronous Extended Transfer Cycle	1	300
Refresh Rate (Typically performed every 15.1 μs)		500 (min)
Bus Master Access to System Board RAM		300 (min)
DMA Controller (10 MHz – 100 ns Clock):		
Single Transfer: 300 + I/O Access + Memory Access		
Burst Transfers: 300 + (I/O Access + Memory Access)N *		
System Board Memory Access		300
Default Transfer Cycle:		
I/O Access		300
Memory Access		200
Synchronous Extended Transfer Cycle		300

* N is the number of transfers in the burst.

Figure 1-5. Performance Specifications – Type 1 System Board

The Type 2 system board supports two speeds of system board memory. Cycle times differ depending upon the speed. See Figure 3-7 on page 3-11.

Device	Number of Waits	Cycle Time (ns)
Microprocessor (10 MHz – 100 ns Clock):		
Access to System Board RAM:		
85-ns Memory	0	200
120-ns Memory	1	300
Access to System Board ROM	1	300
Access to Channel:		
Default Transfer Cycle:		
I/O Access	1	300
Memory Access	0	200
Synchronous Extended Transfer Cycle	1	300
Refresh Rate (Typically performed every 15.1 μs)		500 (min)
Bus Master Access to System Board RAM		300 (min)
DMA Controller (10 MHz – 100 ns Clock):		
Single Transfer: 300 + I/O Access + Memory Access		
Burst Transfers: 300 + (I/O Access + Memory Access)N *		
System Board Memory Access		
85-ns Memory		200
120-ns Memory		300
Default Transfer Cycle:		
I/O Access		300
Memory Access		200
Synchronous Extended Transfer Cycle		300

* N is the number of transfers in the burst.

Figure 1-6. Performance Specifications – Type 2 System Board

Size:	
Width	360 mm (14.1 in)
Depth	420 mm (16.5 in)
Height	140 mm (5.5 in)
Weight	9.55 kg (21 lb)
Cables:	
Power Cable	1.8 m (6 ft)
Keyboard Cable	1.8 m (6 ft)
Air Temperature:	
System On	15.6 to 32.2°C (60 to 90°F)
System Off	10.0 to 43.0°C (50 to 110°F)
Humidity:	
System On	8% to 80%
System Off	20% to 80%
Maximum Altitude	2133.6 m (7000 ft)
Heat Output	144.8 W/hr
Acoustical Readings from 1 m (3.28 ft):	
Operating	46 dB average
Idle	40 dB average
Electrical:	
Power Supply Input	
Low Range	90 (min) - 137 (max)Vac
High Range	180 (min) - 265 (max)Vac
Maximum Current Draw:	
Low Range	2.7 A
High Range	1.4 A
Frequency	50 - 60Hz ± 3Hz
Maximum Allowable Current:	
Keyboard Port	275 mA
Auxiliary Device Port	300 mA
Electromagnetic Compatibility	FCC Class B

Figure 1-7. Physical Specifications

Notes:

Section 2. Programmable Option Select

Description 2-3
POS Address Map 2-5
Card Selected Feedback 2-6
System Board Setup 2-6
System Board Enable/Setup Register (Hex 0094) 2-7
System Board POS Register 2 (Hex 0102) 2-8
System Board POS Register 3 (Hex 0103) 2-9
Adapter Enable/Setup Register (Hex 0096) 2-10

Notes:

Description

Programmable Option Select (POS) eliminates the need for switches by replacing their function with programmable registers. This section describes the POS information used on the Model 50 system board. For additional POS information, refer to the *Hardware Interface Technical Reference*.

Warning:

- IBM recommends that programmable options be set only through the System Configuration utilities. Directly setting the POS registers or CMOS RAM POS parameters can cause multiple assignment of the same system resource, improper operation of the feature, loss of data, or possible damage to the hardware.
- Application programs should not use the adapter identification (ID) unless absolutely necessary. Compatibility problems can result.
- If an adapter and the system board are in setup mode at the same time, bus contention will occur, no useful programming can take place, and damage to the hardware can occur.
- After setup operations are complete, the Adapter Enable/Setup register (hex 0096) should be set to hex 00, and the System Board Enable/Setup register (hex 0094) should be set to hex FF.
- The channel reset bit (bit 7) in the Adapter Enable/Setup register must be 0 to program the adapters.
- The system board does not support 16-bit I/O operations to 8-bit POS registers. Using 16-bit I/O instructions on 8-bit POS registers will cause erroneous data to be written to or read from the registers. Only 8-bit transfers are supported for setup operations.

Setup functions respond to I/O addresses hex 0100 through 0107 only when their unique setup signal is active. The following precautions must be taken before setting individual bits in the POS registers.

System Board Video Subsystem Setup:

- Bit 5 in the System Board Enable/Setup register (hex 0094) must be set to 0 to place the system board video into the setup mode.
- Bit 3 in the Adapter Enable/Setup register (hex 0096) must be set to 0 to avoid driving a 'setup' signal to an adapter.
- Bit 7 in the System Board Enable/Setup register (hex 0094) must be set to 1 to avoid driving a 'setup' signal to other system board functions.

Adapter Setup:

- Bit 3 in the Adapter Enable/Setup register (hex 0096) must be set to 1 to allow adapter setup.
- Bit 5 in the System Board Enable/Setup register (hex 0094) must be set to 1 to avoid driving a 'setup' signal to the Video Subsystem.
- Bit 7 in the System Board Enable/Setup register (hex 0094) must be set to 1 to avoid driving a 'setup' signal to a system board function.

Other System Board Setup:

- Bit 7 in the System Board Enable/Setup register (hex 0094) must be set to 0 to allow setup of other system board functions.
- Bit 3 in the Adapter Enable/Setup register (hex 0096) must be set to 0 to avoid driving a 'setup' signal to an adapter.
- Bit 5 in the System Board Enable/Setup register (hex 0094) must be set to 1 to avoid driving a 'setup' signal to the Video Subsystem.

POS Address Map

The following figure shows the organization of the I/O address space used by POS. The POS registers are 8-bit registers. Bit 0 of POS Register 2 and bits 6 and 7 of POS Register 5 are defined. All other bits in POS Registers 2 through 5 are free-form.

Address (Hex)	Function
0094	System Board Enable/Setup Register
0096	Adapter Enable/Setup Register
0100	POS Register 0 – Adapter Identification Byte (Low Byte)
0101	POS Register 1 – Adapter Identification Byte (High Byte)
0102	POS Register 2 – Option Select Data Byte 1 Bit 0 is Card Enable.
0103	POS Register 3 – Option Select Data Byte 2
0104	POS Register 4 – Option Select Data Byte 3
0105	POS Register 5 – Option Select Data Byte 4 Bit 7 is the channel check active indicator. Bit 6 is the channel check status-available indicator.
0106	POS Register 6 – Subaddress Extension (Low Byte)
0107	POS Register 7 – Subaddress Extension (High Byte)

Figure 2-1. POS I/O Address Map

Card Selected Feedback

When the adapter is addressed, it responds by setting the '-card selected feedback' signal (-CD SFDBK) to active. -CD SFDBK is derived from the address decode and driven by a totem pole driver. It is latched by the system board and can be read through the Card Selected Feedback register at address hex 0091. Diagnostic and automatic configuration programs use this signal to verify the operation of an adapter at a given address or DMA port. This signal must not be active during a setup cycle.

The Card Selected Feedback register is a read-only register at address hex 0091. It allows programs to monitor -CD SFDBK and thereby determine if the video subsystem, system board I/O, or an adapter is addressed and functioning.

Bit	Function
7 - 1	Reserved
0	-Card Selected Feedback

Figure 2-2. Card Selected Feedback Register (Hex 0091)

Bits 7 - 1 Reserved.

Bit 0 This bit is set to 1 whenever -CD SFDBK was active on a previous cycle or whenever the system board I/O functions (diskette drive, serial, or parallel interfaces) are accessed by an I/O cycle. Reading this register resets the bit to 0.

System Board Setup

The integrated I/O functions on the system board use POS information during setup. The diskette drive controller, serial port, and parallel port are treated as a single device. The video subsystem is also an integrated part of the system board; however, POS treats it as a separate device. The System Board Enable/Setup register is used to place the system board or video subsystem into the setup mode.

System Board Enable/Setup Register (Hex 0094)

This is a read/write register; all bits in this register default to 1 (enabled).

Bit	Function
7	Enable/-Setup System Board Functions
6	Reserved
5	Enable/-Setup Video Subsystem
4 - 0	Reserved

Figure 2-3. System Board Enable/Setup Register (Hex 0094)

Bit 7 When this bit is set to 0, various system board I/O functions are placed in the setup mode. The diskette drive controller, serial port, and parallel port are controlled through System Board POS Register 2 (hex 0102). The memory is controlled through System Board POS Register 3 (hex 0103).

When this bit is set to 1, the system board functions are enabled.

Bit 6 Reserved.

Bit 5 When this bit is set to 0, the video subsystem is placed in the setup mode and controlled through POS Register 2 (hex 0102). When set to 1, and bit 0 in hex 0102 is set to 1, video is enabled.

Bit 0 of POS Register 2 is the video enable bit. When this bit is set to 0, the video subsystem does not respond to commands, addresses, or data. If video is being generated when the video enable bit is set to 0, the output is still generated. For information on BIOS calls to enable or disable the video, see the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference.*

Note: When the video is disabled, access to the video DAC registers is disabled.

Bits 4 - 0 Reserved.

System Board POS Register 2 (Hex 0102)

When the system board is in the setup mode, the diskette drive controller, serial port, and parallel port are controlled by this read/write register. Reading this register returns the current state of these system board functions.

Bit	Function
7	Disable Parallel Port Extended Mode
6, 5	Parallel Port Select
4	Enable Parallel Port
3	Serial Port Select
2	Enable Serial Port
1	Enable Diskette Drive Interface
0	Enable System Board

Figure 2-4. System Board POS Register 2 (Hex 0102)

Bit 7 When set to 0, this bit allows the parallel port to be configured as an 8-bit, parallel, bidirectional interface. When set to 1, this bit disables the bidirectional mode. This bit is set to 0 at power-on, and POST sets it to 1.

Bits 6, 5 These bits select the configuration of the system board parallel port.

Bits 6 5	Assignment	Hex Address	Interrupt Level
0 0	Parallel 1	03BC - 03BF	7
0 1	Parallel 2	0378 - 037B	7
1 0	Parallel 3	0278 - 027B	7
1 1	Reserved	----------	-

Figure 2-5. Parallel Port Select Bits

Bit 4 When this bit and bit 0 are set to 1, the system board parallel port is enabled.

Bit 3 When set to 1, this bit sets the system board serial port as Serial 1 (addresses hex 03F8 through 03FF), which uses interrupt level 4. When set to 0, this bit sets the serial port as Serial 2 (addresses hex 02F8 through 02FF), which uses interrupt level 3.

Bit 2 When this bit and bit 0 are set to 1, the system board serial port is enabled.

Bit 1 When this bit and bit 0 are set to 1, the diskette drive interface is enabled.

Bit 0 When set to 1, this bit allows bits 4, 2, and 1 to enable and disable their respective devices. When set to 0, this bit disables the diskette drive interface, system board serial port, and system board parallel port, regardless of the state of bits 4, 2, and 1.

System Board POS Register 3 (Hex 0103)

With the system board in setup, this register controls the system board memory.

Bit	Function
7 - 1	Reserved
0	Enable System Board RAM

Figure 2-6. System Board POS Register (Hex 0103)

Bit 7 - 1 Reserved.

Bit 0 When set to 1, this bit enables the system board memory. When set to 0, this bit disables the memory. All system board memory is enabled or disabled by this operation.

Adapter Enable/Setup Register (Hex 0096)

The Adapter Enable/Setup register selects the connector to be configured.

Bit	Symbol
7	Channel Reset
6 - 4	Reserved
3	Card Setup Enable
2 - 0	Channel Select 2 - 0

Figure 2-7. Adapter Enable/Setup Register (Hex 0096)

Bit 7 When set to 1, this bit activates the 'channel reset' signal to all connectors.

Bits 6 - 4 These bits are reserved.

Bit 3 When set to 1, this bit enables the '-card setup' signal selected by bits 2 through 0.

Bits 2 - 0 These bits are the address bits for the '-card setup' signal Connectors 1 through 4 are addressed as 0 through 3, respectively. When bit 3 is set to 1, these bits select the connector that is put into setup.

Each channel connector has a unique '-card setup' signal (-CD SETUP) associated with it. This signal is used to put the adapters in the setup mode, which allows access to the POS registers. The individual connectors are selected through the Adapter Enable/Setup register. Setup information is then read from or written to the selected adapter through I/O addresses hex 0100 through 0107.

Notes:

1. -CD SETUP only goes active when an operation is performed in the I/O address range hex 0100 through 0107.
2. The status of port hex 0096 can be read by software. However, when the port is read, bits 6, 5, and 4 are set to 1.

Section 3. System Board

Description 3-3
Micro Channel 3-3
  Adapter Identification 3-3
  Central Arbiter 3-3
    Arbitration Bus Priority Assignments 3-4
    Central Arbiter Programming 3-5
Memory 3-7
  Read-Only Memory Subsystem 3-7
  Random Access Memory Subsystem 3-7
    System Memory Maps 3-8
    System Board Memory Connector 3-9
  Real-Time Clock/Complementary Metal-Oxide Semiconductor RAM 3-12
    RT/CMOS Address Register and NMI Mask (Hex 0070) . . 3-13
    RT/CMOS Data Register (Hex 0071) 3-13
    RT/CMOS RAM I/O Operations 3-14
    Real-Time Clock Bytes (Hex 000-00D) 3-15
    CMOS RAM Configuration 3-19
Miscellaneous System Functions 3-24
  Nonmaskable Interrupt 3-24
  System Control Port B (Hex 0061) 3-24
  System Control Port A (Hex 0092) 3-26
  Power-On Password 3-28
Type 2 Serial Port Controller 3-29
Hardware Compatibility 3-29

Notes:

Description

This section provides information about the Model 50 implementation of the Micro Channel, memory subsystem, and miscellaneous system ports. For additional information about these and other topics, refer to the *Hardware Interface Technical Reference.*

Micro Channel

POST routines and registers used to implement programmable features of the Micro Channel architecture may vary from system to system. This section provides information about the Model 50 implementation of the Micro Channel architecture.

Adapter Identification

When the system is powered on, an adapter can issue an ID of hex 0000 for up to 1 second after channel reset to indicate it is not ready. Any adapter that continues to issue an adapter ID of hex 0000 (not ready) for more than 1 second is considered defective. On the Model 60, if the channel reset was caused by a keyboard sequence or a program routine, this time is reduced to 740 milliseconds.

Central Arbiter

The central arbitration control point gives intelligent subsystems on the channel the ability to share and control the system. It allows burst data transfers and prioritization of control between devices. This arbiter supports up to 16 arbitrating devices.

Arbitration Bus Priority Assignments

The following figure shows the assignment of arbitration levels. The functions with the lowest arbitration level have the highest priority.

ARB Level	Primary Assignment
-2	Memory Refresh
-1	NMI
0	DMA Channel 0 (Programmable to any arbitration level)
1	DMA Channel 1
2	DMA Channel 2
3	DMA Channel 3
4	DMA Channel 4 (Programmable to any arbitration level)
5	DMA Channel 5
6	DMA Channel 6
7	DMA Channel 7
8-E	Available
F	System Microprocessor

Figure 3-1. Arbitration Bus Priority Assignments

Note: Devices designed for arbitration level 0 or 1 should have limited bandwidth or short bursts so diskette overruns can be prevented or recovered by retry operations. The diskette drive controller, on arbitration level 2, can be held inactive by devices on levels 0 and 1, by a refresh operation, and by the previous controlling master. The diskette drive controller should not be held inactive for more than 12 microseconds to prevent overrun.

NMI service is executed at a priority level higher than 0, called -1. Memory refresh is prioritized at -2, two levels higher than 0. Levels -1 and -2 are reached on the system board only, while the 'arbitrate/-grant' signal (ARB/-GNT)) is in the arbitrate state.

When the central arbitration control point receives a level -1 request (NMI, a system-board internal signal), it activates -PREEMPT, waits for the end of transfer, and then places ARB/-GNT in the arbitrate state, which denies channel activity to arbitrating devices. The central arbitration control point gives the grant to the level -1 request, and holds ARB/-GNT in the arbitrate state until the operation is complete and the NMI is reset.

Central Arbiter Programming

The central arbitration control point provides access to programmable options through the Arbitration register, which is accessed at I/O address hex 0090. The bits are defined differently for read and write operations, as shown in the following figures.

Bit	Definition
7	Enable System Microprocessor Cycle
6	Arbitration Mask
5	Enable Extended Arbitration
4 - 0	Reserved

Figure 3-2. Arbitration Register, Write to Hex 0090

Bit	Definition
7	Enable System Microprocessor Cycle
6	Arbitration Masked by NMI
5	Bus Time-out
4	Reserved
3 - 0	Value of Arbitration Bus During Previous Grant State

Figure 3-3. Arbitration Register, Read Hex 0090

Bit 7 Setting this bit to 1 enables system microprocessor cycles during arbitration cycles. This bit can be set to 0 if an arbitrating device requires total control of the channel bandwidth. This bit is set to 0 by a system reset.

Reading this bit as a 1 indicates system microprocessor cycles are enabled during arbitration.

Bit 6 Setting this bit to 1 causes the central arbitration control point to enter the arbitration state. The system microprocessor controls the channel until this bit is reset to 0. This bit is set to 0 by a system reset.

Reading this bit as a 1 indicates that an NMI has occurred and has masked arbitration.

Warning: This bit should be set to 1 only by diagnostic routines and system error-recovery routines.

Bit 5 Setting this bit to 1 enables extended arbitration. The minimum arbitration cycle is 300 nanoseconds; this bit extends that minimum cycle to 600 nanoseconds. This bit is set to 0 during a system reset.

Reading this bit as a 1 indicates that a bus time-out has occurred, and resets bit 6 in this register to 0.

Bit 4 This bit is reserved and should be 0.

Bits 3 - 0 These bits are undefined for a write operation and should be set to 0.

Reading these bits returns the arbitration level of the arbiter controlling the channel during the most recent grant state. This information allows the system microprocessor to determine the arbitration level of the device that caused a bus time-out.

Memory

The Model 50 system uses the following types of memory:

- Read-only memory (ROM)
- Random access memory (RAM)
- Real-time Clock and CMOS RAM.

Read-Only Memory Subsystem

The ROM subsystem on the Type 1 system board consists of four 32Kb by 8-bit modules in a 64Kb by 16-bit arrangement (Kb equals 1024 bits). The ROM subsystem on the Type 2 system board consists of two 64Kb by 8-bit modules in a 64Kb by 16-bit arrangement.

ROM is not parity-checked and is assigned addresses at the top of the first and last 1MB of address space (hex 0E0000 and FE0000).

Random Access Memory Subsystem

The RAM subsystem on the system board starts at address hex 000000 of the 16MB address space. It consists of 1MB of RAM modules.

The Type 1 system board uses two, 512KB by 9-bit memory module packages that operate with one wait state. The Type 2 system board supports the following memory module packages:

- 1MB, 120 nanosecond (one wait state)
- 1MB, 85 nanosecond (zero wait state)
- 2MB, 85 nanosecond (zero wait state).

Memory-refresh typically occurs once every 15 microseconds.

The timing of the memory signals on the Type 1 system board is similar to the Hitachi[1] HB61009BR-15, 150-nanosecond DRAM or equivalent.

The timing of the memory signals on the Type 2 system board with 85-nanosecond memory installed is similar to the Hitachi HM51256CP-8, 85-nanosecond DRAM or equivalent. The timing of the memory signals on the Type 2 system board with 120-nanosecond memory installed is similar to the Hitachi HM51256CP-12, 120-nanosecond DRAM or equivalent.

System Memory Maps

The following figure shows the mapping of the memory locations on the Model 50. A 256-byte portion of this RAM is reserved as a BIOS data area. A 1KB portion of this RAM is reserved as an extended BIOS data area. See the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference* for details.

Hex Range	Function
000000 to 09FFFF	640KB System Board RAM *
0A0000 to 0BFFFF	128KB Video RAM
0C0000 to 0DFFFF	128KB I/O Expansion ROM
0E0000 to 0FFFFF	128KB System Board ROM
100000 to 15FFFF	384KB System Board RAM
160000 to FDFFFF	Channel Expansion Memory Addresses
FE0000 to FFFFFF	128KB System Board ROM

* With the system board in the setup mode, the system board memory can be disabled through System Board POS Register 3 (see page 2-9).

Figure 3-4. System Memory Map

[1] Hitachi is a trademark of the Hitachi Corporation.

System Board Memory Connector

The Type 1 and 2 system boards use different types of memory connectors. The Type 1 system board has two 30-pin connectors and comes with two 512KB memory module packages installed. The Type 2 system board has one 72-pin connector and comes with one 1MB memory module package installed. The following figures show the pin assignments and signal definitions for these connectors.

Pin	I/O	Signal	Pin	I/O	Signal
1	N/A	5 Vdc	2	I	-Column Address Strobe
3	I/O	Data Bit 1	4	I	Address Bit 1
5	I	Address Bit 2	6	I/O	Data Bit 2
7	I	Address Bit 3	8	I	Address Bit 4
9	N/A	Ground	10	I/O	Data Bit 3
11	I	Address Bit 5	12	I	Address Bit 6
13	I/O	Data Bit 4	14	I	Address Bit 7
15	I	Address Bit 8	16	I/O	Data Bit 5
17	I	Address Bit 9	18	N/A	No Connection
19	I	Row Address Strobe 1	20	I/O	Data Bit 6
21	I	-Write Strobe	22	N/A	Ground
23	I/O	Data Bit 7	24	O	Presence Detect
25	I/O	Data Bit 8	26	O	Presence Detect
27	I	Row Address Strobe	28	N/A	No Connection
29	I/O	Data Bit 9 (Parity)	30	N/A	+5 Vdc

Figure 3-5. System Board Memory Connector, Type 1

Type 2 Memory Connector

Pin	I/O	Signal	Pin	I/O	Signal
1	N/A	Ground	37	I/O	Parity Data 1
2	I/O	Data 0	38	I/O	Parity Data 3
3	I/O	Data 16	39	N/A	Ground
4	I/O	Data 1	40	O	Column Address Strobe 0
5	I/O	Data 17	41	O	Column Address Strobe 2
6	I/O	Data 2	42	O	Column Address Strobe 3
7	I/O	Data 18	43	O	Column Address Strobe 1
8	I/O	Data 3	44	O	Row Address Strobe 0
9	I/O	Data 19	45	O	Row Address Strobe 1
10	O	+5 Vdc	46	O	Block Select 1
11	O	-Column Address Strobe P	47	O	Write Enable
12	O	Address 0	48	N/A	Reserved
13	O	Address 1	49	I/O	Data 8
14	O	Address 2	50	I/O	Data 24
15	O	Address 3	51	I/O	Data 9
16	O	Address 4	52	I/O	Data 25
17	O	Address 5	53	I/O	Data 10
18	O	Address 6	54	I/O	Data 26
19	N/A	Reserved	55	I/O	Data 11
20	I/O	Data 4	56	I/O	Data 27
21	I/O	Data 20	57	I/O	Data 12
22	I/O	Data 5	58	I/O	Data 28
23	I/O	Data 21	59	O	+5 Vdc
24	I/O	Data 6	60	I/O	Data 29
25	I/O	Data 22	61	I/O	Data 13
26	I/O	Data 7	62	I/O	Data 30
27	I/O	Data 23	63	I/O	Data 14
28	O	Address 7	64	I/O	Data 31
29	O	Block Select 0	65	I/O	Data 15
30	O	+5 Vdc	66	O	Block Select 2
31	O	Address 8	67	I	Presence Detect 0
32	N/A	Reserved	68	I	Presence Detect 1
33	O	Row Address Strobe 3	69	I	Presence Detect 2
34	O	Row Address Strobe 2	70	I	Presence Detect 3
35	I/O	Parity Data 2	71	O	Block Select 3
36	I/O	Parity Data 0	72	N/A	Ground

Figure 3-6. System Board Memory Connector, Type 2

The 'presence detect' signals are used by the system to determine memory card size and memory speed. The pins are either connected to ground (G) or not connected (N). The following table shows those combinations supported by Model 50.

Type of Memory Module Package	Presence Detect Signals 0	1	2	3
1 MB Memory at 120 ns	G	N	N	N
1 MB Memory at 85 ns	G	N	N	G
2 MB Memory at 85 ns	N	G	N	G

Figure 3-7. Presence Detect Encoding

Real-Time Clock/Complementary Metal-Oxide Semiconductor RAM

The real-time clock/complementary metal-oxide semiconductor RAM (RT/CMOS) chip contains the real-time clock and 64 bytes of CMOS RAM. The internal clock circuitry uses 14 bytes of this memory, and the rest is allocated to configuration and system status information.

A 6-Vdc lithium battery maintains voltage to the RT/CMOS RAM when the power supply is not in operation.

The system cover can be locked to prevent battery removal and loss of password and configuration information.

The following figure shows the RT/CMOS RAM bytes and their addresses.

Address (Hex)	RT/CMOS RAM Bytes
000 - 00D	Real-Time Clock Bytes
00E	Diagnostic Status Byte
00F	Shutdown Status Byte
010	Diskette Drive Type Byte
011	First Fixed Disk Drive Type Byte
012	Second Fixed Disk Drive Type Byte
013	Reserved
014	Equipment Byte
015 - 016	Low and High Base Memory Bytes
017 - 018	Low and High Expansion Memory Bytes
019 - 031	Reserved
032 - 033	Configuration CRC Bytes
034 - 036	Reserved
037	Date Century Byte
038 - 03F	Reserved

Figure 3-8. RT/CMOS RAM Address Map

RT/CMOS Address Register and NMI Mask (Hex 0070)

This register is used in conjunction with the port at hex 0071 to read and write the RT/CMOS RAM bytes.

Bit	Function
7	NMI Mask
6	Reserved
5 - 0	RT/CMOS RAM Address

Figure 3-9. RT/CMOS Address Register and NMI Mask (Hex 0070)

Warning: The operation following a write to hex 0070 should access port hex 0071; otherwise intermittent malfunctions and unreliable operation of the RT/CMOS RAM can occur.

Bit 7 When this bit is set to 1, the NMI is masked off (the NMI is disabled). This bit is set to 1 by a power-on reset and is a write-only bit.

Bit 6 Reserved.

Bits 5 - 0 These bits are used to select RT/CMOS RAM addresses.

RT/CMOS Data Register (Hex 0071)

This port is used in conjunction with the address register at hex 0070 to read and write the RT/CMOS RAM bytes.

Bit	Function
7 - 0	RT/CMOS Data

Figure 3-10. RT/CMOS Data Register (Hex 0071)

RT/CMOS RAM I/O Operations

During I/O operations to the RT/CMOS RAM addresses, interrupts should be masked to prevent other interrupt service routines from changing the CMOS address register before data is read or written. After I/O operations, the RT/CMOS and NMI Mask register (hex 0070) should be left pointing to Status Register D (hex 00D).

Warning: The operation following a write to hex 0070 should access hex 0071; otherwise intermittent malfunctions and unreliable operation of the RT/CMOS RAM can occur.

The following steps are required to perform I/O operations to the RT/CMOS RAM addresses:

1. Write the RT/CMOS RAM address to the RT/CMOS and NMI Mask register (hex 0070).
2. Write the data to address hex 0071.

Reading RT/CMOS RAM requires the following steps:

1. Write the RT/CMOS RAM address to the RT/CMOS and NMI Mask register (hex 0070).
2. Read the data from address hex 0071.

Real-Time Clock Bytes (Hex 000-00D)

Bit definitions and addresses for the real-time clock bytes are shown in the following figure.

Address (Hex)	Function	Byte Number
000	Seconds	0
001	Second Alarm	1
002	Minutes	2
003	Minute Alarm	3
004	Hours	4
005	Hour Alarm	5
006	Day of Week	6
007	Date of Month	7
008	Month	8
009	Year	9
00A	Status Register A	10
00B	Status Register B	11
00C	Status Register C	12
00D	Status Register D	13

Figure 3-11. Real-Time Clock Bytes

Note: The Setup program initializes status registers A, B, C, and D when the time and date are set. Interrupt hex 1A is the BIOS interface to read and set the time and date and it initializes the register the same way as the Setup program.

Status Register A (Hex 00A)

Bit	Function
7	Update in Progress
6 - 4	22-Stage Divider
3 - 0	Rate Selection Bits

Figure 3-12. Status Register A

Bit 7 When set to 1, this bit indicates the time-update cycle is in progress. When set to 0, it indicates the current date and time can be read.

Bits 6 - 4 These three divider-selection bits identify which time-base frequency is being used. The system initializes these bits to binary 010, which selects a 32.768 kHz time-base. This is the only value supported by the system for proper time-keeping.

Bits 3 - 0 These bits allow the selection of a divider output frequency. The system initializes the rate selection bits to a binary 0110, which selects a 1.024 kHz square-wave output frequency and a 976.562-microsecond periodic interrupt rate.

Status Register B (Hex 00B)

Bit	Function
7	Set
6	Periodic Interrupt Enable
5	Alarm Interrupt Enable
4	Update-Ended Interrupt Enabled
3	Square Wave Enabled
2	Date Mode
1	24-Hour Mode
0	Daylight Savings Enabled

Figure 3-13. Status Register B

Bit 7 When set to 0, this bit updates the cycle, normally by advancing the counts at a rate of one per second. When set to 1, this bit immediately ends any update cycle in progress, and the program can initialize the 14 time bytes without any further updates occurring until this bit is set to 0.

Bit 6 This bit is a read/write bit that allows an interrupt to occur at a rate specified by the rate and divider bits in Status Register A. When set to 1, this bit enables the interrupt. The system initializes this bit to 0.

Bit 5 When set to 1, this bit enables the alarm interrupt. The system initializes this bit to 0.

Bit 4 When set to 1, this bit enables the update-ended interrupt. The system initializes this bit to 0.

Bit 3 When set to 1, this bit enables the square-wave frequency as set by the rate-selection bits in Status Register A. The system initializes this bit to 0.

Bit 2 This bit indicates if the time-and-date calendar updates use binary or binary-coded-decimal (BCD) formats. When set to 1, this bit indicates a binary format. The system initializes this bit to 0.

Bit 1 This bit establishes if the hours byte is in the 24-hour or 12-hour mode. When set to 1, this bit indicates the 24-hour mode. The system initializes this bit to 1.

Bit 0 When set to 1, this bit enables the daylight savings time mode. When set to 0, it disables the mode, and the clock reverts to standard time. The system initializes this bit to 0.

Status Register C (Hex 00C)

Bit	Function
7	Interrupt Request Flag
6	Periodic Interrupt Flag
5	Alarm Interrupt Flag
4	Update-Ended Interrupt Flag
3 - 0	Reserved

Figure 3-14. Status Register C

Note: Interrupts are enabled by bits 6, 5, and 4 in Status Register B.

Bit 7 This bit is used in conjunction with bits 6, 5, and 4. When set to 1, this bit indicates that an interrupt has occurred; bits 6, 5, and 4 indicate the type of interrupt.

Bit 6 When set to 1, this bit indicates that a periodic interrupt occurred.

Bit 5 When set to 1, this bit indicates that an alarm interrupt occurred.

Bit 4 When set to 1, this bit indicates that an update-ended interrupt occurred.

Bits 3 - 0 Reserved.

Status Register D (Hex 00D)

Bit	Function
7	Valid RAM
6 - 0	Reserved

Figure 3-15. Status Register D

Bit 7 This read-only bit monitors the power-sense pin. A low state of this pin indicates a loss of power to the real-time clock (dead battery). When set to 1, this bit indicates that the real-time clock has power. When set to 0, it indicates that the real-time clock has lost power.

Bits 6 - 0 Reserved.

CMOS RAM Configuration

The following shows the bit definitions for the CMOS RAM configuration bytes.

Diagnostic Status Byte (Hex 00E)

Bit	Function
7	Real-Time Clock Chip Power
6	Configuration Record and Checksum Status
5	Incorrect Configuration
4	Memory Size Miscompare
3	Fixed Disk Controller/Drive C Initialization Status
2	Time Status Indicator
1	Adapter Configuration Miscompare
0	Adapter ID Time-Out

Figure 3-16. Diagnostic Status Byte

Bit 7 When set to 1, this bit indicates the real-time clock chip lost power.

Bit 6 When this bit is set to 1, the checksum is incorrect.

Bit 5 This is a check, at power-on time, of the Equipment byte. When set to 1, the configuration information is incorrect. Power-on checks require that at least one diskette drive be installed (bit 0 of the Equipment byte, hex 014, is set to 1).

Bit 4 When set to 1, this bit indicates the power-on check determined that the memory size is not the same as in the configuration record.

Bit 3 When set to 1, this bit indicates that the controller or drive C failed initialization, which prevents the system from attempting a power-on reset.

Bit 2 When set to 0, this bit indicates the time is valid. When set to 1, this bit indicates the time is invalid.

Bit 1 This bit indicates if the installed adapters match the configuration information. When this bit is set to 1, the adapters do not match the configuration information.

Bit 0 When set to 1, this bit indicates a time-out occurred while an adapter ID was being read.

Shutdown Status Byte (Hex 00F): This byte is defined by the power-on diagnostic programs.

Diskette Drive Type Byte (Hex 010): This byte indicates the type of diskette drive installed.

Bit	Function
7 - 4	First Diskette Drive Type
3 - 0	Second Diskette Drive Type

Figure 3-17. Diskette Drive Type Byte

Bits 7 - 4 These bits indicate the first diskette drive type, as shown in the following figure.

Bits 7 6 5 4	Function
0 0 0 0	No drive present
0 0 0 1	Double-sided diskette drive (48 tracks per inch, 360KB)
0 0 1 1	High-capacity diskette drive (720KB)
0 1 0 0	High-density diskette drive (1.44MB)

Note: All combinations that are not shown are reserved.

Figure 3-18. Diskette Drive Type Byte (Bits 7 - 4)

Bits 3 - 0 These bits indicate the second diskette drive type, as shown in the following figure.

Bits 3 2 1 0	Function
0 0 0 0	No drive present
0 0 0 1	Double-sided diskette drive (48 tracks per inch, 360KB)
0 0 1 1	High-capacity diskette drive (720KB)
0 1 0 0	High-density diskette drive (1.44MB)

Note: All combinations that are not shown are reserved.

Figure 3-19. Diskette Drive Type Byte (Bits 3 - 0)

First Fixed Disk Drive Type Byte (Hex 011): This byte defines the type of the first fixed disk drive (drive C). Hex 00 indicates that a fixed disk drive is *not* installed.

Second Fixed Disk Drive Type Byte (Hex 012): This byte defines the type of the second fixed disk drive (drive D). Hex 00 indicates that a fixed disk drive is *not* installed.

Note: For more information about fixed disk drive types, refer to the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference.*

Reserved Byte (Hex 013): This byte is reserved.

Equipment Byte (Hex 014): The equipment byte defines the basic equipment in the system for the power-on diagnostic tests.

Bit	Function
7, 6	Number of Diskette Drives
5, 4	Display Operating Mode
3, 2	Reserved
1	Math Coprocessor Presence
0	Diskette Drive Presence

Figure 3-20. Equipment Byte

Bits 7, 6 These bits indicate the number of diskette drives installed, as shown in the following figure.

Bits 7 6	Number of Diskette Drives
0 0	One Drive
0 1	Two Drives
1 0	Reserved
1 1	Reserved

Figure 3-21. Equipment Byte (Bits 7, 6)

Bits 5, 4 These bits indicate the operating mode of the display attached to the video port, as shown in the following figure.

Bits 5 4	Display Operating Mode
0 0	Reserved
0 1	40-Column Mode
1 0	80-Column Mode
1 1	Monochrome Mode

Figure 3-22. Equipment Byte (Bits 5, 4)

Bits 3, 2 Reserved.

Bit 1 When set to 1, this bit indicates that a math coprocessor is installed.

Bit 0 When set to 1, this bit indicates that a diskette drive is installed.

Low and High Base Memory Bytes (Hex 015 and 016): These bytes define the amount of memory below the 640KB address space.

The value from these bytes represents the number of 1KB blocks of base memory. For example, hex 0280 is equal to 640KB. The low byte is hex 15; the high byte is hex 16.

Low and High Expansion Memory Bytes (Hex 017 and 018): These bytes define the amount of memory above the 1MB address space.

The hexadecimal values in these bytes represent the number of 1KB blocks of expansion memory. For example, hex 0800 is equal to 2048KB. The low byte is hex 17; the high byte is hex 18.

Reserved Bytes (Hex 019 through 031): These bytes are reserved.

Configuration CRC Bytes (Hex 032 and 033): These bytes contain the cyclic-redundancy-check data for bytes hex 010 through hex 031 of the 64-byte CMOS RAM. The low byte is hex 33; the high byte is hex 32.

Reserved Bytes (Hex 034 through 036): These bytes are reserved.

Date Century Byte (Hex 037): Bits 7 through 0 of this byte contain the BCD value for the century. Refer to the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference* for information about reading and setting this byte.

Reserved Bytes (Hex 038 through 03F): These bytes are reserved.

Miscellaneous System Functions

Nonmaskable Interrupt

The nonmaskable interrupt (NMI) signals the system microprocessor that a parity error, a channel check, a system channel time-out, or a system Watchdog time-out has occurred. The NMI stops all arbitration on the bus until bit 6 of the Arbitration register (I/O address hex 0090) is set to 0. This can result in lost data or an overrun error on some I/O devices. The NMI masks all other interrupts and the IRET instruction restores the interrupt flag to the state it was in prior to the interrupt. A system reset causes a reset of the NMI.

Nonmaskable interrupt requests from system board parity and channel check are subject to mask control with the NMI mask bit in the RT/CMOS Address register. The Watchdog Timer and system channel time-out are not masked by this bit. (See "RT/CMOS RAM I/O Operations" on page 3-14). The power-on default of the NMI mask is 1 (NMI disabled). Prior to enabling the NMI after a power-on reset, the parity check and channel check state are initialized by the POST.

Warning: The operation following a write to hex 0070 should access port hex 0071; otherwise intermittent malfunctions and unreliable operation of the RT/CMOS RAM can occur.

System Control Port B (Hex 0061)

Bit definitions for the read and write functions of this port are shown in the following figures.

Bit	Function
7	Reset Timer 0 Output Latch (IRQ0)
6 - 4	Reserved
3	Enable Channel Check
2	Enable Parity Check
1	Speaker Data Enable
0	Timer 2 Gate to Speaker

Figure 3-23. System Control Port B (Write)

Bit	Function
7	Parity Check
6	Channel Check
5	Timer 2 Output
4	Toggle with Each Refresh Request
3	Enable Channel Check
2	Enable Parity Check
1	Speaker Data Enable
0	Timer 2 Gate to Speaker

Figure 3-24. System Control Port B (Read)

Bit 7 Setting this bit to 1 during a write operation, resets IRQ 0. When this port is read and this bit is set to 1, a parity check has occurred.

Bit 6 When this port is read and this bit is set to 1, a channel check has occurred.

Bit 5 When this port is read, this bit indicates the condition of the timer/counter 2 'output' signal.

Bit 4 When this port is being read, this bit toggles for each refresh request.

Bit 3 Setting this bit to 0 enables channel check. It is set to 1 during a power-on reset.

Bit 2 Setting this bit to 0 enables parity checking. This bit is set to 1 during a power-on reset.

Bit 1 Setting this bit to 1 enables speaker data. This bit is set to 0 during a power-on reset.

Bit 0 Setting this bit to 1 enables the timer 2 gate.

System Control Port A (Hex 0092)

Bit	Function
7, 6	Fixed Disk Activity Light
5	Reserved
4	Watchdog Timer Status
3	Security Lock Latch
2	Reserved
1	Alternate Gate A20
0	Alternate Hot Reset

Figure 3-25. System Control Port A (Hex 0092)

Bits 7, 6 These bits control the fixed-disk activity light. Setting either bit to 1 turns the fixed-disk activity light on. Setting both bits to 0 turns the light off. The power-on reset condition of each bit is 0.

Bit 5 Reserved.

Bit 4 This read-only bit indicates the Watchdog Timer status. When this bit is set to 1, a Watchdog time-out has occurred. For more information about the Watchdog Timer, refer to the *Hardware Interface Technical Reference.*

Bit 3 This bit provides the security lock for the secured area of RT/CMOS. Setting this bit to 1 electrically locks the 8-byte, power-on password. Once this bit is set by POST, it can only be cleared by turning the system off.

Bit 2 Reserved.

Bit 1 This bit is used to enable the 'address 20' signal (A20) when the microprocessor is in the real address mode. When this bit is set to 0, A20 cannot be used in real mode addressing. This bit is set to 0 during a system reset.

Bit 0 This bit provides an alternate method of resetting the system microprocessor. This alternate method supports operating systems requiring faster operation than was provided on the IBM Personal Computer AT®. Resetting the system microprocessor is used to switch the microprocessor from the protected mode to the real address mode. The alternate reset takes 13.4 microseconds.

This bit is set to 0 either by a system reset or a Write operation. When a Write operation changes this bit from 0 to 1, the alternate reset pin is pulsed high for 100 to 125 nanoseconds. The reset occurs after a minimum delay of 6.72 microseconds. While the reset is occurring, the latch remains set so that POST can read this bit. If the bit is 0, POST assumes the system was just powered on. If the bit is 1, POST assumes a switch from the protected mode to the real mode has taken place.

IBM Personal Computer AT is a registered trademark of the International Business Machines Corporation.

Power-On Password

RT/CMOS RAM has 8 bytes reserved for the power-on password and its check character. The 8 bytes are initialized to hex 00. The microprocessor can only access these bytes during power-on self-test (POST). After POST is completed, if a power-on password is installed, the password bytes are locked and cannot be accessed by a program. A power-on password can be from 1 to 7 characters.

During power-on password installation, the password (1 to 7 keyboard scan codes), is stored in the security space.

Power-on password installation is a function of a program contained on the Reference diskette. Once the power-on password utility has been installed, the password can be changed only during the POST. When the new power-on password is installed, changed, or removed, the password is not visible on the display.

The system unit cover can be physically locked to prevent unauthorized access to the battery. This helps prevent unauthorized battery removal and loss of power-on password and configuration information.

For information about the keyboard password, see the "Keyboard and Auxiliary Device Controller" section in the *Hardware Interface Technical Reference*.

Type 2 Serial Port Controller

The Model 50 does not support FIFO mode operations even if a Type 2 serial controller is installed. Do not enable FIFO operations if the model/submodel byte is hex FC/04.

The serial controller on some Type 2 system boards does not respond the same as other Personal System/2 products. After the initial transmitter-holding-register-empty interrupt, rewriting bit 1 of the Interrupt Enable register as a 1 will not reissue an interrupt. The interrupt is generated only when the Transmitter Holding register becomes empty.

Hardware Compatibility

The Model 50 maintains many of the interfaces used by the IBM Personal Computer AT. In most cases, command and status organization of these interfaces is maintained.

The functional interfaces for the Personal System/2 products are compatible with the following interfaces:

- The Intel[2] 8259 interrupt controllers (without edge triggering).
- The Intel 8253 timers driven from 1.193 MHz (timer 0 and 2 only).
- The Intel 8237 DMA controller-address/transfer counters, page registers and status fields only. The Command and Request registers are not supported. The rotate and mask functions are not supported. The Mode register is partially supported.
- Generally compatible with the NS16450 serial communications controller.
- The Intel 8088, 8086, and 80286 microprocessors.
- The Intel 8272 diskette drive controller.
- The Motorola[3] MC146818 Time of Day Clock command and status (CMOS reorganized).
- The Intel 8042 keyboard port at address hex 0060.

---

2 Intel is a trademark of the Intel Corporation.

3 Motorola is a trademark of Motorola, Inc.

- Display modes supported by the IBM Monochrome Display and Printer Adapter, the IBM Color/Graphics Monitor Adapter, and the IBM Enhanced Graphics Adapter.
- The parallel printer ports (Parallel 1, Parallel 2, and Parallel 3) in compatibility mode.
- Generally compatible with the Intel 80287 and 8087 math coprocessors.

Index

A

activity light, fixed disk 3-26
adapter ID (0000) 3-3
adapter setup 2-4
address maps
  memory 3-8
  POS 2-5
  system 1-6
address space, POS 2-5
alternate reset 3-27
arbiter programming 3-5
arbiter refresh 3-4
arbitration level 3-4
arbitration register 3-5
arbitration, central control point 3-5
A20 gate 3-26

B

base memory bytes 3-23
battery 3-12
block diagram, system board 1-5
bus contention 2-3
bus priority assignments 3-4

C

card selected feedback 2-6
-CD SFDBK 2-6
central arbitration control point 3-5
channel-check latch 3-25
CMOS RAM 3-12
configuration CRC bytes 3-23
configuration, RT/CMOS 3-19
connectors, memory 3-9
control port B 3-24

D

date century byte 3-23
description
  programmable option select 2-3
diagnostic status byte (RT/CMOS) 3-19
diskette drive enable 2-9
diskette drive type byte 3-20
DMA channels 3-4

E

electrical requirements 1-9
enable I/O functions 2-8
enable NMI 3-13
enable/setup register 2-7
enable, channel check 3-25
equipment byte 3-22
extended arbitration 3-6

F

features, system 1-4
feedback, card selected 2-6
first fixed disk type byte 3-21
fixed disk drive initialization 3-19
fixed-disk activity light 3-26

G

gate A20 3-26

H

heat output 1-9

I

I/O address map 1-6
I/O function byte 2-8
Identification, adapter 3-3
interrupt 0 reset 3-25
interrupts

K

keyboard (electrical) 1-9

M

map, memory 3-8
math coprocessor 1-4
memory
  connectors 3-9
  control register 2-9
  map 3-8
  memory module packages 3-7
  performance 1-7
  random access 3-7
  read only 3-7
  refresh 1-7
memory refresh 3-4
memory size miscompare 3-19
microprocessor
  alternate reset 3-27
  real address mode 3-26
miscellaneous ports 3-24
mode switch compatibility 3-27
model byte 1-3
modules, memory 3-7

N

NMI mask 3-13
NMI service 3-4
NMI signal 3-4
nonmaskable interrupt (NMI) 3-24

P

parallel port 2-8
parity check enable 3-25
password, power-on 3-28
performance, system 1-7
pointing device (electrical) 1-9
port B 3-24
POS
  See programmable option select
POS registers 2-8
power specifications 1-9
programmable option select 2-3
  enable/setup register 2-7
  registers 2-5
  system board I/O byte 2-8
  system board setup 2-6
  system configuration
    utilities 2-3
programming, arbiter 3-5

R

RAM enable 2-9
random access memory (RAM) 3-7
read-only memory 3-7
real mode addressing 3-26
real-time clock bytes 3-15
refresh 3-4, 3-7
refresh request 3-25
registers
  card selected feedback 2-6
  enable/setup 2-7
  memory control 2-9
  miscellaneous system 3-24
  port B 3-24
  POS 2-5, 2-8
  RT/CMOS and NMI mask 3-13

registers *(continued)*
RT/CMOS status registers 3-16
reset, alternate 3-27
RT/CMOS address register 3-13
RT/CMOS RAM 3-12
password 3-26, 3-28
secured area 3-26

S

second fixed disk type byte 3-21
serial communications controller 3-29
serial port 2-8
serial port controller 3-29
setup
system board 2-6
system board I/O byte 2-8
warning 2-3
shutdown status byte 3-20
size 1-9
speaker data enable 3-25
specifications 1-7
status registers (RT/CMOS) 3-16
status, watchdog 3-26
switch to real mode 3-27
system address map 1-6
system board
block diagram 1-5
features 1-4
I/O address map 1-6
I/O byte 2-8
memory 3-7
memory connectors 3-9
memory enable 2-9
memory map 3-8
setup 2-4
system board setup 2-6
system configuration utilities 2-3
system control port A 3-26
system ports, miscellaneous 3-24
system reset 3-24
system unit specifications 1-7

T

time status indicator 3-19
timer status, watchdog 3-26
type 1 memory connector 3-9
type 2 memory connector 3-10

U

utilities, configuration 2-3

V

valid RAM bit 3-18
video subsystem setup 2-4

W

warning, POS 2-3
watchdog status 3-26
weight 1-9

Notes:

Personal System/2
Model 57
Technical Reference

Note

Before using this information and the product it supports, be sure to read the general information under "Notices" on page vii.

Second Edition (March 1992)

The following paragraph does not apply to the United Kingdom or any country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.

It is possible that this publication may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

Requests for technical information about IBM products should be made to your IBM Authorized Dealer or your IBM Marketing Representative.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, Purchase, NY 10577.

© Copyright International Business Machines Corporation 1991, 1992. All rights reserved.
Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Figures v

Notices vii
Trademarks and Service Marks vii

Preface ix

Section 1. System Overview 1-1
Description 1-3
| System Board Devices and System Features 1-4
System Board I/O Address Map 1-10
Micro Channel Compatibility 1-12
Specifications 1-13
Power Supply 1-16
Outputs 1-17
Output Protection 1-18
Voltage Sequencing 1-18
Power Supply Connectors 1-18

Section 2. Programmable Option Select 2-1
Description 2-3
POS Address Map 2-5
Card Selected Feedback Register (Hex 0091) 2-6
System Board Setup 2-6
System Board Enable/Setup Register (Hex 0094) 2-7
System Board Group 1 POS Register 2 (Hex 0102) 2-8
System Board Group 1 POS Register 3 (Hex 0103) 2-10
System Board Group 1 POS Register 4 (Hex 0104) 2-11
System Board Group 1 POS Register 5 (Hex 0105) 2-11
System Board Group 2 POS Register 2 (Hex 0102) 2-12
System Board Group 2 POS Register 3 (Hex 0103) 2-12
System Board Group 3 POS Register 4 (Hex 0104) 2-13
System Board SCSI Controller POS Register 2 (Hex 0102) . . 2-13
System Board SCSI Controller POS Register 3 (Hex 0103) . . 2-14
Adapter Enable/Setup Register (Hex 0096) 2-15

Section 3. System Board 3-1
Description 3-3
| Microprocessor—Type 1 Systems 3-3
| Microprocessor—Type 2 Systems 3-3
| Programming Model 3-4
| Instructions 3-4

© Copyright IBM Corp. 1991, 1992

| Cache Operation 3-4
| Cacheable Address Space 3-5
Math Coprocessor 3-6
Micro Channel Implementation 3-6
Central Arbiter 3-6
Bus Adapter 3-9
Diskette Drive Connector 3-10
System Board Memory Connectors 3-11
SCSI-Device Connector 3-12
Memory 3-13
ROM Subsystem 3-14
RAM Subsystem 3-14
Memory Interleaving Option 3-15
Memory-Presence Detect Signals 3-15
System Memory Map 3-16
Memory Controller Registers 3-16
RT/CMOS RAM 3-18
NVRAM 3-27
Miscellaneous System Functions and Ports 3-28
Nonmaskable Interrupt 3-28
System Control Port B (Hex 0061) 3-28
System Control Port A (Hex 0092) 3-29
Power-On Password 3-31
Selectable Boot Sequence 3-31
Hardware Compatibility 3-32
Error Codes 3-33

| **Appendix. 386SLC Instruction Cycle Times** A-1

Index X-1

Figures

1-1.	Model and Submodel Bytes	1-3
1-2.	Model 57 SX System Board Devices and System Features	1-4
1-3.	Model 57 SLC System Board Devices and System Features	1-6
1-4.	Model M57 SLC System Board Devices and System Features	1-8
1-5.	System Board I/O Address Map	1-10
1-6.	Performance Specifications	1-13
1-7.	Physical Specifications—Model 57 SX and Model 57 SLC	1-14
1-8.	Physical Specifications—Model M57 SLC	1-15
1-9.	Acoustical Readings	1-16
1-10.	Component Maximum Current	1-17
1-11.	Channel Load Current	1-17
1-12.	Voltage Assignments for the Internal Power-Supply Connectors	1-18
2-1.	POS I/O Address Map	2-5
2-2.	Card Selected Feedback Register (Hex 0091)	2-6
2-3.	Devices Supported in Setup Mode	2-7
2-4.	System Board Enable/Setup Register (Hex 0094)	2-7
2-5.	System Board POS Register 2 (Hex 0102)	2-9
2-6.	Parallel Port Select Bits	2-9
2-7.	System Board Group 1 POS Register 3 (Hex 0103)—Write	2-10
2-8.	System Board POS Group 1 Register 3 (Hex 0103)—Read Index 00	2-10
2-9.	System Board POS Group 1 Register 3 (Hex 0103)—Read Index 04	2-11
2-10.	System Board Group 1 POS Register 4 (Hex 0104)	2-11
2-11.	System Board Group 1 POS Register 5 (Hex 0105)	2-11
2-12.	System Board Group 2 POS Register 2 (Hex 0102)	2-12
2-13.	System Board Group 2 POS Register 3 (Hex 0103)	2-12
2-14.	Serial Port Addresses	2-12
2-15.	System Board Group 3 POS Register 4 (Hex 0104)	2-13
2-16.	System Board SCSI Controller POS Register 2 (Hex 0102)	2-13
2-17.	System Board SCSI Controller POS Register 3 (Hex 0103)	2-14
2-18.	System Board SCSI Controller ID	2-14
2-19.	Adapter Enable/Setup Register (Hex 0096)	2-15
3-1.	Arbitration Bus Priority Assignments	3-7

© Copyright IBM Corp. 1991, 1992

3-2. Arbitration Register (Hex 0090)—Write 3-8
3-3. Arbitration Register (Hex 0090)—Read 3-8
3-4. Diskette Drive Connector 3-10
3-5. System Board Memory Connectors 3-11
3-6. SCSI Device Cable Internal Connector Pin Assignments 3-12
3-7. System Board SCSI Device External Connector 3-13
3-8. Memory Interleaving Configurations 3-15
3-9. Presence-Detect Bits 3-15
| 3-10. System Memory Map 3-16
3-11. Memory-Fault RAS Capture Register (I/O Port Hex 00E2) 3-17
3-12. Fixed RAS Signals 3-17
3-13. Memory-Controller-Fault Address Register (I/O Port Hex 00E4) 3-18
3-14. Memory-Controller-Fault Address Registers (I/O Port Hex 00E5) 3-18
3-15. RT/CMOS RAM Address Map 3-19
3-16. RT/CMOS Address and NMI Mask Register (Hex 0070) 3-19
3-17. RT/CMOS Data Register (Hex 0071) 3-20
3-18. Real-Time Clock Bytes (Hex 000—00D) 3-21
3-19. Status Register A (Hex 00A) 3-21
3-20. Status Register B (Hex 00B) 3-22
3-21. Status Register C (Hex 00C) 3-23
3-22. Status Register D (Hex 00D) 3-23
3-23. Diagnostic Status Byte (Hex 00E) 3-24
3-24. Diskette Drive Type Byte (Hex 010) 3-25
3-25. Diskette Drive Type Byte (Bits 7—4) 3-25
3-26. Diskette Drive Type Byte (Bits 3—0) 3-25
3-27. Equipment Byte (Hex 014) 3-26
3-28. Installed Diskette Drive Bits 3-26
3-29. Display Operating-Mode Bits 3-26
3-30. System Control Port B (Hex 0061)—Write 3-28
3-31. System Control Port B (Hex 0061)—Read 3-29
3-32. System Control Port A (Hex 0092) 3-29
| 3-33. POST Error Messages 3-34
| A-1. Instruction Cycle Times A-1

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights or other legally protectible rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, programs, or services, except those expressly designated by IBM, are the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, Purchase, NY 10577.

Trademarks and Service Marks

The following terms, denoted by an asterisk (*) in this publication, are trademarks of the IBM Corporation in the United States and/or other countries:

IBM
Micro Channel
Personal Computer AT
Personal System/2
PS/2
XGA

The following terms, denoted by a double asterisk (**) in this publication, are trademarks of other companies as follows:

Intel	Intel Corporation
Motorola	Motorola, Incorporated

© Copyright IBM Corp. 1991, 1992

Notes:

Preface

The Technical Reference library is intended for those who develop hardware and software products for IBM Personal Computers and IBM Personal System/2 products. Users should understand computer architecture and programming concepts.

This manual consists of the following sections:

Section 1, "System Overview," describes the system, features, and specifications.

Section 2, "Programmable Option Select," describes the registers used for configuration.

Section 3, "System Board," describes the system-specific hardware implementations of the Micro Channel architecture.

This technical reference should be used with the following publications. These publications contain additional information on many of the subjects discussed in this technical reference.

IBM Personal System/2 Hardware Interface Technical Reference—Architectures

IBM Personal System/2 Hardware Interface Technical Reference—Common Interfaces

IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference.

Information about diskette drives, fixed disk drives, adapters, and external options are in separate option technical references.

Warning: The term *reserved* describes certain signals, bits, and registers that should not be changed. Use of reserved areas can cause compatibility problems, loss of data, or permanent damage to the hardware. When the contents of a register are changed, the state of the reserved bits must be preserved. When possible, read the register first and change only the bits that must be changed.

© Copyright IBM Corp. 1991, 1992

Notes:

Section 1. System Overview

Description 1-3
System Board Devices and System Features 1-4
System Board I/O Address Map 1-10
Micro Channel Compatibility 1-12
Specifications 1-13
Power Supply 1-16
  Outputs 1-17
  Output Protection 1-18
  Voltage Sequencing 1-18
  Power Supply Connectors 1-18

© Copyright IBM Corp. 1991, 1992

Notes:

Description

The IBM* Personal System/2* (PS/2*) Model 57 systems are self-contained desktop computer systems with a keyboard. The systems can support a maximum of four internal drives. Three of these drives can be either removable-media direct access storage devices (DASD) (diskette, tape, optical), or three can be nonremovable-media DASD devices.

Programs can identify the system board by reading the model and submodel bytes. Interrupt 15H, function code (AH) = C0H, returns the model byte, submodel byte, and BIOS revision code. The microprocessor on Type 2 systems contains an internal cache and cache controller.

The following figure lists the model and submodel bytes, system-clock speed, and system board types for all Model 57 system boards.

Model Name	Model Byte (Hex)	Submodel Byte (Hex)	System Board Type	System Clock
Model 57 SX	F8	26	Type 1	20 MHz
Model 57 SLC	F8	25	Type 2	20 MHz
Model M57 SLC	F8	25	Type 2	20 MHz

Figure 1-1. Model and Submodel Bytes

Refer to the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference* for a listing of other systems.

* Trademark of the IBM Corporation.

System Board Devices and System Features

The following figure lists the system board devices and system features for the Model 57 SX. The *IBM Personal System/2 Hardware Interface Technical Reference — Common Interfaces* describes devices common to PS/2 products by type number.

Device	Type	Features
Microprocessor	—	Type 1 systems - 80386SX - 24-bit address, 16-bit data interface
System Timers	1	Channel 0-System timer Channel 2-Tone generation for speaker Channel 3-Watchdog timer
ROM Subsystem	—	128KB logical ROM (KB equals 1024 bytes) 256KB physical ROM
RAM Subsystem	—	Three connectors (1MB — 16MB) (MB equals 1048576 bytes)
CMOS RAM Subsystem	—	64-byte CMOS RAM with real-time clock/calendar 8KB CMOS RAM extension Battery backup
Audio Subsystem	1	Driven by - System-timer channel 2 - The 'audio sum node' signal
DMA Controller	1	Eight independent DMA channels Single or burst transfers with read verification
Video Subsystem	1	16-bit VGA - 132-column text mode - Mainframe Interactive (MFI) support

Figure 1-2 (Part 1 of 2). Model 57 SX System Board Devices and System Features

Device	Type	Features
Interrupt Controller	1	16 levels of system interrupts Interrupts are level sensitive
Keyboard/Auxiliary Device Controller	2	Keyboard connector Auxiliary device connector Password security
Diskette Drive Controller	2	Supports - 3.5-in. diskette drive (1.44MB and 2.88MB) - 5.25-in. diskette drive (360KB and 1.2MB)
Serial Controller Port	4	One RS-232C interface - One 25-pin connector Programmable as serial ports 1 – 16 DMA, FIFO, and character modes
Parallel Controller Port	3	Programmable as parallel port 1 – 4 Supports bidirectional input and output Supports DMA operation
SCSI Controller	1	Supports up to three internal SCSI devices Supports up to seven total SCSI devices
Micro Channel* Bus Adapter	—	Five connectors for Type 3 adapters - One 16-bit connector with an auxiliary video extension - Four 16-bit connectors
Math Coprocessor Socket	—	Supports 80387SX math coprocessor option with the same clock speed as the system microprocessor

Figure 1-2 (Part 2 of 2). Model 57 SX System Board Devices and System Features

* Trademark of the IBM Corporation.

The following figure lists the system board devices and system features for the Model 57 SLC. The *IBM Personal System/2 Hardware Interface Technical Reference—Common Interfaces* describes devices common to PS/2 products by type number.

Device	Type	Features
Microprocessor	—	Type 2 systems - IBM 386SLC - 24-bit address, 16-bit data interface - 8KB instruction and data cache with parity
System Timers	1	Channel 0-System timer Channel 2-Tone generation for speaker Channel 3-Watchdog timer
ROM Subsystem	—	128KB logical ROM (KB equals 1024 bytes) 256KB physical ROM
RAM Subsystem	—	Three connectors (1MB—16MB) (MB equals 1048576 bytes)
CMOS RAM Subsystem	—	64-byte CMOS RAM with real-time clock/calendar 8KB CMOS RAM extension Battery backup
Audio Subsystem	1	Driven by - System-timer channel 2 - The 'audio sum node' signal
DMA Controller	1	Eight independent DMA channels Single or burst transfers with read verification
Video Subsystem	1	16-bit VGA - 132-column text mode - Mainframe interactive (MFI) support

Figure 1-3 (Part 1 of 2). Model 57 SLC System Board Devices and System Features

Device	Type	Features
Interrupt Controller	1	16 levels of system interrupts Interrupts are level sensitive
Keyboard/Auxiliary Device Controller	2	Keyboard connector Auxiliary device connector Password security
Diskette Drive Controller	2	Supports - 3.5-in. diskette drive (1.44MB and 2.88MB) - 5.25-in. diskette drive (360KB and 1.2MB)
Serial Controller Port	4	One RS-232C interface - One 25-pin connector Programmable as serial ports 1–16 DMA, FIFO, and character modes
Parallel Controller Port	3	Programmable as parallel port 1–4 Supports bidirectional input and output Supports DMA operation
SCSI Controller	1	Supports up to three internal SCSI devices Supports up to seven total SCSI devices
Micro Channel Bus Adapter	—	Five connectors for Type 3 adapters - One 16-bit connector with an auxiliary video extension - Four 16-bit connectors
Math Coprocessor Socket	—	Supports 80387SX math coprocessor option with the same clock speed as the system microprocessor

Figure 1-3 (Part 2 of 2). Model 57 SLC System Board Devices and System Features

The following figure lists the system board devices and system features for the Model M57 SLC. The *IBM Personal System/2 Hardware Interface Technical Reference—Common Interfaces* describes devices common to PS/2 products by type number.

Device	Type	Features
Microprocessor	—	Type 2 systems - IBM 386SLC - 24-bit address, 16-bit data interface - 8KB instruction and data cache with parity
System Timers	1	Channel 0-System timer Channel 2-Tone generation for speaker Channel 3-Watchdog timer
ROM Subsystem	—	128KB logical ROM (KB equals 1024 bytes) 256KB physical ROM
RAM Subsystem	—	Three connectors (1MB—16MB) (MB equals 1048576 bytes)
CMOS RAM Subsystem	—	64-byte CMOS RAM with real-time clock/calendar 8KB CMOS RAM extension Battery backup
Audio Subsystem	1	Driven by - System-timer channel 2 - The 'audio sum node' signal M-Audio Capture and Playback Adapter† Multimedia front panel - Stereo headphone output - Mono microphone input - Enhanced speaker
DMA Controller	1	Eight independent DMA channels Single or burst transfers with read verification

† For information on the M-Audio Capture and Playback Adapter, see the *M-Audio Capture and Playback Adapter Installation and Technical Reference Manual.*

Figure 1-4 (Part 1 of 2). Model M57 SLC System Board Devices and System Features

Device	Type	Features
Video Subsystem	2	16-bit Extended graphics array (XGA*) adapter - 132-column text mode
Interrupt Controller	1	16 levels of system interrupts Interrupts are level sensitive
Keyboard/Auxiliary Device Controller	2	Keyboard connector Auxiliary device connector Password security
Diskette Drive Controller	2	Supports - 3.5-in. diskette drive (1.44MB and 2.88MB) - 5.25-in. diskette drive (360KB and 1.2MB)
CD-ROM	—	CD-ROM-II† - 600MB capacity - CD-XA capability - 330 ms average access time
Serial Controller Port	4	One RS-232C interface - One 25-pin connector Programmable as serial ports 1–16 DMA, FIFO, and character modes
Parallel Controller Port	3	Programmable as parallel port 1–4 Supports bidirectional input and output Supports DMA operation
SCSI Controller	1	Supports up to three internal SCSI devices Supports up to seven total SCSI devices
Micro Channel Bus Adapter	—	Five connectors for Type 3 adapters - One 16-bit connector with an auxiliary video extension - Four 16-bit connectors
Math Coprocessor Socket	—	Supports 80387SX math coprocessor option with the same clock speed as the system microprocessor

† For information on the IBM Personal System/2 CD-ROM-II, see the *IBM Personal System/2 CD-ROM-II Technical Reference*.

Figure 1-4 (Part 2 of 2). Model M57 SLC System Board Devices and System Features

* Trademark of the IBM Corporation.

System Board I/O Address Map

The following figure is the Model 57 system board I/O address map.

Address (Hex)	Device
0000–001F	DMA controller (0–3)
0020, 0021	Interrupt controller (master)
0040, 0042–0044, 0047	System timers
0060	Keyboard, auxiliary device
0061	System Control Port B
0064	Keyboard, auxiliary device
0070, 0071	RT/CMOS and NMI mask
0074, 0075, 0076	8KB x8 NVRAM
0081–0083, 0087	DMA Page registers (0–3)
0089–008B, 008F	DMA Page registers (4–7)
0090	Central arbitration control point
0091	Card Selected Feedback register
0092	System Control Port A
0094	System Board Enable/Setup register
0096	Adapter Enable/Setup register
00A0, 00A1	Interrupt controller (slave)
00C0–00DF	DMA controller (4–7)
00E0–00E5	Memory controller
00F0–00FF	Math coprocessor
0100–0107	Programmable option select
0278–027D	Parallel Port 3
02F8–02FF	Serial Port 2
0378–037D	Parallel Port 2
03B4, 03B5, 03BA	Video subsystem
03BC–03BF	Parallel Port 1
03C0–03C5	Video subsystem
03C6–03C9	Video DAC
03CA, 03CC, 03CE, 03CF	Video subsystem
03D4, 03D5, 03DA	Video subsystem
03F0–03F7	Diskette drive controller
03F8–03FF	Serial Port 1
1278–127D	Enhanced Parallel Port 1
1378–137D	Parallel Port 4

Figure 1-5 (Part 1 of 2). System Board I/O Address Map

Address (Hex)	Device
2100–210F	Video instance 0
2110–211F	Video instance 1
2120–212F	Video instance 2
2130–213F	Video instance 3
2140–214F	Video instance 4
2150–215F	Video instance 5
2160–216F	Video instance 6
2170–217F	Video instance 7
3220–3227	Serial Port 3
3228–322F	Serial Port 4
3540–3547	SCSI controller
3A20–3A27	Serial Port 9
3A28–3A2F	Serial Port 10
3A30–3A37	Serial Port 11
3A38–3A3F	Serial Port 12
3A80–3A87	Serial Port 13
3A88–3A8F	Serial Port 14
3A90–3A97	Serial Port 15
3A98–3A9F	Serial Port 16
4220–4227	Serial Port 5
4228–422F	Serial Port 6
4620–462F	Serial port arbitration-level control
5220–5227	Serial Port 7
5228–522F	Serial Port 8
82F8–82FF	Enhanced Serial Port 2
83F8–83FF	Enhanced Serial Port 1
B220–B227	Enhanced Serial Port 3
B228–B22F	Enhanced Serial Port 4
BA20–BA27	Enhanced Serial Port 9
BA28–BA2F	Enhanced Serial Port 10
BA30–BA37	Enhanced Serial Port 11
BA38–BA3F	Enhanced Serial Port 12
BA80–BA87	Enhanced Serial Port 13
BA88–BA8F	Enhanced Serial Port 14
BA90–BA97	Enhanced Serial Port 15
BA98–BA9F	Enhanced Serial Port 16
C220–C227	Enhanced Serial Port 5
C228–C22F	Enhanced Serial Port 6
D220–D227	Enhanced Serial Port 7
D228–D22F	Enhanced Serial Port 8

Figure 1-5 (Part 2 of 2). System Board I/O Address Map

Micro Channel Compatibility

The following terms describe system limitations supporting Micro Channel.

Streaming Data

The system board supports 10-MHz streaming-data procedures involving transfers between adapters; but the devices on the system board do not *use* the streaming-data procedures. Transfers default to the basic transfer procedure.

Data Parity

The system board supports data-parity procedures involving transfers between adapters; but the devices on the system board do not *use* the data-parity procedures. Transfers default to the basic transfer procedure.

Select Feedback Return Indicator (Port Hex 91)

The system board and the processor complex support the 'selected feedback return' indicator except for the I/O devices at addresses below hex 0100. These devices are controlled by the system master only and should not be accessed by bus masters.

Synchronous Channel Check

Synchronous channel checks for errors other than Micro Channel data-parity errors are not supported in this system.

Synchronous channel checks for data-parity errors, typically issued by slaves, are the responsibility of the master that owned the channel and issued the access to the slave. However, in this system, these channel checks are detected by logic on the system board, resulting in a nonmaskable interrupt (NMI).

DMA

In this system, the DMA controller supports DMA slaves with 8-bit or 16-bit data widths.

Specifications

The following figure contains the performance specifications for all Model 57 systems.

Device	Cycle Time (ns)
Microprocessor (20 MHz – 50 ns Clock)	
Pipelined access to system board RAM†	
Memory read/write (page hit)	100
Memory read/write (page miss/bank miss)	150
Memory read/write (page miss)	200
Nonpipelined access to system board RAM†	
Memory read/write (page hit)	150
Memory read/write (page miss/bank miss)	200
Memory read/write (page miss)	250
Access to Channel	
Default transfer cycle	250
Synchronous transfer cycle	350
Access to ROM	200
Refresh Rate (typically performed every 15.1 μs)	500 (min) 800 (typical)
Bus-Master Access to System Board RAM	
Page hit	320 (min)
Page miss	370 (min)
DMA Controller (10 MHz – 100 ns Clock)	
Single transfer: 300 + (I/O access + memory access)	
Burst transfer: 300 + (I/O access + memory access)*n*‡	
System board memory access	200 – 350
Default transfer cycle	350 (min)
Synchronous transfer cycle	450 (min)

† Adapters installed in the channel should not rely on monitoring system board memory access, because channel-memory control signals might not be present during these accesses.

‡ *n* equals the number of transfers in the burst.

Note: The cycle times shown for access to system board RAM are based on 70-nanosecond memory.

Figure 1-6. Performance Specifications

The following figure describes the physical specifications of the Model 57 SX and the Model 57 SLC systems.

Specification	Measurements
Size	
Width	440 mm (17.3 in.)
Depth	394 mm (15.5 in.)
Height	168 mm (6.6 in.)
Weight	
Minimum configuration	12.0 kg (26 lb)
Maximum configuration	15.9 kg (35 lb)
Cables	
Power cable	1.8 m (6 ft)
Keyboard cable	1.8 m (6 ft)
Air Temperature	
System on	10.0 to 35.0°C (50 to 95°F) 0 – 7000 ft
System off	10.0 to 43.0°C (50 to 110°F)
Humidity	
System on	8% to 85%
System off	5% to 85%
Maximum Altitude	2133.6 m (7000 ft)
Heat Output	(300 watts) 1030 BTUs per hour
Acoustical Readings	See Figure 1-9 on page 1-16
Electrical	
Input voltage (range is switch selected; sinewave input is required)	
Low range	90 (min) – 137 (max) V ac
High range	180 (min) – 265 (max) V ac
Input frequency	50 ± 3 Hz or 60 ± 3 Hz
Maximum current draw	
Low range	4.6 A
High range	2.0 A
Input in kilovolt-ampere (kVA)	
Minimum configuration (as shipped by IBM)	0.07 kVA
Maximum configuration	0.50 kVA
Electromagnetic Compatibility	FCC Class B

Figure 1-7. Physical Specifications—Model 57 SX and Model 57 SLC

The following figure describes the physical specifications of the Model M57 SLC.

Specification	Measurements
Size	
Width	440 mm (17.3 in.)
Depth	394 mm (15.5 in.)
Height	168 mm (6.6 in.)
Weight	
Minimum configuration	14.5 kg (32 lb)
Maximum configuration	15.9 kg (35 lb)
Cables	
Power cable	1.8 m (6 ft)
Keyboard cable	1.8 m (6 ft)
Air Temperature	
System on	10.0 to 35.0°C (50 to 95°F) 0 – 7000 ft
System off	10.0 to 43.0°C (50 to 110°F)
Humidity	
System on	8% to 85%
System off	5% to 85%
Maximum Altitude	2133.6 m (7000 ft)
Heat Output	(300 watts) 1030 BTUs per hour
Acoustical Readings	See Figure 1-9 on page 1-16
Electrical	
Input voltage (range is switch selected; sinewave input is required)	
Low range	90 (min) – 137 (max) V ac
High range	180 (min) – 265 (max) V ac
Input frequency	50 ± 3 Hz or 60 ± 3 Hz
Maximum current draw	
Low range	4.6 A
High range	2.0 A
Input in kilovolt-ampere (kVA)	
Minimum configuration (as shipped by IBM)	0.09 kVA
Maximum configuration	0.50 kVA
Electromagnetic Compatibility	FCC Class B

Figure 1-8. Physical Specifications—Model M57 SLC

| The following figure contains the acoustical readings for Model 57 | systems.

Description	L_{WAd} in bels Operate	Idle	L_{pAm} in dB Operate	Idle	$<L_{pA}>_m$ in dB Operate	Idle
Model 57	5.0	5.0	41	41	36	35

Notes

L_{WAd} is the declared sound power level for the random sample of machines.

L_{pAm} is the mean value of the A-weighted sound pressure levels at the operator position (if any) for the random sample of machines.

$<L_{pA}>_m$ is the mean value of the A-weighted sound pressure levels at the one-meter positions for the random sample of machines.

All measurements are made in accordance with ANSI standard 12.10 and reported in conformance with ISO DIS 9296.

The measurements are preliminary data and subject to change.

Figure 1-9. Acoustical Readings

Power Supply

The power supply converts the ac input voltage to three dc output voltages and provides power for the following:

- System board
- Micro Channel adapters
- Internal DASD drives
- Auxiliary devices
- Keyboard.

The power switch and two light-emitting diodes (LED) are on the front of the system unit. The power supply is operating properly when the green LED is lit. The amber LED is the fixed-disk-activity indicator.

Outputs

The power supply provides separate voltage sources for the system board and the drives. The system board voltages are +5 V dc, +12 V dc, and −12 V dc. The drive voltages are +5 V dc and +12 V dc.

The following figure shows the maximum current provided for system components.

System Component	Maximum Current	
	+12 V dc	+5 V dc
Internal fixed disk drives†	4.5 A	5.0 A
Auxiliary device	None	300 mA
Keyboard	None	275 mA
† Total for three power-supply connectors.		

Figure 1-10. Component Maximum Current

The following figure shows the maximum load currents allowed for each channel connector.

