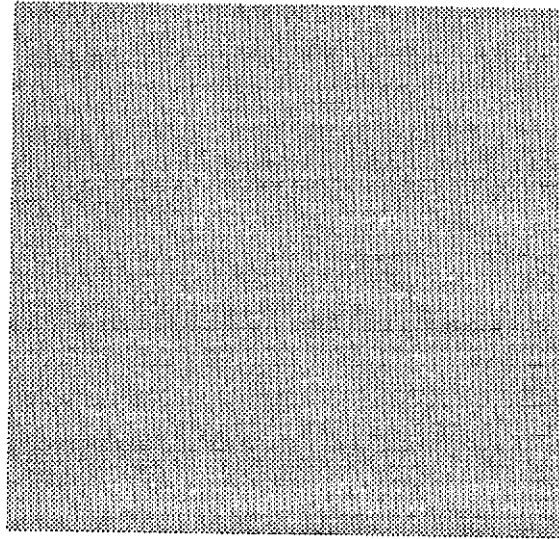


82C5087



IBM PC-AT Host
SCSI Controller

Datasheet

May, 1991

P R E L I M I N A R Y

Copyright Notice

Copyright © 1991, Chips and Technologies, Inc.

Trademark Acknowledgement

Chips and Technologies™, CHIPS™, CHIPS/250™, CHIPS/280™, CHIPS/450™, CHIPSpak™, CHIPSport™, CHIPSet™, CHIPSlink™, LeAPset™, LeAPsetsx™, SMARTMAP™, are trademarks of CHIPS and Technologies, Inc.

IBM AT™, IBM XT™, PS/2™, and Personal System/2™ are trademarks and IBM® is a registered trademark of International Business Machines Corporation.

Disclaimer

This data sheet is provided for the general information of the customer. Chips and Technologies, Inc., reserves the right to modify these parameters as necessary and the customer should ensure that it has the most recent revision of the data sheet. CHIPS makes no warranty for the use of its products and bears no responsibility for any errors which may appear in this document. The customer should be on notice that the field of personal computers is the subject of many patents held by different parties. Customers should ensure that they take appropriate action so that their use of the products does not infringe upon any patents. It is the policy of CHIPS and Technologies, Inc. to respect the valid patent rights of third parties and not to infringe upon or assist others to infringe upon such rights.



Preface

Scope

This manual contains the information needed to program the CHIPS F82C5087 SCSI-Bus Interface Controller to implement the SCSI (Small Computer System Interface) interface on a host computer or a device controller. A working knowledge of the SCSI interface is assumed.

Contents

The information in this manual is divided into five chapters, one appendix, and a glossary.

- Chapter 1 provides an overview of the CHIPS F82C5087 IBM AT HOST SCSI Controller.
- Chapter 2 describes the F82C5087 hardware specifications. It supplies physical and functional pin specifications, signal descriptions, electrical specifications, and packaging specifications.
- Chapter 3 describes the operational modes of the F82C5087.
- Chapter 4 provides a detailed description of the function and operation of each of the 24 internal registers of the F82C5087.
- Chapter 5 provides a detailed description of the operation of each of the 46 commands of the F82C5087.
- Appendix A contains a sample flow chart that illustrates the steps involved in completing an asynchronous data transfer to a target device.
- The glossary provides a list of abbreviations and definitions of the key terms used throughout this manual

Related Publications

- *CHIPS 5086B SCSI Bus Interface controller Reference Manual*
- *CHIPS 5055B DMA Memory Controller/Programmable Data Sequencer Reference Manual*
- *CHIPS 5080 SCSI Multifunctional Device Reference Manual.*
- *American National Standard for Information Systems - Small Computer System Interface (SCSI). (X3.131-1986.)*

Notational Conventions

The following conventions are used throughout this manual:

- | | |
|-----------|--|
| UPPERCASE | is used to indicate names of commands, signals, SCSI bus phases. |
| - | a minus sign prefix to a signal name indicates an active low polarity. |
| + | a plus sign prefix to a signal name indicates an active high polarity. |

Contents

Chapter 1: Overview	1-1
Introduction	1-1
Hardware Features	1-2
Firmware Features	1-3
Typical Applications	1-4
Host Adapter	1-4
Motherboard	1-4
Operational Modes	1-6
Initiator or Target Mode	1-6
Non-Differential Mode	1-7
Asynchronous or Synchronous Mode	1-7
Master or Slave Mode	1-7
System Interface	1-8
I/O Interface	1-8
Command Set	1-9
Memory Structure	1-10
Performance Specifications	1-10
Chapter 2: Hardware Specifications	2-1
Physical Specifications	2-1
Pin Descriptions	2-2
Package Specifications	2-4
Functional Chip Specifications	2-5
Functionality of the F82C5087 Host Adapter	2-5
Interfacing with the SCSI Control Bus	2-7
Interfacing with the SCSI Data Bus	2-7
Interfacing with the Host Computer's MPU	2-7

Interfacing with the Host Computer's DMA Controller	2-8
Signal Descriptions	2-9
I/O Signals	2-9
Signals Specific to 8085/8051 Mode	2-16
Signals Specific to Z8 Mode	2-17
Signals Specific to Wide Data Bus Transfers	2-18
Pad Drive Capacity	2-19
Electrical Specifications	2-20
Clock Timing	2-20
Z8 Mode Timing	2-21
8085/8051 Mode Timing	2-23
Non-multiplexed Mode Timing	2-25
Timing for Data Transfers to System Memory	2-27
Timing for I/O Interface to Buffer Memory	2-32
SCSI Interface Timing	2-34
Timing for the SCSI Bus ARBITRATION Phase	2-34
Timing for SCSI Bus SELECT/RESELECT Phases	2-35
Timing for Asynchronous/Synchronous Transfers	2-36
D.C. Specifications	2-44
Absolute Maximum Ratings	2-44
Standard Test Conditions	2-44
D.C. Characteristics	2-44
D. C. Characteristics for Drivers/Receivers	2-45
Chapter 3: Operation	3-1
Modes of Operation	3-1
Initiator and Target Modes	3-1
Role of Initiator	3-1
Role of the Target	3-2
Transfer Modes	3-2
Asynchronous Transfers	3-2
Synchronous Transfers	3-2
Master/Slave Mode	3-5
Master/Master Mode	3-6
Slave/Master Mode	3-6
Slave/Slave Mode	3-6

Synchronizing Memory Data Bus Transfers	3-6
Synchronizing SCSI Bus Data Transfers	3-7
Hardware Specifications	3-7
Method of Operation	3-7
I/O Interface	3-8
Example 1. I/O Interface Providing a Path to the SCSI Bus.	3-8
Example 2. I/O Interface to Buffer Memory.	3-8
Example 3. I/O Interface to the MPU.	3-8
Example 4. I/O Interface to the Host Computer.	3-8
Command Set	3-9
Memory Structure	3-9
Data FIFO	3-10
MCS Script	3-10
MCS Address Pointers	3-11
MCS Auto Advance	3-11
Pipelining Structure	3-11
Example 1. SCSI Transfer Command	3-13
Example 2. SCSI Select/Reselect Process.	3-14
Target ID	3-15
Identify Message	3-15
Tag Message	3-15
Multi-byte Commands	3-15
Chapter 4: Registers	4-1
Function of Registers	4-1
Register Description	4-2
Registers 0-3 (0-3h)	4-3
Register 0-3 (0-3h) - Write Operation	4-3
Registers 0-3 (0-3h) - Read Operation	4-5
Register 4 (4h)	4-7
Register 4 (4h) - Write Operation	4-7
Register 4 - Read Operation	4-8
Register 5 (5h)	4-9
Register 5 (5h) - Write Operation	4-9
Register 5 (5h) - Read Operation	4-9

Register 6 (6h)	4-12
Register 6 (6h)- Write Operation	4-12
Register 6 (6h) - Read Operation	4-12
Register 7 (7h)	4-15
Register 7 (7h) - Write Operation	4-15
Register 7 (7h) - Read Operation	4-18
Register 8 (8h)	4-19
Register 8 (8h) - Write Operation (Lower 4 bits only)	4-19
Register 8 (8h) - Read Operation	4-19
Register 9 (9h)	4-20
Register 9 (9h) - Write Operation	4-20
Register 9 (9h) - Read Operation	4-20
Register 10 (0Ah)	4-21
Register 10 (0Ah) - Write/Read Operation	4-21
Register 11 (0Bh)	4-22
Register 11 (0Bh) - Write Operation	4-22
Register 11 (0Bh) - Read Operation	4-22
Register 12 (0Ch)	4-23
Register 12 (0Ch) - Write/Read Operation	4-23
Register 13 (0Dh)	4-24
Register 13 (0Dh)- Write Operation	4-24
Register 13 (0Dh)- Read Operation	4-24
Register 14 (0Eh)	4-25
Register 14 (0Eh) - Write Operation	4-25
Register 14 (0Eh) - Read Operation	4-25
Register 15 (0Fh)	4-27
Register 15 (0Fh) - Write Operation	4-27
Register 15 (0Fh) - Read Operation	4-27
Registers 16 - 18 (10 - 12h)	4-28
Register 16 - 18 (10 - 12h)- Write Operation	4-28
Registers 16 - 18 (10 - 12h) - Read Operation	4-28
Register 19 (13h)	4-29
Register 19 (13h) - Write Operation	4-29
Register 19 (13h) - Read Operation	4-30
Register 20 (14h)	4-33

Register 20 (14h) - Write Operation	4-33
Register 20 (14h) - Read Operation	4-34
Register 21 (15h)	4-36
Register 21 (15h)- Write Operation	4-36
Register 21 (15h) - Read Operation	4-36
Register 22 (16h)	4-37
Register 22 (16h) - Write Operation	4-37
Register 22 (16h) - Read Operation	4-38
Register 23 (17h)	4-39
Register 24 (18h)	4-40
Register 25 (19h)	4-41
Register 25 (19h) - Write Function	4-41
Chapter 5: Commands	5-1
Command Functions	5-1
Command Description	5-2
Command Code	5-2
Bus Condition	5-3
Execution Time	5-3
Operational Description	5-3
Possible Errors	5-3
Control Command Descriptions	5-4
CNTL NOP - No Operation	5-5
Operational Description	5-5
CNTL RST FIFO - Control Reset Data FIFO	5-6
Operational Description	5-6
CNTL RST ON - Control RESET On	5-7
Operational Description	5-7
CNTL RST OFF - Control RESET	5-7
Operational Description	5-7
CNTL ATN ON - Control Attention On	5-8
Operational Description	5-8
CNTL ATN OFF - Control Attention Off	5-8
Operational Description	5-8
CNTL ENB SEL - Control Enable Select	5-9

Operational Description 5-9

CNTL DIS SEL - Control Disable Select 5-10

Operational Description 5-10

CNTL ENB RESEL - Control Enable Reselect 5-11

Operational Description 5-11

CNTL DIS RESEL - Control Disable Reselect 5-12

Operational Description 5-12

CNTL ENB ADV - Control Enable MCS Auto Advance 5-13

Operational Description 5-13

CNTL DIS ADV - Control Disable MCS Auto Advance 5-14

Operational Description 5-14

CNTL DATA PTM - Control Data
from MPU to Memory Data Bus 5-15

Operational Description 5-15

CNTL DATA MTP - Control Data
from Memory Data Bus to MPU 5-16

Operational Description 5-16

Initiator Command Descriptions 5-17

INIT SWOA SEQ - Initiator Select without Attention Sequence 5-18

Operational Description 5-18

Possible Errors 5-18

INIT SWA SEQ - Initiator Select with Attention Sequence 5-19

Operational Description 5-19

Possible Error 5-19

INIT SWOA CMD - Initiator Select without Attention Command 5-21

Operational Description 5-21

Possible Errors 5-21

INIT SWA CMD - Initiator Select with Attention Command 5-22

Operational Description 5-22

Possible Errors 5-22

INIT REC INFO - Initiator Receive Information 5-23

Operational Description 5-23

Possible Errors 5-23

INIT DATA MTS - Initiator Receive Data
from Memory Data Bus to SCSI 5-24

Operational Description 5-24

Possible Errors	5-24
INIT DATA PTS - Initiator Receive Data from MPU to SCSI Bus	5-25
Operational Description	5-25
Possible Errors	5-25
INIT DATA STM - Initiator Receive Data from SCSI to Memory Data Bus	5-26
Operational Description	5-26
Possible Errors	5-26
INIT DATA STP - Initiator Receive Data from SCSI to MPU Bus	5-27
Operational Description	5-27
Possible Errors	5-27
INIT SEND MSG - Initiator Send Message	5-28
Operational Description	5-28
Possible Errors	5-28
INIT SEND CMD - Initiator Send Command	5-29
Operational Description	5-29
Possible Errors	5-29
INIT XFER PAD - Initiator Transfer Pad	5-31
Operational Description	5-31
Possible Error	5-31
INIT XFER BYTE - Initiator Transfer Byte	5-32
Operational Description	5-32
Possible Error	5-32
INIT CMDC SEQ - Initiator COMMAND COMPLETE/DISCONNECT Sequence	5-33
Operational Description	5-33
Possible Errors	5-33
INIT WFR CMD - Initiator Wait for Reselect	5-34
Operational Description	5-34
Possible Error	5-34
INIT WFR SEQ - Initiator Wait for Reselect Sequence	5-35
Operational Description	5-35
Possible Errors	5-35

Target Command Descriptions	5-37
TGT SEND MSG - Target Send Message	5-38
Operational Description	5-38
Possible Error	5-38
TGT SEND STAT - Target Send Status	5-39
Operational Description	5-39
Possible Errors	5-39
TGT DATA MTS - Target Receive Data from Memory Data Bus to SCSI Bus	5-40
Operational Description	5-40
Possible Errors	5-40
TGT DATA PTS - Target Receive Data from MPU to SCSI Bus	5-41
Operational Description	5-41
Possible Errors	5-41
TGT REC CMD - Target Receive Command	5-42
Operational Description	5-42
Possible Errors	5-42
TGT REC MSG - Target Receive Message	5-43
Operational Description	5-43
Possible Errors	5-43
TGT DATA STM - Target Receive Data from SCSI Bus to Memory Data Bus	5-44
Operational Description	5-44
Possible Errors	5-44
TGT DATA STP - Target Receive Data from SCSI to MPU Bus	5-45
Operational Description	5-45
Possible Errors	5-45
TGT CMDC SEQ - Target COMMAND COMPLETE Sequence	5-46
Operational Description	5-46
Possible Errors	5-46
TGT XFER BYTE - Target Transfer Byte	5-47
Operational Description	5-47
Possible Error	5-47
TGT DISC CMD - Target Disconnect Command	5-48
Operational Description	5-48
TGT DISC SEQ - Target DISCONNECT Sequence	5-49

Operational Description	5-49
Possible Error	5-49
TGT RCNT SEQ - Target Reconnect Sequence	5-50
Operational Description	5-50
Possible Errors	5-50
TGT SEND BSY - Target Send Busy Command	5-51
Operational Description	5-51
Possible Errors	5-51
TGT WFS CMD - Target Wait for Select Command	5-52
Operational Description	5-52
Possible Error	5-52
TGT WFS SEQ - Target Wait for Select Sequence	5-53
Operational Description	5-53
Possible Errors	5-53

Appendix A

A-1

Glossary

Figures

Chapter 1: Overview	1-1
Figure 1-1.	
Typical F82C5087 Application	1-1
Figure 1-2.	
Block Diagram of the F82C5087	1-2
Figure 1-3.	
An AT Host Adapter System Configuration	1-5
Figure 1-4.	
SCSI on Motherboard System Configuration	1-6
Figure 1-5.	
Typical Wide SCSI Bus Configuration	1-8
Chapter 2: Hardware Specifications	2-1
Figure 2-1.	
Physical Pin-out of the 80-pin PFP Package	2-2
Figure 2-2.	
80-pin PFP Package Specification	2-4
Figure 2-3.	
Functional Pin out of the F82C5087 in Host Adapter Mode	2-6
Figure 2-4.	
Clock Timing Characteristics	2-20
Figure 2-5.	
Write Operation Z8 Mode Timing Characteristics	2-21
Figure 2-6.	
Read Operation Z8 Mode Timing Characteristics	2-22
Figure 2-7.	
Write Operation 8085/8051 Mode Timing Characteristics	2-23

Figure 2-8.
Read Operation 8085/8051 Mode Timing Characteristics 2-24

Figure 2-9.
Write Operation, Non-multiplexed MPU Timing Characteristics 2-25

Figure 2-10.
Read Operation, Non-multiplexed MPU Timing Characteristics . 2-26

Figure 2-11.
MPU Write IORDY Timing Characteristics 2-28

Figure 2-12.
MPU Read IORDY Timing Characteristics 2-29

Figure 2-13.
FIFO_RDY with Address Timing Characteristics 2-30

Figure 2-14.
FIFO_RDY without Address Timing Characteristics 2-31

Figure 2-15.
DMA Interface, Write to F82C5087 Timing Characteristics . . . 2-32

Figure 2-16.
DMA Interface, Read from F82C5087 Timing Characteristics . . 2-33

Figure 2-17.
SCSI Arbitration Timing Characteristics 2-34

Figure 2-18.
SCSI Select/Reselect Timing Characteristics 2-35

Figure 2-19.
SCSI Target Asynchronous Transfer
Output Timing Characteristics 2-36

Figure 2-20.
SCSI Target Asynchronous Transfer
Input Timing Characteristics 2-37

Figure 2-21.
SCSI Initiator Asynchronous Transfer
Output Timing Characteristics 2-38

Figure 2-22.
SCSI Initiator Asynchronous Transfer
Input Timing Characteristics 2-39

Figure 2-23.
SCSI Target Synchronous Transfer
Output Timing Characteristics 2-40

Figure 2-24.

SCSI Target Synchronous Transfer
 Input Timing Characteristics 2-41

Figure 2-25.

SCSI Initiator Synchronous Transfer
 Output Timing Characteristics 2-42

Figure 2-26.

SCSI Initiator Synchronous Transfer
 Input Timing Characteristics 2-43

Chapter 3: Operation 3-1

Figure 3-1.

Functional Pin out of Cascaded 5087s 3-4

Figure 3-2.

Conceptual Diagram of the F82C5087's Memory 3-9

Figure 3-3.

F82C5087's Command Queueing 3-12

Appendix A A-1

Figure A-1.

Asynchronous Data Transfer Flow Chart A-2

Tables

Chapter 1: Overview 1-1

Table 1-1.

F82C5087 Command Set	1-9
--------------------------------	-----

Table 1-1.

F82C5087 Command Set (continued)	1-10
--	------

Chapter 2: Hardware Specifications 2-1

Table 2-1.

A Pin List of the 80-pin Package	2-3
--	-----

Table 2-2.

Input/Output Signals	2-9
--------------------------------	-----

Table 2-3.

Signals Specific to 8085/8051 Mode	2-16
--	------

Table 2-4.

Signals Specific to Z8 Mode	2-17
---------------------------------------	------

Table 2-5.

Signals Specific to Wide Data Bus Transfers	2-18
---	------

Table 2-6.

Pad Drive Capacity	2-19
------------------------------	------

Chapter 3: Operation 3-1

Table 3-1.

Mode Control Register Setting	3-5
---	-----

Table 3-2.

Master/Slave Mode	3-5
-----------------------------	-----

Chapter 4: Registers **4-1**

Table 4-1.

Addressing the F82C5087 Registers 4-1

Table 4-1.

Addressing the F82C5087 Registers (continued) 4-2

Chapter 5: Commands **5-1**

Table 5-1.

F82C5087 Control Commands 5-4

Table 5-2.

Initiator Commands 5-17

Table 5-3.

Command Length Codes 5-29

Table 5-4.

F82C5087 Target Commands 5-37

Chapter 1

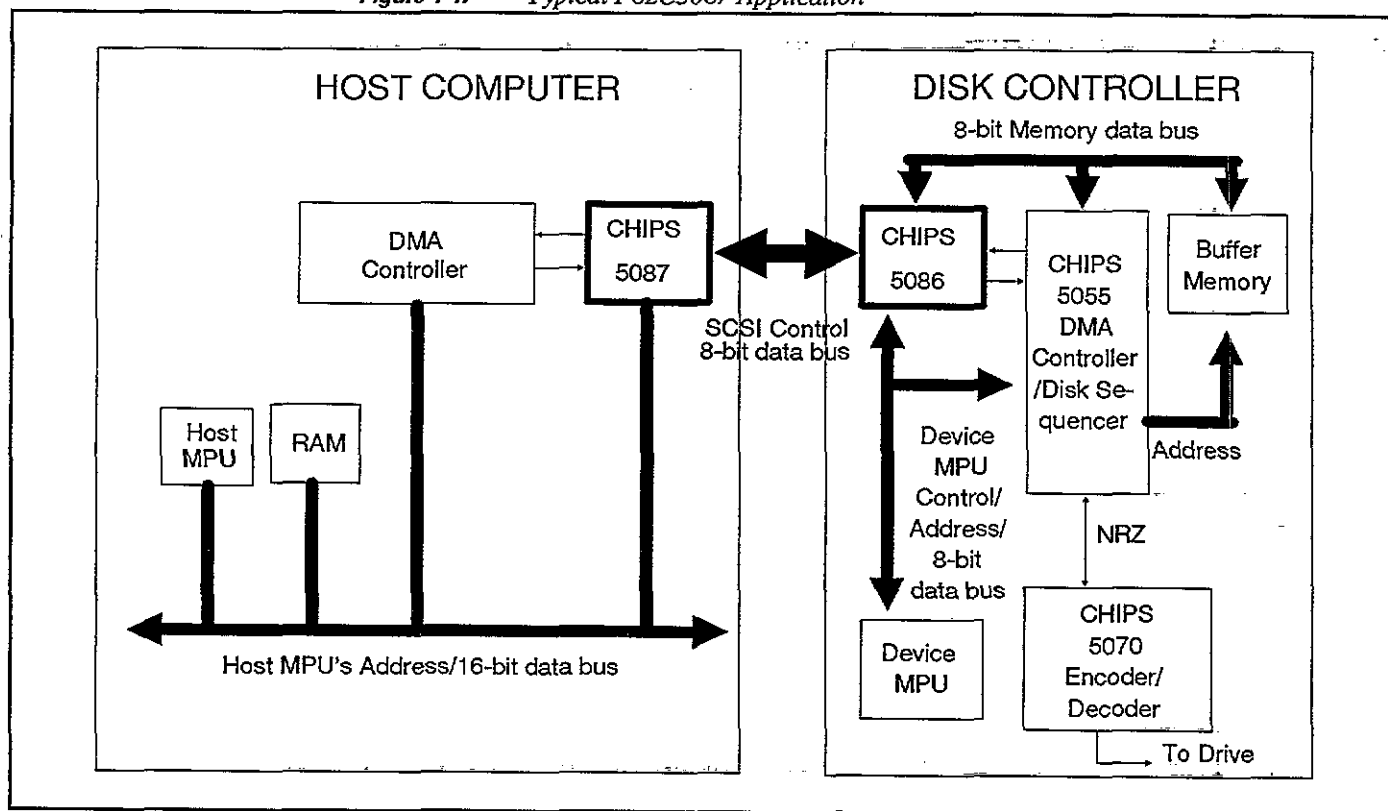
Overview

Introduction

The CHIPS F82C5087 SCSI controller provides all the functions needed to implement the SCSI interface on a host computer or a device controller. The chip provides an I/O interface to the SCSI bus and to the other system components (e.g., buffer memory, the on-board microprocessor (MPU), and the host computer).

Used as a host adapter, the F82C5087 functions as a bridge between the host computer bus and the SCSI bus. In this application, the F82C5087 connects an 8- or 16-bit host computer data bus to an 8-bit SCSI data bus. The F82C5087 interfaces such peripheral equipment as a Winchester drive to the SCSI bus. Refer to Figure 1-1 for an illustration of these two applications. For more details on F82C5087 applications refer to the section "Typical Applications."

Figure 1-1. Typical F82C5087 Application



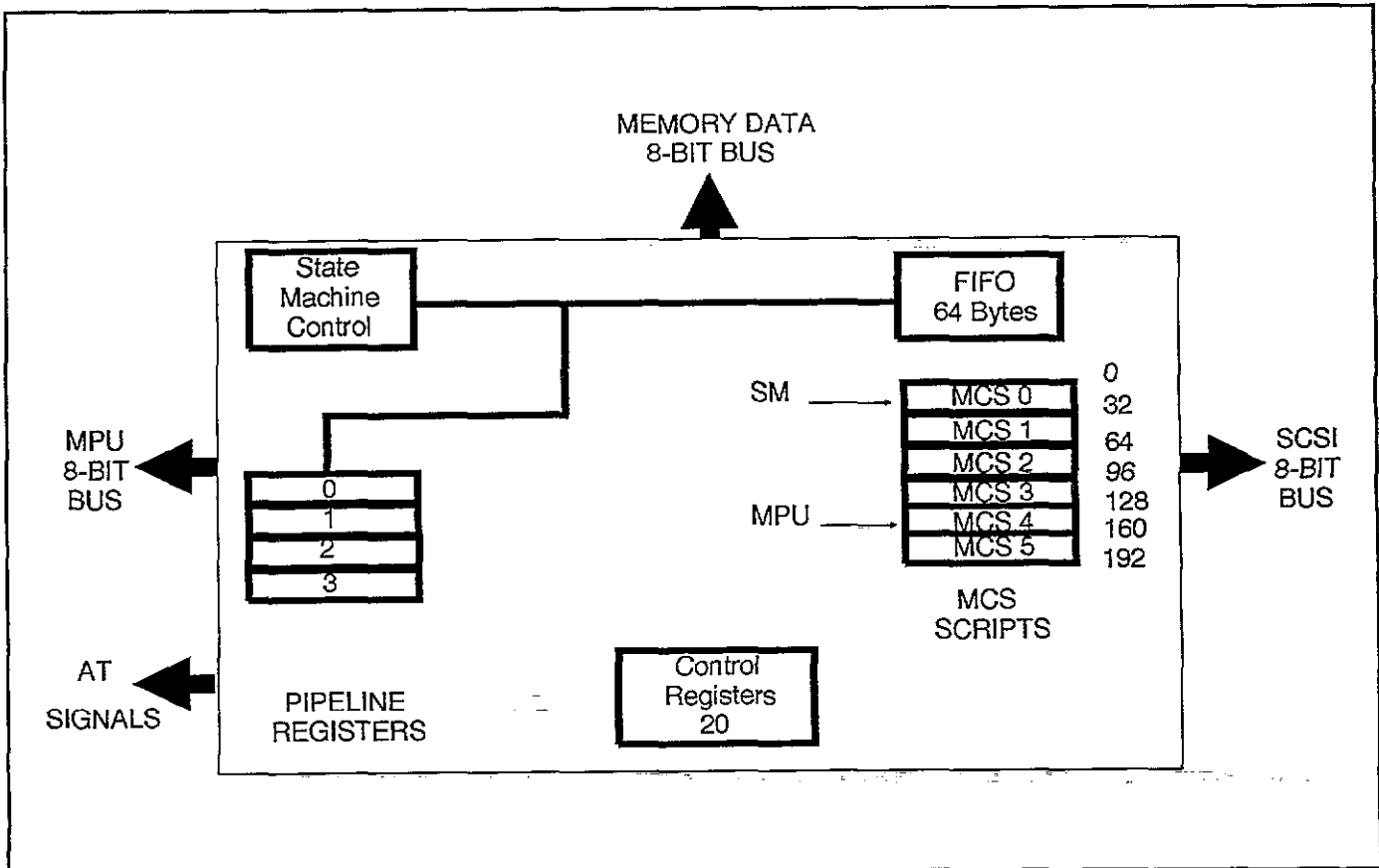
The F82C5087's hardware and firmware make it a flexible, as well as a powerful chip. It is available in an 80-pin flat pack, and can be purchased with or without firmware, standalone, or as part of a host adapter board provided as a Turnkey Manufacturing Package (MK5087 SCSI Host Adapter for PC/AT).

Hardware Features

The F82C5087's memory structure consists of 64 bytes of FIFO and six individual 32-byte Message/Command Space (MCS) scripts. The large FIFO buffer is used to store all the data transferred between the MPU's bus, the memory data bus or the SCSI bus. This increases system throughput by smoothing out the system's data transfers.

The MCS is a valuable tool for tailoring the F82C5087 to the needs of particular applications. It can be used to store blocks of frequently used commands, messages, and status. For instance, if the F82C5087 is used as a host adapter operating as an initiator in a system configuration with an 80286 host MPU, the firmware might decide to use the MCS to store messages frequently sent by the initiator. This improves system performance because the information can be accessed as a block, and does not have to be retrieved from system memory. Because the F82C5087 provides two MCS address pointers the MPU can be processing one MCS segment, while the F82C5087 state machine (SM) is simultaneously processing another 32-byte MCS segment. Figure 1-2 shows a simplified block diagram of the chip's functions.

Figure 1-2. Block Diagram of the F82C5087



The F82C5087's 24 registers provide a means to control the F82C5087's internal operation, and to gain status on the F82C5087's operation. The F82C5087's first four registers, the command pipeline registers, in conjunction with the MCS, enable up to four F82C5087 commands to be queued.

The F82C5087 hardware has an internal state machine that manages the system flow of SCSI information. Because the F82C5087 hardware is managing this lower-level task, the firmware is free to manage such higher-level system activities as task scheduling and resource allocation.

The F82C5087's bus structure increases system performance and provides system design flexibility. The F82C5087 supports an 8-bit MPU data bus, an 8-bit SCSI bus, and an 8-bit high-speed memory data bus. This separate high-speed data bus increases system throughput by enabling data and control activities to be independent. The F82C5087's bus structure also adds to the chip's versatility. For instance, the bandwidth of the SCSI bus and the memory data bus can be extended from 8 to 16 or 32 bits by cascading F82C5087s. The F82C5087 also supports the option of extending the MPU data bus to 16 bits by using the memory data bus as the MPU's upper data bus. Figure 1-3 shows a F82C5087 host adapter supporting 16-bit MPU data transfers. For more information on cascading F82C5087s refer to the section "Master or Slave Mode."

The F82C5087 supports two user-controlled power management features making it ideal for notebook, palmtop and laptop applications. The sleep mode registers allows the clock input to the chip to be turned off. Termination power resistors to the SCSI connector can be switched on or off through a pin on the F82C5087.

Other F82C5087 hardware features include a power-on flag, and input pins that support jumper-selectable options that reduce on-board circuitry.

Because the F82C5087 is built using proven CMOS low-power technology it requires only a single +5 Volt supply.

Firmware Features

Such firmware features as an extensive command set provide firmware engineers with a high degree of control over the F82C5087's operation. The F82C5087's large set of 46 commands enables firmware engineers to exercise a great deal of control over the SCSI bus activities, and to manage the system flow of SCSI information. These commands perform such simple tasks as transferring a byte from the SCSI bus into the MCS, or such series of tasks as receiving a command complete sequence from the SCSI bus, and transferring it into the MCS. Certain control commands can be used to establish a data path between the MPU's data bus, the memory data bus, and the SCSI bus.

One of the F82C5087's main firmware features is command queuing which reduces command overhead. The F82C5087 enables up to four F82C5087 commands to be queued in the pipeline registers. These commands can enable the F82C5087 to sequence through several SCSI bus phases without firmware intervention. For example, the F82C5087 command can handle a complete SCSI command sequence from disconnect, to connect, and to bus free. For more details refer to the section "Command Queuing".

The F82C5087's set of 24 registers provide maximum programming flexibility because registers record the status of all commands executed, and keep records of previous executions. This facilitates the monitoring of command flow, and error recovery.

Another important feature is the F82C5087's support of programmable conditions for interrupt generation. This enables the firmware to dictate the conditions under which an interrupt should be generated to the on-board MPU. This is useful because the conditions for interrupt generation will vary depending on the amount of firmware control desired, and whether the F82C5087 is used as an initiator or a target. Thus, the firmware can tailor the conditions for interrupt generation to fit their particular applications.

Typical Applications

The F82C5087 can be designed into a system as a host adapter or can provide the SCSI host function on a PC-AT motherboard. It provides system design flexibility. For instance, as well as supporting the option of cascading, the F82C5087 provides two different ways of wiring the F82C5087 to support data transfers to/from system memory. The F82C5087's FIFO_RDY pin can be connected to a DMA controller on the system's memory data bus, or it can be connected to the F82C5087's IORDY pin to the MPU's wait state generator to support burst transfers (string operation) from the MPU to the F82C5087.

Host Adapter

Used as a host adapter, the F82C5087 functions as a bridge between the PC-AT host computer bus and the SCSI bus. The 8-bit bidirectional MPU address/data bus and its associated control signals connect the F82C5087 to the host computer's MPU. The F82C5087 uses the SCSI control bus, comprised of nine F82C5087 control signals, as a path to the SCSI bus. For 16-bit host computer MPU's, the 8-bit memory data bus is used to transfer the high byte. Figure 1-3 shows a configuration in which the F82C5087 is used as a host adapter.

Typically, the host computer requires such information as status or data from the devices connected to the SCSI bus. The host adapter queues up these requests, and sends them to the appropriate device(s) via the SCSI bus. The host adapter enables SCSI peripheral devices to communicate with such host systems as an IBM PC AT that do not directly support the SCSI interface. In host adapter mode, the MPU data bus can be extended up to 16 bits; the memory data bus functions as the MPU's upper data bus. In this mode, the F82C5087 performs byte to word packing and unpacking. It provides 24 mA drive capability for direct interface to a PC-AT bus.

Motherboard

The F82C5087 can be used to implement a SCSI port on a PC-AT motherboard. It requires only the termination resistors and connector, besides the chip, to provide an optimized SCSI host solution. The F82C5087 has two useful power management, sleep mode and termination power disable, that make it ideal for laptop and notebook environments. Figure 1-4 shows such a configuration.