Supply Voltage	Maximum Load Current
+ 5.0 V dc	1.6 A
+12.0 V dc	0.175 A
−12.0 V dc	0.040 A

Figure 1-11. Channel Load Current

The formulas used to determine the power requirements and the voltage regulation tolerances are in the Micro Channel architecture information in the *IBM Personal System/2 Hardware Interface Technical Reference − Architectures.*

Output Protection

A short circuit placed on any dc output (between outputs or between an output and dc return) latches all dc outputs into a shutdown state, with no damage to the power supply.

If an overvoltage fault occurs in the power supply, the power supply latches all dc outputs into a shutdown state before any output exceeds 130% of the nominal value of the power supply.

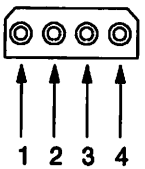
If either of these shutdown states occurs, the power supply returns to normal operation only after the fault has been removed and the power switch has been turned off for at least ten seconds.

Voltage Sequencing

During power-on, the output voltages track within 50 milliseconds of each other when measured at the 50% points.

Power Supply Connectors

The power supply provides three 4-pin connectors for internal devices. The total power used by both connectors must not exceed the amount shown in Figure 1-10 on page 1-17.



Pin	Signal	Pin	Signal
1	+12 V dc	3	dc return
2	dc return	4	+5 V dc

Figure 1-12. Voltage Assignments for the Internal Power-Supply Connectors

Section 2. Programmable Option Select

Description 2-3
POS Address Map 2-5
Card Selected Feedback Register (Hex 0091) 2-6
System Board Setup 2-6
System Board Enable/Setup Register (Hex 0094) 2-7
System Board Group 1 POS Register 2 (Hex 0102) 2-8
System Board Group 1 POS Register 3 (Hex 0103) 2-10
System Board Group 1 POS Register 4 (Hex 0104) 2-11
System Board Group 1 POS Register 5 (Hex 0105) 2-11
System Board Group 2 POS Register 2 (Hex 0102) 2-12
System Board Group 2 POS Register 3 (Hex 0103) 2-12
System Board Group 3 POS Register 4 (Hex 0104) 2-13
System Board SCSI Controller POS Register 2 (Hex 0102) . . 2-13
System Board SCSI Controller POS Register 3 (Hex 0103) . . 2-14
Adapter Enable/Setup Register (Hex 0096) 2-15

© Copyright IBM Corp. 1991, 1992

Notes:

Description

Programmable Option Select (POS) eliminates the need for switches by replacing their function with programmable registers. This section describes the POS information used on all Model 57 system boards. For additional POS information, refer to the *IBM Personal System/2 Hardware Interface Technical Reference—Architectures*.

Note: For Model M57 SLC, there is additional POS information for the M-Audio Capture and Playback Adapter. Refer to the *M-Audio Capture and Playback Adapter Installation and Technical Reference Manual* for more information.

Warning:

- IBM recommends that programmable options be set only through the System Configuration utilities. Directly setting the POS registers or CMOS RAM POS parameters can result in multiple assignments of the same system resource, improper operation of the feature, loss of data, or damage to the hardware.
- Application programs should not use the adapter identification (ID) unless absolutely necessary. Compatibility problems can result.
- If an adapter and the system board are in setup mode at the same time, bus contention occurs, no useful programming can take place, and damage to the hardware can occur.
- After setup operations are complete, the Adapter Enable/Setup register (hex 0096) should be set to hex 00, and the System Board Enable/Setup register (hex 0094) should be set to hex FF.
- The channel-reset bit (bit 7) in the Adapter Enable/Setup register (hex 0096) must be set to 0 to program the adapters.
- The system board does not support 16-bit I/O operations to 8-bit POS registers. Using 16-bit I/O instructions on 8-bit POS registers causes incorrect data to be written to or read from the registers. Only 8-bit transfers are supported for setup operations.

Setup functions respond to I/O addresses hex 0100 through hex 0107 only when their unique setup signal is active. The following precautions must be taken before setting individual bits in the POS registers.

System Board Video Subsystem Setup:

- Bit 5 in the System Board Enable/Setup register (hex 0094) must be set to 0 to place the system board video into the setup mode.
- Bit 3 in the Adapter Enable/Setup register (hex 0096) must be set to 0 to avoid driving a 'setup' signal to an adapter.
- Bits 7, 6, 4, and 3 in the System Board Enable/Setup register must be set to 1 to avoid driving a 'setup' signal to other system board functions.

Adapter Setup:

- Bit 3 in the Adapter Enable/Setup register must be set to 1 to allow adapter setup.
- Bits 7 – 3 in the System Board Enable/Setup register must be set to 1 to avoid driving a 'setup' signal to system board or video subsystem functions.

System Board Setup Group 1:

- Bit 7 in the System Board Enable/Setup register must be set to 0 to allow setup of other system board functions.
- Bit 3 in the Adapter Enable/Setup register must be set to 0 to avoid driving a 'setup' signal to an adapter.
- Bits 6 – 3 in the System Board Enable/Setup register must be set to 1 to avoid driving a 'setup' signal to the video subsystem or other system board functions.

System Board Setup Group 2:

- Bit 6 in the System Board Enable/Setup register must be set to 0 to allow setup of other system board functions.
- Bit 3 in the Adapter Enable/Setup register must be set to 0 to avoid driving a 'setup' signal to an adapter.
- Bits 7, 5, 4, and 3 in the System Board Enable/Setup register must be set to 1 to avoid driving a 'setup' signal to the video subsystem or other system board functions.

System Board Setup Group 3:

- Bit 4 in the System Board Enable/Setup register must be set to 0 to allow setup of other system board functions.
- Bit 3 in the Adapter Enable/Setup register must be set to 0 to avoid driving a 'setup' signal to an adapter.
- Bits 7, 6, 5, and 3 in the System Board Enable/Setup register must be set to 1 to avoid driving a 'setup' signal to the video subsystem or other system board functions.

System Board SCSI Controller Setup:

- Bit 3 in the System Board Enable/Setup register must be set to 0 to allow setup of other system board functions.
- Bit 3 in the Adapter Enable/Setup register must be set to 0 to avoid driving a 'setup' signal to an adapter.
- Bits 7–4 in the System Board Enable/Setup register must be set to 1 to avoid driving a 'setup' signal to the video subsystem or other system board functions.

POS Address Map

The following figure shows the organization of the I/O address space used by POS. Bit 0 of POS Register 2 and bits 6 and 7 of POS Register 5 are fixed. All other bits in POS Registers 2 through 5 are free form.

Address (Hex)	Function
0094	System Board Enable/Setup register
0096	Adapter Enable/Setup register
0100	POS Register 0–Adapter Identification byte (low byte)
0101	POS Register 1–Adapter Identification byte (high byte)
0102	POS Register 2–Option Select Data byte 1–Bit 0 is card enable
0103	POS Register 3–Option Select Data byte 2
0104	POS Register 4–Option Select Data byte 3
0105	POS Register 5–Option Select Data byte 4
0106	Reserved
0107	Reserved

Figure 2-1. POS I/O Address Map

Card Selected Feedback Register (Hex 0091)

Whenever an adapter is addressed, it responds by setting the '-card selected feedback' (-CD SFDBK) signal to active. -CD SFDBK is derived from the address decode, driven by a totem-pole driver, and latched by the system board. It can be read through the Card Selected Feedback register at address hex 0091. Diagnostic and automatic configuration programs can use this signal to verify the operation of an adapter at a given address or DMA port. This signal must not be active during a setup cycle.

The Card Selected Feedback register is a read-only register at address hex 0091. It allows programs to monitor -CD SFDBK and determine if the video subsystem, system board I/O, or an adapter is addressed and functioning.

Bit	Function
7–1	Reserved
0	Card selected feedback

Figure 2-2. Card Selected Feedback Register (Hex 0091)

Bits 7–1 These bits are reserved.

Bit 0 This bit is set to 1 whenever -CD SFDBK was active on a previous cycle or whenever the system board I/O functions (diskette drive, serial, or parallel interfaces) are accessed by an I/O cycle. Reading this register resets the bit to 0.

System Board Setup

The integrated I/O functions on the system board use POS information during setup. The diskette drive controller, serial port, and parallel port are treated as a single device. The video subsystem also is an integrated part of the system board, but POS treats it as a separate device. The System Board Enable/Setup register is used to place the system board or the video subsystem in the setup mode.

Figure 2-3 indicates the group, POS register number, and I/O function supported.

Group	POS Register	Device
1	2	Diskette drive Parallel port controllers
1	3	Memory connectors
1	4	Diskette arbitration level
1	5	Parallel port arbitration level
2	2, 3	Serial port controller
3	4	SCSI controller arbitration level
SCSI Controller	2, 3	Enable SCSI controller

Figure 2-3. Devices Supported in Setup Mode

System Board Enable/Setup Register (Hex 0094)

This is a read/write register. All bits in this register default to 1 (enabled).

Bit	Function
7	Enable/setup system board group 1
6	Enable/setup system board group 2
5	Enable/setup video subsystem
4	Enable/setup system board group 3
3	Enable/setup SCSI controller
2–0	Reserved

Figure 2-4. System Board Enable/Setup Register (Hex 0094)

Bit 7 When this bit is set to 0, it places the diskette drive controller, parallel port controllers, memory connectors, diskette arbitration level, and parallel port arbitration level in setup mode.

When this bit is set to 1, it enables the system board functions.

Bit 6 When this bit is set to 0, it places the system board serial controller in setup mode. System Board Group 2 POS Register 2 (hex 0102) includes the serial port enable and interrupt-level selection. The serial port address selection is in System Board Group 2 POS Register 3 (hex 0103).

Bit 5 When this bit is set to 0, the video subsystem is placed in the setup mode and controlled through Group 1 POS Register 2 (hex 0102). When this bit is set to 1 and bit 0 in Group 1 POS Register 2 is set to 1, video is enabled.

Bit 0 of Group 1 POS Register 2 is the video-enable bit. When this bit is set to 0, the video subsystem does not respond to commands, addresses, or data. If video is being generated when the video-enable bit is set to 0, the output is still generated. For information on BIOS calls to enable or disable the video, see the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference.*

Note: When the video subsystem is disabled, access to the Digital-to-Analog Converter (DAC) registers is disabled.

Bit 4 When this bit is set to 0, the system board bus controller is placed in setup mode and controlled through POS Register 4 (hex 0104). POS Register 4 includes the system board SCSI-controller arbitration level. All other bits in POS Register 4 and POS Register 5 are reserved.

Bit 3 When this bit is set to 0, the system board SCSI controller is placed in setup mode and is controlled through POS Register 2 (hex 0102). When this bit is set to 1 and bit 0 in POS Register 2 is set to 1, the system board SCSI controller is enabled.

Bits 2–0 These bits are reserved.

System Board Group 1 POS Register 2 (Hex 0102)

When the system board is in the setup mode, this read/write register controls the diskette drive controller, serial port, and parallel port. Reading this register returns the current state of these system board functions.

Bit	Function
7	Disable parallel port extended mode
6, 5	Select parallel port
4	Enable parallel port
3	Reserved
2	Reserved
1	Enable diskette drive interface
0	Enable system board

Figure 2-5. System Board POS Register 2 (Hex 0102)

Bit 7 When this bit is set to 0, this bit allows the parallel port to be configured as an 8-bit, parallel, bidirectional (extended mode) interface. When this bit is set to 1, this bit disables the bidirectional mode. This bit is set to 0 after power-on.

Bits 6, 5 These bits select the configuration of the system board parallel port.

Bits 6 5	Assignment	Hex Address	Interrupt Level
0 0	Parallel 1	03BC–03BF†	7
0 1	Parallel 2	0378–037D	7
1 0	Parallel 3	0278–027D	7
1 1	Parallel 4	1378–137D	7

† The addresses are hex 1278–127D for the DMA operations.

Figure 2-6. Parallel Port Select Bits

Bit 4 When this bit and bit 0 are set to 1, the system board parallel port is enabled.

Bits 3, 2 These bits are reserved.

Bit 1 When this bit and bit 0 are set to 1, the diskette drive interface is enabled.

Bit 0 When this bit is set to 1, bits 4 and 1 can enable and disable their respective devices. When this bit is set to 0, this bit disables the diskette drive interface and system board serial and parallel ports, regardless of the state of bits 4 and 1.

System Board Group 1 POS Register 3 (Hex 0103)

The system board has connectors that provide for installation of three single inline memory modules (memory modules) (for more information about these memory connectors, see "System Board Memory Connectors" on page 3-11). Information about each memory module is contained in four read-only data bits in POS Register 3. Each set of data bits is selected by writing the appropriate index to the write-only index register.

POS Register 3 provides access to the read-only data bits and a write-only index register. To determine the memory information for a particular connector, write the appropriate index to POS Register 3, then read POS Register 3. The following figure shows the relationship of the index value, the memory connector, and the available data register information.

Index Value (Hex)	Function
00	Presence detection information for memory connectors 2 and 1
04	Presence detection information for memory connector 3

Note: All values not shown are reserved.

Figure 2-7. System Board Group 1 POS Register 3 (Hex 0103) – Write

The following information is returned by reading POS Register 3 after the appropriate index value has been written to POS Register 3. For coding of the presence detection signals, see Figure 3-9 on page 3-15.

Index 00 Bit	Function
7	Presence detect 3, memory connector 2
6	Presence detect 2, memory connector 2
5	Presence detect 1, memory connector 2
4	Presence detect 0, memory connector 2
3	Presence detect 3, memory connector 1
2	Presence detect 2, memory connector 1
1	Presence detect 1, memory connector 1
0	Presence detect 0, memory connector 1

Figure 2-8. System Board POS Group 1 Register 3 (Hex 0103) – Read Index 00

Index 04 Bit	Function
7–4	Reserved
3	Presence detect 3, memory connector 3
2	Presence detect 2, memory connector 3
1	Presence detect 1, memory connector 3
0	Presence detect 0, memory connector 3

Figure 2-9. System Board POS Group 1 Register 3 (Hex 0103) – Read Index 04

System Board Group 1 POS Register 4 (Hex 0104)

This register is used to specify the diskette arbitration level.

Bit	Function
7–4	Reserved
3–0	Diskette arbitration level

Figure 2-10. System Board Group 1 POS Register 4 (Hex 0104)

Bits 7–4 These bits are reserved.

Bits 3–0 These bits select the diskette arbitration level.

System Board Group 1 POS Register 5 (Hex 0105)

This register is used to specify the arbitration level used by the parallel port controller.

Bit	Function
7–4	Reserved
3–0	Parallel port arbitration level

Figure 2-11. System Board Group 1 POS Register 5 (Hex 0105)

Bits 7–4 These bits are reserved.

Bits 3–0 These bits are used to select the arbitration level used by the parallel port controller.

System Board Group 2 POS Register 2 (Hex 0102)

This register is used to enable the system board serial port.

Bit	Function
7–4	Reserved
3	Enable serial
2–1	Reserved
0	Select serial interrupt

Figure 2-12. System Board Group 2 POS Register 2 (Hex 0102)

Bits 7–4 These bits are reserved.

Bit 3 When this bit is set to 1 and bit 0 in POS Register 2 (Group 1) is set to 1, the serial port is enabled.

Bits 2–1 These bits are reserved.

Bit 0 When this bit is set to 1, the serial port uses IRQ4. When this bit is set to 0, the serial port uses IRQ3.

System Board Group 2 POS Register 3 (Hex 0103)

This register is used to select the address of the serial port.

Bit	Function
7–4	Reserved
3–0	Serial port address

Figure 2-13. System Board Group 2 POS Register 3 (Hex 0103)

Bits 7–4 These bits are reserved.

Bits 3–0 These bits are used to determine the address of the serial port.

Bits 3 2 1 0	Assignment	Hex Address	
		Compatible	Extended
0 0 0 0	Serial 1	03F8 – 03FF	83F8 – 83FF
0 0 0 1	Serial 2	02F8 – 02FF	82F8 – 82FF
0 0 1 0	Serial 3	3220 – 3227	B220 – B227
0 0 1 1	Serial 4	3228 – 322F	B228 – B22F
0 1 0 0	Serial 5	4220 – 4227	C220 – C227
0 1 0 1	Serial 6	4228 – 422F	C228 – C22F
0 1 1 0	Serial 7	5220 – 5227	C220 – D227
0 1 1 1	Serial 8	5228 – 522F	D228 – D22F
1 0 0 0	Serial 9	3A20 – 3A27	BA20 – BA27
1 0 0 1	Serial 10	3A28 – 3A2F	BA28 – BA2F
1 0 1 0	Serial 11	3A30 – 3A37	BA30 – BA37
1 0 1 1	Serial 12	3A38 – 3A3F	BA38 – BA3F
1 1 0 0	Serial 13	3A80 – 3A87	BA80 – BA87
1 1 0 1	Serial 14	3A88 – 3A8F	BA88 – BA8F
1 1 1 0	Serial 15	3A90 – 3A97	BA90 – BA97
1 1 1 1	Serial 16	3A98 – 3A9F	BA98 – BA9F

Figure 2-14. Serial Port Addresses

System Board Group 3 POS Register 4 (Hex 0104)

This register is used to select the arbitration level of the SCSI controller on the system board.

Bit	Function
7 – 4	SCSI arbitration level (system board)
3 – 0	Reserved

Figure 2-15. System Board Group 3 POS Register 4 (Hex 0104)

Bits 7 – 4 These bits select the SCSI arbitration level.

Bits 3 – 0 These bits are reserved.

System Board SCSI Controller POS Register 2 (Hex 0102)

This register is used to enable the system board SCSI controller.

Bit	Function
7 – 1	Reserved
0	Enable SCSI

Figure 2-16. System Board SCSI Controller POS Register 2 (Hex 0102)

Bits 7–1 These bits are reserved.

Bit 0 When this bit is set to 1, this bit enables the system board SCSI controller.

System Board SCSI Controller POS Register 3 (Hex 0103)

This register is used to select the SCSI ID for the system board SCSI controller.

Bit	Function
7–5	SCSI ID
4–0	Reserved

Figure 2-17. System Board SCSI Controller POS Register 3 (Hex 0103)

Bits 7–5 These bits determine the SCSI ID for the system board SCSI controller as follows:

Bits 7 6 5	System Board SCSI ID
0 0 0	0
0 0 1	1
0 1 0	2
0 1 1	3
1 0 0	4
1 0 1	5
1 1 0	6
1 1 1	7

Figure 2-18. System Board SCSI Controller ID

Bits 4–0 These bits are reserved.

Adapter Enable/Setup Register (Hex 0096)

This read/write register resets the channel, enables the '-card setup' signal, and configures the connectors.

Bit	Function
7	Channel reset
6–4	Reserved
3	Enable card setup
2–0	Channel select 2–0

Figure 2-19. Adapter Enable/Setup Register (Hex 0096)

Bit 7 When this bit is set to 1, this bit activates the 'channel reset' signal to all connectors.

Bits 6–4 These bits are reserved and must be set to 0.

Bit 3 When this bit is set to 1, this bit enables the '-card setup' signal selected by bits 2–0.

Bits 2–0 These bits are the address bits for the '-card setup' signal to the connectors. Connectors 1 through 5 are addressed as 0 through 4. When bit 3 is set to 1, these bits select the connector or function for setup cycles.

Each channel connector has a unique '-card setup' signal (-CD SETUP) associated with it. This signal is used to select the adapters for setup cycles, which allows access to the POS registers. The individual connectors are selected through the Adapter Enable/Setup register. Setup information is then read from or written to the selected adapter through I/O addresses hex 0100 through 0107.

Note: -CD SETUP goes active only when an operation is performed in the I/O address range hex 0100 through 0107.

Notes:

Section 3. System Board

Description 3-3
Microprocessor—Type 1 Systems 3-3
Microprocessor—Type 2 Systems 3-3
Programming Model 3-4
Instructions 3-4
Cache Operation 3-4
Cacheable Address Space 3-5
Math Coprocessor 3-6
Micro Channel Implementation 3-6
Central Arbiter 3-6
Arbitration Bus Priority Assignments 3-7
Central Arbiter Programming 3-8
Bus Adapter 3-9
Diskette Drive Connector 3-10
System Board Memory Connectors 3-11
SCSI-Device Connector 3-12
Memory 3-13
ROM Subsystem 3-14
RAM Subsystem 3-14
Memory Interleaving Option 3-15
Memory-Presence Detect Signals 3-15
System Memory Map 3-16
Memory Controller Registers 3-16
Memory-Fault RAS Capture Register (I/O Port Hex 00E2) . 3-17
Memory-Controller-Fault Address Registers (I/O Ports Hex 00E4 and Hex 00E5) 3-17
RT/CMOS RAM 3-18
RT/CMOS Address and NMI Mask Register (Hex 0070) . . 3-19
RT/CMOS Data Register (Hex 0071) 3-20
RT/CMOS RAM I/O Operations 3-20
Real-Time Clock Bytes (Hex 000—00D) 3-21
CMOS RAM Configuration 3-24
NVRAM 3-27
Miscellaneous System Functions and Ports 3-28
Nonmaskable Interrupt 3-28
System Control Port B (Hex 0061) 3-28
System Control Port A (Hex 0092) 3-29
Power-On Password 3-31
Selectable Boot Sequence 3-31
Hardware Compatibility 3-32
Error Codes 3-33

© Copyright IBM Corp. 1991, 1992

Notes:

Description

This section describes the microprocessors, math coprocessor, channel differences, memory subsystems, and miscellaneous system ports and connectors for the all Model 57 systems. Additional information on these topics can be found in the *IBM Personal System/2 Hardware Interface Technical Reference—Common Interfaces*.

Type 1 systems contain an 80386SX microprocessor and Type 2 systems contain a 386SLC microprocessor. To identify the microprocessor used in the system unit, use INT 15H, function code 0C9H. This returns the microprocessor type and stepping level.

Microprocessor—Type 1 Systems

The 80386SX microprocessor used in Type 1 systems runs at 20 MHz and has a 24-bit address and a 16-bit data interface. The 80386SX microprocessor is software compatible with the 80386DX microprocessor. The type for the 80386SX is hex 023.

Microprocessor—Type 2 Systems

The 386SLC microprocessor used in Type 2 systems runs at 20 MHz and has an 8KB, 2-way set-associative store-through instruction and data cache. It contains an internal 32-bit data path (including the path to the cache) with external 16-bit data and 24-bit address paths. The 386SLC maintains binary compatibility with programs written for the 80386SX. The type for the IBM 386SLC is hex 0A3.

Programming Model

The IBM 386SLC microprocessor implements the same registers and bit descriptions as the Intel 80386SX with the following additions:

- The 386SLC implements the WP bit in the CR0 of the Intel486* microprocessor.
- The 386SLC implements the PCD bit in the Page Table Entry of the Intel486 microprocessor.

Instructions

The 386SLC microprocessor decodes the same instructions as the 80386SX. The 386SLC microprocessor also decodes the following 6 Intel486 instructions:

- BSWAP
- XADD
- CMPXCHG
- INVD
- WBINVD
- INVLPG.

Cache Operation

The cache in the 386SLC microprocessor enables it to read instructions and data much faster than if the microprocessor had to access system memory. When instructions are first used or data is first read or written, they are transferred to the cache from main memory. This enables future accesses to the instructions or data to occur much faster.

The cache is disabled and empty when the microprocessor comes out of the reset state. The cache is tested and enabled during the power-on self-test (POST).

The cache is loaded from system memory in 16-byte increments. These increments are referred to as a cache line. A cache line is aligned on a paragraph boundary. A reference to any byte contained

---

* Trademark of Intel Corporation.

in the line results in the entire cache line being read into the cache (if the data was not already in the cache). When the microprocessor gives up control of the system bus, the cache enters a "snoop" mode and monitors all write operations. If a memory write occurs to a location that is in the cache, the corresponding cache line is invalidated.

When the microprocessor performs a memory read, the address is used to find the data in the cache. If the data is found (a hit), it is read from the cache and no external bus cycle occurs. If the data is not found (a miss), an external bus cycle is used to read the data from system memory. If the address of the missed data is in cacheable address space, the data is stored in the cache and the rest of the cache line is read.

When the microprocessor performs a memory write, the address is used to search the cache. If found, the data is written to the cache. Then (hit or miss), an external bus cycle is used to write the data to system memory. If the address of the write operation was not in the cache but was in cacheable address space, the data is read back into the cache and the rest of the cache line is read.

Cacheable Address Space

Cacheable address space is defined as system memory that resides on the system board (0–640KB and 1MB–16MB). Nothing in the address range hex A0000–DFFFF is cached, and no memory that is located in any Micro Channel slot is cached. Nothing in the I/O address space is cached.

ROM address space (hex E0000–FFFFF) is cacheable for *read operations only*. If data in this address range is already in cache and the address range is written to, the cached line is invalidated and is read again from the ROM.

Math Coprocessor

The 80387SX math coprocessor is matched to the speed of the system microprocessor and operates in the synchronous mode. The 80387SX math coprocessor is software compatible with the 80387DX math coprocessor.

Micro Channel Implementation

This section describes the implementation of the Micro Channel architecture on Model 57 systems. For general Micro Channel architecture information, refer to the *IBM Personal System/2 Hardware Interface Technical Reference – Architectures.*

Central Arbiter

The central arbitration control point gives programmable subsystems on the channel the ability to share and control the system. It allows burst data transfers and prioritization of control between devices. The central arbiter supports up to 15 arbitrating devices (levels 0 through E) and the system microprocessor (level F).

Note: The system board SCSI controller performs cycles at arbitration level F if the Micro Channel arbitration level is F when the SCSI controller requests the bus.

Arbitration Bus Priority Assignments

The following figure shows the assignment of arbitration levels. The functions with the *lowest* arbitration level have the *highest* priority.

ARB Level	Primary Assignment
-2	Memory refresh
-1	NMI
0	DMA channel 0 (Programmable to any arbitration level)
1	DMA channel 1
2	DMA channel 2
3	DMA channel 3
4	DMA channel 4 (programmable to any arbitration level)
5	DMA channel 5
6	DMA channel 6
7	DMA channel 7
8–E	Available
F	System microprocessor

Figure 3-1. Arbitration Bus Priority Assignments

Note: Devices designed for arbitration level 0 or 1 should have limited bandwidth or short bursts so that diskette overruns can be prevented or recovered by retry operations. The diskette drive controller, normally on arbitration level 2, can be held inactive by devices on levels 0 and 1, by a refresh operation, and by the previous controlling master. To prevent overrun, the diskette drive controller should not be held inactive for more than 12 microseconds.

Nonmaskable interrupt (NMI) service is executed at a priority level higher than 0, called -1. Memory refresh is prioritized at -2, two levels higher than 0. Levels -1 and -2 are reached on the system board only, while the 'arbitrate/-grant' signal (ARB/-GNT) is in the arbitration state.

When the central arbitration control point receives a level -1 request (NMI, a system board internal signal), it activates -PREEMPT, waits for the end of transfer, and then places ARB/-GNT in the arbitrate state, which denies channel activity to arbitrating devices. The central arbitration control point gives the grant to the level -1 request and holds ARB/-GNT in the arbitrate state until the operation is complete and the NMI is reset.

Central Arbiter Programming

The central arbitration control point provides access to programmable options through the Arbitration register, which is accessed at I/O address hex 0090. The bits are defined differently for read and write operations, as shown in the following figures.

Bit	Function
7	Enable system microprocessor cycle
6	Arbitration mask
5	Enable extended arbitration (not supported)
4–0	Reserved

Figure 3-2. Arbitration Register (Hex 0090) – Write

Bit	Function
7	Enable system microprocessor cycle
6	Arbitration masked by NMI
5	Bus time-out
4	Reserved
3–0	Value of arbitration bus during previous grant state

Figure 3-3. Arbitration Register (Hex 0090) – Read

Bit 7 Setting this bit to 1 enables system microprocessor cycles during arbitration cycles. This bit can be set to 0 if an arbitrating device requires total control of the channel bandwidth. This bit is set to 1 by POST.

Reading this bit as a 1 indicates that the system microprocessor cycles are enabled during arbitration.

Bit 6 Setting this bit to 1 causes the central arbitration control point to enter the arbitration state. The system microprocessor controls the channel until this bit is reset to 0 by a system reset.

Reading this bit as a 1 indicates that an NMI has occurred and has masked arbitration.

Warning: This bit should be set to 1 only by diagnostic routines and system error-recovery routines.

Bit 5 Setting this bit to 1 enables extended arbitration. The minimum arbitration cycle is 300 nanoseconds; this bit extends that minimum cycle to 600 nanoseconds. This bit is set to 0 during POST.

Reading this bit as a 1 indicates that a bus time-out has occurred and resets bit 6 in this register to 0.

Bit 4 This bit is reserved and must be set to 0.

Bits 3–0 These bits are undefined for a write operation and must be set to 0.

Reading these bits returns the arbitration level of the arbiter controlling the channel during the most recent grant state. This information allows the system microprocessor to determine the arbitration level of the device that caused a bus time-out.

Bus Adapter

The Model 57 systems use a direct bus attachment board to provide five channel connectors:

- Four 16-bit Micro Channel connectors
- One 16-bit Micro Channel connector with an auxiliary video extension.

The adapters are installed horizontally into the direct bus attachment board.

Diskette Drive Connector

The diskette drive cable connects to the system board and provides three 2- by 17-pin connectors for attaching internal diskette drives. Each connector passes control and data signals between the diskette drive controller on the system board and the drives. It also provides the power to each drive.

The following figure shows the signal assignments and pin numbers for the diskette drive connectors.

Pin	Signal	Pin	Signal
1	Ground	2	Data rate select 1
3	+5 V dc	4	Drive type ID 1
5	Ground	6	+12 V dc
7	Ground	8	-Index
9	Drive type ID 0	10	Reserved
11	Ground	12	-Drive select
13	Ground	14	Reserved
15	Ground	16	-Motor enable
17	Media type ID 1	18	-Direction in
19	Ground	20	-Step
21	Ground	22	-Write data
23	Ground	24	-Write enable
25	Ground	26	-Track 0
27	Media type ID 0	28	-Write protect
29	Ground	30	-Read data
31	Ground	32	-Head 1 select
33	Data rate select 0	34	-Diskette change

Figure 3-4. Diskette Drive Connector

System Board Memory Connectors

The Model 57 system board has three 72-pin connectors for single inline memory modules (memory modules).

The following figure shows the pin assignments for the 1- by 72-pin memory connectors. Refer to "Memory-Presence Detect Signals" on page 3-15 for decoding of the presence-detect signals.

Pin	Signal	Pin	Signal
1	Ground	37	Parity data 1
2	Data 0	38	Parity data 3
3	Data 16	39	Ground
4	Data 1	40	Column address strobe 0
5	Data 17	41	Column address strobe 2
6	Data 2	42	Column address strobe 3
7	Data 18	43	Column address strobe 1
8	Data 3	44	Row address strobe 0
9	Data 19	45	Row address strobe 1
10	+5 V dc	46	Block select 1
11	-Column address strobe P	47	Write enable
12	Address 0	48	Reserved
13	Address 1	49	Data 8
14	Address 2	50	Data 24
15	Address 3	51	Data 9
16	Address 4	52	Data 25
17	Address 5	53	Data 10
18	Address 6	54	Data 26
19	Reserved	55	Data 11
20	Data 4	56	Data 27
21	Data 20	57	Data 12
22	Data 5	58	Data 28
23	Data 21	59	+5 V dc
24	Data 6	60	Data 29
25	Data 22	61	Data 13
26	Data 7	62	Data 30
27	Data 23	63	Data 14
28	Address 7	64	Data 31
29	Block select 0	65	Data 15
30	+5 V dc	66	Block select 2
31	Address 8	67	Presence detect 0
32	Address 9	68	Presence detect 1
33	Row address strobe 3	69	Presence detect 2
34	Row address strobe 2	70	Presence detect 3
35	Parity data 2	71	Block select 3
36	Parity data 0	72	Ground

Figure 3-5. System Board Memory Connectors

SCSI-Device Connector

The SCSI-device cable connects to the system board and provides three 2- by 25-pin connectors for attaching internal SCSI devices. The Model 57 system board has one 2- by 30-pin connector for attaching external SCSI devices.

The following figure shows the pin assignments for the 2- by 25-pin internal SCSI connector.

Pin	Signal	Pin	Signal
1	Ground	26	+Terminal power
2	Data 0	27	Ground
3	Ground	28	Ground
4	Data 1	29	Ground
5	Ground	30	Ground
6	Data 2	31	Ground
7	Ground	32	-Attention
8	Data 3	33	Ground
9	Ground	34	Ground
10	Data 4	35	Ground
11	Ground	36	-Busy
12	Data 5	37	Ground
13	Ground	38	-Acknowledge
14	Data 6	39	Ground
15	Ground	40	-Reset
16	Data 7	41	Ground
17	Ground	42	-Message
18	Data P	43	Ground
19	Ground	44	-Select
20	Ground	45	Ground
21	Ground	46	-Control/data
22	Ground	47	Ground
23	Ground	48	-Request
24	Ground	49	Ground
25	Not connected	50	-Input/output

Figure 3-6. SCSI Device Cable Internal Connector Pin Assignments

The following figure shows the pin assignments for the 2- by 30-pin external SCSI connector.

Pin	Signal	Pin	Signal
1	Ground	31	Ground
2	Data 0	32	-Attention
3	Ground	33	Ground
4	Data 1	34	Ground
5	Ground	35	Ground
6	Data 2	36	-Busy
7	Ground	37	Ground
8	Data 3	38	-Acknowledge
9	Ground	39	Ground
10	Data 4	40	-Reset
11	Ground	41	Ground
12	Data 5	42	-Message
13	Ground	43	Ground
14	Data 6	44	-Select
15	Ground	45	Ground
16	Data 7	46	-Control/data
17	Ground	47	Ground
18	Data P	48	-Request
19	Ground	49	Ground
20	Ground	50	-Input/output
21	Ground	51	Ground
22	Ground	52	Reserved
23	Ground	53	Reserved
24	Ground	54	Reserved
25	Not connected	55	Reserved
26	+Terminal power	56	Reserved
27	Ground	57	Reserved
28	Ground	58	Reserved
29	Ground	59	Reserved
30	Ground	60	Reserved

Figure 3-7. System Board SCSI Device External Connector

Memory

The Model 57 systems use the following types of memory:

- Read-only memory (ROM)
- Random access memory (RAM)
- Real-time clock and CMOS RAM (RT/CMOS RAM)
- Nonvolatile random access memory (NVRAM).

ROM Subsystem

The ROM subsystem consists of 256KB of physical memory in two 128KB pages. Mapped ROM is active at power-on time and is assigned to the top of the first and last 1MB of address space (hex 0E0000–0FFFFF and hex FE0000–FFFFFF). After POST ensures that system memory is operating properly, the ROM code is copied to RAM at the same address space, and ROM is disabled.

Access to ROM or RAM at address space hex 0E0000–0FFFFF is controlled by the ROM enable bit in the Memory-Controller Configuration register. When ROM is enabled, it is not parity-checked, and operates with two 50-nanosecond wait states.

RAM Subsystem

The RAM subsystem on the system board starts at address hex 000000 of the address space. The RAM subsystem is 18 bits wide: 16 data bits and 2 parity bits. One parity bit is generated for each byte of data written. During a read operation, one parity bit is checked for each byte of data that is read by the device controlling the bus.

Three memory connectors are provided on the system board. Each connector can support 2MB, 4MB, or 8MB of 70-nanosecond memory.

The system enables and disables memory in 1MB blocks. Each 1MB block must start on a 1MB boundary. Because 128KB of I/O ROM, 128KB of system board ROM, and 128KB of video memory are mapped within the first 1MB of address space, the first physical 1MB of RAM cannot be mapped to contiguous addresses and is split at the 640KB boundary, creating a 384KB overflow. The high 128KB of the overflow is mapped to the system board ROM address space; the remaining 256KB, called the *split memory block*, is either reserved for system use or mapped to the first available address following the last full 1MB of activated system board memory. Split-memory-block remapping and disabling are controlled by the Memory-Controller Configuration register. The presence of the split memory block in the memory address space, might be affected by nonsystem memory.

Memory Interleaving Option

The system board memory controller supports interleaving for specific memory configurations. The following figure lists the interleaving configurations for the three memory connectors.

Memory Connector 1	Memory Connector 2	Memory Connector 3
2MB	Empty	Empty
2MB	2MB	Empty
2MB	2MB	2MB
4MB	4MB	Empty
4MB	4MB	4MB

Figure 3-8. Memory Interleaving Configurations

Memory-Presence Detect Signals

The memory-presence detect signals are used by the system to determine the size and speed of memory on the memory module. The pins are either not connected (1) or connected to ground (0).

A set of 4 bits shows the state of all memory-presence detect signals for a specific memory connector; the bit is set to 0 when the signal is connected to ground. The following figure shows the information indicated by these bits as well as the bank size of the memory module. All combinations not shown are reserved.

Presence Detect 3 2 1 0	Single Inline Memory Module Definition	Bank Size
1 1 0 0	2MB 70-ns memory	1MB
0 1 0 0	4MB 70-ns memory	4MB
1 0 1 1	8MB 70-ns memory	4MB

Figure 3-9. Presence-Detect Bits

System Memory Map

Memory is mapped by Memory Controller registers.

The first 640KB of system board RAM is mapped starting at address hex 000000. A 256-byte area and a 1KB area of this RAM are reserved as BIOS data areas. See the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference* for details.

The following figure shows the memory map for a properly functioning system. Memory can be mapped differently if POST detects an error in system board memory or RT/CMOS RAM. In the following figure, the variable *x* represents the number of 1MB blocks of system board memory starting at or above the hex 100000 boundary.

Hex Address Range	Function
000000 to 09FFFF†	640KB system board RAM
0A0000 to 0BFFFF	128KB video RAM
0C0000 to 0DFFFF	Feature adapter ROM
0E0000 to 0FFFFF†	128K system board ROM mapped to RAM
100000 to (100000 + *x*MB) − 1†	*x*MB system board RAM
(100000 + *x*MB) to (100000 + *x*MB + 256KB) − 1†	256KB system board RAM (split memory block)
FE0000 to FFFFFF	128KB system board ROM (same as 0E0000 to 0FFFFF)

† These address ranges are cacheable on Type 2 systems.

Figure 3-10. System Memory Map

Memory Controller Registers

The memory controller, which is located on the system board, has several internal registers that are used to control the memory installed in the memory connectors on the system board. The memory controller manages this memory in banks of 1MB or 4MB. Memory is enabled in 1MB increments.

Memory-parity error information is provided through the following internal registers:

- Memory-Fault Row Address Strobe (RAS) Capture register
- Two Memory-Controller Fault Address registers.

Memory-Fault RAS Capture Register (I/O Port Hex 00E2)

Bits in this register are set by a system board parity check. The RAS signal that is active at the time of the parity check is held until the system microprocessor reads this register.

Bit	Function
7, 6	Reserved
5	RAS5
4	RAS4
3	RAS3
2	RAS2
1	RAS1
0	RAS0

Figure 3-11. Memory-Fault RAS Capture Register (I/O Port Hex 00E2)

Bits 7, 6 These bits are reserved.

Bits 5–0 The system board memory connectors are wired with fixed RAS signals as follows:

Memory Connector	RAS Signals
Memory connector 1	RAS0, RAS1
Memory connector 2	RAS2, RAS3
Memory connector 3	RAS4, RAS5

Figure 3-12. Fixed RAS Signals

Memory-Controller-Fault Address Registers (I/O Ports Hex 00E4 and Hex 00E5)

These registers capture the high-order memory address when a system board parity check occurs. The address remains in the registers until the registers are read by the system microprocessor.

The following figure shows the failing memory-address space address bits for the Memory-Controller-Fault Address register (I/O Port Hex 00E4).

I/O Port Hex 00E4	Failing-Address Bit
7	15
6	14
5	13
4	12
3	11
2	10
1	09
0	08

Figure 3-13. Memory-Controller-Fault Address Register (I/O Port Hex 00E4)

The following figure shows the failing memory-address space address bits for the Memory-Controller-Fault Address register (I/O Port Hex 00E5).

I/O Port Hex 00E5	Failing-Address Bit
7	23
6	22
5	21
4	20
3	19
2	18
1	17
0	16

Figure 3-14. Memory-Controller-Fault Address Registers (I/O Port Hex 00E5)

RT/CMOS RAM

The RT/CMOS RAM (real-time clock/complementary metal-oxide semiconductor RAM) module contains the real-time clock and 64 bytes of CMOS RAM. The clock uses 14 bytes of this memory, and the rest is allocated to configuration and system status information. A battery, which is located on the bus adapter, keeps the RT/CMOS RAM active when the power supply is off.

The system board has an additional 8KB CMOS RAM extension that is used to store configuration and other system information. This RAM extension is kept active by the battery on the bus adapter.

The following figure shows the RT/CMOS RAM bytes and their addresses.

Address (Hex)	RT/CMOS RAM Bytes
000–00D	Real-Time Clock bytes
00E	Diagnostic Status byte
00F	Shutdown Status byte
010	Diskette Drive Type byte
011	Fixed Disk Drive Type byte
012	Reserved
013	Reserved
014	Equipment byte
015, 016	Low and High Base-Memory bytes
017, 018	Low and High Expansion-Memory bytes
019–031	Reserved
032, 033	Configuration CRC bytes
034	Reserved
035, 036	Low and high usable memory above 1MB
037	Date-Century byte
038–03F	Reserved

Figure 3-15. RT/CMOS RAM Address Map

RT/CMOS Address and NMI Mask Register (Hex 0070)

This register is used with the RT/CMOS Data register (hex 0071) to read from and write to the RT/CMOS RAM bytes.

Warning: The operation following a write operation to this register should access the RT/CMOS Data register; otherwise, intermittent malfunctions and unreliable operation of the RT/CMOS RAM can occur.

Bit	Function
7	NMI mask
6	Reserved
5–0	RT/CMOS RAM address

Figure 3-16. RT/CMOS Address and NMI Mask Register (Hex 0070)

Bit 7 When this write-only bit is set to 1, the NMI is masked (disabled). This bit is set to 1 by a power-on reset.

Bit 6 This bit is reserved.

Bits 5–0 These bits are used to select RT/CMOS RAM addresses.

RT/CMOS Data Register (Hex 0071)

This register is used with the RT/CMOS Address and NMI Mask register to read from and write to the RT/CMOS RAM bytes.

Bit	Function
7–0	RT/CMOS data

Figure 3-17. RT/CMOS Data Register (Hex 0071)

RT/CMOS RAM I/O Operations

During I/O operations to the RT/CMOS RAM addresses, interrupts must be masked to prevent other interrupt routines from changing the CMOS Address register before data is read from or written to.

Warning: The operation following a write operation to hex 0070 must access hex 0071; otherwise intermittent malfunctions and unreliable operation of the RT/CMOS RAM can occur.

Writing to RT/CMOS RAM requires the following steps:

1. Write the RT/CMOS RAM address to the RT/CMOS and NMI Mask register (hex 0070).
2. Write the data to address hex 0071.
3. Write the address, hex 0F, to the RT/CMOS and NMI Mask register; this leaves hex 0070 pointing to the Shutdown Status byte (hex 0F).
4. Read address hex 0071 to restore the RT/CMOS (see the warning above).

Reading from RT/CMOS RAM requires the following steps:

1. Write the RT/CMOS RAM address to the RT/CMOS and NMI Mask register (hex 0070).
2. Read the data from address hex 0071.
3. Write the address, hex 0F, to the RT/CMOS and NMI Mask register; this leaves hex 0070 pointing to the Shutdown Status byte (hex 0F).
4. Read address hex 0071 to restore the RT/CMOS.

Real-Time Clock Bytes (Hex 000 – 00D)

Bit definitions and addresses for the real-time clock bytes are shown in the following figure.

Address (Hex)	Function	Byte Number
000	Seconds	0
001	Second alarm	1
002	Minutes	2
003	Minute alarm	3
004	Hours	4
005	Hour alarm	5
006	Day of week	6
007	Date of month	7
008	Month	8
009	Year	9
00A	Status Register A	10
00B	Status Register B	11
00C	Status Register C	12
00D	Status Register D	13

Figure 3-18. Real-Time Clock Bytes (Hex 000 – 00D)

Note: The system initializes Status Registers A and B when the time and date are set. Interrupt 1AH is the BIOS interface to read and set the time and date, and it initializes the registers in the same way as the system does.

Status Register A (Hex 00A)

Bit	Function
7	Update in progress
6 – 4	22-stage divider
3 – 0	Rate-selection bits

Figure 3-19. Status Register A (Hex 00A)

Bit 7 When this bit is set to 1, this bit indicates that the time-update cycle is in progress. When this bit is set to 0, it indicates that the current date and time can be read.

Bits 6 – 4 These bits identify which time-base frequency is being used. The system initializes these bits to binary 010, which selects a 32.768-kHz time base. This is the only value supported by the system for proper timekeeping.

Bits 3–0 These bits allow the selection of a divider-output frequency. The system initializes the rate-selection bits to binary 0110, which selects a 1.024-kHz square-wave output frequency and a 976.562-microsecond periodic interrupt rate.

Status Register B (Hex 00B)

Bit	Function
7	Set
6	Enable periodic-interrupt
5	Enable alarm-interrupt
4	Enable update-ended interrupt
3	Enable square wave
2	Date mode
1	24-hour mode
0	Enable daylight saving time

Figure 3-20. Status Register B (Hex 00B)

Bit 7 When this bit is set to 0, it updates the cycle by advancing the counts at a rate of one per second. When this bit is set to 1, it immediately ends any update cycle in progress, and the program can initialize the 14 time bytes without any further updates occurring until this bit is set to 0.

Bit 6 This bit is a read/write bit that allows an interrupt to occur at a rate specified by the rate and divider bits in Status register A. When this bit is set to 1, it enables the interrupt. The system initializes this bit to 0.

Bit 5 When this bit is set to 1, it enables the alarm interrupt. The system initializes this bit to 0.

Bit 4 When this bit is set to 1, it enables the update-ended interrupt. The system initializes this bit to 0.

Bit 3 When this bit is set to 1, it enables the square-wave frequency as set by the rate-selection bits in Status register A. The system initializes this bit to 0.

Bit 2 This bit indicates whether the binary-coded-decimal (BCD) or binary format is used for time-and-date calendar updates. When this bit is set to 1, it indicates the binary format. The system initializes this bit to 0.

Bit 1 This bit indicates if the hours byte is in 12-hour or 24-hour mode. When this bit is set to 1, it indicates the 24-hour mode. The system initializes this bit to 1.

Bit 0 When this bit is set to 1, it enables the daylight-saving-time mode. When this bit is set to 0, it disables the modes and the clock reverts to standard time. The system initializes this bit to 0.

Status Register C (Hex 00C)

Bit	Function
7	Interrupt-request flag
6	Periodic-interrupt flag
5	Alarm-interrupt flag
4	Update-ended interrupt flag
3–0	Reserved

Figure 3-21. Status Register C (Hex 00C)

Note: Interrupts are enabled by bits 6, 5, and 4 in Status register B.

Bit 7 When this bit is set to 1, it indicates that an interrupt has occurred; bits 6, 5, and 4 indicate the type of interrupt.

Bit 6 When this bit is set to 1, it indicates that a periodic interrupt occurred.

Bit 5 When this bit is set to 1, it indicates that an alarm interrupt occurred.

Bit 4 When this bit is set to 1, it indicates that an update-ended interrupt occurred.

Bits 3–0 These bits are reserved.

Status Register D (Hex 00D)

Bit	Function
7	Valid RAM
6–0	Reserved

Figure 3-22. Status Register D (Hex 00D)

Bit 7 This read-only bit monitors the internal battery. When this bit is set to 1, it indicates that the real-time clock has power. When this bit is set to 0, it indicates that the real-time clock has lost power.

Bits 6–0 These bits are reserved.

CMOS RAM Configuration

The following shows the bit definitions for the CMOS RAM configuration bytes.

Diagnostic Status Byte (Hex 00E)

Bit	Function
7	Real-time clock power
6	Configuration record and checksum status
5	Incorrect configuration
4	Memory-size mismatch
3	Fixed disk controller/drive C initialization status
2	Time status indicator
1	Adapter configuration miscompare
0	Adapter ID time-out

Figure 3-23. Diagnostic Status Byte (Hex 00E)

Bit 7 When this bit is set to 1, it indicates that the real-time clock lost power.

Bit 6 When this bit is set to 1, it indicates that a checksum is incorrect.

Bit 5 When this bit is set to 1, it indicates that the configuration information is incorrect.

Bit 4 When this bit is set to 1, it indicates that the memory size does not match the configuration information.

Bit 3 When this bit is set to 1, it indicates that the controller or fixed disk drive failed initialization.

Bit 2 When this bit is set to 1, it indicates that the time and date are invalid.

Bit 1 When this bit is set to 1, it indicates that the adapters do not match the configuration information.

Bit 0 When this bit is set to 1, it indicates that a time-out occurred while an adapter ID was being read.

Shutdown Status Byte (Hex 00F): This byte is used by POST and BIOS.

Diskette Drive Type Byte (Hex 010): This byte indicates the type of diskette drive installed.

Bit	Function
7–4	First diskette drive type
3–0	Second diskette drive type

Figure 3-24. Diskette Drive Type Byte (Hex 010)

Bits 7–4 These bits indicate the first diskette drive type, as shown in the following figure.

Bits 7 6 5 4	Function
0 0 0 0	No drive present
0 0 0 1	5.25-inch, 360KB diskette drive
0 0 1 0	5.25-inch, 1.2MB diskette drive
0 0 1 1	3.5-inch, 720KB diskette drive
0 1 0 0	3.5-inch, 1.44MB diskette drive
0 1 1 0	3.5-inch, 2.88MB diskette drive

Note: All combinations that are not shown are reserved.

Figure 3-25. Diskette Drive Type Byte (Bits 7–4)

Bits 3–0 These bits indicate the second diskette drive type, as shown in the following figure.

Bits 3 2 1 0	Function
0 0 0 0	No drive present
0 0 0 1	5.25-inch, 360KB diskette drive
0 0 1 0	5.25-inch, 1.2MB diskette drive
0 0 1 1	3.5-inch, 720KB diskette drive
0 1 0 0	3.5-inch, 1.44MB diskette drive
0 1 1 0	3.5-inch, 2.88MB diskette drive

Note: All combinations that are not shown are reserved.

Figure 3-26. Diskette Drive Type Byte (Bits 3–0)

Fixed Disk Drive Type Byte (Hex 011): This byte defines the type of fixed disk drive installed. Hex 000 indicates that a fixed disk drive is *not* installed. This byte is not used for SCSI fixed disk drives.

Note: For more information about fixed disk drives, refer to the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference.*

Reserved Bytes (Hex 012, 013): These bytes are reserved.

Equipment Byte (Hex 014): This byte defines the basic equipment in the system for the power-on diagnostic tests.

Bit	Function
7, 6	Number of diskette drives other than physical drive 0
5, 4	Display operating mode
3, 2	Reserved
1	Math coprocessor presence
0	Diskette drive 0 presence

Figure 3-27. Equipment Byte (Hex 014)

Bits 7, 6 These bits indicate the number of diskette drives other than drive 0 installed, as shown in the following figure.

Bits 7 6	Number of Diskette Drives
0 0	No other drive
0 1	One other drive
1 0	Two other drives
1 1	Reserved

Figure 3-28. Installed Diskette Drive Bits

Bits 5, 4 These bits indicate the operating mode of the display attached to the video port, as shown in the following figure.

Bits 5 4	Display Operating Mode
0 0	Reserved
0 1	40-column mode
1 0	80-column mode
1 1	Monochrome mode

Figure 3-29. Display Operating-Mode Bits

Bits 3, 2 These bits are reserved.

Bit 1 When this bit is set to 1, it indicates that a math coprocessor is installed.

Bit 0 When this bit is set to 1, it indicates that physical diskette drive 0 is installed.

Low and High Base-Memory Bytes (Hex 015 and 016): These bytes define the amount of memory below the 640KB address space.

The hexadecimal value in these bytes represents the number of 1KB blocks of *base memory*. For example, hex 0280 indicates 640KB. The low byte is hex 015; the high byte is hex 016.

Low and High Expansion-Memory Bytes (Hex 017 and 018): These bytes define the amount of memory above the 1MB address space.

The hexadecimal value in these bytes represent the number of 1KB blocks of *expansion memory*. For example, hex 0800 indicates 2048KB. The low byte is hex 017; the high byte is hex 018.

Reserved Bytes (Hex 019 through 031): These bytes are reserved.

Configuration CRC Bytes (Hex 032 and 033): These bytes contain the cyclic-redundancy-check (CRC) data for bytes hex 010 through hex 031 of the 64-byte CMOS RAM. The low byte is hex 033; the high byte is hex 032.

Reserved Byte (Hex 034): This byte is reserved.

Low and High Usable Memory Bytes (Hex 035 and 036): These bytes define the total amount of contiguous memory from 1MB to 16MB.

The hexadecimal value in these bytes represent the number of 1KB blocks of usable memory. For example, hex 0800 indicates 2408KB. The low byte is hex 035; the high byte is hex 036.

Date Century Byte (Hex 037): Bits 7 through 0 of this byte contain the BCD value for the century. Refer to the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference* for information about reading and setting this byte.

Reserved Bytes (Hex 038 – 03F): These bytes are reserved.

NVRAM

The NVRAM contains, among other things, the system configuration information. The entire RAM is reserved for system use.

Miscellaneous System Functions and Ports

Nonmaskable Interrupt

The nonmaskable interrupt (NMI) signals the system microprocessor that a parity error, a channel check, a system channel time-out, or a system watchdog time-out has occurred. The NMI (except cache parity error) stops all arbitration on the bus until bit 6 of the Arbitration register (I/O address hex 0090) is set to 0. This can result in lost data or an overrun error on some I/O devices. The NMI masks all other interrupts, and the IRET instruction restores the interrupt flag to the state it was in before the interrupt. A system reset causes a reset of the NMI.

The NMI requests from system board parity and channel check are subject to mask control with the NMI mask bit in the RT/CMOS Address register. The watchdog timer, system channel time-out, and cache parity error are not masked by this bit. See "RT/CMOS Address and NMI Mask Register (Hex 0070)" on page 3-19. The power-on default of the NMI mask is 1 (NMI disabled). Before the NMI is enabled after a power-on reset, the parity-check and channel-check states are initialized by POST.

Warning: The operation following a write operation to this register must access the RT/CMOS Data register; otherwise, intermittent malfunctions and unreliable operation of the RT/CMOS RAM can occur.

System Control Port B (Hex 0061)

Bit definitions for the write and read functions of this port are shown in the following figures.

Bit	Function
7	Reset timer 0 output latch (IRQ0)
6–4	Reserved
3	Enable channel check
2	Enable parity check
1	Enable speaker data
0	Timer 2 gate to speaker

Figure 3-30. System Control Port B (Hex 0061) – Write

Bit	Function
7	Parity check
6	Channel check
5	Timer 2 output
4	Toggles with each refresh request
3	Enable channel check
2	Enable parity check
1	Enable speaker data
0	Timer 2 gate to speaker

Figure 3-31. System Control Port B (Hex 0061) – Read

Bit 7 When this bit is set to 1 during a write operation, IRQ0 is reset. When read as a 1, this bit indicates that a parity check has occurred.

Bit 6 When read as a 1, this bit indicates that a channel check has occurred.

Bit 5 When read, this bit indicates the condition of the timer/counter 2 'output' signal.

Bit 4 When read, this bit toggles for each refresh request.

Bit 3 When this bit is set to 0, it enables a channel check. This bit is set to 1 during a power-on reset.

Bit 2 When this bit is set to 0, it enables a parity check. This bit is set to 1 during a power-on reset.

Bit 1 When this bit is set to 1, it enables speaker data.

Bit 0 When this bit is set to 1, it enables the timer 2 gate.

System Control Port A (Hex 0092)

Bit	Function
7 – 5	Reserved
4	Watchdog timer status
3	Security lock latch
2	Reserved
1	Alternate gate A20
0	Alternate hot reset

Figure 3-32. System Control Port A (Hex 0092)

Bits 7 – 5 These bits are reserved.

Bit 4 This read-only bit indicates the watchdog-timer status. When this bit is set to 1, a watchdog time-out has occurred. For more information about the watchdog timer, refer to the System Timers section of the *IBM Personal System/2 Hardware Interface Technical Reference—Common Interfaces.*

Bit 3 This bit provides the security lock for the secured area of RT/CMOS. When this bit is set to 1, the 8-byte, power-on password is electronically locked. After this bit is set by POST, it can be cleared only by turning the system off.

Bit 2 This bit is reserved.

Bit 1 This bit is ORed with bit 1 in the Keyboard Output port to enable the 'address 20' signal (A20). When both bits are set to 0, A20 is disabled and set to 0. This bit is set to 0 during POST. See the Keyboard and Auxiliary Device Controller section of the *IBM Personal System/2 Hardware Interface Technical Reference—Common Interfaces.*

Bit 0 This bit provides an alternative method of resetting the system microprocessor. This alternative method supports operating systems that require faster operation than was provided on the IBM Personal Computer AT*. Resetting the system microprocessor switches the microprocessor from protected mode to real-address mode. The alternative reset takes 13.4 microseconds.

This bit is set to 0 by either a system reset or a write operation. When a write operation changes this bit from 0 to 1, the alternative reset pin is pulsed high for 100 to 125 nanoseconds. The reset occurs after a minimum delay of 6.72 microseconds. While the reset is occurring, the latch remains set so that POST can read this bit. If the bit is set to 0, POST assumes that the system was just powered-on. If the bit is set to 1, POST assumes that a switch from the protected mode to the real mode has taken place.

When bit 0 is used to reset the system microprocessor to the real mode, the following procedure *must* be used:

1. Disable all maskable and nonmaskable interrupts.
2. Reset the system microprocessor by writing a 1 to bit 0.
3. Issue a Halt instruction to the system microprocessor.

---

* Trademark of the IBM Corporation.

4. Reenable all maskable and nonmaskable interrupts.

If this procedure is not followed, the results will be unpredictable.

Note: Whenever possible, BIOS or ABIOS should be used as an interface to reset the system microprocessor to the real mode. See *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference* for more information about BIOS.

Power-On Password

RT/CMOS RAM has 8 bytes reserved for the power-on password and its check character. The 8 bytes are initialized to hex 00. The microprocessor can access these bytes only during power-on self-test (POST). After POST is completed, if a power-on password is installed, the password bytes are locked and cannot be accessed by a program. During power-on password installation, the password (1 to 7 characters), is stored in the security space.

Installing the password is a function of a program on the Reference Diskette. The power-on password does not appear on the screen when it is installed, changed, or removed. After the power-on password has been installed, it can be changed or removed only during POST.

The Model 57 systems also can have a keyboard password. See the Keyboard and Auxiliary Device Controller section of the *IBM Personal System/2 Hardware Interface Technical Reference—Common Interfaces* for more information.

Selectable Boot Sequence

With selectable boot you can specify the order in which bootable devices are searched when looking for the boot startup record of an operating system. Diskette drives, fixed disk drives, and remote initial program load (RIPL) can be part of the search sequence. Changing the sequence for selectable boot affects the drive-letter assignment each time the system is booted (started).

You can select the boot sequence by using the Select startup sequence menu on the System Reference Diskette, and selecting from a list of possible devices. After the sequence is selected, it is stored in nonvolatile RAM (NVRAM). When the system is booted, it searches for a device that contains a bootable record, using the specified sequence.

After the operating system is booted, the drive lettering is determined by the sequence and by which device contained the bootable source.

Note: Selectable boot affects adapters that use initialization programs. See the Setup section of *IBM Personal System/2 Hardware Interface Technical Reference — Architectures* for more details.

Hardware Compatibility

The Model 57 systems maintain many of the interfaces used by previous PS/2 products. In most cases, command and status organization of these interfaces is maintained.

The functional interfaces for the Model 57 systems are compatible with the following:

- The Intel** 8259 interrupt controllers (without edge triggering).
- The Intel 8253 timers driven from 1.193 MHz (timer 0 and 2 only).
- The Intel 8237 DMA controller-address/transfer counters, page registers and status fields only. The Command and Request registers are not supported. The Rotate and Mask functions are not supported. The Mode register is partially supported.
- The NS16450 serial port.
- The Intel 8088, 8086, 80286, and 80386 microprocessors.
- The Intel 8272 diskette drive controller.
- The Motorola** MC146818 Time of Day Clock command and status (CMOS reorganized).
- The keyboard interface at address hex 0060.

** Intel is a trademark of Intel Corporation, and Motorola is a trademark of Motorola, Incorporated.

- Display modes supported by the IBM Monochrome Display and Printer Adapter, the IBM Color/Graphics Monitor Adapter, and the IBM Enhanced Graphics Adapter.
- The parallel printer ports (Parallel 1, Parallel 2, and Parallel 3) in DOS mode.
- Generally compatible with the Intel 80387, 80287, and 8087 math coprocessors.

Error Codes

POST returns a message in the form of a multiple-character code to indicate the type of test that failed. Two formats are used for these error messages:

- A 12-character code for all errors *that are* related to SCSI devices:
  - The first four characters identify the device type.
  - The next eight characters identify specifics of the error.
- An 8-character error code for all errors *that are not* related to SCSI devices.

The 8-character error code consists of two 4-character elements: the major error code and the minor error code. The following figure lists the failure and the associated error code.

Major Code	Minor Code	Description
0001		System bus error - system board
	0100	Unexpected interrupt
	0200	Timer failure
	0300	Timer interrupt failure
	0400	Protected mode failure
	0500	Last keyboard command not accepted
	0600	Converting logic test
	0700	NMI test failed
	0800	Timer bus-test failed
	0900	Low MB chip select test failed
	10*xx*	System board memory parity error
	11*xx*	I/O channel-check error
	1200	Watchdog time-out
	1300	DMA arbitration time-out
	1400	Channel feature card ROM checksum error
	1500	ROM checksum error or DMA error
	1600	System board port read/write failure
	1800	System board parity
	2000	Microprocessor test error
	6000	System board ID not recognized
	6100	Dead battery
	6300	Clock not updating
	6400	Memory-configuration error
	6500	Adapter ID mismatch
	6600	Adapter-busy error
	6700	Clock not updating
	6900	System board and processor card configuration mismatch
	7100	Rolling-bit-test failure on CMOS shutdown address byte
	7200	Rolling-bit-test failure on NVRAM diagnostic byte
	7300	Bad CMOS/NVRAM checksum
	7400	Bad configuration
	7900	NVRAM error log full
0002		Memory errors
	0*xxx*	Memory error
	1*xxx*	Unrecoverable error in first 1MB
	2100	ROM-to-RAM remapping error
	25*xx*	Unsupported memory-type installed
	2600	Memory installed in unsupported position

Figure 3-33 (Part 1 of 2). POST Error Messages

Major Code	Minor Code	Description
0003		Keyboard
	0100	Keyboard error
	0200	Keyboard locked
	0300	Keyboard-to-system board interface error
	0400	Keyboard clock high
	0500	No keyboard +5 V dc
0006		Diskette
	0100	Diskette drive or controller error
	0200	Diskette IPL boot record not valid
	0400	Non-media sense
0011	0100	Async-port error
0017		Fixed disk error
	8000	Fixed disk 0 failed
	8100	Fixed disk 1 failed
	8200	Disk-controller error
	9000	Fixed disk 0 error
	9100	Fixed disk 1 error
0037	*xxxx*	System board SCSI controller error
0024	0100	System board video error
0086		Mouse
	0100	System-bus error - keyboard/pointing device interface
	0200	Pointing-device error
	0300	Pointing-device or system-bus error
0096	*xxxx*	SCSI adapter with cache error
0112	*xxxx*	SCSI adapter error
0129	0200	Cache error
0208 to 0241	*xxxx*	SCSI device test failure
1999		Special instruction codes
	00*xx*	Initial microcode load (IML) error
	03*xx*	No bootable device
	04*xx*	IML-to-system mismatch

Figure 3-33 (Part 2 of 2). POST Error Messages

Notes:

Appendix. 386SLC Instruction Cycle Times

The following are the instruction timings for the IBM 386SLC. All cache-off timings are considered cache-miss situations. Any instructions not listed are the same as for the Intel 80386SX.

Mnemonic	16-Bit Real Mode Cache-Miss Cycles	16-Bit Real Mode Cache-Hit Cycles	16-Bit Protected Mode Cache-Miss Cycles	16-Bit Protected Mode Cache-Hit Cycles
adc mb,ib	7	5	7	5
adc mb,rb	7	5	7	5
adc mw,iw	7	5	7	5
adc mw,rw	7	5	7	5
adc rb,ib	2	2	2	2
adc rb,mb	6	4	6	4
adc rb,rb	2	2	2	2
adc rw,iw	2	2	2	2
adc rw,mw	6	4	6	4
adc rw,rw	2	2	2	2
add mb,ib	7	5	7	5
add mb,rb	7	5	7	5
add mw,iw	7	5	7	5
add mw,rw	7	5	7	5
add rb,ib	2	2	2	2
add rb,mb	5	3	5	3
add rb,rb	2	2	2	2
add rw,iw	2	2	2	2
add rw,mw	5	3	5	3
add rw,rw	2	2	2	2
and mb,ib	7	5	7	5
and mb,rb	7	5	7	5
and mw,iw	7	5	7	5
and mw,rw	7	5	7	5
and rb,ib	2	2	2	2
and rb,mb	5	3	5	3
and rb,rb	2	2	2	2
and rw,iw	2	2	2	2
and rw,mw	5	3	5	3
and rw,rw	2	2	2	2
bound rdw,mqw	15	10	15	10
bound rw,mdw	11	10	11	10
bsf rdw,mdw	40	37	40	37
bsf rdw,rdw	35	35	35	35
bsf rw,mw	26	25	26	25
bsf rw,rw	23	23	23	23

Figure A-1 (Part 1 of 8). Instruction Cycle Times

© Copyright IBM Corp. 1991, 1992

Mnemonic	16-Bit Real Mode Cache-Miss Cycles	16-Bit Real Mode Cache-Hit Cycles	16-Bit Protected Mode Cache-Miss Cycles	16-Bit Protected Mode Cache-Hit Cycles
bsr rdw,mdw	38	35	38	35
bsr rdw,rdw	33	33	33	33
bsr rw,mw	24	23	24	23
bsr rw,rw	21	21	21	21
bt mdw,ib	8	5	8	5
bt mdw,rdw	14	11	14	11
bt mw,ib	6	5	6	5
bt mw,rw	12	11	12	11
bt rdw,ib	3	3	3	3
bt rdw,rdw	3	3	3	3
bt rw,ib	3	3	3	3
bt rw,rw	3	3	3	3
btc mdw,ib	9	8	9	8
btc mdw,rdw	15	13	15	13
btc mw,ib	8	8	8	8
btc mw,rw	13	13	13	13
btc rdw,ib	6	6	6	6
btc rdw,rdw	6	6	6	6
btc rw,ib	6	6	6	6
btc rw,rw	6	6	6	6
btr mdw,ib	9	8	9	8
btr mdw,rdw	15	13	15	13
btr mw,ib	8	8	8	8
btr mw,rw	13	13	13	13
btr rdw,ib	6	6	6	6
btr rdw,rdw	6	6	6	6
btr rw,ib	6	6	6	6
btr rw,rw	6	6	6	6
bts mdw,ib	9	8	9	8
bts mdw,rdw	15	13	15	13
bts mw,ib	8	8	8	8
bts mw,rw	13	13	13	13
bts rdw,ib	6	6	6	6
bts rdw,rdw	6	6	6	6
bts rw,ib	6	6	6	6
bts rw,rw	6	6	6	6
call cs:ip	20	14	32	19
call disp	9	5	9	5
call far m	27	17	39	22
call m	14	10	14	10
call rw	11	7	11	7
cli	5	5	5	5
cmc	2	2	2	2

Figure A-1 (Part 2 of 8). Instruction Cycle Times

Mnemonic	16-Bit Real Mode Cache-Miss Cycles	16-Bit Real Mode Cache-Hit Cycles	16-Bit Protected Mode Cache-Miss Cycles	16-Bit Protected Mode Cache-Hit Cycles
cmp mb,ib	5	3	5	3
cmp mb,rb	5	3	5	3
cmp mw,iw	5	3	5	3
cmp mw,rw	5	3	5	3
cmp rb,ib	2	2	2	2
cmp rb,mb	6	4	6	4
cmp rb,rb	2	2	2	2
cmp rw,iw	2	2	2	2
cmp rw,mw	6	4	6	4
cmp rw,rw	2	2	2	2
cmpsb	12	10	12	10
cmpsw	12	10	12	10
cmpsd	16	10	16	10
dec mb	6	4	6	4
dec mw	6	4	6	4
dec rb	2	2	2	2
dec rw	2	2	2	2
enter	7	7	7	7
imul mb	6	4	6	4
imul mw	6	4	6	4
imul rb	2	2	2	2
imul rw	2	2	2	2
imul rw,rw,iw	19	19	19	19
in rb,dx	14	14	7	7
in rb,ib	12	12	6	6
in rw,dx	14	14	7	7
in rw,ib	12	12	6	6
inc mb	6	4	6	4
inc mw	6	4	6	4
inc rb	2	2	2	2
inc rw	2	2	2	2
int ib	43	35	76	64
iret	31	26	42	30
jcond not taken	3	3	3	3
jcond taken	9	5	9	5
jcxz not taken	3	3	3	3
jcxz taken	9	5	9	5
jmp cs:ip	16	12	27	16
jmp disp	9	5	9	5
jmp far m	23	15	34	19
jmp m	13	8	13	8
jmp rw	10	6	10	6
lds rw,mw	7	6	23	12

Figure A-1 (Part 3 of 8). Instruction Cycle Times

Mnemonic	16-Bit Real Mode Cache-Miss Cycles	16-Bit Real Mode Cache-Hit Cycles	16-Bit Protected Mode Cache-Miss Cycles	16-Bit Protected Mode Cache-Hit Cycles
leave	5	4	5	4
les rw,mw	7	6	23	12
lodsb	5	5	5	5
lodsd	7	5	7	5
lodsw	5	5	5	5
loop not taken	11	9	11	9
loop taken	9	5	9	5
loopc not taken	14	12	14	12
loopc taken	12	8	12	8
mov m,s	2	2	2	2
mov mb,ib	2	2	2	2
mov mb,rb	2	2	2	2
mov mw,iw	2	2	2	2
mov mw,rw	2	2	2	2
mov rb,ib	2	2	2	2
mov rb,mb	4	2	4	2
mov rb,rb	2	2	2	2
mov rw,iw	2	2	2	2
mov rw,mw	4	2	4	2
mov rw,rw	2	2	2	2
mov rw,s	2	2	2	2
mov s,m	5	4	18	9
mov s,rw	2	2	15	8
movsb	9	8	9	8
movsd	11	8	11	8
movsw	9	8	9	8
movsx rdw,mb	6	5	6	5
movsx rdw,mw	6	5	6	5
movsx rw,mb	6	5	6	5
movzx rdw,mb	6	5	6	5
movzx rdw,mw	6	5	6	5
movzx rw,mb	6	5	6	5
mul mb	16	15	16	15
mul mw	16	15	16	15
mul rb	13	13	13	13
mul rw	13	13	13	13
neg mb	6	4	6	4
neg mw	6	4	6	4
neg rb	2	2	2	2
neg rw	2	2	2	2
not mb	6	4	6	4
not mw	6	4	6	4
not rb	2	2	2	2
not rw	2	2	2	2

Figure A-1 (Part 4 of 8). Instruction Cycle Times

Mnemonic	16-Bit Real Mode Cache-Miss Cycles	16-Bit Real Mode Cache-Hit Cycles	16-Bit Protected Mode Cache-Miss Cycles	16-Bit Protected Mode Cache-Hit Cycles
or mb,ib	7	5	7	5
or mb,rb	7	5	7	5
or mw,iw	7	5	7	5
or mw,rw	7	5	7	5
or rb,ib	2	2	2	2
or rb,mb	5	3	5	3
or rb,rb	2	2	2	2
or rw,iw	2	2	2	2
or rw,mw	5	3	5	3
or rw,rw	2	2	2	2
out dx,rb	12	12	6	6
out dx,rw	12	12	6	6
out ib,rb	11	11	5	5
out	11	11	5	5
pop m	6	5	6	5
pop rw	4	2	4	2
pop s	5	4	18	9
popa	25	24	25	24
popad	33	24	66	65
popf	8	7	6	5
push m	6	5	6	5
push rw	2	2	2	2
push s	2	2	2	2
push iw	2	2	2	2
pusha	24	24	24	24
pushad	32	25	66	66
pushf	7	7	5	5
rep:cmpsb	10+4*n*†	9+2*n*	10+4*n*	9+2*n*
rep:cmpsw	10+4*n*	9+2*n*	10+4*n*	9+2*n*
rep:cmpsd	10+8*n*	9+2*n*	10+8*n*	9+2*n*
rep:lodsb	5+6*n*	5+6*n*	5+6*n*	5+6*n*
rep:lodsd	5+8*n*	5+6*n*	5+8*n*	5+6*n*
rep:lodsw	5+6*n*	5+6*n*	5+6*n*	5+6*n*
rep:movsb	8+5*n*	8+2*n*	8+5*n*	8+2*n*
rep:movsw	8+5*n*	8+2*n*	8+5*n*	8+2*n*
rep:movsd	8+8*n*	8+2*n*	8+8*n*	8+2*n*
rep:scasb	8+4*n*	8+2*n*	8+4*n*	8+2*n*
rep:scasw	8+4*n*	8+2*n*	8+4*n*	8+2*n*
rep:scasd	8+6*n*	8+2*n*	8+6*n*	8+2*n*

† *n* represents the number of additional cycles.

Figure A-1 (Part 5 of 8). Instruction Cycle Times

Mnemonic	16-Bit Real Mode Cache-Miss Cycles	16-Bit Real Mode Cache-Hit Cycles	16-Bit Protected Mode Cache-Miss Cycles	16-Bit Protected Mode Cache-Hit Cycles
rep:stosb	5+2*n*	5+2*n*	5+2*n*	5+2*n*
rep:stosw	5+2*n*	5+2*n*	5+2*n*	5+2*n*
rep:stosd	5+4*n*	5+2*n*	5+4*n*	5+2*n*
ret	11	7	11	7
ret iw	11	7	11	7
retf	21	13	33	19
retf iw	21	13	33	19
rol mb,cl	7	6	7	6
rol mb,ib	7	6	7	6
rol mw,cl	7	6	7	6
rol mw,ib	7	6	7	6
rol rb,cl	3	3	3	3
rol rb,ib	3	3	3	3
rol rw,cl	3	3	3	3
rol rw,ib	3	3	3	3
ror mb,cl	7	6	7	6
ror mb,ib	7	6	7	6
ror mw,cl	7	6	7	6
ror mw,ib	7	6	7	6
ror rb,cl	3	3	3	3
ror rb,ib	3	3	3	3
ror rw,cl	3	3	3	3
ror rw,ib	3	3	3	3
sahf	3	3	3	3
sal mb,cl	7	6	7	6
sal mb,ib	7	6	7	6
sal mw,cl	7	6	7	6
sal mw,ib	7	6	7	6
sal rb,cl	3	3	3	3
sal rb,ib	3	3	3	3
sal rw,cl	3	3	3	3
sal rw,ib	3	3	3	3
sar mb,cl	7	6	7	6
sar mb,ib	7	6	7	6
sar mw,cl	7	6	7	6
sar mw,ib	7	6	7	6
sar rb,cl	3	3	3	3
sar rb,ib	3	3	3	3
sar rw,cl	3	3	3	3
sar rw,ib	3	3	3	3

Figure A-1 (Part 6 of 8). Instruction Cycle Times

Mnemonic	16-Bit Real Mode Cache-Miss Cycles	16-Bit Real Mode Cache-Hit Cycles	16-Bit Protected Mode Cache-Miss Cycles	16-Bit Protected Mode Cache-Hit Cycles
sbb mb,ib	7	5	7	5
sbb mb,rb	7	5	7	5
sbb mw,iw	7	5	7	5
sbb mw,rw	7	5	7	5
sbb rb,ib	2	2	2	2
sbb rb,mb	6	4	6	4
sbb rb,rb	2	2	2	2
sbb rw,iw	2	2	2	2
sbb rw,mw	6	4	6	4
sbb rw,rw	2	2	2	2
scasb	8	7	8	7
scasd	10	7	10	7
scasw	8	7	8	7
shl mb,cl	7	6	7	6
shl mb,ib	7	6	7	6
shl mw,cl	7	6	7	6
shl mw,ib	7	6	7	6
shl rb,cl	3	3	3	3
shl rb,ib	3	3	3	3
shl rw,cl	3	3	3	3
shl rw,ib	3	3	3	3
shld mw,rw,cl	7	6	7	6
shld mw,rw,ib	7	6	7	6
shld rw,rw,cl	3	3	3	3
shld rw,rw,ib	3	3	3	3
shld mdw,rdw,cl	9	6	9	6
shld mdw,rdw,ib	9	6	9	6
shld rdw,rdw,cl	3	3	3	3
shld rdw,rdw,ib	3	3	3	3
shr mb,cl	7	6	7	6
shr mb,ib	7	6	7	6
shr mw,cl	7	6	7	6
shr mw,ib	7	6	7	6
shr rb,cl	3	3	3	3
shr rb,ib	3	3	3	3
shr rw,cl	3	3	3	3
shr rw,ib	3	3	3	3
shrd mw,rw,cl	7	6	7	6
shrd mw,rw,ib	7	6	7	6
shrd rw,rw,cl	3	3	3	3
shrd rw,rw,ib	3	3	3	3
shrd mdw,rdw,cl	9	6	9	6
shrd mdw,rdw,ib	9	6	9	6
shrd rdw,rdw,cl	3	3	3	3
shrd rdw,rdw,ib	3	3	3	3
stc	2	2	2	2

Figure A-1 (Part 7 of 8). Instruction Cycle Times

Mnemonic	16-Bit Real Mode Cache-Miss Cycles	16-Bit Real Mode Cache-Hit Cycles	16-Bit Protected Mode Cache-Miss Cycles	16-Bit Protected Mode Cache-Hit Cycles
std	2	2	2	2
sti	5	5	5	5
stosb	5	5	5	5
stosd	5	5	5	5
stosw	5	5	5	5
sub mb,ib	7	5	7	5
sub mb,rb	7	5	7	5
sub mw,iw	7	5	7	5
sub mw,rw	7	5	7	5
sub rb,ib	2	2	2	2
sub rb,mb	6	4	6	4
sub rb,rb	2	2	2	2
sub rw,iw	2	2	2	2
sub rw,mw	6	4	6	4
sub rw,rw	2	2	2	2
test mb,ib	5	3	5	3
test mw,iw	5	3	5	3
test rb,ib	2	2	2	2
test rb,mb	5	3	5	3
test rb,rb	2	2	2	2
test rw,iw	2	2	2	2
test rw,mw	5	3	5	3
test rw,rw	2	2	2	2
xlat	5	4	5	4
xor mb,ib	7	5	7	5
xor mb,rb	7	5	7	5
xor mw,iw	7	5	7	5
xor mw,rw	7	5	7	5
xor rb,ib	2	2	2	2
xor rb,mb	5	3	5	3
xor rb,rb	2	2	2	2
xor rw,iw	2	2	2	2
xor rw,mw	5	3	5	3
xor rw,rw	2	2	2	2

Figure A-1 (Part 8 of 8). Instruction Cycle Times

Index

A

adapter
  setup 2-4
adapter setup register 2-15
address and NMI mask register, RT/CMOS 3-19
address map
  memory 3-16
  POS I/O 2-5
  system board I/O 1-10
address 20 gate 3-30
alternate reset 3-30
altitude 1-14, 1-15
anticipated page miss 1-13
arbiter programming 3-8
arbitration control point 3-6
arbitration level 3-7
arbitration register 3-8
arbitration, central control point 3-8
assistance, developer 3-33

B

base memory bytes 3-27
battery 3-23
boot sequence 3-31
bus adapter 3-9
bus contention 2-3
bus priority assignments 3-7

C

cables 1-14, 1-15
cache
  address space 3-5
  operation 3-4
card selected feedback register 2-6
-CD SFDBK 2-6
central arbitration control point 3-6
channel 1-4, 3-6
  check enable 3-29
  check latch 3-29
  connector 3-9
  DMA 3-7
clock, real-time 3-18, 3-21
CMOS RAM 3-18—3-27
codes, POST error 3-33
compatibility 3-32
configuration
  CRC bytes 3-27
  RT/CMOS 3-24
connector
  channel 3-9
  drives 3-10
  memory 3-11, 3-12
contention, bus 2-3
control ports, system 3-28—3-30
controller
coprocessor, math 3-6
current, electrical 1-14, 1-15

D

data register, RT/CMOS 3-20
date century byte 3-27
date status indicator 3-24
depth, system unit 1-14, 1-15
description, power supply 1-16
developer assistance 3-33
diagnostic status byte (RT/CMOS) 3-24
diskette drive
diskette drive connector 3-10
diskette drive connectors 3-10
diskette drive information, POS 2-11
diskette drive type byte 3-24
diskette overruns 3-7
display operating mode 3-26
DMA
  channels 3-7
  performance 1-13

© Copyright IBM Corp. 1991, 1992

E

enable NMI 3-19
equipment byte 3-26
error codes, POST 3-33
extended arbitration 3-9

F

failures and messages, POST 3-33
fault, overvoltage 1-18
feedback, card selected 2-6
fixed disk drive
  initialization 3-24
  type byte 3-25

G

gate A20 3-30

H

height, system unit 1-14, 1-15
humidity 1-14, 1-15

I

instruction cycle times A-1
interleaving, memory 3-15
  mem.interleaving 3-15
interrupt
  IRQ0 reset 3-29
  NMI reset 3-28
I/O
  address map 1-10
  RT/CMOS RAM operations 3-20

K

keyboard
  cable 1-14, 1-15
  password 3-31

L

low and high usable memory
  byte 3-27

M

math coprocessor 3-6
measurements, system unit 1-14, 1-15
memory
  bytes 3-27
  connectors 3-11, 3-12
  description 3-13
  performance 1-13
  RAM 3-14
  read 1-13
  refresh 3-7
  ROM 3-14
  RT/CMOS RAM 3-18
  size miscompare 3-24
  system memory map 3-16
  write 1-13
memory information, POS 2-10
memory-presence detect signals 3-15
messages, POST error 3-33
Micro Channel 1-4
Micro Channel architecture
  arbitration control point 3-6
  bus adapter 3-9
  general information 3-6
microprocessor
  alternate reset 3-30
  mode switch compatibility 3-30
  performance 1-13
  386SLC 3-3
  386SLC instructions 3-4
  386SLC programming considerations 3-4
  80386SX 3-3
mode switch, protected 3-30

N

NMI mask register 3-19
NMI service 3-7
NMI signal 3-7
nonmaskable interrupt (NMI) 3-28
nonvolatile RAM (NVRAM) 3-27

O

operating system boot
  sequence 3-31
output voltage sequencing 1-18
outputs, power supply 1-17
overvoltage fault 1-18

P

page hit 1-13
page miss 1-13
parallel port select 2-9
parity check enable 3-29
password
  keyboard 3-31
  power-on 3-30, 3-31
performance, system 1-13
port
  system 3-28
POS
  See programmable option select
    (POS)
POS address map 2-5
POS register 3 2-10
POS register, diskette drive 2-11
POST error codes 3-33
POST (power-on self-test) 3-14
power
  cable 1-14, 1-15
  loss 3-23
  specifications 1-14, 1-15
power supply 1-4
power supply description 1-16
power supply output
  protection 1-18
power supply outputs 1-17
power-on password 3-30, 3-31
power-on self-test (POST) 3-14
programmable option select (POS)
  address map 2-5
  card selected feedback
    register 2-6
  description 2-3
  setup procedures 2-4
  system board enable/setup
    register 2-7
programmable option select (POS)
  (continued)
  system board setup 2-6
  system configuration
    utilities 2-3
  system memory map 3-16
  video 2-8
  warning 2-3
programming, arbiter 3-8
protected mode switch 3-30
protection, power supply 1-18

R

RAM (random access
  memory) 3-14
random access memory
  (RAM) 3-14
read-only memory (ROM) 3-14
real mode switch 3-30
real-time clock 3-18, 3-21
refresh 3-7
refresh request 1-13, 3-29
register
  adapter setup 2-15
  card selected feedback 2-6
  miscellaneous system 3-28
  RT/CMOS address and NMI
    mask 3-19
  RT/CMOS data 3-20
  RT/CMOS status 3-21—3-23
  system board enable/setup 2-7
  system board POS 2-5
  system control 3-28
reset, alternate 3-30
ROM (read-only memory) 3-14
RT/CMOS RAM
  address and NMI mask
    register 3-19
  address map 3-19
  configuration bytes 3-24
  data register 3-20
  description 3-18
  diagnostic status byte 3-24
  I/O operations 3-20
  power-on password 3-31
  secured area 3-30

RT/CMOS RAM *(continued)*
status registers 3-21—3-23

S

selectable boot 3-31
sequencing, output voltage 1-18
set configuration 3-31
setup
bus contention 2-3
system board 2-6
system board group 1 2-4
system board group 2 2-4
system board group 3 2-5
system board SCSI controller setup 2-5
system memory map 3-16
video 2-8
shutdown status byte 3-24
signal
card selected feedback 2-6
size, system unit 1-14, 1-15
speaker data enable 3-29
specifications
performance 1-13
physical 1-14, 1-15
startup sequence 3-31
status registers
(RT/CMOS) 3-21—3-23
switch to real mode 3-30
system
configuration utilities 2-3
control ports 3-28—3-30
functions, miscellaneous 3-28
performance 1-13
reset 3-28
specifications 1-14, 1-15
system board
address map 1-10
description 3-3
enable/setup register 2-7
memory connector 3-11, 3-12
microprocessors 3-3
POS I/O address map 2-5
setup 2-4, 2-5, 2-6
system description 1-3

T

temperature 1-14, 1-15
time status indicator 3-24

V

video
setup 2-8
subsystem 2-4

W

warning
arbitration 3-7
central arbiter 3-8
NMI mask 3-28
POS 2-3
RT/CMOS 3-19
watchdog status 3-30
weight, system unit 1-14, 1-15
width, system unit 1-14, 1-15

Numerics

386SLC microprocessor 1-13, 3-3
80386SX microprocessor 1-13, 3-3
80387SX math coprocessor 3-6

Personal System/2 Model 60 Technical Reference

First Edition (May, 1988)

The following paragraph does not apply to the United Kingdom or any country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.

It is possible that this publication may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

THE PUBLICATION OF THE INFORMATION CONTAINED HEREIN IS NOT INTENDED TO AND DOES NOT CONVEY ANY RIGHTS OR LICENSES, EXPRESS OR IMPLIED, UNDER ANY IBM PATENTS, COPYRIGHTS, TRADEMARKS, MASK WORKS OR ANY OTHER INTELLECTUAL PROPERTY RIGHTS.

Requests for copies of this publication and for technical information about IBM products should be made to your IBM Authorized Dealer or your IBM Marketing Representative.

Personal System/2 is a registered trademark of the International Business Machines Corporation.

© Copyright International Business Machines Corporation 1988. All rights reserved. No part of this work may be reproduced or distributed in any form or by any means without prior permission in writing from the IBM Corporation.

Preface

The Technical Reference library is intended for those who develop hardware and software products for IBM Personal Computers and IBM Personal System/2. Users should understand computer architecture and programming concepts.

This technical reference provides hardware and software interface information for the IBM Personal System/2 Model 60 and should be used with the following publications:

IBM Personal System/2 Hardware Interface Technical Reference

IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference

This manual consists of the following sections:

Section 1, "System Overview," describes the system, features, and specifications.

Section 2, "Programmable Option Select," describes registers used for configuration.

Section 3, "System Board," describes the system specific hardware implementations..

Warning: The term "Reserved" describes certain signals, bits, and registers that should not be changed. Use of reserved areas can cause compatibility problems, loss of data, or permanent damage to the hardware. When the contents of a register are changed, the state of the reserved bits must be preserved. When possible, read the register first and change only the bits that must be changed.

For information about components or devices not described in this manual, refer to the *Hardware Interface Technical Reference.* Information about diskette drives, fixed disk drives, adapters, and external options are in separate option technical references.

Notes:

Contents

Section 1. System Overview 1-1
Description 1-3
System Board Features 1-3
System Board Block Diagram 1-5
System Board I/O Address Map 1-6
Specifications 1-7

Section 2. Programmable Option Select 2-1
Description 2-3
POS Address Map 2-5
Card Selected Feedback 2-6
System Board Setup 2-6
System Board Enable/Setup Register (Hex 0094) 2-7
System Board POS Register 2 (Hex 0102) 2-8
System Board POS Register 3 (Hex 0103) 2-9
Adapter Enable/Setup Register (Hex 0096) 2-10

Section 3. System Board 3-1
Description 3-3
Micro Channel 3-3
Adapter Identification 3-3
Central Arbiter 3-3
Memory 3-7
Read-Only Memory Subsystem 3-7
Random Access Memory Subsystem 3-7
Real-Time Clock/Complementary Metal-Oxide Semiconductor RAM 3-10
Miscellaneous System Functions 3-21
Nonmaskable Interrupt 3-21
System Control Port B (Hex 0061) 3-21
System Control Port A (Hex 0092) 3-23
Power-On Password 3-25
Hardware Compatibility 3-26

Index X-1

Notes:

Figures

1-1. System Board Devices and Features 1-4
1-2. System Board 1-5
1-3. System Board I/O Address Map 1-6
1-4. Performance Specifications — System Board 1-7
1-5. Physical Specifications 1-8
2-1. POS I/O Address Map 2-5
2-2. Card Selected Feedback Register (Hex 0091) 2-6
2-3. System Board Enable/Setup Register (Hex 0094) 2-7
2-4. System Board POS Register 2 (Hex 0102) 2-8
2-5. Parallel Port Select Bits 2-8
2-6. System Board POS Register (Hex 0103) 2-9
2-7. Adapter Enable/Setup Register (Hex 0096) 2-10
3-1. Arbitration Bus Priority Assignments 3-4
3-2. Arbitration Register, Write to Hex 0090 3-5
3-3. Arbitration Register, Read Hex 0090 3-5
3-4. System Memory Map 3-8
3-5. System Board Memory Connector 3-9
3-6. RT/CMOS RAM Address Map 3-10
3-7. RT/CMOS Address Register and NMI Mask (Hex 0070) 3-11
3-8. RT/CMOS Data Register (Hex 0071) 3-11
3-9. Real-Time Clock Bytes 3-13
3-10. Status Register A 3-13
3-11. Status Register B 3-14
3-12. Status Register C 3-15
3-13. Status Register D 3-16
3-14. Diagnostic Status Byte 3-16
3-15. Diskette Drive Type Byte 3-17
3-16. Diskette Drive Type Byte (Bits 7 - 4) 3-18
3-17. Diskette Drive Type Byte (Bits 3 - 0) 3-18
3-18. Equipment Byte 3-19
3-19. Equipment Byte (Bits 7, 6) 3-19
3-20. Equipment Byte (Bits 5, 4) 3-19
3-21. System Control Port B (Write) 3-22
3-22. System Control Port B (Read) 3-22
3-23. System Control Port A (Hex 0092) 3-23

Notes:

Section 1. System Overview

Description 1-3
System Board Features 1-3
System Board Block Diagram 1-5
System Board I/O Address Map 1-6
Specifications 1-7

Notes:

Description

The IBM Personal System/2 Model 60 is a self-contained, floor-standing computer system with a keyboard. It can support two diskette drives and two internal fixed disk drives.

Programs can identify the system by reading the model and submodel bytes. The model and submodel byte for this system is hex FC and 05, respectively. Interrupt hex 15, function code (AH)= hex C0, returns the model and submodel bytes and BIOS revision code.

Refer to the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference* for a listing of other systems, and check the supplements section for updates to that listing.

System Board Features

The following figure lists the system board devices and features. The *Hardware Interface Technical Reference* describes all devices common to PS/2 products by type number.

Device	Type	Features
Microprocessor	---	80286 24-bit address and 16-bit data interface
System Timers	1	Channel 0 – System timer Channel 2 – Tone generation for speaker Channel 3 – Watchdog timer
ROM Subsystem	---	128KB (KB = 1024 bytes)
RAM Subsystem	---	1MB (MB = 1,048,576 bytes) Expandable on the channel
CMOS RAM Subsystem	---	64-byte CMOS RAM with real-time clock/calendar 2KB CMOS RAM extension Battery backup
Video Subsystem	1	Auxiliary connector on the channel Analog output 256KB video memory
Audio Subsystem	1	Driven by: - System-timer channel 2 - The 'audio sum node' signal.
DMA Controller	1	Eight independent DMA channels Single or burst transfers and read verification
Interrupt Controller	1	16 levels of system interrupts Interrupts are level-sensitive
Keyboard/Auxiliary Device Controller	1	Keyboard connector Auxiliary device connector Password security
Diskette Drive Controller	1	Supports: - 720KB formatted diskette density - 1.44MB formatted diskette density.
Serial Controller	1, 2	RS-232C interface Programmable as serial port 1 or 2 FIFO mode* and character mode
Parallel Controller	1	Programmable as parallel port 1, 2, or 3 Supports bidirectional input and output
Micro Channel	---	Eight channel connectors: - 8- or 16-bit device support - One with an auxiliary video extension - One occupied by the fixed disk drive adapter.
Math Coprocessor Socket	---	Supports 80287 math coprocessors Same clock speed as the system microprocessor
Power Supply	2	110 and 220 Vac support

* The FIFO mode is supported by the Type 2 serial controller.

Figure 1-1. System Board Devices and Features

System Board Block Diagram

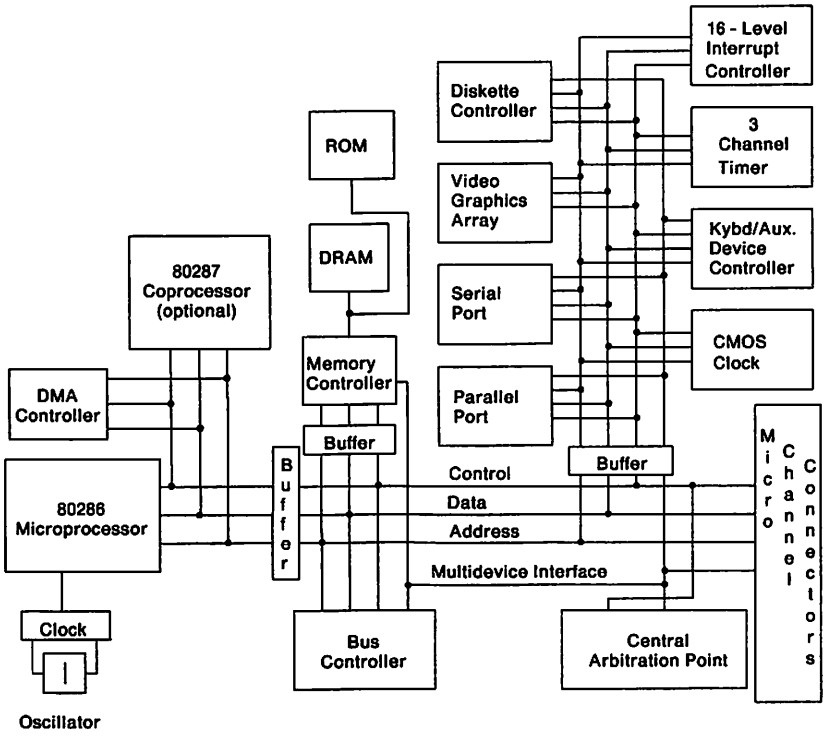


Figure 1-2. System Board

System Board I/O Address Map

Hex Addresses	Device
0000 - 001F	DMA Controller (0 – 3)
0020, 0021	Interrupt Controller (Master)
0040, 0042 - 0044, 0047	System Timers
0060	Keyboard, Auxiliary Device
0061	System Control Port B
0064	Keyboard, Auxiliary Device
0070, 0071	RT/CMOS and NMI Mask
0081 - 0083, 0087	DMA Page Registers (0 – 3)
0089, 008A, 008B, 008F	DMA Page Registers (4 – 7)
0090	Central Arbitration Control Point
0091	Card Selected Feedback Register
0092	System Control Port A
0094	System Board Enable/Setup Register
0096	Adapter Enable/Setup Register
00A0 - 00A1	Interrupt Controller (Slave)
00C0 - 00DF	DMA Controller (4 – 7)
00F0 - 00FF	Math Coprocessor
0100 - 0107	Programmable Option Select
01F0 - 01F8	Fixed Disk Drive Controller
0278 - 027B	Parallel Port 3
02F8 - 02FF	Serial Port 2 (RS-232C)
0378 - 037B	Parallel Port 2
03B4, 03B5, 03BA	Video Subsystem
03BC - 03BF	Parallel Port 1
03C0 - 03C5	Video Subsystem
03C6 - 03C9	Video DAC
03CA, 03CC, 03CE, 03CF	Video Subsystem
03D4, 03D5, 03DA	Video Subsystem
03F0 - 03F7	Diskette Drive Controller
03F8 - 03FF	Serial Port 1 (RS-232C)

Figure 1-3. System Board I/O Address Map

Specifications

Device	Number of Waits	Cycle Time (ns)
Microprocessor (10 MHz – 100 ns Clock):		
Access to System Board RAM	1	300
Access to System Board ROM	1	300
Access to Channel:		
Default Transfer Cycle:		
I/O Access	1	300
Memory Access	0	200
Synchronous Extended Transfer Cycle	1	300
Refresh Rate (Typically performed every 15.1 μs)		500 (min)
Bus Master Access to System Board RAM		300 (min)
DMA Controller (10 MHz – 100 ns Clock):		
Single Transfer: 300 + I/O Access + Memory Access		
Burst Transfers: 300 + (I/O Access + Memory Access)N *		
System Board Memory Access		300
Default Transfer Cycle:		
I/O Access		300
Memory Access		200
Synchronous Extended Transfer Cycle		300

* N is the number of transfers in the burst.

Figure 1-4. Performance Specifications – System Board

Size:	
Width	165 mm (6.5 in)
Width (Feet extended)	318 mm (12.5 in)
Depth	483 mm (19.0 in)
Height	597 mm (23.5 in)
Weight	20 kg (44 lb) with one fixed disk drive
Cables:	
Power Cable	1.8 m (6 ft)
Keyboard Cable	3.05 m (10 ft)
Air Temperature:	
System On	15.6 to 32.2°C (60 to 90°F)
System Off	10.0 to 43.0°C (50 to 110°F)
Humidity:	
System On	8% to 80%
System Off	20% to 80%
Maximum Altitude	2133.6 m (7000 ft)
Heat Output	363.4 W/hr
Acoustical Readings from 1 m (3.28 ft):	
Operating	46 dB average
Idle	40 dB average
Electrical:	
Power Supply Input:	
Low Range	90 (min) - 137 (max)Vac
High Range	180 (min) - 265 (max)Vac
Maximum Current Draw:	
Low Range	5.3 A
High Range	2.7 A
Frequency	50 - 60Hz ± 3Hz
Maximum Allowable Current:	
Keyboard Port	275 mA
Auxiliary Device Port	300 mA
Electromagnetic Compatibility	FCC Class B

Figure 1-5. Physical Specifications

Section 2. Programmable Option Select

Description 2-3
POS Address Map 2-5
Card Selected Feedback 2-6
System Board Setup 2-6
System Board Enable/Setup Register (Hex 0094) 2-7
System Board POS Register 2 (Hex 0102) 2-8
System Board POS Register 3 (Hex 0103) 2-9
Adapter Enable/Setup Register (Hex 0096) 2-10

Notes:

Description

Programmable Option Select (POS) eliminates the need for switches by replacing their function with programmable registers. This section describes the POS information used on the Model 60 system board. For additional POS information, refer to the *Hardware Interface Technical Reference*.

Warning:

- IBM recommends that programmable options be set only through the System Configuration utilities. Directly setting the POS registers or CMOS RAM POS parameters can cause multiple assignment of the same system resource, improper operation of the feature, loss of data, or possible damage to the hardware.
- Application programs should not use the adapter identification (ID) unless absolutely necessary. Compatibility problems can result.
- If an adapter and the system board are in setup mode at the same time, bus contention will occur, no useful programming can take place, and damage to the hardware can occur.
- After setup operations are complete, the Adapter Enable/Setup register (hex 0096) should be set to hex 00, and the System Board Enable/Setup register (hex 0094) should be set to hex FF.
- The channel reset bit (bit 7) in the Adapter Enable/Setup register must be 0 to program the adapters.
- The system board does not support 16-bit I/O operations to 8-bit POS registers. Using 16-bit I/O instructions on 8-bit POS registers will cause erroneous data to be written to or read from the registers. Only 8-bit transfers are supported for setup operations.

Setup functions respond to I/O addresses hex 0100 through 0107 only when their unique setup signal is active. The following precautions must be taken before setting individual bits in the POS registers.

System Board Video Subsystem Setup:

- Bit 5 in the System Board Enable/Setup register (hex 0094) must be set to 0 to place the system board video into the setup mode.
- Bit 3 in the Adapter Enable/Setup register (hex 0096) must be set to 0 to avoid driving a 'setup' signal to an adapter.
- Bit 7 in the System Board Enable/Setup register (hex 0094) must be set to 1 to avoid driving a 'setup' signal to other system board functions.

Adapter Setup:

- Bit 3 in the Adapter Enable/Setup register (hex 0096) must be set to 1 to allow adapter setup.
- Bit 5 in the System Board Enable/Setup register (hex 0094) must be set to 1 to avoid driving a 'setup' signal to the Video Subsystem.
- Bit 7 in the System Board Enable/Setup register (hex 0094) must be set to 1 to avoid driving a 'setup' signal to a system board function.

Other System Board Setup:

- Bit 7 in the System Board Enable/Setup register (hex 0094) must be set to 0 to allow setup of other system board functions.
- Bit 3 in the Adapter Enable/Setup register (hex 0096) must be set to 0 to avoid driving a 'setup' signal to an adapter.
- Bit 5 in the System Board Enable/Setup register (hex 0094) must be set to 1 to avoid driving a 'setup' signal to the Video Subsystem.

POS Address Map

The following figure shows the organization of the I/O address space used by POS. The POS registers are 8-bit registers. Bit 0 of POS Register 2 and bits 6 and 7 of POS Register 5 are defined. All other bits in POS Registers 2 through 5 are free-form.

Address (Hex)	Function
0094	System Board Enable/Setup Register
0096	Adapter Enable/Setup Register
0100	POS Register 0—Adapter Identification Byte (Low Byte)
0101	POS Register 1—Adapter Identification Byte (High Byte)
0102	POS Register 2—Option Select Data Byte 1 Bit 0 is Card Enable.
0103	POS Register 3—Option Select Data Byte 2
0104	POS Register 4—Option Select Data Byte 3
0105	POS Register 5—Option Select Data Byte 4 Bit 7 is the channel check active indicator. Bit 6 is the channel check status-available indicator.
0106	POS Register 6—Subaddress Extension (Low Byte)
0107	POS Register 7—Subaddress Extension (High Byte)

Figure 2-1. POS I/O Address Map

Card Selected Feedback

When the adapter is addressed, it responds by setting the '-card selected feedback' signal (-CD SFDBK) to active. -CD SFDBK is derived from the address decode and driven by a totem pole driver. It is latched by the system board and can be read through the Card Selected Feedback register at address hex 0091. Diagnostic and automatic configuration programs use this signal to verify the operation of an adapter at a given address or DMA port. This signal must not be active during a setup cycle.

The Card Selected Feedback register is a read-only register at address hex 0091. It allows programs to monitor -CD SFDBK and thereby determine if the video subsystem, system board I/O, or an adapter is addressed and functioning.

Bit	Function
7 - 1	Reserved
0	-Card Selected Feedback

Figure 2-2. Card Selected Feedback Register (Hex 0091)

Bits 7 - 1 Reserved.

Bit 0 This bit is set to 1 whenever -CD SFDBK was active on a previous cycle or whenever the system board I/O functions (diskette drive, serial, or parallel interfaces) are accessed by an I/O cycle. Reading this register resets the bit to 0.

System Board Setup

The integrated I/O functions on the system board use POS information during setup. The diskette drive controller, serial port, and parallel port are treated as a single device. The video subsystem is also an integrated part of the system board; however, POS treats it as a separate device. The System Board Enable/Setup register is used to place the system board or video subsystem into the setup mode.

System Board Enable/Setup Register (Hex 0094)

This is a read/write register; all bits in this register default to 1 (enabled).

Bit	Function
7	Enable/-Setup System Board Functions
6	Reserved
5	Enable/-Setup Video Subsystem
4 - 0	Reserved

Figure 2-3. System Board Enable/Setup Register (Hex 0094)

Bit 7 When this bit is set to 0, various system board I/O functions are placed in the setup mode. The diskette drive controller, serial port, and parallel port are controlled through System Board POS Register 2 (hex 0102). The memory is controlled through System Board POS Register 3 (hex 0103).

When this bit is set to 1, the system board functions are enabled.

Bit 6 Reserved.

Bit 5 When this bit is set to 0, the video subsystem is placed in the setup mode and controlled through POS Register 2 (hex 0102). When set to 1, and bit 0 in hex 0102 is set to 1, video is enabled.

Bit 0 of POS Register 2 is the video enable bit. When this bit is set to 0, the video subsystem does not respond to commands, addresses, or data. If video is being generated when the video enable bit is set to 0, the output is still generated. For information on BIOS calls to enable or disable the video, see the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference.*

Note: When the video is disabled, access to the video DAC registers is disabled.

Bits 4 - 0 Reserved.

System Board POS Register 2 (Hex 0102)

When the system board is in the setup mode, the diskette drive controller, serial port, and parallel port are controlled by this read/write register. Reading this register returns the current state of these system board functions.

Bit	Function
7	Disable Parallel Port Extended Mode
6, 5	Parallel Port Select
4	Enable Parallel Port
3	Serial Port Select
2	Enable Serial Port
1	Enable Diskette Drive Interface
0	Enable System Board

Figure 2-4. System Board POS Register 2 (Hex 0102)

Bit 7 When set to 0, this bit allows the parallel port to be configured as an 8-bit parallel bidirectional interface. When set to 1, this bit disables the bidirectional mode. This bit is set to 0 at power-on, and POST sets it to 1.

Bits 6, 5 These bits select the configuration of the system board parallel port.

Bits 6 5	Assignment	Hex Address	Interrupt Level
0 0	Parallel 1	03BC - 03BF	7
0 1	Parallel 2	0378 - 037B	7
1 0	Parallel 3	0278 - 027B	7
1 1	Reserved	----------	-

Figure 2-5. Parallel Port Select Bits

Bit 4 When this bit and bit 0 are set to 1, the system board parallel port is enabled.

Bit 3 When set to 1, this bit sets the system board serial port as Serial 1 (addresses hex 03F8 through 03FF), which uses interrupt level 4. When set to 0, this bit sets the serial port as Serial 2 (addresses hex 02F8 through 02FF), which uses interrupt level 3.

Bit 2 When this bit and bit 0 are set to 1, the system board serial port is enabled.

Bit 1 When this bit and bit 0 are set to 1, the diskette drive interface is enabled.

Bit 0 When set to 1, this bit allows bits 4, 2, and 1 to enable and disable their respective devices. When set to 0, this bit disables the diskette drive interface, system board serial port, and system board parallel port, regardless of the state of bits 4, 2, and 1.

System Board POS Register 3 (Hex 0103)

With the system board in setup, this register controls the system board memory.

Bit	Function
7 - 1	Reserved
0	Enable System Board RAM

Figure 2-6. System Board POS Register (Hex 0103)

Bit 7-1 Reserved.

Bit 0 When set to 1, this bit enables the system board memory. When set to 0, this bit disables the memory. All system board memory is enabled or disabled by this operation.

Adapter Enable/Setup Register (Hex 0096)

The Adapter Enable/Setup register selects the connector to be configured.

Bit	Symbol
7	Channel Reset
6 - 4	Reserved
3	Card Setup Enable
2 - 0	Channel Select 2 - 0

Figure 2-7. Adapter Enable/Setup Register (Hex 0096)

Bit 7 When set to 1, this bit activates the 'channel reset' signal to all connectors.

Bits 6 - 4 These bits are reserved.

Bit 3 When set to 1, this bit enables the '-card setup' signal selected by bits 2 through 0.

Bits 2 - 0 These bits are the address bits for the '-card setup' signal Connectors 1 through 8 are addressed as 0 through 7, respectively. When bit 3 is set to 1, these bits select the connector that is put into setup.

Each channel connector has a unique '-card setup' signal (-CD SETUP) associated with it. This signal is used to put the adapters in the setup mode, which allows access to the POS registers. The individual connectors are selected through the Adapter Enable/Setup register. Setup information is then read from or written to the selected adapter through I/O addresses hex 0100 through 0107.

Notes:

1. -CD SETUP only goes active when an operation is performed in the I/O address range hex 0100 through 0107.
2. The status of port hex 0096 can be read by software. However, when the port is read, bits 6, 5, and 4 are set to 1.

Section 3. System Board

Description 3-3
Micro Channel 3-3
  Adapter Identification 3-3
  Central Arbiter 3-3
    Arbitration Bus Priority Assignments 3-4
    Central Arbiter Programming 3-5
Memory 3-7
  Read-Only Memory Subsystem 3-7
  Random Access Memory Subsystem 3-7
    System Memory Maps 3-8
    System Board Memory Connector 3-9
  Real-Time Clock/Complementary Metal-Oxide Semiconductor RAM 3-10
    RT/CMOS Address Register and NMI Mask (Hex 0070) . . 3-11
    RT/CMOS Data Register (Hex 0071) 3-11
    RT/CMOS RAM I/O Operations 3-12
    Real-Time Clock Bytes (Hex 000-00D) 3-13
    CMOS RAM Configuration 3-16
Miscellaneous System Functions 3-21
  Nonmaskable Interrupt 3-21
  System Control Port B (Hex 0061) 3-21
  System Control Port A (Hex 0092) 3-23
  Power-On Password 3-25
Hardware Compatibility 3-26

Notes:

Description

This section provides information about the Model 60 implementation of the Micro Channel, memory subsystem, and miscellaneous system ports. For additional information about these and other areas not discussed, refer to the *Hardware Interface Technical Reference*.

Micro Channel

POST routines and registers used to implement programmable features of the Micro Channel architecture may vary from system to system. This section provides information about the Model 60 implementation of the Micro Channel architecture.

Adapter Identification

When the system is powered on, an adapter can issue an ID of hex 0000 for up to 1 second after channel reset to indicate it is not ready. Any adapter that continues to issue an adapter ID of hex 0000 (not ready) for more than 1 second is considered defective. On the Model 60, if the channel reset was caused by a keyboard sequence or a program routine, this time is reduced to 740 milliseconds.

Central Arbiter

The central arbitration control point gives intelligent subsystems on the channel the ability to share and control the system. It allows burst data transfers and prioritization of control between devices. This arbiter supports up to 16 arbitrating devices.

Arbitration Bus Priority Assignments

The following figure shows the assignment of arbitration levels. The functions with the lowest arbitration level have the highest priority.

ARB Level	Primary Assignment
-2	Memory Refresh
-1	NMI
0	DMA Channel 0 (Programmable to any arbitration level)
1	DMA Channel 1
2	DMA Channel 2
3	DMA Channel 3
4	DMA Channel 4 (Programmable to any arbitration level)
5	DMA Channel 5
6	DMA Channel 6
7	DMA Channel 7
8-E	Available
F	System Microprocessor

Figure 3-1. Arbitration Bus Priority Assignments

Note: Devices designed for arbitration level 0 or 1 should have limited bandwidth or short bursts so diskette overruns can be prevented or recovered by retry operations. The diskette drive controller, on arbitration level 2, can be held inactive by devices on levels 0 and 1, by a refresh operation, and by the previous controlling master. The diskette drive controller should not be held inactive for more than 12 microseconds to prevent overrun.

NMI service is executed at a priority level higher than 0, called -1. Memory refresh is prioritized at -2, two levels higher than 0. Levels -1 and -2 are reached on the system board only, while the 'arbitrate/-grant' signal (ARB/-GNT)) is in the arbitrate state.

When the central arbitration control point receives a level -1 request (NMI, a system-board internal signal), it activates -PREEMPT, waits for the end of transfer, and then places ARB/-GNT in the arbitrate state, which denies channel activity to arbitrating devices. The central arbitration control point gives the grant to the level -1 request, and holds ARB/-GNT in the arbitrate state until the operation is complete and the NMI is reset.

Central Arbiter Programming

The central arbitration control point provides access to programmable options through the Arbitration register, which is accessed at I/O address hex 0090. The bits are defined differently for read and write operations, as shown in the following figures.

Bit	Definition
7	Enable System Microprocessor Cycle
6	Arbitration Mask
5	Enable Extended Arbitration
4 - 0	Reserved

Figure 3-2. Arbitration Register, Write to Hex 0090

Bit	Definition
7	Enable System Microprocessor Cycle
6	Arbitration Masked by NMI
5	Bus Time-out
4	Reserved
3 - 0	Value of Arbitration Bus During Previous Grant State

Figure 3-3. Arbitration Register, Read Hex 0090

Bit 7 Setting this bit to 1 enables system microprocessor cycles during arbitration cycles. This bit can be set to 0 if an arbitrating device requires total control of the channel bandwidth. This bit is set to 0 by a system reset.

Reading this bit as a 1 indicates system microprocessor cycles are enabled during arbitration.

Bit 6 Setting this bit to 1 causes the central arbitration control point to enter the arbitration state. The system microprocessor controls the channel until this bit is reset to 0. This bit is set to 0 by a system reset.

Reading this bit as a 1 indicates that an NMI has occurred and has masked arbitration.

Warning: This bit should be set to 1 only by diagnostic routines and system error-recovery routines.

Bit 5 Setting this bit to 1 enables extended arbitration. The minimum arbitration cycle is 300 nanoseconds; this bit extends that minimum cycle to 600 nanoseconds. This bit is set to 0 during a system reset.

Reading this bit as a 1 indicates that a bus time-out has occurred, and resets bit 6 in this register to 0.

Bit 4 This bit is reserved and should be 0.

Bits 3 - 0 These bits are undefined for a write operation and should be set to 0.

Reading these bits returns the arbitration level of the arbiter controlling the channel during the most recent grant state. This information allows the system microprocessor to determine the arbitration level of the device that caused a bus time-out.

Memory

The Model 60 system uses the following types of memory:

- Read-only memory (ROM)
- Random access memory (RAM)
- Real-time Clock and CMOS RAM.

Read-Only Memory Subsystem

The ROM subsystem system board consists of four 32Kb by 8-bit modules in a 64Kb by 16-bit arrangement (Kb equals 1024 bits).

ROM is not parity-checked and is assigned addresses at the top of the first and last 1MB of address space (hex 0E0000 and FE0000).

Random Access Memory Subsystem

The RAM subsystem on the system board starts at address hex 000000 of the 16MB address space. It consists of 1MB of RAM modules.

The system board uses four memory module packages that operate with one wait state. Memory-refresh typically occurs once every 15 microseconds.

The timing of the memory signals on the system board is similar to the Hitachi[1] HB61009BR-15, 150-nanosecond DRAM or equivalent.

[1] Hitachi is a trademark of the Hitachi Corporation.

System Memory Maps

The following figure shows the mapping of the memory locations on the Model 60. A 256-byte portion of this RAM is reserved as a BIOS data area. A 1KB portion of this RAM is reserved as an extended BIOS data area. See the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference* for details.

Hex Range	Function
000000 to 09FFFF	640KB System Board RAM *
0A0000 to 0BFFFF	128KB Video RAM
0C0000 to 0DFFFF	128KB I/O Expansion ROM
0E0000 to 0FFFFF	128KB System Board ROM
100000 to 15FFFF	384KB System Board RAM
160000 to FDFFFF	Channel Expansion Memory Addresses
FE0000 to FFFFFF	128KB System Board ROM

* With the system board in the setup mode, the system board memory can be disabled through System Board POS Register 3 (see page 2-9).

Figure 3-4. System Memory Map

System Board Memory Connector

The system board has four 30-pin connectors and comes with four 256KB memory module packages installed. The following figure shows the pin assignments and signal definitions for these connectors.

Pin	I/O	Signal	Pin	I/O	Signal
1	N/A	+ 5 Vdc	2	I	-Column Address Strobe
3	I/O	Data Bit 1	4	I	Address Bit 1
5	I	Address Bit 2	6	I/O	Data Bit 2
7	I	Address Bit 3	8	I	Address Bit 4
9	N/A	Ground	10	I/O	Data Bit 3
11	I	Address Bit 5	12	I	Address Bit 6
13	I/O	Data Bit 4	14	I	Address Bit 7
15	I	Address Bit 8	16	I/O	Data Bit 5
17	I	Address Bit 9	18	N/A	No Connection
19	I	Reserved	20	I/O	Data Bit 6
21	I	-Write Strobe	22	N/A	Ground
23	I/O	Data Bit 7	24	O	Presence Detect
25	I/O	Data Bit 8	26	O	Presence Detect
27	I	Reserved	28	N/A	No Connection
29	I/O	Data Bit 9 (Parity)	30	N/A	+5 Vdc

Figure 3-5. System Board Memory Connector

Real-Time Clock/Complementary Metal-Oxide Semiconductor RAM

The real-time clock/complementary metal-oxide semiconductor RAM (RT/CMOS) chip contains the real-time clock and 64 bytes of CMOS RAM. The internal clock circuitry uses 14 bytes of this memory, and the rest is allocated to configuration and system status information.

In addition to the 64 bytes of CMOS RAM, a 2KB CMOS RAM extension is provided for configuration and other system information.

A 6-Vdc lithium battery maintains voltage to the RT/CMOS RAM and 2KB CMOS RAM extension when the power supply is not in operation.

The system cover can be locked to prevent battery removal and loss of password and configuration information.

The following figure shows the RT/CMOS RAM bytes and their addresses.

Address (Hex)	RT/CMOS RAM Bytes
000 - 00D	Real-Time Clock Bytes
00E	Diagnostic Status Byte
00F	Shutdown Status Byte
010	Diskette Drive Type Byte
011	First Fixed Disk Drive Type Byte
012	Second Fixed Disk Drive Type Byte
013	Reserved
014	Equipment Byte
015 - 016	Low and High Base Memory Bytes
017 - 018	Low and High Expansion Memory Bytes
019 - 031	Reserved
032 - 033	Configuration CRC Bytes
034 - 036	Reserved
037	Date Century Byte
038 - 03F	Reserved

Figure 3-6. RT/CMOS RAM Address Map

RT/CMOS Address Register and NMI Mask (Hex 0070)

This register is used in conjunction with the port at hex 0071 to read and write the RT/CMOS RAM bytes.

Bit	Function
7	NMI Mask
6	Reserved
5 - 0	RT/CMOS RAM Address

Figure 3-7. RT/CMOS Address Register and NMI Mask (Hex 0070)

Warning: The operation following a write to hex 0070 should access port hex 0071; otherwise intermittent malfunctions and unreliable operation of the RT/CMOS RAM can occur.

Bit 7 When this bit is set to 1, the NMI is masked off (the NMI is disabled). This bit is set to 1 by a power-on reset and is a write-only bit.

Bit 6 Reserved.

Bits 5 - 0 These bits are used to select RT/CMOS RAM addresses.

RT/CMOS Data Register (Hex 0071)

This port is used in conjunction with the address register at hex 0070 to read and write the RT/CMOS RAM bytes.

Bit	Function
7 - 0	RT/CMOS Data

Figure 3-8. RT/CMOS Data Register (Hex 0071)

RT/CMOS RAM I/O Operations

During I/O operations to the RT/CMOS RAM addresses, interrupts should be masked to prevent other interrupt service routines from changing the CMOS address register before data is read or written. After I/O operations, the RT/CMOS and NMI Mask register (hex 0070) should be left pointing to Status Register D (hex 00D).

Warning: The operation following a write to hex 0070 should access hex 0071; otherwise intermittent malfunctions and unreliable operation of the RT/CMOS RAM can occur.

The following steps are required to perform I/O operations to the RT/CMOS RAM addresses:

1. Write the RT/CMOS RAM address to the RT/CMOS and NMI Mask register (hex 0070).
2. Write the data to address hex 0071.

Reading RT/CMOS RAM requires the following steps:

1. Write the RT/CMOS RAM address to the RT/CMOS and NMI Mask register (hex 0070).
2. Read the data from address hex 0071.

Real-Time Clock Bytes (Hex 000-00D)

Bit definitions and addresses for the real-time clock bytes are shown in the following figure.

Address (Hex)	Function	Byte Number
000	Seconds	0
001	Second Alarm	1
002	Minutes	2
003	Minute Alarm	3
004	Hours	4
005	Hour Alarm	5
006	Day of Week	6
007	Date of Month	7
008	Month	8
009	Year	9
00A	Status Register A	10
00B	Status Register B	11
00C	Status Register C	12
00D	Status Register D	13

Figure 3-9. Real-Time Clock Bytes

Note: The Setup program initializes status registers A, B, C, and D when the time and date are set. Interrupt hex 1A is the BIOS interface to read and set the time and date and it initializes the register the same way as the Setup program.

Status Register A (Hex 00A)

Bit	Function
7	Update in Progress
6 - 4	22-Stage Divider
3 - 0	Rate Selection Bits

Figure 3-10. Status Register A

Bit 7 When set to 1, this bit indicates the time-update cycle is in progress. When set to 0, it indicates the current date and time can be read.

Bits 6 - 4 These three divider-selection bits identify which time-base frequency is being used. The system initializes these bits to binary 010, which selects a 32.768 kHz time-base. This is the only value supported by the system for proper time-keeping.

Bits 3 - 0 These bits allow the selection of a divider output frequency. The system initializes the rate selection bits to a binary 0110, which selects a 1.024 kHz square-wave output frequency and a 976.562-microsecond periodic interrupt rate.

Status Register B (Hex 00B)

Bit	Function
7	Set
6	Periodic Interrupt Enable
5	Alarm Interrupt Enable
4	Update-Ended Interrupt Enabled
3	Square Wave Enabled
2	Date Mode
1	24-Hour Mode
0	Daylight Savings Enabled

Figure 3-11. Status Register B

Bit 7 When set to 0, this bit updates the cycle, normally by advancing the counts at a rate of one per second. When set to 1, this bit immediately ends any update cycle in progress, and the program can initialize the 14 time bytes without any further updates occurring until this bit is set to 0.

Bit 6 This bit is a read/write bit that allows an interrupt to occur at a rate specified by the rate and divider bits in Status Register A. When set to 1, this bit enables the interrupt. The system initializes this bit to 0.

Bit 5 When set to 1, this bit enables the alarm interrupt. The system initializes this bit to 0.

Bit 4 When set to 1, this bit enables the update-ended interrupt. The system initializes this bit to 0.

Bit 3 When set to 1, this bit enables the square-wave frequency as set by the rate-selection bits in Status Register A. The system initializes this bit to 0.

Bit 2 This bit indicates if the time-and-date calendar updates use binary or binary-coded-decimal (BCD) formats. When set to 1, this bit indicates a binary format. The system initializes this bit to 0.

Bit 1 This bit establishes if the hours byte is in the 24-hour or 12-hour mode. When set to 1, this bit indicates the 24-hour mode. The system initializes this bit to 1.

Bit 0 When set to 1, this bit enables the daylight savings time mode. When set to 0, it disables the mode, and the clock reverts to standard time. The system initializes this bit to 0.

Status Register C (Hex 00C)

Bit	Function
7	Interrupt Request Flag
6	Periodic Interrupt Flag
5	Alarm Interrupt Flag
4	Update-Ended Interrupt Flag
3 - 0	Reserved

Figure 3-12. Status Register C

Note: Interrupts are enabled by bits 6, 5, and 4 in Status Register B.

Bit 7 This bit is used in conjunction with bits 6, 5, and 4. When set to 1, this bit indicates that an interrupt has occurred; bits 6, 5, and 4 indicate the type of interrupt.

Bit 6 When set to 1, this bit indicates that a periodic interrupt occurred.

Bit 5 When set to 1, this bit indicates that an alarm interrupt occurred.

Bit 4 When set to 1, this bit indicates that an update-ended interrupt occurred.

Bits 3 - 0 Reserved.

Status Register D (Hex 00D)

Bit	Function
7	Valid RAM
6 - 0	Reserved

Figure 3-13. Status Register D

Bit 7 This read-only bit monitors the power-sense pin. A low state of this pin indicates a loss of power to the real-time clock (dead battery). When set to 1, this bit indicates that the real-time clock has power. When set to 0, it indicates that the real-time clock has lost power.

Bits 6 - 0 Reserved.

CMOS RAM Configuration

The following shows the bit definitions for the CMOS RAM configuration bytes.

Diagnostic Status Byte (Hex 00E)

Bit	Function
7	Real-Time Clock Chip Power
6	Configuration Record and Checksum Status
5	Incorrect Configuration
4	Memory Size Miscompare
3	Fixed Disk Controller/Drive C Initialization Status
2	Time Status Indicator
1	Adapter Configuration Miscompare
0	Adapter ID Time-Out

Figure 3-14. Diagnostic Status Byte

Bit 7 When set to 1, this bit indicates the real-time clock chip lost power.

Bit 6 When this bit is set to 1, the checksum is incorrect.

Bit 5 This is a check, at power-on time, of the Equipment byte. When set to 1, the configuration information is incorrect. Power-on checks require that at least one diskette drive be installed (bit 0 of the Equipment byte, hex 014, is set to 1).

Bit 4 When set to 1, this bit indicates the power-on check determined that the memory size is not the same as in the configuration record.

Bit 3 When set to 1, this bit indicates that the controller or drive C failed initialization, which prevents the system from attempting a power-on reset.

Bit 2 When set to 0, this bit indicates the time is valid. When set to 1, this bit indicates the time is invalid.

Bit 1 This bit indicates if the installed adapters match the configuration information. When this bit is set to 1, the adapters do not match the configuration information.

Bit 0 When set to 1, this bit indicates a time-out occurred while an adapter ID was being read.

Shutdown Status Byte (Hex 00F): This byte is defined by the power-on diagnostic programs.

Diskette Drive Type Byte (Hex 010): This byte indicates the type of diskette drive installed.

Bit	Function
7 - 4	First Diskette Drive Type
3 - 0	Second Diskette Drive Type

Figure 3-15. Diskette Drive Type Byte

Bits 7 - 4 These bits indicate the first diskette drive type, as shown in the following figure.

Bits 7 6 5 4	Function
0 0 0 0	No drive present
0 0 0 1	Double-sided diskette drive (48 tracks per inch, 360KB)
0 0 1 1	High-capacity diskette drive (720KB)
0 1 0 0	High-density diskette drive (1.44MB)

Note: All combinations that are not shown are reserved.

Figure 3-16. Diskette Drive Type Byte (Bits 7 - 4)

Bits 3 - 0 These bits indicate the second diskette drive type, as shown in the following figure.

Bits 3 2 1 0	Function
0 0 0 0	No drive present
0 0 0 1	Double-sided diskette drive (48 tracks per inch, 360KB)
0 0 1 1	High-capacity diskette drive (720KB)
0 1 0 0	High-density diskette drive (1.44MB)

Note: All combinations that are not shown are reserved.

Figure 3-17. Diskette Drive Type Byte (Bits 3 - 0)

First Fixed Disk Drive Type Byte (Hex 011): This byte defines the type of the first fixed disk drive (drive C). Hex 00 indicates that a fixed disk drive is *not* installed.

Second Fixed Disk Drive Type Byte (Hex 012): This byte defines the type of the second fixed disk drive (drive D). Hex 00 indicates that a fixed disk drive is *not* installed.

Note: For more information about fixed disk drive types, refer to the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference*.

Reserved Byte (Hex 013): This byte is reserved.

Equipment Byte (Hex 014): The equipment byte defines the basic equipment in the system for the power-on diagnostic tests.

Bit	Function
7, 6	Number of Diskette Drives
5, 4	Display Operating Mode
3, 2	Reserved
1	Math Coprocessor Presence
0	Diskette Drive Presence

Figure 3-18. Equipment Byte

Bits 7, 6 These bits indicate the number of diskette drives installed, as shown in the following figure.

Bits 7 6	Number of Diskette Drives
0 0	One Drive
0 1	Two Drives
1 0	Reserved
1 1	Reserved

Figure 3-19. Equipment Byte (Bits 7, 6)

Bits 5, 4 These bits indicate the operating mode of the display attached to the video port, as shown in the following figure.

Bits 5 4	Display Operating Mode
0 0	Reserved
0 1	40-Column Mode
1 0	80-Column Mode
1 1	Monochrome Mode

Figure 3-20. Equipment Byte (Bits 5, 4)

Bits 3, 2 Reserved.

Bit 1 When set to 1, this bit indicates that a math coprocessor is installed.

Bit 0 When set to 1, this bit indicates that a diskette drive is installed.

Low and High Base Memory Bytes (Hex 015 and 016): These bytes define the amount of memory below the 640KB address space.

The value from these bytes represents the number of 1KB blocks of base memory. For example, hex 0280 is equal to 640KB. The low byte is hex 15; the high byte is hex 16.

Low and High Expansion Memory Bytes (Hex 017 and 018): These bytes define the amount of memory above the 1MB address space.

The hexadecimal values in these bytes represent the number of 1KB blocks of expansion memory. For example, hex 0800 is equal to 2048KB. The low byte is hex 17; the high byte is hex 18.

Reserved Bytes (Hex 019 through 031): These bytes are reserved.

Configuration CRC Bytes (Hex 032 and 033): These bytes contain the cyclic-redundancy-check data for bytes hex 010 through hex 031 of the 64-byte CMOS RAM. The low byte is hex 33; the high byte is hex 32.

Reserved Bytes (Hex 034 through 036): These bytes are reserved.

Date Century Byte (Hex 037): Bits 7 through 0 of this byte contain the BCD value for the century. Refer to the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference* for information about reading and setting this byte.

Reserved Bytes (Hex 038 through 03F): These bytes are reserved.

Miscellaneous System Functions

Nonmaskable Interrupt

The nonmaskable interrupt (NMI) signals the system microprocessor that a parity error, a channel check, a system channel time-out, or a system Watchdog time-out has occurred. The NMI stops all arbitration on the bus until bit 6 of the Arbitration register (I/O address hex 0090) is set to 0. This can result in lost data or an overrun error on some I/O devices. The NMI masks all other interrupts and the IRET instruction restores the interrupt flag to the state it was in prior to the interrupt. A system reset causes a reset of the NMI.

Nonmaskable interrupt requests from system board parity and channel check are subject to mask control with the NMI mask bit in the RT/CMOS Address register. The Watchdog Timer and system channel time-out are not masked by this bit. (See "RT/CMOS RAM I/O Operations" on page 3-12). The power-on default of the NMI mask is 1 (NMI disabled). Prior to enabling the NMI after a power-on reset, the parity check and channel check state are initialized by the POST.

Warning: The operation following a write to hex 0070 should access port hex 0071; otherwise intermittent malfunctions and unreliable operation of the RT/CMOS RAM can occur.

System Control Port B (Hex 0061)

Bit definitions for the read and write functions of this port are shown in the following figures.

Bit	Function
7	Reset Timer 0 Output Latch (IRQ0)
6 - 4	Reserved
3	Enable Channel Check
2	Enable Parity Check
1	Speaker Data Enable
0	Timer 2 Gate to Speaker

Figure 3-21. System Control Port B (Write)

Bit	Function
7	Parity Check
6	Channel Check
5	Timer 2 Output
4	Toggle with Each Refresh Request
3	Enable Channel Check
2	Enable Parity Check
1	Speaker Data Enable
0	Timer 2 Gate to Speaker

Figure 3-22. System Control Port B (Read)

Bit 7 Setting this bit to 1 during a write operation, resets IRQ 0. When this port is read and this bit is set to 1, a parity check has occurred.

Bit 6 When this port is read and this bit is set to 1, a channel check has occurred.

Bit 5 When this port is read, this bit indicates the condition of the timer/counter 2 'output' signal.

Bit 4 When this port is being read, this bit toggles for each refresh request.

Bit 3 Setting this bit to 0 enables channel check. It is set to 1 during a power-on reset.

Bit 2 Setting this bit to 0 enables parity checking. This bit is set to 1 during a power-on reset.

Bit 1 Setting this bit to 1 enables speaker data. This bit is set to 0 during a power-on reset.

Bit 0 Setting this bit to 1 enables the timer 2 gate.

System Control Port A (Hex 0092)

Bit	Function
7, 6	Fixed Disk Activity Light
5	Reserved
4	Watchdog Timer Status
3	Security Lock Latch
2	Reserved
1	Alternate Gate A20
0	Alternate Hot Reset

Figure 3-23. System Control Port A (Hex 0092)

Bits 7, 6 These bits control the fixed-disk activity light. Setting either bit to 1 turns the fixed-disk activity light on. Setting both bits to 0 turns the light off. The power-on reset condition of each bit is 0.

Bit 5 Reserved.

Bit 4 This read-only bit indicates the Watchdog Timer status. When this bit is set to 1, a Watchdog time-out has occurred. For more information about the Watchdog Timer, refer to the *Hardware Interface Technical Reference*.

Bit 3 This bit provides the security lock for the secured area of RT/CMOS. Setting this bit to 1 electrically locks the 8-byte, power-on password. Once this bit is set by POST, it can only be cleared by turning the system off.

Bit 2 Reserved.

Bit 1 This bit is used to enable the 'address 20' signal (A20) when the microprocessor is in the real address mode. When this bit is set to 0, A20 cannot be used in real mode addressing. This bit is set to 0 during a system reset.

Bit 0 This bit provides an alternate method of resetting the system microprocessor. This alternate method supports operating systems requiring faster operation than was provided on the IBM Personal Computer AT®. Resetting the system microprocessor is used to switch the microprocessor from the protected mode to the real address mode. The alternate reset takes 13.4 microseconds.

This bit is set to 0 either by a system reset or a Write operation. When a Write operation changes this bit from 0 to 1, the alternate reset pin is pulsed high for 100 to 125 nanoseconds. The reset occurs after a minimum delay of 6.72 microseconds. While the reset is occurring, the latch remains set so that POST can read this bit. If the bit is 0, POST assumes the system was just powered on. If the bit is 1, POST assumes a switch from the protected mode to the real mode has taken place.

IBM Personal Computer AT is a registered trademark of the International Business Machines Corporation.

Power-On Password

RT/CMOS RAM has 8 bytes reserved for the power-on password and its check character. The 8 bytes are initialized to hex 00. The microprocessor can only access these bytes during power-on self-test (POST). After POST is completed, if a power-on password is installed, the password bytes are locked and cannot be accessed by a program. A power-on password can be from 1 to 7 characters.

During power-on password installation, the password (1 to 7 keyboard scan codes), is stored in the security space.

Power-on password installation is a function of a program contained on the Reference diskette. Once the power-on password utility has been installed, the password can be changed only during the POST. When the new power-on password is installed, changed, or removed, the password is not visible on the display.

The system unit cover can be physically locked to prevent unauthorized access to the battery. This helps prevent unauthorized battery removal and loss of power-on password and configuration information.

For information about the keyboard password, see the "Keyboard and Auxiliary Device Controller" section in the *Hardware Interface Technical Reference.*

Hardware Compatibility

The Model 60 maintains many of the interfaces used by the IBM Personal Computer AT. In most cases command and status organization of these interfaces is maintained.

The functional interfaces for the Personal System/2 products are compatible with the following interfaces:

- The Intel[2] 8259 interrupt controllers (without edge triggering).
- The Intel 8253 timers driven from 1.193 MHz (timer 0 and 2 only).
- The Intel 8237 DMA controller-address/transfer counters, page registers and status fields only. The Command and Request registers are not supported. The rotate and mask functions are not supported. The Mode register is partially supported.
- The NS16450 serial port.
- The Intel 8088, 8086, and 80286 microprocessors.
- The Intel 8272 diskette drive controller.
- The Motorola[3] MC146818 Time of Day Clock command and status (CMOS reorganized).
- The Intel 8042 keyboard port at address hex 0060.
- Display modes supported by the IBM Monochrome Display and Printer Adapter, the IBM Color/Graphics Monitor Adapter, and the IBM Enhanced Graphics Adapter.
- The parallel printer ports (Parallel 1, Parallel 2, and Parallel 3) in compatibility mode.
- Generally compatible with the Intel 80287 and 8087 math coprocessors.

---

[2] Intel is a registered trademark of the Intel Corporation.

[3] Motorola is a trademark of Motorola, Inc.

Index

A

activity light, fixed disk 3-23
adapter ID (0000) 3-3
adapter setup 2-4
address maps
  memory 3-8
  POS 2-5
  system 1-6
address space, POS 2-5
alternate reset 3-24
arbiter programming 3-5
arbiter refresh 3-4
arbitration level 3-4
arbitration register 3-5
arbitration, central control point 3-5
A20 gate 3-23

B

base memory bytes 3-20
battery 3-10
block diagram, system board 1-5
bus contention 2-3
bus priority assignments 3-4

C

card selected feedback 2-6
-CD SFDBK 2-6
central arbitration control point 3-5
channel check latch 3-22
CMOS RAM 3-10
configuration CRC bytes 3-20
configuration, RT/CMOS 3-16
connectors, memory 3-9
control ports, system 3-21

D

date century byte 3-20
diagnostic status byte (RT/CMOS) 3-16
diskette drive enable 2-9
diskette drive type byte 3-17
DMA channels 3-4

E

electrical requirements 1-7
enable I/O functions 2-8
enable NMI 3-11
enable/setup register 2-7
enable, channel check 3-22
equipment byte 3-19
extended arbitration 3-6

F

features, system 1-4
feedback, card selected 2-6
first fixed disk type byte 3-18
fixed disk drive initialization 3-17
fixed-disk activity light 3-23

G

gate A20 3-23

H

heat output 1-7

I

I/O address map 1-6
I/O function byte 2-8
identification, adapter 3-3
interrupt 0 reset 3-22
interrupts

K

keyboard (electrical) 1-7

M

map, memory 3-8
math coprocessor 1-4
memory
  connectors 3-9
  control register 2-9
  map 3-8
  memory module packages 3-7
  performance 1-7
  random access 3-7
  read only 3-7
  refresh 1-7
memory refresh 3-4
memory size miscompare 3-17
microprocessor
  alternate reset 3-24
  real address mode 3-23
miscellaneous ports 3-21
mode switch compatibility 3-24
model byte 1-3
modules, memory 3-7

N

NMI mask 3-11
NMI service 3-4
NMI signal 3-4
nonmaskable interrupt (NMI) 3-21

P

parallel port 2-8
parity check enable 3-22
password, power-on 3-25
performance, system 1-7
pointing device (electrical) 1-7
port, system control 3-21, 3-23
POS
  See programmable option select
power specifications (POS) 1-7
programmable option select 2-3
  enable/setup register 2-7
  registers 2-5
  system board I/O byte 2-8
  system board setup 2-6
  system configuration
    utilities 2-3
programming, arbiter 3-5

R

RAM enable 2-9
random access memory (RAM) 3-7
read-only memory 3-7
real mode addressing 3-23
real-time clock bytes 3-13
refresh 3-4, 3-7
refresh request 3-22
registers
  card selected feedback 2-6
  enable/setup 2-7
  memory control 2-9
  miscellaneous system 3-21
  port A 3-23
  port B 3-21
  POS 2-5, 2-8
  RT/CMOS and NMI mask 3-11
  RT/CMOS status registers 3-13
reset, alternate 3-24
RT/CMOS address register 3-11
RT/CMOS RAM 3-10
  password 3-23, 3-25
  secured area 3-23

S

second fixed disk type byte 3-18
serial port 2-8
setup
  system board 2-6
  system board I/O byte 2-8
  warning 2-3
shutdown status byte 3-17
size 1-7
speaker data enable 3-22
specifications 1-7
status registers (RT/CMOS) 3-13
status, watchdog 3-23
switch to real mode 3-24
system address map 1-6
system board
  block diagram 1-5
  features 1-4
  I/O address map 1-6
  I/O byte 2-8
  memory 3-7
  memory connectors 3-9
  memory enable 2-9
  memory map 3-8
  setup 2-4, 2-6
system configuration utilities 2-3
system ports, miscellaneous 3-21
system reset 3-21
system unit specifications 1-7

T

time status indicator 3-17
timer status, watchdog 3-23
type 1 memory connector 3-9

V

valid RAM bit 3-16
video subsystem enable/setup 2-7
video subsystem setup 2-4

W

warning, POS 2-3
watchdog status 3-23
weight 1-7

Notes:

Personal System/2 Model 70 Technical Reference

First Edition (May, 1988)

The following paragraph does not apply to the United Kingdom or any country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.

It is possible that this publication may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

THE PUBLICATION OF THE INFORMATION CONTAINED HEREIN IS NOT INTENDED TO AND DOES NOT CONVEY ANY RIGHTS OR LICENSES, EXPRESS OR IMPLIED, UNDER ANY IBM PATENTS, COPYRIGHTS, TRADEMARKS, MASK WORKS OR ANY OTHER INTELLECTUAL PROPERTY RIGHTS.

Requests for copies of this publication and for technical information about IBM products should be made to your IBM Authorized Dealer or your IBM Marketing Representative.

Personal System/2 is a registered trademark of the International Business Machines Corporation.

© Copyright International Business Machines Corporation 1988. All rights reserved. No part of this work may be reproduced or distributed in any form or by any means without prior permission in writing from the IBM Corporation.

Preface

The Technical Reference library is intended for those who develop hardware and software products for IBM Personal Computers and IBM Personal System/2. Users should understand computer architecture and programming concepts.

This technical reference provides hardware and software interface information for the IBM Personal System/2 Model 70 and should be used with the following publications:

IBM Personal System/2 Hardware Interface Technical Reference

IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference

This manual consists of the following sections:

Section 1, "System Overview," describes the system, features, and specifications.

Section 2, "Programmable Option Select," describes the registers used for configuration.

Section 3, "System Board," describes the system-specific hardware implementations.

Warning: The term "Reserved" describes certain signals, bits, and registers that should not be changed. Use of reserved areas can cause compatibility problems, loss of data, or permanent damage to the hardware. When the contents of a register are changed, the state of the reserved bits must be preserved. When possible, read the register first and change only the bits that must be changed.

For information about components or devices not described in this manual, refer to the *Hardware Interface Technical Reference.* Information about diskette drives, fixed disk drives, adapters, and external options are in separate option technical references.

Notes:







Contents

Section 1. System Overview 1-1
Description 1-3
System Board Features 1-3
System Board Block Diagram 1-5
System Board I/O Address Map 1-6
Specifications 1-7

Section 2. Programmable Option Select 2-1
Description 2-3
POS Address Map 2-5
Card Selected Feedback 2-6
System Board Setup 2-6
System Board Enable/Setup Register (Hex 0094) 2-7
System Board POS Register 2 (Hex 0102) 2-8
System Board POS Register 3 (Hex 0103) 2-9
Adapter Enable/Setup Register (Hex 0096) 2-10

Section 3. System Board 3-1
Description 3-3
80387 Math Coprocessor 3-3
Micro Channel 3-3
Matched-Memory Cycles 3-3
Central Arbiter 3-4
Diskette Drive and Fixed Disk Connectors 3-7
Memory 3-11
Read-Only Memory Subsystem 3-11
Random Access Memory Subsystem 3-12
Error Recovery 3-13
Memory Subsystem Control 3-14
System Memory Maps 3-17
System Board Memory Connectors 3-22
Real-Time Clock/Complementary Metal-Oxide Semiconductor RAM 3-24
Miscellaneous System Functions 3-36
Nonmaskable Interrupt 3-36
System Control Port B (Hex 0061) 3-36
System Control Port A (Hex 0092) 3-38
Power-On Password 3-40
Hardware Compatibility 3-41

Index X-1

Figures

1-1. Model and Submodel Bytes 1-3
1-2. System Board Devices and Features 1-4
1-3. System Board 1-5
1-4. System Board I/O Address Map 1-6
1-5. Performance Specifications—Type 1 1-7
1-6. Performance Specifications—Type 2 1-8
1-7. Physical Specifications 1-9
1-8. Accoustical Readings 1-10
2-1. POS I/O Address Map 2-5
2-2. Card Selected Feedback Register (Hex 0091) 2-6
2-3. System Board Enable/Setup Register (Hex 0094) 2-7
2-4. System Board POS Register 2 (Hex 0102) 2-8
2-5. Parallel Port Select Bits 2-8
2-6. System Board POS Register 3 (Hex 0103) 2-9
2-7. Adapter Enable/Setup Register (Hex 0096) 2-10
3-1. Arbitration Bus Priority Assignments 3-4
3-2. Arbitration Register, Write to Hex 0090 3-5
3-3. Arbitration Register, Read Hex 0090 3-5
3-4. Direct Bus Attachment 3-7
3-5. Diskette Drive Connectors 3-9
3-6. Fixed Disk Connector 3-10
3-7. Memory Error Address Reassignment 3-13
3-8. Memory Encoding Register 1 3-14
3-9. Memory Encoding Register 2 3-16
3-10. System Memory Map 1 3-17
3-11. System Memory Map 2 3-18
3-12. System Memory Map 3 3-18
3-13. System Memory Map 4 3-19
3-14. System Memory Map 5 3-19
3-15. System Memory Map 6 3-20
3-16. System Memory Map 7 3-20
3-17. System Memory Map 8 3-21
3-18. System Board Memory Connector 3-22
3-19. Presence Detect Encoding 3-23
3-20. RT/CMOS RAM Address Map 3-24
3-21. RT/CMOS Address Register and NMI Mask (Hex 0070) 3-25
3-22. RT/CMOS Data Register (Hex 0071) 3-25
3-23. Real-Time Clock Bytes 3-27
3-24. Status Register A 3-28

3-25. Status Register B 3-28
3-26. Status Register C 3-30
3-27. Status Register D 3-30
3-28. Diagnostic Status Byte 3-31
3-29. Diskette Drive Type Byte 3-32
3-30. Diskette Drive Type Byte (Bits 7 - 4) 3-32
3-31. Diskette Drive Type Byte (Bits 3 - 0) 3-32
3-32. Equipment Byte 3-34
3-33. Equipment Byte (Bits 7, 6) 3-34
3-34. Equipment Byte (Bits 5, 4) 3-34
3-35. System Control Port B (Write) 3-37
3-36. System Control Port B (Read) 3-37
3-37. System Control Port A 3-38

Section 1. System Overview

Description 1-3
System Board Features 1-3
System Board Block Diagram 1-5
System Board I/O Address Map 1-6
Specifications 1-7

Notes:

Description

The IBM Personal System/2 Model 70 is a self-contained, desktop computer system with a keyboard. It can support two diskette drives and one internal fixed disk drive.

A system can have either a Type 1 or Type 2 system board. The major differences between the system boards are system clock speed, component layout, and memory. Programs identify the type by reading the model and submodel bytes. Interrupt hex 15, function code (AH) = hex C0, returns the model and submodel bytes. The following figure shows these bytes, system board types, and system clock speeds.

Model Byte	Submodel Byte	Revision Code	System Board	System Clock
F8	09	00	Type 1	16 MHz
F8	04	00	Type 2	20 MHz

Figure 1-1. Model and Submodel Bytes

Refer to the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference* for a listing of other systems, and check the supplements section for updates to that listing.

System Board Features

The following figure lists the system board devices and features. The *Hardware Interface Technical Reference* describes devices common to PS/2 products by type number.

Device	Type	Features
Microprocessor	---	80386 32-bit address and 32-bit data interface
System Timers	1	Channel 0 – System timer Channel 2 – Tone generation for speaker Channel 3 – Watchdog timer
ROM Subsystem	---	128KB (KB = 1024 bytes)
RAM Subsystem	---	Type 1 system board, 1 to 6MB (MB = 1,048,576 bytes) Type 2 system board, 2 to 6MB Expandable on the channel
CMOS RAM Subsystem	---	64-byte CMOS RAM with real-time clock/calendar 2KB CMOS RAM extension Battery backup
Video Subsystem	1	Auxiliary connector on the channel Analog output 256KB video memory
Audio Subsystem	1	Driven by: - System-timer channel 2 - The 'audio sum node' signal.
DMA Controller	1	Eight independent DMA channels Single or burst transfers and read verification
Interrupt Controller	1	16 levels of system interrupts Interrupts are level-sensitive
Keyboard/Auxiliary Device Controller	1	Keyboard connector Auxiliary device connector Password security
Diskette Drive Controller	1	Supports: - 720KB formatted diskette density - 1.44MB formatted diskette density.
Serial Controller	2	RS-232C interface Programmable as serial port 1 or 2 FIFO mode and character mode
Parallel Controller	1	Programmable as parallel port 1, 2, or 3 Supports bidirectional input and output
Micro Channel	---	Three channel connectors: - One 16-bit connector with an auxiliary video extension - Two 32-bit connectors with matched-memory extension.
Math Coprocessor Socket	---	Supports 80387 math coprocessor Same clock speed as the system microprocessor
Power Supply	1	110 and 220 Vac support

Figure 1-2. System Board Devices and Features

System Board Block Diagram

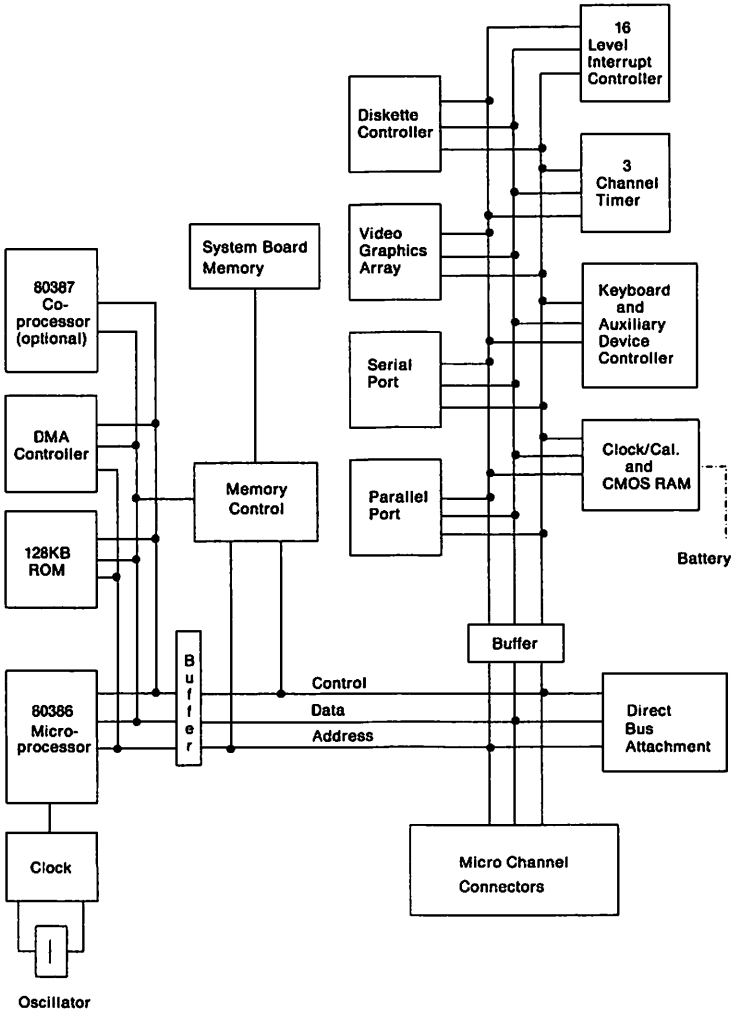


Figure 1-3. System Board

Note: Additional memory can be added in any of the 32-bit Micro Channel connectors. However, the total amount of memory installed should not exceed the 16MB addressing limit of the DMA controller.