Figure 1-3. An AT Host Adapter System Configuration

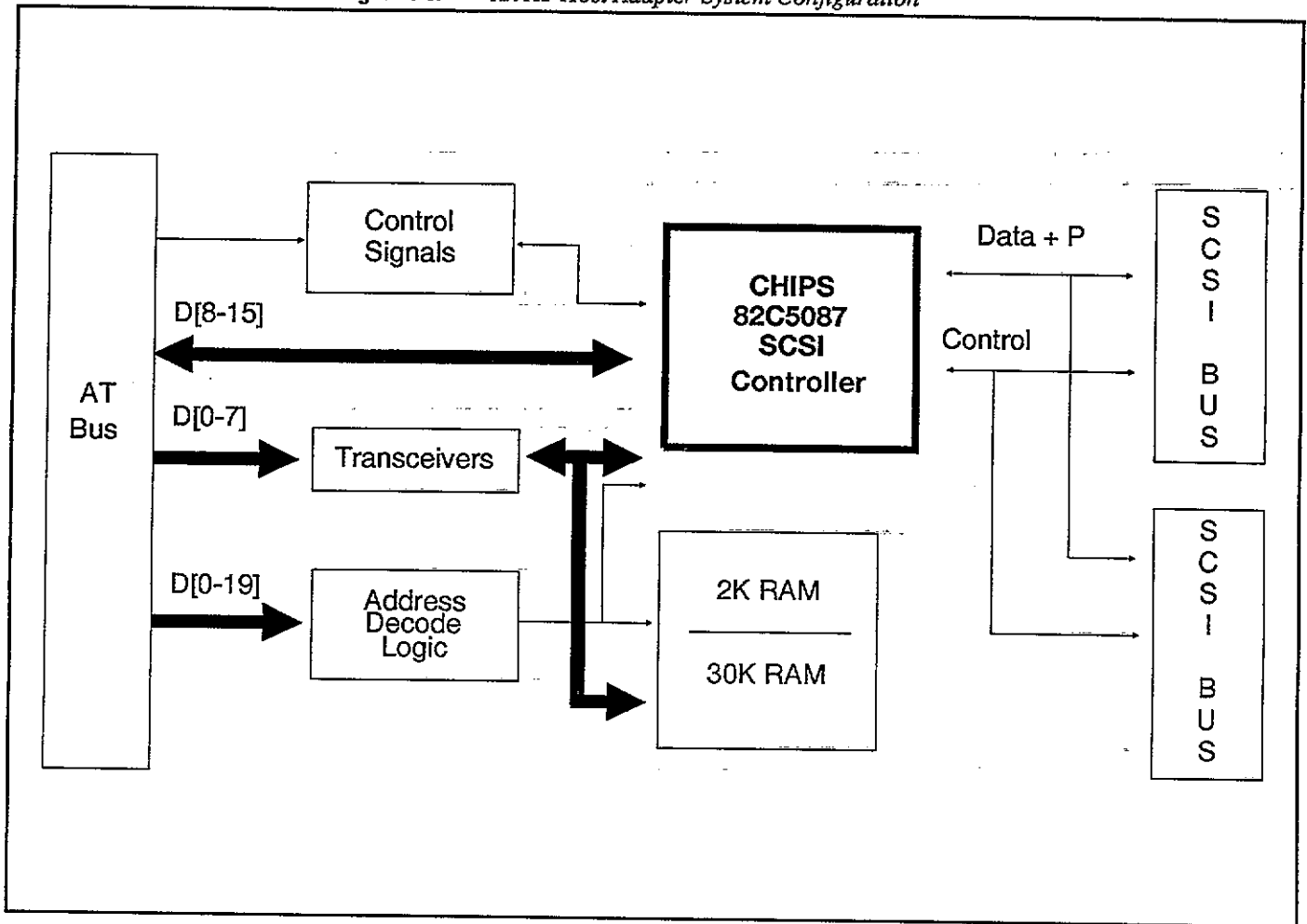
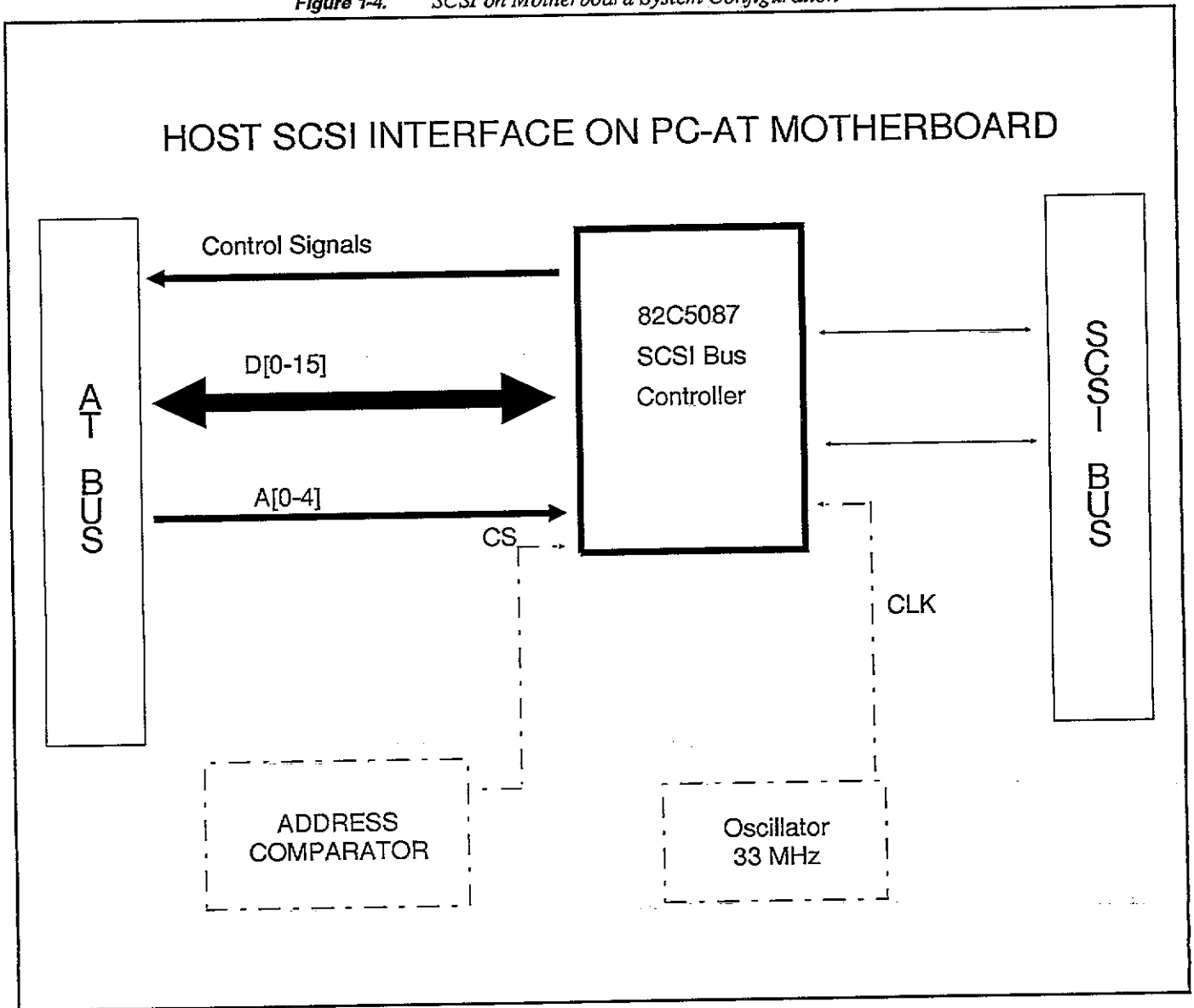


Figure 1-4. SCSI on Motherboard System Configuration



Operational Modes

The F82C5087 offers application flexibility because of the wide range of operational modes it supports. The F82C5087 supports the SCSI bus initiator or target mode, asynchronous or synchronous transfer mode, and master or slave mode (wide SCSI).

Initiator or Target Mode

Systems can be configured with single or multiple initiators and targets. Typically, the initiator is the host adapter responsible for originating the operation. The F82C5087 initiator arbitrates for the SCSI bus, selects the target device, and sends it a stored multi-byte SCSI command from the MCS.

A F82C5087 operating in target mode saves the initiator's device ID, receives the multi-byte command, sends a DISCONNECT message or terminates the sequence, then switches to the BUS FREE phase.

Non-Differential Mode

SCSI bus signals transmitted between initiators and targets can be single-ended or differential. The SCSI interface controller must operate in non-differential mode if single-ended drivers/receivers are used. The F82C5087 supports on-board 48 mA drivers for the single-ended SCSI bus. A six meter cable is the maximum supported in non-differential mode. The F82C5087 does not support a differential SCSI bus.

Asynchronous or Synchronous Mode

The F82C5087 supports both asynchronous and synchronous data transfers over the SCSI bus. For high-performance systems, the F82C5087 device can operate in synchronous mode to support higher transfer rates.

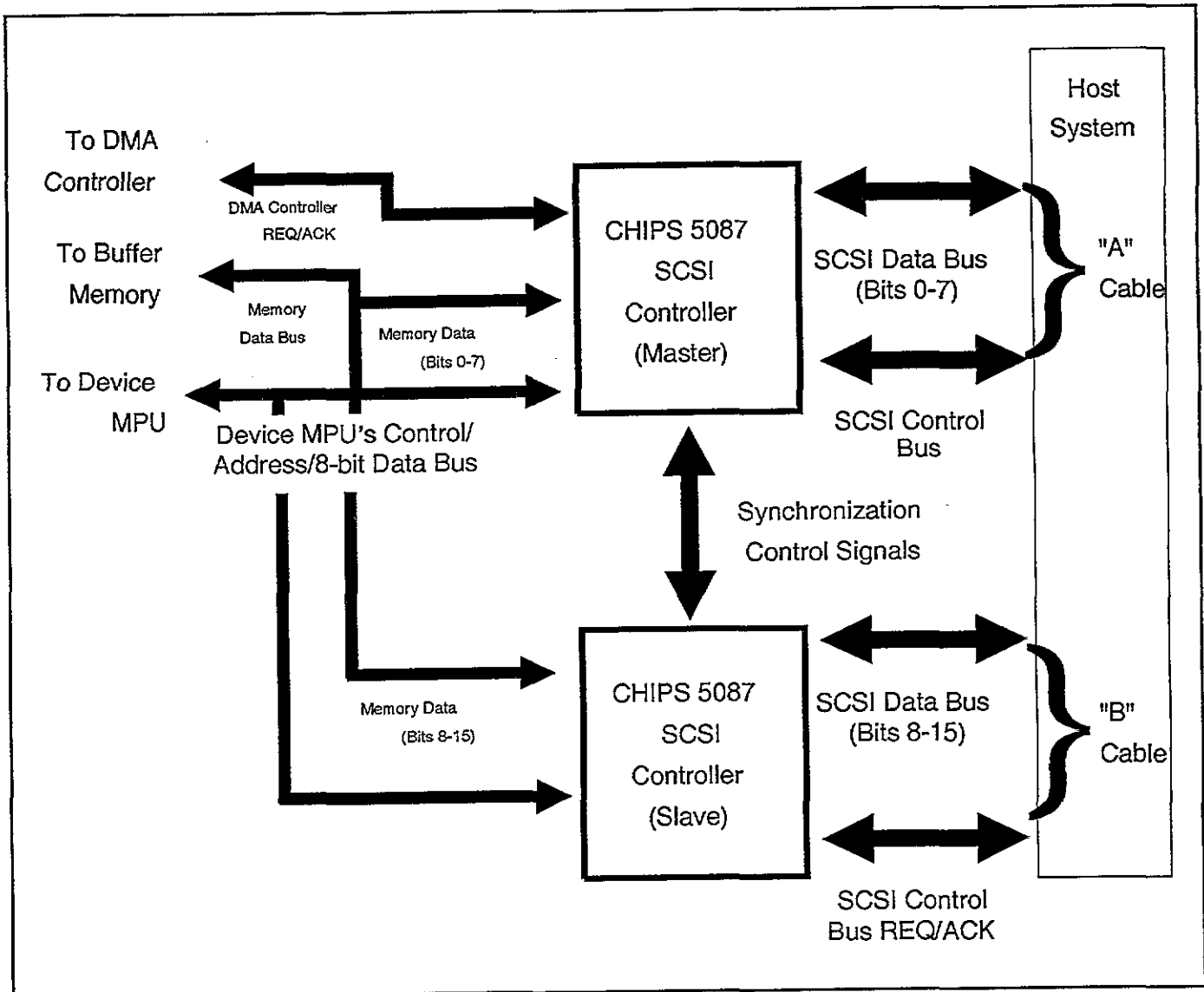
In asynchronous and synchronous transfer mode, one byte of information is transferred over the SCSI bus with each REQ/ACK handshake. The F82C5087's large, internal FIFO acts as a buffer that evens out data transfers between the SCSI bus and buffer memory to support transfers at 4 and 5.3 megabytes/second. For systems with sustained transfers this buffering can increase system performance. Three of the F82C5087's internal registers, Registers 16-18, provide two 24-bit transfer counters that track data transferred to/from the FIFO.

When the F82C5087 is operating in synchronous mode, a transfer rate and an offset must be specified. The F82C5087's clock rate of 33 MHz, results in the F82C5087's unbeatable synchronous transfer rate of 5.3 megabytes/second. The maximum offset is 63 bytes.

Master or Slave Mode

To extend the bandwidth of the SCSI bus and the memory data bus, the F82C5087 can be cascaded. Refer to Figure 1-5. Cascading is supported by the F82C5087's ability to operate in master or slave mode. Through the Mode Control Register, the F82C5087 can be configured as a master or slave chip for its main operational mode (e.g., performing such basic operations as transferring data to/from buffer memory), and for its SCSI bus operational mode. In slave mode, the F82C5087 is under the control of the F82C5087 master chip. For instance, the master F82C5087 is responsible for synchronizing the slaves' transfer of data.

Figure 1-5. Typical Wide SCSI Bus Configuration



System Interface

The F82C5087 contains functional blocks that support these operational modes (e.g., master/slave, synchronous/asynchronous). These blocks provide the logic necessary for the F82C5087 to interface with the other system components, and to manage the flow of SCSI information through the system. They can be grouped into three categories: the I/O interface, the set of F82C5087 commands, and the memory structure.

I/O Interface

It is through its I/O interface that the F82C5087 is able to send/receive signals, addresses, and data to/from the system's other components. This interface

enables the F82C5087 to interface to the buffer memory, the on-board MPU, and the SCSI bus.

Command Set

Through its set of commands, the F82C5087 provides the SCSI protocol for the system flow of command, data, message and status, and allows the firmware to transfer data between the MPU and memory data bus. These commands are arranged in control, initiator, and target groups. Refer to Table 1-1.

Table 1-1.

F82C5087 Command Set

Command	Function	Command	Function
CONTROL COMMANDS			
CNTL NOP FIFO.	No Operation	CNTL RST FIFO	Reset 5087 data
CNTL RST ON	Assert reset signal to SCSI bus.	CNTL RST OFF	Deassert reset signal to SCSI bus.
CNTL ATN ON	Assert ATN signal to SCSI bus.	CNTL ATN OFF	Deassert ATN signal to SCSI bus.
CNTL ENB SEL	Enable selection.	CNTL DIS SEL	Disable selection.
CNTL ENB RESEL	Enable reselection.	CNTL DIS RESEL	Disable reselection.
CNTL ENB ADV	Enable MCS auto advance.	CNTL DIS ADV	Disable MCS auto advance.
CNTL DATA PTM	Receive MPU data and send it to the memory data bus.	CNTL DATA MTP	Receive data from the memory data bus and send it to MPU.
INITIATOR COMMANDS			
INIT SWOA SEQ	Select target without attention sequence.	INIT SWA SEQ	Select target with attention sequence.
INIT SWOA CMD	Select target without attention.	INIT SWA CMD	Select target with attention.
INIT REC INFO	Receive information.	INIT DATA MTS	Receive data from the memory data bus and send it to the SCSI data bus.
INIT DATA PTS	Receive data from MPU and send it to the SCSI data bus.	INIT DATA STM	Receive data from SCSI bus and send it to the memory data bus.
INIT DATA STP	Receive data from SCSI bus and send it to MPU's data bus.	INIT SEND MSG	Send message
INIT SEND CMD	Send multi-byte command.	INIT XFER PAD	Transfer filler data.
INIT XFER BYTE	Transfer byte.	INIT CMDC SEQ	Perform COMMAND COMPLETE/DISCONNECT sequence.
INIT WFR CMD	Wait for reselect phase.	INIT WFR SEQ	Wait to be reselected, then reconnect.

Table 1-1.

F82C5087 Command Set (continued)

TARGET COMMANDS			
TGT SEND MSG	Send message.	TGT SEND STAT	Send status to the initiator.
TGT DATA MTS	Receive data from memory data bus and send it to SCSI bus.	TGT DATA PTS	Receive data from MPU bus and send it to the SCSI bus.
TGT REC CMD	Receive a multi-byte command.	TGT REC MSG	Receive a single-byte or multi-byte message.
TGT DATA STM	Receive data from SCSI bus and send it to memory data bus.	TGT DATA STP	Receive data from SCSI bus and send it to MPU's data bus.
TGT CMDC SEQ	Perform COMMAND COMPLETE sequence.	TGT XFER BYTE	Transfer byte.
TGT DISC CMD	Disconnect from the SCSI bus.	TGT DISC SEQ	Perform DISCONNECT sequence.
TGT RCNT SEQ	Perform RECONNECT sequence.	TGT SEND BSY	Wait for select, then send busy response and disconnect.
TGT WFS CMD	Wait for select.	TGT WFS SEQ	Perform Wait for Select sequence.

Memory Structure

The first segment of F82C5087 memory, the 64-byte FIFO, is used in SCSI bus data transfers, and in data transfers between the MPU's data bus and the memory data bus. The second segment, the MCS, stores SCSI sequences and supports the F82C5087's command queuing capability.

Performance Specifications

When the F82C5087 is operating in synchronous mode, a transfer rate of 5.3 megabytes/second at a clock rate of 33 MHz is supported. The transfer rate in asynchronous mode is 4 megabytes/second (over a 20 foot cable length). Command overhead is 25 microseconds.

Chapter 2

Hardware Specifications

This chapter is directed at hardware engineers intending to design the F82C5087 SCSI-bus interface controller into their systems. Physical and functional pin specifications, signal descriptions, timing specifications, and electrical specifications are provided.

Physical Specifications

The F82C5087 device is available in a 80-pin PFP package. It supports the following functions:

- Asynchronous SCSI data transfers
- Synchronous, as well as faster asynchronous data transfers, are supported. The asynchronous transfer rate with the F82C5087 is 4 megabytes per second. Synchronous data transfer rates at 5.3 megabytes per second are supported by the F82C5087.
- The F82C5087 has on-chip memory that enables up to four F82C5087 commands to be queued.
- The F82C5087 provides 64 bytes of data FIFO to even out data transfers and to increase system performance.
- The F82C5087 command set, which is comprised of 46 commands, performs the SCSI protocol for command, data, message and status flow. This enables the firmware to manage system level activities.
- Supports non-differential, SCSI interface with 48 mA drive capability.
- Connection to MPUs that have non-multiplexed address/data buses, as well as connection to multiplexed MPUs
- Support of wide SCSI and wide memory data bus transfers of 16 or 32 bits, as well as 8-bit data transfers.
- AT host SCSI support through the following signals: -16BHAM, -MPU_ACK, and -MPU_AEN.

Pin Descriptions

Refer to Figure 2-1 for a physical pin out of the 80-pin package. Table 2-1 provides a pin list for the 80-pin package. Note an active low polarity is indicated by a minus sign (-) prefix; a plus sign (+) prefix indicates high polarity. For information on this package's pin functions refer to the "Functional Chip Specifications" section later in this chapter.

Figure 2-1. Physical Pin-out of the 80-pin PFP Package

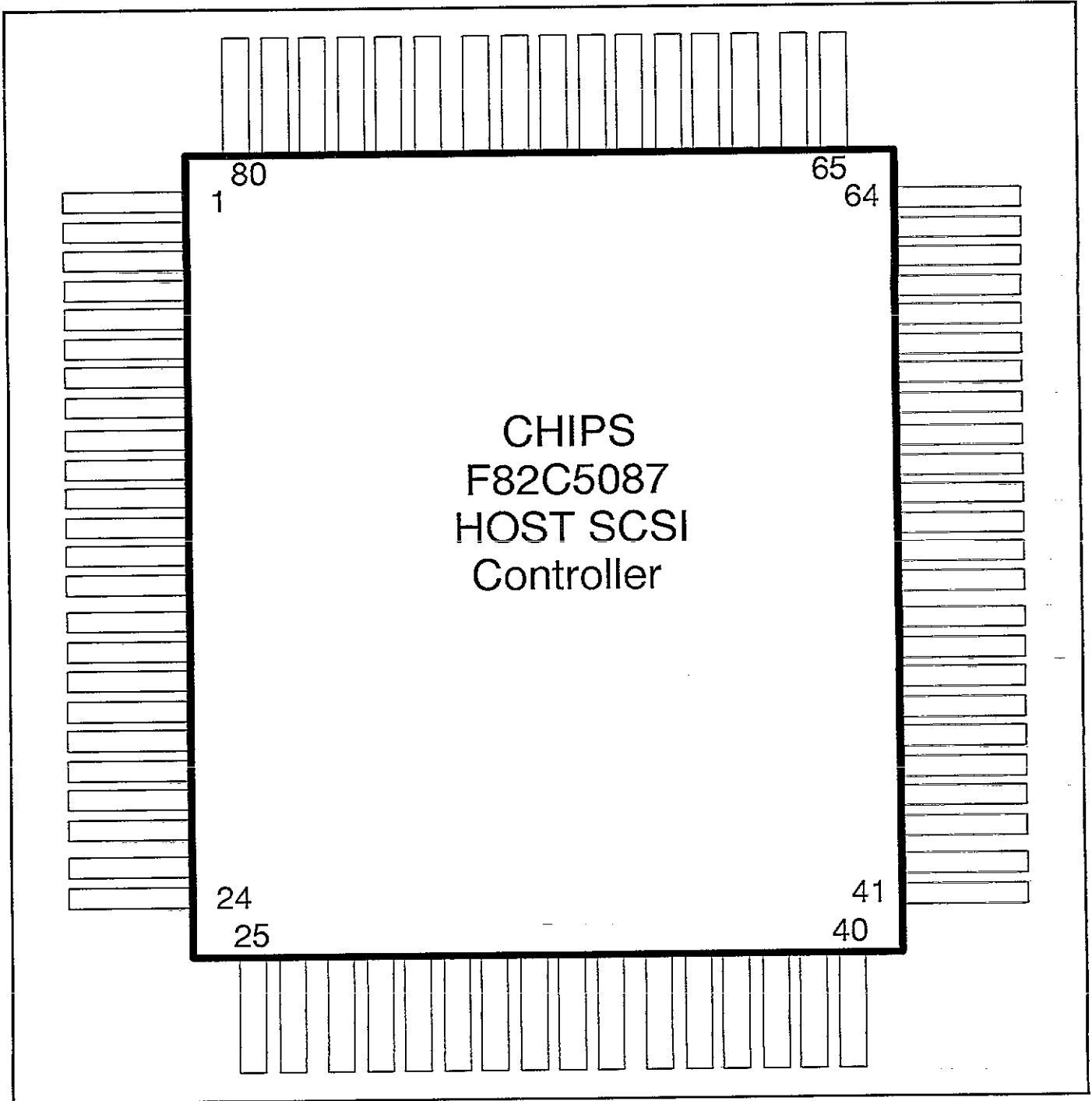


Table 2-1.

A Pin List of the 80-pin Package

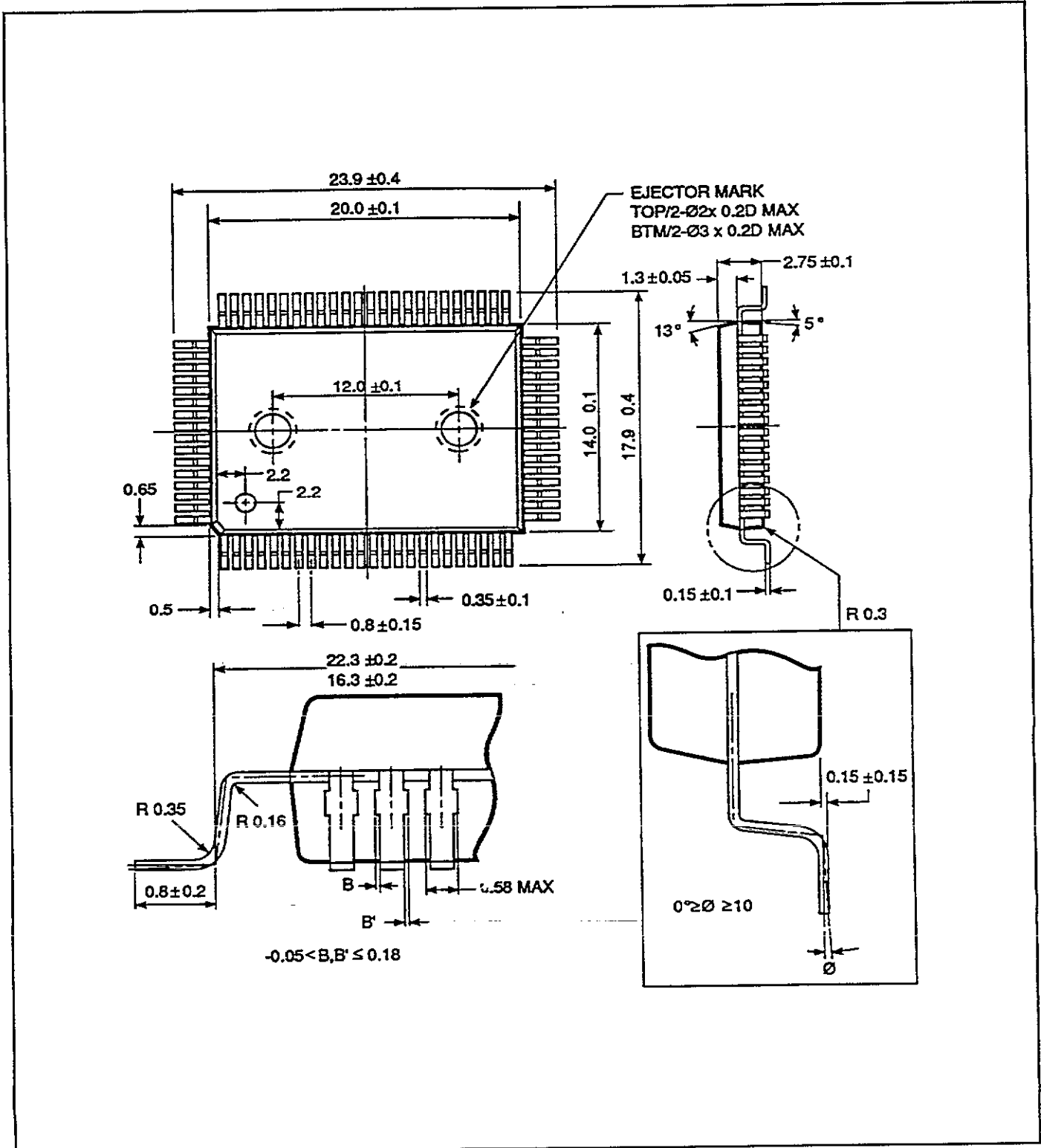
Pin Number	Pin Name	I/O	Pin Number	Pin Name	I/O
1	+INTERRUPT	O	41	-16BHAM	O
2	MPU_AEN	I	42	+INP1[3]/TERMPWR	I/O
3	-MPU_ACK	I	43	+INP1[4]	I/O
4	-XOR[7](A2)	I	44	+BUF_IN	I
5	-XOR[6](A1)	I	45	+IORDY	O
6	-XOR[5](A0)	I	46	+FIFO_RDY	O
7	+CONFIG1	I	47	+A3	I
8	+IO-MEM_-DM	I	48	-DMA_ACK	I
9	-IORD_-DS	I	49	-DMA_REQ	O
10	-IOWR_R-W	I	50	GND_2	I
11	+ALE_-AS	I	51	+MEM_D[0]	I/O
12	GND_0	I	52	+MEM_D[1]	I/O
13	+A_D[0]	I/O	53	+MEM_D[2]	I/O
14	+A_D[1]	I/O	54	+MEM_D[3]	I/O
15	+A_D[2]	I/O	55	VDD_2	I
16	+A_D[3]	I/O	56	+MEM_D[4]	I/O
17	VDD_0	I	57	+MEM_D[5]	I/O
18	+A_D[4]	I/O	58	+MEM_D[6]	I/O
19	+A_D[5]	I/O	59	+MEM_D[7]	I/O
20	+A_D[6]	I/O	60	GND_3	I
21	+A_D[7]	I/O	61	+CONFIG2	I
22	GND_1	I	62	-WPAR	I/O
23	-HOST_D_P	I/O	63	-DSYNC	I/O
24	HOST_D[7]	I/O	64	-ATN	I/O
25	DRV_GND_0	I	65	-BSY	I/O
26	-HOST_D[6]	I/O	66	DRV_GND_3	I
27	-HOST_D[5]	I/O	67	-ACK	I/O
28	-HOST_D[4]	I/O	68	-RESET_IN	I/O
29	-HOST_D[3]	I/O	69	VDD_3	I
30	DRV_GND_1	I	70	-MSG	I/O
31	-HOST_D[2]	I/O	71	-SEL	I/O
32	-SSYNC	I/O	72	DRV_GND_4	I
33	-HOST_D[1]	I/O	73	+A4	I
34	VDD_1	I	74	-C_D	I/O
35	-HOST_D[0]	I/O	75	-REQ	I/O
36	+BUF_OUT	O	76	-I_O	I/O
37	DRV_GND_2	I	77	-RESET_CAP	I
38	+INP1[0]	I	78	DRV_GND_5	I
39	+INP1[1]	I	79	-RESET_OUT	O
40	+INP1[2]	I	80	+CLOCK	I

Note: +INP1[3] could be also configured as the terminator power control through Register 25, bit 1.

Package Specifications

Figure 2-2 provides a diagram of the package specifications for the 80-pin PFP package.

Figure 2-2 80-pin PFP Package Specification



Functional Chip Specifications

The F82C5087 can be used as a host adapter or a device controller. When used as a host adapter, the F82C5087 connects an 8- or 16-bit, host computer data bus to an 8-bit SCSI data bus. As a device controller, the F82C5087 interfaces such peripheral equipment as a Winchester drive to the SCSI bus, and provides a data path between the MPU and memory data buses.

It is through its I/O interface, that the F82C5087 controller can receive/send signals, addresses, and data from/to the system's other components (e.g., DMA controller, MPU). Refer to the following two sections for annotated illustrations of the F82C5087 host adapter.

Functionality of the F82C5087 Host Adapter

As a host adapter, the F82C5087 functions as a bridge between the host computer bus and the SCSI bus. Typically, the host computer requires information (e.g., status) and/or data from the devices connected to the SCSI bus. The F82C5087 host adapter can queue up these requests and send them to the appropriate device(s) via the SCSI bus. Consequently, a host adapter enables SCSI peripheral devices to communicate with host systems that do not directly support the SCSI interface (e.g., IBM PC AT).

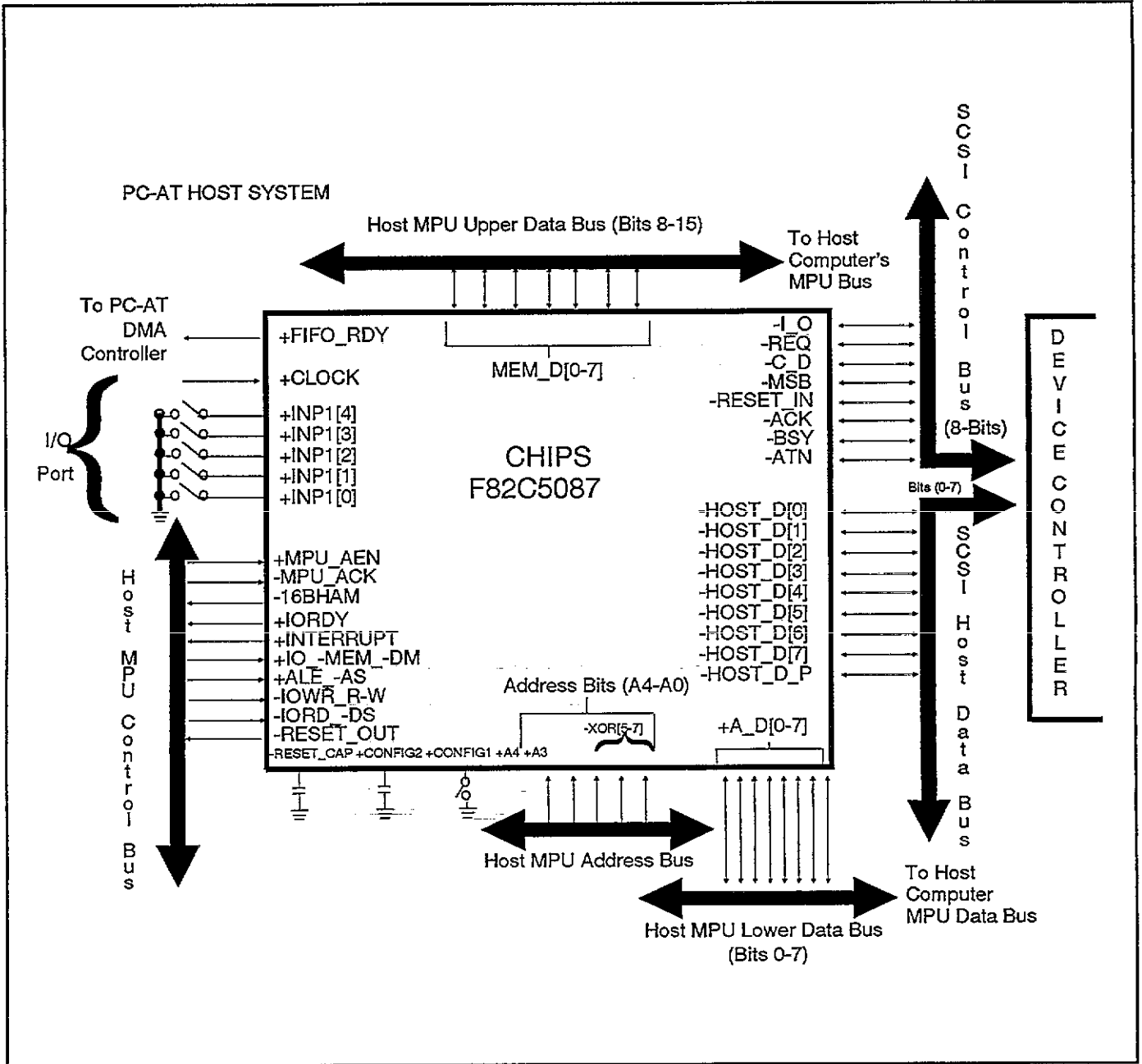
For specifically the AT bus, and to use the DMA on board the AT system, the F82C5087 offers a more optimized host adapter solution. However, the F82C5087 can be used in a host adapter application with a bus master chip and/or bus interface chip for VME, MULTIBUS, MICROCHANNEL, NUBUS, etc.

Refer to Figure 2-3 for a functional pin out of a F82C5087 in an AT host adapter. The F82C5087 host adapter is configured for the following system: (1) the system is operating with 16 bits of MPU data and has an Intel 80286 with a non-multiplexed address/data bus; the bus has been de-multiplexed, and (2) the system has an 8-bit wide SCSI data bus.