System Board I/O Address Map

Hex Addresses	Device
0000 - 001F	DMA Controller (0−3)
0020, 0021	Interrupt Controller (Master)
0040, 0042 - 0044, 0047	System Timers
0060	Keyboard, Auxiliary Device
0061	System Control Port B
0064	Keyboard, Auxiliary Device
0070, 0071	RT/CMOS and NMI Mask
0081 - 0083, 0087	DMA Page Registers (0−3)
0089 - 008B, 008F	DMA Page Registers (4−7)
0090	Central Arbitration Control Point
0091	Card Selected Feedback Register
0092	System Control Port A
0094	System Board Enable/Setup Register
0096	Adapter Enable/Setup Register
00A0 - 00A1	Interrupt Controller (Slave)
00C0 - 00DF	DMA Controller (4−7)
00E0, 00E1	Memory Encoding Registers
00F0 - 00FF	Math Coprocessor
0100 - 0107	Programmable Option Select
01F0 - 01F8	Fixed Disk Drive Controller
0278 - 027B	Parallel Port 3
02F8 - 02FF	Serial Port 2 (RS-232C)
0378 - 037B	Parallel Port 2
03B4, 03B5, 03BA	Video Subsystem
03BC - 03BF	Parallel Port 1
03C0 - 03C5	Video Subsystem
03C6 - 03C9	Video DAC
03CA, 03CC, 03CE, 03CF	Video Subsystem
03D4, 03D5, 03DA	Video Subsystem
03F0 - 03F7	Diskette Drive Controller
03F8 - 03FF	Serial Port 1 (RS-232C)

Figure 1-4. System Board I/O Address Map

Specifications

Device	Number of Waits	Cycle Time (ns)
Microprocessor (16 MHz – 62.5 ns Clock):		
Access to System Board RAM: *		
Memory Read (Page Hit)	0	125
Memory Read (Page Miss)	2	250
Memory Write (Page Hit)	1	187.5
Memory Write (Page Miss)	2	250
Access to Channel:		
Default Transfer Cycle	2	250
Extended Transfer Cycle	4	375
Refresh Rate (Typically performed every 15.1 μs)		625 (min)
Bus Master Access to System Board RAM		300 (min)
DMA Controller (8 MHz – 125 ns Clock):		
Single Transfer: 375 + I/O Access + Memory Access		
Burst Transfers: 375 + (I/O Access + Memory Access)N **		
System Board Memory Access		375
Default Transfer Cycle		250
Extended Transfer Cycle		375

* Adapters installed in the channel should not rely on monitoring system board memory accesses because channel memory control signals may not be present during these accesses.

** N is the number of transfers in the burst.

Figure 1-5. Performance Specifications – Type 1

Device	Number of Waits	Cycle Time (ns)
Microprocessor (20 MHz – 50 ns Clock):		
Access to System Board RAM: *		
Memory Read (Page Hit)	0	100
Memory Read (Page Miss)	2	200
Memory Write (Page Hit)	1	150
Memory Write (Page Miss)	2	200
Access to Channel:		
Default Transfer Cycle	2	200
Extended Transfer Cycle	4	300
Refresh Rate (Typically performed every 15.1 μs)		500 (min)
Bus Master Access to System Board RAM		300 (min)
DMA Controller (10 MHz – 100 ns Clock):		
Single Transfer: 300 + I/O Access + Memory Access		
Burst Transfers: 300 + (I/O Access + Memory Access)N **		
System Board Memory Access		300
Default Transfer Cycle		200
Extended Transfer Cycle		300

* Adapters installed in the channel should not rely on monitoring system board memory accesses because channel memory control signals may not be present during these accesses.
** N is the number of transfers in the burst.

Figure 1-6. Performance Specifications – Type 2

Size:	
Width	360 mm (14.1 in)
Depth	420 mm (16.5 in)
Height	140 mm (5.5 in)
Weight	9.55 kg (21 lb)
Cables:	
Power Cable	1.8 m (6 ft)
Keyboard Cable	1.8 m (6 ft)
Air Temperature:	
System On	15.6 to 32.2°C (60 to 90°F)
System Off	10.0 to 43.0°C (50 to 110°F)
Humidity:	
System On	8% to 80%
System Off	20% to 80%
Maximum Altitude	2133.6 m (7000 ft)
Heat Output	220 W/hr
Accoustical Readings	
See Figure 1-8 on page 1-10	
Electrical:	
Power Supply Input:	
Low Range	90 (min) - 137 (max) Vac
High Range	180 (min) - 265 (max) Vac
Maximum Current Draw :	
Low Range	5.0 A
High Range	3.0 A
Frequency	50 ± 3Hz/60 ± 3Hz
Maximum Current Allowable:	
Keyboard Port	275 mA
Auxiliary Device Port	300 mA
Electromagnetic Compatibility	FCC Class B

Figure 1-7. Physical Specifications

Description	L_{WAd} in bels Operate	Idle	L_{pAm} in dB Operate	Idle	$<L_{pA}>_m$ in dB Operate	Idle
Model 70	5.2	5.2	43	43	40	40

Notes:

L_{WAd} is the declared sound power level for the random sample of machines.

L_{pAm} is the mean value of the A-weighted sound pressure levels at the operator position (if any) for the random sample of machines.

$<L_{pA}>_m$ is the mean value of the A-weighted sound pressure levels at the one-meter positions for the random sample of machines.

All measurements made in accordance with ANSI S12.10, and reported in conformance with ISO DIS 9296.

Figure 1-8. Accoustical Readings

Section 2. Programmable Option Select

Description 2-3
POS Address Map 2-5
Card Selected Feedback 2-6
System Board Setup 2-6
System Board Enable/Setup Register (Hex 0094) 2-7
System Board POS Register 2 (Hex 0102) 2-8
System Board POS Register 3 (Hex 0103) 2-9
Adapter Enable/Setup Register (Hex 0096) 2-10

Notes:

Description

Programmable Option Select (POS) eliminates the need for switches by replacing their function with programmable registers. This section describes the POS information used on the Model 70 system boards. For additional POS information, refer to the *Hardware Interface Technical Reference*.

Warning:

- IBM recommends that programmable options be set only through the System Configuration utilities. Directly setting the POS registers or CMOS RAM POS parameters can result in multiple assignment of the same system resource, improper operation of the feature, loss of data, or possible damage to the hardware.
- Application programs should not use the adapter identification (ID) unless absolutely necessary. Compatibility problems can result.
- If an adapter and the system board are in setup mode at the same time, bus contention will occur, no useful programming can take place, and damage to the hardware can occur.
- After setup operations are complete, the Adapter Enable/Setup register (hex 0096) should be set to hex 00, and the System Board Enable/Setup register (hex 0094) should be set to hex FF.
- The channel reset bit (bit 7) in the Adapter Enable/Setup register must be 0 to program the adapters.
- The system board does not support 16-bit I/O operations to 8-bit POS registers. Using 16-bit I/O instructions on 8-bit POS registers will cause erroneous data to be written to or read from the registers. Only 8-bit transfers are supported for setup operations.

Setup functions respond to I/O addresses hex 0100 through 0107 only when their unique setup signal is active. The following precautions must be taken before setting individual bits in the POS registers.

System Board Video Subsystem Setup:

- Bit 5 in the System Board Enable/Setup register (hex 0094) must be set to 0 to place the system board video into the setup mode.
- Bit 3 in the Adapter Enable/Setup register (hex 0096) must be set to 0 to avoid driving a 'setup' signal to an adapter.
- Bit 7 in the System Board Enable/Setup register must be set to 1 to avoid driving a 'setup' signal to other system board functions.

Adapter Setup:

- Bit 3 in the Adapter Enable/Setup register must be set to 1 to allow adapter setup.
- Bit 5 in the System Board Enable/Setup register must be set to 1 to avoid driving a 'setup' signal to the Video Subsystem.
- Bit 7 in the System Board Enable/Setup register must be set to 1 to avoid driving a 'setup' signal to a system board function.

Other System Board Setup:

- Bit 7 in the System Board Enable/Setup register must be set to 0 to allow setup of other system board functions.
- Bit 3 in the Adapter Enable/Setup register must be set to 0 to avoid driving a 'setup' signal to an adapter.
- Bit 5 in the System Board Enable/Setup register must be set to 1 to avoid driving a 'setup' signal to the Video Subsystem.

POS Address Map

The following figure shows the organization of the I/O address space used by POS. Bit 0 of POS Register 2 and bits 6 and 7 of POS Register 5 are fixed. All other bits in POS Registers 2 through 5 are free-form.

Address (Hex)	Function
0094	System Board Enable/Setup Register
0096	Adapter Enable/Setup Register
0100	POS Register 0—Adapter Identification Byte (Low Byte)
0101	POS Register 1—Adapter Identification Byte (High Byte)
0102	POS Register 2—Option Select Data Byte 1 Bit 0 is Card Enable.
0103	POS Register 3—Option Select Data Byte 2
0104	POS Register 4—Option Select Data Byte 3
0105	POS Register 5—Option Select Data Byte 4 Bit 7 is the channel check active indicator Bit 6 is the channel check status-available indicator
0106	POS Register 6—Subaddress Extension (Low Byte)
0107	POS Register 7—Subaddress Extension (High Byte)

Figure 2-1. POS I/O Address Map

Card Selected Feedback

When an adapter is addressed, it responds by setting the '-card selected feedback' signal (-CD SFDBK) to active. -CD SFDBK is derived from the address decode and driven by a totem pole driver. It is latched by the system board and can be read through the Card Selected Feedback register at address hex 0091. Diagnostic and automatic configuration programs use this signal to verify the operation of an adapter at a given address or DMA port. This signal must not be active during a setup cycle.

The Card Selected Feedback register is a read-only register at address hex 0091. It allows programs to monitor -CD SFDBK and thereby determine if the video subsystem, system board I/O, or an adapter is addressed and functioning.

Bit	Function
7 - 1	Reserved
0	-Card Selected Feedback

Figure 2-2. Card Selected Feedback Register (Hex 0091)

Bits 7 - 1 Reserved.

Bit 0 This bit is set to 1 whenever -CD SFDBK was active on a previous cycle or whenever the system board I/O functions (diskette drive, serial, or parallel interfaces) are accessed by an I/O cycle. Reading this register resets the bit to 0.

System Board Setup

The integrated I/O functions on the system board use POS information during setup. The diskette drive controller, serial port, and parallel port are treated as a single device. The video subsystem is also an integrated part of the system board, however, POS treats it as a separate device. The System Board Enable/Setup register is used to place the system board or the video subsystem into the setup mode.

System Board Enable/Setup Register (Hex 0094)

This is a read/write register; all bits in this register default to 1 (enabled).

Bit	Function
7	Enable/-Setup System Board Functions
6	Reserved
5	Enable/-Setup Video Subsystem
4 - 0	Reserved

Figure 2-3. System Board Enable/Setup Register (Hex 0094)

Bit 7 When this bit is set to 0, various system board I/O functions are placed in the setup mode. The diskette drive controller, serial port, and parallel port are controlled through System Board POS Register 2 (hex 0102). The POS information for memory is in System Board POS Register 3 (hex 0103).

When this bit is set to 1, the system board function is enabled.

Bit 6 Reserved.

Bit 5 When this bit is set to 0, the video subsystem is placed in the setup mode and controlled through POS Register 2 (hex 0102). When set to 1, and bit 0 in hex 0102 is set to 1, video is enabled.

Bit 0 of POS Register 2 is the video enable bit. When this bit is set to 0, the video subsystem does not respond to commands, addresses, or data. If video is being generated when the video enable bit is set to 0, the output is still generated. For information on BIOS calls to enable or disable the video, see the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference.*

Note: When video is disabled, access to the video DAC registers is disabled.

Bits 4 - 0 Reserved.

System Board POS Register 2 (Hex 0102)

When the system board is in the setup mode, the diskette drive controller, serial port, and parallel port are controlled by this read/write register. Reading this register returns the current state of these system board functions.

Bit	Function
7	Disable Parallel Port Extended Mode
6, 5	Parallel Port Select
4	Enable Parallel Port
3	Serial Port Select
2	Enable Serial Port
1	Enable Diskette Drive Interface
0	Enable System Board

Figure 2-4. System Board POS Register 2 (Hex 0102)

Bit 7 When set to 0, this bit allows the parallel port to be configured as an 8-bit, parallel, bidirectional interface. When set to 1, this bit disables the bidirectional mode. This bit is set to 0 at power-on and POST sets it to 1.

Bits 6, 5 These bits select the configuration of the system board parallel port.

Bits 6 5	Assignment	Hex Address	Interrupt Level
0 0	Parallel 1	03BC - 03BF	7
0 1	Parallel 2	0378 - 037B	7
1 0	Parallel 3	0278 - 027B	7
1 1	Reserved	----------	-

Figure 2-5. Parallel Port Select Bits

Bit 4 When this bit and bit 0 are set to 1, the system board parallel port is enabled.

Bit 3 When set to 1, this bit sets the system board serial port as Serial 1 (addresses hex 03F8 through 03FF), which uses interrupt level 4. When set to 0, this bit sets the serial port as Serial 2 (addresses hex 02F8 through 02FF), which uses interrupt level 3.

Bit 2 When this bit and bit 0 are set to 1, the system board serial port is enabled.

Bit 1 When this bit and bit 0 are set to 1, the diskette drive interface is enabled.

Bit 0 When set to 1, this bit allows bits 4, 2, and 1 to enable and disable their respective devices. When set to 0, this bit disables the diskette drive interface, system board serial port, and system board parallel port, regardless of the state of bits 4, 2, and 1.

System Board POS Register 3 (Hex 0103)

This read-only register is accessed while the system board is in the setup mode. It contains information about the presence and type of memory card installed in each system board connector. The memory connectors are numbered 1-2-3 from left to right as viewed from the front of the system.

Bit	Function
7	Reserved
6	2MB/-1MB Card, Connector 3
5	-Card Present, Connector 3
4	Reserved
3	2MB/-1MB Card, Connector 2
2	-Card Present, Connector 2
1	2MB/-1MB Card, Connector 1
0	-Card Present, Connector 1

Figure 2-6. System Board POS Register 3 (Hex 0103)

Bits 7, 4 These bits are reserved.

Bits 6, 3, 1 Each bit indicates the type of memory card installed in the connector indicated. When set to 1, the bit indicates that the card installed is a 2MB memory card. When set to 0, the bit indicates that the card installed is a 1MB memory card.

Bits 5, 2, 0 When set to 0, each bit indicates that a memory card is installed in the connector indicated.

Adapter Enable/Setup Register (Hex 0096)

The Adapter Enable/Setup register selects the connector to be configured.

Bit	Symbol
7	Channel Reset
6 - 4	Reserved
3	Card Setup Enable
2 - 0	Channel Select 2 - 0

Figure 2-7. Adapter Enable/Setup Register (Hex 0096)

Bit 7 When set to 1, this bit activates the 'channel reset' signal to all connectors.

Bits 6 - 4 These bits are reserved.

Bit 3 When set to 1, this bit enables the '-card setup' signal selected by bits 2 through 0.

Bits 2 - 0 These bits are the address bits for the '-card setup' signal. Connectors 1 through 3 are addressed as 0 through 2, respectively, and the fixed disk connector is addressed as 3. When bit 3 is set to 1, these bits select the connector that is put into setup.

Each channel connector has a unique '-card setup' signal (-CD SETUP) associated with it. This signal is used to put the adapters in the setup mode, which allows access to the POS registers. The individual connectors are selected through the Adapter Enable/Setup register. Setup information is then read from or written to the selected adapter through I/O addresses hex 0100 through 0107.

Notes:

1. -CD SETUP only goes active when an operation is performed in the I/O address range hex 0100 through 0107.
2. The status of port hex 0096 can be read by software. However, when the port is read, bits 6, 5, and 4 are set to 1.

Section 3. System Board

Description 3-3
80387 Math Coprocessor 3-3
Micro Channel 3-3
  Matched-Memory Cycles 3-3
  Central Arbiter 3-4
    Arbitration Bus Priority Assignments 3-4
    Central Arbiter Programming 3-5
  Diskette Drive and Fixed Disk Connectors 3-7
    Direct Bus Attachment Connector 3-7
    Diskette Drive Connectors 3-9
    Fixed Disk Drive Connector 3-10
Memory 3-11
  Read-Only Memory Subsystem 3-11
  Random Access Memory Subsystem 3-12
  Error Recovery 3-13
  Memory Subsystem Control 3-14
    Memory Encoding Register 1 (Hex 00E1) 3-14
    Memory Encoding Register 2 (Hex 00E0) 3-16
  System Memory Maps 3-17
  System Board Memory Connectors 3-22
  Real-Time Clock/Complementary Metal-Oxide Semiconductor RAM 3-24
    RT/CMOS Address Register and NMI Mask (Hex 0070) . . 3-25
    RT/CMOS Data Register (Hex 0071) 3-25
    RT/CMOS RAM I/O Operations 3-26
    Real-Time Clock Bytes (Hex 000-00D) 3-27
    CMOS RAM Configuration 3-31
Miscellaneous System Functions 3-36
  Nonmaskable Interrupt 3-36
  System Control Port B (Hex 0061) 3-36
  System Control Port A (Hex 0092) 3-38
  Power-On Password 3-40
Hardware Compatibility 3-41

Notes:

Description

This section describes the math coprocessor, channel differences, memory subsystems, and miscellaneous system ports and connectors for the Model 70. For additional information about these and other topics, refer to the *Hardware Interface Technical Reference.*

80387 Math Coprocessor

The 80387 math coprocessors are matched to the speed of the system microprocessor and operate in the synchronous mode. The 16 MHz and 20 MHz math coprocessors are not interchangeable; their logical operation is the same.

Micro Channel

This section describes the implementation of the Micro Channel architecture on Model 70 systems. For general Micro Channel information, refer to the *Hardware Interface Technical Reference.*

Matched-Memory Cycles

Each 32-bit channel connector has a matched-memory extension. The system microprocessor activates the '-matched memory cycle' signal (-MMC) when it does a memory access. This allows the Model 70 to support the same memory adapter design as the PS/2 Model 80 system board.

Although a memory adapter can respond to -MMC active by driving the '-matched memory cycle request' signal (-MMCR) active, the Model 70 system board ignores -MMCR and performs a default cycle if the 'channel ready' signal (CD CHRDY) is driven active, or a synchronous-extended cycle if CD CHRDY is inactive. The Model 70 also drives the '-matched memory cycle command' signal active at the same time as the '-command' signal.

Central Arbiter

The central arbitration control point gives intelligent subsystems on the channel the ability to share and control the system. It allows burst data transfers and prioritization of control between devices. This arbiter supports up to 16 arbitrating devices.

Arbitration Bus Priority Assignments

The following figure shows the assignment of arbitration levels. The functions with the lowest arbitration level have the highest priority.

ARB Level	Primary Assignment
-2	Memory Refresh
-1	NMI
0	DMA Channel 0 (Programmable to any arbitration level)
1	DMA Channel 1
2	DMA Channel 2
3	DMA Channel 3
4	DMA Channel 4 (Programmable to any arbitration level)
5	DMA Channel 5
6	DMA Channel 6
7	DMA Channel 7
8 - E	Available
F	System Microprocessor

Figure 3-1. Arbitration Bus Priority Assignments

Note: Devices designed for arbitration level 0 or 1 should have limited bandwidth or short bursts so diskette overruns can be prevented or recovered by retry operations. The diskette drive controller, on arbitration level 2, can be held inactive by devices on levels 0 and 1, by a refresh operation, and by the previous controlling master. The diskette drive controller should not be held inactive for more than 12 microseconds to prevent overrun.

NMI service is executed at a priority level higher than 0, called -1. Memory refresh is prioritized at -2, two levels higher than 0. Levels -1 and -2 are reached on the system board only, while the 'arbitrate/-grant' signal is in the arbitrate state.

When the central arbitration control point receives a level -1 request (NMI, a system-board internal signal), it activates -PREEMPT, waits for

the end of transfer, and then places ARB/-GNT in the arbitrate state, which denies channel activity to arbitrating devices. The central arbitration control point gives the grant to the level -1 request, and holds ARB/-GNT in the arbitrate state until the operation is complete and the NMI is reset.

Central Arbiter Programming

The central arbitration control point provides access to programmable options through the Arbitration register, which is accessed at I/O address hex 0090. The bits are defined differently for read and write operations, as shown in the following figures.

Bit	Definition
7	Enable System Microprocessor Cycle
6	Arbitration Mask
5	Enable Extended Arbitration
4 - 0	Reserved

Figure 3-2. Arbitration Register, Write to Hex 0090

Bit	Definition
7	Enable System Microprocessor Cycle
6	Arbitration Masked by NMI
5	Bus Time-out
4	Reserved
3 - 0	Value of Arbitration Bus During Previous Grant State

Figure 3-3. Arbitration Register, Read Hex 0090

Bit 7 Setting this bit to 1 enables system microprocessor cycles during arbitration cycles. This bit can be set to 0 if an arbitrating device requires total control of the channel bandwidth. This bit is set to 0 by a system reset.

Reading this bit as a 1 indicates system microprocessor cycles are enabled during arbitration.

Bit 6 Setting this bit to 1 causes the central arbitration control point to enter the arbitration state. The system microprocessor controls the channel until this bit is reset to 0. This bit is set to 0 by a system reset.

Reading this bit as a 1 indicates that an NMI has occurred and has masked arbitration.

Warning: This bit should be set to 1 only by diagnostic routines and system error-recovery routines.

Bit 5 Setting this bit to 1 enables extended arbitration. The minimum arbitration cycle is 300 nanoseconds; this bit extends that minimum cycle to 600 nanoseconds. This bit is set to 0 during a system reset.

Reading this bit as a 1 indicates that a bus time-out has occurred, and resets bit 6 in this register to 0.

Bit 4 This bit is reserved and should be 0.

Bits 3 - 0 These bits are undefined for a write operation and should be set to 0.

Reading these bits returns the arbitration level of the arbiter controlling the channel during the most recent grant state. This information allows the system microprocessor to determine the arbitration level of the device that caused a bus time-out.

Diskette Drive and Fixed Disk Connectors

The system board has a 2- by 56-pin connector that provides direct bus attachment for two diskette drives and one fixed disk through the fixed disk and diskette drive adapter.

For the diskette drives, this adapter passes control and data signals between the diskette drive controller on the system board and the drives. For the fixed disk drive, the adapter provides a path for the Micro Channel signals necessary for fixed disk operations. It also provides the power required for fixed disk and diskette drive operation.

When in the setup mode, the fixed disk is addressed as channel connector 4 through the Adapter Enable/Setup register. For more information about this register, see "POS Information" in the *Hardware Interface Technical Reference*. For more POS information about the fixed disk drive, refer to the technical reference for that drive.

Direct Bus Attachment Connector

The following figure shows the 2- by 56-pin system board connector.

Side B			Side A		
Pin	**I/O**	**Signal**	**Pin**	**I/O**	**Signal**
1	O	-High Density Select	1	I	-2nd Drive Installed
2	O	+12 Vdc	2	O	+12 Vdc
3	O	+12 Vdc	3	O	+12 Vdc
4	O	+5 Vdc	4	I	CD CHRDY
5	I	-Index	5	O	M/-IO
6	O	-Motor Enable 1	6	N/A	Ground
7	O	-Drive Select 0	7	O	-S1
8	N/A	Ground	8	O	+5 Vdc
9	O	-Drive Select 1	9	O	-S0
10	O	-Motor Enable 0	10	N/A	Ground
11	O	-Direction In	11	N/A	Reserved
12	N/A	Ground	12	O	-TC
13	O	-Step	13	O	ARB/-GNT
14	O	-Write Data	14	N/A	Ground
15	O	-Write Enable	15	I/O	ARB 03

Figure 3-4 (Part 1 of 2). Direct Bus Attachment

Side B			Side A		
Pin	I/O	Signal	Pin	I/O	Signal
16	N/A	Frame Ground	16	I/O	ARB 02
17	I	-Track 0	17	I/O	ARB 01
18	I	-Write Protect	18	N/A	Frame Ground
19	I	-Read Data	19	N/A	Reserved
20	O	-Head 1 Select	20	I/O	ARB 00
21	I	-Diskette Change	21	I	-BURST
22	I	-IRQ 14	22	N/A	Ground
23	I	-CD DS 16	23	I	-PREEMPT
24	N/A	Ground	24	O	+5 Vdc
25	O	-SBHE	25	O	-ADL
26	I/O	D13	26	N/A	Ground
27	O	+12 Vdc	27	O	+12 Vdc
28	I/O	D11	28	O	A00
29	I/O	D10	29	O	A01
30	I/O	D07	30	N/A	Ground
31	I/O	D06	31	O	A02
32	N/A	Ground	32	O	+5 Vdc
33	I/O	D05	33	O	A03
34	I/O	D02	34	N/A	Ground
35	O	+12 Vdc	35	O	+12 Vdc
36	I/O	D00	36	O	A04
37	I/O	D15	37	O	A05
38	I/O	D14	38	N/A	Ground
39	I/O	D12	39	O	A06
40	N/A	Ground	40	O	+5 Vdc
41	I/O	D09	41	O	A07
42	I/O	D08	42	N/A	Ground
43	O	CHRESET	43	N/A	Reserved
44	O	+5 Vdc	44	N/A	Reserved
45	I/O	D04	45	O	A08
46	N/A	Key	46	N/A	Key
47	N/A	Key	47	N/A	Key
48	N/A	Ground	48	O	+5 Vdc
49	I/O	D03	49	O	A09
50	I/O	D01	50	N/A	Ground
51	I	-CD SFDBK	51	O	A10
52	N/A	Ground	52	O	+5 Vdc
53	O	-CMD	53	O	A11
54	O	A12	54	N/A	Ground
55	O	14.3 MHz Osc	55	O	A13
56	N/A	Ground	56	O	+5 Vdc
57	O	A14	57	O	-CD SETUP
58	O	A15	58	N/A	Ground

Figure 3-4 (Part 2 of 2). Direct Bus Attachment

Diskette Drive Connectors

The following figure shows the signal assignment and pin numbering for the diskette drive connectors on the adapter. The diskette drive connectors are 2- by 20-pin connectors that are numbered with the odd numbers on the top.

Pin	I/O	Signal	Pin	I/O	Signal
1	I	-2nd Drive Installed	2	O	-High Density Select
3	N/A	Reserved	4	N/A	Reserved
5	N/A	Ground	6	N/A	Reserved
7	N/A	Signal Ground	8	I	-Index
9	N/A	Signal Ground	10	N/A	Reserved
11	N/A	Signal Ground	12	O	-Drive Select
13	N/A	Ground	14	N/A	Reserved
15	N/A	Signal Ground	16	O	-Motor Enable
17	N/A	Signal Ground	18	O	-Direction In
19	N/A	Signal Ground	20	O	-Step
21	N/A	Signal Ground	22	O	-Write Data
23	N/A	Signal Ground	24	O	-Write Enable
25	N/A	Signal Ground	26	I	-Track 0
27	N/A	Signal Ground	28	I	-Write Protect
29	N/A	Signal Ground	30	I	-Read Data
31	N/A	Signal Ground	32	O	-Head 1 Select
33	N/A	Signal Ground	34	I	-Diskette Change
35	N/A	Frame Ground	36	N/A	Frame Ground
37	N/A	Ground	38	O	+5 Vdc
39	N/A	Ground	40	O	+12 Vdc

Figure 3-5. Diskette Drive Connectors

Fixed Disk Drive Connector

The following figure shows the signal assignment and pin numbering for the fixed disk drive connector on the adapter. The fixed disk drive connector is a 2- by 36-pin connector. Side A of the connector is the top and Side B is the bottom.

Side A			Side B		
Pin	**I/O**	**Signal**	**Pin**	**I/O**	**Signal**
1	O	-CD SETUP	1	O	A15
2	O	A13	2	O	A14
3	N/A	Ground	3	N/A	Ground
4	O	A11	4	O	14.3 MHz Osc
5	O	A10	5	N/A	Ground
6	O	A09	6	O	A12
7	O	+5 Vdc	7	O	-CMD
8	O	A08	8	I	-CD SFDBK
9	O	A07	9	N/A	Ground
10	O	A06	10	I/O	D01
11	N/A	Ground	11	I/O	D03
12	O	A05	12	I/O	D04
13	O	A04	13	N/A	Ground
14	O	A03	14	O	CHRESET
15	O	+5 Vdc	15	I/O	D08
16	O	A02	16	I/O	D09
17	O	A01	17	N/A	Ground
18	O	A00	18	I/O	D12
19	O	+12 Vdc	19	I/O	D14
20	O	-ADL	20	I/O	D15
21	I	-PREEMPT	21	N/A	Ground
22	I	-BURST	22	I/O	D00
23	O	+5 Vdc	23	I/O	D02
24	I	ARB 00	24	I/O	D05
25	I	ARB 01	25	N/A	Ground
26	I	ARB 02	26	I/O	D06
27	N/A	+12 Vdc	27	I/O	D07
28	I	ARB 03	28	I/O	D10
29	O	ARB/-GNT	29	N/A	Ground
30	O	-TC	30	I/O	D11
31	N/A	+5 Vdc	31	I/O	D13
32	O	-S0	32	O	-SBHE
33	O	-S1	33	N/A	Ground
34	O	M/-IO	34	I	-CD DS 16
35	N/A	Ground	35	I	-IRQ 14
36	I	CD CHRDY	36	N/A	Ground

Figure 3-6. Fixed Disk Connector

Memory

Model 70 systems use the following types of memory:

- Read-only memory (ROM)
- Random access memory (RAM)
- Real-time Clock and CMOS RAM.

Read-Only Memory Subsystem

The ROM subsystem consists of 128KB in a 64Kb by 16-bit arrangement. ROM is active at power-on at address space hex 000E0000 to 000FFFFF. After POST ensures the proper operation of system memory, the ROM code is copied to RAM at the same address space, and ROM is disabled. ROM is assigned the top of the first and last 1MB of address space (000E0000 and FFFE0000).

ROM or RAM access at address space hex 000E0000 to 000FFFFF is controlled by the ROMEN bit in Memory Encoding Register 1 (hex 00E1). When enabled, ROM is not parity-checked and operates with four 50-nanosecond wait states.

Random Access Memory Subsystem

The RAM subsystem on the system board starts at address hex 00000000 of the address space. Memory is attached to the system board by three connectors. The RAM subsystem is 36 bits wide: 32 data bits and 4 parity bits. One parity bit is generated for each byte of data written. During a read operation, one parity bit is checked for each byte of data read by the device controlling the bus.

The Model 70 enables and disables memory in 1MB blocks. Each 1MB block must start on a 1MB boundary. Because 128KB of I/O ROM, and 128KB of video memory are mapped within the first 1MB address space, an overflow is created from the first 1MB of RAM installed in the system. The first 1MB of RAM can be split at either 512KB or 640KB; the memory from the split to the start of the RAM that contains the ROM code is called the *split memory block*. This split memory block can be disabled, or it can be remapped to the first address following the last 1MB of memory. Split memory block remapping and disabling are controlled by the memory encoding registers.

Note: Additional memory can be added in either of the 32-bit Micro Channel connectors. The total amount of memory installed should not exceed the 16MB addressing limit of the DMA controller.

Error Recovery

If POST detects a memory error in the first 512KB of memory, the first physical 1MB block of memory is deactivated. The addresses assigned to the deactivated block are reassigned to the second physical block of system board memory (the first 1MB of system memory must reside in connector 1 or 2). If 1MB of valid memory cannot be allocated from connector 1 or 2, the system cannot recover from the error. The first 1MB of memory address space cannot be assigned to memory on the channel.

POST does not deactivate a memory block if an error is detected in the second 512KB of that block. If an error is detected in that area of memory, the 1MB block containing the error can be deactivated and have its addresses reassigned by running the customer-level memory diagnostics program on the Reference Diskette supplied with the system. Once a block of memory is deactivated, it is ignored by POST on subsequent power-ons.

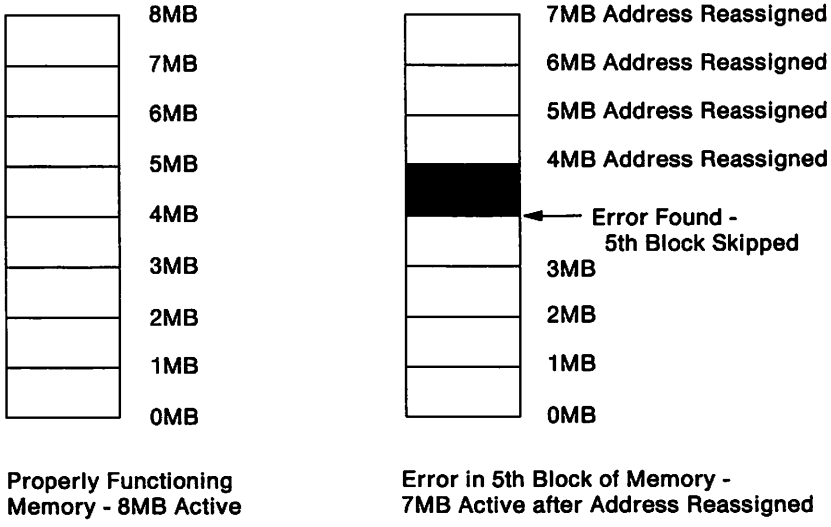


Figure 3-7. Memory Error Address Reassignment

Memory Subsystem Control

These read/write registers define and control the ROM and RAM subsystems.

Memory Encoding Register 1 (Hex 00E1)

This register determines how the first 1MB of memory is addressed and is used with Memory Encoding Register 2 to determine the amount of system board memory enabled.

Bit	Function
7	-Card 2 EN2
6	-Card 2 EN1
5	-Card 1 EN2
4	-Card 1 EN1
3	-ENSPLIT
2	-640
1	ROMEN
0	-ENPLRPCH

Figure 3-8. Memory Encoding Register 1

Bits 7, 6 These bits define system board memory in connector 2. When they are set to 0, bit 6 enables the first 1MB block and bit 7 enables the second 1MB block.

Bits 5, 4 These bits define system board memory in connector 1. When they are set to 0, bit 4 enables the first 1MB block and bit 5 enables the second 1MB block.

Bit 3 This bit determines whether the split-memory block is assigned addresses or is disabled. The top 128KB of the first 1MB is always mapped into the ROM address space and the code in ROM is copied into this space.

When this bit is set to 0, the split-memory block is enabled. The split-memory block size is dependent on the value of the 640 bit. Its address is determined in Memory Encoding Register 2. When this bit is set to 1, the split-memory block is disabled.

Note: If the total system memory is equal to or greater than 16MB, the split-memory block cannot be used.

Bit 2 This bit determines where the first active 1MB of memory is split.

When this bit is set to 0, the system maps 640KB of the first 1MB to address hex 00000000 to 0009FFFF. The split-memory block, 256KB, is mapped to the address specified in Memory Encoding register 2.

When this bit is set to 1, the system maps 512KB of the first 1MB to address hex 00000000 to 0007FFFF. The split-memory block, 384KB, is mapped to the address specified in the Memory Encoding register 2. The addresses hex 00080000 to 0009FFFF are unassigned.

Bit 1 This bit determines how addresses hex 000E0000 to 000FFFFF are assigned.

When this bit is set to 1, ROM is enabled and the read-access addresses are assigned to ROM; the write-access addresses are assigned to RAM. When this bit is set to 0, ROM is disabled and read addresses are assigned to RAM; the write addresses are disabled while ROM is disabled.

Note: When this bit is set to 1, system performance will be substantially slower because of 16-bit accesses to ROM and a longer cycle.

Bit 0 When set to 0, this bit enables parity checking of system board memory. To clear a parity error, this bit must be set to 1 and then to 0.

Memory Encoding Register 2 (Hex 00E0)

This register is used with Memory Encoding Register 1 to determine the amount of system board memory that is enabled. It also contains the address for the split-memory block.

Bit	Function
7, 6	Reserved = 1
5	-Card 3 EN2
4	-Card 3 EN1
3	SPA23
2	SPA22
1	SPA21
0	SPA20

Figure 3-9. Memory Encoding Register 2

Bits 7, 6 These bits are reserved.

Bits 5, 4 These bits define the system board memory in connector 3. When they are set to 0, bit 4 enables the first 1MB block and bit 5 enables the second 1MB block.

Bits 3 - 0 These bits define the starting address of the split-memory block. When split memory is enabled, these bits determine its starting address. The starting location can be at any 1MB boundary from 1MB to 15MB.

System Memory Maps

Memory is mapped by the Memory Encoding registers. The mapping results in either 512KB or 640KB of system board RAM starting at address hex 00000000. A 256-byte and 1KB portion and of this RAM is reserved as BIOS data areas. See the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference* for details.

In the following figures, the variable *X* represents the number of 1MB blocks of system board memory starting at or above the hex 00100000 boundary. The variable *Y* represents the number of 1MB blocks of memory installed in the channel starting at or above the hex 00100000 boundary (*Y* = 0 to 15).

The following figure shows the memory mapping when:

Enable Split bit = 1
640 bit = 1
ROM enable bit = 1

Hex Range	Function
00000000 to 0007FFFF	512KB System Board RAM
00080000 to 0009FFFF	Not Used
000A0000 to 000BFFFF	128KB Video RAM
000C0000 to 000DFFFF	Channel ROM
000E0000 to 000FFFFF	128KB System Board ROM
00100000 to (00100000 + XMB)	XMB System Board RAM
(00100000 + XMB) to (00100000 + XMB + YMB)	YMB Channel RAM
(00100000 + XMB + YMB) to FFFDFFFF	Not Used
FFFE0000 to FFFFFFFF	128KB System Board ROM (Same as 000E0000 to 000FFFFF)

Figure 3-10. System Memory Map 1

The following figure shows the memory mapping when:

Enable Split bit = 1
640 bit = 0
ROM enable bit = 1

Hex Range	Function
00000000 to 0009FFFF	640KB System Board RAM
000A0000 to 000BFFFF	128KB Video RAM
000C0000 to 000DFFFF	Channel ROM
000E0000 to 000FFFFF	128KB System Board ROM
00100000 to (00100000 + *XMB*)	*XMB* System Board RAM
(00100000 + *XMB*) to (00100000 + *XMB* + *YMB*)	*YMB* Channel RAM
(00100000 + *XMB* + *YMB*) to FFFDFFFF	Not Used
FFFE0000 to FFFFFFFF	128KB System Board ROM (Same as 000E0000 to 000FFFFF)

Figure 3-11. System Memory Map 2

The following figure shows the memory mapping when:

Enable Split bit = 0
640 bit = 1
ROM enable bit = 1
Split address bits = 1 + *X* + *Y* (Total Range = 1 to 15).

Hex Range	Function
00000000 to 0007FFFF	512KB System Board RAM
00080000 to 0009FFFF	Not Used
000A0000 to 000BFFFF	128KB Video RAM
000C0000 to 000DFFFF	Channel ROM
000E0000 to 000FFFFF	128KB System Board ROM
00100000 to (00100000 + *XMB*)	*XMB* System Board RAM
(00100000 + *XMB*) to (00100000 + *XMB* + *YMB*)	*YMB* Channel RAM
(00100000 + *XMB* + *YMB*) to (00100000 + *XMB* + *YMB* +384KB − 1)	384KB System Board RAM
(00100000 + *XMB* + *YMB* + 384KB) to FFFDFFFF	Not Used
FFFE0000 to FFFFFFFF	128KB System Board ROM (Same as 000E0000 to 000FFFFF)

Figure 3-12. System Memory Map 3

The following figure shows the memory mapping when:

Enable Split bit = 0
640 bit = 0
ROM enable bit = 1
Split address bits = 1 + *X* + *Y* (Total Range = 1 to 15).

Hex Range	Function
00000000 to 0009FFFF	640KB System Board RAM
000A0000 to 000BFFFF	128KB Video RAM
000C0000 to 000DFFFF	Channel ROM
000E0000 to 000FFFFF	128KB System Board ROM
00100000 to (00100000 + *X*MB)	*X*MB System Board RAM
(00100000 + *X*MB) to (00100000 + *X*MB + *Y*MB)	*Y*MB Channel RAM
(00100000 + *X*MB + *Y*MB) to (00100000 + *X*MB + *Y*MB + 256KB − 1)	256KB System Board RAM
(00100000 + *X*MB + *Y*MB + 256KB) to FFFDFFFF	Not Used
FFFE0000 to FFFFFFFF	128KB System Board ROM (Same as 000E0000 to 000FFFFF)

Figure 3-13. System Memory Map 4

The following figure shows the memory mapping when:

Enable Split bit = 1
640 bit = 1
ROM enable bit = 0

Hex Range	Function
00000000 to 0007FFFF	512KB System Board RAM
00008000 to 0009FFFF	Not Used
000A0000 to 000BFFFF	128KB Video RAM
000C0000 to 000DFFFF	Channel ROM
000E0000 to 000FFFFF	128KB System Board ROM mapped to RAM
00100000 to (00100000 + *X*MB)	*X*MB System Board RAM
(00100000 + *X*MB) to (00100000 + *X*MB + *Y*MB)	*Y*MB Channel RAM
(00100000 + *X*MB + *Y*MB) to FFFDFFFF	Not Used
FFFE0000 to FFFFFFFF	128KB System Board ROM

Figure 3-14. System Memory Map 5

The following figure shows the memory mapping when:

Enable Split bit = 1
640 bit = 0
ROM enable bit = 0

Hex Range	Function
00000000 to 0009FFFF	640KB System Board RAM
000A0000 to 000BFFFF	128KB Video RAM
000C0000 to 000DFFFF	Channel ROM
000E0000 to 000FFFFF	128KB System Board ROM mapped to RAM
00100000 to (00100000 + *XMB*)	*XMB* System Board RAM
(00100000 + *XMB*) to (00100000 + *XMB* + *YMB*)	*YMB* Channel RAM
(00100000 + *XMB* + *YMB*) to FFFDFFFF	Not Used
FFFE0000 to FFFFFFFF	128KB System Board ROM

Figure 3-15. System Memory Map 6

The following figure shows the memory mapping when:

Enable Split bit = 0
640 bit = 1
ROM enable bit = 0
Split address bits = 1 + *X* + *Y* (Total Range = 1 to 15).

Hex Range	Function
00000000 to 0007FFFF	512KB System Board RAM
00080000 to 0009FFFF	Not Used
000A0000 to 000BFFFF	128KB Video RAM
000C0000 to 000DFFFF	Channel ROM
000E0000 to 000FFFFF	128KB System Board ROM mapped to RAM
00100000 to (00100000 + *XMB*)	*XMB* System Board RAM
(00100000 + *XMB*) to (00100000 + *XMB* + *YMB*)	*YMB* Channel RAM
(00100000 + *XMB* + *YMB*) to (00100000 + *XMB* + *YMB* + 384KB − 1)	384KB of First 1MB at Split Address
(00100000 + *XMB* + *YMB* + 384KB) to FFFDFFFF	Not Used
FFFE0000 to FFFFFFFF	128KB System Board ROM

Figure 3-16. System Memory Map 7

The following figure shows the memory mapping when:

Enable Split bit = 0
640 bit = 0
ROM enable bit = 0
Split address bits = 1 + *X* + *Y* (Total Range = 1 to 15).

Hex Range	Function
00000000 to 0009FFFF	640KB System Board RAM
000A0000 to 000BFFFF	128KB Video RAM
000C0000 to 000DFFFF	Channel ROM
000E0000 to 000FFFFF	128K System Board ROM mapped to RAM
00100000 to (00100000 + *X*MB)	*X*MB System Board RAM
(00100000 + *X*MB) to (00100000 + *X*MB + *Y*MB)	*Y*MB Channel RAM
(00100000 + *X*MB + *Y*MB) to (00100000 + *X*MB + *Y*MB + 256KB-1)	256KB of First 1MB at Split Address
(00100000 + *X*MB + *Y*MB + 256KB) to FFFDFFFF	Not Used
FFFE0000 to FFFFFFFF	128KB System Board ROM (Same as 000E0000 to 000FFFFF)

Figure 3-17. System Memory Map 8

System Board Memory Connectors

The system board has three 72-pin memory connectors that support the 1MB and the 2MB memory cards. The three connectors are numbered 1-2-3 from the left as viewed from the front of the system. Memory-refresh requests typically occur once every 15 microseconds.

The Type 1 and Type 2 system boards support 1MB and 2MB memory cards. The following figure shows the pin assignments for the 1- by 72-pin memory connectors.

Pin	I/O	Signal	Pin	I/O	Signal
1	N/A	Ground	37	I/O	Parity Data 1
2	I/O	Data 0	38	I/O	Parity Data 3
3	I/O	Data 16	39	N/A	Ground
4	I/O	Data 1	40	O	Column Address Strobe 0
5	I/O	Data 17	41	O	Column Address Strobe 2
6	I/O	Data 2	42	O	Column Address Strobe 3
7	I/O	Data 18	43	O	Column Address Strobe 1
8	I/O	Data 3	44	O	Row Address Strobe 0
9	I/O	Data 19	45	O	Row Address Strobe 1
10	O	+5 Vdc	46	O	Block Select 1
11	O	-Column Address Strobe P	47	O	Write Enable
12	O	Address 0	48	N/A	Reserved
13	O	Address 1	49	I/O	Data 8
14	O	Address 2	50	I/O	Data 24
15	O	Address 3	51	I/O	Data 9
16	O	Address 4	52	I/O	Data 25
17	O	Address 5	53	I/O	Data 10
18	O	Address 6	54	I/O	Data 26
19	N/A	Reserved	55	I/O	Data 11
20	I/O	Data 4	56	I/O	Data 27
21	I/O	Data 20	57	I/O	Data 12
22	I/O	Data 5	58	I/O	Data 28
23	I/O	Data 21	59	O	+5 Vdc
24	I/O	Data 6	60	I/O	Data 29
25	I/O	Data 22	61	I/O	Data 13
26	I/O	Data 7	62	I/O	Data 30
27	I/O	Data 23	63	I/O	Data 14
28	O	Address 7	64	I/O	Data 31
29	O	Block Select 0	65	I/O	Data 15

Figure 3-18 (Part 1 of 2). System Board Memory Connector

Pin	I/O	Signal	Pin	I/O	Signal
30	O	+5 Vdc	66	O	Block Select 2
31	O	Address 8	67	I	Presence Detect 0
32	N/A	Reserved	68	I	Presence Detect 1
33	O	Row Address Strobe 3	69	I	Presence Detect 2
34	O	Row Address Strobe 2	70	I	Presence Detect 3
35	I/O	Parity Data 2	71	O	Block Select 3
36	I/O	Parity Data 0	72	N/A	Ground

Figure 3-18 (Part 2 of 2). System Board Memory Connector

The 'presence detect' signals are used by the system to determine memory card size and memory speed. The pins are either connected to ground (G) or not connected (N). The following table shows those combinations supported by Model 70.

	Presence Detect Signals			
Card Type	**0**	**1**	**2**	**3**
1 MB Memory at 100 ns	G	N	G	G
1 MB Memory at 85 ns	G	N	N	G
2 MB Memory at 85 ns	N	G	N	G

Figure 3-19. Presence Detect Encoding

Real-Time Clock/Complementary Metal-Oxide Semiconductor RAM

The real-time clock/complementary metal-oxide semiconductor RAM (RT/CMOS) chip contains the real-time clock and 64 bytes of CMOS RAM. The internal clock circuitry uses 14 bytes of this memory, and the rest is allocated to configuration and system status information.

In addition to the 64 bytes of CMOS RAM, a 2KB CMOS RAM extension is provided for configuration and other system information.

A 6-Vdc lithium battery maintains voltage to the RT/CMOS RAM and 2KB CMOS RAM extension when the power supply is not in operation.

The system cover can be locked to prevent battery removal and loss of password and configuration information.

The following figure shows the RT/CMOS RAM bytes and their addresses.

Address (Hex)	RT/CMOS RAM Bytes
000 - 00D	Real-Time Clock Bytes
00E	Diagnostic Status Byte
00F	Shutdown Status Byte
010	Diskette Drive Type Byte
011	First Fixed Disk Drive Type Byte
012	Second Fixed Disk Drive Type Byte
013	Reserved
014	Equipment Byte
015 - 016	Low and High Base Memory Bytes
017 - 018	Low and High Expansion Memory Bytes
019 - 031	Reserved
032 - 033	Configuration CRC Bytes
034 - 036	Reserved
037	Date Century Byte
038 - 03F	Reserved

Figure 3-20. RT/CMOS RAM Address Map

RT/CMOS Address Register and NMI Mask (Hex 0070)

This register is used in conjunction with the port at hex 0071 to read and write the RT/CMOS RAM bytes.

Bit	Function
7	NMI Mask
6	Reserved
5 - 0	RT/CMOS RAM Address

Figure 3-21. RT/CMOS Address Register and NMI Mask (Hex 0070)

Warning: The operation following a write to hex 0070 should access port hex 0071; otherwise intermittent malfunctions and unreliable operation of the RT/CMOS RAM can occur.

Bit 7 When this bit is set to 1, the NMI is masked off (the NMI is disabled). This bit is set to 1 by a power-on reset. This is a write-only bit.

Bit 6 Reserved.

Bits 5 - 0 These bits are used to select RT/CMOS RAM addresses.

RT/CMOS Data Register (Hex 0071)

This port is used in conjunction with the address register at hex 0070 to read and write the RT/CMOS RAM bytes.

Bit	Function
7 - 0	RT/CMOS Data

Figure 3-22. RT/CMOS Data Register (Hex 0071)

RT/CMOS RAM I/O Operations

During I/O operations to the RT/CMOS RAM addresses, interrupts should be masked to prevent other interrupt routines from changing the CMOS Address register before data is read or written. After I/O operations, the RT/CMOS and NMI Mask register (hex 0070) should be left pointing to Status Register D (hex 00D).

Warning: The operation following a write to hex 0070 should access hex 0071; otherwise intermittent malfunctions and unreliable operation of the RT/CMOS RAM can occur.

The following steps are required to perform I/O operations to the RT/CMOS RAM addresses:

1. Write the RT/CMOS RAM address to the RT/CMOS and NMI Mask register (hex 0070).
2. Write the data to address hex 0071.

Reading RT/CMOS RAM requires the following steps:

1. Write the RT/CMOS RAM address to the RT/CMOS and NMI Mask register (hex 0070).
2. Read the data from address hex 0071.

Real-Time Clock Bytes (Hex 000-00D)

Bit definitions and addresses for the real-time clock bytes are shown in the following figure.

Address (Hex)	Function	Byte Number
000	Seconds	0
001	Second Alarm	1
002	Minutes	2
003	Minute Alarm	3
004	Hours	4
005	Hour Alarm	5
006	Day of Week	6
007	Date of Month	7
008	Month	8
009	Year	9
00A	Status Register A	10
00B	Status Register B	11
00C	Status Register C	12
00D	Status Register D	13

Figure 3-23. Real-Time Clock Bytes

Note: The Setup program initializes status registers A, B, C, and D when the time and date are set. Interrupt hex 1A is the BIOS interface to read and set the time and date and it initializes the register the same way as the Setup program.

Status Register A (Hex 00A)

Bit	Function
7	Update in Progress
6 - 4	22-Stage Divider
3 - 0	Rate Selection Bits

Figure 3-24. Status Register A

Bit 7 When set to 1, this bit indicates the time-update cycle is in progress. When set to 0, it indicates the current date and time can be read.

Bits 6 - 4 These three divider-selection bits identify which time-base frequency is being used. The system initializes these bits to binary 010, which selects a 32.768 kHz time base. This is the only value supported by the system for proper time-keeping.

Bits 3 - 0 These bits allow the selection of a divider output frequency. The system initializes the rate selection bits to a binary 0110, which selects a 1.024 kHz square-wave output frequency and a 976.562-microsecond periodic interrupt rate.

Status Register B (Hex 00B)

Bit	Function
7	Set
6	Periodic Interrupt Enable
5	Alarm Interrupt Enable
4	Update-Ended Interrupt Enabled
3	Square Wave Enabled
2	Date Mode
1	24-Hour Mode
0	Daylight Savings Enabled

Figure 3-25. Status Register B

Bit 7 When set to 0, this bit updates the cycle, normally by advancing the counts at a rate of one per second. When set to 1, this bit immediately ends any update cycle in progress, and the program can initialize the 14 time bytes without any further updates occurring until this bit is set to 0.

Bit 6 This bit is a read/write bit that allows an interrupt to occur at a rate specified by the rate and divider bits in Status Register A. When set to 1, this bit enables the interrupt. The system initializes this bit to 0.

Bit 5 When set to 1, this bit enables the alarm interrupt. The system initializes this bit to 0.

Bit 4 When set to 1, this bit enables the update-ended interrupt. The system initializes this bit to 0.

Bit 3 When set to 1, this bit enables the square-wave frequency as set by the rate-selection bits in Status Register A. The system initializes this bit to 0.

Bit 2 This bit indicates if the time-and-date calendar updates use binary or binary-coded-decimal (BCD) formats. When set to 1, this bit indicates a binary format. The system initializes this bit to 0.

Bit 1 This bit establishes if the hours byte is in the 24-hour or 12-hour mode. When set to 1, this bit indicates the 24-hour mode. The system initializes this bit to 1.

Bit 0 When set to 1, this bit enables the daylight savings time mode. When set to 0, it disables the mode, and the clock reverts to standard time. The system initializes this bit to 0.

Status Register C (Hex 00C)

Bit	Function
7	Interrupt Request Flag
6	Periodic Interrupt Flag
5	Alarm Interrupt Flag
4	Update-Ended Interrupt Flag
3 - 0	Reserved

Figure 3-26. Status Register C

Note: Interrupts are enabled by bits 6, 5, and 4 in Status Register B.

Bit 7 This bit is used in conjunction with bits 6, 5, and 4. When set to 1, this bit indicates that an interrupt has occurred; bits 6, 5, and 4 indicate the type of interrupt.

Bit 6 When set to 1, this bit indicates that a periodic interrupt occurred.

Bit 5 When set to 1, this bit indicates that an alarm interrupt occurred.

Bit 4 When set to 1, this bit indicates that an update-ended interrupt occurred.

Bits 3 - 0 Reserved.

Status Register D (Hex 00D)

Bit	Function
7	Valid RAM
6 - 0	Reserved

Figure 3-27. Status Register D

Bit 7 This read-only bit monitors the power-sense pin. A low state of this pin indicates a loss of power to the real-time clock (dead battery). When set to 1, this bit indicates that the real-time clock has power. When set to 0, it indicates that the real-time clock has lost power.

Bits 6 - 0 Reserved.

CMOS RAM Configuration

The following shows the bit definitions for the CMOS RAM configuration bytes.

Diagnostic Status Byte (Hex 00E)

Bit	Function
7	Real-Time Clock Chip Power
6	Configuration Record and Checksum Status
5	Incorrect Configuration
4	Memory Size Miscompare
3	Fixed Disk Controller/Drive C Initialization Status
2	Time Status Indicator
1	Adapter Configuration Miscompare
0	Adapter ID Time-Out

Figure 3-28. Diagnostic Status Byte

Bit 7 When set to 1, this bit indicates the real-time clock chip lost power.

Bit 6 When this bit is set to 1, the checksum is incorrect.

Bit 5 This is a check, at power-on time, of the Equipment byte. When set to 1, the configuration information is incorrect. Power-on checks require that at least one diskette drive be installed (bit 0 of the Equipment byte, hex 014, is set to 1).

Bit 4 When set to 1, this bit indicates the power-on check determined that the memory size is not the same as in the configuration record.

Bit 3 When set to 1, this bit indicates that the controller or drive C failed initialization, which prevents the system from attempting a power-on reset.

Bit 2 When set to 0, this bit indicates the time is valid. When set to 1, this bit indicates the time is invalid.

Bit 1 This bit indicates if the installed adapters match the configuration information. When this bit is set to 1, the adapters do not match the configuration information.

Bit 0 When set to 1, this bit indicates a time-out occurred while an adapter ID was being read.

Shutdown Status Byte (Hex 00F): This byte is defined by the power-on diagnostic programs.

Diskette Drive Type Byte (Hex 010): This byte indicates the type of diskette drive installed.

Bit	Function
7 - 4	First Diskette Drive Type
3 - 0	Second Diskette Drive Type

Figure 3-29. Diskette Drive Type Byte

Bits 7 - 4 These bits indicate the first diskette drive type, as shown in the following figure.

Bits 7 6 5 4	Function
0 0 0 0	No drive present
0 0 0 1	Double-sided diskette drive (48 tracks per inch, 360KB)
0 0 1 1	High-capacity diskette drive (720KB)
0 1 0 0	High-density diskette drive (1.44MB)

Note: All combinations that are not shown are reserved.

Figure 3-30. Diskette Drive Type Byte (Bits 7 - 4)

Bits 3 - 0 These bits indicate the second diskette drive type, as shown in the following figure.

Bits 3 2 1 0	Function
0 0 0 0	No drive present
0 0 0 1	Double-sided diskette drive (48 tracks per inch, 360KB)
0 0 1 1	High-capacity diskette drive (720KB)
0 1 0 0	High-density diskette drive (1.44MB)

Note: All combinations that are not shown are reserved.

Figure 3-31. Diskette Drive Type Byte (Bits 3 - 0)

First Fixed Disk Drive Type Byte (Hex 011): This byte defines the type of the first fixed disk drive (drive C). Hex 00 indicates that a fixed disk drive is *not* installed.

Second Fixed Disk Drive Type Byte (Hex 012): This byte defines the type of the second fixed disk drive (drive D). Hex 00 indicates that a fixed disk drive is *not* installed.

Note: For more information about fixed disk drive types, refer to the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference.*

Reserved Byte (Hex 013): This byte is reserved.

Equipment Byte (Hex 014): The equipment byte defines the basic equipment in the system for the power-on diagnostic tests.

Bit	Function
7, 6	Number of Diskette Drives
5, 4	Display Operating Mode
3, 2	Reserved
1	Math Coprocessor Presence
0	Diskette Drive Presence

Figure 3-32. Equipment Byte

Bits 7, 6 These bits indicate the number of diskette drives installed, as shown in the following figure.

Bits 7 6	Number of Diskette Drives
0 0	One Drive
0 1	Two Drives
1 0	Reserved
1 1	Reserved

Figure 3-33. Equipment Byte (Bits 7, 6)

Bits 5, 4 These bits indicate the operating mode of the display attached to the video port, as shown in the following figure.

Bits 5 4	Display Operating Mode
0 0	Reserved
0 1	40-Column Mode
1 0	80-Column Mode
1 1	Monochrome Mode

Figure 3-34. Equipment Byte (Bits 5, 4)

Bits 3, 2 Reserved.

Bit 1 When set to 1, this bit indicates that a math coprocessor is installed.

Bit 0 When set to 1, this bit indicates that a diskette drive is installed.

Low and High Base Memory Bytes (Hex 015 and 016): These bytes define the amount of memory below the 640KB address space.

The value from these bytes represents the number of 1KB blocks of base memory. For example, hex 0280 is equal to 640KB. The low byte is hex 15; the high byte is hex 16.

Low and High Expansion Memory Bytes (Hex 017 and 018): These bytes define the amount of memory above the 1MB address space.

The hexadecimal values in these bytes represent the number of 1KB blocks of expansion memory. For example, hex 0800 is equal to 2048KB. The low byte is hex 17; the high byte is hex 18.

Reserved Bytes (Hex 019 through 031): These bytes are reserved.

Configuration CRC Bytes (Hex 032 and 033): These bytes contain the cyclic-redundancy-check data for bytes hex 010 through hex 031 of the 64-byte CMOS RAM. The low byte is hex 33; the high byte is hex 32.

Reserved Bytes (Hex 034 through 036): These bytes are reserved.

Date Century Byte (Hex 037): Bits 7 through 0 of this byte contain the BCD value for the century. Refer to the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference* for information about reading and setting this byte.

Reserved Bytes (Hex 038 through 03F): These bytes are reserved.

Miscellaneous System Functions

Nonmaskable Interrupt

The nonmaskable interrupt (NMI) signals the system microprocessor that a parity error, a channel check, a system channel time-out, or a system Watchdog time-out has occurred. The NMI stops all arbitration on the bus until bit 6 of the Arbitration register (I/O address hex 0090) is set to 0. This can result in lost data or an overrun error on some I/O devices. The NMI masks all other interrupts and the IRET instruction restores the interrupt flag to the state it was in prior to the interrupt. A system reset causes a reset of the NMI.

Nonmaskable interrupt requests from system board parity and channel check are subject to mask control with the NMI mask bit in the RT/CMOS Address register. The Watchdog Timer and system channel time-out are not masked by this bit. (See "RT/CMOS Address Register and NMI Mask (Hex 0070)" on page 3-25). The power-on default of the NMI mask is 1 (NMI disabled). Prior to enabling the NMI after a power-on reset, the parity check and channel check state are initialized by the POST.

Warning: The operation following a write to hex 0070 should access port hex 0071; otherwise intermittent malfunctions and unreliable operation of the RT/CMOS RAM can occur.

System Control Port B (Hex 0061)

Bit definitions for the read and write functions of this port are shown in the following figures.

Bit	Function
7	Reset Timer 0 Output Latch (IRQ0)
6 - 4	Reserved
3	Enable Channel Check
2	Enable Parity Check
1	Speaker Data Enable
0	Timer 2 Gate to Speaker

Figure 3-35. System Control Port B (Write)

Bit	Function
7	Parity Check
6	Channel Check
5	Timer 2 Output
4	Toggles with Each Refresh Request
3	Enable Channel Check
2	Enable Parity Check
1	Speaker Data Enable
0	Timer 2 Gate to Speaker

Figure 3-36. System Control Port B (Read)

Bit 7 Setting this bit to 1 resets IRQ0. Reading this bit as a 1 indicates a parity check has occurred.

Bit 6 Reading this bit as a 1 indicates a channel check has occurred.

Bit 5 This bit indicates the condition of the timer/counter 2 'output' signal.

Bit 4 This bit toggles for each refresh request.

Bit 3 Setting this bit to 0 enables channel check. This bit is set to 1 during a power-on reset.

Bit 2 Setting this bit to 0 enables parity check. This bit is set to 1 during a power-on reset.

Bit 1 Setting this bit to 1 enables speaker data.

Bit 0 Setting this bit to 1 enables the timer 2 gate.

System Control Port A (Hex 0092)

Bit	Function
7, 6	Fixed-Disk Activity Light
5	Reserved
4	Watchdog Timer Status
3	Security Lock Latch
2	Reserved = 0
1	Alternate Gate A20
0	Alternate Hot Reset

Figure 3-37. System Control Port A

Bits 7, 6 These bits control the fixed-disk activity light. Setting either bit to 1 turns the fixed-disk activity light on. Setting both bits to 0 turns the light off. The power-on reset condition of each bit is 0.

Bit 5 Reserved.

Bit 4 This read-only bit indicates the Watchdog Timer status. When this bit is set to 1, a Watchdog time-out has occurred. For more information about the Watchdog Timer, refer to the *Hardware Interface Technical Reference*.

Bit 3 This bit provides the security lock for the secured area of RT/CMOS. Setting this bit to 1 electrically locks the 8-byte, power-on password. Once this bit is set by POST, it can only be cleared by turning the system off.

Bit 2 Reserved.

Bit 1 This bit is used to enable the 'address 20' signal (A20) when the microprocessor is in the real address mode. When this bit is set to 0, A20 cannot be used in real mode addressing. This bit is set to 0 during a system reset.

Bit 0 This bit provides an alternate method of resetting the system microprocessor. This alternate method supports operating systems requiring faster operation than was provided on the IBM Personal Computer AT®. Resetting the system microprocessor is used to switch the microprocessor from the protected mode to the real address mode. The alternate reset takes 13.4 microseconds.

This bit is set to 0 either by a system reset or a Write operation. When a Write operation changes this bit from 0 to 1, the alternate reset pin is pulsed high for 100 to 125 nanoseconds. The reset occurs after a minimum delay of 6.72 microseconds. While the reset is occurring, the latch remains set so that POST can read this bit. If the bit is 0, POST assumes the system was just powered on. If the bit is 1, POST assumes a switch from the protected mode to the real mode has taken place.

IBM Personal Computer AT is a registered trademark of the International Business Machines Corporation.

Power-On Password

RT/CMOS RAM has 8 bytes reserved for the power-on password and its check character. The 8 bytes are initialized to hex 00. The microprocessor can only access these bytes during power-on self-test (POST). After POST is completed, if a power-on password is installed, the password bytes are locked and cannot be accessed by a program. A power-on password can be from 1 to 7 characters.

During power-on password installation, the password (1 to 7 keyboard scan codes), is stored in the security space.

Power-on password installation is a function of a program contained on the Reference diskette. Once the power-on password utility has been installed, the password can be changed only during the POST. When the new power-on password is installed, changed, or removed, the password is not visible on the display.

The system unit cover can be physically locked to prevent unauthorized access to the battery. This helps prevent unauthorized battery removal and loss of power-on password and configuration information.

For information about the keyboard password, see the "Keyboard and Auxiliary Device Controller" section in the *Hardware Interface Technical Reference.*

Hardware Compatibility

The Model 70 maintain many of the interfaces used by the IBM Personal Computer AT. In most cases command and status organization of these interfaces is maintained.

The functional interfaces for the Model 70 are compatible with the following interfaces:

- The Intel[1] 8259 interrupt controllers (without edge triggering).
- The Intel 8253 timers driven from 1.193 MHz (timer 0 and 2 only).
- The Intel 8237 DMA controller-address/transfer counters, page registers and status fields only. The Command and Request registers are not supported. The rotate and mask functions are not supported. The Mode register is partially supported.
- The NS16450 serial port.
- The Intel 8088, 8086, and 80286 microprocessors.
- The Intel 8272 diskette drive controller.
- The Motorola[2] MC146818 Time of Day Clock command and status (CMOS reorganized).
- The Intel 8042 keyboard port at address hex 0060.
- Display modes supported by the IBM Monochrome Display and Printer Adapter, the IBM Color/Graphics Monitor Adapter, and the IBM Enhanced Graphics Adapter.
- The parallel printer ports (Parallel 1, Parallel 2, and Parallel 3) in compatibility mode.
- Generally compatible with the Intel 80287 and 8087 math coprocessors.

---

1 Intel is a trademark of the Intel Corporation.

2 Motorola is a trademark of Motorola, Inc.

Notes:

Index

A

acoustic readings 1-10
activity light 3-38
adapter setup register 2-10
adapter, fixed disk and diskette drive 3-7
address maps
  memory 3-17
  POS 2-5
  reassignment 3-13
  system 1-6
address 20 gate 3-38
address, split memory 3-16
alternate reset 3-39
arbiter programming 3-5
arbitration control point 3-4
arbitration level 3-4
arbitration register 3-5
arbitration, central control point 3-5

B

base memory bytes 3-35
battery 3-24
block diagram, system board 1-5
block, split memory 3-12
bus attachment, direct 3-7
bus contention 2-3
bus priority assignments 3-4

C

card present bits 2-9
card selected feedback 2-6
-CD SFDBK 2-6
central arbitration control point 3-4
channel 1-4, 3-3
channel check enable 3-37
channel check latch 3-37
CMOS RAM 3-24
configuration CRC bytes 3-35
configuration, RT/CMOS 3-31
connector, drives 3-7
connector, memory 3-22
connectors, fixed disk and diskette drive 3-7

D

date century byte 3-35
description
  math coprocessor 3-3
  memory 3-11
  NMI 3-36
  password 3-40
  programmable option select 2-3
  real-time clock 3-24
  system 1-3
  system board 1-3
diagnostic status byte (RT/CMOS) 3-31
diagram, system board 1-5
direct bus attachment 3-7
diskette drive connector 3-9
diskette drive type byte 3-32
diskette overruns 3-4
DMA channels 3-4

E

enable error checking 3-15
enable NMI 3-25
enable, memory 3-14
equipment byte 3-34
error recovery 3-13
extended arbitration 3-6

F

feedback, card selected 2-6
fixed disk activity light 3-38
fixed disk connector 3-10
fixed disk drive initialization 3-31
fixed disk type byte 3-33

G

gate A20 3-38

I

I/O address map 1-6
identifier, model 1-3
interrupts
  IRQ0 reset 3-37
  NMI reset 3-36

L

list of features 1-4

M

matched memory 3-3
math coprocessor 3-3
maximum current 1-9
measurements 1-9
memory
  connectors 3-22
  description 3-11
  enable bits 3-14
  error recovery 3-13
  performance 1-7
  reassigning 3-13
memory *(continued)*
  refresh 3-4
  split block 3-14
  system memory maps 3-17
memory card definition 2-9
memory encoding registers 3-14
memory size miscompare 3-31
micro channel 1-4, 3-3
microprocessor
  alternate reset 3-39
  mode switch compatibility 3-39
  performance 1-7
  real address mode 3-38
mode switching 3-39
model/submodel bytes 1-3

N

NMI mask 3-25
NMI service 3-4
NMI signal 3-4
noise readings 1-10
nonmaskable interrupt (NMI) 3-36

P

parallel port select 2-8
parity check 3-15
parity check enable 3-37
password, power-on 3-40
performance 1-7, 3-3
ports, system 3-36
POS register 2 2-8
POS register 3 2-9
power specifications 1-9
power supply 1-4
programmable option select (POS) 2-3
  address decode 2-6
  address map 2-5
  registers 2-5
  setup procedures 2-4
  setup register 2-7
  system board 2-6

programmable option select (POS) *(continued)*
system board I/O 2-8
system configuration utilities 2-3
system memory maps 3-17
video 2-7
warning 2-3
programming, arbiter 3-5
protected mode switch 3-39

R

random access memory 3-12
read-only memory (ROM) 3-11
real mode switch 3-39
real-time clock bytes 3-27
refresh 3-4
refresh request 3-37
registers
adapter setup 2-10
card feedback 2-6
memory encoding 3-14
miscellaneous system 3-36
RT/CMOS and NMI mask 3-25
RT/CMOS status registers 3-28
system board POS 2-7
system control 3-36
ROM enable 3-15
RT/CMOS address register 3-25
RT/CMOS RAM 3-24
password 3-40
secured area 3-38

S

second fixed disk type byte 3-33
setup
bus contention 2-3
registers 2-7
system board 2-6
system board I/O 2-8
system memory maps 3-17
video 2-7
shutdown status byte 3-32
size 1-9
size determine, memory 2-9
sound specifications 1-10
speaker data enable 3-37
specifications, system unit 1-9
split address 3-16
split memory block 3-12, 3-14
status registers (RT/CMOS) 3-28
subsystems, memory 3-11
switch to real 3-39
system address map 1-6
system board 1-3
block diagram 1-5
features 1-3
I/O register 2-8
memory connector 3-22
memory registers 3-14
model identifier 1-3
POS registers 2-7
setup 2-6
system configuration utilities 2-3
system control port A 3-38
system control ports 3-36
system description 1-3
system performance 1-7
system ports, miscellaneous 3-36
system reset 3-36
system unit specifications 1-9

T

time status indicator 3-31
type identifier 1-3

V

valid RAM bit 3-30
video setup 2-7

W

warnings
  arbitration 3-4
  central arbiter 3-6
  NMI mask 3-36
  POS 2-3
  real-time clock 3-25
  reserved areas iii
watchdog status 3-38

Numerics

80386 performance 1-7
80387 math coprocessor 3-3



Personal System/2 Model 80 Technical Reference

First Edition (May, 1988)

The following paragraph does not apply to the United Kingdom or any country where such provisions are inconsistent with local law:INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.

It is possible that this publication may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

THE PUBLICATION OF THE INFORMATION CONTAINED HEREIN IS NOT INTENDED TO AND DOES NOT CONVEY ANY RIGHTS OR LICENSES, EXPRESS OR IMPLIED, UNDER ANY IBM PATENTS, COPYRIGHTS, TRADEMARKS, MASK WORKS OR ANY OTHER INTELLECTUAL PROPERTY RIGHTS.

Requests for copies of this publication and for technical information about IBM products should be made to your IBM Authorized Dealer or your IBM Marketing Representative.

Personal System/2 is a registered trademark of the International Business Machines Corporation.

© Copyright International Business Machines Corporation 1988. All rights reserved. No part of this work may be reproduced or distributed in any form or by any means without prior permission in writing from the IBM Corporation.

Preface

The Technical Reference library is intended for those who develop hardware and software products for IBM Personal Computers and IBM Personal System/2. Users should understand computer architecture and programming concepts.

This technical reference provides hardware and software interface information for the IBM Personal System/2 Model 80 and should be used with the following publications:

IBM Personnal System/2 Hardware Interface Technical Reference

IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference

This manual consists of the following sections:

Section 1, "System Overview," describes the system, features, and specifications.

Section 2, "Programmable Option Select," describes registers used for configuration.

Section 3, "System Board," describes the system-specific hardware implementations.

Warning: The term "Reserved" describes certain signals, bits, and registers that should not be changed. Use of reserved areas can cause compatibility problems, loss of data, or permanent damage to the hardware. When the contents of a register are changed, the state of the reserved bits must be preserved. When possible, read the register first and change only the bits that must be changed.

For information about components or devices not described in this manual, refer to the *Hardware Interface Technical Reference.* Information about diskette drives, fixed disk drives, adapters, and external options are in separate option technical references.