As illustrated, the F82C5087 host adapter provides an interface with the following system components:

- SCSI control bus
- SCSI data bus
- Host computer's MPU address/control/data bus
- Host computer's DMA controller.

Figure 2-3. Functional Pin out of the F82C5087 in Host Adapter Mode



Interfacing with the SCSI Control Bus

The SCSI control bus consists of nine bidirectional lines. Firmware can control these SCSI bus control signals (e.g., -BSY and -ATN) by means of the F82C5087 command set. For example, by issuing the CNTL ATN ON (Control Attention On) pipeline command, firmware can assert the -ATN signal on the SCSI bus. By reading the contents of certain F82C5087 internal registers, firmware can monitor the activities and contents of the SCSI bus.

Interfacing with the SCSI Data Bus

SCSI data bus signals enable the F82C5087 host adapter to handle the device controller's requests for information transfers to/from the host computer's memory.

Interfacing with the Host Computer's MPU

IORDY, MPU_AEN, -MPU_ACK, and -16BHAM are the four pins that enable the F82C5087 to interface with the PC-AT host MPU. Whenever the MPU accesses the MPU FIFO Buffer Register and data is not available to be read/written, the IORDY signal is deasserted. Using the IORDY pin is only one of several ways the MPU can check if data is available or if the FIFO is ready for another byte. (The description of MPU FIFO Buffer Register in Chapter 4 describes the other options.)

MPU_AEN and -MPU_ACK are used to override the MPU address bits, and force access to the MPU FIFO Buffer register. These signals enable the F82C5087 to send data to the host (AT) MPU without the host MPU generating a valid address to the MPU FIFO Buffer Register.

The -16BHAM AT host adapter signal enables 16-bit data transfers between the host computer's MPU and the SCSI data bus. This signal is asserted when the MPU read/writes 16 bits of data to/from the F82C5087 host adapter. Bits 0-7 are sent/received via pins A_D[0-7] that comprise the lower MPU data bus. Bits 8-15 are transferred simultaneously via pins MEM_D[0-7], the memory data bus.

CONFIG1 is the line that is pulled up internally, and used to select the MPU's strobe input. This signal is grounded when the system MPU is an Intel 80286, since it uses separate read/write signals.

CONFIG2 is also pulled up internally, and held low because the F82C5087 host adapter is configured to operate with the non-multiplexed MPU address/data bus.

INTERRUPT is a programmable pin that enables the firmware engineer to decide under what conditions (e.g., a parity error) the F82C5087 host adapter should generate an interrupt to the system MPU.

-RESET_CAP is tied to an external capacitor for a reset pulse greater than 5 microseconds at power up. -RESET_OUT is asserted any time the F82C5087 is reset. The F82C5087 can be reset by asserting the -RESET_CAP or -RESET_IN signals, or by a programmed reset.

Interfacing with the Host Computer's DMA Controller

Typically, the FIFO_RDY pin is connected to the host computer's DMA controller. When this pin is asserted, it indicates the F82C5087 is ready to process more data. When FIFO_RDY is asserted, the host DMA controller will send a new byte to a F82C5087 internal register, the MPU FIFO Buffer Register 12. Thus this register can be used to pass data to/from the MPU and the F82C5087's internal FIFO.

Refer to Tables 2-2 through 2-5 later in this chapter for a complete list of the F82C5087's I/O signals and their functions.

Signal Descriptions

Tables 2-1 lists the signals in sequential order. Tables 2-2 through 2-5 list signals in alphabetical order. Note that in all the tables an active low state is indicated by a negative sign (-) prefix. When the pin(s) can be configured for more than one function, such as A_D[0]-A_D[7], the other functions are listed as a subset. An N/A under a pin number indicates that the pin is not available on that particular version of the F82C5087.

Note that in all the signal descriptions, references to direction (e.g., MESSAGE OUT phase) is in reference to the initiator. For example, a MESSAGE OUT phase indicates the initiator wants to send a message to the target. Refer to Table 2-6 for pad drive capacity of the F82C5087's pins.

I/O Signals

Table 2-2 lists the F82C5087 input/output signals and their functions.

Table 2-2.

Input/Output Signals

Signal Symbol	Signal Name	I/O	Pin #	Function
A_D[0-7]	Address/Data Bus	I/O	13-16 18-21	These are active high, tri-state signals. When the active high CONFIG2 pin is high, these address/data lines interface with the MPU's lower 8-bit address/data bus. The addresses are latched into the internal, address register by the falling edge of the ADDRESS LATCH ENABLE signal (ALE). If the address is within the range of the internal chip select, the 8-bit data is written/read to/from a register, depending on the I/O write or I/O read input control lines.
D0-D7	Data Bus	I/O		When CONFIG2 is pulled low, these lines carry only data.
A3	Address Bit 3	I	47	When the active high CONFIG2 signal is low, this active high line provides address bit 3 for internal register selection. Refer to Table 4-1 for 5087 register addresses.
A4	Address Bit 4	I	73	When the active high CONFIG2 signal is low, this active high line provides address bit 4 for internal register selection. Refer to Table 4-1 for the addresses of the 5087 internal registers

Table 2-2.

Input/Output Signals (continued)

Signal Symbol	Signal Name	I/O	Pin #	Function
-ACK	SCSI Acknowledge	I/O	67	The initiator drives this active low signal to indicate acknowledgment of a REQ/ACK handshake for a SCSI data bus transfer. In target mode, the 5087 receives this bidirectional I/O line. In initiator mode, this line is driven by the 5087 in response to the target's request for data to be transferred over the SCSI data bus.
-ATN	SCSI Attention	I/O	64	Initiator asserts this active low, SCSI bus control signal to indicate the ATTENTION condition and to request a MESSAGE OUT phase.
-BSY	SCSI Busy	I/O	65	This active low, SCSI bus control signal indicates the bus is currently in use.
+BUF_IN	Buffer In	I	44	This input signal provides direct control of the BUF_OUT signal if SCSI differential mode is disabled. Its value can be read by the MPU.
+BUF_OUT	Buffer Out	O	36	This open drain 48 mA output line can be driven from the BUF_IN signal in non-differential (single-ended drivers/receivers) SCSI mode. This signal provides the reset out signal to the SCSI bus when differential drivers (multiple-ended drivers) are connected.
-C_D	SCSI Control/Data	I/O	74	In target mode, the 5087 drives this bidirectional I/O line for all SCSI command and data transfers. In initiator mode, the 5087 receives this active low signal that indicates the target has SCSI command/data to transfer.
CLOCK	Clock	I	80	This active high, input signal is for a free running clock used for all internal timing. Recommended clock frequency is 24 to 32 MHz.

Table 2-2.

Input/Output Signals (continued)

Signal Symbol	Signal Name	I/O	Pin #	Function
CONFIG1	Configuration1	I	7	This line is pulled up internally and used to select the MPU's strobe inputs. When this active high, input signal is grounded, the 5087 chip is configured for an 8085/8051-type MPU that uses individual read and write strobes. When this signal is left open, the 5087 chip is configured for a Z8-type MPU that uses separate strobe and read/write signals.
CONFIG2	Configuration2	I	61	When this active high signal is high, the 5087 chip is configured to operate with a multiplexed address and data MPU bus. When this line is low, pins A0-A4 (-XOR[5-7], A3, and A4) are used to select the internal registers. Refer to Table 4-1 for a list of the addresses of the 5087 internal registers. This signal is pulled up internally. Typically it will be pulled up by a resistance of 25 Kohms.
-DMA_ACK	DMA Acknowledge	I	48	This active low, input signal provides the 5087 with the acknowledge handshake signal from the memory data bus DMA controller. When this active low signal is asserted, the 5087 gates data onto the memory data bus (write), or clocks data into (read) the 5087 on the trailing edge of -DMA_ACK.
-DMA_REQ	DMA Request	O	49	This active low, output signal provides the memory data bus DMA controller with a request handshake signal from the 5087.
DRV_GND[0-5]	Drive ground	I	25,30,37 66,72,78	Drive ground. Required around 48 mA drivers.
GND[0-3]	Ground	I	12,22, 50,60	Ground
-HOST_D[0-7]	SCSI Host Data 0-7	I/O	24, 26-29, 31,33,35	These eight active low signals comprise the SCSI data bus. This 8-bit bidirectional data bus is used to transfer parallel data to/from the host system.

Table 2-2.

Input/Output Signals (continued)

Signal Symbol	Signal Name	I/O	Pin #	Function
-HOST_D_P	SCSI Host Data Parity	I/O	23	When the 5087 is driving the SCSI bus with data, this active low signal indicates odd parity (flow through). When the 5087 receives data from the SCSI host data bus, this odd parity bit is used to check for valid parity.
INP1[0-2]	Input Port 1	I	39-40	These active high signals can be used for setup information and are pulled high internally. For example, the SCSI controller ID can be set with this port's bits. These bits are accessible to the MPU by reading register 8.
INP1[3]		I/O	42	If bit 1 of Auxiliary Control 3 Register 25 is set, INP1[3] may be used to control power to SCSI bus terminators. The output is low when the 5087 is active, and high when it is in the sleep mode. If this feature is not being used, it forms bit 3 of the input port.
INP1[4]		I/O	43	The INP1[4] pin can be used for a parity bit on the memory data bus by setting bit 5 of the Mode Control Register 7. Otherwise, it is bit 4 of the input port.
INTERRUPT	Interrupt	O	1	This signal can be programmed to control when the 5087 should generate an interrupt to the MPU. This signal is active low on reset. If it is enabled, output is active when any enabled interrupting sequence is detected. For latched bits, it is cleared when the MPU reads Interrupt Status Register 5. For unlatched bits, it is cleared when the pass-through interrupt condition disappears.
-I_O	SCSI Input/Output	I/O	76	In target mode, this bidirectional, active low I/O line is driven by the 5087 to indicate the target has data/command/status to transfer to the initiating controller. In initiator mode, the 5087 receives this line to indicate the direction of the data transfer.

Table 2-2.

Input/Output Signals (continued)

Signal Symbol	Signal Name	I/O	Pin #	Function									
IO_MEM_-DM	I/O Data Memory	I	8	This chip enable pin controls access to the 5087. Its asserted polarity is discussed in Tables 2-3 and 2-4, and is shown as follows: <table border="1" data-bbox="1146 453 1480 723"> <tbody> <tr> <td></td> <td>CONFIG1 Low</td> <td>CONFIG1 High</td> </tr> <tr> <td>CONFIG2 Low</td> <td>Low</td> <td>High</td> </tr> <tr> <td>CONFIG2 High</td> <td>High</td> <td>Low</td> </tr> </tbody> </table>		CONFIG1 Low	CONFIG1 High	CONFIG2 Low	Low	High	CONFIG2 High	High	Low
	CONFIG1 Low	CONFIG1 High											
CONFIG2 Low	Low	High											
CONFIG2 High	High	Low											
MEM_D[0-7]	Memory Data Bus	I/O	51-54, 56-59	These eight active high signals comprise the 8-bit bidirectional data bus to/from the buffer memory. This high-speed data bus is called the memory data bus. In 16-bit host adapter mode these lines are used as the MPU's upper data bus to transfer bits 8-15 to/from buffer memory. This enables two bytes per cycle to be transferred.									
-MSG	SCSI Message	I/O	70	This active low signal is driven by the target during a MESSAGE phase. In target mode, the 5087 drives this I/O signal to indicate a MESSAGE phase (e.g., MESSAGE IN) is in progress. In initiator mode, the 5087 receives this signal.									
-REQ	SCSI Request	I/O	75	In target mode, the 5087 drives this active low I/O signal to request data transfer to/from the initiator. In initiator mode, the 5087 receives this signal that indicates the target has command/data to transfer.									

Table 2-2.

Input/Output Signals (continued)

Signal Symbol	Signal Name	I/O	Pin #	Function
RESET_CAP	Reset Capacitor	I	77	The 5087 contains an internal POR reset circuit that generates a reset pulse for approximately 5 μ s at power up. If a longer pulse is required, the hardware design engineer should connect this signal to an external capacitor to provide a power-on reset signal of externally controllable pulse width. An internal pull-up resistor of 100 Kohms and a voltage threshold of 3.2 Volts should be used for reset pulse width calculations.
-RESET_IN	SCSI Reset	I/O	68	When this active low SCSI bus control signal is asserted, the 5087 assumes a known reset condition. This signal's unlatched status can be obtained by reading bit 7 (RESET Pin) of Input Port 1 Register 8. The latched status can be read from bit 6 (SCSI RESET) of the Interrupt Status Register 5.
-RESET_OUT	Reset Out	O	79	This active low output signal is always true during power up. It can also be asserted during a SCSI bus reset if the firmware engineer has deasserted bit 4 (Disable SCSI RESET_OUT Pin) of Auxiliary Control Register 19.
-SEL	SCSI Select	I/O	71	Initiator uses this active low signal to select the target, or the target uses it to reselect the initiator.
VDD[0-3]	VDD	I	17,34, 55,69	+5 VDC

Table 2-2.

Input/Output Signals (continued)

Signal Symbol	Signal Name	I/O	Pin #	Function
-XOR[5-7]	Exclusive-OR Address 5-7	I	6-4	When the active high CONFIG2 pin is high (multiplexed MPU bus addressing), these active low input signals are used for internal chip select. They are pulled up internally. They control the polarity of the corresponding address line. For example, if these pins equal 1 1 1, then the 5087 will respond only to addresses below 32.
A0-A2	Address Bits 0-2	I		When the CONFIG2 signal is low, these active high lines connect to address bits 0-2 (A0-A2). Address bits 0-2 are used for internal register selection.
FIFO_RDY	FIFO Request	O	46	This tri-state active high signal is asserted when the FIFO is waiting for a new byte from the MPU, or it has a byte ready to send to the MPU. When the MPU FIFO Buffer Register 12 is accessed, this signal is cleared. This pin is enabled by bit 2 (FIFO RDY Pin) of the Mode Control Register 7.
IORDY	I/O Ready	O	45	This open drain pin sinks 24 mA. It is enabled by bit 0 (Host Adapter Pins) of the Mode Control Register 7. Whenever the MPU accesses the MPU FIFO Buffer Register 12 and data is not available to be read/written, this signal is deasserted. Hardware engineers should tie this output signal to the MPU Wait State Generator.
-16BHAM	16-bit Host adapter mode	O	41	This open drain pin is active low, sinks 24mA. Its output is asserted when the MPU reads/writes 16-bit data from/to the host adapter (MPU FIFO Data Register 12).
+MPU_AEN	MPU Address Enable	I	2	This active high signal is used as a qualifier to override the MPU address bits. It is internally pulled up.

Table 2-2.

Input/Output Signals (continued)

Signal Symbol	Signal Name	I/O	Pin #	Function
-MPU_ACK	MPU Acknowledge	I	3	This active low signal is used as a qualifier to override the MPU address bits and also forces an internal chip select. It is internally pulled up.

Signals Specific to 8085/8051 Mode

Table 2-3 describes the signals used only when the F82C5087 is operating in 8085/8051 mode (multiplexed mode). Note the 8085/8051 uses multiplexed address/data buses.

Note: In this mode the CONFIG1 signal is low.

Table 2-3.

Signals Specific to 8085/8051 Mode

Signal Symbol	Signal Name	I/O	Pin	Function
ALE_-AS	Address Latch Enable	I	11	This input strobe is for latching the address from the MPU into the internal address register on the falling edge of ALE. The address bits are used for internal chip and register selection.
IO_-MEM_-DM	I/O Data Memory	I	8	If true (high), this input signal enables I/O Read and I/O Write signals. If this I/O Data Memory signal is false (low), I/O Read and I/O Write signals are disabled.
-IORD_-DS	I/O Read	I	9	MPU uses this active low input signal to read 5087 registers.
-IOWR_R-W	I/O Write	I	10	MPU uses this active low input signal to write 5087 registers.

Signals Specific to Z8 Mode

Table 2-4 provides signal descriptions for signals used only when the F82C5087 is interfacing with a Z8 microprocessor (multiplexed mode).

Note: In this mode the CONFIG1 signal is high.

Table 2-4.

Signals Specific to Z8 Mode

Signal Symbol	Signal Name	I/O	Pin #	Function
ALE_-AS	Address Strobe	I	11	This active low input strobe is for latching the address from the MPU into the internal address register. Address bits are used for internal chip and register selection.
IO_-MEM_-DM	Data Memory	I	8	If true (low), this input signal enables data strobe (DS). If this active low signal is false (high), data strobe is disabled.
-IORD_-DS	Data Strobe	I	9	If R/W is true (high), the MPU uses this active, low input to read information from the 5087. If R/W is false (low), the MPU uses it to load information into the 5087.
-IOWR_R-W	Read/Write	I	10	This active high input signal determines whether the MPU wants to perform a read operation (R/-W is true), or write operation (R/-W is false).

Signals Specific to Wide Data Bus Transfers

Table 2-5 describes signals used only when two or more 5087s are cascaded to support wide SCSI bus data transfers. Refer to Figure 1-3 for a diagram of a system configuration with two 5087s cascaded to support 16-bit data transfers over the SCSI bus. Figure 3-2 provides a functional pin out of 5087s cascaded.

Table 2-5.

Signals Specific to Wide Data Bus Transfers

Signal Symbol	Signal Name	I/O	Pin #	Function
-DSYNC	DMA Sync	I/O	63	<p>The master 5087 does not generate a DMA request (assert -DMA_REQ) signal until this active low input signal is deasserted.</p> <p>The slave(s) 5087 asserts this signal to indicate a not ready condition. Hardware engineers should tie this pin on all master and slave 5087s together.</p> <p>In master mode, this pin is an input and is pulled high. In slave mode, this pin is an open drain output. For more details refer to Chapter 3's section "Master/Slave Mode."</p>
-SSYNC	SCSI Sync	I/O	32	<p>The master 5087 does not generate a SCSI request/acknowledge (REQ/ACK) signal until this active low input signal is deasserted.</p> <p>To indicate a not ready condition the slave(s) 5087 asserts this signal. Hardware engineers should tie this pin on all master and slave 5087s together.</p> <p>In master mode, this pin is an input and is pulled high. In slave mode, this pin is an open drain output. For more details refer to Chapter 3's section "Master/Slave Mode."</p>
-WPAR	Wide Data Bus Parity	I/O	62	<p>In asynchronous or synchronous mode, this active low input signal indicates the external 5087 slave chip(s) has a data parity error or over/under run error.</p> <p>In master mode, this signal is an input and is pulled high. In slave mode, this signal is an open drain output. Hardware engineers should tie this pin on all masters and slave 5087s together.</p>

Pad Drive Capacity

Refer to Table 2-6 for a list of the pad drive capacity of the F82C5087 pins.

Table 2-6.

Pad Drive Capacity

Pin Name	Drive Pad Capacity (mA)	Pin Name	Drive Pad Capacity (mA)
-ACK	48	+INR_ENB	4
-ATN	48	+INTERRUPT	16
-BSY	48	+DIF_DIR	4
+BUF_OUT	48	+DMA_REQ	4
-C_D	48	+A_D[0-7]	8
-HOST_D[0-7]	48	-HOST_D_P	48
-I_O	48	+ARB	4
-MSG	48	-RESET_OUT	4
-REQ	48	+TAR_ENB	4
-RESET_IN	48	+INP1[4]	8
-SEL	48	-DSYNC	4
IORDY	24	-WPAR	4
+FIFO_RDY	16	-SSYNC	4
-16BHAM	24	INP1[3]	16
+A_D[0-7]	24	+MEM_D[0-7]	24

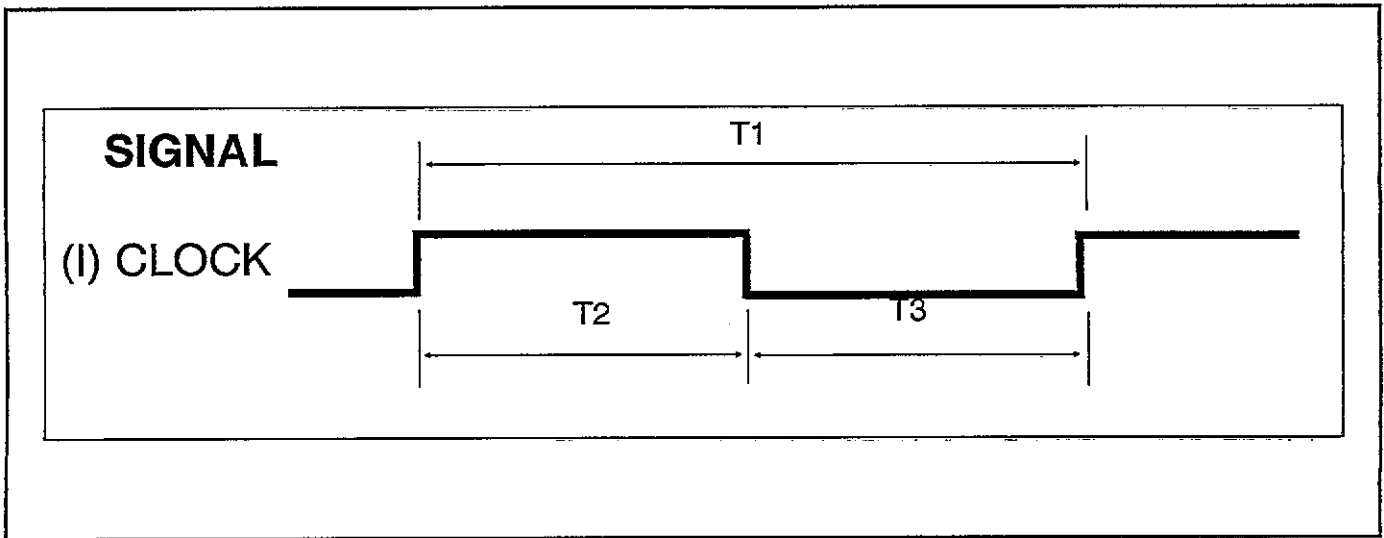
Electrical Specifications

The relevant timing diagrams and A.C. characteristics for interfacing the F82C5087 are provided in the following timing specification sections. For more information on the microprocessor timing refer to the documentation for the particular microprocessor.

Clock Timing

The F82C5087 chip has a CLOCK signal for a free running clock that is used for all internal timing. The recommended clock frequency for the F82C5087 chip is 24 to 33 MHz. Figure 2-4 presents the waveform and AC characteristics of the CLOCK signal timing.

Figure 2-4. Clock Timing Characteristics

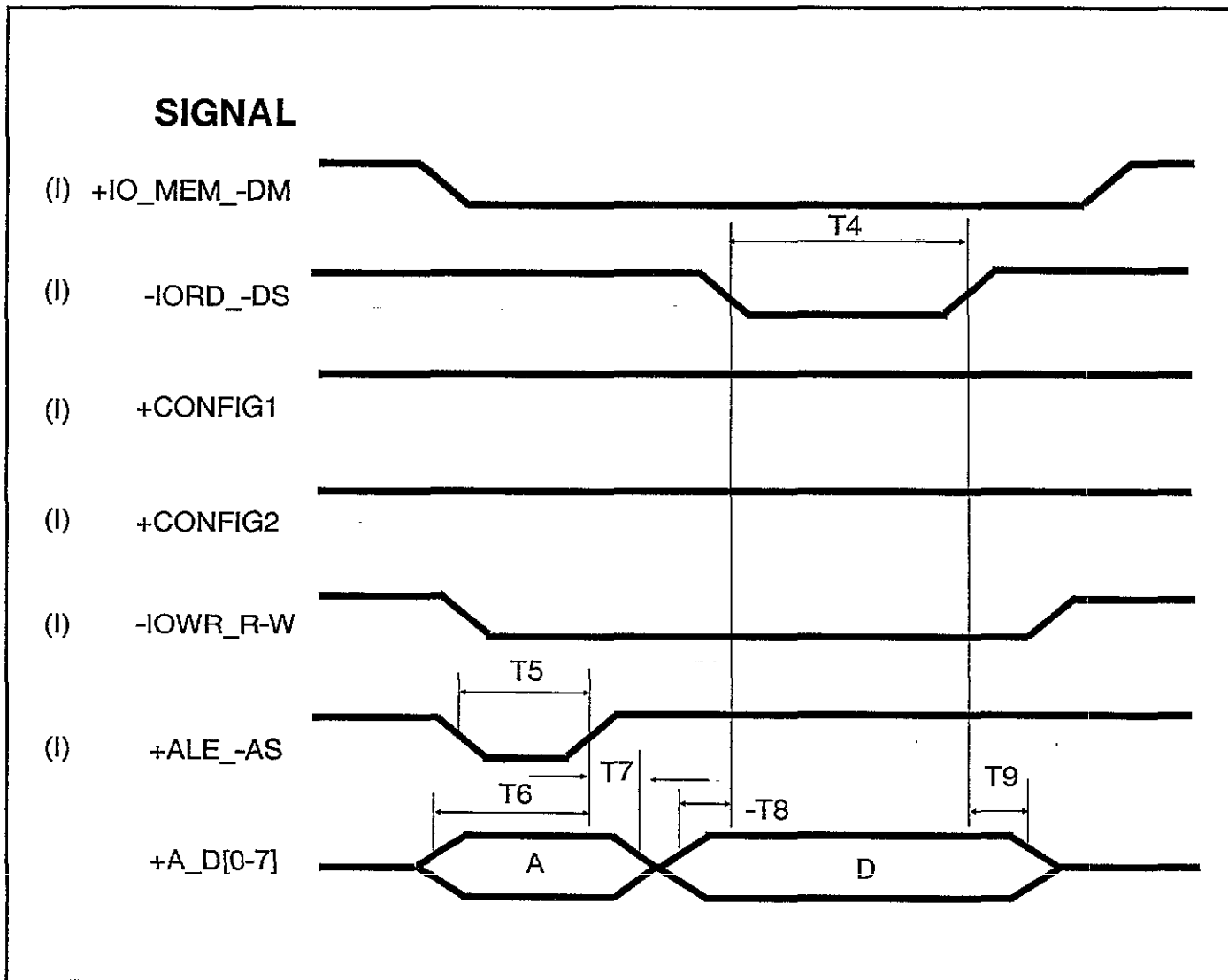


Symbol	Parameter	Min.	Typ.	Max.	Unit
T1	Clock Period (CP)	30			ns
T2	Clock pulse width high	40% of CP			ns
T3	Clock pulse width low	40% of CP			ns

Z8 Mode Timing

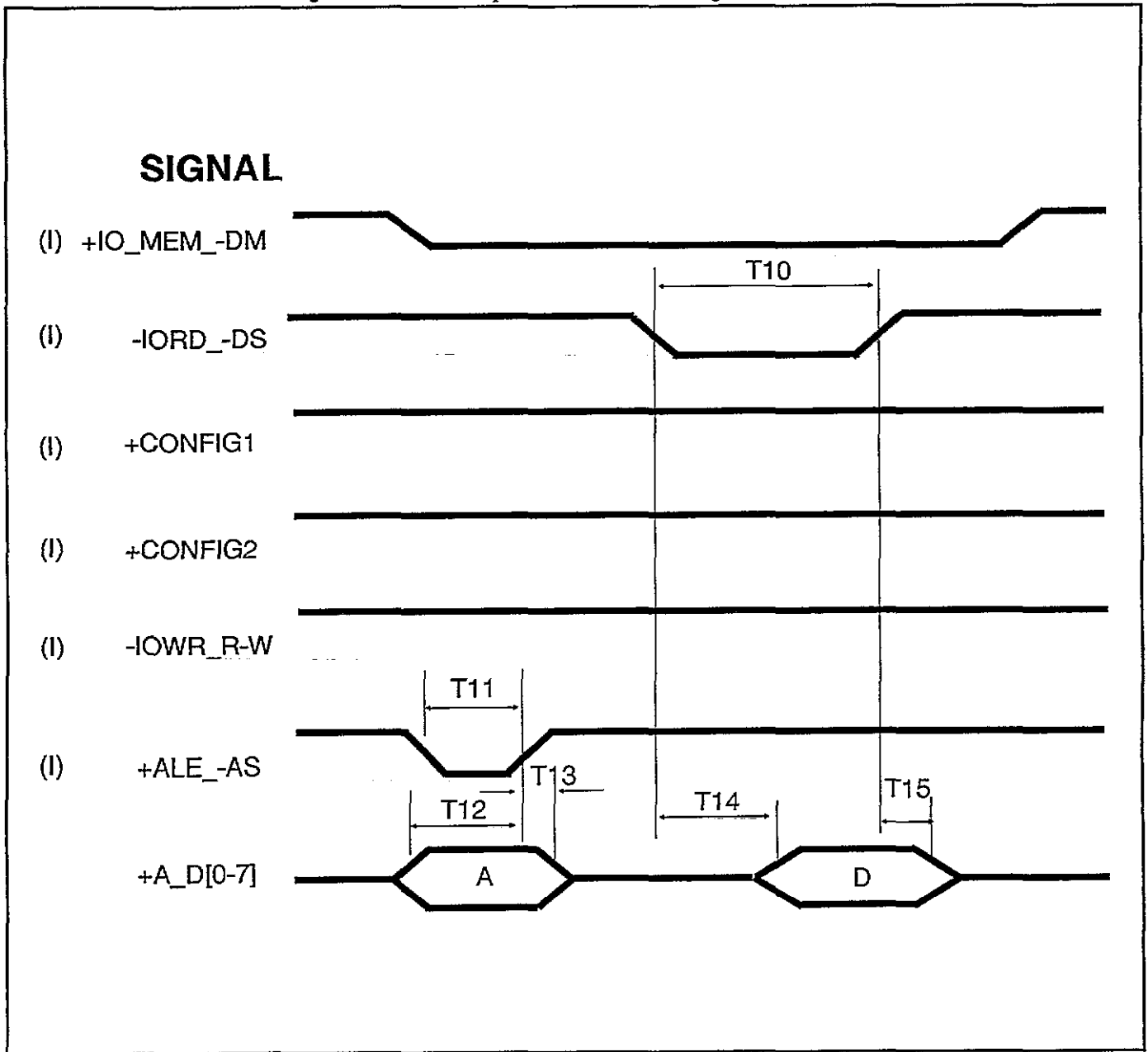
When the F82C5087 is interfacing with the Z8 that has a multiplexed address/data bus, the F82C5087 operates in multiplexed mode. Figures 2-5 and 2-6 respectively present signal timings and their AC characteristics for Z8 mode write and read operations. For more details on these signals, refer to in the section "Signals Specific to Z8 Mode" earlier in this chapter.

Figure 2-5. Write Operation Z8 Mode Timing Characteristics



Symbol	Parameter	Min.	Typ.	Max.	Unit
T4	-IOWR_-DS low pulse width	60			ns
T5	+ALE_-AS low pulse width	30			ns
T6	Address setup to +ALE_-AS high	25			ns
T7	Address hold after +ALE_-AS high	25			ns
T8	Data setup to -IORD_-DS low	25			ns
T9	Data hold after -IORD_-DS high	25			ns

Figure 2-6. Read Operation Z8 Mode Timing Characteristics

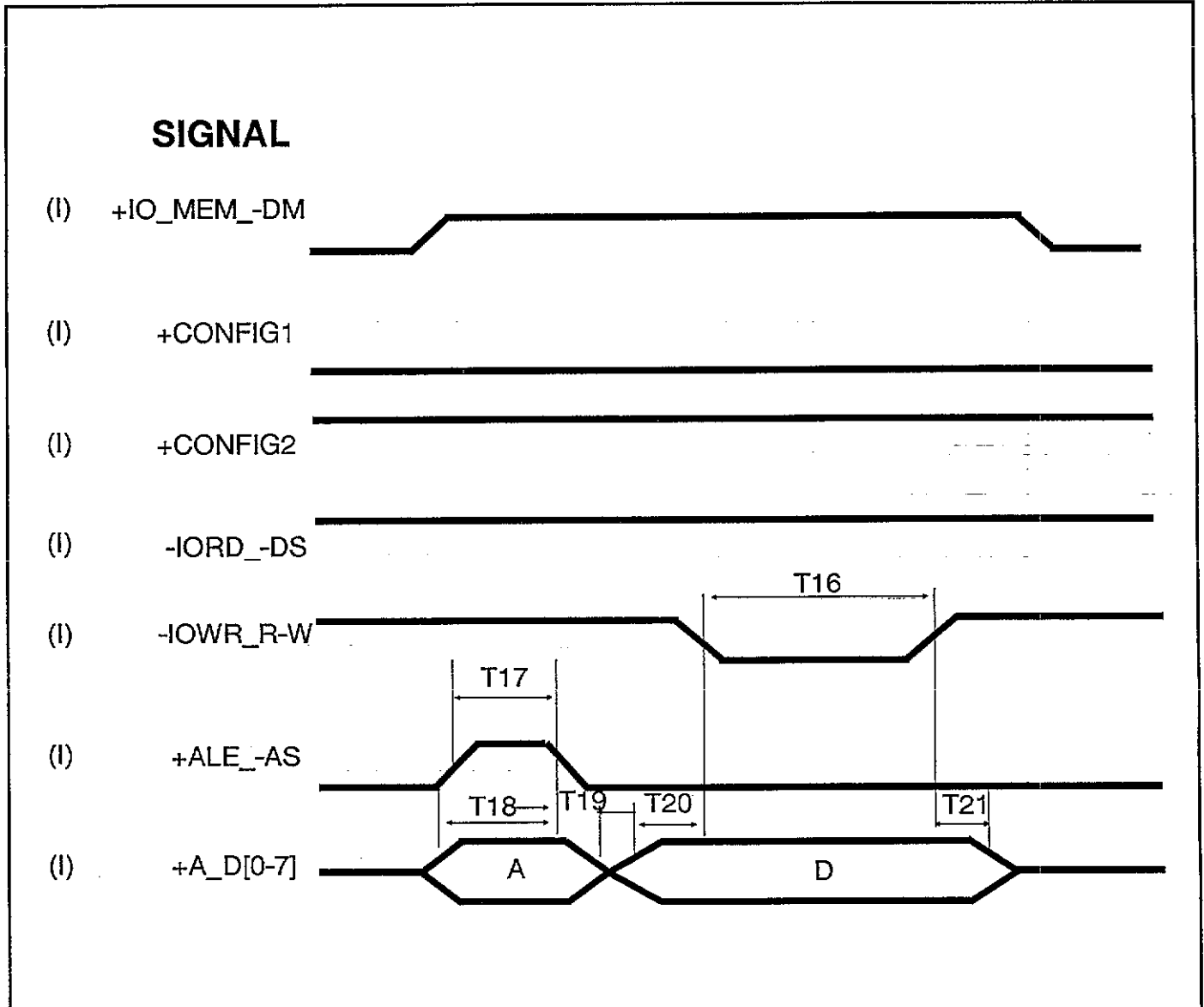


Symbol	Parameter	Min.	Typ.	Max.	Unit
T10	-IORD_DS low pulse width	60			ns
T11	+ALE_AS low pulse width	30			ns
T12	Address setup to +ALE_AS high	25			ns
T13	Address hold after +ALE_AS high	25			ns
T14	Data valid from -IORD_DS low			50	ns
T15	Data float after -IORD_DS high		35		ns

8085/8051 Mode Timing

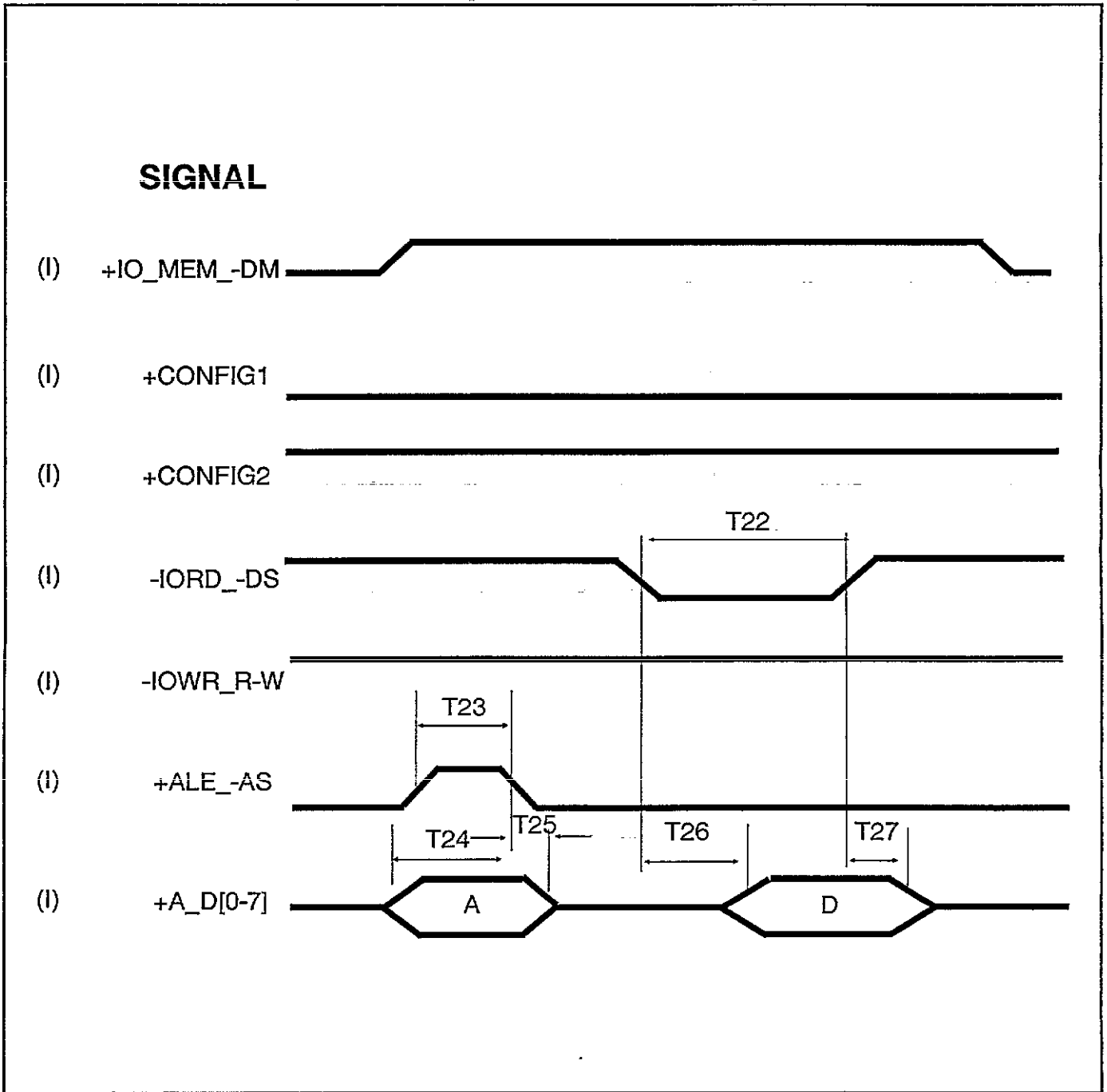
The F82C5087 can operate in 8085/8051 mode (multiplexed mode). Figures 2-7 and 2-8 respectively present the signal timings and their AC characteristics for 8085/8051 mode write and read operations.

Figure 2-7. Write Operation 8085/8051 Mode Timing Characteristics



Symbol	Parameter	Min.	Typ.	Max.	Unit
T16	-IOWR_R-W low pulse width	60			ns
T17	+ALE_AS high pulse width	30			ns
T18	Address setup to +ALE_AS low	25			ns
T19	Address hold after +ALE_AS low	25			ns
T20	Data setup to -IOWR_R-W low	0			ns
T21	Data hold after -IOWR_R-W high	25			ns

Figure 2-8. Read Operation 8085/8051 Mode Timing Characteristics

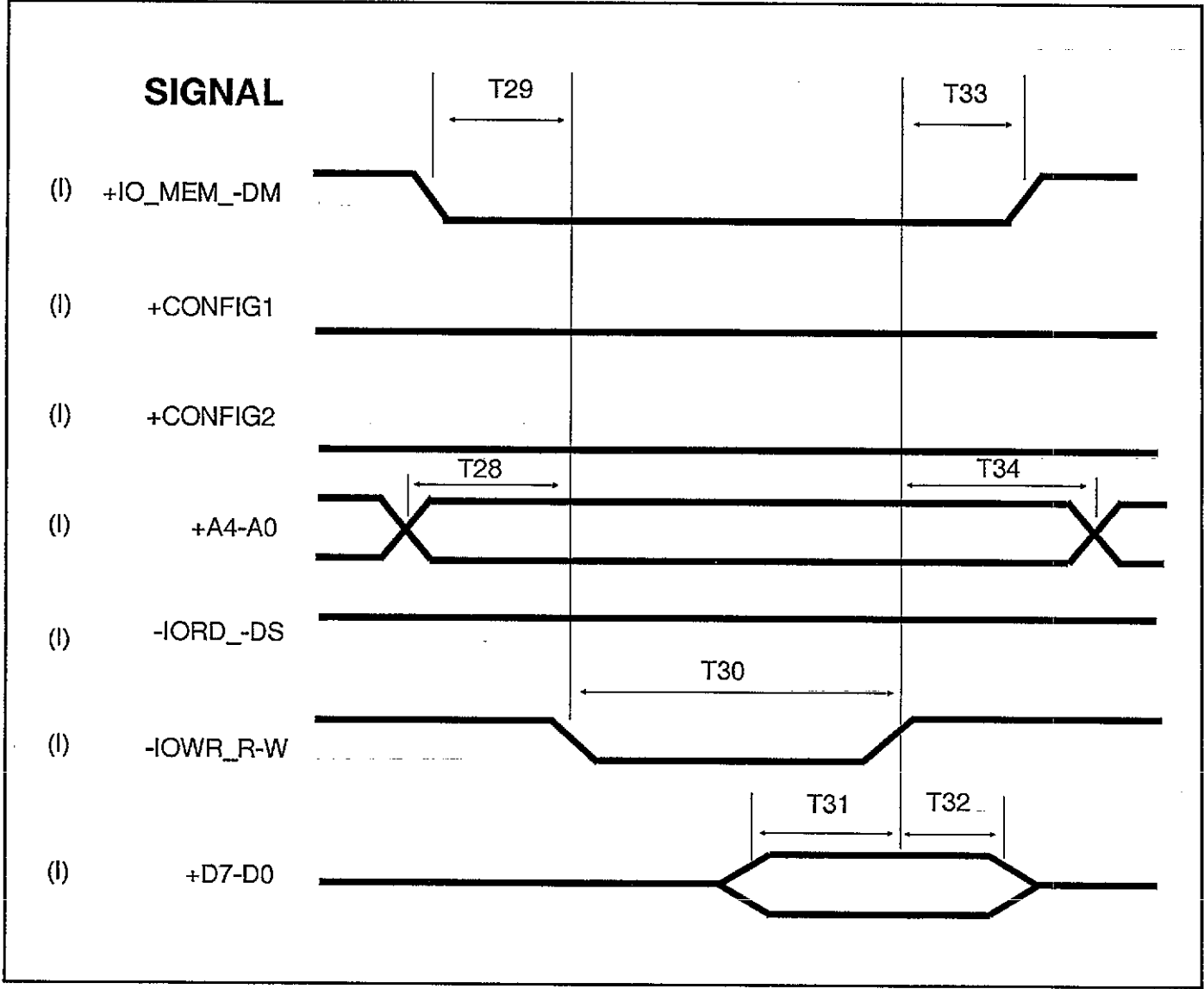


Symbol	Parameter	Min.	Typ.	Max.	Unit
T22	-IOWR_R-W low pulse width	60			ns
T23	+ALE_-AS high pulse width	30			ns
T24	Address setup to +ALE_-AS low	25			ns
T25	Address hold after +ALE_-AS low	25			ns
T26	Data valid from -IORD_-DS low			50	ns
T27	Data float after -IORD_-DS high		35		ns

Non-multiplexed Mode Timing

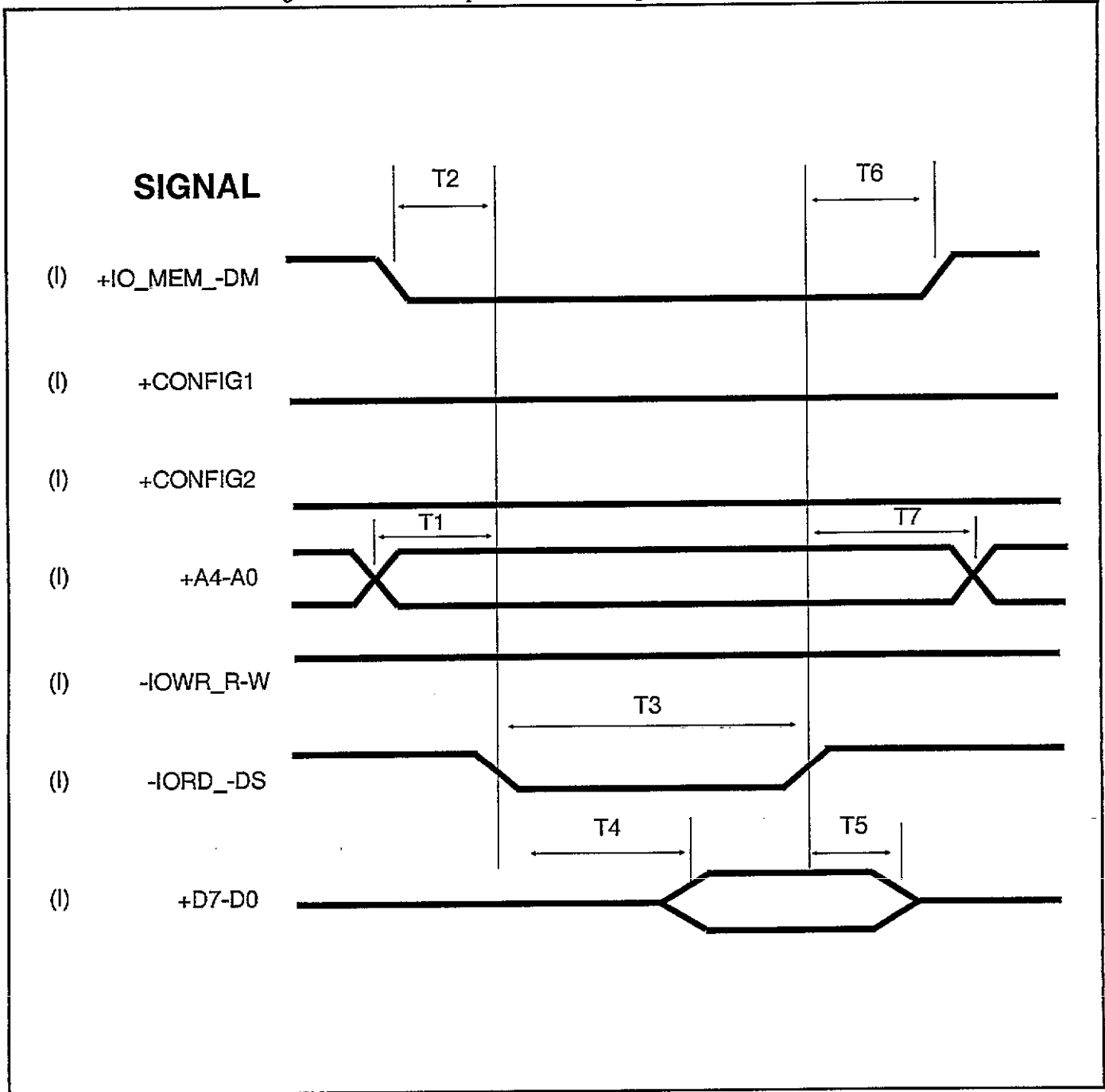
The F82C5087 chip can operate in non-multiplexed mode in order to interface with microprocessors that have non-multiplexed address/data buses. Figures 2-9 and 2-10 respectively present signal timings and their AC characteristics of write and read operations for non-multiplexed mode.

Figure 2-9. Write Operation, Non-multiplexed MPU Timing Characteristics



Symbol	Parameter	Min.	Typ.	Max.	Unit
T28	Address setup	30			ns
T29	+IO_MEM_-DM setup	20			ns
T30	-IOWR_R-W width	2CP			ns
T31	Data bus setup	30			ns
T32	Data bus hold	20			ns
T33	+IO_MEM_-DM hold	10			ns
T34	Address hold	10			ns

Figure 2-10. Read Operation, Non-multiplexed MPU Timing Characteristics



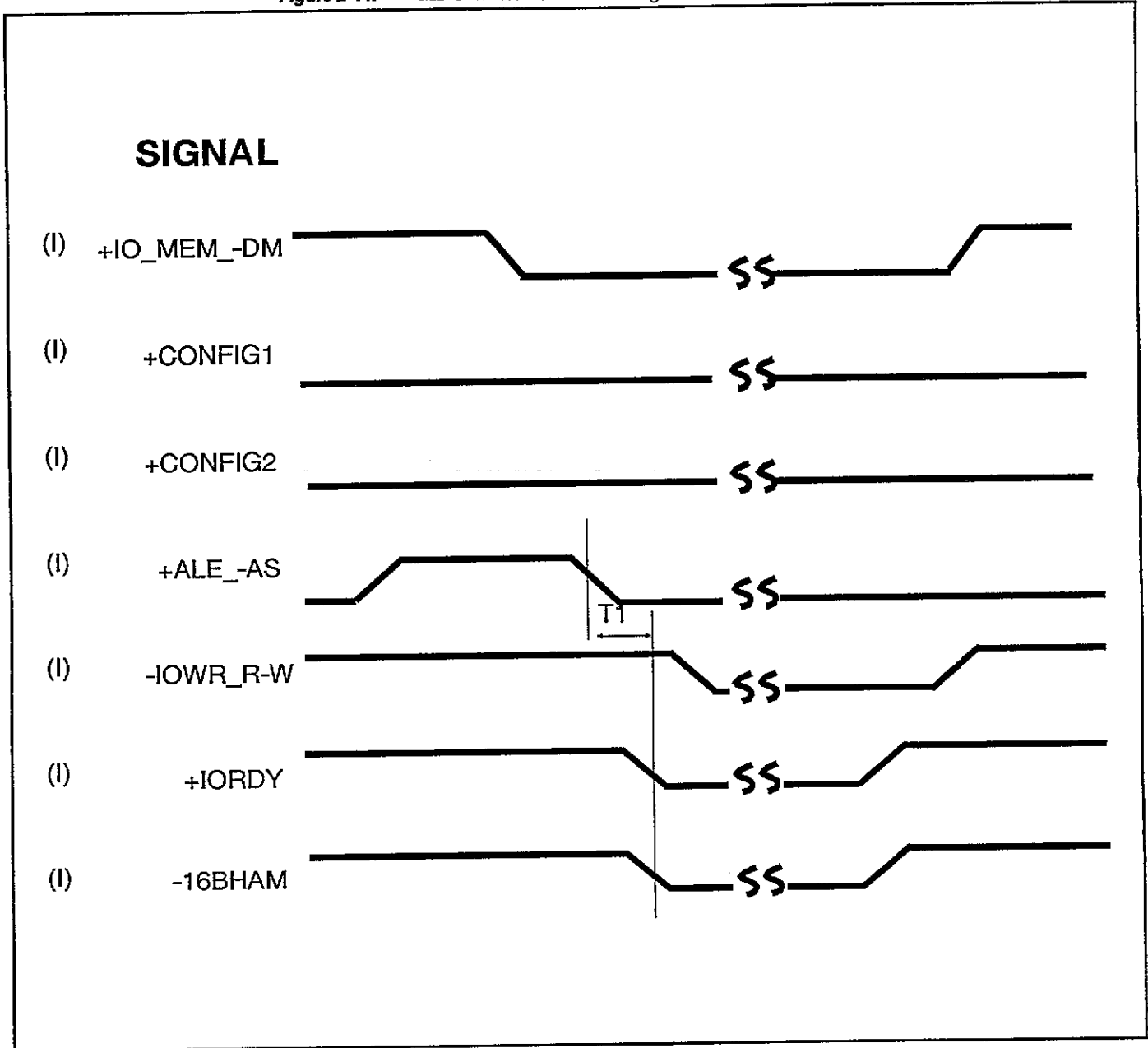
Symbol	Parameter	Min.	Typ.	Max.	Unit
T1	Address setup	30			ns
T2	+IO_MEM_-DM setup	20			ns
T3	-IORD_-DS width	2CP			ns
T4	Data valid from strobe	10		50	ns
T5	Data float after strobe		35		ns
T6	+IO_MEM_-DM hold	10			ns
T7	Address hold	10			ns

Timing for Data Transfers to System Memory

System design engineers have two alternative ways of wiring the F82C5087 to support data transfers to/from system memory. First, the F82C5087's IORDY pin can be connected to the MPU's wait state generator to support burst transfers (string operation) from the MPU to the F82C5087 chip. Figures 2-11 and 2-12 respectively present the signal timings and their AC characteristics of the IORDY signal and its associated signals for a MPU write and read operation.

The second method of supporting data transfers to/from system memory is to connect the F82C5087's FIFO_RDY pin to a DMA controller on the system's memory data bus. Figures 2-13 and 2-14 respectively present the signal timings and AC characteristics for the FIFO_RDY signal and its associated signals.

Figure 2-11. MPU Write IORDY Timing Characteristics

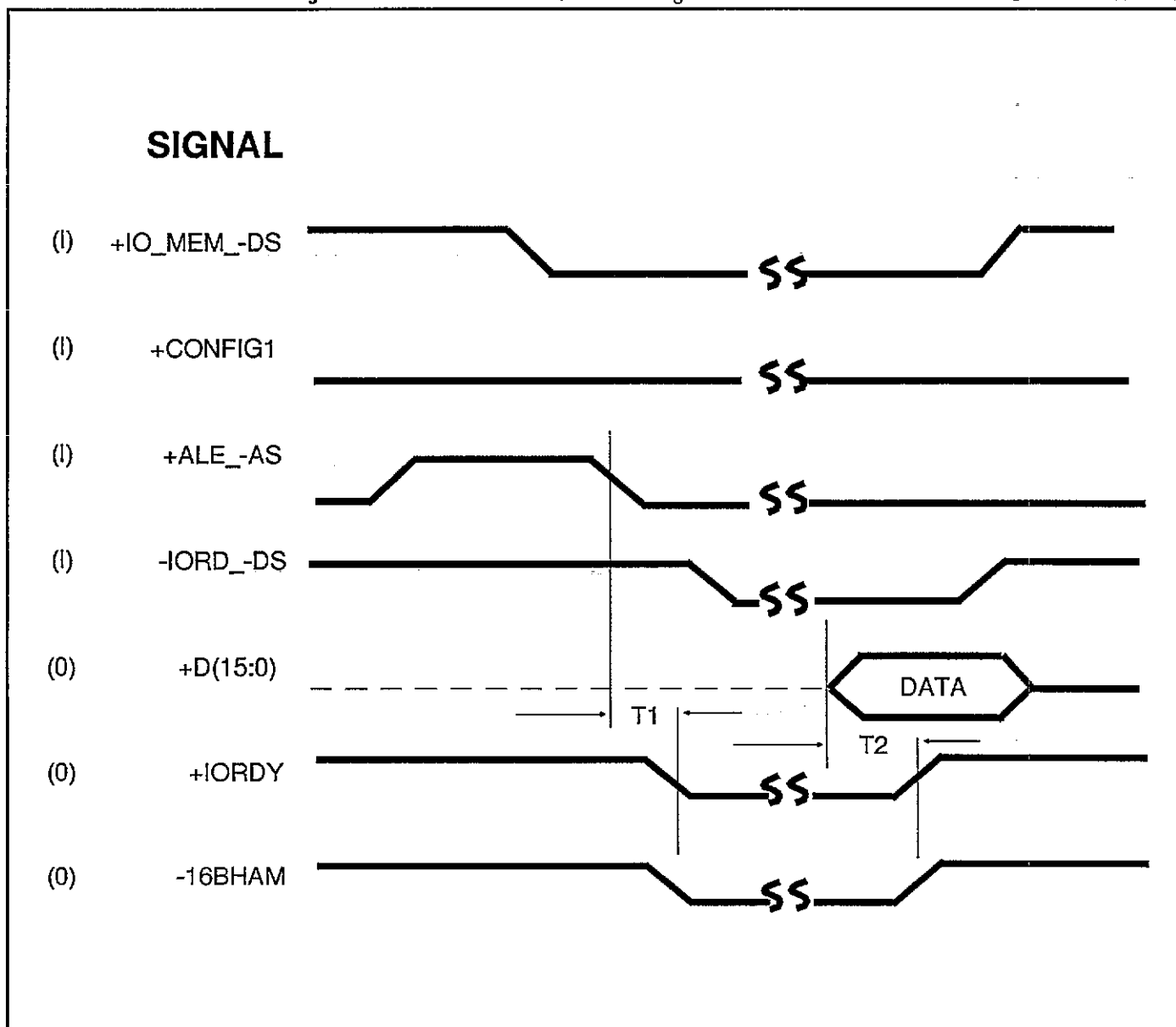


Symbol	Parameter	Min.	Typ.	Max.	Unit
T1	+IO_MEM_DM and +ALE_AS deasserted to IORDY low			50	ns

Note:

1. +IORDY is driven low only if the pin is enabled and the 5087 is not ready to accept more data.
2. +IORDY will remain low until the 5087 accepts data.
3. +IORDY will only be low for Register 11 or 12 access.
4. +IORDY is not a function of -IOWR_R-W
5. -16BHAM pin must be enabled and a 16-bit transfer requested.

Figure 2-12. MPU Read IORDY Timing Characteristics

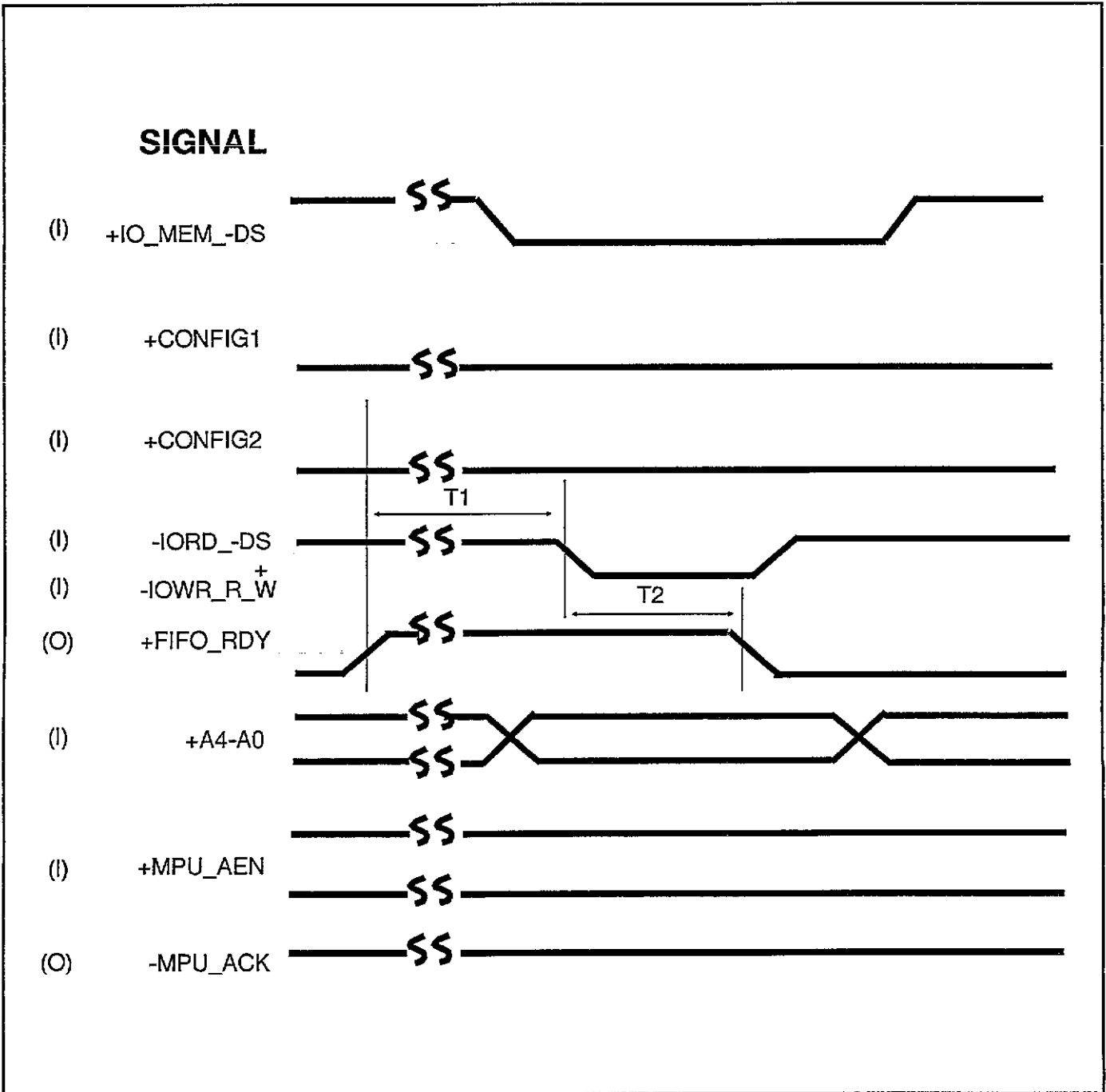


Symbol	Parameter	Min.	Typ.	Max.	Unit
T1	+IO_MEM_-DS and +ALE_-AS deasserted to +IORDY low			50	ns
T2	Data valid to +IORDY high	10			

Note:

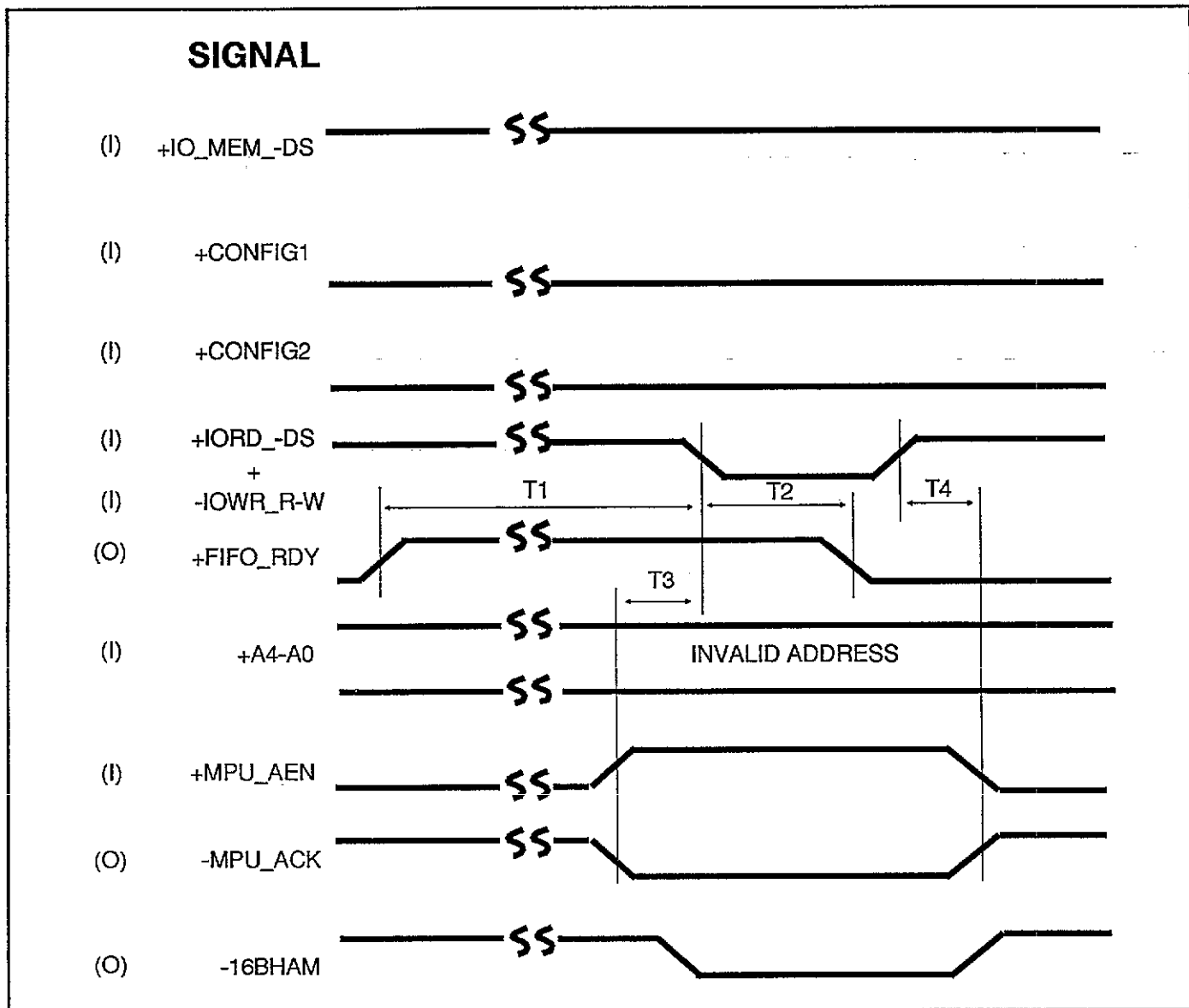
1. +IORDY is driven low only if the pin is enabled and the 5087 does not have data to send to the MPU.
2. +IORDY will remain low until data is available.
3. +IORDY will only be low for Register 11 or 12 access.
4. +IORDY is not a function of -IOWR_-DS.
5. -16BHAM pin must be enabled and a 16-bit transfer requested.

Figure 2-13. FIFO_RDY with Address Timing Characteristics



Symbol	Parameter	Min.	Typ.	Max.	Unit
T1	+FIFO_RDY high to access strobe	0			ns
T2	Access strobe low to +FIFO_RDY low			CP+80	ns

Figure 2-14. FIFO_RDY without Address Timing Characteristics



Symbol	Parameter	Min.	Typ.	Max.	Unit
T1	+FIFO_RDY high to access strobe	0			ns
T2	Access strobe low to +FIFO_RDY low			CP+80	ns
T3	+MPU_AEN high and -MPU_ACK low to access strobe	95			ns
T4	+MPU_AEN and -MPU_ACK hold time	30			ns

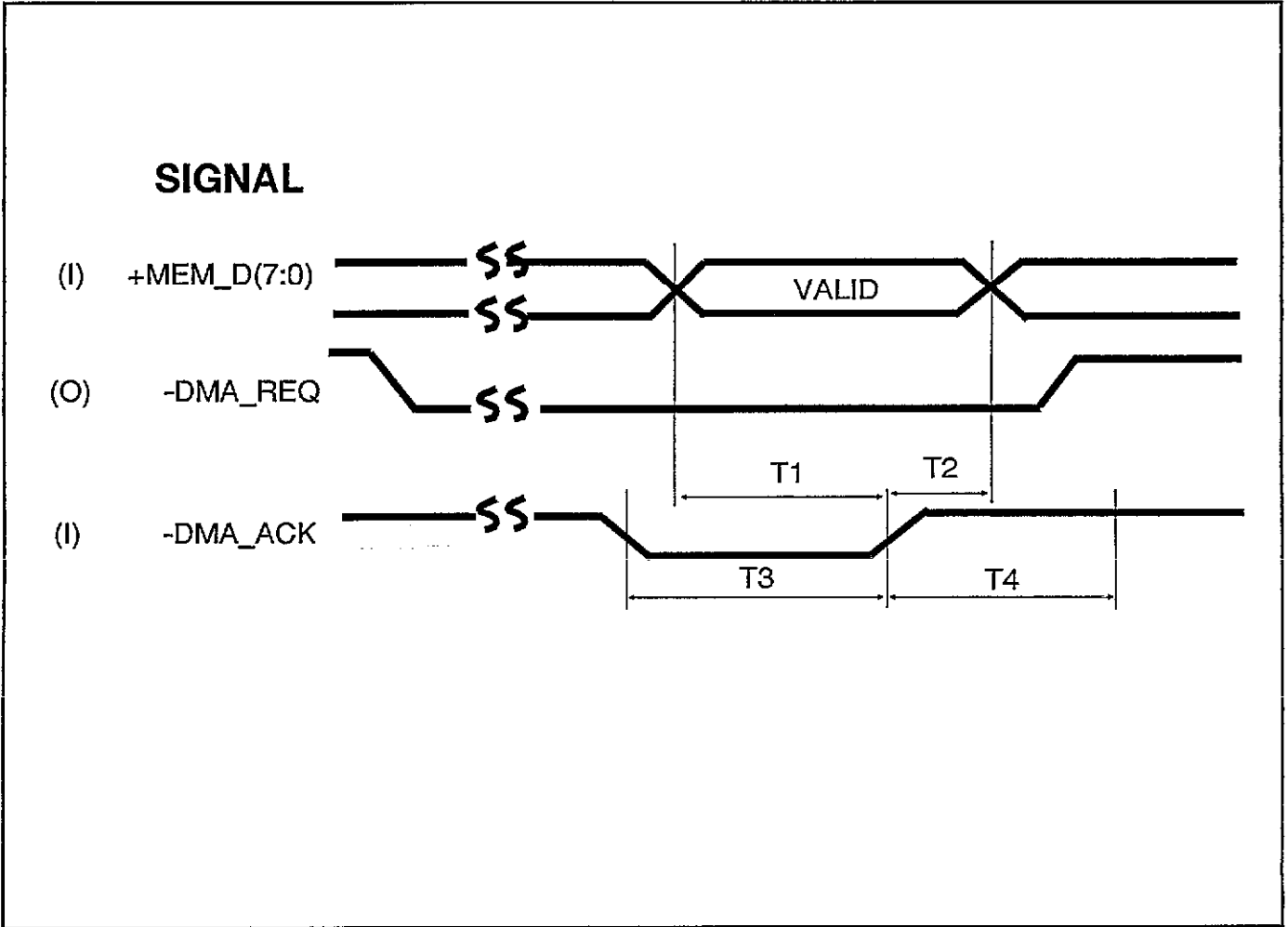
Note:

- +MPU_AEN high and -MPU_ACK low and +FIFO_RDY high and access strobe forces Register 12 access.
- 16BHAM pin must be enabled in order to be asserted and a 16 bit transfer is requested.