Notes:

Contents

Section 1. System Overview 1-1
Description 1-3
System Board Features 1-3
System Board Block Diagram 1-5
System Board I/O Address Map 1-6
Specifications 1-7

Section 2. Programmable Option Select 2-1
Description 2-3
POS Address Map 2-5
Card Selected Feedback 2-6
System Board Setup 2-6
System Board Enable/Setup Register (Hex 0094) 2-7
System Board POS Register 2 (Hex 0102) 2-8
System Board POS Register 3 (Hex 0103)—Type 1 2-9
System Board POS Register 3 (Hex 0103)—Type 2 2-10
Adapter Enable/Setup Register (Hex 0096) 2-12

Section 3. System Board 3-1
Description 3-3
80387 Math Coprocessor 3-3
Micro Channel 3-3
Matched Memory, Type 1 Implementation 3-3
Matched Memory, Type 2 Implementation 3-12
Central Arbiter 3-12
Memory 3-16
Read-Only Memory Subsystem (Type 1) 3-16
Read-Only Memory Subsystem (Type 2) 3-16
Random Access Memory Subsystem 3-17
Error Recovery 3-18
Memory Subsystem Control 3-19
Memory Registers—Type 1 3-20
Memory Registers—Type 2 3-23
System Memory Maps 3-26
System Board Memory Connectors 3-33
Real-Time Clock/Complementary Metal-Oxide Semiconductor RAM 3-36
Miscellaneous System Functions 3-48
Nonmaskable Interrupt 3-48

System Control Port B (Hex 0061) 3-48
System Control Port A (Hex 0092) 3-50
Power-On Password 3-52
Hardware Compatibility 3-53

Index X-1

Figures

1-1. Model and Submodel Bytes 1-3
1-2. System Board Devices and Features 1-4
1-3. System Board 1-5
1-4. System Board I/O Address Map 1-6
1-5. Performance Specifications—Type 1 1-7
1-6. Performance Specifications—Type 2 1-8
1-7. Physical Specifications 1-9
2-1. POS I/O Address Map 2-5
2-2. Card Selected Feedback Register (Hex 0091) 2-6
2-3. System Board Enable/Setup Register (Hex 0094) 2-7
2-4. System Board POS Register 2 (Hex 0102) 2-8
2-5. Parallel Port Select Bits 2-8
2-6. System Board POS Register 3, Type 1 (Hex 0103) 2-9
2-7. POS Register 3 (Bits 3, 2) Type 1 2-10
2-8. POS Register 3 (Bits 1, 0) Type 1 2-10
2-9. POS Register 3, Type 2 2-11
2-10. POS Register 3 (Bits 3, 2), Type 2 2-11
2-11. R1 and T1 Bit Values, Type 2 2-11
2-12. Adapter Enable/Setup Register (Hex 0096) 2-12
3-1. Matched-Memory Extension Voltage and Signal Assignments 3-4
3-2. Matched-Memory Cycle (No Waits) 3-9
3-3. Matched-Memory Cycle with One Wait 3-11
3-4. Arbitration Bus Priority Assignments 3-13
3-5. Arbitration Register, Write to Hex 0090 3-14
3-6. Arbitration Register, Read Hex 0090 3-14
3-7. Memory Error Address Reassignment 3-18
3-8. Memory Encoding Register (Type 1) 3-20
3-9. Memory Encoding Register, Bits 7 and 6 (Type 1) . . . 3-20
3-10. Memory Encoding Register, Bits 5 and 4 (Type 1) . . . 3-21
3-11. Memory Encoding Register, Bits 3, 2, and 1 (Type 1) . 3-22
3-12. Split Address Register (Type 1) 3-22
3-13. Memory Encoding Register 1 (Type 2) 3-23
3-14. Memory Encoding Register 1, Bits 5 and 4 (Type 2) . . 3-23
3-15. Memory Encoding Register 1, Bits 3—1 (Type 2) 3-25
3-16. Memory Encoding Register 2 (Type 2) 3-25
3-17. Memory Encoding Register 2, Bits 5 and 4 (Type 2) . . 3-26
3-18. System Memory Map 1, (Type 1 and Type 2) 3-27
3-19. System Memory Map 2, (Type 1 and Type 2) 3-27

3-20. System Memory Map 3, (Type 1) 3-28
3-21. System Memory Map 3, (Type 2) 3-28
3-22. System Memory Map 4, (Type 1) 3-29
3-23. System Memory Map 4, (Type 2) 3-29
3-24. System Memory Map 5, (Type 1) 3-30
3-25. System Memory Map 5, (Type 2) 3-30
3-26. System Memory Map 6, (Type 1) 3-31
3-27. System Memory Map 6, (Type 2) 3-31
3-28. System Memory Map 7, (Type 2) 3-32
3-29. System Memory Map 8, (Type 2) 3-33
3-30. System Board Memory Connector Pin Locations 3-33
3-31. System Board Memory Connector 3-35
3-32. RT/CMOS RAM Address Map 3-36
3-33. RT/CMOS Address Register and NMI Mask (Hex 0070) 3-37
3-34. RT/CMOS Data Register (Hex 0071) 3-37
3-35. Real-Time Clock Bytes 3-39
3-36. Status Register A 3-40
3-37. Status Register B 3-40
3-38. Status Register C 3-42
3-39. Status Register D 3-42
3-40. Diagnostic Status Byte 3-43
3-41. Diskette Drive Type Byte 3-44
3-42. Diskette Drive Type Byte (Bits 7 - 4) 3-44
3-43. Diskette Drive Type Byte (Bits 3 - 0) 3-44
3-44. Equipment Byte 3-46
3-45. Equipment Byte (Bits 7, 6) 3-46
3-46. Equipment Byte (Bits 5, 4) 3-46
3-47. System Control Port B (Write) 3-49
3-48. System Control Port B (Read) 3-49
3-49. System Control Port A 3-50

Section 1. System Overview

Description 1-3
System Board Features 1-3
System Board Block Diagram 1-5
System Board I/O Address Map 1-6
Specifications 1-7

Notes:

Description

The IBM Personal System/2 Model 80 is a self-contained, floor-standing computer system with a keyboard. It can support two diskette drives and two internal fixed disk drives.

A system can have either a Type 1 or Type 2 system board. The major differences between the system boards are system clock speed, component layout, and memory. Programs identify the type by reading the model and submodel bytes. Interrupt hex 15, function code (AH) = hex C0, returns the model and submodel bytes. The following figure shows these bytes, system board types, and system clock speeds.

Model Byte	Submodel Byte	Revision Code	System Board	System Clock
F8	00	00	Type 1	16 MHz
F8	01	00	Type 2	20 MHz

Figure 1-1. Model and Submodel Bytes

Refer to the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference* for a listing of other systems, and check the supplements section for updates to that listing.

System Board Features

The following figure lists the system board devices and features. The *Hardware Interface Technical Reference* describes devices common to PS/2 products by type number.

Device	Type	Features
Microprocessor	—	80386 32-bit address and 32-bit data interface
System Timers	1	Channel 0 – System timer Channel 2 – Tone generation for speaker Channel 3 – Watchdog timer
ROM Subsystem	—	128KB (KB = 1024 bytes)
RAM Subsystem	—	Type 1 system board, 1 or 2MB (MB = 1,048,576 bytes) Type 2 system board, 2 or 4MB Expandable on the channel
CMOS RAM Subsystem	—	64-byte CMOS RAM with real-time clock/calendar 2KB CMOS RAM extension Battery backup
Video Subsystem	1	Auxiliary connector on the channel Analog output 256KB video memory
Audio Subsystem	1	Driven by: - System-timer channel 2 - The 'audio sum node' signal.
DMA Controller	1	Eight independent DMA channels Single or burst transfers and read verification
Interrupt Controller	1	16 levels of system interrupts Interrupts are level-sensitive
Keyboard/Auxiliary Device Controller	1	Keyboard connector Auxiliary device connector Password security
Diskette Drive Controller	1	Supports: - 720KB formatted diskette density - 1.44MB formatted diskette density.
Serial Controller	1, 2	RS-232C interface Programmable as serial port 1 or 2 FIFO mode* and character mode
Parallel Controller	1	Programmable as parallel port 1, 2, or 3 Supports bidirectional input and output
Micro Channel	—	Eight channel connectors: - Three 32-bit connectors with a matched-memory extension - Five 16-bit connectors: - One with a video extension - One occupied by the fixed disk drive adapter
Math Coprocessor Socket	—	Supports 80387 math coprocessor Same clock speed as the system microprocessor
Power Supply	2	110 and 220 Vac support

* The FIFO mode is only supported by the Type 2 serial controller.

Figure 1-2. System Board Devices and Features

System Board Block Diagram

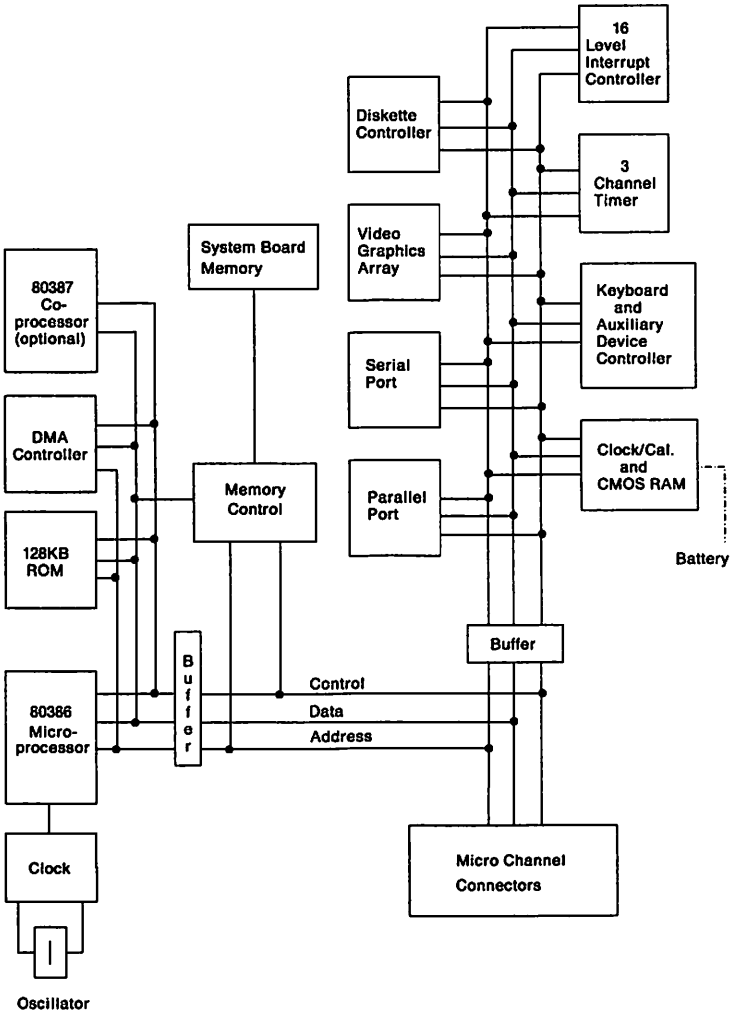


Figure 1-3. System Board

Note: Additional memory can be added in any of the 32-bit Micro Channel connectors. However, the total amount of memory installed should not exceed the 16MB addressing limit of the DMA controller.

System Board I/O Address Map

Hex Addresses	Device
0000 - 001F	DMA Controller (0–3)
0020, 0021	Interrupt Controller (Master)
0040, 0042 - 0044, 0047	System Timers
0060	Keyboard, Auxiliary Device
0061	System Control Port B
0064	Keyboard, Auxiliary Device
0070, 0071	RT/CMOS and NMI Mask
0081 - 0083, 0087	DMA Page Registers (0–3)
0089 - 008B, 008F	DMA Page Registers (4–7)
0090	Central Arbitration Control Point
0091	Card Selected Feedback Register
0092	System Control Port A
0094	System Board Enable/Setup Register
0096	Adapter Enable/Setup Register
00A0 - 00A1	Interrupt Controller (Slave)
00C0 - 00DF	DMA Controller (4–7)
00E0, 00E1	Memory Encoding Registers
00F0 - 00FF	Math Coprocessor
0100 - 0107	Programmable Option Select
01F0 - 01F8	Fixed Disk Drive Controller
0278 - 027B	Parallel Port 3
02F8 - 02FF	Serial Port 2 (RS-232C)
0378 - 037B	Parallel Port 2
03B4, 03B5, 03BA	Video Subsystem
03BC - 03BF	Parallel Port 1
03C0 - 03C5	Video Subsystem
03C6 - 03C9	Video DAC
03CA, 03CC, 03CE, 03CF	Video Subsystem
03D4, 03D5, 03DA	Video Subsystem
03F0 - 03F7	Diskette Drive Controller
03F8 - 03FF	Serial Port 1 (RS-232C)

Figure 1-4. System Board I/O Address Map

Specifications

Device	Number of Waits	Cycle Time (ns)
Microprocessor (16 MHz – 62.5 ns Clock):		
Access to System Board RAM	1	187.5
Access to System Board ROM	1	187.5
Access to Channel:		
Default Transfer Cycle	2	250
Synchronous Extended Transfer Cycle	4	375
Refresh Rate (Typically performed every 15.1 μs)		625 (min)
Bus Master Access to System Board RAM		300 (min)
DMA Controller (8 MHz – 125 ns Clock):		
Single Transfer: 375 + I/O Access + Memory Access		
Burst Transfers: 375 + (I/O Access + Memory Access)N *		
System Board Memory Access		375
Default Transfer Cycle		250
Synchronous Extended Transfer Cycle		375

* N is the number of transfers in the burst.

Figure 1-5. Performance Specifications – Type 1

Device	Number of Waits	Cycle Time (ns)
Microprocessor (20 MHz – 50 ns Clock):		
Access to System Board RAM: *		
Memory Read (Page Hit)	0	100
Memory Read (Page Miss)	2	200
Memory Write (Page Hit)	1	150
Memory Write (Page Miss)	2	200
Access to Channel:		
Default Transfer Cycle	2	200 (min)
Synchronous Extended Transfer Cycle	4	300 (min)
Refresh Rate (Typically performed every 15.1 μs)		500 (min)
Bus Master Access to System Board RAM		300 (min)
DMA Controller (10 MHz – 100 ns Clock):		
Single Transfer: 300 + I/O Access + Memory Access		
Burst Transfers: 300 + (I/O Access + Memory Access)N **		
System Board Memory Access		300
Default Transfer Cycle		200
Synchronous Extended Transfer Cycle		300

* Adapters installed in the channel should not rely on monitoring system board memory accesses because channel memory control signals may not be present during these accesses.
** N is the number of transfers in the burst.

Figure 1-6. Performance Specifications – Type 2

Size:	
Width	165 mm (6.5 in)
Width (feet extended)	318 mm (12.5 in)
Depth	483 mm (19.0 in)
Height	597 mm (23.5 in)
Weight	
(with one fixed disk drive)	20.6 kg (45.3 lb)
Cables:	
Power Cable	1.8 m (6 ft)
Keyboard Cable	3.05 m (10 ft)
Air Temperature:	
System On	15.6 to 32.2° C (60 to 90° F)
System Off	10.0 to 43.0° C (50 to 110° F)
Humidity:	
System On	8% to 80%
System Off	20% to 80%
Maximum Altitude	2133.6 m (7000 ft)
Heat Output	365 W/hr
Acoustical Readings	
from 1 meter (3.28 ft):	
Operating	46 dB average
Idle	40 dB average
Electrical:	
Power Supply Input:	
Low Range	90 (min) - 137 (max) Vac
High Range	180 (min) - 265 (max) Vac
Maximum Current Draw :	
Low Range	5.3 A
High Range	2.7 A
Frequency	50 ± 3Hz/60 ± 3Hz
Maximum Current Allowable:	
Keyboard Port	275 mA
Auxiliary Device Port	300 mA
Electromagnetic Compatibility	FCC Class B

Figure 1-7. Physical Specifications

Notes:

Section 2. Programmable Option Select

Description 2-3
POS Address Map 2-5
Card Selected Feedback 2-6
System Board Setup 2-6
System Board Enable/Setup Register (Hex 0094) 2-7
System Board POS Register 2 (Hex 0102) 2-8
System Board POS Register 3 (Hex 0103)—Type 1 2-9
System Board POS Register 3 (Hex 0103)—Type 2 2-10
Adapter Enable/Setup Register (Hex 0096) 2-12

Notes:

Description

Programmable Option Select (POS) eliminates the need for switches by replacing their function with programmable registers. This section describes the POS information used on the Model 80 system boards. For additional POS information, refer to the *Hardware Interface Technical Reference*.

Warning:

- IBM recommends that programmable options be set only through the System Configuration utilities. Directly setting the POS registers or CMOS RAM POS parameters can result in multiple assignment of the same system resource, improper operation of the feature, loss of data, or possible damage to the hardware.
- Application programs should not use the adapter identification (ID) unless absolutely necessary. Compatibility problems can result.
- If an adapter and the system board are in setup mode at the same time, bus contention will occur, no useful programming can take place, and damage to the hardware can occur.
- After setup operations are complete, the Adapter Enable/Setup register (hex 0096) should be set to hex 00, and the System Board Enable/Setup register (hex 0094) should be set to hex FF.
- The channel reset bit (bit 7) in the Adapter Enable/Setup register must be 0 to program the adapters.
- The system board does not support 16-bit I/O operations to 8-bit POS registers. Using 16-bit I/O instructions on 8-bit POS registers will cause erroneous data to be written to or read from the registers. Only 8-bit transfers are supported for setup operations.

Setup functions respond to I/O addresses hex 0100 through 0107 only when their unique setup signal is active. The following precautions must be taken before setting individual bits in the POS registers.

System Board Video Subsystem Setup:

- Bit 5 in the System Board Enable/Setup register (hex 0094) must be set to 0 to place the system board video into the setup mode.
- Bit 3 in the Adapter Enable/Setup register (hex 0096) must be set to 0 to avoid driving a 'setup' signal to an adapter.
- Bit 7 in the System Board Enable/Setup register must be set to 1 to avoid driving a 'setup' signal to other system board functions.

Adapter Setup:

- Bit 3 in the Adapter Enable/Setup register must be set to 1 to allow adapter setup.
- Bit 5 in the System Board Enable/Setup register must be set to 1 to avoid driving a 'setup' signal to the Video Subsystem.
- Bit 7 in the System Board Enable/Setup register must be set to 1 to avoid driving a 'setup' signal to a system board function.

Other System Board Setup:

- Bit 7 in the System Board Enable/Setup register must be set to 0 to allow setup of other system board functions.
- Bit 3 in the Adapter Enable/Setup register must be set to 0 to avoid driving a 'setup' signal to an adapter.
- Bit 5 in the System Board Enable/Setup register must be set to 1 to avoid driving a 'setup' signal to the Video Subsystem.

POS Address Map

The following figure shows the organization of the I/O address space used by POS. Bit 0 of POS Register 2 and bits 6 and 7 of POS Register 5 are fixed. All other bits in POS Registers 2 through are free-form.

Address (Hex)	Function
0094	System Board Enable/Setup Register
0096	Adapter Enable/Setup Register
0100	POS Register 0—Adapter Identification Byte (Low Byte)
0101	POS Register 1—Adapter Identification Byte (High Byte)
0102	POS Register 2—Option Select Data Byte 1
	Bit 0 is Card Enable.
0103	POS Register 3—Option Select Data Byte 2
0104	POS Register 4—Option Select Data Byte 3
0105	POS Register 5—Option Select Data Byte 4
	Bit 7 is the channel check active indicator
	Bit 6 is the channel check status-available indicator
0106	POS Register 6—Subaddress Extension (Low Byte)
0107	POS Register 7—Subaddress Extension (High Byte)

Figure 2-1. POS I/O Address Map

Card Selected Feedback

When an adapter is addressed, it responds by setting the '-card selected feedback' signal (-CD SFDBK) to active. -CD SFDBK is derived from the address decode and driven by a totem pole driver. It is latched by the system board and can be read through the Card Selected Feedback register at address hex 0091. Diagnostic and automatic configuration programs use this signal to verify the operation of an adapter at a given address or DMA port. This signal must not be active during a setup cycle.

The Card Selected Feedback register is a read-only register at address hex 0091. It allows programs to monitor -CD SFDBK and thereby determine if the video subsystem, system board I/O, or an adapter is addressed and functioning.

Bit	Function
7 - 1	Reserved
0	-Card Selected Feedback

Figure 2-2. Card Selected Feedback Register (Hex 0091)

Bits 7 - 1 Reserved.

Bit 0 This bit is set to 1 whenever -CD SFDBK was active on a previous cycle or whenever the system board I/O functions (diskette drive, serial, or parallel interfaces) are accessed by an I/O cycle. Reading this register resets the bit to 0.

System Board Setup

The integrated I/O functions on the system board use POS information during setup. The diskette drive controller, serial port, and parallel port are treated as a single device. The video subsystem is also an integrated part of the system board, however, POS treats it as a separate device. The System Board Enable/Setup register is used to place the system board or the video subsystem into the setup mode.

System Board Enable/Setup Register (Hex 0094)

This is a read/write register; all bits in this register default to 1 (enabled).

Bit	Function
7	Enable/-Setup System Board Functions
6	Reserved
5	Enable/-Setup Video Subsystem
4 - 0	Reserved

Figure 2-3. System Board Enable/Setup Register (Hex 0094)

Bit 7 When this bit is set to 0, various system board I/O functions are placed in the setup mode. The diskette drive controller, serial port, and parallel port are controlled through System Board POS Register 2 (hex 0102). The POS information for memory is in System Board POS Register 3 (hex 0103).

When this bit is set to 1, the system board functions are enabled.

Bit 6 Reserved.

Bit 5 When this bit is set to 0, the video subsystem is placed in the setup mode and controlled through POS Register 2 (hex 0102). When set to 1, and bit 0 in hex 0102 is set to 1, video is enabled.

Bit 0 of POS Register 2 is the video enable bit. When this bit is set to 0, the video subsystem does not respond to commands, addresses, or data. If video is being generated when the video enable bit is set to 0, the output is still generated. For information on BIOS calls to enable or disable the video, see the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference.*

Note: When video is disabled, access to the video DAC registers is disabled.

Bits 4 - 0 Reserved.

System Board POS Register 2 (Hex 0102)

When the system board is in the setup mode, the diskette drive controller, serial port, and parallel port are controlled by this read/write register. Reading this register returns the current state of these system board functions.

Bit	Function
7	Disable Parallel Port Extended Mode
6, 5	Parallel Port Select
4	Enable Parallel Port
3	Serial Port Select
2	Enable Serial Port
1	Enable Diskette Drive Interface
0	Enable System Board

Figure 2-4. System Board POS Register 2 (Hex 0102)

Bit 7 When set to 0, this bit allows the parallel port to be configured as an 8-bit, parallel, bidirectional interface. When set to 1, this bit disables the bidirectional mode. This bit is set to 0 at power-on and POST sets it to 1.

Bits 6, 5 These bits select the configuration of the system board parallel port.

Bits 6 5	Assignment	Hex Address	Interrupt Level
0 0	Parallel 1	03BC - 03BF	7
0 1	Parallel 2	0378 - 037B	7
1 0	Parallel 3	0278 - 027B	7
1 1	Reserved	----------	-

Figure 2-5. Parallel Port Select Bits

Bit 4 When this bit and bit 0 are set to 1, the system board parallel port is enabled.

Bit 3 When set to 1, this bit sets the system board serial port as Serial 1 (addresses hex 03F8 through 03FF), which uses interrupt level 4. When set to 0, this bit sets the serial port as Serial 2 (addresses hex 02F8 through 02FF), which uses interrupt level 3.

Bit 2 When this bit and bit 0 are set to 1, the system board serial port is enabled.

Bit 1 When this bit and bit 0 are set to 1, the diskette drive interface is enabled.

Bit 0 When set to 1, this bit allows bits 4, 2, and 1 to enable and disable their respective devices. When set to 0, this bit disables the diskette drive interface, system board serial port, and system board parallel port, regardless of the state of bits 4, 2, and 1.

System Board POS Register 3 (Hex 0103) – Type 1

This read-only register is accessed while the system board is in the setup mode. It contains information about the presence and type of memory card installed in each system board connector. Connector 1 is located closest to the power supply; connector 2 is located furthest from the power supply.

Note: The Type 1 system board supports one or two 1MB memory cards.[1]

Bit	Function
7 - 4	Reserved
3	-1MB Card, Connector 2
2	-Card Present, Connector 2
1	-1MB Card, Connector 1
0	-Card Present, Connector 1

Figure 2-6. System Board POS Register 3, Type 1 (Hex 0103)

Bits 7 - 4 Reserved.

Bits 3, 2 These bits indicate if memory is installed in memory connector 2 as shown in the following figure.

[1] 80386 System Board Memory Expansion Kit.

Bits 3 2	Available RAM
0 0	1MB Memory Card in Connector 2
0 1	Reserved
1 0	Reserved
1 1	No Memory Card Present in Connector 2

Figure 2-7. POS Register 3 (Bits 3, 2) Type 1

Bits 1, 0 These bits indicate if memory is installed in memory connector 1 as shown in the following figure.

Bits 1 0	Available RAM
0 0	1MB Card in Connector 1
0 1	Reserved
1 0	Reserved
1 1	No Card Present in Connector 1

Figure 2-8. POS Register 3 (Bits 1, 0) Type 1

Additional registers at address hex 00E0 and 00E1 are used during setup to define how memory is used. See "Random Access Memory Subsystem" on page 3-17 for more information.

System Board POS Register 3 (Hex 0103) – Type 2

This read-only register is accessed while the system board is in the setup mode. It contains information about the presence and type of memory card installed in each system board connector. Connector 1 is located closest to the power supply; connector 2 is located furthest from the power supply.

Note: The Type 2 system board supports one or two 2MB memory cards.[2]

[2] 80386 System Board 2MB Memory Expansion Kit

Bit	Function
7 - 4	Reserved
3	2MB Card, Connector 2 (T Signal)
2	-Card Present, Connector 2 (R Signal)
1	2MB Card, Connector 1 (T Signal)
0	-Card Present, Connector 1 (R Signal)

Figure 2-9. POS Register 3, Type 2

Bits 7, 4 These bits are reserved.

Bits 3, 2 These bits indicate if memory is installed in memory connector 2 as shown in the following figure.

Bits 3 2	Available RAM
0 0	Reserved
0 1	Reserved
1 0	2MB Card in Connector 2
1 1	No Card in Connector 2

Figure 2-10. POS Register 3 (Bits 3, 2), Type 2

Bits 1, 0 These bits indicate if memory is installed in memory connector 1 as shown in the following figure.

Bits 1 0	Available RAM
0 0	Reserved
0 1	Reserved
1 0	2MB Card in Connector 1
1 1	Invalid (At Least 1MB Must Be Available in Connector 1)

Figure 2-11. R1 and T1 Bit Values, Type 2

Additional registers at address hex 00E0 and 00E1 are used during setup to define how memory is used. See "Random Access Memory Subsystem" on page 3-17 for more information.

Adapter Enable/Setup Register (Hex 0096)

The Adapter Enable/Setup register selects the connector to be setup.

Bit	Symbol
7	Channel Reset
6 - 4	Reserved
3	Card Setup Enable
2 - 0	Channel Select 2 - 0

Figure 2-12. Adapter Enable/Setup Register (Hex 0096)

Bit 7 When set to 1, this bit activates the 'channel reset' signal to all connectors.

Bits 6 - 4 These bits are reserved.

Bit 3 When set to 1, this bit enables the '-card setup' signal selected by bits 2 through 0.

Bits 2 - 0 These bits are the address bits for the '-card setup' signal. Connectors 1 through 8 are addressed as 0 through 7, respectively. When bit 3 is set to 1, these bits select the connector that is put into setup.

Each channel connector has a unique '-card setup' signal (-CD SETUP) associated with it. This signal is used to put the adapters in the setup mode, which allows access to the POS registers. The individual connectors are selected through the Adapter Enable/Setup register. Setup information is then read from or written to the selected adapter through I/O addresses hex 0100 through 0107.

Notes:

1. -CD SETUP only goes active when an operation is performed in the I/O address range hex 0100 through 0107.
2. The status of port hex 0096 can be read by software. However, when the port is read, bits 6, 5, and 4 are set to 1.

Section 3. System Board

Description 3-3
80387 Math Coprocessor 3-3
Micro Channel 3-3
Matched Memory, Type 1 Implementation 3-3
Matched-Memory Extension 3-4
Signal Descriptions 3-4
Matched-Memory Cycle 3-6
Matched-Memory Cycle Timing (No Waits) 3-8
Matched-Memory Cycle Timing (One Wait) 3-10
Matched Memory, Type 2 Implementation 3-12
Central Arbiter 3-12
Arbitration Bus Priority Assignments 3-12
Central Arbiter Programming 3-14
Memory 3-16
Read-Only Memory Subsystem (Type 1) 3-16
Read-Only Memory Subsystem (Type 2) 3-16
Random Access Memory Subsystem 3-17
Error Recovery 3-18
Memory Subsystem Control 3-19
Memory Registers—Type 1 3-20
Memory Encoding Register — Type 1 (Hex 00E1) 3-20
Split Address Register — Type 1 (Hex 00E0) 3-22
Memory Registers—Type 2 3-23
Memory Encoding Register 1 — Type 2 (Hex 00E1) 3-23
Memory Encoding Register 2 — Type 2 (Hex 00E0) 3-25
System Memory Maps 3-26
System Board Memory Connectors 3-33
Real-Time Clock/Complementary Metal-Oxide Semiconductor RAM 3-36
RT/CMOS Address Register and NMI Mask (Hex 0070) . . 3-37
RT/CMOS Data Register (Hex 0071) 3-37
RT/CMOS RAM I/O Operations 3-38
Real-Time Clock Bytes (Hex 000-00D) 3-39
CMOS RAM Configuration 3-43
Miscellaneous System Functions 3-48
Nonmaskable Interrupt 3-48
System Control Port B (Hex 0061) 3-48
System Control Port A (Hex 0092) 3-50
Power-On Password 3-52

Hardware Compatibility 3-53

Description

This section describes the Model 80 implementation of the 80387 math coprocessor, Micro Channel architecture, memory subsystem, and miscellaneous system ports. For additional information about these and other areas not discussed, refer to the *Hardware Interface Technical Reference.*

80387 Math Coprocessor

The 80387 math coprocessors are matched to the speed of the system microprocessor and operate in the synchronous mode. The 16 MHz and 20 MHz math coprocessors are not interchangeable; their logical operation is the same.

Micro Channel

The Type 1 and Type 2 system boards implement the Micro Channel architecture the same way, with the exception of matched memory.

Matched Memory, Type 1 Implementation

The 32-bit channel connector with matched-memory extension is a dual 93-pin, 50-mil card edge design. The matched-memory extension provides three signal lines, two ground lines, and three reserved lines.

Matched-Memory Extension

This extension to the 32-bit Micro Channel connector provides the signals necessary to accommodate matched-memory cycles.

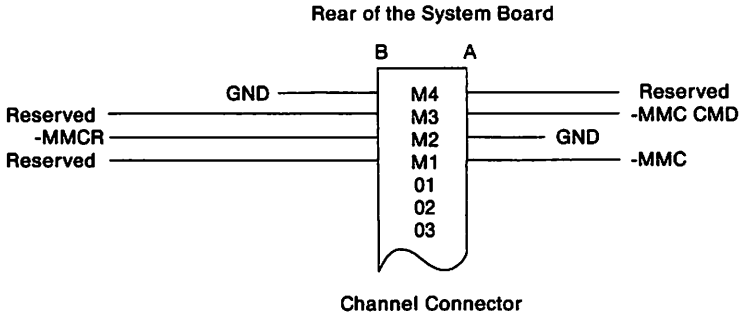


Figure 3-1. Matched-Memory Extension Voltage and Signal Assignments

Signal Descriptions

The following are descriptions of the matched memory signals.

-MMC: -Matched Memory Cycle: This signal is driven by the system board logic to indicate to the channel slaves that the system microprocessor is the controlling master and is able to run a matched-memory cycle. -MMC is driven by a tri-state driver that can sink 24 milliamps.

-MMCR: -Matched Memory Cycle Request: This is a bus cycle control input signal. A 16- or 32-bit channel slave drives -MMCR to request the faster cycle available on the system bus. This signal is driven active from an address decode and M/-IO.

If -MMCR is driven active by an 8-bit channel slave, or if it is driven active by a 16- or 32-bit channel slave during a cycle other than a microprocessor bus cycle, a default transfer cycle is run and -MMCR is not honored. CD CHRDY is used to extend the matched-memory cycle, as needed. -ADL and -CMD remain inactive for the entire matched-memory cycle.

Only the system microprocessor can drive matched-memory cycles. This signal is wired separately to each of the 32-bit channel connectors on the system board and ORed by the system logic. This signal is driven by a totem-pole driver that can sink 6 milliamps.

-MMC CMD: -Matched Memory Cycle Command: This output signal to the bus and is generated for system microprocessor bus cycles only. It is used during a matched-memory cycle to define when the data on the bus is valid. The trailing edge of this signal indicates the end of the matched-memory cycle. As with -CMD,, this signal indicates the time that the data is valid on the bus, and indicates when to either latch the data from the bus or stop driving the data onto the bus. -MMC CMD is driven by a tri-state driver that can sink 24 milliamps.

Matched-Memory Cycle

This section describes the Type 1 matched-memory bus operation supported by the Micro Channel. The Type 1 implementation is such that a channel slave can be either a memory or I/O slave with a data bus width of 16 or 32 bits. No 8-bit devices are allowed to run matched-memory cycles.

In addition to the three basic types of cycles defined by the Micro Channel (default, synchronous extended, and asynchronous extended), the Type 1 system board also supports the matched-memory cycle. The matched-memory cycle is a system-unique function supported by the Micro Channel, and is provided for memory subsystem expansion. This function allows the most efficient data transfer capability between the 16 MHz 80386 system microprocessor and channel memory. The system board ROM and RAM (held on the local bus) and the IBM 80386 Memory Expansion Adapter (held on the channel) adhere to the matched-memory cycle protocol described in this section.

Matched-memory cycles are a minimum of 3 (16 MHz) clocks or a cycle time of 187.5 nanosecond. (Default transfer cycles are a minimum of 4 clocks or a cycle time of 250 nanoseconds.) The target channel slave must request the matched-memory cycle on a cycle-by-cycle basis through -MMCR, or it will get the default transfer cycle.

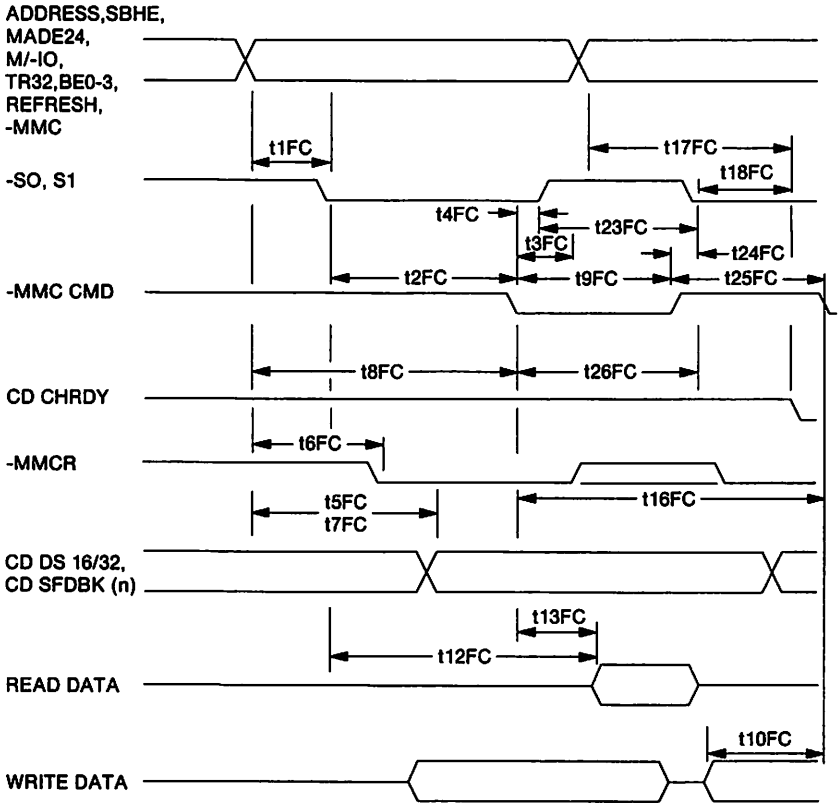
A system microprocessor bus cycle begins by the 'address', 'byte enables', and bus definition signals becoming valid in the first clock cycle of the bus cycle. Either of the status bits driven active signals the start of the channel bus cycle. The status state (Ts) is the first clock cycle. After Ts, the command state (Tc) is entered. Channel slaves respond to the bus operation during Tc, either transferring read data to or accepting write data from the system bus. Tc states can be repeated as often as necessary to assure sufficient time for the channel slave to respond. CD CHRDY is used to determine whether Tc is repeated. A repeated Tc state is called a 62.5-nanosecond wait state.

If -MMCR is driven active by the channel slave, then -MMC CMD is driven by the system microprocessor.

If -MMCR is not returned from the channel slave, a default cycle is run. Default transfer and matched-memory bus cycles can be mixed in any manner because the mechanism is completely dynamic on a bus cycle basis. Bus cycles can be extended until CD CHRDY is sampled as active.

CD CHRDY is driven inactive by a slave to allow additional time to complete a default or matched-memory cycle when the normal cycle length is not sufficient to service that device. The timings for the matched-memory cycles are shown on the following pages.

Matched-Memory Cycle Timing (No Waits)



Symbol	Description	Min/Max
t1 FC	ADDR valid to Status active	10 / - ns
t2 FC	Status valid to -MMC CMD active	82 / - ns
t3 FC	ADDR hold from -MMC CMD active	20 / - ns
t4 FC	Status hold from -MMC CMD active	25 / - ns
t5 FC	CD DS 16/32 active from ADDR valid	- / 55 ns
t6 FC	-MMCR active from ADDR valid	- / 55 ns
t7 FC	-CD SFDBK active from ADDR valid	- / 60 ns
t8 FC	ADDR valid to -MMC CMD active	100 / - ns
t9 FC	-MMC CMD pulse width	85 / - ns
t10 FC	Write data valid to -MMC CMD active	0 / - ns
t11 FC	Write data hold from -MMC CMD inactive	30 / - ns
t12 FC	Read data valid from Status active	- /145 ns
t13 FC	Read data valid from -MMC CMD active	- / 60 ns
t14 FC	Read data hold from -MMC CMD inactive	0 / - ns
t15 FC	Read data off delay from -MMC CMD inactive	- / 40 ns
t16 FC	-MMC CMD active to next -MMC CMD active	180 / - ns
t17 FC	CD CHRDY valid from ADDR valid	- / 70 ns
t18 FC	CD CHRDY valid from Status active	- / 30 ns
t23 FC	Status inactive pulse width	30 / - ns
t24 FC	Next Status active to -MMC CMD inactive	- / 5 ns
t25 FC	-MMC CMD inactive pulse width	85 / - ns
t26 FC	-MMC CMD active to Status active	90 / - ns

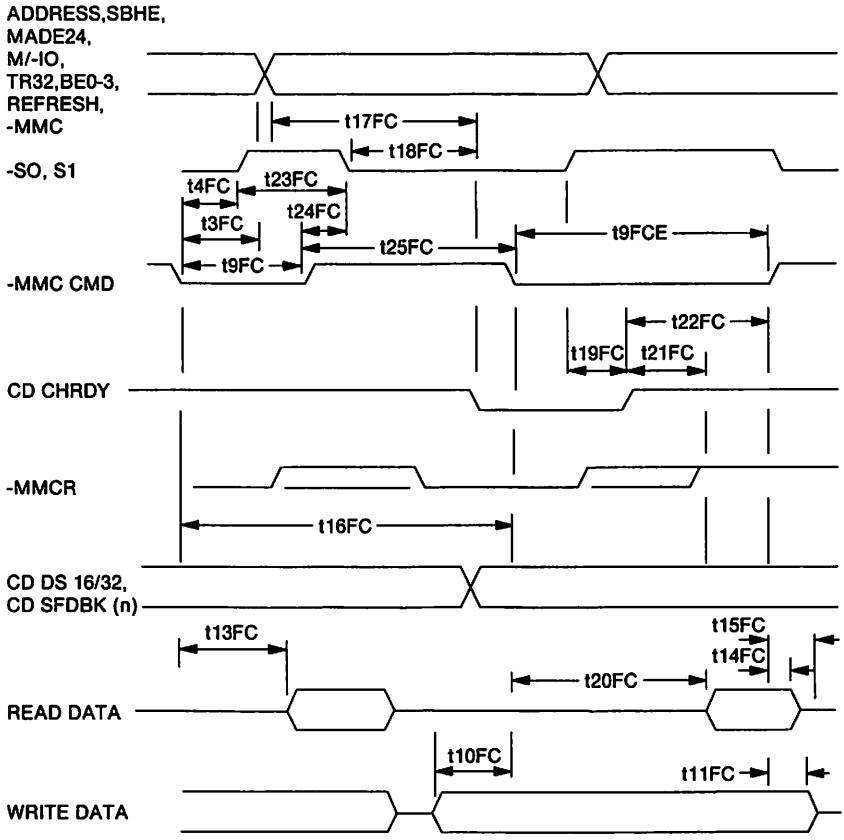
Figure 3-2. Matched-Memory Cycle (No Waits)

Matched-memory cycles are supported for 32-bit and 16-bit channel devices only. Eight-bit channel devices cannot run matched-memory cycles.

Note: Adapters using this protocol can only be installed in the three 32-bit channel connectors and therefore sacrifice some compatibility. These adapters may not fit or operate in some other systems.

When -MMC is active, 32-bit and 16-bit devices should not use -MADE 24, A0, A1, or -SBHE in logic that generates -MMCR, -CD DS 16/32, -CD SFDBK, and CD CHRDY.

Matched-Memory Cycle Timing (One Wait)



Symbol	Description	Min/Max
t3 FC	ADDR hold from -MMC CMD active	20 / - ns
t4 FC	Status hold from -MMC CMD active	25 / - ns
t9 FC	-MMC CMD pulse width	85 / - ns
t9 FCE	-MMC CMD pulse width (Extended Cycle)	145 / - ns
t10 FC	Write data valid to -MMC CMD active	0 / - ns
t11 FC	Write data hold from -MMC CMD inactive	30 / - ns
t13 FC	Read data valid from -MMC CMD active	- / 60 ns
t14 FC	Read data hold from -MMC CMD inactive	0 / - ns
t15 FC	Read data off delay from -MMC CMD inactive	- / 40 ns
t16 FC	-MMC CMD active to next -MMC CMD active	240 / - ns
t17 FC	CD CHRDY valid from ADDR valid	- / 70 ns
t18 FC	CD CHRDY valid from Status active	- / 30 ns
t19 FC	CD CHRDY active from Status inactive	- / 9 ns
t20 FC	Read data valid from -MMC CMD active	- /125 ns
t21 FC	Read data valid from CD CHRDY active	- / 45 ns
t22 FC	-MMC CMD inactive from CD CHRDY active	65 / - ns
t23 FC	Status inactive pulse width	30 / -
t24 FC	Next Status active to -MMC CMD inactive	- / 5 ns
t25 FC	-MMC CMD inactive pulse width	85 / - ns

Figure 3-3. Matched-Memory Cycle with One Wait

Matched-memory cycles are supported for 32-bit and 16-bit channel devices only. Eight-bit channel devices cannot run matched-memory cycles.

Note: Adapters using this protocol can only be installed in the three 32-bit channel connectors and therefore sacrifice some *portability*. These adapters may not fit or operate in some other systems.

When -MMC is active, 32-bit and 16-bit devices should not use -MADE 24, A0, A1, or -SBHE in logic that generates -MMCR, -CD DS 16/32, -CD SFDBK, and CD CHRDY.

Matched Memory, Type 2 Implementation

The channel connectors are physically identical to those on the Type 1 system board including the presence of the matched-memory extensions on the 32-bit connectors. The system microprocessor drives -MMC active when it does a memory access. This allows the Type 2 system board to support the same memory adapter design as the Type 1 system board.

Although a memory adapter can respond to -MMC active by driving -MMCR active, the Type 2 system board ignores -MMCR and performs a default cycle if the 'channel ready' signal (CD CHRDY) is driven active, or a synchronous-extended cycle if CD CHRDY is inactive. The Type 2 system board also drives -MMC CMD active at the same time as -CMD.

Central Arbiter

The central arbitration control point gives intelligent subsystems on the channel the ability to share and control the system. It allows burst data transfers and prioritization of control between devices. This arbiter supports up to 16 arbitrating devices.

Arbitration Bus Priority Assignments

The following figure shows the assignment of arbitration levels. The functions with the lowest arbitration level have the highest priority.

ARB Level	Primary Assignment
-2	Memory Refresh
-1	NMI
0	DMA Channel 0 (Programmable to any arbitration level)
1	DMA Channel 1
2	DMA Channel 2
3	DMA Channel 3
4	DMA Channel 4 (Programmable to any arbitration level)
5	DMA Channel 5
6	DMA Channel 6
7	DMA Channel 7
8 - E	Available
F	System Microprocessor

Figure 3-4. Arbitration Bus Priority Assignments

Note: Devices designed for arbitration level 0 or 1 should have limited bandwidth or short bursts so diskette overruns can be prevented or recovered by retry operations. The diskette drive controller, on arbitration level 2, can be held inactive by devices on levels 0 and 1, by a refresh operation, and by the previous controlling master. The diskette drive controller should not be held inactive for more than 12 microseconds to prevent overrun.

NMI service is executed at a priority level higher than 0, called -1. Memory refresh is prioritized at -2, two levels higher than 0. Levels -1 and -2 are reached on the system board only, while the 'arbitrate/-grant' signal is in the arbitrate state.

When the central arbitration control point receives a level -1 request (NMI, a system-board internal signal), it activates -PREEMPT, waits for the end of transfer, and then places ARB/-GNT in the arbitrate state, which denies channel activity to arbitrating devices. The central arbitration control point gives the grant to the level -1 request, and holds ARB/-GNT in the arbitrate state until the operation is complete and the NMI is reset.

Central Arbiter Programming

The central arbitration control point provides access to programmable options through the Arbitration register, which is accessed at I/O address hex 0090. The bits are defined differently for read and write operations, as shown in the following figures.

Bit	Definition
7	Enable System Microprocessor Cycle
6	Arbitration Mask
5	Enable Extended Arbitration
4 - 0	Reserved

Figure 3-5. Arbitration Register, Write to Hex 0090

Bit	Definition
7	Enable System Microprocessor Cycle
6	Arbitration Masked by NMI
5	Bus Time-out
4	Reserved
3 - 0	Value of Arbitration Bus During Previous Grant State

Figure 3-6. Arbitration Register, Read Hex 0090

Bit 7 Setting this bit to 1 enables system microprocessor cycles during arbitration cycles. This bit can be set to 0 if an arbitrating device requires total control of the channel bandwidth. This bit is set to 0 by a system reset.

Reading this bit as a 1 indicates system microprocessor cycles are enabled during arbitration.

Bit 6 Setting this bit to 1 causes the central arbitration control point to enter the arbitration state. The system microprocessor controls the channel until this bit is reset to 0. This bit is set to 0 by a system reset.

Reading this bit as a 1 indicates that an NMI has occurred and has masked arbitration.

Warning: This bit should be set to 1 only by diagnostic routines and system error-recovery routines.

Bit 5 Setting this bit to 1 enables extended arbitration. The minimum arbitration cycle is 300 nanoseconds; this bit extends that minimum cycle to 600 nanoseconds. This bit is set to 0 during a system reset.

Reading this bit as a 1 indicates that a bus time-out has occurred, and resets bit 6 in this register to 0.

Bit 4 This bit is reserved and should be 0.

Bits 3 - 0 These bits are undefined for a write operation and should be set to 0.

Reading these bits returns the arbitration level of the arbiter controlling the channel during the most recent grant state. This information allows the system microprocessor to determine the arbitration level of the device that caused a bus time-out.

Memory

The Model 80 systems use the following types of memory:

- Read-only memory (ROM)
- Random access memory (RAM)
- Real-time Clock and CMOS RAM.

Read-Only Memory Subsystem (Type 1)

The ROM subsystem on the Type 1 system board consists of four 32Kb by 8-bit modules in a 32Kb by 32-bit arrangement (Kb equals 1024 bits). ROM is assigned at the top of the first and last 1MB address space (000E0000 and FFFE0000). ROM is not parity-checked. It operates with one 62.5-nanosecond wait state.

Read-Only Memory Subsystem (Type 2)

The ROM subsystem on the Type 2 system board consists of two 64Kb by 8-bit modules in a 64Kb by 16-bit arrangement. ROM is active at power-on at the top of the first and last 1MB of address space (000E0000 and FFFE0000). After POST ensures the proper operation of system memory, the ROM code is copied to RAM at the same address space, and ROM is disabled.

ROM or RAM access at address space hex 000E0000 to 000FFFFF is controlled by the ROMEN bit in Memory Encoding Register 1 (hex 00E1). When enabled, ROM is not parity-checked and operates with four 50-nanosecond wait states.

Random Access Memory Subsystem

The RAM subsystem on the system board consists of memory modules mounted on a printed-circuit board, which attaches to the system board through a 3- by 32-pin connector. Two connectors are provided on the system board. The Type 1 system board supports the 1MB memory cards; the Type 2 system board supports the 2MB memory cards. Type 2 system board POST routines require a minimum of 1MB of functional memory installed in connector 1. The Type 1 system board also requires a minimum of 1MB of functional memory, however it can be installed in either connector 1 or connector 2. The RAM subsystem is 36 bits wide: 32 data bits and 4 parity bits. One parity bit is generated for each byte of data written. During a read operation, one parity bit is checked for each each byte of data read by the device controlling the bus.

Note: Additional memory can be added in any of the 32-bit Micro Channel connectors. The total amount of memory installed should not exceed the 16MB addressing limit of the DMA controller.

Error Recovery

If POST detects a memory error in the first 512KB of memory, the first physical 1MB block of memory is deactivated. The addresses assigned to the deactivated block are reassigned to the second physical block of system board memory, if installed. In the event an error is also detected in the first 512KB of the second 1MB block of system board memory, the system cannot recover. The first 512KB of memory address space cannot be assigned to reside on an adapter installed in the channel.

POST does not deactivate a memory block if an error is detected in the second 512KB of that block. If an error is detected in that area of memory, the 1MB block containing the error can be deactivated and have its addresses reassigned by running the customer-level memory diagnostics program on the Reference Diskette supplied with the system. Once a block of memory is deactivated, it is ignored by POST on subsequent power-ons.

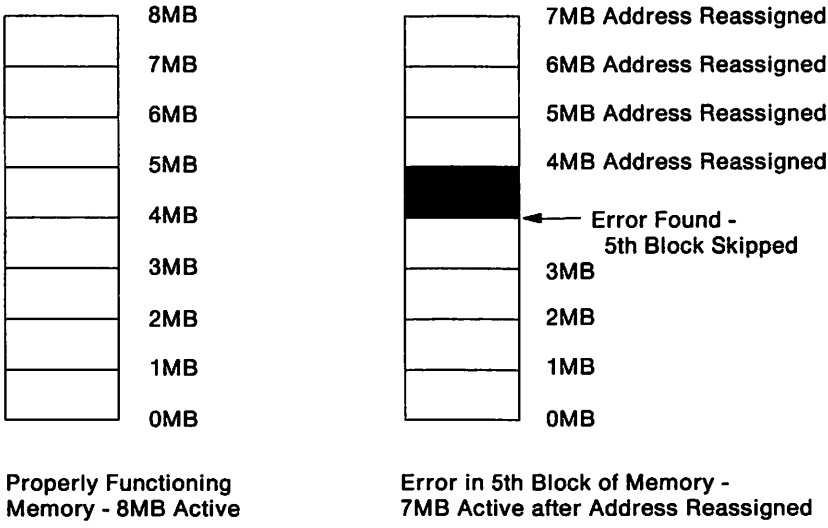


Figure 3-7. Memory Error Address Reassignment

Memory Subsystem Control

The Type 1 and Type 2 system boards enable memory in 1MB blocks. Each block must start on a 1MB boundary. Because 128KB of system board ROM, 128KB of I/O ROM, and 128KB of video RAM are mapped within the first 1MB address space, the first physical 1MB of memory cannot be mapped to contiguous addresses and is therefore split at either the 512KB or 640KB boundary. The first 512KB or 640KB is addressed beginning at hex 00000000; the remaining unmapped memory, called the *split-memory block*, can be mapped in several ways:

- Type 1 system board – The entire split-memory block is mapped to the first available address following the last full 1MB block of activated memory. Alternately, the split-memory block can be handled in either of the following ways:
  - It can be totally disabled.
  - 128KB of the split-memory block can be mapped to the system board ROM address space, replacing the system board ROM. The remainder of the split-memory block is disabled.
- Type 2 system board – In addition to mapping the first 512KB or 640KB of physical RAM, the Type 2 system board automatically maps 128KB of the first 1MB of RAM to the system board ROM address space. The ROM code is then copied into RAM at this location. The remaining unmapped memory, the *split-memory block*, can be either disabled or mapped to the first available address following the last full 1MB block of activated memory .

The size of the split-memory block, memory mapping, and disabling of memory blocks are controlled by registers residing at addresses hex 00E0 and 00E1. The Type 1 and Type 2 system boards define these registers differently.

Memory Registers – Type 1

Two registers define how the system board memory is addressed and used:

- Memory Encoding Register (hex 00E1)
- Split Address Register (hex 00E0).

Memory Encoding Register – Type 1 (Hex 00E1)

This read/write register determines the amount of enabled system board memory and how it is used.

Bit	Function
7	EN4
6	EN3
5	EN2
4	EN1
3	-ENSPLIT
2	-640
1	ROMEN
0	-ENPLRPCH

Figure 3-8. Memory Encoding Register (Type 1)

Bits 7, 6 These bits define the amount of enabled memory in system board memory connector 2.

Bits 7 6	Amount of RAM	Function
0 0	Reserved	
0 1	Reserved	
1 0	1MB	1MB Card enabled in Connector 2
1 1	0MB	Card Disabled or No Card Present in Connector 2

Figure 3-9. Memory Encoding Register, Bits 7 and 6 (Type 1)

Bits 5, 4 These bits define the amount of enabled memory in system board memory connector 1.

Bits 5 4	Amount of RAM	Function
0 0	Reserved	
0 1	Reserved	
1 0	1MB	1MB Card enabled in Connector 1
1 1	0MB	Card Disabled or No Card Present in Connector 1

Figure 3-10. Memory Encoding Register, Bits 5 and 4 (Type 1)

Bit 3 This bit determines whether the split-memory block is assigned addresses or disabled.

When this bit is set to 0, the split-memory block is enabled. The split-memory block size is 384KB or 512KB, depending on the 640 bit, and its address is determined in the Split Address register. When this bit is set to 1, the split-memory block is disabled; however, 128KB of the split-memory block can be used in the ROM address space by setting bit 1 (ROMEN) to 0.

Bit 2 This bit determines where the first active 1MB block of RAM is split.

When this bit is set to 0, 640KB of the first 1MB of memory is mapped into the first 640KB of address space (hex 00000000 to 0009FFFF). When set to 1, 512KB of the first 1MB of memory is mapped into the first 512KB of address space (hex 00000000 to 0007FFFF) and addresses 00080000 through 0009FFFF are unused.

Bit 1 This bit can deactivate the system board ROM and use its address space for system board RAM.

When set to 0, this bit places the ROM address space (hex 000E0000 to 000FFFFF) into RAM. The split-memory block must be disabled. 256KB or 384KB of memory is unused (depending on the 640 bit) and the ROM modules are deactivated. When this bit is set to 1, the ROM modules are left active.

The following figure shows how bits 3, 2, and 1 are used together to define the various configurations.

Bits 3 2 1	Function
0 0 0	Invalid
0 0 1	ROM Enabled, Split at 640KB, High 384KB at Split Address
0 1 0	Invalid
0 1 1	ROM Enabled, Split at 512KB, High 512KB at Split Address
1 0 0	ROM Disabled, Split at 640KB, 128KB of High 384KB at ROM Address
1 0 1	ROM Enabled, Split at 640KB, High 384KB Not used
1 1 0	ROM Disabled, Split at 512KB, 128KB of High 512KB at ROM Address
1 1 1	ROM Enabled, Split at 512KB, High 512KB Not Used

Figure 3-11. Memory Encoding Register, Bits 3, 2, and 1 (Type 1)

Bit 0 When set to 0, this bit enables parity checking of system board memory. To clear a parity error, this bit must be set to 1 and then to 0.

Split Address Register – Type 1 (Hex 00E0)

This read/write register defines the starting address of the split-memory block.

Bit	Function
7 - 4	Reserved
3	SPA23
2	SPA22
1	SPA21
0	SPA20

Figure 3-12. Split Address Register (Type 1)

Bits 7 - 4 These bits are reserved and must be set to 0.

Bits 3 - 0 These bits define the starting address of the split-memory block.

These bits must not be set to a value of 0 unless -ENSPLIT is set to 1. When -ENSPLIT is set to 0, these bits determine the starting address of the split-memory block. The starting address can be at any 1MB boundary from 1MB to 15MB.

Note: If the total system memory is equal to or greater than 16MB, the split-memory block cannot be used.

Memory Registers – Type 2

Two registers define how the system board memory is addressed and used:

- Memory Encoding Register 1 (hex 00E1)
- Memory Encoding Register 2 (hex 00E0).

Memory Encoding Register 1 – Type 2 (Hex 00E1)

This read/write register determines the amount of enabled system board memory in connector 1 and how it is used.

Bit	Function
7, 6	Reserved
5	-Card 1 EN2
4	-Card 1 EN1
3	-ENSPLIT
2	-640
1	ROMEN
0	-ENPLRPCH

Figure 3-13. Memory Encoding Register 1 (Type 2)

Bits 7, 6 These bits are reserved and must be set to 1.

Bits 5 - 4 These bits define the amount of enabled memory in system board memory connector 1.

Bits 5 4	Amount of RAM	Function
0 0	2MB	2MB Card Enabled in Connector 1
0 1	1MB	First 1MB Disabled on 2MB Card in Connector 1
1 0	1MB	Second 1MB Disabled on 2MB Card in Connector 1
1 1	0MB	Invalid (1MB Minimum Must Be Present)

Figure 3-14. Memory Encoding Register 1, Bits 5 and 4 (Type 2)

Bit 3 This bit determines whether the split-memory block is assigned addresses or is disabled. The top 128KB of the first 1MB is always mapped into the ROM address space and the code in ROM is copied into this space.

When this bit is set to 0, the split-memory block is enabled. The split-memory block size is dependent on the value of the 640 bit. Its address is determined in Memory Encoding Register 2. When this bit is set to 1, the split-memory block is disabled.

Note: If the total system memory is equal to or greater than 16MB, the split-memory block cannot be used.

Bit 2 This bit determines where the first active 1MB of memory is split.

When this bit is set to 0, the system maps 640KB of the first 1MB to address hex 00000000 to 0009FFFF. The split-memory block, 256KB, is mapped to the address specified in Memory Encoding register 2.

When this bit is set to 1, the system maps 512KB of the first 1MB to address hex 00000000 to 0007FFFF. The split-memory block, 384KB, is mapped to the address specified in the Memory Encoding register 2. The addresses hex 00080000 to 0009FFFF are unassigned.

Bit 1 This bit determines how addresses hex 000E0000 to 000FFFFF are assigned.

When this bit is set to 1, ROM is enabled and the read-access addresses are assigned to ROM; the write-access addresses are assigned to RAM. When this bit is set to 0, ROM is disabled and read addresses are assigned to RAM; the write addresses are disabled while ROM is disabled.

Note: When this bit is set to 1, system performance will be substantially slower because of 16-bit accesses to ROM and a longer cycle.

The following figure shows how bits 3, 2, and 1 are used together to define the various configurations.

Bits 3 2 1	Function
0 0 0	ROM Disabled, Split at 640KB, 256KB at Split Address
0 0 1	ROM Enabled (ROM Mapped to RAM Disabled), Split at 640KB, 256KB at Split Address
0 1 0	ROM Disabled, Split at 512KB, 384KB at Split Address
0 1 1	ROM Enabled (ROM Mapped to RAM Disabled), Split at 512KB, 384KB at Split Address
1 0 0	ROM Disabled, Split at 640KB, 256KB at Split Address Disabled
1 0 1	ROM Enabled (ROM Mapped to RAM Disabled), Split at 640KB, 256KB at Split Address Disabled
1 1 0	ROM Disabled, Split at 512KB, 384KB at Split Address Disabled
1 1 1	ROM Enabled (ROM Mapped to RAM Disabled), Split at 512KB, 384KB at Split Address Disabled

Figure 3-15. Memory Encoding Register 1, Bits 3 – 1 (Type 2)

Bit 0 When set to 0, this bit enables parity checking of system board memory. To clear a parity error, this bit must be set to 1 and then to 0.

Memory Encoding Register 2 – Type 2 (Hex 00E0)

This read/write register determines the amount of enabled system board memory in connector 2 and the starting address of the split-memory block.

Bit	Function
7, 6	Reserved
5	-Card 2 EN2
4	-Card 2 EN1
3	SPA23
2	SPA22
1	SPA21
0	SPA20

Figure 3-16. Memory Encoding Register 2 (Type 2)

Bits 7, 6 These bits are reserved and must be set to 1.

Bits 5 - 4 These bits define the amount of enabled memory in system board memory connector 2.

Bits 5 4	Amount	Function of RAM
0 0	2MB	2MB Card Enabled in Connector 2
0 1	1MB	First 1MB Disabled on 2MB Card in Connector 2
1 0	1MB	Second 1MB Disabled on 2MB Card in Connector 2
1 1	0MB	All Disabled Or No Card Present.

Figure 3-17. Memory Encoding Register 2, Bits 5 and 4 (Type 2)

Bits 3 - 0 This register defines the starting address of the split-memory block.

These bits must not be set to a value of 0 unless split memory is disabled. When split memory is enabled, these bits determine its starting address. The starting address can be at any 1MB boundary from 1MB to 15MB.

Note: If the total system memory is equal to or greater than 16MB, the split-memory block cannot be used.

System Memory Maps

Memory is mapped by registers at hex 00E0 and 00E1. The mapping results in either 512KB or 640KB of system board RAM starting at address hex 00000000. A 256-byte and 1KB portion and of this RAM is reserved as BIOS data areas. See the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference* for details.

In the following figures, the variable *X* represents the number of 1MB blocks of system board memory starting at or above the hex 00100000 boundary. The variable *Y* represents the number of 1MB blocks of memory installed in the channel starting at or above the hex 00100000 boundary (*Y* = 0 to 15).

The following figure shows the memory mapping when:

Enable Split bit = 1
640 bit = 1
ROM enable bit = 1

Hex Range	Function
00000000 to 0007FFFF	512KB System Board RAM
00080000 to 0009FFFF	Not Used
000A0000 to 000BFFFF	128KB Video RAM
000C0000 to 000DFFFF	Channel ROM
000E0000 to 000FFFFF	128KB System Board ROM
00100000 to (00100000 + XMB)	XMB System Board RAM
(00100000 + XMB) to (00100000 + XMB + YMB)	YMB Channel RAM
(00100000 + XMB + YMB) to FFFDFFFF	Not Used
FFFE0000 to FFFFFFFF	128KB System Board ROM (Same as 000E0000 to 000FFFFF)

Figure 3-18. System Memory Map 1, (Type 1 and Type 2)

The following figure shows the memory mapping when:

Enable Split bit = 1
640 bit = 0
ROM enable bit = 1

Hex Range	Function
00000000 to 0009FFFF	640KB System Board RAM
000A0000 to 000BFFFF	128KB Video RAM
000C0000 to 000DFFFF	Channel ROM
000E0000 to 000FFFFF	128KB System Board ROM
00100000 to (00100000 + XMB)	XMB System Board RAM
(00100000 + XMB) to (00100000 + XMB + YMB)	YMB Channel RAM
(00100000 + XMB + YMB) to FFFDFFFF	Not Used
FFFE0000 to FFFFFFFF	128KB System Board ROM (Same as 000E0000 to 000FFFFF)

Figure 3-19. System Memory Map 2, (Type 1 and Type 2)

The following figures show the memory mapping when:

Enable Split bit = 0
640 bit = 1
ROM enable bit = 1
Split address byte = 1 + *X* + *Y* (Total Range = 1 to 15).

Hex Range	Function
00000000 to 0007FFFF	512KB System Board RAM
00080000 to 0009FFFF	Not Used
000A0000 to 000BFFFF	128KB Video RAM
000C0000 to 000DFFFF	Channel ROM
000E0000 to 000FFFFF	128KB System Board ROM
00100000 to (00100000 + *XMB*)	*XMB* System Board RAM
(00100000 + *XMB*) to (00100000 + *XMB* + *YMB*)	*YMB* Channel RAM
(00100000 + *XMB* + *YMB*) to (00100000 + *XMB* + *YMB* + 512KB - 1)	512KB System Board RAM
(00100000 + *XMB* + *YMB* + 512KB) to FFFDFFFF	Not Used
FFFE0000 to FFFFFFFF	128KB System Board ROM (Same as 000E0000 to 000FFFFF)

Figure 3-20. System Memory Map 3, (Type 1)

Hex Range	Function
00000000 to 0007FFFF	512KB System Board RAM
00080000 to 0009FFFF	Not Used
000A0000 to 000BFFFF	128KB Video RAM
000C0000 to 000DFFFF	Channel ROM
000E0000 to 000FFFFF	128KB System Board ROM
00100000 to (00100000 + *XMB*)	*XMB* System Board RAM
(00100000 + *XMB*) to (00100000 + *XMB* + *YMB*)	*YMB* Channel RAM
(00100000 + *XMB* + *YMB*) to (00100000 + *XMB* + *YMB* +384KB – 1)	384KB System Board RAM
(00100000 + *XMB* + *YMB* + 384KB) to FFFDFFFF	Not Used
FFFE0000 to FFFFFFFF	128KB System Board ROM (Same as 000E0000 to 000FFFFF)

Figure 3-21. System Memory Map 3, (Type 2)

The following figures show the memory mapping when:

Enable Split bit = 0
640 bit = 0
ROM enable bit = 1
Split address byte = 1 + *X* + *Y* (Total Range = 1 to 15).

Hex Range	Function
00000000 to 0009FFFF	640KB System Board RAM
000A0000 to 000BFFFF	128KB Video RAM
000C0000 to 000DFFFF	Channel ROM
000E0000 to 000FFFFF	128KB System Board ROM
00100000 to (00100000 + XMB)	XMB System Board RAM
(00100000 + XMB) to (00100000 + XMB + YMB)	YMB Channel RAM
(00100000 + XMB + YMB) to (00100000 + XMB + YMB + 384KB - 1)	384KB System Board RAM
(00100000 + XMB + YMB + 384KB) to FFFDFFFF	Not Used
FFFE0000 to FFFFFFFF	128KB System Board ROM (Same as 000E0000 to 000FFFFF)

Figure 3-22. System Memory Map 4, (Type 1)

Hex Range	Function
00000000 to 0009FFFF	640KB System Board RAM
000A0000 to 000BFFFF	128KB Video RAM
000C0000 to 000DFFFF	Channel ROM
000E0000 to 000FFFFF	128KB System Board ROM
00100000 to (00100000 + XMB)	XMB System Board RAM
(00100000 + XMB) to (00100000 + XMB + YMB)	YMB Channel RAM
(00100000 + XMB + YMB) to (00100000 + XMB + YMB + 256KB − 1)	256KB System Board RAM
(00100000 + XMB + YMB + 256KB) to FFFDFFFF	Not Used
FFFE0000 to FFFFFFFF	128KB System Board ROM (Same as 000E0000 to 000FFFFF)

Figure 3-23. System Memory Map 4, (Type 2)

The following figures show the memory mapping when:

Enable Split bit = 1
640 bit = 1
ROM enable bit = 0

Hex Range	Function
00000000 to 0007FFFF	512KB System Board RAM
00008000 to 0009FFFF	Not Used
000A0000 to 000BFFFF	128KB Video RAM
000C0000 to 000DFFFF	Channel ROM
000E0000 to 000FFFFF	128KB System Board ROM mapped to RAM
00100000 to (00100000 + XMB)	XMB System Board RAM
(00100000 + XMB) to (00100000 + XMB + YMB)	YMB Channel RAM
(00100000 + XMB + YMB) to FFFDFFFF	Not Used
FFFE0000 to FFFFFFFF	128KB System Board ROM

Figure 3-24. System Memory Map 5, (Type 1)

Hex Range	Function
00000000 to 0007FFFF	512KB System Board RAM
00008000 to 0009FFFF	Not Used
000A0000 to 000BFFFF	128KB Video RAM
000C0000 to 000DFFFF	Channel ROM
000E0000 to 000FFFFF	128KB System Board ROM mapped to RAM
00100000 to (00100000 + XMB)	XMB System Board RAM
(00100000 + XMB) to (00100000 + XMB + YMB)	YMB Channel RAM
(00100000 + XMB + YMB) to FFFDFFFF	Not Used
FFFE0000 to FFFFFFFF	128KB System Board ROM

Figure 3-25. System Memory Map 5, (Type 2)

The following figures show the memory mapping when:

Enable Split bit = 1
640 bit = 0
ROM enable bit = 0

Hex Range	Function
00000000 to 0009FFFF	640KB System Board RAM
000A0000 to 000BFFFF	128KB Video RAM
000C0000 to 000DFFFF	Channel ROM
000E0000 to 000FFFFF	128KB System Board ROM mapped to RAM
00100000 to (00100000 + XMB)	XMB System Board RAM
(00100000 + XMB) to (00100000 + XMB + YMB)	YMB Channel RAM
(00100000 + XMB + YMB) to FFFDFFFF	Not Used
FFFE0000 to FFFFFFFF	128KB System Board ROM

Figure 3-26. System Memory Map 6, (Type 1)

Hex Range	Function
00000000 to 0009FFFF	640KB System Board RAM
000A0000 to 000BFFFF	128KB Video RAM
000C0000 to 000DFFFF	Channel ROM
000E0000 to 000FFFFF	128KB System Board ROM mapped to RAM
00100000 to (00100000 + XMB)	XMB System Board RAM
(00100000 + XMB) to (00100000 + XMB + YMB)	YMB Channel RAM
(00100000 + XMB + YMB) to FFFDFFFF	Not Used
FFFE0000 to FFFFFFFF	128KB System Board ROM

Figure 3-27. System Memory Map 6, (Type 2)

The following figure shows the memory mapping when:

Enable Split bit = 0
640 bit = 1
ROM enable bit = 0
Split address bits = 1 + *X* + *Y* (Total Range = 1 to 15).

Hex Range	Function
00000000 to 0007FFFF	512KB System Board RAM
00080000 to 0009FFFF	Not Used
000A0000 to 000BFFFF	128KB Video RAM
000C0000 to 000DFFFF	Channel ROM
000E0000 to 000FFFFF	128KB System Board ROM mapped to RAM
00100000 to (00100000 + *X*MB)	*X*MB System Board RAM
(00100000 + *X*MB) to (00100000 + *X*MB + *Y*MB)	*Y*MB Channel RAM
(00100000 + *X*MB + *Y*MB) to (00100000 + *X*MB + *Y*MB + 384KB − 1)	384KB of First 1MB at Split Address
(00100000 + *X*MB + *Y*MB + 384KB) to FFFDFFFF	Not Used
FFFE0000 to FFFFFFFF	128KB System Board ROM

Figure 3-28. System Memory Map 7, (Type 2)

The following figure shows the memory mapping when:

Enable Split bit = 0
640 bit = 0
ROM enable bit = 0
Split address bits = 1 + *X* + *Y* (Total Range = 1 to 15).

Hex Range	Function
00000000 to 0009FFFF	640KB System Board RAM
000A0000 to 000BFFFF	128KB Video RAM
000C0000 to 000DFFFF	Channel ROM
000E0000 to 000FFFFF	128KB System Board ROM mapped to RAM
00100000 to (00100000 + *XMB*)	*XMB* System Board RAM
(00100000 + *XMB*) to (00100000 + *XMB* + *YMB*)	*YMB* Channel RAM
(00100000 + *XMB* + *YMB*) to (00100000 + *XMB* + *YMB* + 256KB-1)	256KB of First 1MB at Split Address
(00100000 + *XMB* + *YMB* + 256KB) to FFFDFFFF	Not Used
FFFE0000 to FFFFFFFF	128KB System Board ROM

Figure 3-29. System Memory Map 8, (Type 2)

System Board Memory Connectors

The Type 1 and 2 system boards are equipped with two 3- by 32-pin system board memory connectors. A memory card must be installed in connector 1 before additional memory can be used in the Micro Channel.

The following figure shows the pin locations of the 3- by 32-pin system board memory connectors. The pin locations are the same for connectors 1 and 2. Connector 1 is located closest to the power supply. Pin 32 is closest to the rear of the system board.

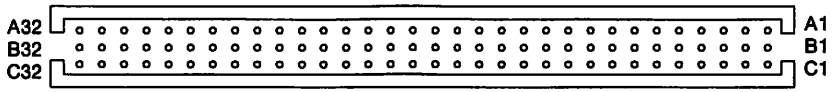


Figure 3-30. System Board Memory Connector Pin Locations

The following is a list of signals used by the system board memory connectors.

-MW -Memory Write

-MA(0-8) -Memory Address (0 through 8)

-RAS(0-3) -Row Address Strobe (0 through 3)

-CAS(0-3) -Column Address Strobe (0 through 3)

-CASP -Column Address Strobe Parity

MDP(0-3) Memory Data Parity (0 through 3)

MD(0-31) Memory Data (0 through 31)

-BE(0-3) -Byte Enable (0 through 3)

R This signal is used by POS Register 3 during system board setup to determine the presence of a memory card in each of the two system board memory connectors.

Note: Type 2 system board POST routines require a minimum of 1MB of functional memory installed in connector 1. The Type 1 system board also requires a minimum of 1MB of functional memory, however it may be installed in either connector.

T This signal is used by POS Register 3 during system board setup to determine the type of memory card installed in each of the two system board memory connectors. See Section 2, "Programmable Option Select" for additional information.

The following figure shows the pin assignments for the system board memory connector. Inputs (I) to and Outputs (O) from the system board are also shown.

Row A			Row B			Row C		
Pin	**I/O**	**Signal**	**Pin**	**I/O**	**Signal**	**Pin**	**I/O**	**Signal**
A1	N/A	Reserved	B1	N/A	Ground	C1	I/O	MD0
A2	O	-MW	B2	O	+5 Vdc	C2	I/O	MD1
A3	O	MA0	B3	N/A	Ground	C3	I/O	MD2
A4	O	MA1	B4	O	+5 Vdc	C4	I/O	MD3
A5	O	MA2	B5	N/A	Ground	C5	I/O	MD4
A6	O	MA3	B6	O	+5 Vdc	C6	I/O	MD5
A7	O	MA4	B7	N/A	Ground	C7	I/O	MD6
A8	O	MA5	B8	O	+5 Vdc	C8	I/O	MD7
A9	O	MA6	B9	N/A	Ground	C9	I/O	MD8
A10	O	MA7	B10	O	+5 Vdc	C10	I/O	MD9
A11	O	MA8	B11	N/A	Ground	C11	I/O	MD10
A12	O	-RAS0	B12	O	+5 Vdc	C12	I/O	MD11
A13	O	-RAS1	B13	N/A	Ground	C13	I/O	MD12
A14	O	-RAS2	B14	O	+5 Vdc	C14	I/O	MD13
A15	O	-RAS3	B15	N/A	Ground	C15	I/O	MD14
A16	N/A	Reserved	B16	O	+5 Vdc	C16	I/O	MD15
A17	I	R	B17	N/A	Ground	C17	I/O	MD16
A18	N/A	Reserved	B18	O	+5 Vdc	C18	I/O	MD17
A19	O	-CAS0	B19	N/A	Ground	C19	I/O	MD18
A20	O	-CAS1	B20	O	+5 Vdc	C20	I/O	MD19
A21	O	-CAS2	B21	N/A	Ground	C21	I/O	MD20
A22	O	-CAS3	B22	O	+5 Vdc	C22	I/O	MD21
A23	I/O	MDP0	B23	N/A	Ground	C23	I/O	MD22
A24	I/O	MDP1	B24	O	+5 Vdc	C24	I/O	MD23
A25	I/O	MDP2	B25	N/A	Ground	C25	I/O	MD24
A26	I/O	MDP3	B26	O	+5 Vdc	C26	I/O	MD25
A27	I	-BE0	B27	N/A	Ground	C27	I/O	MD26
A28	I	-BE1	B28	O	+5 Vdc	C28	I/O	MD27
A29	I	-BE2	B29	N/A	Ground	C29	I/O	MD28
A30	I	-BE3	B30	O	+5 Vdc	C30	I/O	MD29
A31	I	-CASP	B31	N/A	Ground	C31	I/O	MD30
A32	I	T	B32	O	+5 Vdc	C32	I/O	MD31

Figure 3-31. System Board Memory Connector

Real-Time Clock/Complementary Metal-Oxide Semiconductor RAM

The real-time clock/complementary metal-oxide semiconductor RAM (RT/CMOS) chip contains the real-time clock and 64 bytes of CMOS RAM. The internal clock circuitry uses 14 bytes of this memory, and the rest is allocated to configuration and system status information.

In addition to the 64 bytes of CMOS RAM, a 2KB CMOS RAM extension is provided for configuration and other system information.

A 6-Vdc lithium battery maintains voltage to the RT/CMOS RAM and 2KB CMOS RAM extension when the power supply is not in operation.