Timing for I/O Interface to Buffer Memory

The F82C5087 communicates with the DMA controller over a bidirectional memory data bus via DMA request/acknowledge handshake control signals. Figures 2-15 and 2-16 respectively present the signal timings and their AC characteristics for a write and a read operation to/from buffer memory.

Figure 2-15. DMA Interface, Write to F82C5087 Timing Characteristics

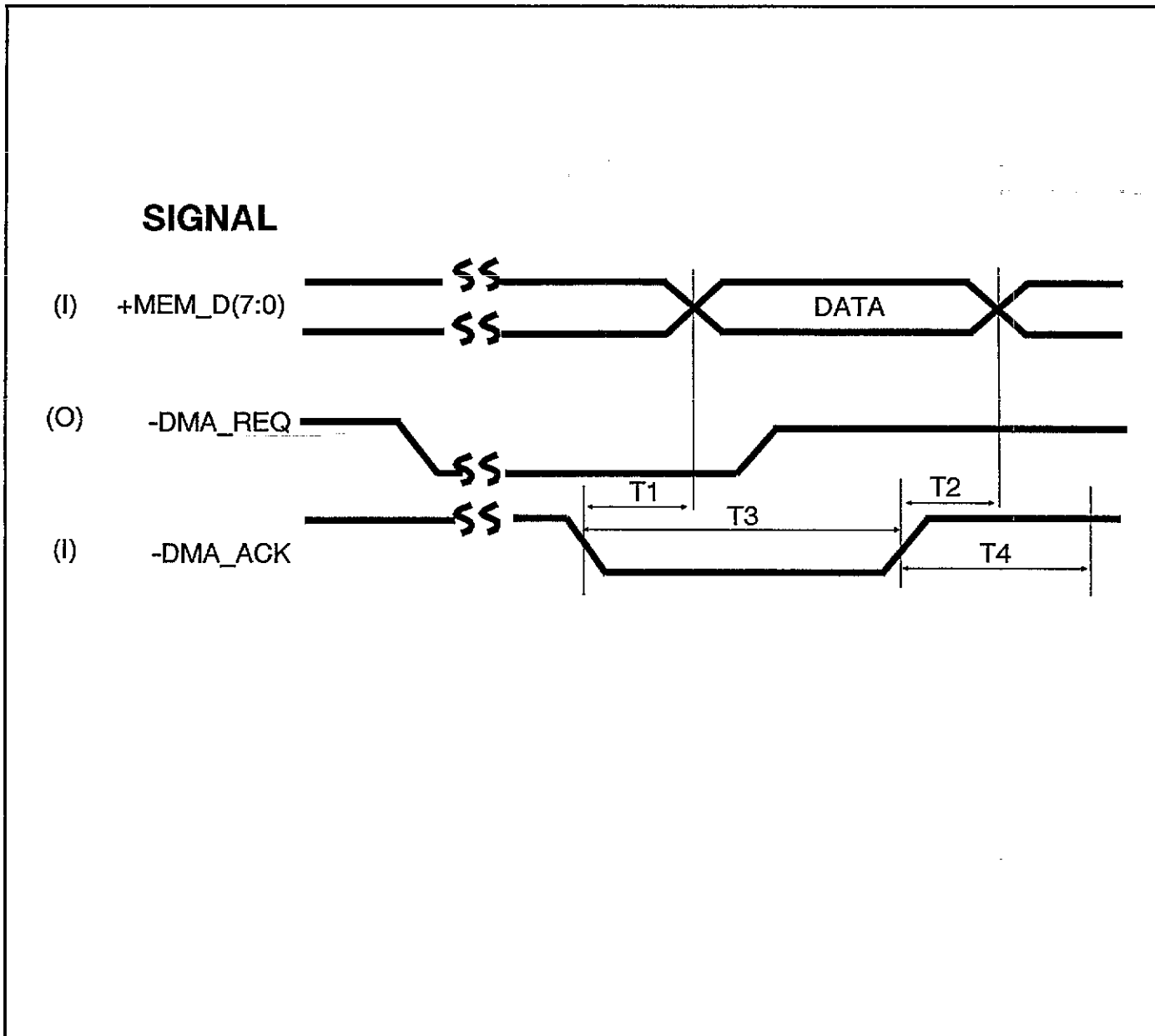


Symbol	Parameter	Min.	Typ.	Max.	Unit
T1	Data valid to -DMA_ACK high	10			ns
T2	Data hold time	10			ns
T3	-DMA_ACK low	2 CP			ns
T4	-DMA_ACK high	2 CP			ns

Note:

- Under normal operation -DMA_REQ will not be asserted for a second transfer until -DMA_ACK has been deasserted from the first transfer. -DMA_REQ can be asserted earlier if the fast DMA transfer bit (Register 7, bit 3) is asserted.

Figure 2-16. DMA Interface, Read from F82C5087 Timing Characteristics



Symbol	Parameter	Min.	Typ.	Max.	Unit
T1	Data valid to -DMA_ACK low to data valid			25	ns
T2	Data hold from -DMA_ACK high	10			ns
T3	-DMA_ACK low	2 CP			ns
T4	-DMA_ACK high	2 CP			ns

Note:

1. Under normal operation -DMA_REQ will not be asserted for a second transfer until -DMA_ACK has been deasserted from the first transfer. -DMA_REQ can be asserted earlier if the fast DMA transfer bit (Register 7, bit 3) is asserted.

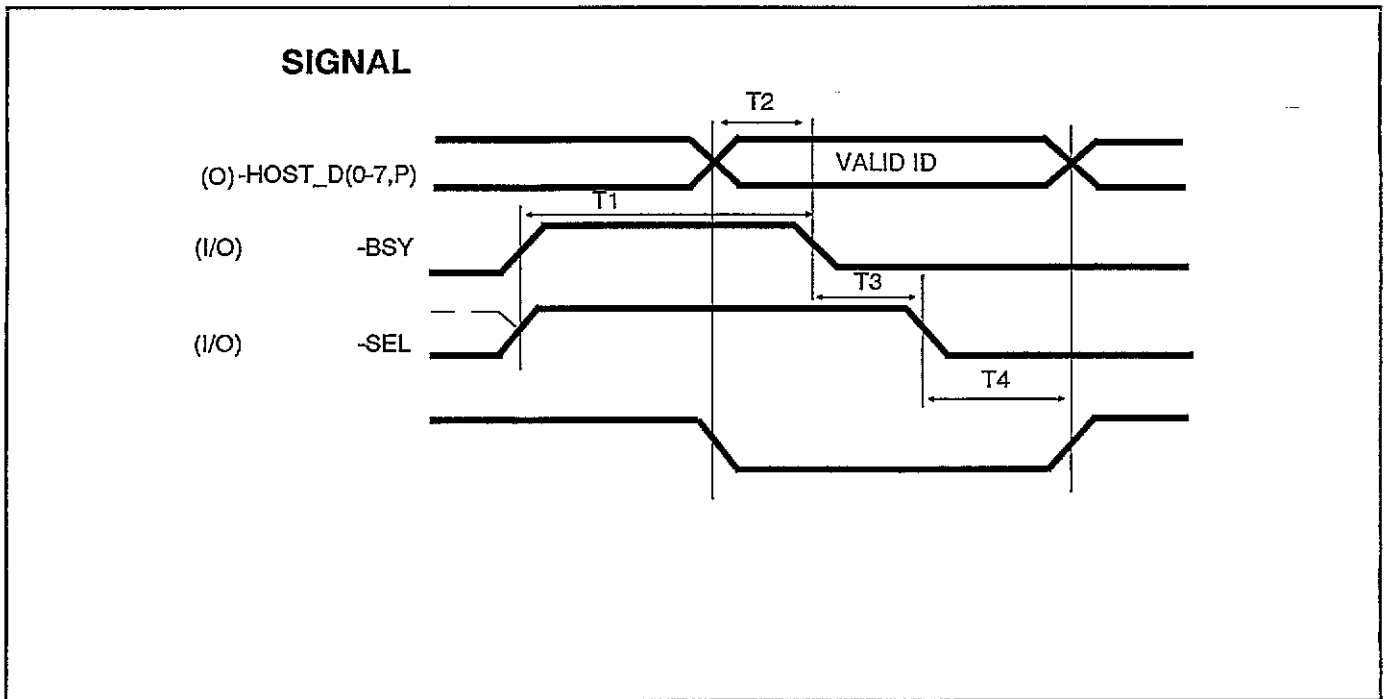
SCSI Interface Timing

Figures 2-17 through 2-26 present signal timings and their AC characteristics for the F82C5087 signals that support such SCSI interface functions as arbitration of the SCSI bus, selection/reselection of a SCSI device, and asynchronous and synchronous data transfers. Note that signals -HOST_D(0-7) are depicted in all of these figures. These eight lines comprise the 8-bit SCSI data bus. This bus is used to transfer data to/from the host computer. The -HOST_D_P signal is the SCSI data bus parity bit.

Timing for the SCSI Bus ARBITRATION Phase

Figure 2-17 presents the signal timings and their AC characteristics for the signals involved in a SCSI device's arbitration of the SCSI bus.

Figure 2-17. SCSI Arbitration Timing Characteristics



Symbol	Parameter	Min.	Typ.	Max.	Unit
T1	Bus free to -BSY low	3 SC		4 SC+50	ns
T2	Valid bus ID to -BSY low	0			ns
T3	-BSY asserted to -SEL low	6 SC			ns
T4	-Valid ID hold after -SEL low	3 SC			ns

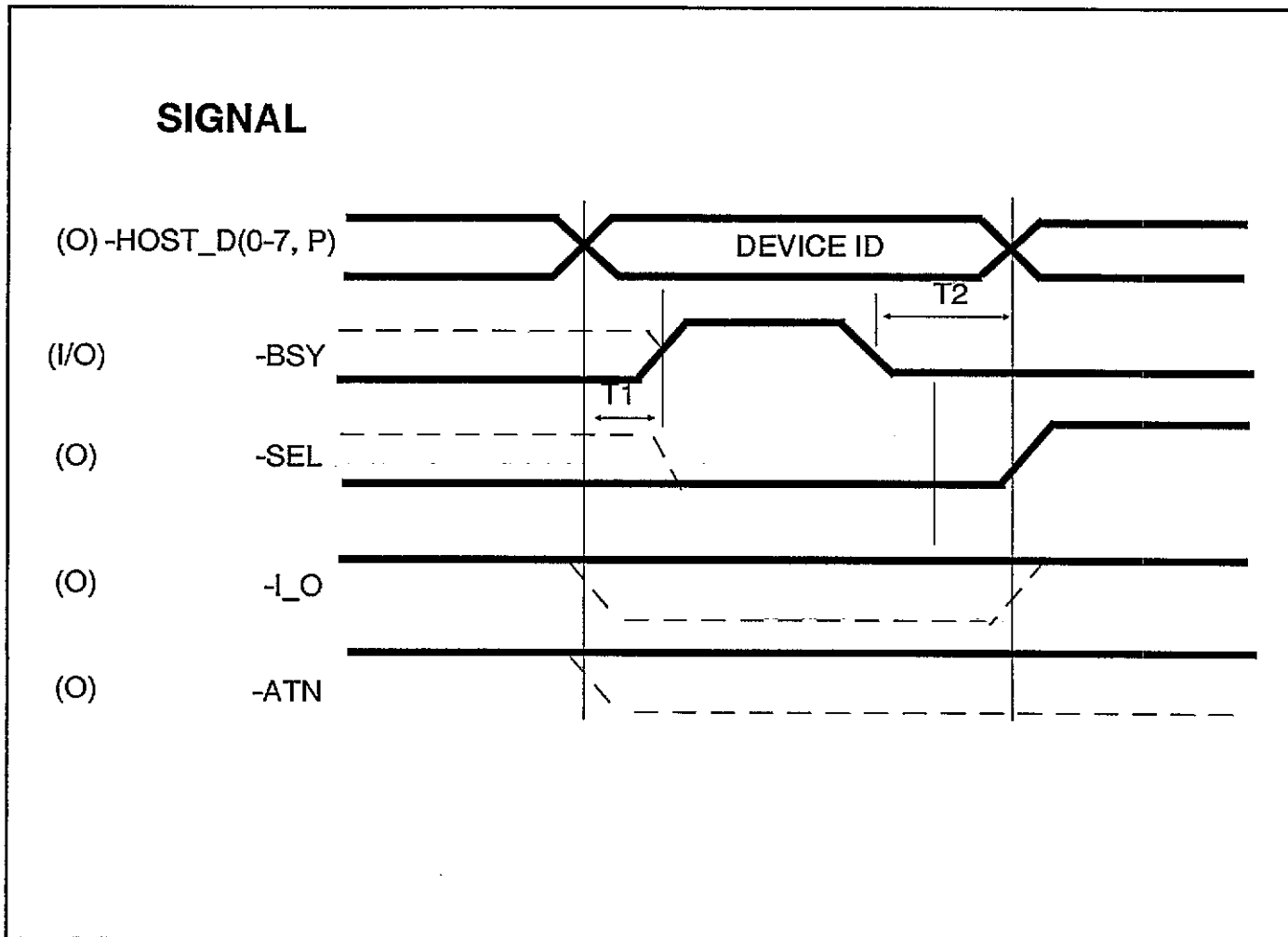
Note:

1. SC stands for SCSI Clock which is set by the value loaded into Register 8.
2. In this diagram SCSI arbitration is won, if arbitration is lost -BSY and -HOST_D(0-7, P) are deasserted within 2 SC max.

Timing for SCSI Bus SELECT/RESELECT Phases

Figure 2-18 presents the signal timings and their AC characteristics for the signals involved in the SCSI bus SELECT/RESELECT phases.

Figure 2-18. SCSI Select/Reselect Timing Characteristics



Symbol	Parameter	Min.	Typ.	Max.	Unit
T1	Valid device ID and -I_O to -BSY high or -SEL low	4 CP			ns
T2	-BSY low by device to 5087 -SEL, device ID, and -I_O hold	1 SC			ns

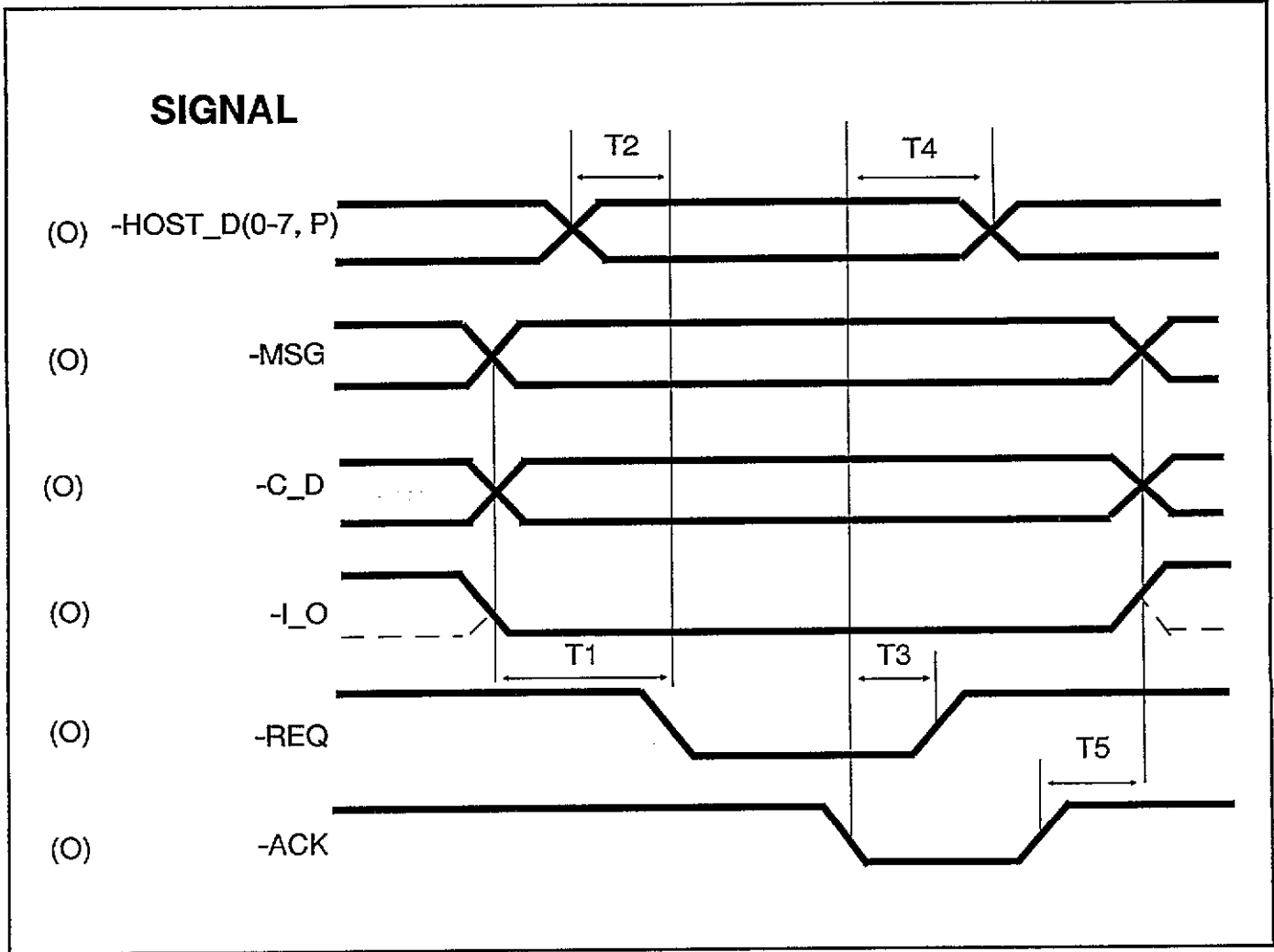
Note:

1. SC stands for SCSI Clock which is set by the value loaded into Register 8.
2. Dashed lines for -BSY and -SEL indicate non-arbitrating systems.
3. Dashed line for -I_O indicates reselection.

Timing for Asynchronous/Synchronous Transfers

SCSI bus data transfers can be asynchronous or synchronous. Figures 2-19 through 2-22 present signal timings and their AC characteristics for asynchronous data transfers. Figures 2-23 through 2-26 present signal timings and their AC characteristics for synchronous data transfers.

Figure 2-19. SCSI Target Asynchronous Transfer Output Timing Characteristics

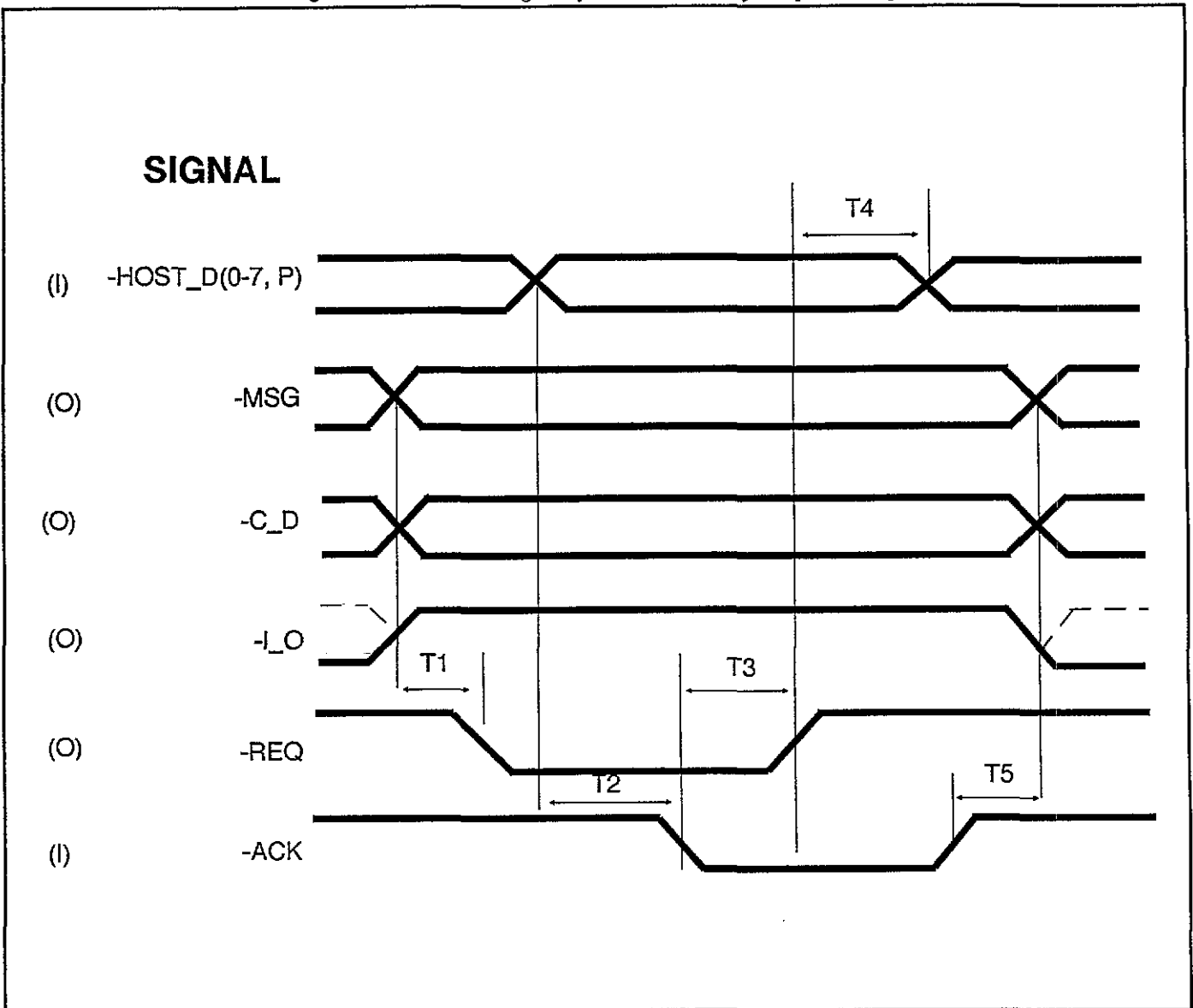


Symbol	Parameter	Min.	Typ.	Max.	Unit
T1	Phase change to -REQ low	1 SC			ns
T2	Data valid to -REQ low	2 CP-5			ns
T3	-ACK low to -REQ high	10		2 CP+10	ns
				(DATA Phase)	
T4	Data hold after -ACK high	CP			ns
T5	-ACK high to phase change	1 SC			ns

Note:

1. SC stands for SCSI Clock which is set by the value loaded into Register 8.

Figure 2-20. SCSI Target Asynchronous Transfer Input Timing Characteristics

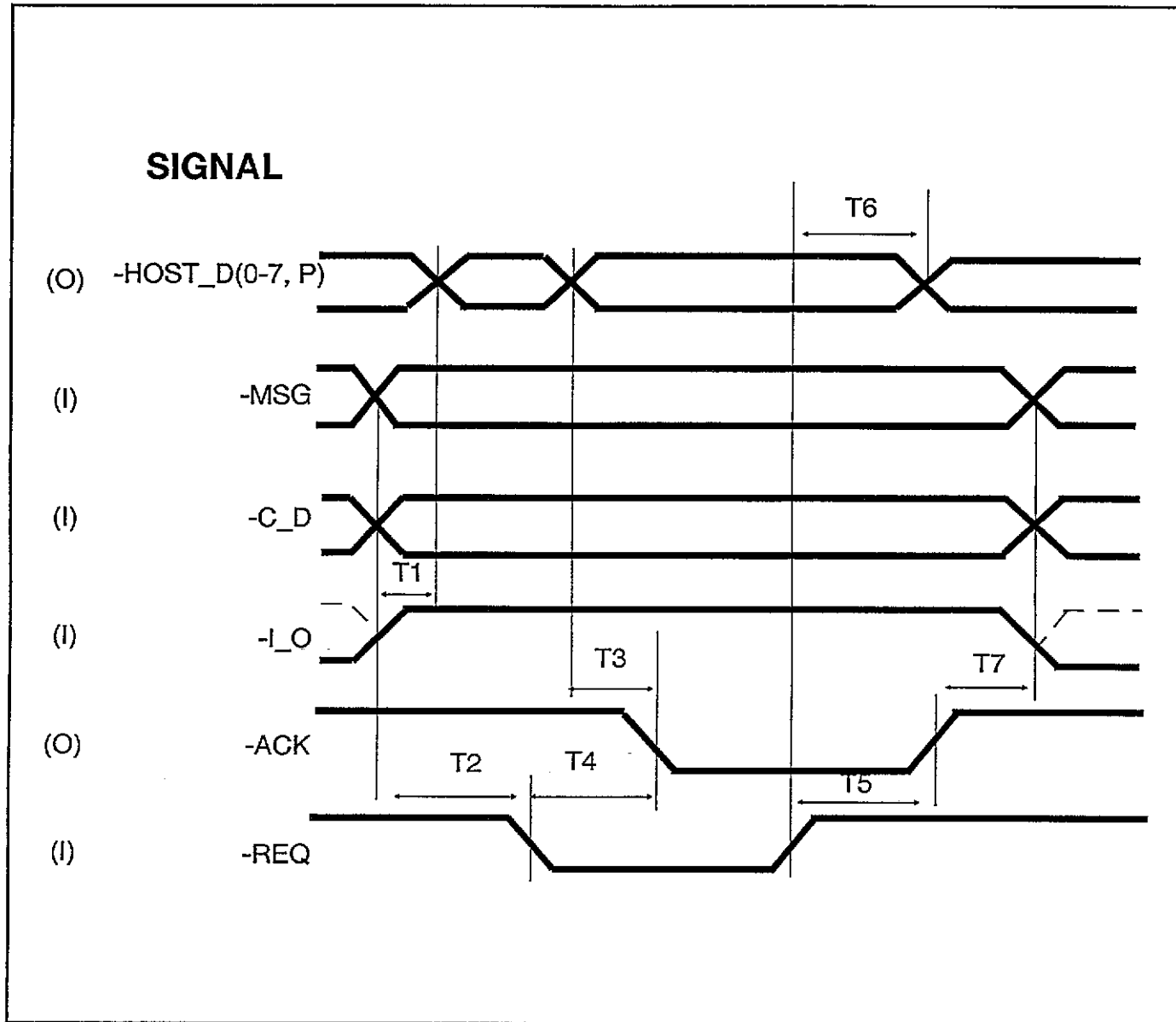


Symbol	Parameter	Min.	Typ.	Max.	Unit
T1	Phase change to -REQ low	1 SC			ns
T2	Data valid to -ACK low	0			ns
T3	-ACK low to -REQ high	10		2 CP+10	ns (DATA Phase)
T4	Data hold after -REQ high	0			ns
T5	-ACK high to phase change	1 SC			ns

Note:

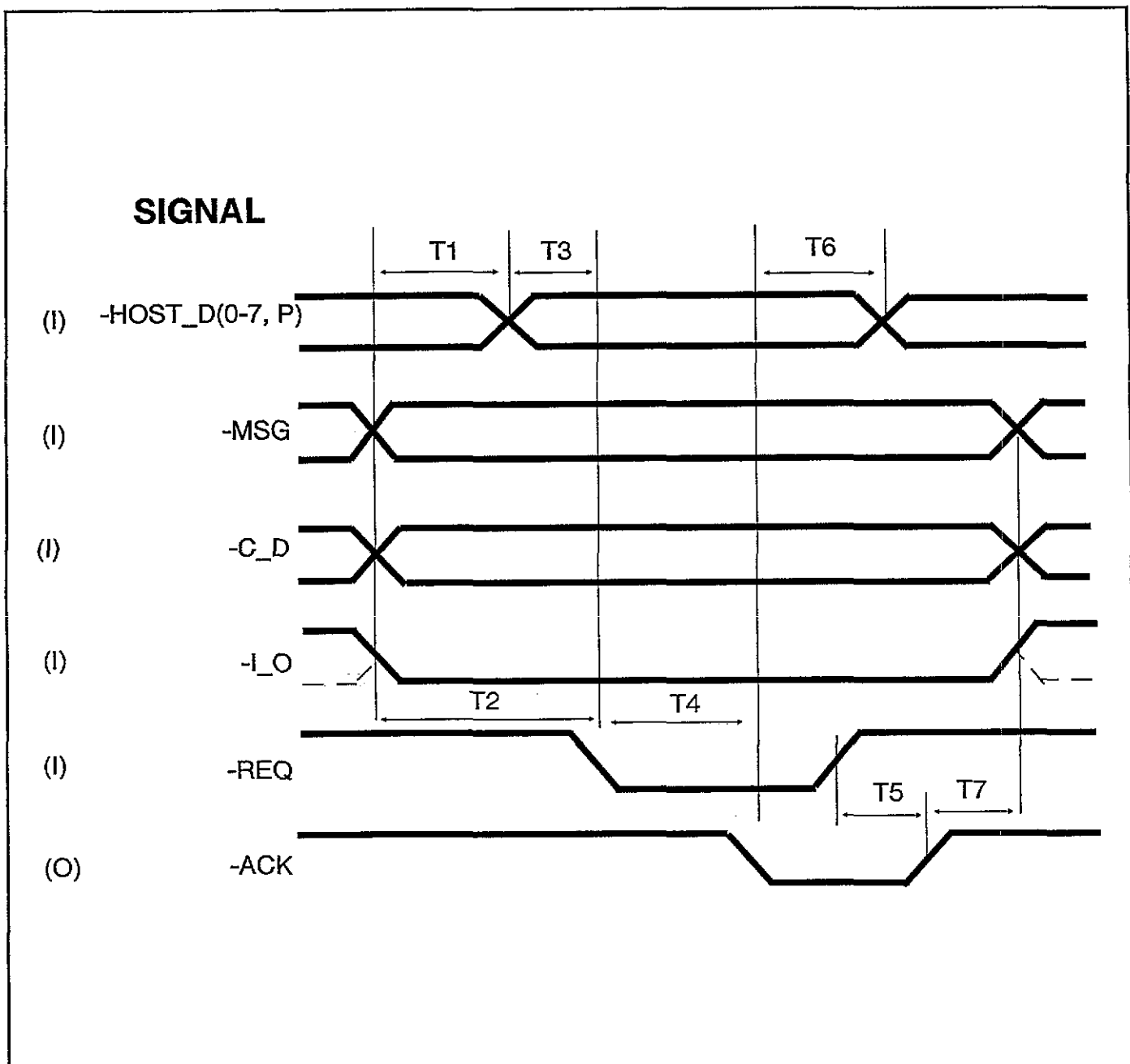
1. SC stands for SCSI Clock which is set by the value loaded into Register 8.