The system cover can be locked to prevent battery removal and loss of password and configuration information.

The following figure shows the RT/CMOS RAM bytes and their addresses.

Address (Hex)	RT/CMOS RAM Bytes
000 - 00D	Real-Time Clock Bytes
00E	Diagnostic Status Byte
00F	Shutdown Status Byte
010	Diskette Drive Type Byte
011	First Fixed Disk Drive Type Byte
012	Second Fixed Disk Drive Type Byte
013	Reserved
014	Equipment Byte
015 - 016	Low and High Base Memory Bytes
017 - 018	Low and High Expansion Memory Bytes
019 - 031	Reserved
032 - 033	Configuration CRC Bytes
034 - 036	Reserved
037	Date Century Byte
038 - 03F	Reserved

Figure 3-32. RT/CMOS RAM Address Map

RT/CMOS Address Register and NMI Mask (Hex 0070)

This register is used in conjunction with the port at hex 0071 to read and write the RT/CMOS RAM bytes.

Bit	Function
7	NMI Mask
6	Reserved
5 - 0	RT/CMOS RAM Address

Figure 3-33. RT/CMOS Address Register and NMI Mask (Hex 0070)

Warning: The operation following a write to hex 0070 should access port hex 0071; otherwise intermittent malfunctions and unreliable operation of the RT/CMOS RAM can occur.

Bit 7 When this bit is set to 1, the NMI is masked off (the NMI is disabled). This bit is set to 1 by a power-on reset. This is a write-only bit.

Bit 6 Reserved.

Bits 5 - 0 These bits are used to select RT/CMOS RAM addresses.

RT/CMOS Data Register (Hex 0071)

This port is used in conjunction with the address register at hex 0070 to read and write the RT/CMOS RAM bytes.

Bit	Function
7 - 0	RT/CMOS Data

Figure 3-34. RT/CMOS Data Register (Hex 0071)

RT/CMOS RAM I/O Operations

During I/O operations to the RT/CMOS RAM addresses, interrupts should be masked to prevent other interrupt routines from changing the CMOS Address register before data is read or written. After I/O operations, the RT/CMOS and NMI Mask register (hex 0070) should be left pointing to Status Register D (hex 00D).

Warning: The operation following a write to hex 0070 should access hex 0071; otherwise intermittent malfunctions and unreliable operation of the RT/CMOS RAM can occur.

The following steps are required to perform I/O operations to the RT/CMOS RAM addresses:

1. Write the RT/CMOS RAM address to the RT/CMOS and NMI Mask register (hex 0070).
2. Write the data to address hex 0071.

Reading RT/CMOS RAM requires the following steps:

1. Write the RT/CMOS RAM address to the RT/CMOS and NMI Mask register (hex 0070).
2. Read the data from address hex 0071.

Real-Time Clock Bytes (Hex 000-00D)

Bit definitions and addresses for the real-time clock bytes are shown in the following figure.

Address (Hex)	Function	Byte Number
000	Seconds	0
001	Second Alarm	1
002	Minutes	2
003	Minute Alarm	3
004	Hours	4
005	Hour Alarm	5
006	Day of Week	6
007	Date of Month	7
008	Month	8
009	Year	9
00A	Status Register A	10
00B	Status Register B	11
00C	Status Register C	12
00D	Status Register D	13

Figure 3-35. Real-Time Clock Bytes

Note: The Setup program initializes status registers A, B, C, and D when the time and date are set. Interrupt hex 1A is the BIOS interface to read and set the time and date and it initializes the register the same way as the Setup program.

Status Register A (Hex 00A)

Bit	Function
7	Update in Progress
6 - 4	22-Stage Divider
3 - 0	Rate Selection Bits

Figure 3-36. Status Register A

Bit 7 When set to 1, this bit indicates the time-update cycle is in progress. When set to 0, it indicates the current date and time can be read.

Bits 6 - 4 These three divider-selection bits identify which time-base frequency is being used. The system initializes these bits to binary 010, which selects a 32.768 kHz time base. This is the only value supported by the system for proper time-keeping.

Bits 3 - 0 These bits allow the selection of a divider output frequency. The system initializes the rate selection bits to a binary 0110, which selects a 1.024 kHz square-wave output frequency and a 976.562-microsecond periodic interrupt rate.

Status Register B (Hex 00B)

Bit	Function
7	Set
6	Periodic Interrupt Enable
5	Alarm Interrupt Enable
4	Update-Ended Interrupt Enabled
3	Square Wave Enabled
2	Date Mode
1	24-Hour Mode
0	Daylight Savings Enabled

Figure 3-37. Status Register B

Bit 7 When set to 0, this bit updates the cycle, normally by advancing the counts at a rate of one per second. When set to 1, this bit immediately ends any update cycle in progress, and the program can initialize the 14 time bytes without any further updates occurring until this bit is set to 0.

Bit 6 This bit is a read/write bit that allows an interrupt to occur at a rate specified by the rate and divider bits in Status Register A. When set to 1, this bit enables the interrupt. The system initializes this bit to 0.

Bit 5 When set to 1, this bit enables the alarm interrupt. The system initializes this bit to 0.

Bit 4 When set to 1, this bit enables the update-ended interrupt. The system initializes this bit to 0.

Bit 3 When set to 1, this bit enables the square-wave frequency as set by the rate-selection bits in Status Register A. The system initializes this bit to 0.

Bit 2 This bit indicates if the time-and-date calendar updates use binary or binary-coded-decimal (BCD) formats. When set to 1, this bit indicates a binary format. The system initializes this bit to 0.

Bit 1 This bit establishes if the hours byte is in the 24-hour or 12-hour mode. When set to 1, this bit indicates the 24-hour mode. The system initializes this bit to 1.

Bit 0 When set to 1, this bit enables the daylight savings time mode. When set to 0, it disables the mode, and the clock reverts to standard time. The system initializes this bit to 0.

Status Register C (Hex 00C)

Bit	Function
7	Interrupt Request Flag
6	Periodic Interrupt Flag
5	Alarm Interrupt Flag
4	Update-Ended Interrupt Flag
3 - 0	Reserved

Figure 3-38. Status Register C

Note: Interrupts are enabled by bits 6, 5, and 4 in Status Register B.

Bit 7 This bit is used in conjunction with bits 6, 5, and 4. When set to 1, this bit indicates that an interrupt has occurred; bits 6, 5, and 4 indicate the type of interrupt.

Bit 6 When set to 1, this bit indicates that a periodic interrupt occurred.

Bit 5 When set to 1, this bit indicates that an alarm interrupt occurred.

Bit 4 When set to 1, this bit indicates that an update-ended interrupt occurred.

Bits 3 - 0 Reserved.

Status Register D (Hex 00D)

Bit	Function
7	Valid RAM
6 - 0	Reserved

Figure 3-39. Status Register D

Bit 7 This read-only bit monitors the power-sense pin. A low state of this pin indicates a loss of power to the real-time clock (dead battery). When set to 1, this bit indicates that the real-time clock has power. When set to 0, it indicates that the real-time clock has lost power.

Bits 6 - 0 Reserved.

CMOS RAM Configuration

The following shows the bit definitions for the CMOS RAM configuration bytes.

Diagnostic Status Byte (Hex 00E)

Bit	Function
7	Real-Time Clock Chip Power
6	Configuration Record and Checksum Status
5	Incorrect Configuration
4	Memory Size Miscompare
3	Fixed Disk Controller/Drive C Initialization Status
2	Time Status Indicator
1	Adapter Configuration Miscompare
0	Adapter ID Time-Out

Figure 3-40. Diagnostic Status Byte

Bit 7 When set to 1, this bit indicates the real-time clock chip lost power.

Bit 6 When this bit is set to 1, the checksum is incorrect.

Bit 5 This is a check, at power-on time, of the Equipment byte. When set to 1, the configuration information is incorrect. Power-on checks require that at least one diskette drive be installed (bit 0 of the Equipment byte, hex 014, is set to 1).

Bit 4 When set to 1, this bit indicates the power-on check determined that the memory size is not the same as in the configuration record.

Bit 3 When set to 1, this bit indicates that the controller or drive C failed initialization, which prevents the system from attempting a power-on reset.

Bit 2 When set to 0, this bit indicates the time is valid. When set to 1, this bit indicates the time is invalid.

Bit 1 This bit indicates if the installed adapters match the configuration information. When this bit is set to 1, the adapters do not match the configuration information.

Bit 0 When set to 1, this bit indicates a time-out occurred while an adapter ID was being read.

Shutdown Status Byte (Hex 00F): This byte is defined by the power-on diagnostic programs.

Diskette Drive Type Byte (Hex 010): This byte indicates the type of diskette drive installed.

Bit	Function
7 - 4	First Diskette Drive Type
3 - 0	Second Diskette Drive Type

Figure 3-41. Diskette Drive Type Byte

Bits 7 - 4 These bits indicate the first diskette drive type, as shown in the following figure.

Bits 7 6 5 4	Function
0 0 0 0	No drive present
0 0 0 1	Double-sided diskette drive (48 tracks per inch, 360KB)
0 0 1 1	High-capacity diskette drive (720KB)
0 1 0 0	High-density diskette drive (1.44MB)

Note: All combinations that are not shown are reserved.

Figure 3-42. Diskette Drive Type Byte (Bits 7 - 4)

Bits 3 - 0 These bits indicate the second diskette drive type, as shown in the following figure.

Bits 3 2 1 0	Function
0 0 0 0	No drive present
0 0 0 1	Double-sided diskette drive (48 tracks per inch, 360KB)
0 0 1 1	High-capacity diskette drive (720KB)
0 1 0 0	High-density diskette drive (1.44MB)

Note: All combinations that are not shown are reserved.

Figure 3-43. Diskette Drive Type Byte (Bits 3 - 0)

First Fixed Disk Drive Type Byte (Hex 011): This byte defines the type of the first fixed disk drive (drive C). Hex 00 indicates that a fixed disk drive is *not* installed.

Second Fixed Disk Drive Type Byte (Hex 012): This byte defines the type of the second fixed disk drive (drive D). Hex 00 indicates that a fixed disk drive is *not* installed.

Note: For more information about fixed disk drive types, refer to the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference.*

Reserved Byte (Hex 013): This byte is reserved.

Equipment Byte (Hex 014): The equipment byte defines the basic equipment in the system for the power-on diagnostic tests.

Bit	Function
7, 6	Number of Diskette Drives
5, 4	Display Operating Mode
3, 2	Reserved
1	Math Coprocessor Presence
0	Diskette Drive Presence

Figure 3-44. Equipment Byte

Bits 7, 6 These bits indicate the number of diskette drives installed, as shown in the following figure.

Bits 7 6	Number of Diskette Drives
0 0	One Drive
0 1	Two Drives
1 0	Reserved
1 1	Reserved

Figure 3-45. Equipment Byte (Bits 7, 6)

Bits 5, 4 These bits indicate the operating mode of the display attached to the video port, as shown in the following figure.

Bits 5 4	Display Operating Mode
0 0	Reserved
0 1	40-Column Mode
1 0	80-Column Mode
1 1	Monochrome Mode

Figure 3-46. Equipment Byte (Bits 5, 4)

Bits 3, 2 Reserved.

Bit 1 When set to 1, this bit indicates that a math coprocessor is installed.

Bit 0 When set to 1, this bit indicates that a diskette drive is installed.

Low and High Base Memory Bytes (Hex 015 and 016): These bytes define the amount of memory below the 640KB address space.

The value from these bytes represents the number of 1KB blocks of base memory. For example, hex 0280 is equal to 640KB. The low byte is hex 15; the high byte is hex 16.

Low and High Expansion Memory Bytes (Hex 017 and 018): These bytes define the amount of memory above the 1MB address space.

The hexadecimal values in these bytes represent the number of 1KB blocks of expansion memory. For example, hex 0800 is equal to 2048KB. The low byte is hex 17; the high byte is hex 18.

Reserved Bytes (Hex 019 through 031): These bytes are reserved.

Configuration CRC Bytes (Hex 032 and 033): These bytes contain the cyclic-redundancy-check data for bytes hex 010 through hex 031 of the 64-byte CMOS RAM. The low byte is hex 33; the high byte is hex 32.

Reserved Bytes (Hex 034 through 036): These bytes are reserved.

Date Century Byte (Hex 037): Bits 7 through 0 of this byte contain the BCD value for the century. Refer to the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference* for information about reading and setting this byte.

Reserved Bytes (Hex 038 through 03F): These bytes are reserved.

Miscellaneous System Functions

Nonmaskable Interrupt

The nonmaskable interrupt (NMI) signals the system microprocessor that a parity error, a channel check, a system channel time-out, or a system Watchdog time-out has occurred. The NMI stops all arbitration on the bus until bit 6 of the Arbitration register (I/O address hex 0090) is set to 0. This can result in lost data or an overrun error on some I/O devices. The NMI masks all other interrupts and the IRET instruction restores the interrupt flag to the state it was in prior to the interrupt. A system reset causes a reset of the NMI.

Nonmaskable interrupt requests from system board parity and channel check are subject to mask control with the NMI mask bit in the RT/CMOS Address register. The Watchdog Timer and system channel time-out are not masked by this bit. (See "RT/CMOS Address Register and NMI Mask (Hex 0070)" on page 3-37). The power-on default of the NMI mask is 1 (NMI disabled). Prior to enabling the NMI after a power-on reset, the parity check and channel check state are initialized by the POST.

Warning: The operation following a write to hex 0070 should access port hex 0071; otherwise intermittent malfunctions and unreliable operation of the RT/CMOS RAM can occur.

System Control Port B (Hex 0061)

Bit definitions for the read and write functions of this port are shown in the following figures.

Bit	Function
7	Reset Timer 0 Output Latch (IRQ0)
6 - 4	Reserved
3	Enable Channel Check
2	Enable Parity Check
1	Speaker Data Enable
0	Timer 2 Gate to Speaker

Figure 3-47. System Control Port B (Write)

Bit	Function
7	Parity Check
6	Channel Check
5	Timer 2 Output
4	Toggles with Each Refresh Request
3	Enable Channel Check
2	Enable Parity Check
1	Speaker Data Enable
0	Timer 2 Gate to Speaker

Figure 3-48. System Control Port B (Read)

Bit 7 Setting this bit to 1 resets IRQ0. Reading this bit as a 1 indicates a parity check has occurred.

Bit 6 Reading this bit as a 1 indicates a channel check has occurred.

Bit 5 This bit indicates the condition of the timer/counter 2 'output' signal.

Bit 4 This bit toggles for each refresh request.

Bit 3 Setting this bit to 0 enables channel check. This bit is set to 1 during a power-on reset.

Bit 2 Setting this bit to 0 enables parity check. This bit is set to 1 during a power-on reset.

Bit 1 Setting this bit to 1 enables speaker data.

Bit 0 Setting this bit to 1 enables the timer 2 gate.

System Control Port A (Hex 0092)

Bit	Function
7, 6	Fixed-Disk Activity Light
5	Reserved
4	Watchdog Timer Status
3	Security Lock Latch
2	Reserved = 0
1	Alternate Gate A20
0	Alternate Hot Reset

Figure 3-49. System Control Port A

Bits 7, 6 These bits control the fixed-disk activity light. Setting either bit to 1 turns the fixed-disk activity light on. Setting both bits to 0 turns the light off. The power-on reset condition of each bit is 0.

Bit 5 Reserved.

Bit 4 This read-only bit indicates the Watchdog Timer status. When this bit is set to 1, a Watchdog time-out has occurred. For more information about the Watchdog Timer, refer to the *Hardware Interface Technical Reference*.

Bit 3 This bit provides the security lock for the secured area of RT/CMOS. Setting this bit to 1 electrically locks the 8-byte, power-on password. Once this bit is set by POST, it can only be cleared by turning the system off.

Bit 2 Reserved.

Bit 1 This bit is used to enable the 'address 20' signal (A20) when the microprocessor is in the real address mode. When this bit is set to 0, A20 cannot be used in real mode addressing. This bit is set to 0 during a system reset.

Bit 0 This bit provides an alternate method of resetting the system microprocessor. This alternate method supports operating systems requiring faster operation than was provided on the IBM Personal Computer AT®. Resetting the system microprocessor is used to switch the microprocessor from the protected mode to the real address mode. The alternate reset takes 13.4 microseconds.

This bit is set to 0 either by a system reset or a Write operation. When a Write operation changes this bit from 0 to 1, the alternate reset pin is pulsed high for 100 to 125 nanoseconds. The reset occurs after a minimum delay of 6.72 microseconds. While the reset is occurring, the latch remains set so that POST can read this bit. If the bit is 0, POST assumes the system was just powered on. If the bit is 1, POST assumes a switch from the protected mode to the real mode has taken place.

IBM Personal Computer AT is a registered trademark of the International Business Machines Corporation.

Power-On Password

RT/CMOS RAM has 8 bytes reserved for the power-on password and its check character. The 8 bytes are initialized to hex 00. The microprocessor can only access these bytes during power-on self-test (POST). After POST is completed, if a power-on password is installed, the password bytes are locked and cannot be accessed by a program. A power-on password can be from 1 to 7 characters.

During power-on password installation, the password (1 to 7 keyboard scan codes), is stored in the security space.

Power-on password installation is a function of a program contained on the Reference diskette. Once the power-on password utility has been installed, the password can be changed only during the POST. When the new power-on password is installed, changed, or removed, the password is not visible on the display.

The system unit cover can be physically locked to prevent unauthorized access to the battery. This helps prevent unauthorized battery removal and loss of power-on password and configuration information.

For information about the keyboard password, see the "Keyboard and Auxiliary Device Controller" section in the *Hardware Interface Technical Reference.*

Hardware Compatibility

The Model 80 maintains many of the interfaces used by the IBM Personal Computer AT. In most cases command and status organization of these interfaces is maintained.

The functional interfaces for the Model 80 are compatible with the following interfaces:

- The Intel[1] 8259 interrupt controllers (without edge triggering).
- The Intel 8253 timers driven from 1.193 MHz (timer 0 and 2 only).
- The Intel 8237 DMA controller-address/transfer counters, page registers and status fields only. The Command and Request registers are not supported. The rotate and mask functions are not supported. The Mode register is partially supported.
- The NS16450 serial port.
- The Intel 8088, 8086, and 80286 microprocessors.
- The Intel 8272 diskette drive controller.
- The Motorola[2] MC146818 Time of Day Clock command and status (CMOS reorganized).
- The Intel 8042 keyboard port at address hex 0060.
- Display modes supported by the IBM Monochrome Display and Printer Adapter, the IBM Color/Graphics Monitor Adapter, and the IBM Enhanced Graphics Adapter.
- The parallel printer ports (Parallel 1, Parallel 2, and Parallel 3) in compatibility mode.
- Generally compatible with the Intel 80287 and 8087 math coprocessors.

---

1 Intel is a trademark of the Intel Corporation.

2 Motorola is a trademark of Motorola, Inc.

Notes:

Index

A

activity light 3-50
adapter setup register 2-12
address maps
  memory 3-26
  POS 2-5
  system 1-6
address space, POS 2-5
address 20 gate 3-50
alternate reset 3-51
arbiter programming 3-14
arbitration control point 3-12
arbitration level 3-12
arbitration register 3-14
arbitration, central control point 3-14

B

base memory bytes 3-47
battery 3-36
bit values 2-11
block diagram, system board 1-5
board, system 1-3
bus contention 2-3
bus priority assignments 3-12

C

card selected feedback 2-6
-CD SFDBK 2-6
central arbitration control point 3-12
channel 1-4, 3-3
channel check enable 3-49
channel check latch 3-49
channel connectors 3-4
clock speed 1-3
CMOS RAM 3-36
configuration CRC bytes 3-47
configuration, RT/CMOS 3-43
connector, matched memory 3-4
connectors, memory 3-33
coprocessor, math 3-3
cycles
  asynchronous extended 3-6
  matched memory 3-6
  synchronous extended 3-6

D

date century byte 3-47
default 3-6
description
  memory 3-16
  NMI 3-48
  password 3-52
  real-time clock 3-36
  system board 1-3
diagnostic status byte (RT/CMOS) 3-43
diagram, system board 1-5
diagrams
diskette drive type byte 3-44
diskette overruns 3-13
DMA channels 3-13

E

enable error checking 3-25
enable NMI 3-37
encoding register, type 1 3-20
encoding registers, type 2 3-23
equipment byte 3-46
error recovery 3-18
extended arbitration 3-15
extension, matched memory 3-4

F

features 1-3
features, system board 1-3
feedback, card selected 2-6
fixed disk activity light 3-50
fixed disk drive initialization 3-43
fixed disk type byte 3-45

G

gate A20 3-50

I

I/O address map 1-6
interrupts
  IRQ0 reset 3-49
  NMI reset 3-48

M

matched memory 3-4
matched memory cycle 3-6
matched memory extension 3-4
math coprocessor 3-3
maximum current 1-9
measurements 1-9
memory
  addressing 3-19
  available memory 2-9
  channel RAM 3-27
  channel ROM 3-27
  connectors 3-33
  description 3-16
  encoding register, type 2 3-23
  encoding registers, Type 1 3-20
  error recovery 3-18
  maps 3-26, 3-27
  matched memory cycle 3-6
  matched memory timing (no waits) 3-8
  matched memory timing (one wait) 3-10
  POS register 3, type 2 2-10
memory *(continued)*
  POST routines 3-17, 3-34
  random access memory subsystem 3-17
  read-only Type 1 3-16
  reassigning 3-18
  refresh 3-13
  registers, Type 2 3-23
  ROM enable 3-21
  signals 3-33, 3-35
  split 3-19, 3-21
  split address, Type 1 3-22
  split block 3-23
  split-memory block 3-19
  support 3-17
  system board 3-27
  system memory maps 3-26
  video 3-27
memory connector location 2-9, 2-10
memory size miscompare 3-43
micro channel 1-4
microprocessor
  alternate reset 3-51
  mode switch compatibility 3-51
  real address mode 3-50
microprocessor speed 1-3
-MMC 3-4
-MMC CMD 3-5
-MMCR 3-4
mode switching 3-51
model byte 1-3

N

NMI mask 3-37
NMI service 3-13
NMI signal 3-13
nonmaskable interrupt (NMI) 3-48

P

parallel port select 2-8
parity check 3-25
parity check enable 3-49
password, power-on 3-52
performance 3-3
performance, system 1-7
ports, system control 3-48
POS register 2 2-8
POS register 3, type 1 2-9
POS register 3, type 2 2-10
power specifications 1-9
power supply 1-4
programmable option select 2-3
programmable option select (POS)
  address decode 2-6
  address map 2-5
  enable/-setup register 2-7
  registers 2-5
  setup procedures 2-4
  system board 2-6
  system board I/O 2-8
  system configuration utilities 2-3
  system memory maps 3-26
  video 2-7
  warning 2-3
programming, arbiter 3-14
protected mode switch 3-51

R

RAM 3-17
random access memory 3-17
read-only memory, (Type 2) 3-16
read-only memory, Type 1 3-16
real mode switch 3-51
real-time clock bytes 3-39
refresh 3-13
refresh request 3-49
registers
  adapter setup 2-12
  card feedback 2-6
  enable/-setup register 2-7
registers *(continued)*
  memory encoding, type 1 3-20
  memory encoding, type 2 3-23
  miscellaneous system 3-48
  RT/CMOS and NMI mask 3-37
  RT/CMOS status registers 3-40
  split address, Type 1 3-22
  system control 3-48
ROM enable 3-24
ROM Type 2 3-16
RT/CMOS address register 3-37
RT/CMOS RAM 3-36
  password 3-52
  secured area 3-50
R1,R2 bit values 2-11

S

second fixed disk type byte 3-45
setup
  bus contention 2-3
  enable/-setup register 2-7
  memory encoding register, type 2 3-23
  memory split 3-21
  POS register 3, type 1 2-9
  POS register 3, type 2 2-10
  ROM enable 3-21
  split memory address, Type 1 3-22
  system board 2-6
  system board I/O 2-8
  system memory maps 3-26
  video 2-7
shutdown status byte 3-44
signals
  -MMC 3-4
  -MMC CMD 3-5
  -MMCR 3-4
  matched memory 3-4
  memory 3-33, 3-35
size 1-9
speaker data enable 3-49

specifications, system 1-7
specifications, system unit 1-9
split address register, Type 1 3-22
split memory block 3-19, 3-23
status registers (RT/CMOS) 3-40
subsystems, memory 3-16
switch to real 3-51
system address map 1-6
system board 1-3
  block diagram 1-5
  features 1-3
  I/O register 2-8
  memory connectors 3-33
  memory map 3-27
  setup 2-6
system board features 1-3
system configuration utilities 2-3
system control port A 3-50
system control ports 3-48
system overview 1-3
system ports, miscellaneous 3-48
system reset 3-48
system unit specifications 1-7, 1-9

T

time status indicator 3-43
timing
  matched memory (no waits) 3-8
  matched memory (one wait) 3-10
total system memory 1-3
T1,T2 bit values 2-11

V

valid RAM bit 3-42
video setup 2-7
voltage 1-3

W

warnings
  arbitration 3-13
  central arbiter 3-15
  NMI mask 3-48
  POS 2-3
  real-time clock 3-37
watchdog status 3-50

Numerics

80387 math coprocessor 3-3



Personal System/2
Model 90 XP 486
Technical Reference

First Edition (October, 1990)

The following paragraph does not apply to the United Kingdom or any country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.

It is possible that this publication may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

Requests for technical information about IBM products should be made to your IBM Authorized Dealer or your IBM Marketing Representative.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, Purchase, NY 10577.

© Copyright International Business Machines Corporation 1990. All rights reserved.

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Special Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights or other legally protectible rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, programs, or services, except those expressly designated by IBM, are the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, Purchase, NY 10577.

The following terms, denoted by an asterisk (*) in this publication, are trademarks of the IBM Corporation in the United States and/or other countries:

IBM
Micro Channel
Personal Computer AT
Personal System/2
PS/2

The following terms, denoted by a double asterisk (**) in this publication, are trademarks of other companies as follows:

Intel	Intel Corporation
Motorola	Motorola, Incorporated
National Semiconductor	National Semiconductor Corporation

Notes:

Preface

The Technical Reference library is intended for those who develop hardware and software products for IBM Personal Computers and IBM Personal System/2 products. Users should understand computer architecture and programming concepts.

This manual consists of the following sections:

Section 1, "System Overview," describes the system, features, and specifications.

Section 2, "Programmable Option Select," describes the registers used for configuration.

Section 3, "System Board," describes the system-specific hardware implementations of the Micro Channel architecture.

Section 4, "Processor Complex" describes the components and features of the processor complex.

This technical reference should be used with the following publications. These publications contain additional information on many of the subjects discussed in this technical reference.

IBM Personal System/2 Hardware Interface Technical Reference — Architectures

IBM Personal System/2 Hardware Interface Technical Reference — Common Interfaces

IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference

Information about diskette drives, fixed disk drives, adapters, and external options are in separate option technical references.

Warning: The term "Reserved" describes certain signals, bits, and registers that should not be changed. Use of reserved areas can cause compatibility problems, loss of data, or permanent damage to the hardware. When the contents of a register are changed, the state of the reserved bits must be preserved. When possible, read the register first and change only the bits that must be changed.

© Copyright IBM Corp. 1990

Notes:

Contents

Section 1. System Overview 1-1
Description 1-2
System Board Devices and Features 1-3
System Board I/O Address Map 1-4
Micro Channel Compatibility 1-4
Specifications 1-6
Power Supply 1-8
  Outputs 1-8
  Voltage Sequencing 1-9
  Power Supply Connectors 1-10

Section 2. Programmable Option Select 2-1
Description 2-2
POS Address Map 2-4
Card Selected Feedback 2-5
System Board Setup 2-5
  System Board Enable/Setup Register (Hex 0094) 2-5
  System Board POS Register 2 (Hex 0102) 2-6
  System Board POS Register 3 (Hex 0103) 2-8
  System Board POS Registers 4 (Hex 0104) 2-9
  System Board POS Register 5 (Hex 0105) 2-10
  Diskette Drive POS Registers 4 (Hex 0104) 2-11
  Adapter Enable/Setup Register (Hex 0096) 2-12
  Second Serial Port POS Register 2 (Hex 0102) 2-13
  Second Serial Port POS Register 4 (Hex 0104) 2-13
  Second Serial Port POS Register 5 (Hex 0105) 2-14
Selectable Boot Sequence 2-15

Section 3. System Board 3-1
Description 3-2
  Diskette Drive Connector 3-2
  System Board Memory Connectors 3-3
  Real-Time Clock/Complementary Metal-Oxide Semiconductor RAM 3-5
Miscellaneous System Functions 3-17
  Nonmaskable Interrupt 3-17
  System Control Port B (Hex 0061) 3-17
  System Control Port A (Hex 0092) 3-19
  Power-On Password 3-21
Hardware Compatibility 3-22
Error Codes 3-23

© Copyright IBM Corp. 1990

Section 4. Processor Complex 4-1
Type 1 4-3
Type 2 4-5
I/O Address Map 4-7
Optional Cache 4-7
Micro Channel 4-7
  Central Arbiter 4-8
  Central Arbiter Programming 4-9
Memory Subsystem 4-11
  Memory Presence Detect 4-11
  Memory Controller Registers 4-12
Processor Cache Address Map 4-28
  Read-Only Memory Subsystem 4-29

Index X-1

Figures

1-1. System Board Devices and Features 1-3
1-2. System Board I/O Address Map 1-4
1-3. Physical Specifications 1-6
1-4. Acoustical Readings 1-7
1-5. Power Provided for Internal Devices 1-8
1-6. Load Current for the Micro Channel Connectors 1-9
1-7. Voltage Assignments for the Internal Power-Supply Connectors 1-10
2-1. POS I/O Address Map 2-4
2-2. Card Selected Feedback Register (Hex 0091) 2-5
2-3. System Board Enable/Setup Register (Hex 0094) 2-5
2-4. System Board POS Register 2 (Hex 0102) 2-6
2-5. Parallel Port Select Bits 2-7
2-6. System Board POS Register 3 (Hex 0103) — Write 2-8
2-7. System Board POS Register 3 (Hex 0103) — Read 2-8
2-8. System Board POS Register 4 (Hex 0104) 2-9
2-9. System Board POS Register 5 (Hex 0105) 2-10
2-10. Primary Serial Port Address Select Bits 2-10
2-11. Diskette Drive POS Register 4 (Hex 0104) 2-11
2-12. Adapter Enable/Setup Register (Hex 0096) 2-12
2-13. Second Serial Port POS Register 2 (Hex 0102) 2-13
2-14. Second Serial Port POS Register 4 (Hex 0104) 2-13
2-15. Second Serial Port POS Register 5 (Hex 0105) 2-14
2-16. Second Serial Port Address Select Bits 2-14
3-1. Diskette Drive Connector 3-2
3-2. Memory Connector Numbering 3-3
3-3. System Board Memory Connector 3-4
3-4. RT/CMOS RAM Address Map 3-5
3-5. RT/CMOS Address Register and NMI Mask (Hex 0070) . 3-6
3-6. RT/CMOS Data Register (Hex 0071) 3-6
3-7. Real-Time Clock Bytes 3-8
3-8. Status Register A 3-9
3-9. Status Register B 3-9
3-10. Status Register C 3-11
3-11. Status Register D 3-11
3-12. Diagnostic Status Byte 3-12
3-13. Diskette Drive Type Byte 3-13
3-14. Diskette Drive Type Byte (Bits 7—4) 3-13
3-15. Diskette Drive Type Byte (Bits 3—0) 3-13
3-16. Equipment Byte 3-15
3-17. Equipment Byte (Bits 7, 6) 3-15
3-18. Equipment Byte (Bits 5, 4) 3-15

© Copyright IBM Corp. 1990

3-19. System Control Port B (Hex 0061) – Write 3-18
3-20. System Control Port B (Hex 0061) – Read 3-18
3-21. System Control Port A (Hex 0092) 3-19
3-22. POST Error Message Table 3-24
4-1. Model and Submodel Bytes 4-2
4-2. Type 1 Processor Complex Feature and Devices 4-3
4-3. System Performance Specifications, Type 1 Processor Complex 4-3
4-4. Memory Performance Specifications, Type 1 Processor Complex 4-3
4-5. Type 2 Processor Complex Feature and Devices 4-5
4-6. System Performance Specifications, Type 2 Processor Complex 4-5
4-7. Memory Performance Specifications, Type 2 Processor Complex 4-5
4-8. Processor Complex I/O Address Map 4-7
4-9. Arbitration Bus Priority Assignments 4-8
4-10. Arbitration Register (Hex 0090) – Write 4-9
4-11. Arbitration Register (Hex 0090) – Read 4-9
4-12. Presence-Detect Bits 4-11
4-13. Memory Controller Register Assignments 4-12
4-14. Memory Control 4-14
4-15. Memory Bank Base Address Register (Index Hex 90 – 9F) 4-15
4-16. Split Address Register (Index Hex A0) 4-15
4-17. Memory Encoding Register (Index Hex A1) 4-16
4-18. Cache/Timer Control Register (Index Hex A2) 4-18
4-19. Boundary 0 Cache Control Register (Index Hex A3) . . 4-19
4-20. Lower Boundary 1 Cache Control Register (Index Hex A4) 4-20
4-21. Upper Boundary 1 Cache Control Register (Index Hex A5) 4-20
4-22. Memory Bank Enable Register 1 (Index Hex A6) 4-21
4-23. Memory Bank Enable Register 2 (Index Hex A7) 4-22
4-24. Parity Trap Register (Hex 00E1) 4-25
4-25. Cache Control Register (Hex 00E2) 4-26
4-26. Cache Status Register (Hex 00E3) 4-27
4-27. Cache Address Map Summary 4-28

Section 1. System Overview

Description 1-2
System Board Devices and Features 1-3
System Board I/O Address Map 1-4
Micro Channel Compatibility 1-4
Specifications 1-6
Power Supply 1-8
  Outputs 1-8
    Output Protection 1-9
  Voltage Sequencing 1-9
  Power Supply Connectors 1-10

© Copyright IBM Corp. 1990

Description

The IBM* Personal System/2* Model 90 is a self-contained, desktop computer system with a keyboard. The system can support maximum of four internal drives. Three of these drives can be removable-media DASD devices (diskette, tape, optical) or three can be nonremovable-media DASD devices.

All Model 90 systems use the same system board. The system board has:

- Four Micro Channel* connectors
- Extended Graphics Array (XGA) Video Subsystem
- Diskette drive, parallel port, and two serial port controllers
- Eight memory connectors on two riser cards
- Processor complex connector
- Real-time clock.

The processor complex determines system performance and the model/submodel byte information. The major components on the processor complex are:

- System microprocessor
- DMA controller
- Memory controller.

* IBM, Personal System/2, PS/2, and Micro Channel are trademarks of the International Business Corporation.

System Board Devices and Features

The following table lists the system board devices and features. The *Hardware Interface Technical Reference — Common Interfaces* describes devices common to PS/2 products by type number.

Device	Type	Features
Microprocessor	—	Connector for processor complex
System Timers	1	Channel 0 — System timer Channel 2 — Tone generation for speaker Channel 3 — Watchdog timer
RAM Subsystem	—	Two riser cards provide eight connectors Refer to the processor complex for more information
CMOS RAM Subsystem	—	64-byte CMOS RAM with real-time clock/calendar 8KB CMOS RAM extension (KB = 1,024 bytes) Battery backup
Audio Subsystem	1	Driven by: — System-timer channel 2 — The 'audio sum node' signal
Video Subsystem	2	XGA video functions — VGA mode — Extended graphics mode 512KB or 1MB of video memory
Interrupt Controller	1	16 levels of system interrupts Interrupts are level-sensitive
Keyboard/Auxiliary Device Controller	2	Keyboard connector Auxiliary device connector Password security
Diskette Drive Controller	2	Supports: — 3.5-in. diskette drive (1.44MB) — 5.25-in. diskette drive (360KB and 1.2MB)
Serial Controller	3	Two RS-232C interfaces — One 25-pin connector (primary serial) — One 9-pin connector (second serial) Programmable as serial ports 1–8 DMA, FIFO, and character modes
Parallel Controller	3	Programmable as parallel port 1–4 Supports bidirectional input and output Supports DMA operation
Micro Channel	—	Four connectors for Type 3 adapters: — One 32-bit connector with an auxiliary video extension — Three 32-bit connectors with matched memory extension One connector contains a fixed disk adapter

Figure 1-1. System Board Devices and Features

System Board I/O Address Map

Hex Addresses	Device
0020, 0021	Interrupt Controller (Master)
0040, 0042–0044, 0047	System Timers
0060	Keyboard, Auxiliary Device
0061	System Control Port B
0064	Keyboard, Auxiliary Device
0070, 0071	RT/CMOS and NMI Mask
0091	Card Selected Feedback Register
0092	System Control Port A
0094	System Board Enable/Setup Register
0096	Adapter Enable/Setup Register
00A0–00A1	Interrupt Controller (Slave)
0100–0107	Programmable Option Select
0108–010F	Information Panel
0278–027D	Parallel Port 3
02F8–02FF	Serial Port 2
0378–037D	Parallel Port 2
03B4–03BB	Video Subsystem
03BC–03BF	Parallel Port 1
03C0–03CF	Video Subsystem
03D4–03DB	Video Subsystem
03F0–03F7	Diskette Drive Controller
03F8–03FF	Serial Port 1
1278–127D	Parallel Port 1 (DMA mode)
1378–137D	Parallel Port 4
2100–217F	Video Subsystem
3220–3227	Serial Port 3
3228–322F	Serial Port 4
4220–4227	Serial Port 5
4228–422F	Serial Port 6
5220–5227	Serial Port 7
5228–522F	Serial Port 8
83F8–83FF	Serial Port 1 (DMA mode)
82F8–82FF	Serial Port 2 (DMA mode)
B220–B227	Serial Port 3 (DMA mode)
B228–B22F	Serial Port 4 (DMA mode)
C220–C227	Serial Port 5 (DMA mode)
C228–C22F	Serial Port 6 (DMA mode)
D220–D227	Serial Port 7 (DMA mode)
D228–D22F	Serial Port 8 (DMA mode)

Figure 1-2. System Board I/O Address Map

Micro Channel Compatibility

The following describe the system limitations in the support of Micro Channel procedures.

Streaming Data

The system board supports 10 MHz streaming data procedures involving transfers between adapters.

Devices on the system board and the processor complex work with streaming-data adapters; however, they do not use the streaming data procedure. Transfers default to the basic transfer procedure.

Data Parity

The system board supports data-parity procedures involving transfers between adapters.

Devices on the system board and the processor complex work with adapters that support data parity; however, they do not use the data parity option. Transfers default to the basic transfer procedure.

Address Parity

The system board supports address-parity procedures involving transfers between adapters.

Devices on the system board and the processor complex work with adapters that support address parity; however, they do not use the address parity option. Transfers default to the basic transfer procedure.

Select Feedback Return Signal

The system board and the processor complex support the 'selected feedback return' signal except for I/O devices at addresses below hex 0110. These devices are controlled by the system master only and should not be accessed by bus masters.

Synchronous Channel Check

Synchronous channel checks for errors other than Micro Channel data-parity errors or Micro Channel address-parity errors are not supported in this system.

Synchronous channel checks for data-parity or address-parity errors, typically issued by slaves, are the responsibility of the master that owned the channel and issued the access to the slave. However, in this system, these channel checks can also be detected by logic on the system board, resulting in an NMI.

DMA

In this system, the DMA controller supports DMA slaves with 8-bit or 16-bit data widths. Transfers are limited to the memory address space below 16MB.

Specifications

Size:	
Width	440 mm (17.3 in)
Depth	430 mm (16.9 in)
Height	140 mm (5.5 in)
Weight	12 kg (26 lb)
Cables:	
Power Cable	1.8 m (6 ft)
Keyboard Cable	1.8 m (6 ft)
Air Temperature:	
System On	
0 - 3000 ft	10.0 to 35.0°C (50 to 95°F)
3000 - 7000 ft	10.0 to 32.2°C (50 to 90°F)
System Off	10.0 to 43.0°C (50 to 110°F)
Humidity:	8% to 80%
Maximum Altitude	2134 m (7000 ft)
Heat Output	329 W (1125 BTU per hour)
Acoustical Readings	
See Figure 1-4 on page 1-7	
Electrical:	
Input Voltage:	Range is automatically selected Sinewave input required
Low Range	90 (min) – 137 (max) Vac
High Range	180 (min) – 265 (max) Vac
Input Frequency:	50 ± 3Hz or 60 ± 3Hz
Maximum Current Draw:	
Low Range	4.8 A
High Range	2.4 A
Input in Kilovolt-Ampere (kVA):	
Minimum Configuration (as shipped)	0.15 kVa
Maximum Configuration	0.48 kVa
Electromagnetic Compatibility	FCC Class B

Figure 1-3. Physical Specifications

Description	L_{WAd} in bels Operate	Idle	L_{pAm} in dB Operate	Idle	$<L_{pA}>_m$ in dB Operate	Idle
Model 90	5.1	5.1	43	43	37	37

Notes:

L_{WAd} is the declared sound power level for the random sample of machines.

L_{pAm} is the mean value of the A-weighted sound pressure levels at the operator position (if any) for the random sample of machines.

$<L_{pA}>_m$ is the mean value of the A-weighted sound pressure levels at the one-meter positions for the random sample of machines.

All measurements made in accordance with ANSI S12.10, and reported in conformance with ISO DIS 9296.

The measurements are preliminary data and subject to change.

Figure 1-4. Acoustical Readings

Power Supply

The power supply requires a sinewave input and converts the ac input voltage to three dc output voltages. The power supply automatically adjusts to match the input voltage range. The power supply provides power for the following:

- System board
- Processor complex
- Micro Channel adapters
- Internal diskette drives
- Internal fixed disk drives
- Auxiliary device
- Keyboard.

The power switch and two light-emitting diode (LED) are on the front of the system unit. The power supply is operating properly when the green LED is lit. The amber LED is the fixed-disk-activity indicator.

Outputs

The power supply provides separate voltage sources for the system board and the drives. The system-board voltages are +5 Vdc, +12 Vdc, and −12 Vdc. The voltages for the internal drives are +5 Vdc and +12 Vdc. The following is the approximate power provided for system components.

System Component	Maximum Current	
	+12	+5
Internal Fixed Disk Drives *	3.2 A	2.3 A
Auxiliary Device		300 mA
Keyboard		275 mA
.* Total for both power supply connectors		

Figure 1-5. Power Provided for Internal Devices

The following are the load currents allowed for each channel connector. The formulas used to determine the power requirements can be found in the section of the *Hardware Interface Technical Reference — Architectures* that describes the Micro Channel architecture.

Supply Voltage	Maximum Current
+ 5.0 Vdc	2.0 A
+12.0 Vdc	0.175 A
−12.0 Vdc	0.040 A

Figure 1-6. Load Current for the Micro Channel Connectors

Output Protection

A short circuit placed on any dc output (between outputs or between an output and dc return) latches all dc outputs into a shutdown state with no damage to the power supply.

If an overvoltage fault occurs (internal to the power supply), the power supply latches all dc outputs into a shutdown state before any output exceeds 130% of its nominal value.

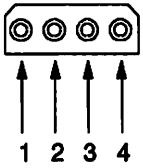
If either of these shutdown states is actuated, the power supply returns to normal operation only after the fault has been removed and the power switch has been turned off for at least one second.

Voltage Sequencing

At power-on time, the output voltages track within 50 milliseconds of each other when measured at the 50% points.

Power Supply Connectors

The power supply provides two 4-pin connectors for internal devices. The total power used by both connectors must not exceed the amount shown in Figure 1-5 on page 1-8.



Pin	Signal	Pin	Signal
1	+12 Vdc	3	DC Return
2	DC Return	4	+5 Vdc

Figure 1-7. Voltage Assignments for the Internal Power-Supply Connectors

Section 2. Programmable Option Select

Description 2-2
POS Address Map 2-4
Card Selected Feedback 2-5
System Board Setup 2-5
  System Board Enable/Setup Register (Hex 0094) 2-5
  System Board POS Register 2 (Hex 0102) 2-6
  System Board POS Register 3 (Hex 0103) 2-8
  System Board POS Registers 4 (Hex 0104) 2-9
  System Board POS Register 5 (Hex 0105) 2-10
  Diskette Drive POS Registers 4 (Hex 0104) 2-11
  Adapter Enable/Setup Register (Hex 0096) 2-12
  Second Serial Port POS Register 2 (Hex 0102) 2-13
  Second Serial Port POS Register 4 (Hex 0104) 2-13
  Second Serial Port POS Register 5 (Hex 0105) 2-14
Selectable Boot Sequence 2-15

© Copyright IBM Corp. 1990

Description

Programmable Option Select (POS) eliminates the need for switches by replacing their function with programmable registers. This section describes the POS information used on the Model 90 system board. For additional POS information, refer to the *Hardware Interface Technical Reference*.

Warning:

- IBM recommends that programmable options be set only through the System Configuration utilities. Directly setting the POS registers or CMOS RAM POS parameters can result in multiple assignment of the same system resource, improper operation of the feature, loss of data, or possible damage to the hardware.
- Application programs should not use the adapter identification (ID) unless absolutely necessary. Compatibility problems can result.
- If an adapter and the system board are in setup mode at the same time, bus contention will occur, no useful programming can take place, and damage to the hardware can occur.
- After setup operations are complete, the Adapter Enable/Setup register (hex 0096) should be set to hex 00, and the System Board Enable/Setup register (hex 0094) should be set to hex FF.
- Bit 7 (channel reset) in the Adapter Enable/Setup register must be 0 to program the adapters.
- Only 8-bit instructions are supported for setup operations. Using 32- or 16-bit I/O instructions on 8-bit POS registers will cause erroneous data to be written or read.

Setup functions respond to I/O addresses hex 0100 through 0107 only when their unique setup signal is active. The following precautions must be taken before setting individual bits in the POS registers.

Primary System Board Setup

- Bit 7 must be set to 0 while other bits in the System Board Enable/Setup register are set to 1 to setup the primary system board functions.
- Bit 3 in the Adapter Enable/Setup register must be set to 0 to avoid driving a 'setup' signal to an adapter.

System Board Video Subsystem Setup:

- Bit 5 in the System Board Enable/Setup register (hex 0094) must be set to 0 to place the system board video into the setup mode.
- Bits 7 and 6 in the System Board Enable/Setup register must be set to 1 to avoid driving a 'setup' signal to other system board functions.
- Bit 3 in the Adapter Enable/Setup register (hex 0096) must be set to 0 to avoid driving a 'setup' signal to an adapter.

Diskette Drive Setup

- Bit 6 must be set to 0 while other bits in the System Board Enable/Setup register are set to 1 to setup the diskette drive functions on the system board.
- Bit 3 in the Adapter Enable/Setup register must be set to 0 to avoid driving a 'setup' signal to an adapter.

Second Serial Port Setup

- Only bits 3, 2, and 0 in the Adapter Enable/Setup register are set to 1 to setup the second serial port on the system board (9-pin connector).
- Bits 7 – 0 in the System Board Enable/Setup register must be set to 1 to avoid driving a 'setup' signal to other system board functions.

Adapter Setup:

- Bit 3 in the Adapter Enable/Setup register must be set to 1 to allow adapter setup.
- Bits 7–0 in the System Board Enable/Setup register must be set to 1 to avoid driving a 'setup' signal to system board functions.

POS Address Map

The following table shows the organization of the I/O address space used by POS. Bit 0 of POS Register 2 and bits 6 and 7 of POS Register 5 are fixed. All other bits in POS Registers 2 through 5 are free-form.

Address (Hex)	Function
0094	System Board Enable/Setup Register
0096	Adapter Enable/Setup Register
0100	POS Register 0—Adapter Identification Byte (Low Byte)
0101	POS Register 1—Adapter Identification Byte (High Byte)
0102	POS Register 2—Option Select Data Byte 1 Bit 0 is Card Enable.
0103	POS Register 3—Option Select Data Byte 2
0104	POS Register 4—Option Select Data Byte 3
0105	POS Register 5—Option Select Data Byte 4
0106	Reserved
0107	Reserved

Figure 2-1. POS I/O Address Map

Card Selected Feedback

When an adapter is addressed, it responds by setting the '-card selected feedback' signal (-CD SFDBK) to active. -CD SFDBK is derived from the address decode and driven by a totem pole driver. It is latched by the system board and can be read through the Card Selected Feedback register at address hex 0091.

The Card Selected Feedback register is a read-only register at address hex 0091. It allows programs to monitor -CD SFDBK and thereby determine if the system board I/O or an adapter is addressed and functioning.

Bit	Function
7–1	Reserved
0	-Card Selected Feedback

Figure 2-2. Card Selected Feedback Register (Hex 0091)

Bits 7–1 These bits are reserved.

Bit 0 This bit is set to 1 whenever -CD SFDBK was active on a previous cycle or whenever the system board I/O functions (diskette drive, primary serial, or parallel interfaces) are accessed by an I/O or DMA cycle. Reading this register resets the bit to 0.

System Board Setup

The integrated I/O functions on the system board use POS information during setup. The diskette drive controller, primary serial port, and parallel port are treated as a single device. The video subsystem is treated as a second, separate device on the system board. The System Board Enable/Setup register is used to place the system board into the setup mode.

System Board Enable/Setup Register (Hex 0094)

This is a read/write register; all bits in this register default to 1 (enabled). Setup functions should only be accessed one at a time.

Bit	Function
7	Enable/-Setup System Board Functions
6	Enable/-Setup Diskette Drive Functions
5	Enable/-Setup Video Subsystem
4–0	Reserved

Figure 2-3. System Board Enable/Setup Register (Hex 0094)

Bit 7 When this bit is set to 0, various system board I/O functions are placed in the setup mode. The diskette drive controller, primary serial port, and parallel port are controlled through System Board POS Register 2 (hex 0102).

When this bit is set to 1, the system board function is enabled.

Bit 6 Additional setup for diskette drive. When this bit is set to 0, these setup functions are available through the Diskette Drive POS Register 4.

Bit 5 When this bit is set to 0, the video subsystem is placed in setup mode. While in setup mode, the video subsystem is controlled through the video subsystem POS registers. When bit 5 is set to a 1, the video subsystem is enabled. (See the Video Subsystem section of the Hardware Interface Technical Reference Manual for operation of the POS registers).

Bits 4–0 These bits are reserved.

System Board POS Register 2 (Hex 0102)

When the system board is in the setup mode, the diskette drive controller, primary serial port, and parallel port are controlled by this read/write register. Reading this register returns the current state of these system board functions.

Bit	Function
7	Disable Parallel Port Extended Mode
6, 5	Parallel Port Select
4	Enable Parallel Port
3	Primary Serial IRQ Select
2	Enable Primary Serial Port
1	Enable Diskette Drive Interface
0	Enable System Board

Figure 2-4. System Board POS Register 2 (Hex 0102)

Bit 7 When set to 0, this bit allows the parallel port to be configured as an 8-bit, parallel, bidirectional interface. When set to 1, this bit disables the bidirectional mode. This bit is set to 0 after power-on.

Bits 6, 5 These bits select the configuration of the system board parallel port.

Bits 6 5	Assignment	Hex Address	Interrupt Level
0 0	Parallel 1	03BC–03BF*	7
0 1	Parallel 2	0378–037D	7
1 0	Parallel 3	0278–027D	7
1 1	Parallel 4	1378–137D	7

* The addresses are hex 1278–127D for the DMA operations.

Figure 2-5. Parallel Port Select Bits

Bit 4 When this bit and bit 0 are set to 1, the system-board parallel port is enabled.

Bit 3 When this bit is set to 1, the primary serial port (25-pin connector) uses interrupt request level 4. When set to 0, the primary serial port uses interrupt request level 3.

Bit 2 When this bit and bit 0 are set to 1, the primary serial port is enabled.

Bit 1 When this bit and bit 0 are set to 1, the diskette drive interface is enabled.

Bit 0 When set to 1, this bit allows bits 4, 2, and 1 to enable and disable their respective devices. When set to 0, this bit disables the diskette drive interface, primary system-board serial port, and system-board parallel port, regardless of the state of bits 4, 2, and 1.

System Board POS Register 3 (Hex 0103)

The system board has connectors that provide installation of eight memory cards (for more information about these memory connectors, see "System Board Memory Connectors" on page 3-3). Information about each memory card is contained in four read-only data registers; each data register is selected by writing its index to the write-only index register.

POS Register 3 provides access to the read-only data registers and a write-only index register. To determine the memory information for a particular connector, write the appropriate index to POS Register 3, then read POS Register 3. The following shows the relationship of the index value, the memory connector, and the data register information available.

Index Value (Hex)	Description
01	Presence Detect Information for A1 and B1
05	Presence Detect Information for A2 and B2
09	Presence Detect Information for A3 and B3
0D	Presence Detect Information for A4 and B4
All values not show are reserved.	

Figure 2-6. System Board POS Register 3 (Hex 0103) — Write

The following information is returned by reading POS Register 3 when the appropriate index value is written to POS Register 3. For coding of the presence detect signals, see Figure 4-12 on page 4-11.

Bit	Description
7	Presence Detect 3, Connector B
6	Presence Detect 2, Connector B
5	Presence Detect 1, Connector B
4	Presence Detect 0, Connector B
3	Presence Detect 3, Connector A
2	Presence Detect 2, Connector A
1	Presence Detect 1, Connector A
0	Presence Detect 0, Connector A

Figure 2-7. System Board POS Register 3 (Hex 0103) — Read

System Board POS Registers 4 (Hex 0104)

This register is accessed when bit 7 of the System Board Enable/Setup register is set to 0. The register is used to specify the arbitration levels used by the primary serial port during transmit and receive operations in the DMA mode.

Bit	Description
7–4	Receive Arbitration Level
3–0	Transmit Arbitration Level

Figure 2-8. System Board POS Register 4 (Hex 0104)

Bits 7–4 These bits select the arbitration level used for the transfer of data being received through the primary serial port.

Bits 3–0 These bits select the arbitration level used for the transfer of data being transmitted through the primary serial port.

System Board POS Register 5 (Hex 0105)

This register is used to specify the arbitration level used by the parallel port and the I/O addresses used by the primary serial port.

Bit	Description
7	Reserved
6–4	Primary Serial Address Select
3–0	Parallel Port Arbitration Level

Figure 2-9. System Board POS Register 5 (Hex 0105)

Bit 7 This bit is reserved.

Note: This bit may read as 0 but does not indicate any occurrence of channel check.

Bits 6–4 These bits are used to select the I/O addresses used by the primary serial port on the system board. The I/O addresses can be the compatible addresses or the extended address (see Serial Port Controller in the *Hardware Interface Technical Reference – Common Interfaces* for more information on addressing).

Bits 6 5 4	Assignment	Hex Address	
		Compatible	Extended
0 0 0	Serial 1	03F8–03FF	83F8–83FF
0 0 1	Serial 2	02F8–02FF	82F8–82FF
0 1 0	Serial 3	3220–3227	B220–B227
0 1 1	Serial 4	3228–322F	B228–B22F
1 0 0	Serial 5	4220–4227	C220–C227
1 0 1	Serial 6	4228–422F	C228–C22F
1 1 0	Serial 7	5220–5227	D220–D227
1 1 1	Serial 8	5228–522F	D228–D22F

Figure 2-10. Primary Serial Port Address Select Bits

Bits 3–0 These bits select the arbitration level used by the parallel port controller.

Diskette Drive POS Registers 4 (Hex 0104)

This register is accessed when the bit 6 of the System Board Enable/Setup register is set to 0. The register is used to specify the arbitration level used by the diskette drive controller.

Bit	Description
7–4	Reserved
3–0	Diskette Drive Arbitration Level

Figure 2-11. Diskette Drive POS Register 4 (Hex 0104)

Bits 7–4 These bits are reserved.

Bits 3–0 These bits select the arbitration level used by the diskette drive controller.

Adapter Enable/Setup Register (Hex 0096)

This read/write register selects the connector to be configured.

Bit	Description
7	Channel Reset
6–4	Reserved = 0
3	Card Setup Enable
2–0	Channel Select 2–0

Figure 2-12. Adapter Enable/Setup Register (Hex 0096)

Bit 7 When set to 1, this bit activates the 'channel reset' signal to all connectors.

Bits 6–4 These bits are reserved and must be set to 0.

Bit 3 When set to 1, this bit enables the '-card setup' signal selected by bits 2 through 0.

Bits 2–0 These bits are the address bits for the '-card setup' signal to the connectors and the second serial port. Connectors 1 through 4 are addressed as 0 through 3; the second serial port is addressed as 5. When bit 3 is set to 1, these bits select the connector or function for setup cycles.

Each channel connector has a unique '-card setup' signal (-CD SETUP) associated with it. This signal is used to select the adapters for setup cycles, which allows access to the POS registers. The individual connectors are selected through the Adapter Enable/Setup register. Setup information is then read from or written to the selected adapter through I/O addresses hex 0100 through 0107.

Note: -CD SETUP only goes active when an operation is performed in the I/O address range hex 0100 through 0107.

Second Serial Port POS Register 2 (Hex 0102)

While the second serial port is selected for setup cycles, this read/write register is used to control the second serial port interface.

Bit	Description
7–4	Reserved
3	Second Serial IRQ Select
2	Second Serial Enable
1	Reserved
0	Second Serial Enable

Figure 2-13. Second Serial Port POS Register 2 (Hex 0102)

Bits 7–4 These bits are reserved.

Bit 3 When this bit is set to 1, the second serial port uses IRQ4; when it is set to 0, the second serial port uses IRQ3,

Bit 2 When this bit and bit 0 are set to 1, the second serial port is enabled. When either bit is 0, the port is disabled.

Bit 1 This bit is reserved.

Bit 0 When this bit and bit 2 are set to 1, the second serial port is enabled. When either bit is 0, the port is disabled.

Second Serial Port POS Register 4 (Hex 0104)

While the second serial port is selected for setup cycles, this read/write register selects the arbitration level used.

Bit	Description
7–4	Receive Arbitration Level
3–0	Transmit Arbitration Level

Figure 2-14. Second Serial Port POS Register 4 (Hex 0104)

Bits 7–4 These bits select the arbitration level used in the transfer of data being received by the second serial port.

Bits 3–0 These bits select the arbitration level used in the transfer of data being transmitted by the second serial port.

Second Serial Port POS Register 5 (Hex 0105)

While the second serial port is selected for setup cycles, this read/write register selects the I/O address of the second serial port.

Bit	Description
7	Reserved
6-4	I/O Address Select
3-0	Reserved

Figure 2-15. Second Serial Port POS Register 5 (Hex 0105)

Bit 7 This bit is reserved.

Note: This bit may read as 0 but does not indicate any occurrence of channel check.

Bits 6–4 These bits are used to select the I/O addresses used by the second serial port on the system board. The I/O addresses can be the compatible addresses or the extended address.

Bits 6 5 4	Assignment	Hex Address	
		Compatible	Extended
0 0 0	Serial 1	03F8–03FF	83F8–83FF
0 0 1	Serial 2	02F8–02FF	82F8–82FF
0 1 0	Serial 3	3220–3227	B220–B227
0 1 1	Serial 4	3228–322F	B228–B22F
1 0 0	Serial 5	4220–4227	C220–C227
1 0 1	Serial 6	4228–422F	C228–C22F
1 1 0	Serial 7	5220–5227	D220–D227
1 1 1	Serial 8	5228–522F	D228–D22F

Figure 2-16. Second Serial Port Address Select Bits

Bits 3–0 These bits are reserved.

Selectable Boot Sequence

Selectable boot allows the user to specify the order in which the drives are searched when looking for the boot record of an operating system. Only the diskette and fixed disk drives can be part of the search sequence. Changing the sequence for selectable boot affects the drive letter ordering the next time the system is booted.

The user selects the boot sequence with the Select Startup feature in the Set Features utility. The user is presented with a list of possible drives from which to select. After the sequence is selected, it is stored in nonvolatile RAM (NVRAM). When the system boots, it searches for a drive that contains a bootable record using the specified sequence.

After the operating system boots, the drive lettering is determined by the sequence and which drive contained the bootable source. If the operating system boots from a diskette drive, that drive is drive A. If the operating system boots from a fixed disk drive, that fixed disk drive is drive C. Other drives specified in the sequence can change also.

Section 3. System Board

Description 3-2
Diskette Drive Connector 3-2
System Board Memory Connectors 3-3
Real-Time Clock/Complementary Metal-Oxide Semiconductor RAM 3-5
RT/CMOS Address Register and NMI Mask (Hex 0070) . . . 3-6
RT/CMOS Data Register (Hex 0071) 3-6
RT/CMOS RAM I/O Operations 3-7
Real-Time Clock Bytes (Hex 000–00D) 3-8
CMOS RAM Configuration 3-12
Miscellaneous System Functions 3-17
Nonmaskable Interrupt 3-17
System Control Port B (Hex 0061) 3-17
System Control Port A (Hex 0092) 3-19
Power-On Password 3-21
Hardware Compatibility 3-22
Error Codes 3-23

© Copyright IBM Corp. 1990

Description

This section describes miscellaneous system ports and connectors for the Model 90. For additional information about these and other topics, refer to other descriptions in the *Hardware Interface Technical Reference*.

Diskette Drive Connector

The system provides three 2-by-17-pin connectors for attaching internal diskette drives. Each connector passes control and data signals between the diskette drive controller on the system board and the drives and provides the power to each drive.

The following figure shows the signal assignments and pin numbers for the diskette drive connectors.

Pin	Signal	Pin	Signal
1	Ground	2	Data Rate Select 1
3	+5VDC	4	Drive Type ID 1
5	Ground	6	+12DC
7	Ground	8	-Index
9	Drive Type ID 0	10	Reserved
11	Ground	12	-Drive Select
13	Ground	14	Reserved
15	Ground	16	-Motor Enable
17	Media Type ID 1	18	-Direction In
19	Ground	20	-Step
21	Ground	22	-Write Data
23	Ground	24	-Write Enable
25	Ground	26	-Track 0
27	Media Type ID 0	28	-Write Protect
29	Ground	30	-Read Data
31	Ground	32	-Head 1 Select
33	Data Rate Select 0	34	-Diskette Change

Figure 3-1. Diskette Drive Connector

System Board Memory Connectors

The Model 90 system board has two connectors for riser cards. Each riser card provides four 72-pin memory connectors that support a variety of memory cards.

The following figure shows the location of the riser cards and memory connectors as viewed from the front of the system. The individual connectors are logically identified as A1 through A4 and B1 through B4; on both riser cards, these connectors are physically identified as J1 through J4, as shown in parentheses.

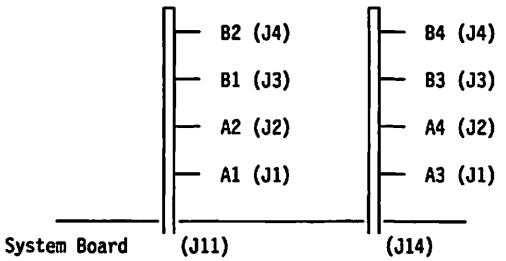


Figure 3-2. Memory Connector Numbering

The following figure shows the pin assignments for the 1- by 72-pin memory connectors. Refer to "Memory Presence Detect" on page 4-11 for decoding of the presence detect signals.

Pin	Signal	Pin	Signal
1	Ground	37	Parity Data 1
2	Data 0	38	Parity Data 3
3	Data 16	39	Ground
4	Data 1	40	Column Address Strobe 0
5	Data 17	41	Column Address Strobe 2
6	Data 2	42	Column Address Strobe 3
7	Data 18	43	Column Address Strobe 1
8	Data 3	44	Row Address Strobe 0
9	Data 19	45	Row Address Strobe 1
10	+5 Vdc	46	Block Select 1
11	-Column Address Strobe P	47	Write Enable
12	Address 0	48	Reserved
13	Address 1	49	Data 8
14	Address 2	50	Data 24
15	Address 3	51	Data 9
16	Address 4	52	Data 25
17	Address 5	53	Data 10
18	Address 6	54	Data 26
19	Reserved	55	Data 11
20	Data 4	56	Data 27
21	Data 20	57	Data 12
22	Data 5	58	Data 28
23	Data 21	59	+5 Vdc
24	Data 6	60	Data 29
25	Data 22	61	Data 13
26	Data 7	62	Data 30
27	Data 23	63	Data 14
28	Address 7	64	Data 31
29	Block Select 0	65	Data 15
30	+5 Vdc	66	Block Select 2
31	Address 8	67	Presence Detect 0
32	Address 9	68	Presence Detect 1
33	Row Address Strobe 3	69	Presence Detect 2
34	Row Address Strobe 2	70	Presence Detect 3
35	Parity Data 2	71	Block Select 3
36	Parity Data 0	72	Ground

Figure 3-3. System Board Memory Connector

Real-Time Clock/Complementary Metal-Oxide Semiconductor RAM

The real-time clock/complementary metal-oxide semiconductor RAM (RT/CMOS) chip contains the real-time clock and 64 bytes of CMOS RAM. The internal clock circuitry uses 14 bytes of this memory, and the rest is allocated for configuration and system-status information.

In addition to the 64 bytes of CMOS RAM, an 8KB CMOS RAM extension is provided for configuration and other system information.

A battery maintains voltage to the RT/CMOS RAM and 8KB CMOS RAM extension when the power supply is not in operation.

The system cover can be locked to prevent battery removal and, consequently, prevents the loss of password and configuration information.

The following table shows the RT/CMOS RAM bytes and their addresses.

Address (Hex)	RT/CMOS RAM Bytes
000–00D	Real-Time Clock Bytes
00E	Diagnostic Status Byte
00F	Shutdown Status Byte
010	Diskette Drive Type Byte
011	First Fixed Disk Drive Type Byte
012	Second Fixed Disk Drive Type Byte
013	Reserved
014	Equipment Byte
015, 016	Low and High Base Memory Bytes
017, 018	Low and High Expansion Memory Bytes
019–031	Reserved
032, 033	Configuration CRC Bytes
034	Reserved
035, 036	Low and High Usable Memory Above 1MB
037	Date Century Byte
038–3F	Reserved

Figure 3-4. RT/CMOS RAM Address Map

RT/CMOS Address Register and NMI Mask (Hex 0070)

This read/write register is used in conjunction with the port at hex 0071 to read and write the RT/CMOS RAM bytes.

Bit	Function
7	NMI Mask
6	Reserved
5–0	RT/CMOS RAM Address

Figure 3-5. RT/CMOS Address Register and NMI Mask (Hex 0070)

Warning: The operation following a write to hex 0070 must access port hex 0071; otherwise intermittent malfunctions and unreliable operation of the RT/CMOS RAM can occur.

Bit 7 When this bit is set to 1, the NMI is masked off (the NMI is disabled). This bit is set to 1 by a power-on reset. This is a write-only bit.

Bit 6 This bit is reserved.

Bits 5–0 These bits are used to select RT/CMOS RAM addresses.

RT/CMOS Data Register (Hex 0071)

This port is used in conjunction with the address register at hex 0070 to read and write the RT/CMOS RAM bytes.

Bit	Function
7–0	RT/CMOS Data

Figure 3-6. RT/CMOS Data Register (Hex 0071)

RT/CMOS RAM I/O Operations

During I/O operations to the RT/CMOS RAM addresses, interrupts must be masked to prevent other interrupt routines from changing the CMOS Address register before data is read or written.

Warning: The operation following a write to hex 0070 must access hex 0071; otherwise intermittent malfunctions and unreliable operation of the RT/CMOS RAM can occur.

Writing RT/CMOS RAM requires the following steps:

1. Write the RT/CMOS RAM address to the RT/CMOS and NMI Mask register (hex 0070).
2. Write the data to address hex 0071.
3. Write the address, hex 0D, to the RT/CMOS and NMI Mask register; this leaves hex 0070 pointing to Status Register D.
4. Read address hex 0071 to restore the RT/CMOS (see the warning above).

Reading RT/CMOS RAM requires the following steps:

1. Write the RT/CMOS RAM address to the RT/CMOS and NMI Mask register (hex 0070).
2. Read the data from address hex 0071.
3. Write the address, hex 0D, to the RT/CMOS and NMI Mask register; this leaves hex 0070 pointing to Status Register D.
4. Read address hex 0071 to restore the RT/CMOS.

Real-Time Clock Bytes (Hex 000 – 00D)

Bit definitions and addresses for the real-time clock bytes are shown in the following table.

Address (Hex)	Function	Byte Number
000	Seconds	0
001	Second Alarm	1
002	Minutes	2
003	Minute Alarm	3
004	Hours	4
005	Hour Alarm	5
006	Day of Week	6
007	Date of Month	7
008	Month	8
009	Year	9
00A	Status Register A	10
00B	Status Register B	11
00C	Status Register C	12
00D	Status Register D	13

Figure 3-7. Real-Time Clock Bytes

Note: The Setup program initializes status registers A, B, C, and D when the time and date are set. Interrupt hex 1A is the BIOS interface to read and set the time and date. It initializes the register the same way as the Setup program.

Status Register A (Hex 00A)

Bit	Function
7	Update in Progress
6–4	22-Stage Divider
3–0	Rate Selection Bits

Figure 3-8. Status Register A

Bit 7 When set to 1, this bit indicates the time-update cycle is in progress. When set to 0, it indicates the current date and time can be read.

Bits 6–4 These three divider-selection bits identify which time-base frequency is being used. The system initializes these bits to binary 010, which selects a 32.768 kHz time base. This is the only value supported by the system for proper time-keeping.

Bits 3–0 These bits allow the selection of a divider output frequency. The system initializes the rate selection bits to a binary 0110, which selects a 1.024 kHz square-wave output frequency and a 976.562-microsecond periodic interrupt rate.

Status Register B (Hex 00B)

Bit	Function
7	Set
6	Periodic Interrupt Enable
5	Alarm Interrupt Enable
4	Update-Ended Interrupt Enable
3	Square Wave Enable
2	Date Mode
1	24-Hour Mode
0	Daylight Savings Enable

Figure 3-9. Status Register B

Bit 7 When set to 0, this bit updates the cycle, normally by advancing the counts at a rate of one per second. When set to 1, this bit immediately ends any update cycle in progress, and the program can initialize the 14 time bytes without any further updates occurring until this bit is set to 0.

Bit 6 This bit is a read/write bit that allows an interrupt to occur at a rate specified by the rate and divider bits in Status Register A. When set to 1, this bit enables the interrupt. The system initializes this bit to 0.

Bit 5 When set to 1, this bit enables the alarm interrupt. The system initializes this bit to 0.

Bit 4 When set to 1, this bit enables the update-ended interrupt. The system initializes this bit to 0.

Bit 3 When set to 1, this bit enables the square-wave frequency as set by the rate-selection bits in Status Register A. The system initializes this bit to 0.

Bit 2 This bit indicates if the time-and-date calendar updates use binary or binary-coded-decimal (BCD) formats. When set to 1, this bit indicates a binary format. The system initializes this bit to 0.

Bit 1 This bit establishes if the hours byte is in the 24-hour or 12-hour mode. When set to 1, this bit indicates the 24-hour mode. The system initializes this bit to 1.

Bit 0 When set to 1, this bit enables the daylight savings time mode. When set to 0, it disables the mode, and the clock reverts to standard time. The system initializes this bit to 0.

Status Register C (Hex 00C)

Bit	Function
7	Interrupt Request Flag
6	Periodic Interrupt Flag
5	Alarm Interrupt Flag
4	Update-Ended Interrupt Flag
3–0	Reserved

Figure 3-10. Status Register C

Note: Interrupts are enabled by bits 6, 5, and 4 in Status Register B.

Bit 7 This bit is used in conjunction with bits 6, 5, and 4. When set to 1, this bit indicates that an interrupt has occurred; bits 6, 5, and 4 indicate the type of interrupt.

Bit 6 When set to 1, this bit indicates that a periodic interrupt occurred.

Bit 5 When set to 1, this bit indicates that an alarm interrupt occurred.

Bit 4 When set to 1, this bit indicates that an update-ended interrupt occurred.

Bits 3–0 These bits are reserved.

Status Register D (Hex 00D)

Bit	Function
7	Valid RAM
6–0	Reserved

Figure 3-11. Status Register D

Bit 7 This read-only bit monitors the power-sense pin. A low state of this pin indicates a loss of power to the real-time clock (dead battery). When set to 1, this bit indicates that the real-time clock has power. When set to 0, it indicates that the real-time clock has lost power.

Bits 6–0 These bits are reserved.

CMOS RAM Configuration

The following figure shows the bit definitions for the CMOS RAM configuration bytes.

Diagnostic Status Byte (Hex 00E)

Bit	Function
7	Real-Time Clock Chip Power
6	Configuration Record and Checksum Status
5	Incorrect Configuration
4	Memory Size Miscompare
3	Fixed Disk Controller/Drive C Initialization Status
2	Time Status Indicator
1	Adapter Configuration Miscompare
0	Adapter ID Time-Out

Figure 3-12. Diagnostic Status Byte

Bit 7 When set to 1, this bit indicates the real-time clock chip lost power.

Bit 6 When this bit is set to 1, the checksum is incorrect.

Bit 5 This is a check, at power-on time, of the Equipment byte. When set to 1, the configuration information is incorrect. Power-on checks require that at least one diskette drive be installed (bit 0 of the Equipment byte, hex 014, is set to 1).

Bit 4 When set to 1, this bit indicates the power-on check determined that the memory size is not the same as in the configuration record.

Bit 3 When set to 1, this bit indicates that the controller or drive C failed initialization, which prevents the system from attempting a power-on reset.

Bit 2 When set to 0, this bit indicates the time is valid. When set to 1, this bit indicates the time is invalid.

Bit 1 This bit indicates if the installed adapters match the configuration information. When this bit is set to 1, the adapters do not match the configuration information.

Bit 0 When set to 1, this bit indicates a time-out occurred while an adapter ID was being read.

Shutdown Status Byte (Hex 00F): This byte is defined by the power-on diagnostic programs.

Diskette Drive Type Byte (Hex 010): This byte indicates the type of diskette drive installed.

Bit	Function
7–4	First Diskette Drive Type
3–0	Second Diskette Drive Type

Figure 3-13. Diskette Drive Type Byte

Bits 7–4 These bits indicate the first diskette drive type, as shown in the following table.

Bits 7 6 5 4	Function
0 0 0 0	No Drive Present
0 0 0 1	Double-sided Diskette Drive (48 tracks per inch, 360KB)
0 0 1 0	Double-sided Diskette Drive (80 tracks per inch, 1.2MB)
0 0 1 1	High-capacity Diskette Drive (720KB)
0 1 0 0	High-density Diskette Drive (1.44MB)
All combinations that are not shown are reserved.	

Figure 3-14. Diskette Drive Type Byte (Bits 7–4)

Bits 3–0 These bits indicate the second diskette drive type, as shown in the following table.

Bits 3 2 1 0	Function
0 0 0 0	No Drive Present
0 0 0 1	Double-sided Diskette Drive (48 tracks per inch, 360KB)
0 0 1 0	Double-sided Diskette Drive (80 tracks per inch, 1.2MB)
0 0 1 1	High-capacity Diskette Drive (720KB)
0 1 0 0	High-density Diskette Drive (1.44MB)
All combinations that are not shown are reserved.	

Figure 3-15. Diskette Drive Type Byte (Bits 3–0)

First Fixed Disk Drive Type Byte (Hex 011): This byte defines the type of the first fixed disk drive (drive C). Hex 00 indicates that a fixed disk drive is *not* installed.

Second Fixed Disk Drive Type Byte (Hex 012): This byte defines the type of the second fixed disk drive (drive D). Hex 00 indicates that a fixed disk drive is *not* installed.

Note: For more information about fixed disk drive types, refer to the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference.*

Reserved Byte (Hex 013): This byte is reserved.

Equipment Byte (Hex 014): The equipment byte, for the power-on diagnostic tests, defines the basic equipment in the system.

Bit	Function
7, 6	Number of Diskette Drives
5, 4	Display Operating Mode
3, 2	Reserved
1	Math Coprocessor Presence
0	Diskette Drive Presence

Figure 3-16. Equipment Byte

Bits 7, 6 These bits indicate the number of diskette drives installed, as shown in the following table.

Bits 7 6	Number of Diskette Drives
0 0	One Drive
0 1	Two Drives
1 0	Three Drives
1 1	Reserved

Figure 3-17. Equipment Byte (Bits 7, 6)

Bits 5, 4 These bits indicate the operating mode of the display attached to the video port, as shown in the following table.

Bits 5 4	Display Operating Mode
0 0	Reserved
0 1	40-Column Mode
1 0	80-Column Mode
1 1	Monochrome Mode

Figure 3-18. Equipment Byte (Bits 5, 4)

Bits 3, 2 These bits are reserved.

Bit 1 When set to 1, this bit indicates that a math coprocessor is installed.

Bit 0 When set to 1, this bit indicates that a diskette drive is installed.

Low and High Base Memory Bytes (Hex 015 and 016): These bytes define the amount of memory below the 640KB address space.

The value from these bytes represents the number of 1KB blocks of base memory. For example, hex 0280 is equal to 640KB. The low byte is hex 15; the high byte is hex 16.

Low and High Expansion Memory Bytes (Hex 017 and 018): These bytes define the amount of memory above the 1MB address space.

The value from these bytes represents the number of 1KB blocks of expansion memory. For example, hex 0800 is equal to 2048KB. The low byte is hex 17; the high byte is hex 18.

Reserved Bytes (Hex 019 through 031): These bytes are reserved.

Configuration CRC Bytes (Hex 032 and 033): These bytes contain the cyclic-redundancy-check (CRC) data for bytes hex 010 through hex 031 of the 64-byte CMOS RAM. The low byte is hex 33; the high byte is hex 32.

Reserved Byte (Hex 034): This byte is reserved.

Low and High Useable Memory Bytes (Hex 035 and 036): These bytes define the total amount of contiguous useable memory from 1MB to 16MB.

The hexadecimal values in these bytes represent the number of 1KB blocks of useable memory. For example, hex 0800 is equal to 2048KB. The low byte is hex 35; the high byte is hex 36.

Date Century Byte (Hex 037): Bits 7 through 0 of this byte contain the BCD value for the century. Refer to the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference* for information about reading and setting this byte.

Reserved Bytes (Hex 038 through 03F): These bytes are reserved.

Miscellaneous System Functions

Nonmaskable Interrupt

The nonmaskable interrupt (NMI) signals the system microprocessor that a parity error, a channel check, a system channel time-out, or a system Watchdog time-out has occurred. The NMI stops all arbitration on the bus until bit 6 of the Arbitration register (I/O address hex 0090) is set to 0. This can result in lost data or an overrun error on some I/O devices. The NMI masks all other interrupts and the IRET instruction restores the interrupt flag to the state it was in prior to the interrupt. A system reset causes a reset of the NMI.

The NMI requests that are caused by channel check and system-board memory parity errors are subject to mask control with the NMI mask bit in the RT/CMOS Address register. The Watchdog Timer and system channel time-out are not masked by this bit. (See "RT/CMOS Address Register and NMI Mask (Hex 0070)" on page 3-6). The power-on default of the NMI mask is 1 (NMI disabled). Prior to enabling the NMI after a power-on reset, the parity check and channel check state are initialized by the POST.

Warning: The operation following a write to hex 0070 should access port hex 0071; otherwise intermittent malfunctions and unreliable operation of the RT/CMOS RAM can occur.

System Control Port B (Hex 0061)

Bit definitions for the read and write functions of this port are shown in the following tables.

Bit	Function
7	Reset Timer 0 Output Latch (IRQ0)
6–4	Reserved
3	Enable Channel Check
2	Enable Parity Check
1	Speaker Data Enable
0	Timer 2 Gate to Speaker

Figure 3-19. System Control Port B (Hex 0061) – Write

Bit	Function
7	Memory Parity Check
6	Channel Check
5	Timer 2 Output
4	Toggles with Each Refresh Request
3	Enable Channel Check
2	Enable Parity Check
1	Speaker Data Enable
0	Timer 2 Gate to Speaker

Figure 3-20. System Control Port B (Hex 0061) – Read

Bit 7 Setting this bit to 1 resets IRQ0. Reading this bit as a 1 indicates a memory parity check has occurred.

Bit 6 Reading this bit as a 1 indicates a channel check has occurred.

Bit 5 This bit indicates the condition of the timer/counter 2 'output' signal.

Bit 4 This bit toggles for each refresh request.

Bit 3 Setting this bit to 0 enables channel check. Setting this bit to 1 disables channel check and clears the channel check latch.

Bit 2 Setting this bit to 0 enables system board memory parity check. This bit is set to 1 during a power-on reset. To clear a parity error, set this bit to 1 and then to 0.

Bit 1 Setting this bit to 1 enables speaker data.

Bit 0 Setting this bit to 1 enables the timer 2 gate.

System Control Port A (Hex 0092)

Bit	Function
7–5	Reserved
4	Watchdog Timer Status
3	Security Lock Latch
2	Reserved = 0
1	Alternate Gate A20
0	Alternate Hot Reset

Figure 3-21. System Control Port A (Hex 0092)

Bits 7–5 These bits are reserved.

Bit 4 This read-only bit indicates the watchdog-timer status. When this bit is set to 1, a watchdog time-out has occurred. For more information about the Watchdog Timer, refer to System Timers in *Hardware Interface Technical Reference – Common Interfaces.*

Bit 3 This bit provides the security lock for the secured area of RT/CMOS. Setting this bit to 1 electrically locks the 8-byte, power-on password. Once this bit is set by POST, it can only be cleared by turning the system off.

Bit 2 This bit is reserved.

Bit 1 This bit is ORed with bit 1 in the keyboard output port to enable the 'address 20' signal (A20). When both bits are set to 0, A20 is disabled and set to 0. This bit is set to 0 during POST. (See the "Keyboard Controller" section in the *Hardware Interface Technical Reference.*)

Bit 0 This bit provides an alternate method of resetting the system microprocessor. This alternate method supports operating systems requiring faster operation than was provided on the IBM Personal Computer AT*. Resetting the system microprocessor is used to switch the microprocessor from the protected mode to the real address mode. The alternate reset takes 13.4 microseconds.

This bit is set to 0 either by a system reset or a Write operation. When a Write operation changes this bit from 0 to 1, the alternate reset pin is pulsed high for 100 to 125 nanoseconds. The reset occurs after a minimum delay of 6.72 microseconds. While the reset is occurring, the latch remains set so that POST can read this bit. If the bit is 0, POST assumes the system was just powered on. If the bit is 1, POST assumes a switch from the protected mode to the real mode has taken place.

---

* Personal Computer AT is a trademark of the International Business Machines Corporation.

Power-On Password

RT/CMOS RAM has eight bytes reserved for the power-on password and its check character. The eight bytes are initialized to hex 00. The microprocessor can only access these bytes during power-on self-test (POST). After POST is completed, if a power-on password is installed, the password bytes are locked and cannot be accessed by a program. A power-on password can be from 1 to 7 characters.

During power-on password installation, the password (1 to 7 keyboard scan codes), is stored in the security space.

Power-on password installation is a function of a program contained on the Reference Diskette. Once the power-on password utility has been installed, the password can be changed only during the POST. When the new power-on password is installed, changed, or removed, the password is not visible on the display.

The system unit cover can be physically locked to prevent unauthorized access to the battery. This helps prevent unauthorized battery removal and loss of power-on password and configuration information.

For information about the keyboard password, see the "Keyboard and Auxiliary Device Controller" section in the *Hardware Interface Technical Reference.*

Hardware Compatibility

The Model 90 maintains many of the interfaces used by the IBM Personal Computer AT. In most cases command and status organization of these interfaces is maintained.

The functional interfaces for the Model 90 are compatible with the following interfaces:

- The Intel** 8259 interrupt controllers (without edge triggering).
- The Intel 8253 timers driven from 1.193 MHz (timer 0 and 2 only).
- The Intel 8237 DMA controller address/transfer counters, page registers and status fields only. The Command and Request registers are not supported. The rotate and mask functions are not supported. The Mode register is partially supported.
- The NS16450 serial port.
- The Intel 8088, 8086, 80286, and 80386 microprocessors.
- The Intel 8272 diskette drive controller.
- The Motorola** MC146818A Time of Day Clock command and status (CMOS reorganized).
- The keyboard interface at address hex 0060.
- Display modes supported by the IBM Monochrome Display and Printer Adapter, the IBM Color/Graphics Monitor Adapter, and the IBM Enhanced Graphics Adapter.
- The parallel printer ports (Parallel 1, Parallel 2, and Parallel 3) in compatibility mode.
- Generally compatible with the Intel 8087, 80287, and 80387 math coprocessors.

Error Codes

POST returns a message in the form of a multiple-character code to indicate the type of test that failed. Two formats are used for these error messages:

- An eight-character error code for all errors not related to SCSI devices
- An 11-character code for all errors related to SCSI devices
  - The first three characters identify the device type
  - The next eight characters identify specifics of the error.

The eight-character error code consists of two 4-character elements, the major error code and the minor error code. The following table gives the failure indicated with the associated error code.

Note: For information on unlisted error codes, contact Developer Assistance at 1-800-IBM-7763 (this number is for developers who are registered with IBM).

Major Code	Minor Code	Description
0001		System Bus Error - system board or processor complex
	0100	Unexpected interrupt
	0200	Timer failure
	0300	Timer interrupt failure
	0400	Protected mode failure
	0500	Last keyboard command not accepted
	0600	Converting logic test
	0700	NMI test failed
	0800	Timer bus-test failed
	10xx	System board memory parity error
	11xx	I/O channel check error
	1200	Watchdog timeout
	1300	DMA arbitration timeout
	1400	Channel feature card ROM checksum error
	1500	ROM checksum error or DMA error
	1600	System board port read/write failure
	1800	System board parity or L2-cache error during previous power-on
	2000	Microprocessor test error
	6100	Dead battery
	6300	Clock not updating
	6400	Memory configuration error
	6500	Adapter ID mismatch
	6600	Adapter busy error
	6700	Clock not updating
	6900	Processor complex configuration error
	7000	ASCII setup conflict error
	7100	Rolling bit test failure on CMOS shutdown address byte
	7200	Rolling bit test failure on NVRAM diagnostic byte
	7300	Bad CMOS/NVRAM checksum
	7400	Bad configuration

Figure 3-22 (Part 1 of 2). POST Error Message Table

Major Code	Minor Code	Description
0002		Memory errors
	0xxx	Memory error
	1xxx	Unrecoverable error in first 1MB
	2100	ROM to RAM remap error
	25xx	Unsupported memory-type installed or memory pair mismatch
0003		Keyboard
	0100	Keyboard error
	0200	Keyboard locked
	0300	Keyboard to system board interface error
	0400	Keyboard clock high
	0500	No keyboard +5V
0004	0100	Display error
0005	0100	Display error
0006		Diskette
	0100	Diskette drive or controller error
	0200	Diskette IPL boot record not valid
	0604	Non-media sense
0011	0100	Async chip error
0017		Fixed disk error
	8000	Fixed disk 0 failed
	8100	Fixed disk 1 failed
	8200	Disk controller error
	9000	Hard file 0 error
	9100	Hard file 1 error
0024	0100	System board video error
0086		Mouse
	0100	System bus error - keyboard/pointing device interface
	0200	Pointing device error
	0300	Pointing device or system bus error
0129		Cache
	0x00	Cache error
1999		Special instruction codes
	00xx	Initial microcode load (IML) error
	03xx	No bootable device
	04xx	IML to system mismatch

Figure 3-22 (Part 2 of 2). POST Error Message Table

Notes:

Section 4. Processor Complex

Type 1 4-3
Type 2 4-5
I/O Address Map 4-7
Optional Cache 4-7
Micro Channel 4-7
  Central Arbiter 4-8
    Arbitration Bus Priority Assignments 4-8
  Central Arbiter Programming 4-9
Memory Subsystem 4-11
  Memory Presence Detect 4-11
  Memory Controller Registers 4-12
    Memory Bank Base Address Register (Index Hex 90—9F) 4-15
    Split Address Register (Index Hex A0) 4-15
    Memory Encoding Register (Index Hex A1) 4-16
    Cache/Timer Control Register (Index Hex A2) 4-18
    Boundary 0 Cache Control Register (Index Hex A3) 4-19
    Boundary 1 Cache Control Registers (Index Hex A4 and A5) 4-20
    Memory Bank Enable Register 1 (Index Hex A6) 4-21
    Memory Bank Enable Register 2 (Index Hex A7) 4-22
    System Bus Request Timer Register (Index Hex A8) 4-23
    System Bus Hold Timer Register (Index Hex A9) 4-24
    System Bus Idle Timer Register (Index Hex AA) 4-24
    Parity Trap Register (Hex 00E1) 4-25
    Cache Control Register (Hex 00E2) 4-26
    Cache Status Register (Hex 00E3) 4-27
Processor Cache Address Map 4-28
  Read-Only Memory Subsystem 4-29

© Copyright IBM Corp. 1990

The processor complex provides certain devices and features that work in conjunction with those devices and features provided by the system board. In addition, the processor complex determines system performance and certain channel timings. This section describes these aspects of the processor complex.

A system can have either a Type 1 or Type 2 processor complex. The major difference between the two types is speed. Programs can identify the type by reading the model and submodel bytes and the BIOS revision level (Interrupt hex 15, function code AH = hex C0). The following table shows these bytes and the processor complex types.

Model Byte	Submodel Byte	Revision Level	Speed	Processor Complex
F8	11	00	25 MHz	Type 1
F8	13	00	33 MHz	Type 2

Figure 4-1. Model and Submodel Bytes

Refer to the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference* for a listing of other systems, and check the supplements section for updates to that listing.

Type 1

The following are the devices and features for the Type 1 processor complex.

Device	Type	Features
Microprocessor		80486 – 25 MHz 32-bit address and 32-bit data interface
Cache	—	8KB, 4-way, write-through, internal cache Optional L2 cache card interface
ROM Subsystem	—	128KB physical ROM
RAM Subsystem	—	Eight connectors support 4 to 32MB, interleaved Expandable on the system board only
DMA Controller	1	Eight independent DMA channels Single or burst transfers and read verification
Math Coprocessor	—	Math coprocessor functions included in 80486

Figure 4-2. Type 1 Processor Complex Feature and Devices

The following shows the performance and other timing information for the Type 1 processor complex. These timings affect the channel performance for operations involving the processor complex or system board memory.

Device	Cycle Time (ns)
Microprocessor (25 MHz – 40 ns Clock):	
Access to Channel:	
Default Transfer Cycle	240
Extended Transfer Cycle	320
Refresh Rate (Typically performed every 15.1 μs)	500 (min)
DMA Controller (10 MHz – 100 ns Clock):	
Single Transfer: 300 + I/O Access + Memory Access	
Burst Transfers: 300 + (I/O Access + Memory Access)N *	
System Board Memory Access	300
Default Transfer Cycle	200
Synchronous Extended Transfer Cycle	300
* N is the number of transfers in the burst transfer.	

Figure 4-3. System Performance Specifications, Type 1 Processor Complex

Device	Cycle Time (ns) per Memory Type		
	70 ns	80 ns	85 ns
Microprocessor (25 MHz – 40 ns Clock):			
Memory Read (Cache Hit)	40	40	40
Memory Read (Cache Miss, Page Hit, Burst)	320	320	400
Memory Read (Cache Miss, Page Miss, Burst)	440	480	560
Memory Write (Page Hit)	160	160	200
Memory Write (Page Miss)	280	320	360
Microprocessor – with Optional L2 Cache			
Memory Read (Cache Hit, Optional Cache Hit)	40	40	40
Memory Read (Cache Miss, Optional Cache Hit, Burst)	200	200	200
Memory Read (Cache Miss, Optional Cache Miss, Page Hit, Burst)	320	320	400
Memory Read (Cache Miss, Optional Cache Card Miss, Page Miss, Burst)	440	480	560
Memory (Page Hit)	160	160	200
Memory Write (Page Miss)	280	320	360
Bus Master Access to System Board RAM			
Page Hit (minimum)	300	300	300
Page Miss (minimum)	300	300	300

Figure 4-4. Memory Performance Specifications, Type 1 Processor Complex

Type 2

The following are the devices and features for the Type 2 processor complex.

Device	Type	Features
Microprocessor		80486—33 MHz 32-bit address and 32-bit data interface
Cache	—	8KB, 4-way, write-through, internal cache Optional cache card interface
ROM Subsystem	—	128KB physical ROM
RAM Subsystem	—	Eight connector support 4 to 32MB interleaved Expandable on the system board only
DMA Controller	1	Eight independent DMA channels Single or burst transfers and read verification
Math Coprocessor	—	Math coprocessor functions included in 80486

Figure 4-5. Type 2 Processor Complex Feature and Devices

The following shows the performance and other timing information for the Type 2 processor complex. These timings affect the channel performance for operations involving the processor complex or system board memory.

Device			Cycle Time (ns)
Microprocessor (33 MHz — 30 ns Clock):			
Access to Channel:			
Default Transfer Cycle			240
Extended Transfer Cycle			320
Refresh Rate (Typically performed every 15.1 μs)			500 (min)
DMA Controller (10 MHz — 100 ns Clock):			
Single Transfer:	300 + I/O Access + Memory Access		
Burst Transfers:	300 + (I/O Access + Memory Access)N *		
System Board Memory Access		300	
Default Transfer Cycle		200	
Synchronous Extended Transfer Cycle		300	
* N is the number of transfers in the burst transfer.			

Figure 4-6. System Performance Specifications, Type 2 Processor Complex

Device	Cycle Time (ns) per Memory Type		
	70 ns	80 ns	85 ns
Microprocessor (33 MHz – 30 ns Clock):			
Memory Read (Cache Hit)	30	30	30
Memory Read (Cache Miss, Page Hit, Burst)	240	300	300
Memory Read (Cache Miss, Page Miss, Burst)	360	420	420
Memory Write (Page Hit)	120	150	150
Memory Write (Page Miss)	240	270	270
Microprocessor – with Optional L2 Cache			
Memory Read (Cache Hit, Optional Cache Hit)	30	30	30
Memory Read (Cache Miss, Optional Cache Hit, Burst)	150	150	150
Memory Read (Cache Miss, Optional Cache Miss, Page Hit, Burst)	240	300	300
Memory Read (Cache Miss, Optional Cache Card Miss, Page Miss, Burst)	360	420	420
Memory Write (Page Hit)	120	150	150
Memory Write (Page Miss)	240	270	270
Bus Master Access to System Board RAM			
Page Hit (minimum)	300	300	300
Page Miss (minimum)	300	300	300

Figure 4-7. Memory Performance Specifications, Type 2 Processor Complex

I/O Address Map

The following shows the I/O address for devices located on the processor complex.

Hex Addresses	Device
0000–001F	DMA Controller (0–3)
0081–0083, 0087	DMA Page Registers (0–3)
0089–008B, 008F	DMA Page Registers (4–7)
0090	Central Arbitration Control Point
00C0–00DF	DMA Controller (4–7)
00E0	Memory Control Register Select
00E1	Parity Trap Register
00E2	Cache Control Register
00E3	Cache Status Register
00E4	Memory Control Data Read/Write

Figure 4-8. Processor Complex I/O Address Map

Optional Cache

The Type 1 and Type 2 processor complexes have a 160-pin connector for an optional cache card (L2 cache). The optional cache card provides secondary cache and cache controller for the system microprocessor.

The optional cache card provides the system microprocessor with an additional 256KB cache. It operates as 2-way set-associative, write-through cache with parity. The primary cache in the 80486 must be enabled to operate the secondary cache. The L2 cache is controlled through the Cache Control Register (Hex 00E2), and its status is indicated in the Cache Status Register (Hex 00E3).

Micro Channel

This section describes the implementation of the Micro Channel architecture on Model 90 systems. For general Micro Channel information, refer to *Hardware Interface Technical Reference — Architectures*.

Note: The system does not support adapters that use the matched memory extension.

Central Arbiter

The central arbitration control point gives intelligent subsystems on the channel the ability to share and control the system. It allows burst data transfers and prioritization of control between devices. This arbiter supports up to 16 arbitrating devices.

Arbitration Bus Priority Assignments

The following table shows the assignment of arbitration levels. The functions with the lowest arbitration level have the highest priority.

ARB Level	Primary Assignment
-2	Memory Refresh
-1	NMI
0	DMA Channel 0 (Programmable to any arbitration level)
1	DMA Channel 1
2	DMA Channel 2
3	DMA Channel 3
4	DMA Channel 4 (Programmable to any arbitration level)
5	DMA Channel 5
6	DMA Channel 6
7	DMA Channel 7
8–B	Available
C	SCSI Adapter
D, E	Available
F	System Microprocessor

Figure 4-9. Arbitration Bus Priority Assignments

Note: Devices designed for arbitration level 0 or 1 should have limited bandwidth or short bursts so diskette overruns can be prevented or recovered by retry operations. The diskette drive controller, on arbitration level 2, can be held inactive by devices on levels 0 and 1, by a refresh operation, and by the previous controlling master. The diskette drive controller should not be held inactive for more than 12 microseconds to prevent overrun.

Nonmaskable interrupt (NMI) service is executed at a priority level higher than 0, called -1. Memory refresh is prioritized at -2, two levels higher than 0. Levels -1 and -2 are reached on the system board only, while the 'arbitrate/-grant' signal is in the arbitrate state.

When the central arbitration control point receives a level -1 request (NMI, a system-board internal signal), it activates -PREEMPT, waits for the end of transfer, and then places ARB/-GNT in the arbitrate state, which denies channel activity to arbitrating devices. The central arbitration control point gives the grant to the level -1 request, and holds ARB/-GNT in the arbitrate state until the operation is complete and the NMI is reset.

Central Arbiter Programming

The central arbitration control point provides access to programmable options through the Arbitration register, which is accessed at I/O address hex 0090. The bits are defined differently for read and write operations, as shown in the following tables.

Bit	Definition
7	Enable System Microprocessor Cycle
6	Arbitration Mask
5	Enable Extended Arbitration
4–0	Reserved

Figure 4-10. Arbitration Register (Hex 0090) – Write

Bit	Definition
7	Enable System Microprocessor Cycle
6	Arbitration Masked by NMI
5	Bus Time-out
4	Reserved
3–0	Value of Arbitration Bus During Previous Grant State

Figure 4-11. Arbitration Register (Hex 0090) – Read

Bit 7 Setting this bit to 1 enables system microprocessor cycles during arbitration cycles. This bit can be set to 0 if an arbitrating device requires total control of the channel bandwidth. This bit is set to 0 by a system reset.

Reading this bit as a 1 indicates system microprocessor cycles are enabled during arbitration.

Bit 6 Setting this bit to 1 causes the central arbitration control point to enter the arbitration state. The system microprocessor controls the channel until this bit is reset to 0. This bit is set to 0 by a system reset.

Reading this bit as a 1 indicates that an NMI has occurred and has masked arbitration.

Warning: This bit should be set to 1 only by diagnostic routines and system error-recovery routines.

Bit 5 Setting this bit to 1 enables extended arbitration. The minimum arbitration cycle is 300 nanoseconds; this bit extends that minimum cycle to 600 nanoseconds. This bit is set to 0 during a system reset.

Reading this bit as a 1 indicates that a bus time-out has occurred, and resets bit 6 in this register to 0.

Bit 4 This bit is reserved and should be 0.

Bits 3–0 These bits are undefined for a write operation and should be set to 0.

Reading these bits returns the arbitration level of the arbiter controlling the channel during the most recent grant state. This information allows the system microprocessor to determine the arbitration level of the device that caused a bus time-out.

Memory Subsystem

Memory on the system board is parity-checked, interleaved random-access memory and must be installed in matched pairs: one in an A connector and one in the matching B connector (for memory connector information, see "System Board Memory Connectors" on page 3-3). The memory in each pair must be the same speed and size. Memory on adapters in the channel are not mapped into the cacheable address space of the system microprocessor.

System board memory is designed so that the system microprocessor can access memory while a bus master has control of the channel. The memory controller manages this dual access by interleaving the microprocessor and bus master accesses. In addition, system board memory is organized into pairs of banks so that the microprocessor can perform burst cycles with two-way interleaved accesses. Because of this organization, the microprocessor can perform zero-wait-state transfers for every other 32-bit transfer in the burst cycle.

Memory Presence Detect

The presence-detect signals are used by the system to determine size and speed of memory on the memory card. The pins are either not connected (1), or connected to ground (0).

A set of four bits shows the state of all presence-detect signals for a specific memory connector; the bit is 0 when the signal is connected to ground. The following figure shows the information indicated by these bits as well as the bank size of the memory card. All combinations not shown are reserved.

Presence Detect 3 2 1 0	Memory Card Definition	Bank Size
0 1 0 0	4MB 70-ns Memory	4MB
0 0 0 0	4MB 80-ns Memory	4MB
1 1 0 0	2MB 70-ns Memory	1MB
1 0 0 1	2MB 80-ns Memory	1MB
0 1 0 1	2MB 85-ns Memory	1MB
0 1 1 0	1MB 85-ns Memory	1MB
1 1 1 1	Connector is Empty	

Figure 4-12. Presence-Detect Bits

Memory Controller Registers

The memory controller, located on the processor complex, has several internal registers that are used to control the memory in the memory connectors on the system board. The memory controller manages this memory in banks of 1MB or 4MB. The size of the bank determines its granularity.

Connectors A1 through A4 contain banks 0 through 7; connectors B1 through B4 contain banks 8 through hex F.

Each internal register is accessed by writing its internal address (index) to the Memory Controller Address register at hex 00E0 and then accessing the data through the Memory Controller Data register at hex 00E4. These registers can be accessed only from the system microprocessor.

I/O port 00E2 controls the internal 80486 cache and the optional cache functions. I/O port 00E3 is used for cache status and memory bank parity error determination. I/O port 00E1 is also used for memory bank parity error determination.

All registers in the memory controller are 8-bit read/write registers. The following shows the function and index for each internal register. These registers are programmed during POST to configure the memory controller according to the type and amount of memory and other system requirements.

Index	Register Function
90	Memory Bank 0 Base Address
91	Memory Bank 1 Base Address
92	Memory Bank 2 Base Address
93	Memory Bank 3 Base Address
94	Memory Bank 4 Base Address
95	Memory Bank 5 Base Address
96	Memory Bank 6 Base Address
97	Memory Bank 7 Base Address
98	Memory Bank 8 Base Address
99	Memory Bank 9 Base Address
9A	Memory Bank A Base Address
9B	Memory Bank B Base Address
9C	Memory Bank C Base Address
9D	Memory Bank D Base Address
9E	Memory Bank E Base Address
9F	Memory Bank F Base Address
A0	Split Address
A1	Memory Encoding
A2	Cache/Timer Control
A3	Boundary 0 Cache Control
A4	Boundary 1 Cache Control (Low Address)
A5	Boundary 1 Cache Control (High Address)
A6	Memory Bank Enable 1
A7	Memory Bank Enable 2
A8	System Bus Request Timer
A9	System Bus Hold Timer
AA	System Bus Idle Timer

Figure 4-13. Memory Controller Register Assignments

The 16 Memory Bank Base Address registers and individual bits in the two Bank Enable registers control the enabling and mapping of all memory on the system board. Individual bits in the Parity Trap register (hex 00E1) indicate whether the respective bank has detected a parity error.

The following figure shows the relationship of each of the control or status registers to the bank of memory that it represents. The error-indicator bit is in the Parity Trap register (hex 00E1).

Bank Number	Index of Memory Bank Address Register	Memory Bank Enable Register and Bit	Error Indicator Bit
0	90	Register 1, Bit 0	0
1	91	Register 1, Bit 1	1
2	92	Register 1, Bit 2	2
3	93	Register 1, Bit 3	3
4	94	Register 1, Bit 4	4
5	95	Register 1, Bit 5	5
6	96	Register 1, Bit 6	6
7	97	Register 1, Bit 7	7
8	98	Register 2, Bit 0	0
9	99	Register 2, Bit 1	1
A	9A	Register 2, Bit 2	2
B	9B	Register 2, Bit 3	3
C	9C	Register 2, Bit 4	4
D	9D	Register 2, Bit 5	5
E	9E	Register 2, Bit 6	6
F	9F	Register 2, Bit 7	7

Figure 4-14. Memory Control

Memory Bank Base Address Register (Index Hex 90 – 9F)

Each bank of memory has an 8-bit, read/write register that determines the base address for that bank. The banks can be set to respond to any starting 1MB address from 0 to 255MB. 4MB memory cards must be on 4MB address boundaries.

The 16 Base Address registers are at indexes hex 90 to 9F.

Bit	Function
7	Address Bit A27
6	Address Bit A26
5	Address Bit A25
4	Address Bit A24
3	Address Bit A23
2	Address Bit A22
1	Address Bit A21
0	Address Bit A20

Figure 4-15. Memory Bank Base Address Register (Index Hex 90 – 9F)

Split Address Register (Index Hex A0)

This 8-bit, read/write register at index hex A0 determines the starting address for the split memory block. The split-memory block can be positioned at any 1MB boundary from 1MB to 255MB; it must not be positioned below 1MB.

The bit definitions and functions are described below:

Bit	Function
7	Split Address Bit A27
6	Split Address Bit A26
5	Split Address Bit A25
4	Split Address Bit A24
3	Split Address Bit A23
2	Split Address Bit A22
1	Split Address Bit A21
0	Split Address Bit A20

Figure 4-16. Split Address Register (Index Hex A0)

Bits 7 – 0 These bits define the starting address of the split-memory block. When split memory is enabled, these bits determine its starting address. The starting location can be at any 1MB boundary from 1MB to 255MB. This register must not be set to 00.

Memory Encoding Register (Index Hex A1)

This 8-bit, read/write register at index hex A1 controls miscellaneous functions within the memory controller.

Bit	Function
7	-System Bus Enable
6	-Reserved
5	-Disable ROM Space Decode
4	-Lock
3	-Enable Split
2	-640
1	ROM enable
0	-Enable Planar Parity Check

Figure 4-17. Memory Encoding Register (Index Hex A1)

Bit 7 This bit is used to isolate system-board memory from the channel during initialization of the memory controller. When the bit is set to 1, channel accesses to system board memory are disabled and the memory controller ignores all channel requests. When the bit is set to 0, the channel is enabled.

Bit 6 This bit is reserved.

Bit 5 This bit controls the memory-controller decode of the first 4096 locations in system ROM space from 000E0000 to 000E0FFF. When the bit is set to 0, the decode of these addresses is disabled; when set to 1, the memory controller selects ROM or system RAM for all addresses from hex 000E0000 to 000FFFFF and from hex FFFE0000 to FFFFFFFF.

While this bit is 0, accesses to addresses hex 000E0000 to 000E0FFF are not directed to the space allocated for system ROM code and can be decoded on the channel.

Bit 4 This bit determines whether locked-access operations from the system microprocessor are enabled. When this bit is set to 0, lock is enabled and microprocessor cycles are performed as follows:

- When the system microprocessor owns the channel and is accessing memory with the 'lock' signal active, the microprocessor must complete the entire operation before another master can be granted control of the channel. This includes the DMA controller.

- When a bus-master owns the channel and the microprocessor tries to run a locked cycle operation, the microprocessor waits until the bus master releases the channel.

Bit 3 This bit determines if the split-memory block is assigned addresses or is disabled. The top 128KB of the first 1MB is always mapped into the ROM address space and the code in ROM is copied into this space.

When this bit is set to 0, the split-memory block is enabled. The split-memory block size is dependent on the value of the 640 bit. Its address is determined in Memory Encoding Register 2. When this bit is set to 1, the split-memory block is disabled.

Bit 2 This bit determines where the first active 1MB of memory is split. When this bit is set to 0, the system maps 640KB of the first 1MB to addresses hex 00000000 to 0009FFFF. The split-memory block, 256KB, is mapped to the address specified in the Split Address register.

When this bit is set to 1, the system maps 512KB of the first 1MB to addresses hex 00000000 to 0007FFFF. The split-memory block, 384KB, is mapped to the address specified in Split Address register. The addresses hex 00080000 to 0009FFFF are unassigned.

Note: The memory address space from the memory split to 896KB is not cached.

Bit 1 This bit determines how addresses hex 000E0000 to 000FFFFF are assigned.

When this bit is set to 1, ROM is enabled and the read-access addresses are assigned to ROM; the write-access addresses are assigned to RAM. When this bit is set to 0, ROM is disabled and read accesses are assigned to RAM; write accesses are disabled while ROM is disabled.

Note: This bit should be set to 0 because system performance can be degraded when ROM is enabled.

Bit 0 When set to 0, this bit enables parity checking of system board memory. Setting this bit to 1 clears the pending parity error. For compatibility with other systems, enabling and disabling of parity checking should also be done through System Control Port B at address hex 0061

(see the register definition on "System Control Port B (Hex 0061)" on page 3-17).

Note: When a parity error occurs, clear the error using this bit then clear the error using the bit in System Control Port B.

Cache/Timer Control Register (Index Hex A2)

This 8-bit, read/write register at index hex A2 controls the cache and timer functions.

Bit	Function
7	-Enable Processor Preempt Timers
6–3	Reserved
2	-Enable Boundary 1 Check
1	-Enable Boundary 0 Check
0	-Enable Cache

Figure 4-18. Cache/Timer Control Register (Index Hex A2)

Bit 7 This bit controls the timers used to allocate bus cycles to the microprocessor. In system environments with several bus-master-type adapters, the system microprocessor can be prevented from gaining control of the channel for long periods of time. When this bit is set to 0, the system-bus timer registers (index hex A8 through AA) are enabled, and the system microprocessor can be ensured a percentage of the channel bandwidth.

Bits 6–3 Reserved

Bit 2 This bit controls the boundary check for a noncacheable address from 16MB to 256MB. When the bit is set to 0, boundary checks are enabled. For detail, see "Boundary 1 Cache Control Registers (Index Hex A4 and A5)" on page 4-20.

Bit 1 This bit controls the boundary check for a noncacheable address below 16MB. When the bit is set to 0, boundary checks are enabled. For detail, see "Boundary 0 Cache Control Register (Index Hex A3)" on page 4-19.

Bit 0 This bit enables or disables the internal cache in the system microprocessor and the external cache, if it is installed. When this bit is set to 0, both caches are enabled. When this bit is set to 1, both caches are disabled.

After disabling the caches and before enabling them, they should be flushed (see "Cache Control Register (Hex 00E2)" on page 4-26).

Boundary 0 Cache Control Register (Index Hex A3)

This 8-bit, read/write register at index hex A3 contains the upper-range and lower-range control bits. This register determines the memory range, from 1MB up to 16MB, that will not be cached.

When the cache is enabled, memory below 1MB is cached except for the following:

- The addresses from hex 000A0000 to 000DFFFF are never cached.
- The addresses from hex 000E0000 to 000FFFFF are not cached when ROM is enabled, and are cached when ROM is disabled.
- The addresses from hex 00080000 to 000DFFFF are not cached when the memory split is at 512KB.

The range check for noncacheable addresses is enabled by bit 1 in the Cache/Timer Control register. When the boundary 0 range check is disabled, all addresses from 1MB to 16MB are cacheable and the addresses are not checked. When the range check is enabled, the memory controller checks the range of the memory address to determine if it is cacheable. If the address is equal to or greater than the lower boundary limit *and* equal to or less than the upper boundary limit, it is considered to be noncacheable.

Bit	Function
7–4	Upper Limit for Boundary 0
3–0	Lower Limit for Boundary 0

Figure 4-19. Boundary 0 Cache Control Register (Index Hex A3)

Bits 7–4 These bits contain the upper boundary limit and are compared to address bits A23 through A20 (address bits A24 thru A31 must be 0).

Bits 3–0 These bits contain the lower boundary limit and are compared to address bits A23–A20 (address bits A24 thru A31 must be 0).

Boundary 1 Cache Control Registers (Index Hex A4 and A5)

These two registers are 8-bit, read/write registers at index hex A4 and A5. The lower-boundary register (index hex A4) contains the lower-range control bits and the upper-boundary register (index hex A5) contains the upper-range control bits. The upper-range and lower-range control bits determine the noncacheable addresses in the memory space from 16MB to 256MB.

When the cache is enabled, memory below 1MB is cached except for the following:

- The addresses from hex 000A0000 to 000DFFFF are never cached.
- The addresses from hex 000E0000 to 000FFFFF are not cached when ROM is enabled, and are cached when ROM is disabled.
- The addresses from hex 00080000 to 000DFFFF are not cached when the memory split is at 512KB.

The range check for noncacheable addresses is enabled by bit 2 in the Cache/Timer Control register. When the boundary 1 range check is disabled, all addresses from 16MB to 256MB are cacheable and the addresses are not checked. When the range check is enabled, the memory controller checks the range of the memory address to determine if it is cacheable. If the address is equal to or greater than the lower boundary limit *and* equal to or less than the upper boundary limit, it is considered to be noncacheable.

The Lower Boundary 1 Cache Control register contains the lower boundary limit and is compared to address bits A27 through A20 (address bits A28 through A31 must be 0).

The Upper Boundary 1 Cache Control register contains the upper boundary and is compared to address bits A27 through A20 (address bits A28 through A31 must be 0). When the boundary includes the first 1MB of memory, any area that normally would be cacheable (for example hex 0E0000 to 0FFFFF) will be noncacheable.

Bit	Function
7	Lower Limit Boundary A27
6	Lower Limit Boundary A26
5	Lower Limit Boundary A25
4	Lower Limit Boundary A24
3	Lower Limit Boundary A23
2	Lower Limit Boundary A22
1	Lower Limit Boundary A21
0	Lower Limit Boundary A20

Figure 4-20. Lower Boundary 1 Cache Control Register (Index Hex A4)

Bit	Function
7	Upper Limit Boundary A27
6	Upper Limit Boundary A26
5	Upper Limit Boundary A25
4	Upper Limit Boundary A24
3	Upper Limit Boundary A23
2	Upper Limit Boundary A22
1	Upper Limit Boundary A21
0	Upper Limit Boundary A20

Figure 4-21. Upper Boundary 1 Cache Control Register (Index Hex A5)

Memory Bank Enable Register 1 (Index Hex A6)

This 8-bit, read/write register at index A6 controls the enabling of the first eight banks of memory in the system-board memory connectors A1 through A4.

Bit	Function
7	Disable Bank 7 (Port 97)
6	Disable Bank 6 (Port 96)
5	Disable Bank 5 (Port 95)
4	Disable Bank 4 (Port 94)
3	Disable Bank 3 (Port 93)
2	Disable Bank 2 (Port 92)
1	Disable Bank 1 (Port 91)
0	Disable Bank 0 (Port 90)

Figure 4-22. Memory Bank Enable Register 1 (Index Hex A6)

Bits 7–0 Each bit controls the enabling of memory for the specified bank. When the bit is set to 1, the bank is disabled. When the bit is 0, the bank is enabled and responds to the address specified in its Base Address register.

Note: Because memory must be enabled in pairs, corresponding banks should be enabled in Memory Bank Enable Register 2. These two registers should be set to the same value.

Memory Bank Enable Register 2 (Index Hex A7)

This 8-bit, read/write register at index hex A7 controls the enabling of the second 8 banks of memory in the system-board memory connectors B1 through B4.

Bit	Function
7	Disable Bank F (Port 9F)
6	Disable Bank E (Port 9E)
5	Disable Bank D (Port 9D)
4	Disable Bank C (Port 9C)
3	Disable Bank B (Port 9B)
2	Disable Bank A (Port 9A)
1	Disable Bank 9 (Port 99)
0	Disable Bank 8 (Port 98)

Figure 4-23. Memory Bank Enable Register 2 (Index Hex A7)

Bits 7–0 Each bit controls the enabling of memory for the specified bank. When the bit is set to 1, the bank is disabled. When the bit is 0, the bank is enabled and responds to the address specified in its Base Address register.

Note: Because memory must be enabled in pairs, corresponding banks should be enabled in Memory Bank Enable Register 1. These two register should be set to the same value.

System Bus Request Timer Register (Index Hex A8)

The System Bus Request Timer Register, the System Bus Hold Timer Register, and the System Bus Idle Timer Register work together to determine the percentage of time that the system microprocessor owns the channel. Each register is an 8-bit read/write register and they are active only when bit 7 of the Cache/Timer Control register is set to 1.

The values in these three registers are moved to counters, which are decremented for every other clock cycle of the system microprocessor.

For the Type 1 Processor Complex, each count represents 80 nanoseconds. For the Type 2 Processor Complex, each count represents 60 nanoseconds.

The System Bus Request Timer register at index hex A8 determines how long the system microprocessor will wait before preempting for use of the channel.

The value in this register is moved to a counter when the system microprocessor loses ownership of the channel. Decrementing the counter begins when the system microprocessor wants to use the channel and another master owns it.

If the count reaches 0 and the system microprocessor has not yet gained control of the channel, -PREEMPT is driven active.

System Bus Hold Timer Register (Index Hex A9)

This register determines the maximum amount of time that the system microprocessor maintains ownership of the channel.

The value in this register is moved to a counter when the system microprocessor loses ownership of the channel. Decrementing the counter begins as soon as the system microprocessor is granted ownership of the channel.

If -PREEMPT was driven active because of the System Bus Request Timer register the system microprocessor still owns the channel when the count reaches 0, the system microprocessor gives up ownership of the channel.

Note: If the system microprocessor gained control of the channel without driving -PREEMPT active, the channel is given up when another device requests the channel.)

System Bus Idle Timer Register (Index Hex AA)

This register determines the maximum amount of time that system microprocessor will retain control of the channel without using it.

The value in this register is moved to a counter each time the system microprocessor accesses the channel. Decrementing begins when the system microprocessor starts the channel access.

If the system microprocessor still owns the channel when the count reaches 0, the system microprocessor gives up ownership of the channel

Parity Trap Register (Hex 00E1)

This 8-bit, read-only register at address hex 00E1 is used to determine if system-board memory caused a parity error. When a parity error occurs, the state of 'row address select' signals is latched into this register.

If a bit is a 1, it indicates that the memory in that pair of matching connectors (A and B) was the cause of the error. To determine which memory card caused the error, test bits 0 and 1 in the Cache Status register (Hex 00E3).

If all bits in this register are 0, the error was not caused by the memory on the system board.

After each parity check, this register must be cleared using the parity-check bit (bit 0 of the Memory Controller register, index A1) to reset the latch.

Bit	Function
7	Error Indicator Connector A4 or B4
6	Error Indicator Connector A4 or B4
5	Error Indicator Connector A3 or B3
4	Error Indicator Connector A3 or B3
3	Error Indicator Connector A2 or B2
2	Error Indicator Connector A2 or B2
1	Error Indicator Connector A1 or B1
0	Error Indicator Connector A1 or B1

Figure 4-24. Parity Trap Register (Hex 00E1)

The bank that generated the parity error can by determined by bits 0 and 1 in the Cache Status Register (Hex 00E3).

Cache Control Register (Hex 00E2)

This 8-bit, read/write register at address hex 00E2

Bit	Function
7	-Flush Level 1 Cache
6	-Flush Level 2 Cache
5	-Enable Level 2 Cache
4–0	Reserved

Figure 4-25. Cache Control Register (Hex 00E2)

Bit 7 This bit controls the flushing of the level 1 cache. When the bit is toggled from 1 to 0, the cache is placed in the flushed state for one processor cycle. When this bit is set to 1, the cache is in the enabled state.

Note: To ensure that the cache is flushed, the cache must be in the enabled state before setting this bit to 0.

Bit 6 This bit controls the flushing of the level 2 cache. When the bit is set to 0, the cache is placed and held in the flushed state. When this bit is set to 1, the state of the level 2 cache is controlled by bit 5.

Bit 5 This bit and bit 6 work together to control the state of the level 2 cache. When this bit is 0, the level 2 cache is enabled.

To disable both caches at the same time, use the Cache/Timer Control register (index hex A2), then use the Cache Control register to flush both caches.

Bits 4–0 These bits are reserved.

Notes:

1. Refer to the 80486 microprocessor specification for information on CR0, which must also be set to enable the internal 8KB cache.
2. The cache must be disabled and flushed before enabling ROM or changing the cacheable boundaries.

Cache Status Register (Hex 00E3)

This 8-bit, read-only register contains diagnostic information related to the cache operation.

Bit	Function
7	Reserved
6	Processor Type
5, 4	-L2 Cache Installed
3	-L2 Cache Parity Error
2	Reserved
1	-Parity Error Bank A
0	-Parity Error Bank B

Figure 4-26. Cache Status Register (Hex 00E3)

Bit 7 This bit is reserved.

Bit 6 This bit indicates the type of processor complex that is present. When the bit is a 0, the Type 1 processor complex is installed; when the bit is 1, the Type 2 is installed.

Bits 5, 4 These bits identify the presence of the level 2 cache. When both bits are 0, the 256KB level-2 cache is present; when both bits are 1, the level 2 cache is not present. Other combinations of these bits are reserved.

Bit 3 When this bit is a 0, a parity error was detected in the level 2 cache.

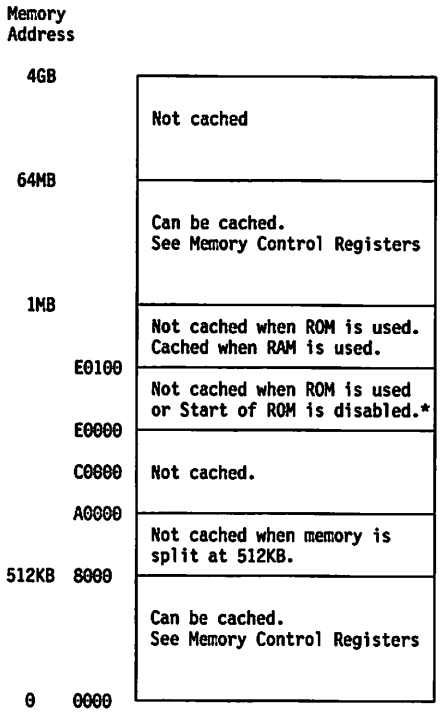
Bit 2 This bit is reserved.

Bit 1 When this bit is a 0, a parity error was detected in system-board memory in an A connector.

Bit 0 When this bit is a 0, a parity error was detected in system-board memory in a B connector.

Processor Cache Address Map

The following map is a summary of the Memory Control register options. Only system-board memory is mapped into the microprocessor and level-2 caches. The level 2 (L2) cache is optional.



* Refer to Bit 5 on page 4-16 for details.

Figure 4-27. Cache Address Map Summary

Read-Only Memory Subsystem

The ROM subsystem is located on the processor complex and consists of 128KB in a 128K by 8-bit arrangement. The ROM is active at power-on at address space hex 000E0000 to 000FFFFF. The ROM contains the code that initializes the system to load and execute POST and BIOS code from a DASD device. After POST ensures the proper operation of system memory, POST and BIOS code from a DASD device can be copied to RAM, and ROM disabled. ROM is also assigned at the top of the 4GB address space from hex FFFE0000 to FFFFFFFF.

Two bits in the Memory Control register (index A1) control ROM addressing. The ROM-enable bit determines whether ROM or RAM is accessed at address space hex 000E0000 to 000FFFFF, and the disable-ROM-space-decode bit selects the system decode and cacheability of the first 4KB of ROM address space.

Index

A

acoustic readings 1-7
adapter setup register 2-12
address maps
  POS 2-4
  system 1-4
address 20 gate 3-19
alternate reset 3-20
arbiter programming 4-9
arbitration control point 4-8
arbitration level 4-8
arbitration register 4-9
arbitration, central control point 4-9
assistance, developer 3-23

B

base memory bytes 3-16
battery 3-5
boot sequence 2-15
bus contention 2-2
bus priority assignments 4-8

C

cache connector 4-7
cache map 4-28
card selected feedback 2-5
-CD SFDBK 2-5
central arbitration control point 4-8
channel 1-3, 4-7
channel check 3-18
channel check latch 3-18
CMOS RAM 3-5
codes, POST error 3-23
configuration CRC bytes 3-16
configuration, RT/CMOS 3-12
connector numbering, memory 3-3
connector, optional cache 4-7

D

date century byte 3-16
description
  NMI 3-17
  password 3-21
  programmable option select 2-2
  real-time clock 3-5
  system 1-2
  system board 1-3
description, power supply 1-8
developer assistance 3-23
diagnostic status byte (RT/CMOS) 3-12
diskette drive connector 3-2
diskette drive connectors 3-2
diskette drive information, POS 2-10, 2-11
diskette drive type byte 3-13
diskette overruns 4-8
DMA channels 4-8

E

enable error checking 4-17
enable NMI 3-6
equipment byte 3-15
error codes, POST 3-23
extended arbitration 4-10

F

failures and messages, POST 3-23
fault, overvoltage 1-9
features
  processor complex 4-2
feedback, card selected 2-5
fixed disk drive activity light 1-8
fixed disk drive initialization 3-12
fixed disk type byte 3-14

© Copyright IBM Corp. 1990

G

gate A20 3-19

H

hardware compatibility 3-22

I

identifier, model 4-2
interrupts
  IRQ0 reset 3-18
  NMI reset 3-17
I/O address map 1-4

L

light, fixed disk drive activity 1-8
light, power 1-8
L2 cache 4-7

M

math coprocessor
maximum current 1-5
measurements 1-5
memory
  connectors 3-3
  performance 4-3
  refresh 4-8
  split block 4-17
memory information, POS 2-8
memory numbering 3-3
memory presence detect 4-11
memory size miscompare 3-12
messages, POST error 3-23
Micro Channel 1-3, 4-7
microprocessor
  alternate reset 3-20
  gate A20 3-19
  mode switch compatibility 3-20
  performance 4-3
mode switching 3-20
model/submodel bytes 4-2

N

NMI mask 3-6
NMI service 4-8
NMI signal 4-8
noise readings 1-7
nonmaskable interrupt (NMI) 3-17
numbering, memory connector 3-3

O

operating system boot
  sequence 2-15
optional cache card 4-7
output voltage sequencing 1-9
outputs, power supply 1-8
overvoltage fault 1-9

P

parallel port select 2-7
parity check 3-18, 4-17
password, power-on 3-21
performance 3-2, 4-3
ports, system 3-17
POS address map 2-4
POS register 2 2-6
POS register 3 2-8
POS register, diskette drive 2-10,
  2-11
POST error codes 3-23
power specifications 1-5
power supply 1-3
power supply description 1-8
power supply LED 1-8
power supply output protection 1-9
power supply outputs 1-8
power-on password 3-21
processor cache map 4-28
processor complex
processor performance 4-3
programmable option select
  (POS) 2-2
    address decode 2-5
    address map 2-4
    registers 2-4
    setup procedures 2-3

programmable option select (POS) *(continued)*
setup register 2-5
system board 2-5
system board I/O 2-6
system configuration utilities 2-2
video 2-6
warning 2-2
programming, arbiter 4-9
protected mode switch 3-20
protection, power supply 1-9

R

real mode switch 3-20
real-time clock bytes 3-8
refresh 4-8
refresh request 3-18
registers
adapter setup 2-12
card feedback 2-5
miscellaneous system 3-17
RT/CMOS and NMI mask 3-6
RT/CMOS status registers 3-9
serial port 2 2-13
system board POS 2-5
system control 3-17
restore channel check 3-18
ROM enable 4-17
RT/CMOS address register 3-6
RT/CMOS RAM 3-5
password 3-21
secured area 3-19

S

second fixed disk type byte 3-14
second serial setup 2-13
selectable boot 2-15
sequencing, output voltage 1-9
serial port address select 2-10
set configuration 2-15
setup
bus contention 2-2
registers 2-5
system board 2-5
setup *(continued)*
system board I/O 2-6
video 2-6
shutdown status byte 3-12
size 1-5
sound specifications 1-7
speaker data enable 3-18
specifications 1-5, 4-3
performance 4-3
physical 1-5
system unit 1-5
split address bits 4-15
split memory block 4-17
status registers (RT/CMOS) 3-9
switch to real 3-20
system address map 1-4
system board 1-2, 1-3
features 1-3
I/O register 2-6
memory connector 3-3
model identifier 4-2
POS registers 2-5
setup 2-5
system configuration utilities 2-2
system control port A 3-19
system control ports 3-17
system description 1-2
system overview 1-1
system ports, miscellaneous 3-17
system reset 3-17
system unit specifications 1-5

T

time status indicator 3-12
type identifier 4-2

V

valid RAM bit 3-11
video setup 2-6

W

warnings
arbitration 4-8
central arbiter 4-10

warnings *(continued)*
NMI mask 3-17
POS 2-2
real-time clock 3-6
reserved areas v
watchdog status 3-19

Numerics

80486 performance 4-3



Personal System/2
Model 95 XP 486
Technical Reference

First Edition (October, 1990)

The following paragraph does not apply to the United Kingdom or any country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.

It is possible that this publication may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

Requests for technical information about IBM products should be made to your IBM Authorized Dealer or your IBM Marketing Representative.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, Purchase, NY 10577.

© Copyright International Business Machines Corporation 1990. All rights reserved.

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Special Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights or other legally protectible rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, programs, or services, except those expressly designated by IBM, are the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, Purchase, NY 10577.

The following terms, denoted by an asterisk (*) in this publication, are trademarks of the IBM Corporation in the United States and/or other countries:

IBM
Micro Channel
Personal Computer AT
Personal System/2
PS/2

The following terms, denoted by a double asterisk (**) in this publication, are trademarks of other companies as follows:

Intel	Intel Corporation
Motorola	Motorola, Incorporated
National Semiconductor	National Semiconductor Corporation

Notes:

Preface

The Technical Reference library is intended for those who develop hardware and software products for IBM Personal Computers and IBM Personal System/2 products. Users should understand computer architecture and programming concepts.

This manual consists of the following sections:

Section 1, "System Overview," describes the system, features, and specifications.

Section 2, "Programmable Option Select," describes the registers used for configuration.

Section 3, "System Board," describes the system-specific hardware implementations of the Micro Channel architecture.

Section 4, "Processor Complex" describes the components and features of the processor complex.

This technical reference should be used with the following publications. These publications contain additional information on many of the subjects discussed in this technical reference.

IBM Personal System/2 Hardware Interface Technical Reference — Architectures

IBM Personal System/2 Hardware Interface Technical Reference — Common Interfaces

IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference

Information about diskette drives, fixed disk drives, adapters, and external options are in separate option technical references.

Warning: The term "Reserved" describes certain signals, bits, and registers that should not be changed. Use of reserved areas can cause compatibility problems, loss of data, or permanent damage to the hardware. When the contents of a register are changed, the state of the reserved bits must be preserved. When possible, read the register first and change only the bits that must be changed.

© Copyright IBM Corp. 1990

Notes:

Contents

Section 1. System Overview 1-1
Description 1-2
System Board Devices and Features 1-3
System Board I/O Address Map 1-4
Micro Channel Compatibility 1-4
Specifications 1-6
Power Supply 1-8
  Outputs 1-8
  Voltage Sequencing 1-9
  Power Supply Connectors 1-10

Section 2. Programmable Option Select 2-1
Description 2-2
POS Address Map 2-3
Card Selected Feedback 2-4
System Board Setup 2-4
  System Board Enable/Setup Register (Hex 0094) 2-4
  System Board POS Register 2 (Hex 0102) 2-5
  System Board POS Register 3 (Hex 0103) 2-6
  System Board POS Registers 4 (Hex 0104) 2-7
  System Board POS Register 5 (Hex 0105) 2-8
  Diskette Drive POS Registers 4 (Hex 0104) 2-9
  Adapter Enable/Setup Register (Hex 0096) 2-10
Selectable Boot Sequence 2-11

Section 3. System Board 3-1
Description 3-2
  Diskette Drive Connector 3-2
  System Board Memory Connectors 3-3
  Real-Time Clock/Complementary Metal-Oxide Semiconductor RAM 3-5
Miscellaneous System Functions 3-17
  Nonmaskable Interrupt 3-17
  System Control Port B (Hex 0061) 3-17
  System Control Port A (Hex 0092) 3-19
  Power-On Password 3-21
Hardware Compatibility 3-22
Error Codes 3-23

Section 4. Processor Complex 4-1
Type 1 4-3
Type 2 4-5

© Copyright IBM Corp. 1990

I/O Address Map 4-7
Optional Cache 4-7
Micro Channel 4-7
Central Arbiter 4-8
Central Arbiter Programming 4-9
Memory Subsystem 4-11
Memory Presence Detect 4-11
Memory Controller Registers 4-12
Processor Cache Address Map 4-28
Read-Only Memory Subsystem 4-29

Index X-1

Figures

1-1. System Board Devices and Features 1-3
1-2. System Board I/O Address Map 1-4
1-3. Physical Specifications 1-6
1-4. Acoustical Readings 1-7
1-5. System-Provided Power 1-8
1-6. Load Current for Micro Channel Adapter Connector . . . 1-9
1-7. Voltage Assignments for the Internal Drive Power-Supply Connectors 1-10
2-1. POS I/O Address Map 2-3
2-2. Card Selected Feedback Register (Hex 0091) 2-4
2-3. System Board Enable/Setup Register (Hex 0094) 2-4
2-4. System Board POS Register 2 (Hex 0102) 2-5
2-5. Parallel Port Select Bits 2-6
2-6. System Board POS Register 3 (Hex 0103) – Write 2-6
2-7. System Board POS Register 3 (Hex 0103) – Read 2-7
2-8. System Board POS Register 4 (Hex 0104) 2-7
2-9. System Board POS Register 5 (Hex 0105) 2-8
2-10. Serial Port Address Select Bits 2-8
2-11. Diskette Drive POS Register 4 (Hex 0104) 2-9
2-12. Adapter Enable/Setup Register (Hex 0096) 2-10
3-1. Diskette Drive Connector 3-2
3-2. Memory Connector Numbering 3-3
3-3. System Board Memory Connector 3-4
3-4. RT/CMOS RAM Address Map 3-5
3-5. RT/CMOS Address Register and NMI Mask (Hex 0070) . 3-6
3-6. RT/CMOS Data Register (Hex 0071) 3-6
3-7. Real-Time Clock Bytes 3-8
3-8. Status Register A 3-9
3-9. Status Register B 3-9
3-10. Status Register C 3-11
3-11. Status Register D 3-11
3-12. Diagnostic Status Byte 3-12
3-13. Diskette Drive Type Byte 3-13
3-14. Diskette Drive Type Byte (Bits 7–4) 3-13
3-15. Diskette Drive Type Byte (Bits 3–0) 3-13
3-16. Equipment Byte 3-15
3-17. Equipment Byte (Bits 7, 6) 3-15
3-18. Equipment Byte (Bits 5, 4) 3-15
3-19. System Control Port B (Hex 0061) – Write 3-18
3-20. System Control Port B (Hex 0061) – Read 3-18
3-21. System Control Port A (Hex 0092) 3-19
3-22. POST Error Message Table 3-24

© Copyright IBM Corp. 1990

4-1. Model and Submodel Bytes 4-2
4-2. Type 1 Processor Complex Feature and Devices 4-3
4-3. System Performance Specifications, Type 1 Processor Complex 4-3
4-4. Memory Performance Specifications, Type 1 Processor Complex 4-3
4-5. Type 2 Processor Complex Feature and Devices 4-5
4-6. System Performance Specifications, Type 2 Processor Complex 4-5
4-7. Memory Performance Specifications, Type 2 Processor Complex 4-5
4-8. Processor Complex I/O Address Map 4-7
4-9. Arbitration Bus Priority Assignments 4-8
4-10. Arbitration Register (Hex 0090) — Write 4-9
4-11. Arbitration Register (Hex 0090) — Read 4-9
4-12. Presence-Detect Bits 4-11
4-13. Memory Controller Register Assignments 4-12
4-14. Memory Control 4-14
4-15. Memory Bank Base Address Register (Index Hex 90 — 9F) 4-15
4-16. Split Address Register (Index Hex A0) 4-15
4-17. Memory Encoding Register (Index Hex A1) 4-16
4-18. Cache/Timer Control Register (Index Hex A2) 4-18
4-19. Boundary 0 Cache Control Register (Index Hex A3) . . 4-19
4-20. Lower Boundary 1 Cache Control Register (Index Hex A4) 4-20
4-21. Upper Boundary 1 Cache Control Register (Index Hex A5) 4-20
4-22. Memory Bank Enable Register 1 (Index Hex A6) 4-21
4-23. Memory Bank Enable Register 2 (Index Hex A7) 4-22
4-24. Parity Trap Register (Hex 00E1) 4-25
4-25. Cache Control Register (Hex 00E2) 4-26
4-26. Cache Status Register (Hex 00E3) 4-27
4-27. Cache Address Map Summary 4-28

Section 1. System Overview

Description 1-2
System Board Devices and Features 1-3
System Board I/O Address Map 1-4
Micro Channel Compatibility 1-4
Specifications 1-6
Power Supply 1-8
  Outputs 1-8
    Output Protection 1-9
  Voltage Sequencing 1-9
  Power Supply Connectors 1-10

© Copyright IBM Corp. 1990

Description

The IBM* Personal System/2* Model 95 is floor-standing computer system with a keyboard. The system can support up to four removable-media DASD devices (diskette, tape, optical) and three nonremovable-media DASD devices.

All Model 95 systems use the same system board. This system board has:

- Eight Micro Channel* connectors
- Eight connectors for memory
- Parallel port, serial port, and diskette drive controllers
- Processor complex connector
- Real-time clock.

The processor complex determines system performance and the model/submodel byte information. The major components on the processor complex are:

- System microprocessor
- DMA controller
- Memory controller.

* IBM, Personal System/2, PS/2, and Micro Channel are trademarks of the International Business Corporation.

System Board Devices and Features

The following table lists the system board devices and features. The *Hardware Interface Technical Reference — Common Interfaces* describes devices common to PS/2 products by type number.

Device	Type	Features
Microprocessor	—	Connector for processor complex
System Timers	1	Channel 0 — System timer Channel 2 — Tone generation for speaker Channel 3 — Watchdog timer
RAM Subsystem	—	Eight connectors on system board Refer to the processor complex for more information
CMOS RAM Subsystem	—	64-byte CMOS RAM with real-time clock/calendar 8KB CMOS RAM extension (KB = 1,024 bytes) Battery backup
Audio Subsystem	1	Driven by: — System-timer channel 2 — The 'audio sum node' signal.
Interrupt Controller	1	16 levels of system interrupts Interrupts are level-sensitive
Keyboard/Auxiliary Device Controller	2	Keyboard connector Auxiliary device connector Password security
Diskette Drive Controller	2	Supports: — 3.5-in. diskette drive (1.44MB) — 5.25-in. diskette drive (360KB and 1.2MB)
Serial Controller	3	RS-232C interface Programmable as serial port 1–8 DMA, FIFO, and character modes Supports DMA operation
Parallel Controller	3	Programmable as parallel port 1–4 Supports bidirectional input and output Supports DMA operation
Micro Channel	—	Eight connectors for Type 3 adapters: — One 32-bit connector with auxiliary video extension — One 32-bit connector with base video extension — Six 32-bit connectors with matched memory extension One connector contains a fixed disk adapter One connector contains a video adapter

Figure 1-1. System Board Devices and Features

System Board I/O Address Map

Hex Addresses	Device
0020, 0021	Interrupt Controller (Master)
0040, 0042–0044, 0047	System Timers
0060	Keyboard, Auxiliary Device
0061	System Control Port B
0064	Keyboard, Auxiliary Device
0070, 0071	RT/CMOS and NMI Mask
0091	Card Selected Feedback Register
0092	System Control Port A
0094	System Board Enable/Setup Register
0096	Adapter Enable/Setup Register
00A0–00A1	Interrupt Controller (Slave)
0100–0107	Programmable Option Select
0108–010F	Information Panel
0278–027D	Parallel Port 3
02F8–02FF	Serial Port 2
0378–037D	Parallel Port 2
03BC–03BF	Parallel Port 1
03F0–03F7	Diskette Drive Controller
03F8–03FF	Serial Port 1
1278–127D	Parallel Port 1 (DMA mode)
1378–137D	Parallel Port 4
3220–3227	Serial Port 3
3228–322F	Serial Port 4
4220–4227	Serial Port 5
4228–422F	Serial Port 6
5220–5227	Serial Port 7
5228–522F	Serial Port 8
83F8–83FF	Serial Port 1 (DMA mode)
82F8–82FF	Serial Port 2 (DMA mode)
B220–B227	Serial Port 3 (DMA mode)
B228–B22F	Serial Port 4 (DMA mode)
C220–C227	Serial Port 5 (DMA mode)
C228–C22F	Serial Port 6 (DMA mode)
D220–D227	Serial Port 7 (DMA mode)
D228–D22F	Serial Port 8 (DMA mode)

Figure 1-2. System Board I/O Address Map

Micro Channel Compatibility

Following is a description of restrictions in the support of certain Micro Channel Procedures.

Streaming Data

The system board supports 10 MHz-streaming data procedures involving transfers between adapters.

Devices on the system board and the processor complex work with streaming-data adapters; however, they do not

use the streaming data procedure. Transfers default to the basic transfer procedure.

Data Parity

The system board supports data-parity procedures involving transfers between adapters.

Devices on the system board and the processor complex work with adapters that support data parity; however, they do not use the data parity option. Transfers default to the basic transfer procedure.

Address Parity

The system board supports address-parity procedures involving transfers between adapters.

Devices on the system board and the processor complex work with adapters that support address parity; however, they do not use the address parity option. Transfers default to the basic transfer procedure.

Select Feedback Return Signal

The system board and the processor complex support the 'selected feedback return' signal except for I/O devices at addresses below hex 0110. These devices are controlled by the system master only and should not be accessed by bus masters.

Synchronous Channel Check

Synchronous channel checks for errors other than Micro Channel data-parity errors or Micro Channel address-parity errors are not supported in this system.

Synchronous channel checks for data-parity or address-parity errors, typically issued by slaves, are the responsibility of the master that owned the channel and issued the access to the slave. However, in this system, these channel checks can also be detected by logic on the system board, resulting in an NMI.

DMA

In this system, the DMA controller supports DMA slaves with 8-bit or 16-bit data widths. Transfers are limited to the memory address space below 16MB.

Specifications

Size:	
Width	203 mm (8.0 in)
Depth	508 mm (20.0 in)
Height	501 mm (19.75 in)
Weight:	
Minimum Configuration	22 kg (50 lb)
Maximum Configuration	30 kg (67 lb)
Cables:	
Power Cable	1.8 m (6 ft)
Keyboard Cable	3.05 m (10 ft)
Air Temperature:	
System On	
0 - 3000 ft	10.0 to 35.0°C (50 to 95°F)
3000 - 7000 ft	10.0 to 32.2°C (50 to 90°F)
System Off	10.0 to 43.0°C (50 to 110°F)
Humidity:	8% to 80%
Maximum Altitude	2134 m (7000 ft)
Heat Output	515 W (1757 BTU per hour)
Acoustical Readings	
See Figure 1-4 on page 1-7	
Electrical:	
Input Voltage:	Range is switch selected. Sinewave input required
Low Range	90 (min) – 137 (max) Vac
High Range	180 (min) – 265 (max) Vac
Input Frequency:	50 ± 3Hz or 60 ± 3Hz
Maximum Current Draw:	
Low Range	9.2 A
High Range	4.6 A
Input in Kilovolt-Ampere (kVA):	
Minimum Configuration (as shipped)	0.18 kVa
Maximum Configuration	0.78 kVa
Electromagnetic Compatibility	FCC Class B

Figure 1-3. Physical Specifications

Description	L_{WAd} in bels		L_{pAm} in dB		$\langle L_{pA} \rangle_m$ in dB	
	Operate	Idle	Operate	Idle	Operate	Idle
Model 95	5.3	5.3	33	33	36	36

Notes:

L_{WAd} is the declared sound power level for the random sample of machines.

L_{pAm} is the mean value of the A-weighted sound pressure levels at the operator position (if any) for the random sample of machines.

$\langle L_{pA} \rangle_m$ is the mean value of the A-weighted sound pressure levels at the one-meter positions for the random sample of machines.

All measurements made in accordance with ANSI S12.10, and reported in conformance with ISO DIS 9296.

The measurements are preliminary data and subject to change.

Figure 1-4. Acoustical Readings

Power Supply

The power supply requires a sinewave input and converts the ac input voltage to three dc output voltages. The switch near the power cord is used to select the input-voltage range. The power supply provides power for the following:

- System board
- Processor complex
- Micro Channel adapters
- Internal diskette drives
- Internal fixed disk drives
- Auxiliary device
- Keyboard.

The power switch and one light-emitting diode (LED) are on the front of the system unit. The power supply is operating properly when the LED is lit.

Outputs

The power supply provides separate voltage sources for the system board and the drives. The system-board voltages are +5 Vdc, +12 Vdc, and −12 Vdc. The voltages for the internal drives are +5 Vdc and +12 Vdc. The following is a list of the power provided for system components.

System Component	Maixmum Current	
	+12	+5
Internal Fixed Disk Drives *	2.5 A	1.5 A
Total of Fixed Disk Drives	7.5 A	4.6 A
Auxiliary Device		300 mA
Keyboard		275 mA
* Per power supply connector		

Figure 1-5. System-Provided Power

The following are the load currents allowed for the connectors for the Type 3 adapters. The formulas used to determine the power requirements and the definition of adapter types can be found in the section of the *Hardware Interface Technical Reference — Architectures* that describes the Micro Channel architecture.

Supply Voltage	Maximum Current
+ 5.0 Vdc	2.0 A
+12.0 Vdc	0.175 A
−12.0 Vdc	0.040 A

Figure 1-6. Load Current for Micro Channel Adapter Connector

Output Protection

A short circuit placed on any dc output (between outputs or between an output and dc return) latches all dc outputs into a shutdown state with no damage to the power supply.

If an overvoltage fault occurs (internal to the power supply), the power supply latches all dc outputs into a shutdown state before any output exceeds 130% of its nominal value.

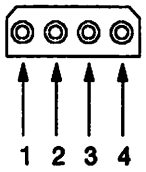
If either of these shutdown states is actuated, the power supply returns to normal operation only after the fault has been removed and the power switch has been turned off for at least one second.

Voltage Sequencing

At power-on time, the output voltages track within 50 milliseconds of each other when measured at the 50% points.

Power Supply Connectors

The power supply provides three 4-pin connectors for internal devices. The connectors can be extended to provide power to more than one internal drive as long as the total power for all connectors does not exceed the connector specifications shown in Figure 1-5 on page 1-8.



Pin	Signal	Pin	Signal
1	+12 Vdc	3	DC Return
2	DC Return	4	+5 Vdc

Figure 1-7. Voltage Assignments for the Internal Drive Power-Supply Connectors

Section 2. Programmable Option Select

Description 2-2
POS Address Map 2-3
Card Selected Feedback 2-4
System Board Setup 2-4
  System Board Enable/Setup Register (Hex 0094) 2-4
  System Board POS Register 2 (Hex 0102) 2-5
  System Board POS Register 3 (Hex 0103) 2-6
  System Board POS Registers 4 (Hex 0104) 2-7
  System Board POS Register 5 (Hex 0105) 2-8
  Diskette Drive POS Registers 4 (Hex 0104) 2-9
  Adapter Enable/Setup Register (Hex 0096) 2-10
Selectable Boot Sequence 2-11

© Copyright IBM Corp. 1990

Description

Programmable Option Select (POS) eliminates the need for switches by replacing their function with programmable registers. This section describes the POS information used on the Model 95 system board. For additional POS information, refer to the *Hardware Interface Technical Reference*.

Warning:

- IBM recommends that programmable options be set only through the System Configuration utilities. Directly setting the POS registers or CMOS RAM POS parameters can result in multiple assignment of the same system resource, improper operation of the feature, loss of data, or possible damage to the hardware.
- Application programs should not use the adapter identification (ID) unless absolutely necessary. Compatibility problems can result.
- If an adapter and the system board are in setup mode at the same time, bus contention will occur, no useful programming can take place, and damage to the hardware can occur.
- After setup operations are complete, the Adapter Enable/Setup register (hex 0096) should be set to hex 00, and the System Board Enable/Setup register (hex 0094) should be set to hex FF.
- Bit 7 (channel reset) in the Adapter Enable/Setup register must be 0 to program the adapters.
- Only 8-bit instructions are supported for setup operations. Using 32- or 16-bit I/O instructions on 8-bit POS registers will cause erroneous data to be written or read.

Setup functions respond to I/O addresses hex 0100 through 0107 only when their unique setup signal is active. The following precautions must be taken before setting individual bits in the POS registers.

Adapter Setup:

- Bit 3 in the Adapter Enable/Setup register must be set to 1 to allow adapter setup.
- Bits 7–0 in the System Board Enable/Setup register must be set to 1 to avoid driving a 'setup' signal to a system board function.

System Board Setup:

- Bits 7 and 6 in the System Board Enable/Setup register must be set to 0 individually while other bits are set to 1 to setup corresponding system board functions.
- Bit 3 in the Adapter Enable/Setup register must be set to 0 to avoid driving a 'setup' signal to an adapter.

POS Address Map

The following table shows the organization of the I/O address space used by POS. Bit 0 of POS Register 2 and bits 6 and 7 of POS Register 5 are fixed. All other bits in POS Registers 2 through 5 are free-form.

Address (Hex)	Function
0094	System Board Enable/Setup Register
0096	Adapter Enable/Setup Register
0100	POS Register 0—Adapter Identification Byte (Low Byte)
0101	POS Register 1—Adapter Identification Byte (High Byte)
0102	POS Register 2—Option Select Data Byte 1 Bit 0 is Card Enable.
0103	POS Register 3—Option Select Data Byte 2
0104	POS Register 4—Option Select Data Byte 3
0105	POS Register 5—Option Select Data Byte 4
0106	Reserved
0107	Reserved

Figure 2-1. POS I/O Address Map

Card Selected Feedback

When an adapter is addressed, it responds by setting the '-card selected feedback' signal (-CD SFDBK) to active. -CD SFDBK is derived from the address decode and driven by a totem pole driver. It is latched by the system board and can be read through the Card Selected Feedback register at address hex 0091.

The Card Selected Feedback register is a read-only register at address hex 0091. It allows programs to monitor -CD SFDBK and thereby determine if the system board I/O or an adapter is addressed and functioning.

Bit	Function
7–1	Reserved
0	-Card Selected Feedback

Figure 2-2. Card Selected Feedback Register (Hex 0091)

Bits 7–1 These bits are reserved.

Bit 0 This bit is set to 1 whenever -CD SFDBK was active on a previous cycle or whenever the system board I/O functions (diskette drive, serial, or parallel interfaces) are accessed by an I/O or DMA cycle. Reading this register resets the bit to 0.

System Board Setup

The integrated I/O functions on the system board use POS information during setup. The diskette drive controller, serial port, and parallel port are treated as a single device. The System Board Enable/Setup register is used to place the system board into the setup mode.

System Board Enable/Setup Register (Hex 0094)

This is a read/write register; all bits in this register default to 1 (enabled). Setup functions should only be accessed one at a time.

Bit	Function
7	Enable/-Setup System Board Functions
6	Enable/-Setup Diskette Drive Functions
5–0	Reserved

Figure 2-3. System Board Enable/Setup Register (Hex 0094)

Bit 7 When this bit is set to 0, various system board I/O functions are placed in the setup mode. The diskette drive controller, serial port, and parallel port are controlled through System Board POS Register 2 (hex 0102).

When this bit is set to 1, the system board function is enabled.

Bit 6 Additional setup for diskette drive. When this bit is set to 0, these setup functions are available through the Diskette Drive POS Register 4.