Figure 2-21. SCSI Initiator Asynchronous Transfer Output Timing Characteristics



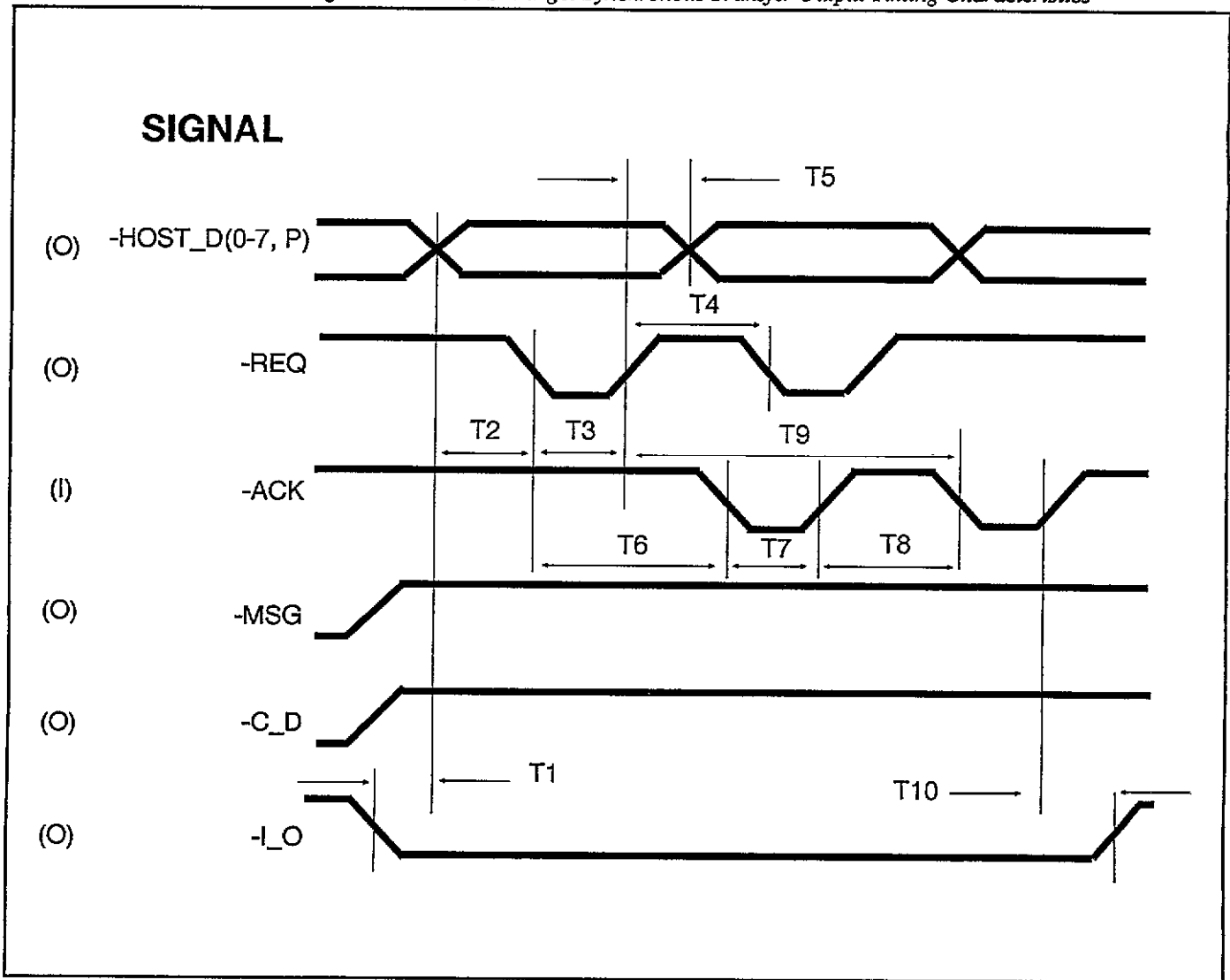
Symbol	Parameter	Min.	Typ.	Max.	Unit
T1	-I/O high to bus enable	15			ns
T2	Phase change to -REQ low	4 CP			ns
T3	Data valid to -ACK low	2 CP-5			ns
T4	-REQ low to -ACK high	10			ns
T5	-REQ high to -ACK high	10		2 CP+10	ns (DATA Phase)
T6	-REQ high to data Invalid	1 CP			ns
T7	-ACK high to phase change	2 CP			ns

Figure 2-22. SCSI Initiator Asynchronous Transfer Input Timing Characteristics



Symbol	Parameter	Min.	Typ.	Max.	Unit
T1	-I_O low to data bus disable	60			ns
T2	Phase change to -REQ low	4 CP			ns
T3	Data valid to -REQ low	10			ns
T4	-REQ low to -ACK low	10			ns
T5	-REQ high to -ACK high	10			ns
T6	-ACK low to data bus hold	0			ns
T7	-ACK high to phase change	2 CP			ns

Figure 2-23. SCSI Target Synchronous Transfer Output Timing Characteristics

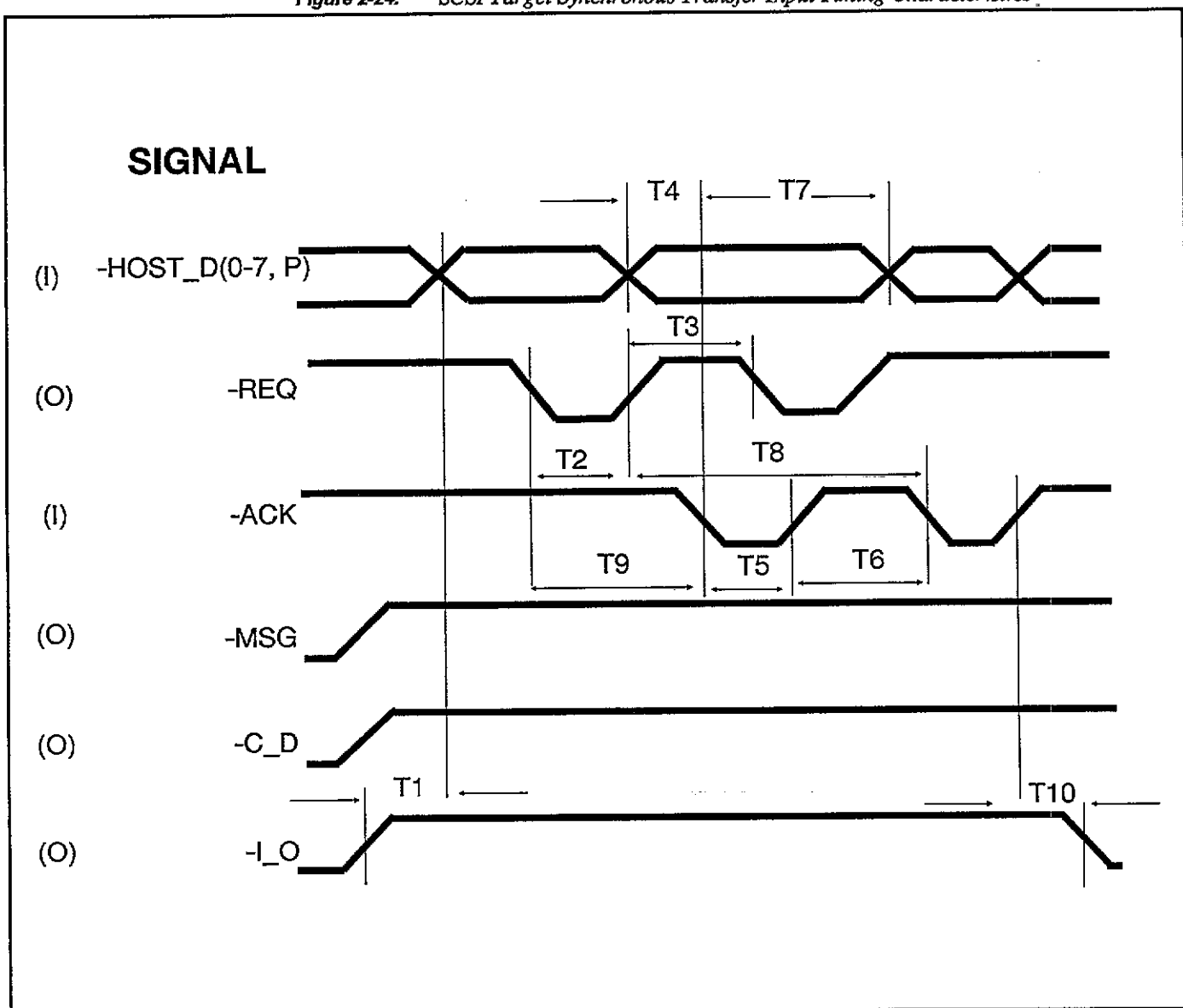


Symbol	Parameter	Min.	Typ.	Max.	Unit
T1	-I_O low to data bus enable	1 SC			ns
T2	Data bus valid to -REQ low	2 CP-5			ns
T3	-REQ assertion period	3 CP			ns
T4	-REQ deassertion period	3 CP			ns
T5	-REQ high to data hold	0			ns
T6	-REQ low to -ACK low	10			ns
T7	-ACK assertion period	2 CP			ns
T8	-ACK deassertion period	2 CP			ns
T9	-ACK period	6 CP			ns
T10	Last -ACK pulse high to phase change	4 CP			ns

Note:

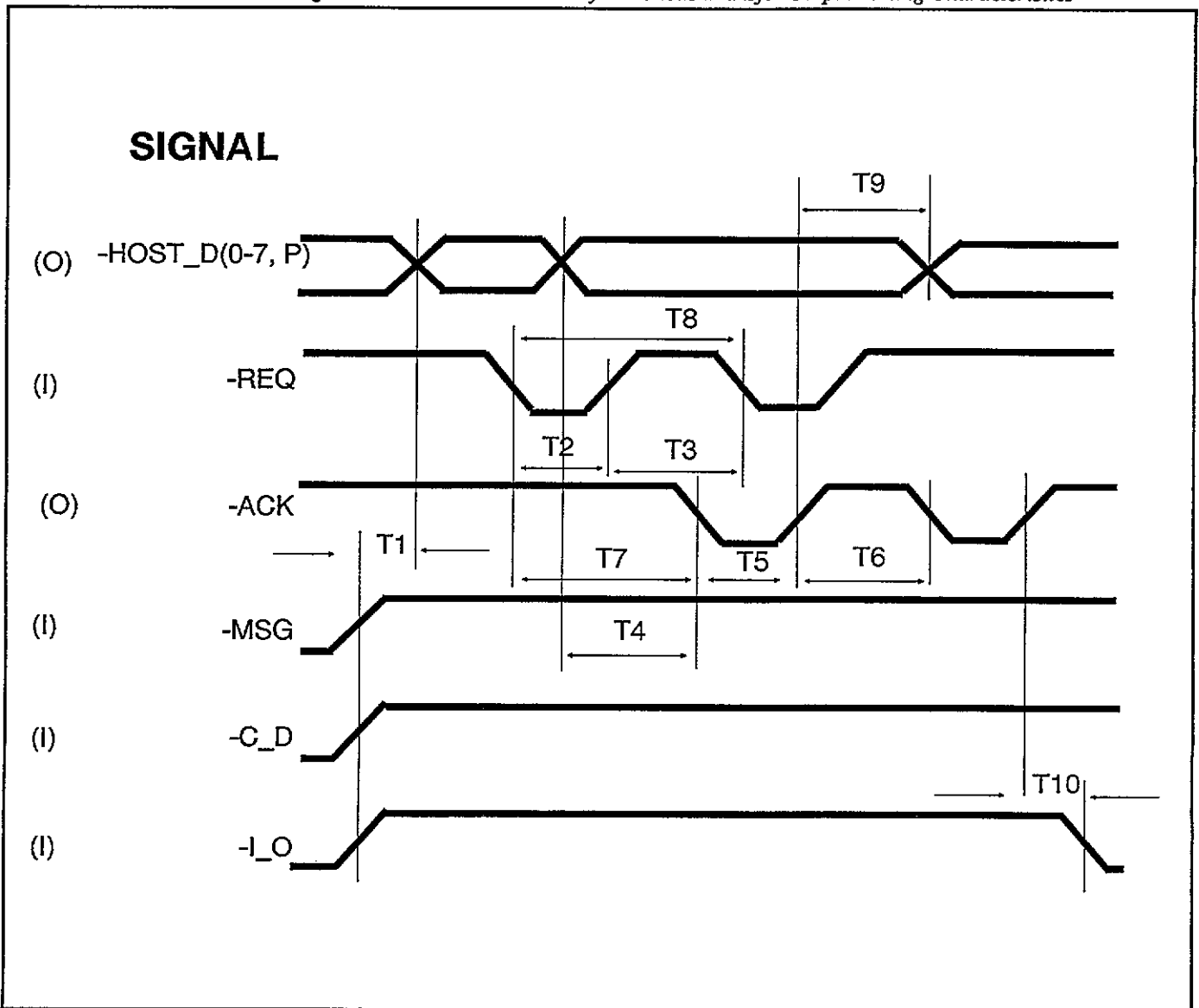
1. SC stands for SCSI clock which is set by the value loaded into Register 8.

Figure 2-24. SCSI Target Synchronous Transfer Input Timing Characteristics



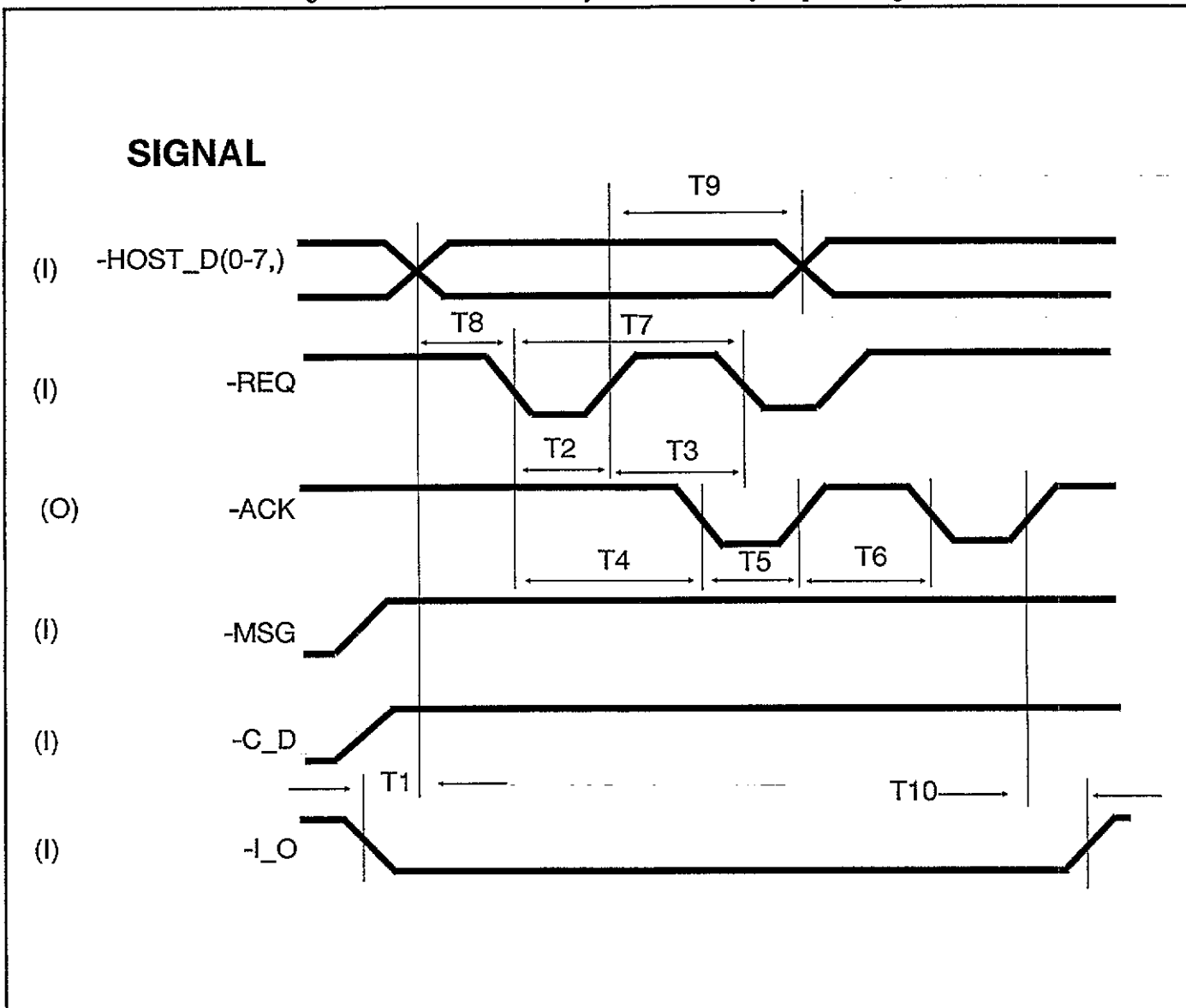
Symbol	Parameter	Min.	Typ.	Max.	Unit
T1	-I_O high to data bus disable			50	ns
T2	-REQ assertion period	3 CP			ns
T3	-REQ deassertion period	3 CP			ns
T4	Data valid to -ACK low	0			ns
T5	-ACK assertion period	2 CP			ns
T6	-ACK deassertion period	2 CP			ns
T7	-ACK low to data hold	45			ns
T8	-ACK period	6 CP			ns
T9	-REQ low to -ACK low	10			ns
T10	Last -ACK pulse high to phase change	4 CP			ns

Figure 2-25. SCSI Initiator Synchronous Transfer Output Timing Characteristics



Symbol	Parameter	Min.	Typ.	Max.	Unit
T1	-I_O high to data bus enable	10			ns
T2	-REQ pulse assertion period	2 CP			ns
T3	-REQ pulse deassertion period	2 CP			ns
T4	Data bus valid to -ACK low	2 CP-5			ns
T5	-ACK assertion period	3 CP			ns
T6	-ACK deassertion period	3 CP			ns
T7	-REQ low to -ACK low	4 CP			ns
T8	-REQ period	6 CP			ns
T9	-ACK high to data hold	0			ns
T10	Last -ACK pulse high to phase change	2 CP			ns

Figure 2-26. SCSI Initiator Synchronous Transfer Input Timing Characteristics



Symbol	Parameter	Min.	Typ.	Max.	Unit
T1	-I_O low to data bus disable			50	ns
T2	-REQ pulse assertion period	2 CP			ns
T3	-REQ pulse deassertion period	2 CP			ns
T4	-REQ low to -ACK low	4 CP			ns
T5	-ACK assertion period	3 CP			ns
T6	-ACK deassertion period	3 CP			ns
T7	-REQ period	6 CP			ns
T8	Data bus valid to -ACK low	10			ns
T9	Data hold after -REQ high	0			ns
T10	Last -ACK pulse high to phase change	4 CP			ns

D.C. Specifications

The F82C5087 is built using CMOS low-power technology. It operates from a single +5 Volt supply. Refer to the following sections for information on absolute maximum ratings and D.C. Characteristics.

Absolute Maximum Ratings

Rating	Minimum	Maximum	Unit
Power supply voltage	-0.3	7.0	VDC
Ambient operating temperature	0.0	+70.0	C degrees
Storage temperature	-65.0	+150.0	C degrees

Caution: Stresses greater than those indicated may cause permanent damage to the chip. Operation of the chip at conditions above those shown is not implied. Exposure to absolute maximum rating conditions for extended periods may affect the chip's reliability.

Standard Test Conditions

The following characteristics apply to standard test conditions unless noted otherwise in test specifications. Voltages are referenced to GND. Positive current flow is into the reference pin. Standard conditions are as follows:

- $V_{cc} = 5.0 \text{ VDC} \pm 0.25 \text{ VDC}$
- $GND = 0 \text{ VDC}$
- $0 \text{ degrees C} < T_A < 70 \text{ degrees C}$

D.C. Characteristics

Parameter	Min	Max	Unit
Input High Voltage	2	V_{cc}	V
Input Low Voltage	-0.3	0.8	V
Output High Voltage	2.4	V_{cc}	V
Output Low Voltage		0.4	V
High-level Output Current		-4	mA
Low-level Output Current		4	mA
Input Leakage	-30	10	uA
Output Leakage		10	uA
V_{cc} Supply Current		50	mA

D. C. Characteristics for Drivers/Receivers

Drivers (SCSI bus)	Sink 48 mA @ 0.5 VDC asserted
Drivers (AT bus)	Sink 24 mA @ 0.5 VDC asserted
Receivers	Asserted at input = 0 to 0.8 VDC Non-asserted at input = 2.0 to 5.25 VDC Minimum hysteresis = 0.2 VDC



Chapter 3

Operation

This chapter provides a description of the modes the F82C5087 can operate in when used as a host adapter or device controller. It also explains how the structure of the F82C5087 architecture supports these various modes.

Modes of Operation

As mentioned previously, the F82C5087 provides the functions necessary to implement the SCSI interface on a host computer or a device controller. Used in either of these applications, the F82C5087 can support the following modes of operation:

- Initiator or target mode
- Differential or non-differential mode
- Asynchronous or synchronous transfer mode
- Master or slave mode.

Initiator and Target Modes

Systems can be configured with single or multiple initiators, and with single or multiple targets. Some SCSI devices can act as an initiator in one SCSI interface transaction, and function as a target in another. The F82C5087 supports both the initiator and target mode.

Role of Initiator

Typically, the initiator is the host adapter. The initiator is responsible for originating the operation that the target will perform. In order to initiate a SCSI bus operation, the initiator must complete the following steps:

- Wait for the SCSI bus to go to the BUS FREE phase.
- Arbitrate for control of the SCSI bus.
- Select the device (target) using the target ID placed in the MCS.
- Optionally send an identify message from the MCS to the target.
- Optionally send a two-byte tag message from the MCS to the target.
- Send the target a multi-byte command from the MCS, defining the activity which is to occur. Refer to the section "Pipelining Structure" later in this chapter for more details on this topic.

The initiator controls the SCSI bus during the ARBITRATION and SELECTION phases. However, once the target is selected, control transfers to the target.

Role of the Target

If the F82C5087 is selected as a target it will perform the following sequence under the control of the firmware:

- Save the ID of the initiator in the MCS.
- Optionally receive the identify message and save it in the MCS.
- Optionally receive the two-byte tag message, and save it in the MCS.
- Receive a multi-byte command, and save it in the MCS.
- If the identify message allows the target to disconnect, then send the DISCONNECT message to the initiator. If the identify message disallows disconnect, the sequence is terminated at this point.
- Go to the SCSI bus BUS FREE phase.

Once the target is selected by the initiator, the target controls the SCSI bus. Targets can also arbitrate for the SCSI bus and reselect an initiator. Refer to the target commands in Chapter 5 for more details.

Transfer Modes

The F82C5087 supports only the single-ended SCSI mode with built-in 48 mA drivers. When the F82C5087 is transferring data over the SCSI bus, the mode of transfer can be asynchronous or synchronous. An asynchronous transfer rate of up to 4 megabytes/second, and synchronous transfers of up to 5.3 megabytes/second are supported (33 MHz clock).

Asynchronous Transfers

Asynchronous transfers use the protocol of the REQ/ACK handshake. With each handshake, one byte of information can be transferred over the SCSI bus. Targets drive the -REQ signal while initiators drive the -ACK signal. In this mode, targets control the directional flow by driving the -I_O signal. When this signal is asserted, the information is transferred from the target to the initiator over the SCSI bus. When the -I_O signal is deasserted, the information is transferred from the initiator to the target.

In this mode, the F82C5087's 64-byte FIFO is used as a buffer between the SCSI bus, and the high-speed memory data bus or MPU bus. Also 5087s can be cascaded to increase the width of SCSI bus and the memory data bus from 8 bits to up to 32 bits. Refer to the section "Master/Slave Mode" later in this chapter for more information on cascading 5087s.

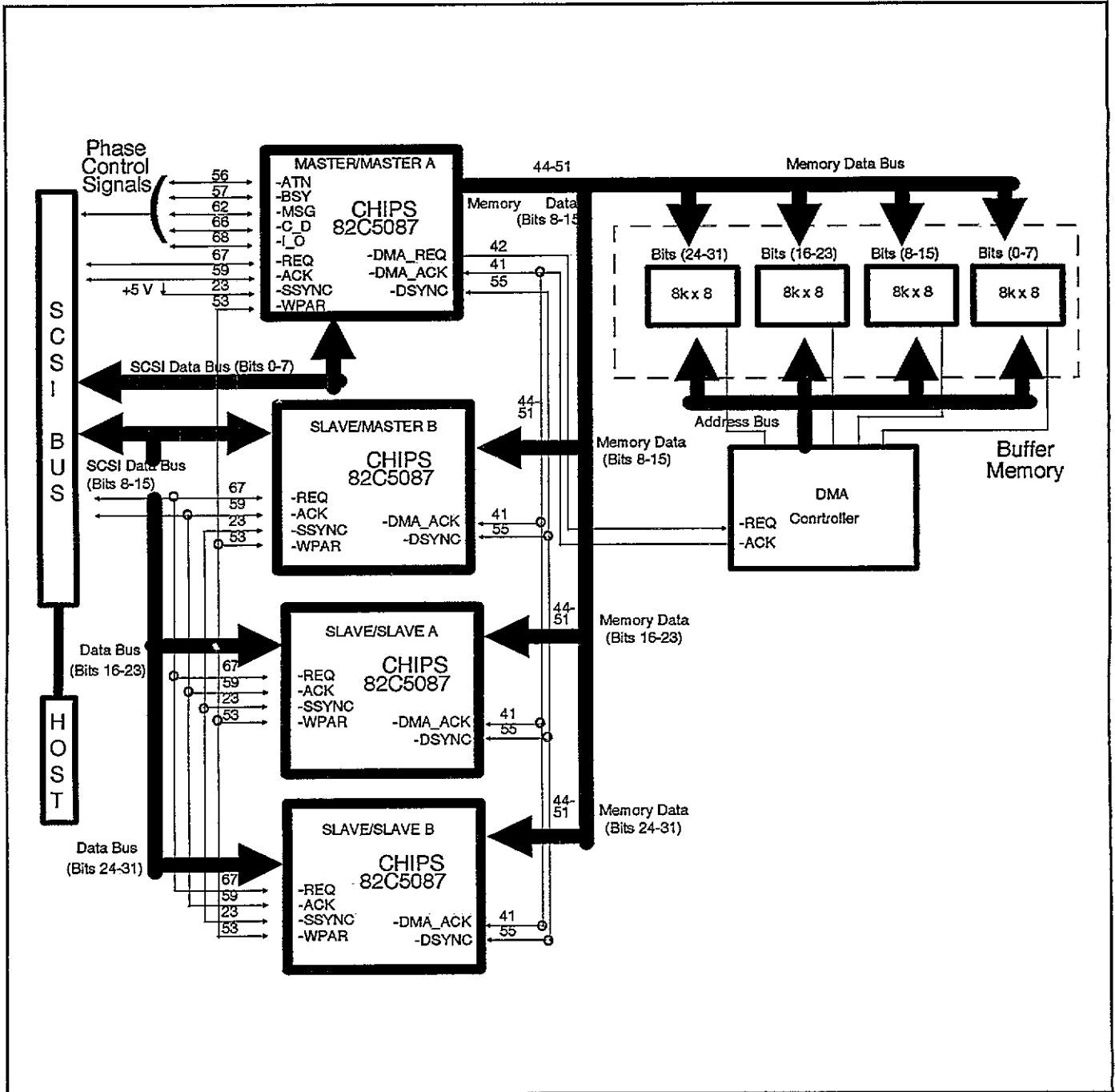
Synchronous Transfers

Like asynchronous transfers, synchronous transfers use a REQ/ACK handshake. Unlike asynchronous transfers, a transfer rate can be specified with synchronous

transfers. This rate can be specified by performing a write operation to Transfer Period Register 15. For targets responsible for driving the -REQ signal, the transfer rate specifies the minimum number of clock cycles between the leading edges of the -REQ signal to the next -REQ pulse. For initiators responsible for driving the -ACK signal, the transfer rate specifies the minimum number of clock cycles between the leading edges of -ACK pulses.

With synchronous transfers, the -REQ and -ACK signals can be offset by a count of up to 63 bytes; a count up to the size of the F82C5087's internal FIFO. This offset can be specified through a write operation to Maximum Sync Offset Register 14. For more information on these registers refer to Chapter 4.

Figure 3-1. Functional Pin out of Cascaded 5087s



Master/Slave Mode

As Figure 3-1 indicates, 5087s can be cascaded to support wide data transfers over the SCSI and the memory data buses. By cascading four F82C5087 chips these buses can be extended to 32 bits.

The following may be specified for each F82C5087 in the system:

- Master or slave mode for the F82C5087(s) basic mode of operation (e.g., transferring data to/from the DMA controller's buffer memory), and
- Master or slave mode for the F82C5087(s) SCSI interface functions.

The mode the chip is operating in to perform SCSI bus activities can be viewed as a subordinate mode of the chip's main mode of operation. For instance, as Figure 3-1 indicates the chip labelled Slave/Master B is operating in slave mode; it is a slave to the F82C5087 labelled Master/Master A. However, when Slave/Master B is performing SCSI interface activities, it is operating in master mode. Therefore for such operations, it will operate as the master chip for its two slave chips, Slave/Slave A and B.

The F82C5087's basic mode of operation as master or slave can be specified with bit 7 (Slave Mode Bit) of Mode Control Register 7. The F82C5087's subordinate mode, the SCSI bus interface mode, can be specified as master or slave with bit 6 (SCSI Bus Slave Mode) of this register.

When bit 7 is asserted, the F82C5087 is operating in slave mode as its basic mode of operation; when this bit is deasserted the F82C5087 is operating in master mode. When bit 6 is asserted, it forces the chip to operate in slave mode when performing SCSI bus interface functions; when this bit is deasserted, the chip will operate in master mode. Bit 6 should be asserted only if bit 7 is also asserted. Consequently, bits 7 and 6 must be set as shown in Table 3-1 for the system configuration that is shown in Figure 3-1:

Table 3-1.

82C5087 Chip	Mode Control Register 7	
	Bit 7 Slave Mode	Bit 6 SCSI Bus Slave
Master/Master A	0	0
Slave/Master B	1	0
Slave/Slave A	1	1
Slave/Slave B	1	1

If bits 7 and 6 are set as indicated in Table 3-1, the F82C5087 chips are configured as shown in Table 3-2.

Table 3-2.

	Basic Mode	SCSI Bus Mode
Master/Master A	Master Mode	Master Mode
Slave/Master B	Slave Mode	Master Mode
Slave/Slave A	Slave Mode	Slave Mode
Slave/Slave B	Slave Mode	Slave Mode

Master/Master Mode

When bits 7 and 6 of Register 7 are deasserted, the F82C5087 chip is in master mode when performing its basic operations (e.g., transferring data to buffer memory), as well as in master mode when performing SCSI bus functions. Refer to the chip labelled Master/Master A in Figure 3-1. The master chip is responsible for controlling the SCSI bus phase control signals and for transferring bits 0-7 during SCSI bus transfers. It is connected to the host system via a cable referred to as Cable A. For more information on this cable refer to the *American National Standard for Information Systems-Small Computer System Interface*.

Slave/Master Mode

When bit 7 is asserted and bit 6 is deasserted, the chip is forced to operate in slave mode for its basic operation, yet operate in master mode for SCSI bus interface functions. Refer to the chip labelled Slave/Master B in Figure 3-1. This chip is functioning as a slave to Master/Master A. It does not control such SCSI bus phase signals as -ATN, -BSY, -C_D; Master/Master A is responsible for this. It functions as a slave in such activities as transferring data to the system's buffer memory.

Yet Slave/Master B acts in master mode when performing SCSI bus data transfers. Therefore, this chip must synchronize the transfer of bits 16-32 by Slave/Slave A and Slave/Slave B as well as transfer bits 8-15. Slave/Master B, Slave/Slave A, and Slave/Slave B are connected to the host system via a cable referred to as Cable B. Whereas, the master chip's Cable A carries SCSI bus phase control signals as well as SCSI data (bits 0-7), Cable B carries only the -REQ and -ACK signals and SCSI data (bits 8-32). Cable A and B are independent cables. For more information on these cables refer to the above-mentioned ANSI document.

Slave/Slave Mode

When bits 7 and 6 are asserted, the chip is forced to operate in slave mode for its basic operation as well as in slave mode for its SCSI bus interface operations. Refer to the chips labelled Slave/Slave A and B in Figure 3-1.

In slave mode, the F82C5087 is under the control of the master 82C5087 chip. Only the FIFO is active in slave mode; SCSI bus phases or MCS access is not used. Asynchronous and synchronous data transfers proceed as in master mode.

Synchronizing Memory Data Bus Transfers

Only the master (Master/Master A) can send request signals -DMA_REQ to the DMA controller. When the slave chips have data ready to transfer over the memory data bus, they all release -DSYNC (DMA Sync); forcing this signal to go high. When the slave(s) -DSYNC signal is high, the master asserts its -DMA_REQ signal to indicate it has data ready to transfer to the system's buffer memory. When the DMA controller is ready to receive that data, it sends an -ACK signal back. The slaves' -DMA_ACK pins should be tied to the master's

-DMA_ACK pin; thereby enabling the slaves to detect the DMA's -ACK signal. In slave mode, the slave F82C5087(s) drives the -WPAR as well as the -DSYNC signals.

Synchronizing SCSI Bus Data Transfers

A chip operating as a SCSI bus master (Master/Master A or Slave/Master B) conducts a request/acknowledge handshake with the SCSI bus for transferring data. A SCSI bus master chip will not handshake a byte on the SCSI bus until the wire-or'ed -SSYNC signal is deasserted (high). SCSI bus slave chips will assert this signal when they are not ready to process more data.

In slave SCSI mode, the F82C5087 chip is under the control of the master SCSI chip. Only the FIFO is active in slave mode; SCSI bus phases or MCS access is not used. SCSI bus data transfers can be asynchronous or synchronous. In asynchronous or synchronous mode, the -WPAR pin indicates the F82C5087 slave chip(s) has a data parity error or an overrun/underrun error condition.

To ensure correct synchronization, the firmware must send pipeline command(s) to the slave F82C5087(s) before sending the identical command(s) to the master F82C5087.

Hardware Specifications

The F82C5087 pins that support the buffer memory interface should be wired as follows: The master chip's -DMA_REQ and -DMA_ACK pins need to be tied to the DMA controller's REQ/ACK pins. The -DMA_ACK pin on the slave(s) should be tied to the master's -DMA_ACK pin. The -DSYNC pin on the slave(s) should be tied to the master's -DSYNC pin.

The F82C5087 pins that support the SCSI bus interface should be wired as follows: The SCSI master chip's -ATN, -BSY, -MSG -C_D, and -I_O, -REQ, and -ACK pins should be tied to the SCSI bus. On the master, the -SSYNC and -WPAR should be externally pulled high, or tied to the respective pins on the slave F82C5087(s). The SCSI bus master chip's -REQ and -ACK pins should be tied to their respective pin on the SCSI bus slave(s). Refer to Figure 3-1.

Method of Operation

Operational modes (e.g., master/slave, synchronous/asynchronous data transfer mode) are supported by functional blocks within the F82C5087. These blocks provide the logic the F82C5087 needs to interface with other system components (e.g., MPU), and to manage the flow of SCSI data, commands, messages, and status through the system. These functional blocks can be grouped into the following categories:

- The I/O interface
- The set of 46 commands, and
- The memory structure that supports the queuing of commands.

It is through its I/O interface that the F82C5087 is able to send/receive signals, addresses, and data to/from the system's other components. Through its command set, the F82C5087 is able to provide the SCSI protocol for command, data, message and status flow through the system. *The F82C5087's memory*

structure consists of a 64-byte FIFO and six, 32-byte Message/Command Space (MCS) scripts. The FIFO enables data transfers to be streamlined, and system performance to be increased. The MCS, in conjunction with the F82C5087's four pipeline registers, enables up to four pipeline commands to be queued.

In the following sections, the first two components, the I/O interface and command set, are briefly discussed. Both of these topics are described in detail in Chapters 2 and 4, respectively. The third component, the F82C5087's memory structure, is described in detail in this section.

I/O Interface

This section provides four examples of how this I/O interface enables the F82C5087 to communicate with other system components.

Example 1. I/O Interface Providing a Path to the SCSI Bus.

There are nine F82C5087 signals (e.g., -ATN) that comprise the SCSI control bus. These bidirectional signals are controlled through the F82C5087's command set. For example, by issuing the CNTL ATN ON pipeline command, the firmware can assert the -ATN signal on the SCSI bus.

Example 2. I/O Interface to Buffer Memory.

The system's buffer memory (memory data bus) is accessible via a DMA controller. Such DMA controllers as the CHIPS 5055 DMA Controller provide the addresses in the buffer memory to which data is to be transferred. The F82C5087 communicates with the DMA controller over a bidirectional memory data bus via DMA request/acknowledge handshake control signals.

Example 3. I/O Interface to the MPU.

The A_D bus and its associated control signals connect the F82C5087 to the MPU. Each of the 24, internal F82C5087 registers has its own address. Through address bits A4-A0 (non-multiplexed addressing) the MPU can specify an address and select any of the F82C5087's internal registers for a read/write operation. For example, by performing a write operation to MPU FIFO Data Register 12, the MPU can deposit data into the F82C5087's FIFO. From the FIFO, the data can be transferred to the buffer memory via the memory data bus or to the SCSI bus.

Example 4. I/O Interface to the Host Computer.

Communication with the host computer is via an 8- or 16-bit bidirectional port. The host computer interface consists of the A_D Bus (bits 0-7), the memory data bus (bits 8-15), and the MPU control signals. Data is transferred to/from the host computer via the SCSI data bus. This bus is comprised of the eight lines named HOST_D[0-7].

Certain signals are used only when the F82C5087 is operating in a specific mode. For instance, the MPU_AEN signal is used only when the F82C5087 is operating in the host adapter mode. Refer to Chapter 2 for signal descriptions.