Bits 5–0 These bits are reserved.

System Board POS Register 2 (Hex 0102)

When the system board is in the setup mode, the diskette drive controller, serial port, and parallel port are controlled by this read/write register. Reading this register returns the current state of these system board functions.

Bit	Function
7	Disable Parallel Port Extended Mode
6, 5	Parallel Port Select
4	Enable Parallel Port
3	Serial Port IRQ Select
2	Enable Serial Port
1	Enable Diskette Drive Interface
0	Enable System Board

Figure 2-4. System Board POS Register 2 (Hex 0102)

Bit 7 When set to 0, this bit allows the parallel port to be configured as an 8-bit, parallel, bidirectional interface. When set to 1, this bit disables the bidirectional mode. This bit is set to 0 after power-on.

Bits 6, 5 These bits select the configuration of the system board parallel port.

Bits 6 5	Assignment	Hex Address	Interrupt Level
0 0	Parallel 1	03BC–03BF*	7
0 1	Parallel 2	0378–037D	7
1 0	Parallel 3	0278–027D	7
1 1	Parallel 4	1378–137D	7

* The addresses are hex 1278–127D for DMA operations

Figure 2-5. Parallel Port Select Bits

Bit 4 When this bit and bit 0 are set to 1, the system-board parallel port is enabled.

Bit 3 When this bit is set to 1, the serial port uses interrupt request level 4. When set to 0, the serial port uses interrupt request level 3.

Bit 2 When this bit and bit 0 are set to 1, the system board serial port is enabled.

Bit 1 When this bit and bit 0 are set to 1, the diskette drive interface is enabled.

Bit 0 When set to 1, this bit allows bits 4, 2, and 1 to enable and disable their respective devices. When set to 0, this bit disables the diskette drive interface, system board serial port, and system board parallel port, regardless of the state of bits 4, 2, and 1.

System Board POS Register 3 (Hex 0103)

The system board has connectors that provide installation of eight memory cards (for more information about these memory connectors, see "System Board Memory Connectors" on page 3-3). Information about each memory card is contained in four read-only data registers; each data register is selected by writing its index to the write-only index register.

POS Register 3 provides access to the read-only data registers and a write-only index register. To determine the memory information for a particular connector, write the appropriate index to POS Register 3, then read POS Register 3. The following shows the relationship of the index value, the memory connector, and the data register information available.

Index Value (Hex)	Description
01	Presence Detect Information for A1 and B1
05	Presence Detect Information for A2 and B2
09	Presence Detect Information for A3 and B3
0D	Presence Detect Information for A4 and B4
All values not show are reserved.	

Figure 2-6. System Board POS Register 3 (Hex 0103) — Write

The following information is returned by reading POS Register 3 when the appropriate value is written to POS Register 3. For coding of the presence detect signals, see Figure 4-12 on page 4-11.

Bit	Description
7	Presence Detect 3, Connector B
6	Presence Detect 2, Connector B
5	Presence Detect 1, Connector B
4	Presence Detect 0, Connector B
3	Presence Detect 3, Connector A
2	Presence Detect 2, Connector A
1	Presence Detect 1, Connector A
0	Presence Detect 0, Connector A

Figure 2-7. System Board POS Register 3 (Hex 0103) — Read

System Board POS Registers 4 (Hex 0104)

This register is accessed when bit 7 of the System Board Enable/Setup register is set to 0. The register is used to specify the arbitration levels used by the serial port during transmit and receive operations in the DMA mode.

Bit	Description
7–4	Receive Arbitration Level
3–0	Transmit Arbitration Level

Figure 2-8. System Board POS Register 4 (Hex 0104)

Bits 7–4 These bits select the arbitration level used for the transfer of data being received through the serial port.

Bits 3–0 These bits select the arbitration level used for the transfer of data being transmitted through the serial port.

System Board POS Register 5 (Hex 0105)

This register is used to specify the arbitration level used by the parallel port and the I/O addresses used by the serial port.

Bit	Description
7	Reserved
6–4	Serial Address Select
3–0	Parallel Port Arbitration Level

Figure 2-9. System Board POS Register 5 (Hex 0105)

Bit 7 This bit is reserved.

Note: This bit may read as 0 but does not indicate any occurrence of channel check.

Bits 6–4 These bits are used to select the I/O addresses used by the serial port on the system board. The I/O addresses can be the compatible addresses or the extended address (see the technical reference for the serial port controller for more information on addressing).

Bits 6 5 4	Assignment	Hex Address	
		Compatible	Extended
0 0 0	Serial 1	03F8–03FF	83F8–83FF
0 0 1	Serial 2	02F8–02FF	82F8–82FF
0 1 0	Serial 3	3220–3227	B220–B227
0 1 1	Serial 4	3228–322F	B228–B22F
1 0 0	Serial 5	4220–4227	C220–C227
1 0 1	Serial 6	4228–422F	C228–C22F
1 1 0	Serial 7	5220–5227	D220–D227
1 1 1	Serial 8	5228–522F	D228–D22F

Figure 2-10. Serial Port Address Select Bits

Bits 3–0 These bits select the arbitration level used by the parallel port controller.

Diskette Drive POS Registers 4 (Hex 0104)

This register is accessed when the bit 6 of the System Board Enable/Setup register is set to 0. The register is used to specify the arbitration level used by the diskette drive controller.

Bit	Description
7–4	Reserved
3–0	Diskette Drive Arbitration Level

Figure 2-11. Diskette Drive POS Register 4 (Hex 0104)

Bits 7–4 These bits are reserved.

Bits 3–0 These bits select the arbitration level used by the diskette drive controller.

Adapter Enable/Setup Register (Hex 0096)

This read/write register selects the connector to be configured.

Bit	Symbol
7	Channel Reset
6–4	Reserved = 0
3	Card Setup Enable
2–0	Channel Select 2–0

Figure 2-12. Adapter Enable/Setup Register (Hex 0096)

Bit 7 When set to 1, this bit activates the 'channel reset' signal to all connectors.

Bits 6–4 These bits are reserved and must be set to 0.

Bit 3 When set to 1, this bit enables the '-card setup' signal selected by bits 2 through 0.

Bits 2–0 These bits are the address bits for the '-card setup' signal. Connectors 1 through 8 are addressed as 0 through 7. When bit 3 is set to 1, these bits select the connector for setup cycles.

Each channel connector has a unique '-card setup' signal (-CD SETUP) associated with it. This signal is used to select the adapters for setup cycles, which allows access to the POS registers. The individual connectors are selected through the Adapter Enable/Setup register. Setup information is then read from or written to the selected adapter through I/O addresses hex 0100 through 0107.

Note: -CD SETUP only goes active when an operation is performed in the I/O address range hex 0100 through 0107.

Selectable Boot Sequence

Selectable boot allows the user to specify the order in which the drives are searched when looking for the boot record of an operating system. Only the diskette and fixed disk drives can be part of the search sequence. Changing the sequence for selectable boot affects the drive letter ordering the next time the system is booted.

The user selects the boot sequence with the Select Startup option in the Set Feature utility. The user is presented with a list of possible drives from which to select. After the sequence is selected, it is stored in nonvolatile RAM (NVRAM). When the system boots, it searches for a drive that contains a bootable record using the specified sequence.

After the operating system boots, the drive lettering is determined by the sequence and which drive contained the bootable source. If the operating system boots from a diskette drive, that drive is drive A. If the operating system boots from a fixed disk drive, that fixed disk drive is drive C. Other drives specified in the sequence can change also.

Section 3. System Board

Description 3-2
Diskette Drive Connector 3-2
System Board Memory Connectors 3-3
Real-Time Clock/Complementary Metal-Oxide Semiconductor RAM 3-5
RT/CMOS Address Register and NMI Mask (Hex 0070) . . . 3-6
RT/CMOS Data Register (Hex 0071) 3-6
RT/CMOS RAM I/O Operations 3-7
Real-Time Clock Bytes (Hex 000–00D) 3-8
CMOS RAM Configuration 3-12
Miscellaneous System Functions 3-17
Nonmaskable Interrupt 3-17
System Control Port B (Hex 0061) 3-17
System Control Port A (Hex 0092) 3-19
Power-On Password 3-21
Hardware Compatibility 3-22
Error Codes 3-23

© Copyright IBM Corp. 1990

Description

This section describes miscellaneous system ports and connectors for the Model 95. For additional information about these and other topics, refer to other descriptions in the *Hardware Interface Technical Reference.*

Diskette Drive Connector

The system provides three 2-by-17-pin connectors for attaching internal diskette drives. Each connector passes control and data signals between the diskette drive controller on the system board and the drives and provides the power to each drive.

The following figure shows the signal assignments and pin numbers for the diskette drive connectors.

Pin	Signal	Pin	Signal
1	Ground	2	Data Rate Select 1
3	+5VDC	4	Drive Type ID 1
5	Ground	6	+12DC
7	Ground	8	-Index
9	Drive Type ID 0	10	Reserved
11	Ground	12	-Drive Select
13	Ground	14	Reserved
15	Ground	16	-Motor Enable
17	Media Type ID 1	18	-Direction In
19	Ground	20	-Step
21	Ground	22	-Write Data
23	Ground	24	-Write Enable
25	Ground	26	-Track 0
27	Media Type ID 0	28	-Write Protect
29	Ground	30	-Read Data
31	Ground	32	-Head 1 Select
33	Data Rate Select 0	34	-Diskette Change

Figure 3-1. Diskette Drive Connector

System Board Memory Connectors

The Model 95 system board has eight 72-pin memory connectors that support a variety of memory cards. The connectors are numbered as viewed from the top of the system.

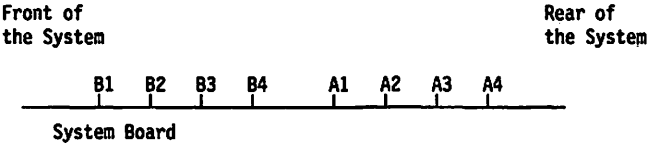


Figure 3-2. Memory Connector Numbering

The following figure shows the pin assignments for the 1- by 72-pin memory connectors. Refer to "Memory Presence Detect" on page 4-11 for decoding of the presence detect signals.

Pin	Signal	Pin	Signal
1	Ground	37	Parity Data 1
2	Data 0	38	Parity Data 3
3	Data 16	39	Ground
4	Data 1	40	Column Address Strobe 0
5	Data 17	41	Column Address Strobe 2
6	Data 2	42	Column Address Strobe 3
7	Data 18	43	Column Address Strobe 1
8	Data 3	44	Row Address Strobe 0
9	Data 19	45	Row Address Strobe 1
10	+5 Vdc	46	Block Select 1
11	-Column Address Strobe P	47	Write Enable
12	Address 0	48	Reserved
13	Address 1	49	Data 8
14	Address 2	50	Data 24
15	Address 3	51	Data 9
16	Address 4	52	Data 25
17	Address 5	53	Data 10
18	Address 6	54	Data 26
19	Reserved	55	Data 11
20	Data 4	56	Data 27
21	Data 20	57	Data 12
22	Data 5	58	Data 28
23	Data 21	59	+5 Vdc
24	Data 6	60	Data 29
25	Data 22	61	Data 13
26	Data 7	62	Data 30
27	Data 23	63	Data 14
28	Address 7	64	Data 31
29	Block Select 0	65	Data 15
30	+5 Vdc	66	Block Select 2
31	Address 8	67	Presence Detect 0
32	Address 9	68	Presence Detect 1
33	Row Address Strobe 3	69	Presence Detect 2
34	Row Address Strobe 2	70	Presence Detect 3
35	Parity Data 2	71	Block Select 3
36	Parity Data 0	72	Ground

Figure 3-3. System Board Memory Connector

Real-Time Clock/Complementary Metal-Oxide Semiconductor RAM

The real-time clock/complementary metal-oxide semiconductor RAM (RT/CMOS) chip contains the real-time clock and 64 bytes of CMOS RAM. The internal clock circuitry uses 14 bytes of this memory, and the rest is allocated for configuration and system-status information.

In addition to the 64 bytes of CMOS RAM, an 8KB CMOS RAM extension is provided for configuration and other system information.

A battery maintains voltage to the RT/CMOS RAM and 8KB CMOS RAM extension when the power supply is not in operation.

The system cover can be locked to prevent battery removal and, consequently, prevents the loss of password and configuration information.

The following table shows the RT/CMOS RAM bytes and their addresses.

Address (Hex)	RT/CMOS RAM Bytes
000–00D	Real-Time Clock Bytes
00E	Diagnostic Status Byte
00F	Shutdown Status Byte
010	Diskette Drive Type Byte
011	First Fixed Disk Drive Type Byte
012	Second Fixed Disk Drive Type Byte
013	Reserved
014	Equipment Byte
015, 016	Low and High Base Memory Bytes
017, 018	Low and High Expansion Memory Bytes
019–031	Reserved
032, 033	Configuration CRC Bytes
034	Reserved
035, 036	Low and High Usable Memory Above 1MB
037	Date Century Byte
038–3F	Reserved

Figure 3-4. RT/CMOS RAM Address Map

RT/CMOS Address Register and NMI Mask (Hex 0070)

This read/write register is used in conjunction with the port at hex 0071 to read and write the RT/CMOS RAM bytes.

Bit	Function
7	NMI Mask
6	Reserved
5–0	RT/CMOS RAM Address

Figure 3-5. RT/CMOS Address Register and NMI Mask (Hex 0070)

Warning: The operation following a write to hex 0070 must access port hex 0071; otherwise intermittent malfunctions and unreliable operation of the RT/CMOS RAM can occur.

Bit 7 When this bit is set to 1, the NMI is masked off (the NMI is disabled). This bit is set to 1 by a power-on reset. This is a write-only bit.

Bit 6 This bit is reserved.

Bits 5–0 These bits are used to select RT/CMOS RAM addresses.

RT/CMOS Data Register (Hex 0071)

This port is used in conjunction with the address register at hex 0070 to read and write the RT/CMOS RAM bytes.

Bit	Function
7–0	RT/CMOS Data

Figure 3-6. RT/CMOS Data Register (Hex 0071)

RT/CMOS RAM I/O Operations

During I/O operations to the RT/CMOS RAM addresses, interrupts must be masked to prevent other interrupt routines from changing the CMOS Address register before data is read or written.

Warning: The operation following a write to hex 0070 must access hex 0071; otherwise intermittent malfunctions and unreliable operation of the RT/CMOS RAM can occur.

Writing RT/CMOS RAM requires the following steps:

1. Write the RT/CMOS RAM address to the RT/CMOS and NMI Mask register (hex 0070).
2. Write the data to address hex 0071.
3. Write the address, hex 0D, to the RT/CMOS and NMI Mask register; this leaves hex 0070 pointing to Status Register D.
4. Read address hex 0071 to restore the RT/CMOS (see the warning above).

Reading RT/CMOS RAM requires the following steps:

1. Write the RT/CMOS RAM address to the RT/CMOS and NMI Mask register (hex 0070).
2. Read the data from address hex 0071.
3. Write the address, hex 0D, to the RT/CMOS and NMI Mask register; this leaves hex 0070 pointing to Status Register D.
4. Read address hex 0071 to restore the RT/CMOS.

Real-Time Clock Bytes (Hex 000 – 00D)

Bit definitions and addresses for the real-time clock bytes are shown in the following table.

Address (Hex)	Function	Byte Number
000	Seconds	0
001	Second Alarm	1
002	Minutes	2
003	Minute Alarm	3
004	Hours	4
005	Hour Alarm	5
006	Day of Week	6
007	Date of Month	7
008	Month	8
009	Year	9
00A	Status Register A	10
00B	Status Register B	11
00C	Status Register C	12
00D	Status Register D	13

Figure 3-7. Real-Time Clock Bytes

Note: The Setup program initializes status registers A, B, C, and D when the time and date are set. Interrupt hex 1A is the BIOS interface to read and set the time and date. It initializes the register the same way as the Setup program.

Status Register A (Hex 00A)

Bit	Function
7	Update in Progress
6–4	22-Stage Divider
3–0	Rate Selection Bits

Figure 3-8. Status Register A

Bit 7 When set to 1, this bit indicates the time-update cycle is in progress. When set to 0, it indicates the current date and time can be read.

Bits 6–4 These three divider-selection bits identify which time-base frequency is being used. The system initializes these bits to binary 010, which selects a 32.768 kHz time base. This is the only value supported by the system for proper time-keeping.

Bits 3–0 These bits allow the selection of a divider output frequency. The system initializes the rate selection bits to a binary 0110, which selects a 1.024 kHz square-wave output frequency and a 976.562-microsecond periodic interrupt rate.

Status Register B (Hex 00B)

Bit	Function
7	Set
6	Periodic Interrupt Enable
5	Alarm Interrupt Enable
4	Update-Ended Interrupt Enable
3	Square Wave Enable
2	Date Mode
1	24-Hour Mode
0	Daylight Savings Enable

Figure 3-9. Status Register B

Bit 7 When set to 0, this bit updates the cycle, normally by advancing the counts at a rate of one per second. When set to 1, this bit immediately ends any update cycle in progress, and the program can initialize the 14 time bytes without any further updates occurring until this bit is set to 0.

Bit 6 This bit is a read/write bit that allows an interrupt to occur at a rate specified by the rate and divider bits in Status Register A. When set to 1, this bit enables the interrupt. The system initializes this bit to 0.

Bit 5 When set to 1, this bit enables the alarm interrupt. The system initializes this bit to 0.

Bit 4 When set to 1, this bit enables the update-ended interrupt. The system initializes this bit to 0.

Bit 3 When set to 1, this bit enables the square-wave frequency as set by the rate-selection bits in Status Register A. The system initializes this bit to 0.

Bit 2 This bit indicates if the time-and-date calendar updates use binary or binary-coded-decimal (BCD) formats. When set to 1, this bit indicates a binary format. The system initializes this bit to 0.

Bit 1 This bit establishes if the hours byte is in the 24-hour or 12-hour mode. When set to 1, this bit indicates the 24-hour mode. The system initializes this bit to 1.

Bit 0 When set to 1, this bit enables the daylight savings time mode. When set to 0, it disables the mode, and the clock reverts to standard time. The system initializes this bit to 0.

Status Register C (Hex 00C)

Bit	Function
7	Interrupt Request Flag
6	Periodic Interrupt Flag
5	Alarm Interrupt Flag
4	Update-Ended Interrupt Flag
3–0	Reserved

Figure 3-10. Status Register C

Note: Interrupts are enabled by bits 6, 5, and 4 in Status Register B.

Bit 7 This bit is used in conjunction with bits 6, 5, and 4. When set to 1, this bit indicates that an interrupt has occurred; bits 6, 5, and 4 indicate the type of interrupt.

Bit 6 When set to 1, this bit indicates that a periodic interrupt occurred.

Bit 5 When set to 1, this bit indicates that an alarm interrupt occurred.

Bit 4 When set to 1, this bit indicates that an update-ended interrupt occurred.

Bits 3–0 These bits are reserved.

Status Register D (Hex 00D)

Bit	Function
7	Valid RAM
6–0	Reserved

Figure 3-11. Status Register D

Bit 7 This read-only bit monitors the power-sense pin. A low state of this pin indicates a loss of power to the real-time clock (dead battery). When set to 1, this bit indicates that the real-time clock has power. When set to 0, it indicates that the real-time clock has lost power.

Bits 6–0 These bits are reserved.

CMOS RAM Configuration

The following figure shows the bit definitions for the CMOS RAM configuration bytes.

Diagnostic Status Byte (Hex 00E)

Bit	Function
7	Real-Time Clock Chip Power
6	Configuration Record and Checksum Status
5	Incorrect Configuration
4	Memory Size Miscompare
3	Fixed Disk Controller/Drive C Initialization Status
2	Time Status Indicator
1	Adapter Configuration Miscompare
0	Adapter ID Time-Out

Figure 3-12. Diagnostic Status Byte

Bit 7 When set to 1, this bit indicates the real-time clock chip lost power.

Bit 6 When this bit is set to 1, the checksum is incorrect.

Bit 5 This is a check, at power-on time, of the Equipment byte. When set to 1, the configuration information is incorrect. Power-on checks require that at least one diskette drive be installed (bit 0 of the Equipment byte, hex 014, is set to 1).

Bit 4 When set to 1, this bit indicates the power-on check determined that the memory size is not the same as in the configuration record.

Bit 3 When set to 1, this bit indicates that the controller or drive C failed initialization, which prevents the system from attempting a power-on reset.

Bit 2 When set to 0, this bit indicates the time is valid. When set to 1, this bit indicates the time is invalid.

Bit 1 This bit indicates if the installed adapters match the configuration information. When this bit is set to 1, the adapters do not match the configuration information.

Bit 0 When set to 1, this bit indicates a time-out occurred while an adapter ID was being read.

Shutdown Status Byte (Hex 00F): This byte is defined by the power-on diagnostic programs.

Diskette Drive Type Byte (Hex 010): This byte indicates the type of diskette drive installed.

Bit	Function
7–4	First Diskette Drive Type
3–0	Second Diskette Drive Type

Figure 3-13. Diskette Drive Type Byte

Bits 7–4 These bits indicate the first diskette drive type, as shown in the following table.

Bits 7 6 5 4	Function
0 0 0 0	No Drive Present
0 0 0 1	Double-sided Diskette Drive (48 tracks per inch, 360KB)
0 0 1 0	Double-sided Diskette Drive (80 tracks per inch, 1.2MB)
0 0 1 1	High-capacity Diskette Drive (720KB)
0 1 0 0	High-density Diskette Drive (1.44MB)
All combinations that are not shown are reserved.	

Figure 3-14. Diskette Drive Type Byte (Bits 7–4)

Bits 3–0 These bits indicate the second diskette drive type, as shown in the following table.

Bits 3 2 1 0	Function
0 0 0 0	No Drive Present
0 0 0 1	Double-sided Diskette Drive (48 tracks per inch, 360KB)
0 0 1 0	Double-sided Diskette Drive (80 tracks per inch, 1.2MB)
0 0 1 1	High-capacity Diskette Drive (720KB)
0 1 0 0	High-density Diskette Drive (1.44MB)
All combinations that are not shown are reserved.	

Figure 3-15. Diskette Drive Type Byte (Bits 3–0)

First Fixed Disk Drive Type Byte (Hex 011): This byte defines the type of the first fixed disk drive (drive C). Hex 00 indicates that a fixed disk drive is *not* installed.

Second Fixed Disk Drive Type Byte (Hex 012): This byte defines the type of the second fixed disk drive (drive D). Hex 00 indicates that a fixed disk drive is *not* installed.

Note: For more information about fixed disk drive types, refer to the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference.*

Reserved Byte (Hex 013): This byte is reserved.

Equipment Byte (Hex 014): The equipment byte, for the power-on diagnostic tests, defines the basic equipment in the system.

Bit	Function
7, 6	Number of Diskette Drives
5, 4	Display Operating Mode
3, 2	Reserved
1	Math Coprocessor Presence
0	Diskette Drive Presence

Figure 3-16. Equipment Byte

Bits 7, 6 These bits indicate the number of diskette drives installed, as shown in the following table.

Bits 7 6	Number of Diskette Drives
0 0	One Drive
0 1	Two Drives
1 0	Three Drives
1 1	Reserved

Figure 3-17. Equipment Byte (Bits 7, 6)

Bits 5, 4 These bits indicate the operating mode of the display attached to the video port, as shown in the following table.

Bits 5 4	Display Operating Mode
0 0	Reserved
0 1	40-Column Mode
1 0	80-Column Mode
1 1	Monochrome Mode

Figure 3-18. Equipment Byte (Bits 5, 4)

Bits 3, 2 These bits are reserved.

Bit 1 When set to 1, this bit indicates that a math coprocessor is installed.

Bit 0 When set to 1, this bit indicates that a diskette drive is installed.

Low and High Base Memory Bytes (Hex 015 and 016): These bytes define the amount of memory below the 640KB address space.

The value from these bytes represents the number of 1KB blocks of base memory. For example, hex 0280 is equal to 640KB. The low byte is hex 15; the high byte is hex 16.

Low and High Expansion Memory Bytes (Hex 017 and 018): These bytes define the amount of memory above the 1MB address space.

The value from these bytes represents the number of 1KB blocks of expansion memory. For example, hex 0800 is equal to 2048KB. The low byte is hex 17; the high byte is hex 18.

Reserved Bytes (Hex 019 through 031): These bytes are reserved.

Configuration CRC Bytes (Hex 032 and 033): These bytes contain the cyclic-redundancy-check (CRC) data for bytes hex 010 through hex 031 of the 64-byte CMOS RAM. The low byte is hex 33; the high byte is hex 32.

Reserved Byte (Hex 034): This byte is reserved.

Low and High Useable Memory Bytes (Hex 035 and 036): These bytes define the total amount of continguous useable memory from 1MB to 16MB.

The hexadecimal values in these bytes represent the number of 1KB blocks of useable memory. For example, hex 0800 is equal to 2048KB. The low byte is hex 35; the high byte is hex 36.

Date Century Byte (Hex 037): Bits 7 through 0 of this byte contain the BCD value for the century. Refer to the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference* for information about reading and setting this byte.

Reserved Bytes (Hex 038 through 03F): These bytes are reserved.

Miscellaneous System Functions

Nonmaskable Interrupt

The nonmaskable interrupt (NMI) signals the system microprocessor that a parity error, a channel check, a system channel time-out, or a system Watchdog time-out has occurred. The NMI stops all arbitration on the bus until bit 6 of the Arbitration register (I/O address hex 0090) is set to 0. This can result in lost data or an overrun error on some I/O devices. The NMI masks all other interrupts and the IRET instruction restores the interrupt flag to the state it was in prior to the interrupt. A system reset causes a reset of the NMI.

The NMI requests that are caused by channel check and system-board memory-parity errors are subject to mask control with the NMI mask bit in the RT/CMOS Address register. The Watchdog Timer and system channel time-out are not masked by this bit. (See "RT/CMOS Address Register and NMI Mask (Hex 0070)" on page 3-6). The power-on default of the NMI mask is 1 (NMI disabled). Prior to enabling the NMI after a power-on reset, the parity check and channel check state are initialized by the POST.

Warning: The operation following a write to hex 0070 should access port hex 0071; otherwise intermittent malfunctions and unreliable operation of the RT/CMOS RAM can occur.

System Control Port B (Hex 0061)

Bit definitions for the read and write functions of this port are shown in the following tables.

Bit	Function
7	Reset Timer 0 Output Latch (IRQ0)
6–4	Reserved
3	Enable Channel Check
2	Enable Parity Check
1	Speaker Data Enable
0	Timer 2 Gate to Speaker

Figure 3-19. System Control Port B (Hex 0061) – Write

Bit	Function
7	Memory Parity Check
6	Channel Check
5	Timer 2 Output
4	Toggles with Each Refresh Request
3	Enable Channel Check
2	Enable Parity Check
1	Speaker Data Enable
0	Timer 2 Gate to Speaker

Figure 3-20. System Control Port B (Hex 0061) – Read

Bit 7 Setting this bit to 1 resets IRQ0. Reading this bit as a 1 indicates a memory parity check has occurred.

Bit 6 Reading this bit as a 1 indicates a channel check has occurred.

Bit 5 This bit indicates the condition of the timer/counter 2 'output' signal.

Bit 4 This bit toggles for each refresh request.

Bit 3 Setting this bit to 0 enables channel check. Setting this bit to 1 disables channel check and clears the channel check latch.

Bit 2 Setting this bit to 0 enables system board memory parity check. This bit is set to 1 during a power-on reset. To clear a parity error, set this bit to 1 and then to 0.

Bit 1 Setting this bit to 1 enables speaker data.

Bit 0 Setting this bit to 1 enables the timer 2 gate.

System Control Port A (Hex 0092)

Bit	Function
7, 6	Fixed-Disk Activity Light
5	Reserved
4	Watchdog Timer Status
3	Security Lock Latch
2	Reserved = 0
1	Alternate Gate A20
0	Alternate Hot Reset

Figure 3-21. System Control Port A (Hex 0092)

Bits 7, 6 These bits control the fixed-disk activity light. Setting either bit to 1 turns the fixed-disk activity light on. Setting both bits to 0 turns the light off. The power-on reset condition of each bit is 0.

Bit 5 This bit is reserved.

Bit 4 This read-only bit indicates the watchdog-timer status. When this bit is set to 1, a watchdog time-out has occurred. For more information about the Watchdog Timer, refer to System Timers in *Hardware Interface Technical Reference — Common Interfaces.*

Bit 3 This bit provides the security lock for the secured area of RT/CMOS. Setting this bit to 1 electrically locks the 8-byte, power-on password. Once this bit is set by POST, it can only be cleared by turning the system off.

Bit 2 This bit is reserved.

Bit 1 This bit is ORed with bit 1 in the keyboard output port to enable the 'address 20' signal (A20). When both bits are set to 0, A20 is disabled and set to 0. This bit is set to 0 during POST. (See the "Keyboard Controller" section in the *Hardware Interface Technical Reference.*)

Bit 0 This bit provides an alternate method of resetting the system microprocessor. This alternate method supports operating systems requiring faster operation than was provided on the IBM Personal Computer AT*. Resetting the system microprocessor is used to switch the microprocessor from the protected mode to the real address mode. The alternate reset takes 13.4 microseconds.

This bit is set to 0 either by a system reset or a Write operation. When a Write operation changes this bit from 0 to 1, the alternate reset pin is pulsed high for 100 to 125 nanoseconds. The reset occurs after a minimum delay of 6.72 microseconds. While the reset is occurring, the latch remains set so that POST can read this bit. If the bit is 0, POST assumes the system was just powered on. If the bit is 1, POST assumes a switch from the protected mode to the real mode has taken place.

---

* Personal Computer AT is a trademark of the International Business Machines Corporation.

Power-On Password

RT/CMOS RAM has eight bytes reserved for the power-on password and its check character. The eight bytes are initialized to hex 00. The microprocessor can only access these bytes during power-on self-test (POST). After POST is completed, if a power-on password is installed, the password bytes are locked and cannot be accessed by a program. A power-on password can be from 1 to 7 characters.

During power-on password installation, the password (1 to 7 keyboard scan codes), is stored in the security space.

Power-on password installation is a function of a program contained on the Reference Diskette. Once the power-on password utility has been installed, the password can be changed only during the POST. When the new power-on password is installed, changed, or removed, the password is not visible on the display.

The system unit cover can be physically locked to prevent unauthorized access to the battery. This helps prevent unauthorized battery removal and loss of power-on password and configuration information.

For information about the keyboard password, see the "Keyboard and Auxiliary Device Controller" section in the *Hardware Interface Technical Reference*.

Hardware Compatibility

The Model 95 maintains many of the interfaces used by the IBM Personal Computer AT. In most cases command and status organization of these interfaces is maintained.

The functional interfaces for the Model 95 are compatible with the following interfaces:

- The Intel** 8259 interrupt controllers (without edge triggering).
- The Intel 8253 timers driven from 1.193 MHz (timer 0 and 2 only).
- The Intel 8237 DMA controller address/transfer counters, page registers and status fields only. The Command and Request registers are not supported. The rotate and mask functions are not supported. The Mode register is partially supported.
- The NS16450 serial port.
- The Intel 8088, 8086, 80286, and 80386 microprocessors.
- The Intel 8272 diskette drive controller.
- The Motorola** MC146818 Time of Day Clock command and status (CMOS reorganized).
- The keyboard interface at address hex 0060.
- Display modes supported by the IBM Monochrome Display and Printer Adapter, the IBM Color/Graphics Monitor Adapter, and the IBM Enhanced Graphics Adapter.
- The parallel printer ports (Parallel 1, Parallel 2, and Parallel 3) in compatibility mode.
- Generally compatible with the Intel 8087, 80287, and 80387 math coprocessors.

Error Codes

POST returns a message in the form of a multiple-character code to indicate the type of test that failed. Two formats are used for these error messages:

- An eight-character error code for all errors *not* related to SCSI devices
- An 11 character code for all errors related to SCSI devices
  - The first three characters identify the device type
  - The next eight characters identify specifics of the error.

An information panel is used to display the first eight characters of an error message in case the system is operating without a video display. Only the code for the first error found during POST is displayed on the information panel; subsequent errors are indicated only on the video display (the information panel will continue to display the first error).

The error code consists of two 4-character elements, the major error code and the minor error code. The following figure gives the failure indicated with the associated error code.

Note: For information on unlisted error codes, contact Developer Assistance at 1-800-IBM-7763 (this number is for developers who are registered with IBM).

Major Code	Minor Code	Description
0001		System Bus Error - system board or processor complex
	0100	Unexpected interrupt
	0200	Timer failure
	0300	Timer interrupt failure
	0400	Protected mode failure
	0500	Last keyboard command not accepted
	0600	Converting logic test
	0700	NMI test failed
	0800	Timer bus-test failed
	10xx	System board memory parity error
	11xx	I/O channel check error
	1200	Watchdog timeout
	1300	DMA arbitration timeout
	1400	Channel feature card ROM checksum error
	1500	ROM checksum error or DMA error
	1600	System board port read/write failure
	1800	System board parity or L2-cache error during previous power-on
	2000	Microprocessor test error
	6100	Dead battery
	6300	Clock not updating
	6400	Memory configuration error
	6500	Adapter ID mismatch
	6600	Adapter busy error
	6700	Clock not updating
	6800	Processor complex configuration error
	7000	ASCII setup conflict error
	7100	Rolling bit test failure on CMOS shutdown address byte
	7200	Rolling bit test failure on NVRAM diagnostic byte
	7300	Bad CMOS/NVRAM checksum
	7400	Bad configuration

Figure 3-22 (Part 1 of 2). POST Error Message Table

Major Code	Minor Code	Description
0002		Memory errors
	0xxx	Memory error
	1xxx	Unrecoverable error in first 1MB
	2100	ROM to RAM remap error
	25xx	Unsupported memory-type installed or memory pair mismatch
0003		Keyboard
	0100	Keyboard error
	0200	Keyboard locked
	0300	Keyboard to system board interface error
	0400	Keyboard clock high
	0500	No keyboard +5V
0004	0100	Display error
0005	0100	Display error
0006		Diskette
	0100	Diskette drive or controller error
	0200	Diskette IPL boot record not valid
	0604	Non-media sense
0011	0100	Async chip error
0017		Fixed disk error
	8000	Fixed disk 0 failed
	8100	Fixed disk 1 failed
	8200	Disk controller error
	9000	Hard file 0 error
	9100	Hard file 1 error
0024	0100	System board video error
0086		Mouse
	0100	System bus error - keyboard/pointing device interface
	0200	Pointing device error
	0300	Pointing device or system bus error
0129		Cache
	0x00	Cache error
I999		Special instruction codes
	00xx	Initial microcode load (IML) error
	03xx	No bootable device
	04xx	IML to system mismatch

Figure 3-22 (Part 2 of 2). POST Error Message Table

Notes:

Section 4. Processor Complex

Type 1 4-3
Type 2 4-5
I/O Address Map 4-7
Optional Cache 4-7
Micro Channel 4-7
  Central Arbiter 4-8
    Arbitration Bus Priority Assignments 4-8
  Central Arbiter Programming 4-9
Memory Subsystem 4-11
  Memory Presence Detect 4-11
  Memory Controller Registers 4-12
    Memory Bank Base Address Register (Index Hex 90—9F) 4-15
    Split Address Register (Index Hex A0) 4-15
    Memory Encoding Register (Index Hex A1) 4-16
    Cache/Timer Control Register (Index Hex A2) 4-18
    Boundary 0 Cache Control Register (Index Hex A3) 4-19
    Boundary 1 Cache Control Registers (Index Hex A4 and A5) 4-20
    Memory Bank Enable Register 1 (Index Hex A6) 4-21
    Memory Bank Enable Register 2 (Index Hex A7) 4-22
    System Bus Request Timer Register (Index Hex A8) 4-23
    System Bus Hold Timer Register (Index Hex A9) 4-24
    System Bus Idle Timer Register (Index Hex AA) 4-24
    Parity Trap Register (Hex 00E1) 4-25
    Cache Control Register (Hex 00E2) 4-26
    Cache Status Register (Hex 00E3) 4-27
Processor Cache Address Map 4-28
  Read-Only Memory Subsystem 4-29

© Copyright IBM Corp. 1990

The processor complex provides certain devices and features that work in conjunction with those devices and features provided by the system board. In addition, the processor complex determines system performance and certain channel timings. This section describes these aspects of the processor complex.

A system can have either a Type 1 or Type 2 processor complex. The major difference between the two types is speed. Programs can identify the type by reading the model and submodel bytes and the BIOS revision level (Interrupt hex 15, function code AH = hex C0). The following table shows these bytes and the processor complex types.

Model Byte	Submodel Byte	Revision Level	Speed	Processor Complex
F8	14	00	25 MHz	Type 1
F8	16	00	33 MHz	Type 2

Figure 4-1. Model and Submodel Bytes

Refer to the *IBM Personal System/2 and Personal Computer BIOS Interface Technical Reference* for a listing of other systems, and check the supplements section for updates to that listing.

Type 1

The following are the devices and features for the Type 1 processor complex.

Device	Type	Features
Microprocessor		80486—25 MHz 32-bit address and 32-bit data interface
Cache	---	8KB, 4-way, write-through, internal cache Optional L2 cache card interface
ROM Subsystem	---	128KB physical ROM
RAM Subsystem	---	Eight connectors support 4 to 32MB, interleaved Expandable on the system board only
DMA Controller	1	Eight independent DMA channels Single or burst transfers and read verification
Math Coprocessor	---	Math coprocessor functions included in 80486

Figure 4-2. Type 1 Processor Complex Feature and Devices

The following shows the performance and other timing information for the Type 1 processor complex. These timings affect the channel performance for operations involving the processor complex or system board memory.

Device		Cycle Time (ns)
Microprocessor (25 MHz — 40 ns Clock):		
Access to Channel:		
Default Transfer Cycle		240
Extended Transfer Cycle		320
Refresh Rate (Typically performed every 15.1 μs)		500 (min)
DMA Controller (10 MHz — 100 ns Clock):		
Single Transfer:	300 + I/O Access + Memory Access	
Burst Transfers:	300 + (I/O Access + Memory Access)N *	
System Board Memory Access	300	
Default Transfer Cycle	200	
Synchronous Extended Transfer Cycle	300	
* N is the number of transfers in the burst transfer.		

Figure 4-3. System Performance Specifications, Type 1 Processor Complex

Device	Cycle Time (ns) per Memory Type		
	70 ns	80 ns	85 ns
Microprocessor (25 MHz – 40 ns Clock):			
Memory Read (Cache Hit)	40	40	40
Memory Read (Cache Miss, Page Hit, Burst)	320	320	400
Memory Read (Cache Miss, Page Miss, Burst)	440	480	560
Memory Write (Page Hit)	160	160	200
Memory Write (Page Miss)	280	320	360
Microprocessor – with Optional L2 Cache			
Memory Read (Cache Hit, Optional Cache Hit)	40	40	40
Memory Read (Cache Miss, Optional Cache Hit, Burst)	200	200	200
Memory Read (Cache Miss, Optional Cache Miss, Page Hit, Burst)	320	320	400
Memory Read (Cache Miss, Optional Cache Card Miss, Page Miss, Burst)	440	480	560
Memory (Page Hit)	160	160	200
Memory Write (Page Miss)	280	320	360
Bus Master Access to System Board RAM			
Page Hit (minimum)	300	300	300
Page Miss (minimum)	300	300	300

Figure 4-4. Memory Performance Specifications, Type 1 Processor Complex

Type 2

The following are the devices and features for the Type 2 processor complex.

Device	Type	Features
Microprocessor		80486—33 MHz 32-bit address and 32-bit data interface
Cache	—	8KB, 4-way, write-through, internal cache Optional cache card interface
ROM Subsystem	—	128KB physical ROM
RAM Subsystem	—	Eight connector support 4 to 32MB interleaved Expandable on the system board only
DMA Controller	1	Eight independent DMA channels Single or burst transfers and read verification
Math Coprocessor	—	Math coprocessor functions included in 80486

Figure 4-5. Type 2 Processor Complex Feature and Devices

The following shows the performance and other timing information for the Type 2 processor complex. These timings affect the channel performance for operations involving the processor complex or system board memory.

Device		Cycle Time (ns)
Microprocessor (33 MHz — 30 ns Clock):		
Access to Channel:		
Default Transfer Cycle		240
Extended Transfer Cycle		320
Refresh Rate (Typically performed every 15.1 μs)		500 (min)
DMA Controller (10 MHz — 100 ns Clock):		
Single Transfer:	300 + I/O Access + Memory Access	
Burst Transfers:	300 + (I/O Access + Memory Access)N *	
System Board Memory Access	300	
Default Transfer Cycle	200	
Synchronous Extended Transfer Cycle	300	

* N is the number of transfers in the burst transfer.

Figure 4-6. System Performance Specifications, Type 2 Processor Complex

Device	Cycle Time (ns) per Memory Type		
	70 ns	80 ns	85 ns
Microprocessor (33 MHz – 30 ns Clock):			
Memory Read (Cache Hit)	30	30	30
Memory Read (Cache Miss, Page Hit, Burst)	240	300	300
Memory Read (Cache Miss, Page Miss, Burst)	360	420	420
Memory Write (Page Hit)	120	150	150
Memory Write (Page Miss)	240	270	270
Microprocessor – with Optional L2 Cache			
Memory Read (Cache Hit, Optional Cache Hit)	30	30	30
Memory Read (Cache Miss, Optional Cache Hit, Burst)	150	150	150
Memory Read (Cache Miss, Optional Cache Miss, Page Hit, Burst)	240	300	300
Memory Read (Cache Miss, Optional Cache Card Miss, Page Miss, Burst)	360	420	420
Memory Write (Page Hit)	120	150	150
Memory Write (Page Miss)	240	270	270
Bus Master Access to System Board RAM			
Page Hit (minimum)	300	300	300
Page Miss (minimum)	300	300	300

Figure 4-7. Memory Performance Specifications, Type 2 Processor Complex

I/O Address Map

The following shows the I/O address for devices located on the processor complex.

Hex Addresses	Device
0000–001F	DMA Controller (0–3)
0081–0083, 0087	DMA Page Registers (0–3)
0089–008B, 008F	DMA Page Registers (4–7)
0090	Central Arbitration Control Point
00C0–00DF	DMA Controller (4–7)
00E0	Memory Control Register Select
00E1	Parity Trap Register
00E2	Cache Control Register
00E3	Cache Status Register
00E4	Memory Control Data Read/Write

Figure 4-8. Processor Complex I/O Address Map

Optional Cache

The Type 1 and Type 2 processor complexes have a 160-pin connector for an optional cache card (L2 cache). The optional cache card provides secondary cache and cache controller for the system microprocessor.

The optional cache card provides the system microprocessor with an additional 256KB cache. It operates as 2-way set-associative, write-through cache with parity. The primary cache in the 80486 must be enabled to operate the secondary cache. The L2 cache is controlled through the Cache Control Register (Hex 00E2), and its status is indicated in the Cache Status Register (Hex 00E3).

Micro Channel

This section describes the implementation of the Micro Channel architecture on Model 95 systems. For general Micro Channel information, refer to *Hardware Interface Technical Reference — Architectures.*

Note: The system does not support adapters that use the matched memory extension.

Central Arbiter

The central arbitration control point gives intelligent subsystems on the channel the ability to share and control the system. It allows burst data transfers and prioritization of control between devices. This arbiter supports up to 16 arbitrating devices.

Arbitration Bus Priority Assignments

The following table shows the assignment of arbitration levels. The functions with the lowest arbitration level have the highest priority.

ARB Level	Primary Assignment
-2	Memory Refresh
-1	NMI
0	DMA Channel 0 (Programmable to any arbitration level)
1	DMA Channel 1
2	DMA Channel 2
3	DMA Channel 3
4	DMA Channel 4 (Programmable to any arbitration level)
5	DMA Channel 5
6	DMA Channel 6
7	DMA Channel 7
8–B	Available
C	SCSI Adapter
D, E	Available
F	System Microprocessor

Figure 4-9. Arbitration Bus Priority Assignments

Note: Devices designed for arbitration level 0 or 1 should have limited bandwidth or short bursts so diskette overruns can be prevented or recovered by retry operations. The diskette drive controller, on arbitration level 2, can be held inactive by devices on levels 0 and 1, by a refresh operation, and by the previous controlling master. The diskette drive controller should not be held inactive for more than 12 microseconds to prevent overrun.

Nonmaskable interrupt (NMI) service is executed at a priority level higher than 0, called -1. Memory refresh is prioritized at -2, two levels higher than 0. Levels -1 and -2 are reached on the system board only, while the 'arbitrate/-grant' signal is in the arbitrate state.

When the central arbitration control point receives a level -1 request (NMI, a system-board internal signal), it activates -PREEMPT, waits for the end of transfer, and then places ARB/-GNT in the arbitrate state, which denies channel activity to arbitrating devices. The central arbitration control point gives the grant to the level -1 request, and holds ARB/-GNT in the arbitrate state until the operation is complete and the NMI is reset.

Central Arbiter Programming

The central arbitration control point provides access to programmable options through the Arbitration register, which is accessed at I/O address hex 0090. The bits are defined differently for read and write operations, as shown in the following tables.

Bit	Definition
7	Enable System Microprocessor Cycle
6	Arbitration Mask
5	Enable Extended Arbitration
4–0	Reserved

Figure 4-10. Arbitration Register (Hex 0090) – Write

Bit	Definition
7	Enable System Microprocessor Cycle
6	Arbitration Masked by NMI
5	Bus Time-out
4	Reserved
3–0	Value of Arbitration Bus During Previous Grant State

Figure 4-11. Arbitration Register (Hex 0090) – Read

Bit 7 Setting this bit to 1 enables system microprocessor cycles during arbitration cycles. This bit can be set to 0 if an arbitrating device requires total control of the channel bandwidth. This bit is set to 0 by a system reset.

Reading this bit as a 1 indicates system microprocessor cycles are enabled during arbitration.

Bit 6 Setting this bit to 1 causes the central arbitration control point to enter the arbitration state. The system microprocessor controls the channel until this bit is reset to 0. This bit is set to 0 by a system reset.

Reading this bit as a 1 indicates that an NMI has occurred and has masked arbitration.

Warning: This bit should be set to 1 only by diagnostic routines and system error-recovery routines.

Bit 5 Setting this bit to 1 enables extended arbitration. The minimum arbitration cycle is 300 nanoseconds; this bit extends that minimum cycle to 600 nanoseconds. This bit is set to 0 during a system reset.

Reading this bit as a 1 indicates that a bus time-out has occurred, and resets bit 6 in this register to 0.

Bit 4 This bit is reserved and should be 0.

Bits 3–0 These bits are undefined for a write operation and should be set to 0.

Reading these bits returns the arbitration level of the arbiter controlling the channel during the most recent grant state. This information allows the system microprocessor to determine the arbitration level of the device that caused a bus time-out.

Memory Subsystem

Memory on the system board is parity-checked, interleaved random-access memory and must be installed in matched pairs: one in an A connector and one in the matching B connector (for memory connector information, see "System Board Memory Connectors" on page 3-3). The memory in each pair must be the same speed and size. Memory on adapters in the channel are not mapped into the cacheable address space of the system microprocessor.

System board memory is designed so that the system microprocessor can access memory while a bus master has control of the channel. The memory controller manages this dual access by interleaving the microprocessor and bus master accesses. In addition, system board memory is organized into pairs of banks so that the microprocessor can perform burst cycles with two-way interleaved accesses. Because of this organization, the microprocessor can perform zero-wait-state transfers for every other 32-bit transfer in the burst cycle.

Memory Presence Detect

The presence-detect signals are used by the system to determine size and speed of memory on the memory card. The pins are either not connected (1), or connected to ground (0).

A set of four bits shows the state of all presence-detect signals for a specific memory connector; the bit is 0 when the signal is connected to ground. The following figure shows the information indicated by these bits as well as the bank size of the memory card. All combinations not shown are reserved.

Presence Detect 3 2 1 0	Memory Card Definition	Bank Size
0 1 0 0	4MB 70-ns Memory	4MB
0 0 0 0	4MB 80-ns Memory	4MB
1 1 0 0	2MB 70-ns Memory	1MB
1 0 0 1	2MB 80-ns Memory	1MB
0 1 0 1	2MB 85-ns Memory	1MB
0 1 1 0	1MB 85-ns Memory	1MB
1 1 1 1	Connector is Empty	

Figure 4-12. Presence-Detect Bits

Memory Controller Registers

The memory controller, located on the processor complex, has several internal registers that are used to control the memory in the memory connectors on the system board. The memory controller manages this memory in banks of 1MB or 4MB. The size of the bank determines its granularity.

Connectors A1 through A4 contain banks 0 through 7; connectors B1 through B4 contain banks 8 through hex F.

Each internal register is accessed by writing its internal address (index) to the Memory Controller Address register at hex 00E0 and then accessing the data through the Memory Controller Data register at hex 00E4. These registers can be accessed only from the system microprocessor.

I/O port 00E2 controls the internal 80486 cache and the optional cache functions. I/O port 00E3 is used for cache status and memory bank parity error determination. I/O port 00E1 is also used for memory bank parity error determination.

All registers in the memory controller are 8-bit read/write registers. The following shows the function and index for each internal register. These registers are programmed during POST to configure the memory controller according to the type and amount of memory and other system requirements.

Index	Register Function
90	Memory Bank 0 Base Address
91	Memory Bank 1 Base Address
92	Memory Bank 2 Base Address
93	Memory Bank 3 Base Address
94	Memory Bank 4 Base Address
95	Memory Bank 5 Base Address
96	Memory Bank 6 Base Address
97	Memory Bank 7 Base Address
98	Memory Bank 8 Base Address
99	Memory Bank 9 Base Address
9A	Memory Bank A Base Address
9B	Memory Bank B Base Address
9C	Memory Bank C Base Address
9D	Memory Bank D Base Address
9E	Memory Bank E Base Address
9F	Memory Bank F Base Address
A0	Split Address
A1	Memory Encoding
A2	Cache/Timer Control
A3	Boundary 0 Cache Control
A4	Boundary 1 Cache Control (Low Address)
A5	Boundary 1 Cache Control (High Address)
A6	Memory Bank Enable 1
A7	Memory Bank Enable 2
A8	System Bus Request Timer
A9	System Bus Hold Timer
AA	System Bus Idle Timer

Figure 4-13. Memory Controller Register Assignments

The 16 Memory Bank Base Address registers and individual bits in the two Bank Enable registers control the enabling and mapping of all memory on the system board. Individual bits in the Parity Trap register (hex 00E1) indicate whether the respective bank has detected a parity error.

The following figure shows the relationship of each of the control or status registers to the bank of memory that it represents. The error-indicator bit is in the Parity Trap register (hex 00E1).

Bank Number	Index of Memory Bank Address Register	Memory Bank Enable Register and Bit	Error Indicator Bit
0	90	Register 1, Bit 0	0
1	91	Register 1, Bit 1	1
2	92	Register 1, Bit 2	2
3	93	Register 1, Bit 3	3
4	94	Register 1, Bit 4	4
5	95	Register 1, Bit 5	5
6	96	Register 1, Bit 6	6
7	97	Register 1, Bit 7	7
8	98	Register 2, Bit 0	0
9	99	Register 2, Bit 1	1
A	9A	Register 2, Bit 2	2
B	9B	Register 2, Bit 3	3
C	9C	Register 2, Bit 4	4
D	9D	Register 2, Bit 5	5
E	9E	Register 2, Bit 6	6
F	9F	Register 2, Bit 7	7

Figure 4-14. Memory Control

Memory Bank Base Address Register (Index Hex 90 – 9F)

Each bank of memory has an 8-bit, read/write register that determines the base address for that bank. The banks can be set to respond to any starting 1MB address from 0 to 255MB. 4MB memory cards must be on 4MB address boundaries.

The 16 Base Address registers are at indexes hex 90 to 9F.

Bit	Function
7	Address Bit A27
6	Address Bit A26
5	Address Bit A25
4	Address Bit A24
3	Address Bit A23
2	Address Bit A22
1	Address Bit A21
0	Address Bit A20

Figure 4-15. Memory Bank Base Address Register (Index Hex 90 – 9F)

Split Address Register (Index Hex A0)

This 8-bit, read/write register at index hex A0 determines the starting address for the split memory block. The split-memory block can be positioned at any 1MB boundary from 1MB to 255MB; it must not be positioned below 1MB.

The bit definitions and functions are described below:

Bit	Function
7	Split Address Bit A27
6	Split Address Bit A26
5	Split Address Bit A25
4	Split Address Bit A24
3	Split Address Bit A23
2	Split Address Bit A22
1	Split Address Bit A21
0	Split Address Bit A20

Figure 4-16. Split Address Register (Index Hex A0)

Bits 7 – 0 These bits define the starting address of the split-memory block. When split memory is enabled, these bits determine its starting address. The starting location can be at any 1MB boundary from 1MB to 255MB. This register must not be set to 00.

Memory Encoding Register (Index Hex A1)

This 8-bit, read/write register at index hex A1 controls miscellaneous functions within the memory controller.

Bit	Function
7	-System Bus Enable
6	-Reserved
5	-Disable ROM Space Decode
4	-Lock
3	-Enable Split
2	-640
1	ROM enable
0	-Enable Planar Parity Check

Figure 4-17. Memory Encoding Register (Index Hex A1)

Bit 7 This bit is used to isolate system-board memory from the channel during initialization of the memory controller. When the bit is set to 1, channel accesses to system board memory are disabled and the memory controller ignores all channel requests. When the bit is set to 0, the channel is enabled.

Bit 6 This bit is reserved.

Bit 5 This bit controls the memory-controller decode of the first 4096 locations in system ROM space from 000E0000 to 000E0FFF. When the bit is set to 0, the decode of these addresses is disabled; when set to 1, the memory controller selects ROM or system RAM for all addresses from hex 000E0000 to 000FFFFF and from hex FFFE0000 to FFFFFFFF.

While this bit is 0, accesses to addresses hex 000E0000 to 000E0FFF are not directed to the space allocated for system ROM code and can be decoded on the channel.

Bit 4 This bit determines whether locked-access operations from the system microprocessor are enabled. When this bit is set to 0, lock is enabled and microprocessor cycles are performed as follows:

- When the system microprocessor owns the channel and is accessing memory with the 'lock' signal active, the microprocessor must complete the entire operation before another master can be granted control of the channel. This includes the DMA controller.

- When a bus-master owns the channel and the microprocessor tries to run a locked cycle operation, the microprocessor waits until the bus master releases the channel.

Bit 3 This bit determines if the split-memory block is assigned addresses or is disabled. The top 128KB of the first 1MB is always mapped into the ROM address space and the code in ROM is copied into this space.

When this bit is set to 0, the split-memory block is enabled. The split-memory block size is dependent on the value of the 640 bit. Its address is determined in Memory Encoding Register 2. When this bit is set to 1, the split-memory block is disabled.

Bit 2 This bit determines where the first active 1MB of memory is split. When this bit is set to 0, the system maps 640KB of the first 1MB to addresses hex 00000000 to 0009FFFF. The split-memory block, 256KB, is mapped to the address specified in the Split Address register.

When this bit is set to 1, the system maps 512KB of the first 1MB to addresses hex 00000000 to 0007FFFF. The split-memory block, 384KB, is mapped to the address specified in Split Address register. The addresses hex 00080000 to 0009FFFF are unassigned.

Note: The memory address space from the memory split to 896KB is not cached.

Bit 1 This bit determines how addresses hex 000E0000 to 000FFFFF are assigned.

When this bit is set to 1, ROM is enabled and the read-access addresses are assigned to ROM; the write-access addresses are assigned to RAM. When this bit is set to 0, ROM is disabled and read accesses are assigned to RAM; write accesses are disabled while ROM is disabled.

Note: This bit should be set to 0 because system performance can be degraded when ROM is enabled.

Bit 0 When set to 0, this bit enables parity checking of system board memory. Setting this bit to 1 clears the pending parity error. For compatibility with other systems, enabling and disabling of parity checking should also be done through System Control Port B at address hex 0061

(see the register definition on "System Control Port B (Hex 0061)" on page 3-17).

Note: When a parity error occurs, clear the error using this bit then clear the error using the bit in System Control Port B.

Cache/Timer Control Register (Index Hex A2)

This 8-bit, read/write register at index hex A2 controls the cache and timer functions.

Bit	Function
7	-Enable Processor Preempt Timers
6–3	Reserved
2	-Enable Boundary 1 Check
1	-Enable Boundary 0 Check
0	-Enable Cache

Figure 4-18. Cache/Timer Control Register (Index Hex A2)

Bit 7 This bit controls the timers used to allocate bus cycles to the microprocessor. In system environments with several bus-master-type adapters, the system microprocessor can be prevented from gaining control of the channel for long periods of time. When this bit is set to 0, the system-bus timer registers (index hex A8 through AA) are enabled, and the system microprocessor can be ensured a percentage of the channel bandwidth.

Bits 6–3 Reserved

Bit 2 This bit controls the boundary check for a noncacheable address from 16MB to 256MB. When the bit is set to 0, boundary checks are enabled. For detail, see "Boundary 1 Cache Control Registers (Index Hex A4 and A5)" on page 4-20.

Bit 1 This bit controls the boundary check for a noncacheable address below 16MB. When the bit is set to 0, boundary checks are enabled. For detail, see "Boundary 0 Cache Control Register (Index Hex A3)" on page 4-19.

Bit 0 This bit enables or disables the internal cache in the system microprocessor and the external cache, if it is installed. When this bit is set to 0, both caches are enabled. When this bit is set to 1, both caches are disabled.

After disabling the caches and before enabling them, they should be flushed (see "Cache Control Register (Hex 00E2)" on page 4-26).

Boundary 0 Cache Control Register (Index Hex A3)

This 8-bit, read/write register at index hex A3 contains the upper-range and lower-range control bits. This register determines the memory range, from 1MB up to 16MB, that will not be cached.

When the cache is enabled, memory below 1MB is cached except for the following:

- The addresses from hex 000A0000 to 000DFFFF are never cached.
- The addresses from hex 000E0000 to 000FFFFF are not cached when ROM is enabled, and are cached when ROM is disabled.
- The addresses from hex 00080000 to 000DFFFF are not cached when the memory split is at 512KB.

The range check for noncacheable addresses is enabled by bit 1 in the Cache/Timer Control register. When the boundary 0 range check is disabled, all addresses from 1MB to 16MB are cacheable and the addresses are not checked. When the range check is enabled, the memory controller checks the range of the memory address to determine if it is cacheable. If the address is equal to or greater than the lower boundary limit *and* equal to or less than the upper boundary limit, it is considered to be noncacheable.

Bit	Function
7–4	Upper Limit for Boundary 0
3–0	Lower Limit for Boundary 0

Figure 4-19. Boundary 0 Cache Control Register (Index Hex A3)

Bits 7–4 These bits contain the upper boundary limit and are compared to address bits A23 through A20 (address bits A24 thru A31 must be 0).

Bits 3–0 These bits contain the lower boundary limit and are compared to address bits A23–A20 (address bits A24 thru A31 must be 0).

Boundary 1 Cache Control Registers (Index Hex A4 and A5)

These two registers are 8-bit, read/write registers at index hex A4 and A5. The lower-boundary register (index hex A4) contains the lower-range control bits and the upper-boundary register (index hex A5) contains the upper-range control bits. The upper-range and lower-range control bits determine the noncacheable addresses in the memory space from 16MB to 256MB.

When the cache is enabled, memory below 1MB is cached except for the following:

- The addresses from hex 000A0000 to 000DFFFF are never cached.
- The addresses from hex 000E0000 to 000FFFFF are not cached when ROM is enabled, and are cached when ROM is disabled.
- The addresses from hex 00080000 to 000DFFFF are not cached when the memory split is at 512KB.

The range check for noncacheable addresses is enabled by bit 2 in the Cache/Timer Control register. When the boundary 1 range check is disabled, all addresses from 16MB to 256MB are cacheable and the addresses are not checked. When the range check is enabled, the memory controller checks the range of the memory address to determine if it is cacheable. If the address is equal to or greater than the lower boundary limit *and* equal to or less than the upper boundary limit, it is considered to be noncacheable.

The Lower Boundary 1 Cache Control register contains the lower boundary limit and is compared to address bits A27 through A20 (address bits A28 through A31 must be 0).

The Upper Boundary 1 Cache Control register contains the upper boundary and is compared to address bits A27 through A20 (address bits A28 through A31 must be 0). When the boundary includes the first 1MB of memory, any area that normally would be cacheable (for example hex 0E0000 to 0FFFFF) will be noncacheable.

Bit	Function
7	Lower Limit Boundary A27
6	Lower Limit Boundary A26
5	Lower Limit Boundary A25
4	Lower Limit Boundary A24
3	Lower Limit Boundary A23
2	Lower Limit Boundary A22
1	Lower Limit Boundary A21
0	Lower Limit Boundary A20

Figure 4-20. Lower Boundary 1 Cache Control Register (Index Hex A4)

Bit	Function
7	Upper Limit Boundary A27
6	Upper Limit Boundary A26
5	Upper Limit Boundary A25
4	Upper Limit Boundary A24
3	Upper Limit Boundary A23
2	Upper Limit Boundary A22
1	Upper Limit Boundary A21
0	Upper Limit Boundary A20

Figure 4-21. Upper Boundary 1 Cache Control Register (Index Hex A5)

Memory Bank Enable Register 1 (Index Hex A6)

This 8-bit, read/write register at index A6 controls the enabling of the first eight banks of memory in the system-board memory connectors A1 through A4.

Bit	Function
7	Disable Bank 7 (Port 97)
6	Disable Bank 6 (Port 96)
5	Disable Bank 5 (Port 95)
4	Disable Bank 4 (Port 94)
3	Disable Bank 3 (Port 93)
2	Disable Bank 2 (Port 92)
1	Disable Bank 1 (Port 91)
0	Disable Bank 0 (Port 90)

Figure 4-22. Memory Bank Enable Register 1 (Index Hex A6)

Bits 7–0 Each bit controls the enabling of memory for the specified bank. When the bit is set to 1, the bank is disabled. When the bit is 0, the bank is enabled and responds to the address specified in its Base Address register.

Note: Because memory must be enabled in pairs, corresponding banks should be enabled in Memory Bank Enable Register 2. These two registers should be set to the same value.

Memory Bank Enable Register 2 (Index Hex A7)

This 8-bit, read/write register at index hex A7 controls the enabling of the second 8 banks of memory in the system-board memory connectors B1 through B4.

Bit	Function
7	Disable Bank F (Port 9F)
6	Disable Bank E (Port 9E)
5	Disable Bank D (Port 9D)
4	Disable Bank C (Port 9C)
3	Disable Bank B (Port 9B)
2	Disable Bank A (Port 9A)
1	Disable Bank 9 (Port 99)
0	Disable Bank 8 (Port 98)

Figure 4-23. Memory Bank Enable Register 2 (Index Hex A7)

Bits 7–0 Each bit controls the enabling of memory for the specified bank. When the bit is set to 1, the bank is disabled. When the bit is 0, the bank is enabled and responds to the address specified in its Base Address register.

Note: Because memory must be enabled in pairs, corresponding banks should be enabled in Memory Bank Enable Register 1. These two register should be set to the same value.

System Bus Request Timer Register (Index Hex A8)

The System Bus Request Timer Register, the System Bus Hold Timer Register, and the System Bus Idle Timer Register work together to determine the percentage of time that the system microprocessor owns the channel. Each register is an 8-bit read/write register and they are active only when bit 7 of the Cache/Timer Control register is set to 1.

The values in these three registers are moved to counters, which are decremented for every other clock cycle of the system microprocessor.

For the Type 1 Processor Complex, each count represents 80 nanoseconds. For the Type 2 Processor Complex, each count represents 60 nanoseconds.

The System Bus Request Timer register at index hex A8 determines how long the system microprocessor will wait before preempting for use of the channel.

The value in this register is moved to a counter when the system microprocessor loses ownership of the channel. Decrementing the counter begins when the system microprocessor wants to use the channel and another master owns it.

If the count reaches 0 and the system microprocessor has not yet gained control of the channel, -PREEMPT is driven active.

System Bus Hold Timer Register (Index Hex A9)

This register determines the maximum amount of time that the system microprocessor maintains ownership of the channel.

The value in this register is moved to a counter when the system microprocessor loses ownership of the channel. Decrementing the counter begins as soon as the system microprocessor is granted ownership of the channel.

If -PREEMPT was driven active because of the System Bus Request Timer register the system microprocessor still owns the channel when the count reaches 0, the system microprocessor gives up ownership of the channel.

Note: If the system microprocessor gained control of the channel without driving -PREEMPT active, the channel is given up when another device requests the channel.)

System Bus Idle Timer Register (Index Hex AA)

This register determines the maximum amount of time that system microprocessor will retain control of the channel without using it.

The value in this register is moved to a counter each time the system microprocessor accesses the channel. Decrementing begins when the system microprocessor starts the channel access.

If the system microprocessor still owns the channel when the count reaches 0, the system microprocessor gives up ownership of the channel

Parity Trap Register (Hex 00E1)

This 8-bit, read-only register at address hex 00E1 is used to determine if system-board memory caused a parity error. When a parity error occurs, the state of 'row address select' signals is latched into this register.

If a bit is a 1, it indicates that the memory in that pair of matching connectors (A and B) was the cause of the error. To determine which memory card caused the error, test bits 0 and 1 in the Cache Status register (Hex 00E3).

If all bits in this register are 0, the error was not caused by the memory on the system board.

After each parity check, this register must be cleared using the parity-check bit (bit 0 of the Memory Controller register, index A1) to reset the latch.

Bit	Function
7	Error Indicator Connector A4 or B4
6	Error Indicator Connector A4 or B4
5	Error Indicator Connector A3 or B3
4	Error Indicator Connector A3 or B3
3	Error Indicator Connector A2 or B2
2	Error Indicator Connector A2 or B2
1	Error Indicator Connector A1 or B1
0	Error Indicator Connector A1 or B1

Figure 4-24. Parity Trap Register (Hex 00E1)

The bank that generated the parity error can by determined by bits 0 and 1 in the Cache Status Register (Hex 00E3).

Cache Control Register (Hex 00E2)

This 8-bit, read/write register at address hex 00E2

Bit	Function
7	-Flush Level 1 Cache
6	-Flush Level 2 Cache
5	-Enable Level 2 Cache
4–0	Reserved

Figure 4-25. Cache Control Register (Hex 00E2)

Bit 7 This bit controls the flushing of the level 1 cache. When the bit is toggled from 1 to 0, the cache is placed in the flushed state for one processor cycle. When this bit is set to 1, the cache is in the enabled state.

Note: To ensure that the cache is flushed, the cache must be in the enabled state before setting this bit to 0.

Bit 6 This bit controls the flushing of the level 2 cache. When the bit is set to 0, the cache is placed and held in the flushed state. When this bit is set to 1, the state of the level 2 cache is controlled by bit 5.

Bit 5 This bit and bit 6 work together to control the state of the level 2 cache. When this bit is 0, the level 2 cache is enabled.

To disable both caches at the same time, use the Cache/Timer Control register (index hex A2), then use the Cache Control register to flush both caches.

Bits 4–0 These bits are reserved.

Notes:

1. Refer to the 80486 microprocessor specification for information on CR0, which must also be set to enable the internal 8KB cache.
2. The cache must be disabled and flushed before enabling ROM or changing the cacheable boundaries.

Cache Status Register (Hex 00E3)

This 8-bit, read-only register contains diagnostic information related to the cache operation.

Bit	Function
7	Reserved
6	Processor Type
5, 4	-L2 Cache Installed
3	-L2 Cache Parity Error
2	Reserved
1	-Parity Error Bank A
0	-Parity Error Bank B

Figure 4-26. Cache Status Register (Hex 00E3)

Bit 7 This bit is reserved.

Bit 6 This bit indicates the type of processor complex that is present. When the bit is a 0, the Type 1 processor complex is installed; when the bit is 1, the Type 2 is installed.

Bits 5, 4 These bits identify the presence of the level 2 cache. When both bits are 0, the 256KB level-2 cache is present; when both bits are 1, the level 2 cache is not present. Other combinations of these bits are reserved.

Bit 3 When this bit is a 0, a parity error was detected in the level 2 cache.

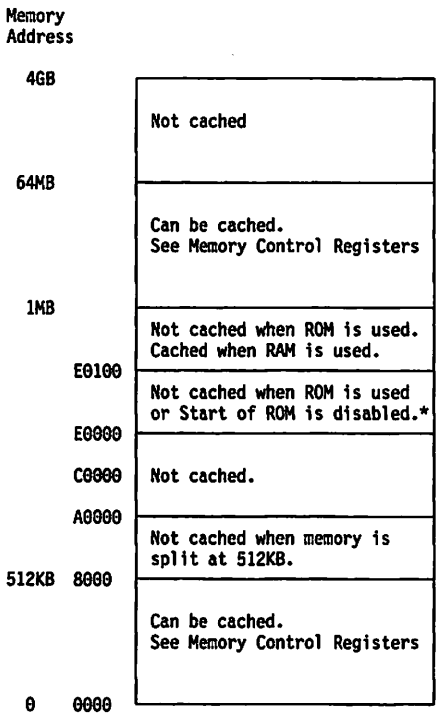
Bit 2 This bit is reserved.

Bit 1 When this bit is a 0, a parity error was detected in system-board memory in an A connector.

Bit 0 When this bit is a 0, a parity error was detected in system-board memory in a B connector.

Processor Cache Address Map

The following map is a summary of the Memory Control register options. Only system-board memory is mapped into the microprocessor and level-2 caches. The level 2 (L2) cache is optional.



* Refer to Bit 5 on page 4-16 for details.

Figure 4-27. Cache Address Map Summary

Read-Only Memory Subsystem

The ROM subsystem is located on the processor complex and consists of 128KB in a 128K by 8-bit arrangement. The ROM is active at power-on at address space hex 000E0000 to 000FFFFF. The ROM contains the code that initializes the system to load and execute POST and BIOS code from a DASD device. After POST ensures the proper operation of system memory, POST and BIOS code from a DASD device can be copied to RAM, and ROM disabled. ROM is also assigned at the top of the 4GB address space from hex FFFE0000 to FFFFFFFF.

Two bits in the Memory Control register (index A1) control ROM addressing. The ROM-enable bit determines whether ROM or RAM is accessed at address space hex 000E0000 to 000FFFFF, and the disable-ROM-space-decode bit selects the system decode and cacheability of the first 4KB of ROM address space.

Index

A

acoustic readings 1-7
activity light 3-19
adapter setup register 2-10
address maps
  POS 2-3
  system 1-4
address 20 gate 3-19
alternate reset 3-20
arbiter programming 4-9
arbitration control point 4-8
arbitration level 4-8
arbitration register 4-9
arbitration, central control point 4-9
assistance, developer 3-23

B

base memory bytes 3-16
battery 3-5
boot sequence 2-11
bus contention 2-2
bus priority assignments 4-8

C

cache connector 4-7
cache map 4-28
card selected feedback 2-4
-CD SFDBK 2-4
central arbitration control point 4-8
channel 1-3, 4-7
channel check 3-18
channel check latch 3-18
CMOS RAM 3-5
codes, POST error 3-23
configuration CRC bytes 3-16
configuration, RT/CMOS 3-12
connector numbering, memory 3-3
connector, optional cache 4-7

D

date century byte 3-16
description
  NMI 3-17
  password 3-21
  programmable option select 2-2
  real-time clock 3-5
  system 1-2
  system board 1-3
description, power supply 1-8
developer assistance 3-23
diagnostic status byte (RT/CMOS) 3-12
diskette drive connector 3-2
diskette drive connectors 3-2
diskette drive information, POS 2-8, 2-9
diskette drive type byte 3-13
diskette overruns 4-8
display panel, information 3-23
DMA channels 4-8

E

enable error checking 4-17
enable NMI 3-6
equipment byte 3-15
error codes, POST 3-23
extended arbitration 4-10

F

failures and messages, POST 3-23
fault, overvoltage 1-9
features
  processor complex 4-2
feedback, card selected 2-4
fixed disk activity light 3-19
fixed disk drive initialization 3-12
fixed disk type byte 3-14

© Copyright IBM Corp. 1990

G

gate A20 3-19

H

hardware compatibility 3-22

I

identifier, model 4-2
information panel 3-23
interrupts
  IRQ0 reset 3-18
  NMI reset 3-17
I/O address map 1-4

L

light, power 1-8
L2 cache 4-7

M

math coprocessor
maximum current 1-6
measurements 1-6
memory
  connectors 3-3
  performance 4-3
  refresh 4-8
  split block 4-17
memory information, POS 2-6
memory numbering 3-3
memory presence detect 4-11
memory size miscompare 3-12
messages, POST error 3-23
micro channel 1-3, 4-7
microprocessor
  alternate reset 3-20
  gate A20 3-19
  mode switch compatibility 3-20
  performance 4-3
mode switching 3-20
model/submodel bytes 4-2

N

NMI mask 3-6
NMI service 4-8
NMI signal 4-8
noise readings 1-7
nonmaskable interrupt (NMI) 3-17
numbering, memory connector 3-3

O

operating system boot
  sequence 2-11
optional cache card 4-7
output voltage sequencing 1-9
outputs, power supply 1-8
overvoltage fault 1-9

P

panel, information 3-23
parallel port select 2-5
parity check 3-18, 4-17
password, power-on 3-21
performance 3-2, 4-3
ports, system 3-17
POS address map 2-3
POS register 2 2-5
POS register 3 2-6
POS register, diskette drive 2-8,
  2-9
POST error codes 3-23
power specifications 1-6
power supply 1-3
power supply description 1-8
power supply LED 1-8
power supply output protection 1-9
power supply outputs 1-8
power-on password 3-21
processor cache map 4-28
processor complex
processor performance 4-3
programmable option select
  (POS) 2-2
    address decode 2-4
    address map 2-3
    registers 2-3

programmable option select (POS) *(continued)*
setup procedures 2-3
setup register 2-4
system board 2-4
system board I/O 2-5
system configuration utilities 2-2
warning 2-2
programming, arbiter 4-9
protected mode switch 3-20
protection, power supply 1-9

R

real mode switch 3-20
real-time clock bytes 3-8
refresh 4-8
refresh request 3-18
registers
adapter setup 2-10
card feedback 2-4
miscellaneous system 3-17
RT/CMOS and NMI mask 3-6
RT/CMOS status registers 3-9
system board POS 2-4
system control 3-17
restore channel check 3-18
ROM enable 4-17
RT/CMOS address register 3-6
RT/CMOS RAM 3-5
password 3-21
secured area 3-19

S

second fixed disk type byte 3-14
selectable boot 2-11
sequencing, output voltage 1-9
serial port address select 2-8
set configuration 2-11
setup
bus contention 2-2
registers 2-4
system board 2-4
system board I/O 2-5
shutdown status byte 3-12
size 1-6
sound specifications 1-7
speaker data enable 3-18
specifications 1-6, 4-3
performance 4-3
physical 1-6
system unit 1-6
split address bits 4-15
split memory block 4-17
status registers (RT/CMOS) 3-9
switch to real 3-20
system address map 1-4
system board 1-2, 1-3
features 1-3
I/O register 2-5
memory connector 3-3
model identifier 4-2
POS registers 2-4
setup 2-4
system configuration utilities 2-2
system control port A 3-19
system control ports 3-17
system description 1-2
system overview 1-1
system ports, miscellaneous 3-17
system reset 3-17
system unit specifications 1-6

T

time status indicator 3-12
type identifier 4-2

V

valid RAM bit 3-11

W

warnings
arbitration 4-8
central arbiter 4-10
NMI mask 3-17
POS 2-2
real-time clock 3-6
reserved areas v

watchdog status 3-19

Numerics

80486 performance 4-3