Command Set

The commands can control the F82C5087's I/O signals, and thus control the F82C5087's communication with the other system components. These commands provide the functions needed to implement the SCSI interface on a host computer or a device controller. Through this set of 46 commands data can be transferred between the MPU and memory data bus. SCSI data, command, message, and status flow can also be managed through the system. Refer to Chapter 5 for command descriptions.

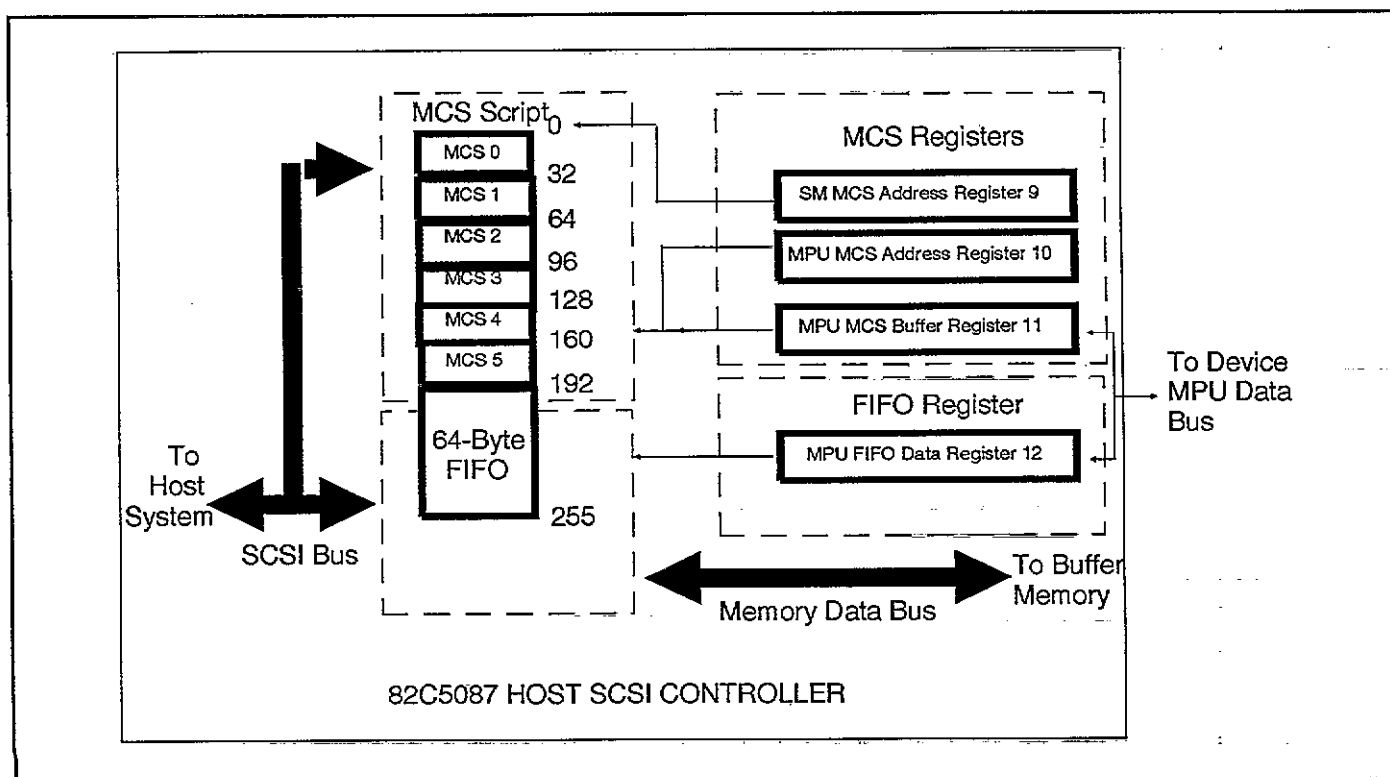
Memory Structure

The F82C5087 provides 256 bytes x 9 of memory. This on-chip memory can be categorized as:

- Data FIFO or,
- Message/Command Space (MCS) script.

The FIFO can be used to smooth out data transfers and increase system throughput. As Figure 3-2 indicates, all data transfers in the system must go through the F82C5087's internal FIFO.

Figure 3-2. Conceptual Diagram of the F82C5087's Memory



The second component of on-chip memory is the MCS. This portion of memory is used by the firmware to store SCSI sequences comprised of SCSI commands, messages, and/or status. These SCSI sequences are used by the pipeline commands that are placed in the F82C5087's pipeline registers. Together, the MCS and the pipeline registers enable the firmware engineer to queue up to four commands, and to use the MCS as an extension of the MPU's memory. Plus, the MCS can be customized for a particular application. Refer to the section "Pipelining Structure" later in this chapter for more details.

Requests to access F82C5087 memory are prioritized in the following descending order: FIFO IN, FIFO OUT, F82C5087 state machine (SM) requests, and MPU requests.

Data FIFO

All data transfers in the system must use the F82C5087's internal FIFO. This 64-byte section of memory is used as a depository for FIFO data and parity. The pipeline commands are used to manage the flow of this SCSI data. This includes asynchronous, as well as synchronous, SCSI bus data transfers.

In addition to SCSI bus data transfers, the FIFO is also used as a depository for data being transferred between the MPU's data bus and the memory data bus. The high-speed memory data bus provides increased system performance because control and data activities are independent. Joining these two buses provides the firmware with a means of accessing the data stored in the buffer memory. For example, if the bad track information from a disk drive is read, that information may have to be transferred from the buffer memory to the MPU memory. Note that for the MPU to transfer data to/from the F82C5087's FIFO, the MPU must perform a write/read to MPU FIFO Data Register 12. Just as MPU MCS Buffer Register 11 is a holding register for MPU transfers to/from the MCS, the MPU FIFO Data Register is a holding register for MPU transfers of data to/from the FIFO.

Bytes are transferred into and out of the FIFO only when the transfer counter is non-zero. The transfer counter can be loaded through Registers 16-18. For diagnostic purposes, Register 13 may be used to preset the FIFO count to a value between 0-63. During normal operation, Register 13 can be read to determine the number of bytes in the FIFO. Write to Register 14 to set an offset of up to 63 bytes for synchronous data transfers. Register 14 can also be used to preset the current offset count. Normally, before the F82C5087 executes any data transfer commands it resets the FIFO. For more details on any of these registers refer to Chapter 4.

MCS Script

Just as the F82C5087's internal FIFO is used as a depository for data that will be routed through the system, the six, 32-byte MCS scripts can be used to store SCSI commands, messages, and status. The flow of the SCSI sequence(s) stored in the MCS is managed by issuing pipeline commands to the F82C5087's pipeline registers. For more information on this topic refer to the section "Pipelining Structure" later in this chapter.

The MCS can be used as a tool to customize the F82C5087 for their particular applications. For instance, if the F82C5087 is configured as a host adapter operating in initiator mode, certain messages frequently sent by the initiator can be stored in the MCS.

MCS Address Pointers

The F82C5087 has two registers that serve as pointers to the MCS. One register, MPU MCS Address Register 10, acts as the MPU's pointer to the MCS. The MPU uses this pointer as the starting address for sending/receiving SCSI commands, status and/or messages to/from the MCS. Every time the MPU processes (reads or writes) a byte from the MCS script, the MPU MCS address pointer (Register 10) is incremented.

The second MCS pointer register, SM MCS Address Register 9, contains a pointer to the MCS for the F82C5087 state machine's use. Because there are two F82C5087 registers that can access the MCS, it is possible for the F82C5087 state machine to be accessing one 32-byte MCS script while the MPU is processing another area in the MCS.

MCS Auto Advance

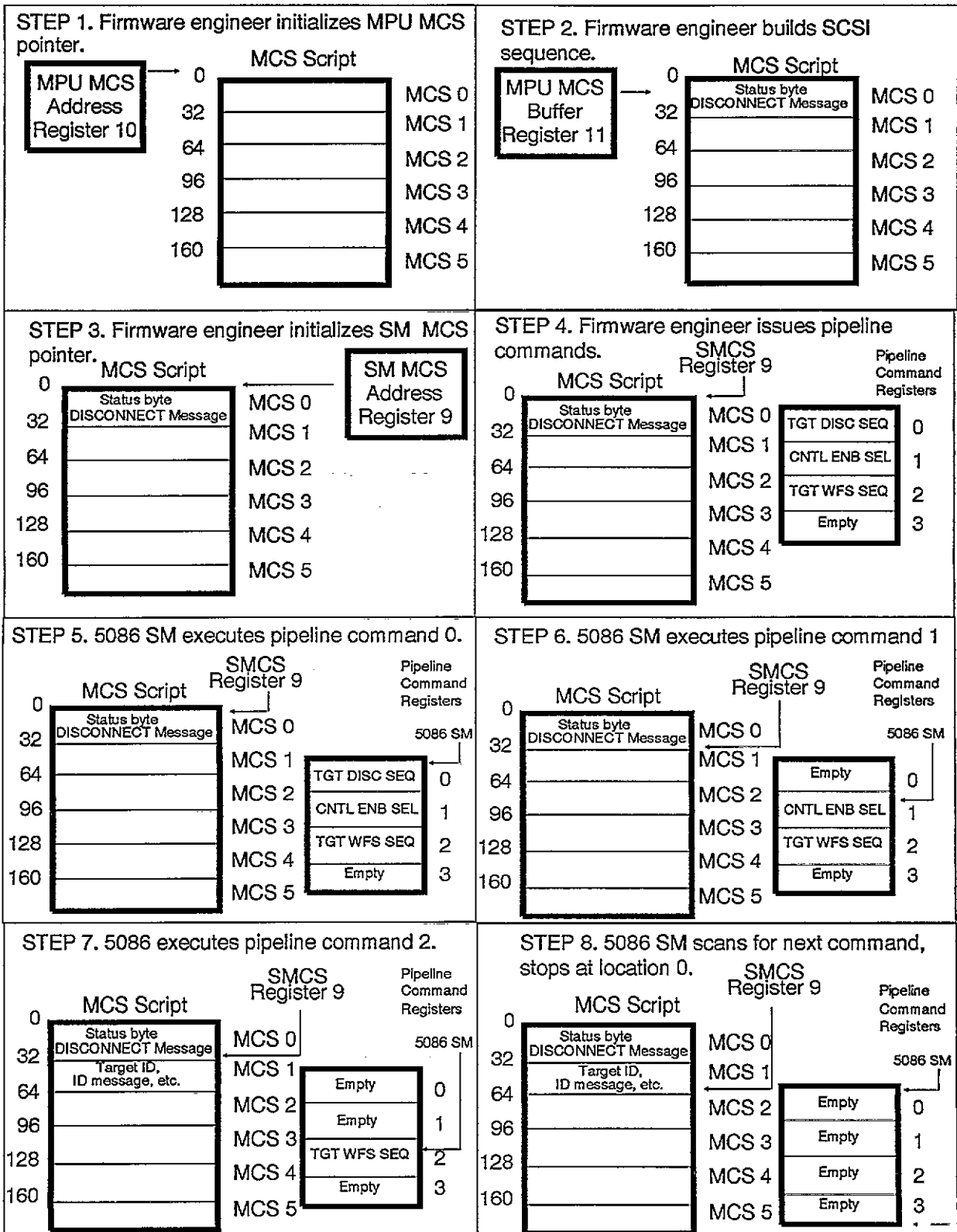
The MCS has an auto advance feature. Before starting to execute a pipeline command that accesses the MCS, the F82C5087 SM will automatically advance the SM MCS address pointer (Register 9) to the beginning of the next MCS. This feature enables the F82C5087 to execute a variable length SCSI read sequence followed by a SCSI write sequence.

This auto advance will not occur if the SM MCS address pointer is already pointing to the beginning of an MCS, or if this feature has been disabled. This can be disabled by issuing the CNTL DIS ADV command to one of the pipeline registers. If this feature is disabled, all F82C5087 state machine accesses to the MCS will be to contiguous memory locations.

Pipelining Structure

As mentioned previously, the F82C5087 state machine uses the MCS in conjunction with the four pipeline registers to enable the firmware to queue up to four commands. The pipeline registers are used as a circular queue. When a command is issued to the pipeline register, it establishes an implied relationship between the position in the MCS to which the SM MCS Address Register is pointing, and the pipeline commands.

Figure 3-3. F82C5087's Command Queuing



Before issuing a pipeline command, that will require access to the MCS, ensure that the SCSI sequence (e.g., SCSI commands, message, status) has already been stored in the MCS. The firmware is responsible for building this information in the MCS by performing consecutive writes to MPU MCS Buffer Register 11. After the firmware has built the SCSI sequence in the MCS, it must point the SM MCS address pointer (Register 9) to the first byte of this sequence.

Once the MCS has been initialized, pipeline commands to one of the pipeline registers can then be issued. The F82C5087 SM scans these four pipeline registers searching for commands to execute. When the F82C5087 SM locates the pipeline command, it executes the command and processes the sequence stored in the MCS. If the F82C5087 detects that all the pipeline registers are empty, it will stop scanning for commands at Register 0. Therefore, it is recommended that the firmware place the first pipeline command in Register 0.

After the data transfer is completed, the target needs to set up the MCS to send a status byte to the initiator and then a DISCONNECT message. Refer to the following two examples of how the pipeline registers and the MCS can be used to queue commands. Figure 3-3 illustrates the steps involved in the first example.

Example 1. SCSI Transfer Command

Before the MPU can send the SCSI transfer command, it must perform the following three steps as illustrated in Figure 3-3:

Step 1. Initialize the MPU MCS pointer.

Set up MPU MCS Address Register 10 to point to the beginning of an MCS script. In this example, assume it uses MCS 0.

Step 2. Build SCSI sequence.

The firmware must perform consecutive write operations to MPU MCS Buffer Register 11 to build the SCSI sequence, in this case a status byte followed by a DISCONNECT message. It starts at location 0 of MCS 0.

Step 3. Initialize SM MCS pointer.

Load SM MCS Address Register 9 with the address in the MCS that contains the first byte of the stored SCSI sequence. In this case, that is location 0.

Once the MCS has been initialized as described above, issue the pipeline command and the F82C5087 SM will attempt to execute the queued command(s). Refer to Steps 4-7 of Figure 3-3.

Step 4. Issue pipeline commands.

Queue a command for execution by performing a write operation to one of the four pipeline registers. In this example, three commands are queued: TGT DISC SEQ (Target Disconnect Sequence) to Register 0, CNTL ENB SEL (Control Enable Select) to Register 1, and TGT WFS SEQ (Target Wait for Select Sequence) to Register 2. As mentioned, it is recommended that the first queued command be placed in Register 0. The first command in the queue will be executed immediately.

Step 5. F82C5087 SM executes pipeline command 0.

Once the TGT DISC SEQ command is placed in Register 0, the F82C5087 SM begins to process the SCSI sequence stored in MCS0. This implied association between the pipeline command and the SCSI sequence is made because the SM MCS address pointer is pointing to location 0 of MCS 0 when the TGT DISC SEQ command is issued.

The F82C5087 SM executes the TGT DISC SEQ command, and clears the command from Register 0. After this command is successfully executed, the SM MCS address pointer is pointing to the next byte beyond the DISCONNECT message.

Step 6. F82C5087 SM executes pipeline command 1.

The F82C5087 SM proceeds to the next pipeline register, Register 1, for another command. Because this register contains a control command, CNTL ENB SEL, the F82C5087 SM executes the command without advancing the SM MCS address pointer to the beginning of the next MCS script (location 32). This is because the control commands do not require MCS access; therefore, their execution does not affect the positioning of the SM MCS address pointer.

Step 7. F82C5087 SM executes pipeline command 2.

The F82C5087 SM executes the TGT WFS SEQ in Command Pipeline Register 2. This command requires that the F82C5087 sequences through several SCSI bus phases, and optionally disconnects from the SCSI bus when the command is completed. Because this command requires MCS access the SM MCS address pointer is incremented to location 32 before command execution begins. As command execution progresses the target ID, identify message, etc. are transferred from the SCSI bus into MCS 1. When the command is successfully completed, the SM MCS address pointer is pointing to the last byte, plus one, that was received from the initiator.

Step 8. F82C5087 SM scans for next command, and stops at location 0.

After executing the TGT WFS SEQ command, the F82C5087 SM proceeds to scan Register 3 which is empty. After scanning this empty register, it circles around and scans Register 0. At this point Register 0 is empty; the F82C5087 SM realizes that the pipeline registers are all empty and stops scanning at Register 0.

Example 2. SCSI Select/Reselect Process.

For operations that require the F82C5087 to select/reselect a SCSI device, the first byte the MCS Address Register points to must be a byte containing the initiator/target SCSI bus ID. For example, if the F82C5087 command to be issued is the Select With ATN Sequence then the firmware must have performed consecutive write operations to MPU MCS Buffer Register 11 to set up the MCS as follows:

MCS Script	
SM MCS Address 0 (Register 9) ----->	Target ID byte
	Identify message
	Optional tag message
	Multi-byte command

Target ID

The initiator can address up to eight peripheral devices that are connected to a device controller. Each SCSI peripheral device attached to the device controller is assigned a SCSI ID at system installation. The eight lines on the SCSI data bus are used to identify which SCSI device(s) is sending a request to the initiator. During the SCSI bus ARBITRATION phase, these device IDs are compared to determine bus usage priority. The device which is tied to the line -HOST_D[7] has an ID of 7 and the highest priority when arbitrating for the SCSI bus.

Identify Message

The identify message is a single-byte message that specifies if a target is allowed to disconnect. It also specifies the initiator's LUN (logical unit number). This optional message is sent by the initiator and stored in the MCS.

Tag Message

This two-byte, optional message enables multiple commands to be sent to a target from a single initiator.

Multi-byte Commands

In this example, this portion of the queued SCSI sequence is a SCSI multi-byte command.

Once the MCS has been set up as shown, an *INIT SWA SEQ* command can be issued to one of the pipeline registers. Again, it is recommended that Register 0 be used. Because the SM MCS address pointer is at the beginning of an MCS script (location 0 of MCS 0), the F82C5087 will not advance this pointer to the beginning of the next MCS (location 32 of MCS 1) before starting to execute the *INIT SWA SEQ* command. The *INIT SWA SEQ* will perform the following steps:

- Arbitrate for the SCSI bus.
- Select the target whose ID has been stored at location 0 of the MCS.
- Send a one-byte identify message to the target to indicate if a disconnect will be allowed. This identify message will also indicate the initiator's logical unit number.
- If the modifier bit of the *INIT SWQ SEQ* command code equals 1, send the target a two-byte tag message.
- Send the target the multi-byte command stored in the MCS.

- Terminate execution at this point if the identify message disallowed a disconnect, or if the Override Disconnect Enable Bit (Register 20's bit 2) is asserted.
- Receive one of the following from the target if a disconnect is allowed:
 - one status byte and one message,
 - or one or two messages.
- Wait for the SCSI bus to go to the BUS FREE phase.

The status byte and/or message received from the target will be placed in MCS 0 after the multi-byte command. The SM MCS address pointer will point to the last byte processed, plus one.

To sort out contiguous status and message bytes, the F82C5087 tags all message bytes received over the SCSI bus. Before reading the MCS Data Register bit 5 of the SCSI Status Register 19 can be checked. If the next read of the MCS Data Register will produce a message byte, this bit will be asserted. Note that Register 19's bit 5 is valid only if Register 19's bit 7 is asserted.

The pipeline status registers contain the counts and the status of the SCSI bus phases that transpired during a command's execution. Refer to Chapter 4's description of the pipeline command registers for more details on the pipeline registers. Chapter 5 provides details on the pipeline commands.

Chapter 4

Registers

This chapter provides a description of the function and operation of each of the F82C5087's internal registers.

Function of Registers

The F82C5087 has 26 internal registers that are accessible from an external circuit. They can be categorized as control (write) registers, or status (read) registers. Their function is two-fold; they can be used to control the F82C5087's internal operation, and also to gain processing status and status results of the F82C5087's operation.

Each register has a unique address. The particular register's address is identified by the F82C5087's address bits A0 through A4, as illustrated in Table 4-1.

Table 4-1.

Addressing the F82C5087 Registers

Address					Register Number	OP	Register Name
A4	A3	A2	A1	A0			
0	0	0	0	0	R0	W R	Command Pipeline 0 Pipeline 0 Status
0	0	0	0	1	R1	W R	Command Pipeline 1 Pipeline 1 Status
0	0	0	1	0	R2	W R	Command Pipeline 2 Pipeline 2 Status
0	0	0	1	1	R3	W R	Command Pipeline 3 Pipeline 3 Status
0	0	1	0	0	R4	W R	SCSI Configuration SCSI Status
0	0	1	0	1	R5	W R	Interrupt Mask Interrupt Status
0	0	1	1	0	R6	W R	Select Timeout Extended Status
0	0	1	1	1	R7	W R	Mode Control Buffer Status
0	1	0	0	0	R8	W R	SCSI Clock Input Port 1
0	1	0	0	1	R9	W R	SM MCS Address SM MCS Address

Table 4-1.

Addressing the F82C5087 Registers (continued)

Address					Register Number	OP	Register Name
A4	A3	A2	A1	A0			
0	1	0	1	0	R10	W R	MPU MCS Address MPU MCS Address
0	1	0	1	1	R11	W R	MPU MCS Buffer MPU MCS Buffer
0	1	1	0	0	R12	W R	MPU FIFO Data MPU FIFO Data
0	1	1	0	1	R13	W R	FIFO Count FIFO Count
0	1	1	1	0	R14	W R	Maximum Sync Offset Current Sync Offset
0	1	1	1	1	R15	W R	Transfer Period Revision Number
1	0	0	0	0	R16	W R	Transfer Counter/ Low Byte
1	0	0	0	1	R17	W R	Transfer Counter/ Middle Byte
1	0	0	1	0	R18	W R	Transfer Counter/ High Byte
1	0	0	1	1	R19	W R	Auxiliary Control 1 Auxiliary Status 1
1	0	1	0	0	R20	W R	Auxiliary Control 2 Auxiliary Status 2
1	0	1	0	1	R21	W R	Sync/Async ID Current Target ID
1	0	1	1	0	R22	W R	CMD Group 6 & 7 Length SCSI Data
1	0	1	1	1	R23	W R	Sync ACK Out Counter Sync ACK Out Counter
1	1	0	0	0	R24	W R	Sync REQ Counter Sync REQ Counter
1	1	0	0	1	R25	W R	Auxiliary Control 3 Not Defined

Register Description

This section provides an operational description of a write or read operation in each of the F82C5087 internal registers. The registers are organized sequentially (0-25).

Note: Unless otherwise noted in the register's operational description, a hard reset condition clears all registers.

Registers 0-3 (0-3h)

Register Number	Operation	Register Name
0	Write	Command Pipeline 0
	Read	Pipeline 0 Status
1	Write	Command Pipeline 1
	Read	Pipeline 1 Status
2	Write	Command Pipeline 2
	Read	Pipeline 2 Status
3	Write	Command Pipeline 3
	Read	Pipeline 3 Status

Register 0-3 (0-3h) - Write Operation

These four pipeline registers are a circular queue to send commands to the F82C5087. A command placed in one of these registers could execute a simple task (e.g., send a message to the SCSI bus), or execute a sequence of tasks. For example, the following series of tasks are executed when the firmware engineer issues the TGT WFS SEQ (Target Wait for Select Sequence):

- Wait for select from initiator
- Receive ID message into MCS script
- Optionally receive tag message into MCS script
- Receive multi-byte command into MCS
- Optionally send disconnect message, and proceed to the SCSI bus phase of BUS FREE

The F82C5087 executes pipeline commands in a circular fashion starting with Register 0. For example if the firmware engineer places two commands in Registers 0 and 2, the F82C5087 starts command scanning at Register 0. Once locating the first command, the F82C5087 starts executing it. If this command completes without an error condition, the F82C5087 scans the next register, Register 1, which is empty. The F82C5087 proceeds to Register 2, locates the second command and starts executing it. If this command completes without an error, the F82C5087 scans Register 3 which is empty. Next the F82C5087 scans Register 0 which is empty. At this point the F82C5087 detects all four pipeline registers are empty, and stops scanning.

In this example, note that the F82C5087 is now positioned to start command execution at Register 0. Consequently, pipeline command sequences should begin at this location. Refer to Chapter 3's section "Pipelining Structure" for more details. For details on pipeline commands refer to Chapter 5.

OP	7	6	5	4	3	2	1	0
W	Pipeline Command Code				INIT Mode	TAR Mode	MOD Bit	INT

Bits 7-4 Pipeline Command Code

These bits contain the command code for the specific pipeline command. For instance, 0000 specifies the NO OPERATION command. Refer to Chapter 5 for details on each pipeline command.

Bit 3 Initiator Mode

If this bit is asserted, the F82C5087 assumes the initiator role on the SCSI bus. This bit and bit 2 should never be enabled within the same pipeline command.

Bit 2 Target Mode

The F82C5087 assumes operation in target mode on the SCSI bus when this bit is asserted. This bit and bit 3 control internal logic; they enable differential SCSI drivers/receivers when the DIF_ENB pin is high. This bit and bit 3 should never be enabled within the same pipeline command.

Bit 1 Modifier Bit

Several pipeline commands use this bit as a command modifier bit. For example, commands that transfer data between the SCSI bus and the MPU use this bit to enable 16-bit host adapter mode. The Initiator Select with ATN Sequence (INT SWA SEQ) uses this bit to allow a second message to be sent after the ID Message. The Target Send Busy Sequence (TGT SEND BSY) uses this bit to prevent the command from being cleared upon completion. Consequently, the command can be repeatedly executed until the sequence is aborted.

Bit 0 Interrupt when Command Complete

When this bit is asserted and this interrupt is enabled by the Interrupt Mask Register 5, the F82C5087 interrupts the MPU upon command completion.

Notes:

- When the F82C5087 is executing a SCSI command or command sequence and an unexpected condition occurs, the F82C5087 clears the current command and any other commands in Registers 0-3. Bit 1 (Extended Status) of the Interrupt Status Register 5 is set, and the Auxiliary Status Register 19 indicates the command number that caused the error. Extended Status Register 6 has the bit(s) set indicating which error occurred (e.g., bus phase error).
- The SM MCS Address Register and the MCS's contents are not affected by any error condition. Therefore, these registers and the pipeline registers can be examined to determine the number and type of SCSI bus phases that occurred prior to the error.

- A pipeline command can be aborted by writing a one to bit 5 (Soft Abort) of the SCSI Configuration Register 4. This soft abort terminates the currently executing command, sets bit 7 (BUSY) of the pipeline status registers to zero, and resets bit 6 of Register 4. The firmware needs to examine the status of the SCSI bus, the F82C5087's internal FIFO, and/or SM MCS Address Register, and deal with any conditions created by the command being aborted prematurely.
- Bits 6 - 0 are not cleared by a hard reset.

Registers 0-3 (0-3h) - Read Operation

The firmware can use these registers to obtain status information and to aid in error recovery.

OP	7	6	5	4	3	2	1	0
R	BUSY PHASE	CMND PHASE	STATUS PHASE	DATA PHASE	MSG_OUT COUNT			MSG_IN COUNT

Bit 7 BUSY

This bit is set when the MPU sends a new command to the pipeline register, and is cleared upon command completion. The firmware can poll this bit to monitor command progress. When a pipeline command terminates with an error, this bit is cleared in all the pipeline status registers.

Bit 6 COMMAND Phase

The SCSI bus COMMAND phase permits targets to request command information from initiators. When the F82C5087 detects a SCSI COMMAND phase during the execution of a pipeline command, this bit is asserted.

Bit 5 STATUS Phase

The SCSI bus STATUS phase permits targets to request status information be sent from targets to initiators. When the F82C5087 detects a SCSI STATUS phase during the execution of a pipeline command, this bit is asserted.

Bit 4 Data Phase

Data phase is a term used to encompass the SCSI bus DATA IN phase and DATA OUT phase. DATA IN phases permit targets to request data be sent from targets to initiators. DATA OUT phases permit targets to request data be sent from initiators to targets. When this bit is asserted, the F82C5087 detects a SCSI bus DATA IN or DATA OUT phase during pipeline command execution.

Bits 3-2 MESSAGE OUT Count

These bits contain an encoded count of the number of SCSI bus MESSAGE OUT phases that transpired during pipeline command execution.

Bits 1-0 MESSAGE IN Count

These bits contain an encoded count of the number of SCSI bus MESSAGE IN phases that occurred during pipeline command execution. For example, if these bits contain a decimal 2 and the F82C5087 is configured as a target, the F82C5087 sent two separate messages to the initiator during command execution. The messages could have been single or multi-byte.

Note:

- The F82C5087 contains no provision for overflow detection from bits 2 and 3, or bits 0 and 1.

Register 4 (4h)

Register Number	Operation	Register Name
4	Write Read	SCSI Configuration SCSI Status.

Register 4 (4h) - Write Operation

This register aids in control of activity on the SCSI bus.

OP	7	6	5	4	3	2	1	0
W	Force Connect	Hard Abort	Soft Abort	DIS ARBIT	Extra Data Setup	SCSI Bus ID		

Bit 7 Force Connect

This bit should only be used for diagnostic purposes. It forces the F82C5087 to connect to the SCSI bus as either initiator or target based on the last executed command mode.

Bit 6 Hard Abort

When this bit is asserted, all pipeline command registers are cleared immediately, and all SM activity is halted. This bit is also cleared.

This bit should be used only after the Soft Abort Bit has failed, and a "hung" command needs to be terminated. When the Hard Abort Bit is asserted, the SM Error Bit (bit 7) of the Extended Status Register (Register 6) is asserted, and a pipeline COMMAND COMPLETE condition is generated.

Bit 5 Soft Abort

When asserted, this bit causes a soft abort. This bit, unlike a hard abort, attempts to terminate the command execution in an orderly way, as soon as possible, rather than aborting the command immediately. For example, asserting this bit might halt command execution after the selection timeout has expired and the target has not responded, rather than immediately aborting.

Bit 5 clears itself and all command pipeline registers, and generates a pipeline COMMAND COMPLETE condition. When this bit is asserted, the SM Error Bit (bit 7) of the Extended Status Register 6 is set.

Because of the abnormal command abort the MPU might have to perform a clean up. Therefore, the MPU should check the status of the SCSI bus, and contents of the F82C5087's internal FIFO and/or the SM MCS Address Register, and status registers after a soft abort.

Bit 4 Disable Arbitration

When this bit is asserted, the F82C5087 omits the ARBITRATION phase and proceeds directly to the SELECTION phase. Because bus arbitration is unnecessary in systems with only one initiator, this bit should be asserted for systems using the single initiator option.

Bit 3 Extra Data Setup

When asserted, this bit instructs the F82C5087 to add two clocks of setup time to the data during asynchronous writes to the SCSI bus. Asserting this bit has no effect on synchronous data transfers because such transfers are locked into the system clocking.

Bits 0-2 SCSI Bus Identifier

A SCSI bus can support up to eight physical SCSI devices. Each physical device must be assigned a unique identifier. Bits 0-2 specify an encoded SCSI bus device identifier. During the SCSI bus ARBITRATION phase, the F82C5087 puts this decoded ID on the SCSI bus.

Devices can be assigned identifiers (IDs) from 0 through 7. These IDs have priorities for bus arbitration purposes. During arbitration, device IDs on the SCSI bus are compared to determine bus usage priority. A SCSI device with an ID of 7 has the highest priority.

Register 4 - Read Operation

This register aids in the monitoring of activity on the SCSI bus.

OP		7		6		5		4		3		2		1		0
R		CON- NECTED		SCSI SEL		SCSI BSY		SCSI ACK		SCSI REQ		SCSI BUS PHASE MSG C/D I/O				

Bit 7 Connected to the SCSI Bus

When this bit is asserted, the F82C5087 is connected to the SCSI bus. Note that references in this manual to the SCSI bus condition as being connected/disconnected are referring to this bit.

Bit 6 SCSI Select

Bit 6 indicates the unlatched status of the SCSI bus -SEL signal.

Bit 5 SCSI Busy

This bit indicates the unlatched status of the SCSI bus -BSY signal.

Bit 4 SCSI Acknowledge

Bit 4 indicates the unlatched status of the SCSI bus -ACK signal.

Bit 3 SCSI REQ

Bit 3 indicates the unlatched status of the SCSI bus -REQ signal.

Bits 0-2 SCSI Bus Phase

These bits indicate the unlatched condition of the following three SCSI bus control signals: -MSG, -C_D, and -I_O.

Note:

- Unlatched signals can be changing asynchronously with respect to the MPU's processing. Therefore, to ensure valid data two readings of unlatched signals should be taken and compared.

Register 5 (5h)

Register Number	Operation	Register Name
5	Write Read	Interrupt Mask Interrupt Status

Typically, when the F82C5087 encounters any of the following events while it is processing a pipeline command, it can be programmed to generate an interrupt to the MPU:

- Command execution completed
- Unexpected condition occurred
- Detection of a parity error or overrun condition
- A phase change occurred
- Detection of the SCSI SELECTION/RESELECTION, RESET, or ATTENTION signal.

Register 5 (5h) - Write Operation

This register is used to specify under what conditions the F82C5087 should generate interrupts to the MPU. They can enable or disable the associated mask bit by writing to this register. If the Interrupt Mask Bit is asserted and the corresponding event occurs (e.g., the command is completed and the COMMAND COMPLETE Mask Bit (bit 0) is asserted), the F82C5087 will generate an interrupt to the MPU.

OP	7	6	5	4	3	2	1	0
W	SCSI ATN Mask	SCSI RST Mask	SEL Mask	Change Phase Mask	FIFO Half-Empty Mask	Parity/Overrun Mask	Extnded Status Mask	CMD CMLPTE Mask

Register 5 (5h) - Read Operation

A read to this register provides the current status of the interrupt events. Interrupting events for bits 0,1, 2, 3, 4, and 6 are latched; bits 5 and 7 are passed unlatched through to the mask bit gate. The trailing edge of the MPU read cycle clears any latched bits.

OP	7	6	5	4	3	2	1	0
R	SCSI ATN	SCSI RST	SEL	Change Phase	FIFO Half-Empty	Parity Error	Extnded Status	CMD CMLPTE

Bit 7 SCSI ATTENTION

This unlatched bit is asserted when the SCSI bus -ATN signal is asserted. The F82C5087 must be connected to the SCSI bus for this bit to be asserted. It is valid in target and initiator mode.

Bit 6 SCSI RESET

This latched bit is asserted when the SCSI bus reset signal has been asserted.

Bit 5 SELECT

When the F82C5087 is currently being selected or reselected on the SCSI bus, this unlatched bit is asserted. Selection and/or reselection must be enabled for this bit to be asserted. If both selection and reselection are enabled, the firmware must read bit 0 of SCSI Status Register 4 to determine if a selection or reselection occurred. If bit 0 is equal to zero, selection occurred. If bit 0 is equal to 1, reselection occurred.

Bit 4 Change Phase

This latched bit indicates a phase change has occurred when the F82C5087 is operating in initiator mode.

Bit 3 FIFO Half-empty

This latched bit indicates that the 32nd byte in the FIFO has been accessed. This bit provides a means for the host adapter driver code to gain control of the host processor when the F82C5087 needs more data (non-DMA mode).

Bit 2 Parity Error/Overrun

When a parity error is detected or an overflow condition has developed, this latched bit is asserted. Refer to the description of Auxiliary Status Register 20 later in this chapter, for further details on the type of parity error and/or overflow condition.

Bit 1 Extended Status

This latched bit is set when a pipeline command is terminated abnormally. This asserted bit indicates that extended status is available in Extended Status Register 6. When this bit is set, all pipeline commands are aborted. The number of the pipeline register last executed is obtained by reading Auxiliary Status Register 19.

Bit 0 Command Complete

This latched bit is asserted when the execution of a pipeline command, that had bit 0 (Interrupt When Command Complete Bit) asserted, is completed. A COMMAND COMPLETE event is also generated when an error condition has developed and the pipeline command was terminated abnormally.

Note:

- As mentioned previously, all latched bits are cleared by the trailing edge of the Interrupt Status Register read strobe. The F82C5087 contains extra logic to ensure that only bits that are active at the beginning of the read strobe are cleared by the strobe's trailing edge. This implies that the firmware does not have to take any precautions to avoid loss of data when reading this register.
- In host adapter mode, the F82C5087 driver code could issue the F82C5087 a command (e.g., INIT DATA PTS) to transfer data from the host MPU bus to the SCSI bus. The driver code could

then burst transfer (IORDY is connected to the MPU Wait State Generator) 64 bytes into the F82C5087's FIFO, enable an interrupt on bit 3, then transfer control back to the operating system. When the target begins to accept data or an error occurs, and the FIFO is half-empty, bit 3 will generate an interrupt. This interrupt causes the host processor to return control to the F82C5087 driver code. At this point, the F82C5087 driver code would burst transfer 32 bytes to the F82C5087, and wait for bit 5 of FIFO Count Register 13 to be deasserted. It would then repeat the process until all the bytes had been transferred into the FIFO.

Register 6 (6h)

Register Number	Operation	Register Name
6	Write Read	Select/Reselect Timeout Extended Status

Register 6 (6h)- Write Operation

This register specifies the selection/reselection timeout . Selection timeout is the maximum time the F82C5087 will wait for a target's response to a SELECT command before terminating the command presently in progress. Reselection timeout is the maximum time an initiator is allowed to respond to a target's RESELECT command.

OP	7	6	5	4	3	2	1	0
W	Select/Reselect timeout							

Bits 0-7 Select/Reselect Timeout

If the value in Register 8 produces a 400 ns period, this register will produce 1.6 ms resolution for the timeout period (400 ns * 4096). To produce a timeout period of 250 ms this register should be loaded with 250 ms divided by 1.6 ms which equals 156.

Register 6 (6h) - Read Operation

If an unexpected condition is detected (Extended Status bit 1 of Register 5 is asserted), this register contains valid data. It indicates why the command was terminated prematurely. The firmware should realize that all pipeline commands have been aborted.

OP	7	6	5	4	3	2	1	0
R	SM Error	Time-out	ARB Error	Soft Reset	Bus Phase Error	Connect Error	ATN Error	Illegal CMD Error

Bit 7 State Machine (SM) Error

A state machine error has occurred when this bit is asserted. Any of the following five conditions cause state machine errors to occur:

- The command was aborted via a soft or hard abort.
- The F82C5087 received an illegal message. When the F82C5087 does not receive the expected message, this bit is set and the command is aborted immediately. For instance, if the F82C5087 is executing the TGT WFS SEQ command and the first message received is not the identify message, the command is aborted immediately, and this bit is set.

In target mode, the illegal message is deposited in the MCS.

In initiator mode, the F82C5087 does not handshake with the illegal message; it leaves it on the bus with the -REQ signal

asserted. The firmware must decide whether or not to receive the message.

- Illegal group code encountered. In order for the F82C5087 to determine the SCSI command length, the group code must be valid. If the F82C5087 encounters a group code of 3, or 4, the command does not have a valid group code.
- While executing the TGT SEND BSY sequence the F82C5087 was reselected. The F82C5087 was expecting to be selected.
- The target disconnected before sending a COMMAND COMPLETE or DISCONNECT message.

Bit 6 Timeout

This bit is asserted when the F82C5087 is selecting/reselecting a SCSI device and a timeout occurs; command execution is terminated. Time period is programmable via the value written to Select Timeout Register 6.

Bit 5 Arbitration Error

If the F82C5087 is selected while arbitrating for the SCSI bus, this bit is asserted. Command execution is terminated.

Bit 4 Soft Reset

A soft reset condition occurs when the SCSI bus reset signal is asserted, and the firmware engineer has asserted the DIS SRI Bit (bit 4 of Auxiliary Control Register 19). Under these conditions, the F82C5087 is not hard reset. Command execution is terminated immediately and this bit is asserted. The soft reset forces the F82C5087 to disconnect from the SCSI bus if connected. All internal registers remain unaltered. Note that when a soft reset occurs, bit 6 of Interrupt Status Register 5 is asserted.

Bit 3 Bus Phase Error

Throughout all initiator commands the correct SCSI bus phase must be maintained. If the target switches to an unexpected phase and asserts the -REQ signal, this bit is asserted. Command execution is terminated immediately.

Bit 2 Connect Error

Any F82C5087 command accessing the SCSI bus checks the connected bit's status. This status is contained in bit 7 of SCSI Status Register 4. Certain commands, such as TGT SEND MSG, require that the F82C5087 is connected to the SCSI bus throughout the execution phase.

Other commands, such as INIT SWA SEQ, require that the connect state changes with execution. For instance, this command requires the F82C5087 to be disconnected from the SCSI bus when execution begins. However, when the F82C5087 wins SCSI bus arbitration, it requires the F82C5087 to remain connected until the DISCONNECT

message is sent. If the target disconnects before command completion, this bit is asserted. Command execution is terminated immediately.

Bit 1 Attention Error

The currently executing target command is terminated with this bit set when all the following conditions exist:

- An asserted SCSI -ATN signal
- The F82C5087 is in target mode
- An enabled -ATN pin (bit 2 of Register 19 is set to zero)
- In a phase other than SELECTION or MESSAGE OUT phase
- Not executing TGT XFER BYTE or TGT SEND BSY command.

When these conditions exist, the command termination behavior of the F82C5087 depends on which of the four types of transfer is transpiring.

- Synchronous Data Transfers. If the F82C5087 is transferring synchronous data, the F82C5087 terminates the command after the -REQ pulse is deasserted. As with asynchronous transfers, data remains in the FIFO and the current MPU and/or memory data bus cycle is allowed to complete. The Offset Counter continues to count initiator -ACK pulses until the target changes bus phase.
- Asynchronous Data Transfers. If the F82C5087 is transferring asynchronous data, the F82C5087 completes the last SCSI bus REQ/ACK handshake. The current MPU and/or memory data bus cycle is allowed to complete then command execution is terminated. Data may remain in the F82C5087's internal FIFO.
- SCSI Bus to MCS Transfers. If the F82C5087 is transferring a command from the SCSI bus into an MCS script, the F82C5087 does not assert -REQ for the next byte. Command execution is terminated.
- MCS to SCSI Bus Transfers. If the F82C5087 is sending messages or status from an MCS script to the SCSI bus, the F82C5087 terminates command execution after the initiator deasserts the -ACK signal.

Bit 0 Illegal Command Error

When the F82C5087 is issued an undefined command, this bit is asserted.

Register 7 (7h)

Register Number	Operation	Register Name
7	Write Read	Mode Control Buffer Status

Register 7 (7h) - Write Operation

This register specifies the F82C5087's operational modes. The F82C5087's main mode of operation can be specified (e.g., mode when it is transferring data to/from buffer memory) with this register's bit 7. This register's bit 6 can be used to specify whether the F82C5087 is operating in master or slave mode when performing SCSI interface operations (e.g., SCSI bus data transfers).

This register also enables/disables several pin functions.

OP	7	6	5	4	3	2	1	0
W	Slave Mode	SCSI Bus Slave	MEM_D Parity Check	SCSI Data Parity	DMA Handshake	FIFO_RDY Pin	FIFO_RST	Host Adapter Pins

Bit 7 Slave Mode

When this bit is asserted, the F82C5087 is operating in slave mode. In this mode, the F82C5087 is under the control of a master F82C5087 chip. Only the FIFO is active in slave mode; SCSI phases or MCS access is not defined. Asynchronous and synchronous data transfers proceed as in master mode. The Transfer Counter can be loaded with the number of bytes to transfer, or an infinite transfer count (Register 20, bit 3) can be programmed. To ensure correct synchronization, the firmware must send pipeline command(s) to the slave F82C5087(s) before sending the identical command(s) to the master F82C5087.

Because the F82C5087 slave does not have a SCSI bus ID, slave mode forces the chip to ID 0. Hence, the firmware specifies synchronous data transfer by setting bit 0 (SCSI bus ID 0) of Register 21, or asynchronous data transfers by clearing bit 0.

When the F82C5087 is operating in slave mode (this bit is asserted), it drives the -WPAR and -DSYNC signals.

Bit 6 SCSI Bus Slave

The firmware should assert this bit only if bit 7 is also asserted. It forces the SCSI bus interface to operate in slave mode. Consequently, REQ/ACK control signals are not generated. Rather the -SSYNC pin outputs a ready condition when the slave F82C5087 SCSI interface is ready for additional data.

Bit 5 Memory Data Bus Parity Check

When this bit is asserted, data received over the memory data bus is parity checked. Data sent to the memory data bus from the F82C5087 contains the flow-through parity bit.

When this bit is deasserted and the F82C5087 receives MPU or memory data bus data, the F82C5087 must calculate parity before sending the data to the FIFO or MCS.

Bit 4 SCSI Data Parity

Asserting this bit enables parity checking on SCSI bus transfers and all FIFO output transfers. When a parity error is detected, bit 2 (*Parity/Overrun*) of *Interrupt Status Register 5* is asserted.

When a parity error is detected and the Halt on Data Parity Error Bit (bit 0 of Auxiliary Control Register 20) is asserted, the F82C5087 terminates command execution with the Interrupt Status Bits 0,1, and 2, and *Extended Status Bit 7* asserted. The precise time of command termination depends on the type of transfer as described below:

- Synchronous/Asynchronous Data Transfers in Target Mode. The F82C5087 receives the byte with the parity error, waits for the completion of the memory data bus and/or MPU cycle then terminates the command. The firmware should read the Up Transfer counter, FIFO Counter, the Offset Counter, and Buffer Status Register to determine the number of transferred bytes.
- Synchronous/Asynchronous Data Transfers in Initiator Mode. The F82C5087 waits for the memory data bus and/or MPU cycle to complete then terminates the command. As mentioned previously, the firmware should check the Up Transfer Counter, FIFO Counter, and the Offset Counter, and Buffer Status Register. Because -ACK is not asserted, the F82C5087 does not receive the byte with the parity error. This provides the firmware with the option of asserting the -ATN signal.
- MCS Transfers in Target Mode. The byte with the parity error is received; however, it is not transferred to the MCS. The F82C5087 then terminates the command.
- MCS Transfers in Initiator Mode. The byte with the parity error is not received. This allows the firmware engineer the option to assert the -ATN signal.

Bit 3 DMA Controller Handshake

Normally the F82C5087 does not assert the -DMA_REQ signal for the next buffer RAM transfer until the DMA controller has deasserted the -DMA_ACK signal. Asserting this bit, enables the F82C5087 to assert the next -DMA_REQ signal while the previous -DMA_ACK signal is still asserted.

Bit 2 FIFO Ready Pin

For AT Bus compatibility, the FIFO_RDY pin is a tri-state output. Asserting this bit, enables the F82C5087 to drive this pin to indicate the MPU FIFO Data Register is ready for additional data. It can be used for 8- or 16-bit MPU data transfers to/from the memory data bus, or the SCSI bus. This pin is typically connected to a DMA controller on the MPU bus.

Bit 1 FIFO RESET

Typically, the F82C5087 resets the internal FIFO before starting data transfers. When this bit is asserted, the F82C5087 does not reset the FIFO before starting to execute commands that will transfer data. The firmware is responsible for resetting the FIFO by issuing the CNTL RST FIFO pipeline command.

Bit 0 Host Adapter Pins

When this bit is asserted, the F82C5087 drives the open drain IORDY and -16BHAM pins. When the MPU FIFO Data Register is accessed yet not ready to receive/send more data from/to the MPU, the IORDY pin is deasserted. Typically this pin is connected to the MPU Wait State Generator. The -16BHAM is asserted when the MPU accesses the MPU FIFO Data Register for a 16 bit data transfer. Typically, this pin enables the upper byte (bits 8-15) to transfer to the MPU memory. Refer to Figure 2-7 for a functional pin out of these host adapter pins.

Notes:

- The SCSI bus mode can be viewed as a subordinate mode of the chip's main mode of operation as a master or slave. For more details on this topic refer to Chapter 3's section "Master/Slave Mode".
- SCSI bus slave chip(s) must have their -SSYNC, -REQ, and -ACK pins connected to the equivalent pins on the bus master chip.
- When parity checking is enabled on memory data bus transfers, pin 4 of Input Port 1 contains the parity bit.
- In the F82C5087, parity is passed through. For instance, parity received from the SCSI bus is passed through to the memory data bus.

Register 7 (7h) - Read Operation

This register provides status information on the FIFO data buffers . Dual buffers (ping pong) are used on the memory data bus and the SCSI data bus to accelerate the speed of data transfers between these buses and the FIFO.

OP	7	6	5	4	3	2	1	0
R	Not Used			MPU BUF	MEM_D BUF_B	MEM_D BUF_A	SCSI BUF_B	SCSI BUF_A

Bit 7-5 Not Used

Bit 4 **Microprocessor Buffer**

When this bit is asserted, it indicates that the MPU's FIFO data buffer contains data.

Bit 3 **Memory Data Buffer B**

When this bit is asserted, it indicates that the memory data buffer B contains data.

Bit 2 **Memory Data Buffer A**

When this bit is asserted, it indicates that the memory data buffer A contains data.

Bit 1 **SCSI Buffer B**

When this bit is asserted, it indicates that the SCSI buffer B contains data.

Bit 0 **SCSI Buffer A**

When this bit is asserted, it indicates that the SCSI buffer A contains data.

Notes:

- The MPU's data bus is not double buffered because the MPU activities usually do not require high speed buffering.
- For error recovery the firmware should also check the FIFO Counter and Transfer Counter's contents.

Register 8 (8h)

Register Number	Operation	Register Name
8	Write Read	SCSI Clock Input Port 1

Register 8 (8h) - Write Operation (Lower 4 bits only)

This register sets the SCSI clock. SCSI bus arbitration times are relatively long in comparison to the width of the system clock input to the F82C5087.

Consequently, this register sets a divide-by-factor to create an approximately 400 ns square wave ($CP \times n = 400 \text{ ns}$, where n =register value). For example, a 406 ns square wave would be produced if the decimal value 12 (the number of clock periods minus one) is placed in this register and the clock input is 33 MHz. To produce a symmetrical square wave, the firmware must load an even count into this register.

OP	7	6	5	4	3	2	1	0
W	Not Used					SCSI Clock		

Register 8 (8h) - Read Operation

The contents of this register provide the firmware engineer with status information concerning the SCSI bus RST, the BUF_IN, and Input Port 1 signals.

OP	7	6	5	4	3	2	1	0
R	RST PIN		BUF IN PIN	INPUT PINS				

Bit 7 Reset Pin

This bit provides the firmware with the unlatched status of the SCSI bus reset signal. Bit 6 (SCSI RST) of the Interrupt Status Register 5 latches the SCSI bus reset signal. When the firmware receives an interrupt from the reset pin, this pin can be read to determine if the RST signal is still asserted.

Bit 6 Reserved

Bit 5 Buffer In Pin

This bit provides the firmware with the status of the BUF_IN pin. In non-differential SCSI mode, the BUF_IN pin is internally connected to the 48 mA BUF_OUT pin.

Bit 4-0 Input Pins # 4-0

These bits provide the firmware with the status of Input Port 1 pins 4-0. The input port is normally used to provide the firmware with the SCSI bus ID.

When the memory data bus parity is enabled, Input Port 1's pin 4 must be used for this parity bit.

Register 9 (9h)

Register Number	Operation	Register Name
9	Write/Read	SM MCS Address Pointer

Register 9 (9h) - Write Operation

Register 9 specifies an address pointer in the MCS for the state machine's use. The F82C5087's state machine (SM) uses this address pointer to the MCS to send SCSI commands, status and/or messages.

The F82C5087 contains 192 bytes of MCS space. Therefore, the address placed in this register should be between decimal 0 and decimal 191. Decimal values 192 through 255 access the F82C5087's other section of memory, the FIFO address space.

If a new pipeline command that requires MCS access begins execution and this register is not pointing to location zero of an MCS script, this register is incremented. It is incremented to point to location zero of the next MCS. For example, if this register contains an MCS address pointer of 5 and the new pipeline command requires MCS access, then this register advances to location 32. Location 32 is the starting address of MCS1. Note in this example, that MCS auto advance is not disabled.

OP	7	6	5	4	3	2	1	0
W	State Machine (SM) Message/Command Script (MCS) Address Pointer							
R								

Notes:

- This register is set up initially by the firmware.
- This register should not be loaded with a value greater than 191.
- The MCS space is considered circular in nature. Consequently, MCS5 follows MCS4 and MCS0 follows MCS5.

Register 9 (9h) - Read Operation

The firmware can read this register to verify execution of a pipeline command, and to aid in error recovery.

If the SM is sending data to the MCS, this register points to the location where the next byte is placed. If the SM is reading data from the MCS, this register points to the location where the next data byte is read.

Note:

- The SM has third priority in accessing F82C5087 memory. FIFO IN requests have first priority; FIFO OUT requests have second.

Register 10 (0Ah)

Register Number	Operation	Register Name
10	Write/Read	MPU MCS Address

Register 10 (0Ah) - Write/Read Operation

This register specifies an address in the MCS for the MPU's use. The MPU uses this address pointer to the MCS to send SCSI commands, status and/or messages.

Because there are 192 bytes of F82C5087 Command/Message space the address placed in this register should be between decimal 0 and decimal 191. Decimal values 192 through 255 access the F82C5087's other section of memory, the FIFO address space.

OP	7	6	5	4	3	2	1	0
W	Microprocessor Message/Command Script (MCS) Address Pointer							
R								

Notes:

- This register is incremented every time the MPU processes a byte from the MCS space.
- The firmware should be careful when writing data in the FIFO address space (addresses 192-255) because erroneous data results if the F82C5087 is concurrently using the FIFO for data transfers.

The firmware can access the MCS at any time. Thus, the firmware could be accessing an MCS script while the F82C5087 is sending SCSI commands, and/or messages to the SCSI bus, or transferring data through the FIFO. For example, the F82C5087 could be executing an INIT DATA STM pipeline command.

- The MPU has the lowest priority in accessing F82C5087 memory. FIFO IN, FIFO OUT, and SM requests have a higher priority.
- Refer to bit 7 of Auxiliary Status Register 19 for a discussion of the MCS Ready Bit.

Register 11 (0Bh)

Register Number	Operation	Register Name
11	Write/Read	MPU MCS Buffer

If the MPU's Wait State Generator is not connected to the F82C5087's IORDY pin, the MPU must poll Register 19's MCS Data Ready Bit (bit 7) before transferring data to/from this register. This F82C5087 internal register provides a buffer between the MPU and the MCS.

Register 11 (0Bh) - Write Operation

Information (e.g., commands, status, messages) written to this register is stored in the MCS at the address specified in the MPU MCS Address Register. Every time information is written to Register 11, the MPU MCS Address Register is incremented by one.

Register 11 (0Bh) - Read Operation

When a MPU read access occurs, the information (e.g., commands, status, messages) at the MCS address specified in the MPU MCS Address Register is latched into Register 11. Every time information is read from Register 11, the MPU MCS Address Register is incremented by one.

Because the F82C5087 must support MPUs that do not have the capability to generate wait states on MPU access, the firmware engineer must perform a dummy read of this register any time the MPU MCS address pointer is changed. This transfers the first byte from the MCS to this register. For write operations, there are no special considerations.

OP		7		6		5		4		3		2		1		0
W	Microprocessor															
R	Message/Command Script Data															

Register 12 (0Ch)

Register Number	Operation	Register Name
12	Write/Read	MPU FIFO Data

Register 12 (0Ch) - Write/Read Operation

This register is used when the MPU is transferring data to/from the SCSI bus or the memory data bus. It is used to pass the transferred data to/from the MPU and the F82C5087's FIFO.

There are several ways the MPU can check if data is available or if the FIFO is ready for another byte:

- Connect the FIFO Request (FIFO_REQ) pin to a DMA controller on the MPU bus. The DMA controller should be programmed to send/receive data from this register. If -MPU_ACK and MPU_AEN are asserted, MPU access is forced to Register 12 and an address is not required.
- Poll Auxiliary Status Register's bit 6 before reading or writing Register 12. An asserted bit indicates there is data ready for transfer.
- Connect the F82C5087's Input/Output Ready (IORDY) pin to the MPU Wait State Generator. Then open loop burst data to this register. If the MPU gets ahead of the F82C5087, the IORDY pin is deasserted. This forces the MPU to lengthen its read or write strobe. When the IORDY pin is reasserted, the access is completed.
- Because the MPU bus does not contain a parity bit, parity is generated on MPU writes to this register. On MPU reads, parity is checked as the byte is read from the FIFO; however, the bit is discarded when it is transferred to the MPU.

OP		7		6		5		4		3		2		1		0
W																
R																

Microprocessor
FIFO Data

Register 13 (0Dh)

Register Number	Operation	Register Name
13	Write/Read	FIFO Count

Register 13 (0Dh)- Write Operation

The F82C5087 has a 64-byte internal FIFO. For diagnostic purposes this register can be preset to any value (0-63). For example, writing a 10 to this register forces the F82C5087 to assume there are 10 remaining bytes in the FIFO.

OP	7	6	5	4	3	2	1	0
W	FIFO Count							

Register 13 (0Dh)- Read Operation

During normal operation, the firmware can read this register to determine the number of bytes in the FIFO. However, if data transfers are in progress, the firmware should use only bits 5, 6 and 7. Bits 0-4 should be disregarded. They are invalid because the count changes asynchronously with respect to the MPU processing activities.

OP	7	6	5	4	3	2	1	0
R	FIFO Empty	FIFO Full	Number of bytes in FIFO					

Bit 7 **FIFO Empty**

When this bit is asserted, the FIFO is empty; there is no data in the FIFO.

Bit 6 **FIFO Full**

When this bit is asserted, the FIFO is full and cannot accept any more data. When it is set, the FIFO contains 64 bytes of data.

Bit 5-0 **Number of bytes in FIFO**

These bits contain a count of the current number of bytes in the FIFO. These bits should be read only when the firmware is certain data transfer has been terminated. Bit 5 can be polled asynchronously and tested for a FIFO half full/empty condition.

Register 14 (0Eh)

Register Number	Operation	Register Name
14	Write Read	Maximum Sync Offset Current Sync Offset

Register 14 (0Eh) - Write Operation

The firmware can use this register to set the maximum synchronous offset, or to preset the current offset count.

OP	7	6	5	4	3	2	1	0
W	Unlimtd Offset	Not Used	Maximum Sync Offset/ Current Sync Offset					

Bit 7 Unlimited Offset

Asserting this bit forces the F82C5087 to an unlimited offset when performing synchronous data transfers.

If the F82C5087 is sending data, it continues to send data until the transfer count equals 0 and the FIFO is empty. In target mode, the initiator's -ACK pulses are disregarded. In initiator mode, -REQ pulses from the target are disregarded.

This bit is disregarded if the F82C5087 is receiving SCSI bus synchronous data.

Bit 6 Not Used

Bits 5-0 Maximum Synchronous Offset/Current Synchronous Offset

If bit 1 of Auxiliary Control Register 2 is deasserted, a write to this register sets the maximum synchronous offset, 63 bytes. If bit 1 of Auxiliary Control Register 2 is asserted, a write to this register presets the current offset count. This is useful for diagnostics and error recovery.

Register 14 (0Eh) - Read Operation

This registers specifies the current offset count. During normal operation, the firmware can read this register to determine the current offset count. However, if data transfers are in progress, the firmware should use only bits 6 and 7. Bits 0-5 should be disregarded because they are invalid; they are not valid because the count changes asynchronously with respect to the MPU processing activities.

OP	7	6	5	4	3	2	1	0
R	Zero Offset	Max Offset	Current Offset Count					

Bit 7 Offset Equals Zero

This bit indicates that the current offset counter is zero.

Bit 6 Offset Equals Maximum

When this bit is asserted, the current offset count is equal to the maximum offset count.

Bits 5-0 Current Offset Count

The firmware can obtain the current offset count by reading these bits.

Register 15 (0Fh)

Register Number	Operation	Register Name
15	Write Read	Transfer Period Revision Number

Register 15 (0Fh) - Write Operation

The F82C5087 can handle synchronous transfers at 5.3 megabytes/second, or 188 ns/ transfer period . If the SCSI device connected to the F82C5087 cannot receive/send data at this rate, delay clocks must be added to decrease the transfer rate. This register specifies the minimum number of clock cycles (minus three) between -REQ and -ACK signals for targets and initiators.

For targets, it specifies the minimum number of clock cycles between the leading edges of the -REQ signal to the next -REQ pulse. For initiators, it specifies the minimum number of clock cycles between the leading edges of -ACK pulses.

Use the following equation: Register Value = (Sync Transfer Period/CP) - 3. For example, for synchronous transfers at 5.3 megabytes/second this register would contain a 3; $[188\text{ns} / (1 / 33 \text{ MHz})] - 3 = 6 - 3 = 3$.

OP	7	6	5	4	3	2	1	0
W	Not Used			Transfer Period				

Note:

- For a write operation, only the lower 5 bits are valid.

Register 15 (0Fh) - Read Operation

This register specifies the F82C5087's current revision number and the status of the power-on flag . The power on flag is asserted at power up, and cleared by asserting bit 6 of Auxiliary Control Register 19.

OP	7	6	5	4	3	2	1	0
R	Power On Flag	Not Used			Current F82C5087 Revision Number			

Bit 7 Power On Flag

This flag is set when power is first supplied to the F82C5087. It is used for such purposes as power up testing, and waiting for a disk drive to spin up. This flag is cleared by writing a one to Auxiliary Control 1's (Register 19) RST Power Flag Bit (bit 6).

Bits 6-4 Not Used

Bits 3-0 Current F82C5087 Revision Number

These bits form the encoded F82C5087 revision number. The count began with the prototype part at number 0.

Registers 16 - 18 (10 - 12h)

Register Number	Operation	Register Name
16	Write/Read	Transfer Counter/Low Byte
17	Write/Read	Transfer Counter/Middle Byte
18	Write/Read	Transfer Counter/High Byte

Register 16 - 18 (10 - 12h)- Write Operation

The F82C5087 contains two 24-bit transfer counters . These registers contain the 3-byte transfer counter. A write to these registers presets the Down Transfer Counter . Once the Down Transfer Counter is loaded, the SM controls it. The SM decrements it when making a request for more data INTO the FIFO. The Up Transfer Counter counts the number of bytes transferred OUT of the F82C5087 FIFO. It is reset when the F82C5087 resets the FIFO. This register is useful for error recovery.

Notes:

- All bytes processed by the F82C5087 are accounted for by the 24-bit Up Transfer Count, FIFO Count, and Buffer Status Registers.
- The 24-bit Down Counter is not cleared by a reset condition; the 24-bit Up Counter is cleared.

Registers 16 - 18 (10 - 12h) - Read Operation

The contents of these registers specify the 3-byte transfer counter. When bit 4 of the Auxiliary Control Register 2 (Register 20) is deasserted, these registers contain the decrementing transfer count. When bit 4 is asserted, these registers contain the incrementing transfer count.

Register 19 (13h)

Register Number	Operation	Register Name
19	Write Read	Auxiliary Control 1 Auxiliary Status 1

Register 19 (13h) - Write Operation

This register specifies pin functions, and controls reset conditions in the F82C5087.

OP	7	6	5	4	3	2	1	0
W	RST 5087	RST Power Flag	RST OF Flags	DIS SRI Pin	DIS RO Pin	DIS ATN Pin	Polarity INT Pin	Interrupt Tri-State Enable

Bit 7 Reset F82C5087

Asserting this unlatched bit resets all internal F82C5087 registers except this one. An external reset via the `-RESET_CAP` or the `-RESET_IN` pin provides the same function, except this register is also reset.

Bit 6 Reset Power-on Flag

At power up, the F82C5087 is reset by approximately a 5 microseconds pulse. This reset signal asserts the power-on flag (bit 7 of Register 15). The firmware can check this flag to determine the type of reset that occurred. The power-on flag is cleared when this unlatched bit is asserted.

Bit 5 Reset Overflow Flags

Asserting this unlatched bit resets the parity and over/underrun flags in Auxiliary Status Register 20.

Bit 4 Disable SCSI RESET_IN Pin

When the firmware asserts this bit, it prevents the SCSI bus `-RESET_IN` signal from hard resetting the F82C5087. The F82C5087 still disconnects from the SCSI bus and interrupts the MPU via bit 6 of the Interrupt Status Register. When this bit is asserted and a SCSI reset occurs, the occurrence is referred to as a soft reset.

Bit 3 Disable RESET_OUT Pin

The `-RESET_OUT` pin is asserted when the F82C5087 is reset via an internal power on, `-RESET_IN` pin, program reset (Register 19's bit 7), or `-RESET_CAP` pin.

If the firmware asserts this bit, the `-RESET_OUT` pin is disabled.

Bit 2 Disable ATTENTION Pin

In initiator mode the F82C5087 does not drive the `-ATN` pin when this bit is asserted. Asserting this bit when the F82C5087 is in target mode, causes the F82C5087 to receive the `-ATN` signal (bit 7 of Register 5). However, the signal does not cause the F82C5087 to stop execution of the pipeline command.

Bit 1 Polarity of Interrupt Pin

When this bit is asserted, the polarity of the interrupt pin is high true.

Bit 0 Interrupt Tri-State Driver Enable

When this bit is set, the +INTERRUPT signal, pin 1, becomes tri-stateable.

Register 19 (13h) - Read Operation

This register provides information on the status of F82C5087 operation.

OP	7	6	5	4	3	2	1	0
R	MCS Data Rdy	FIFO Data Rdy	MSG Byte	Target Mode	Init Mode	Pipeline Empty	CMD MSB	CMD LSB

Bit 7 Message/Command Script (MCS) Data Ready

When this bit is asserted, the MCS Data Register is ready for more data. During normal operation, the firmware sets the MPU MCS Address Register to point to the starting location of an MCS script. Then the firmware makes consecutive reads or writes to the MPU MCS Buffer Register. The F82C5087 transfers data from the MPU MCS Register to the MCS script, then increments the MCS address pointer.

When this bit is asserted, it indicates data has been sent/received to/from the MCS. If the F82C5087 is not transferring data concurrently through the FIFO, the MPU MCS Buffer Register can be processed in approximately .3 microseconds. In this case, the firmware does not have to poll this bit before sending/receiving the next byte to/from the MPU MCS Buffer Register. When data transfers are occurring through the FIFO in parallel with the MPU accessing the MPU MCS Buffer Register, this bit must be polled before a new byte is transferred to/from the F82C5087.

Bit 6 FIFO Data Ready

This bit is asserted when the MPU is involved in data transfers (e.g., sending MPU data to the SCSI bus) and the FIFO is ready for more MPU data. Data transfers between the MPU and F82C5087 FIFO can be performed in any of the following three ways:

- The MPU polls this bit and transfers data to the MPU FIFO Data Register only when this bit is asserted.
- Enable the FIFO_RDY pin, then connect it to a DMA controller on the MPU bus. The DMA controller must send/receive data to/from the MPU FIFO Data Register.
- Enable IORDY pin (Mode Control Register 7's bit 0) and connect this pin to the MPU Wait State Generator. The MPU can open loop transfer the data to the MPU FIFO Data Register. If the register is not ready to transfer data, the F82C5087 deasserts the IORDY pin. This causes the MPU to generate wait states until the IORDY pin is asserted.

Bit 5 Message Byte

The F82C5087 tags all message bytes received from the SCSI bus. Before the firmware reads the MCS Data Register, this bit can be checked to determine if the next byte is a message. Bit 7 must be asserted for this bit to be valid. During SELECTION/RESELECTION, the F82C5087 also tags the Target ID. If the F82C5087 is reselected (-I_O signal is asserted), this bit is asserted. As mentioned previously, the firmware must read this bit before reading the MPU FIFO Data Register.

Note:

- When reading data from the MCS to the MPU MCS Buffer Register, the firmware must perform a dummy read of the MPU MCS Buffer Register. This transfers the first byte from the MCS to the MPU MCS Buffer Register, and also validates this bit. A dummy read is required because the F82C5087 must work with MPUs that do not contain a wait state generator.

Bit 4 Target Mode

When this bit is asserted, the F82C5087 is operating as a target. The target or initiator mode is normally set when a target or initiator command is issued to the F82C5087. For most commands the target or initiator role is maintained throughout the command. However, several pipeline commands can change the mode, depending on whether the F82C5087 is selected or reselected. For example, the following four commands change mode within the command itself because they are all dependent upon the selection/reselection process: (1) TGT WFS SEQ, (2) INIT WFR SEQ, (3) INIT WFR CMD, and (4) TGT WFS CMD.

Bit 3 Initiator Mode

When this bit is asserted, the F82C5087 is operating as an initiator.

Bit 2 Pipeline Register Empty

When this bit is asserted, the four pipeline command registers are empty.

Bits 1-0 Pipeline Command's Most and Least Significant Bits

These bits contain the number of the last pipeline command executed as shown in the following example. Assume the firmware places the following commands in the pipeline registers. When the execution of the first command begins, a zero is written to bit 1 and bit 0 as indicated as follows:

Pipeline Command	CMD MSB (Bit 1)	CMD LSB (Bit 0)	Pipeline Command Last Executed
CNTL ENB SEL	0	0	0
CNTL ENB RESEL	0	1	1
TGT WFS CMD	1	0	2 ←
	1	1	3

The control (CNTL) commands are executed successfully, but the pipeline command in Pipeline Register 2 is aborted. At this point, bits 1 and 0 contain the number of the last pipeline command executed. In this case, that would be TGT WFS CMD. Consequently, the firmware engineer can locate the exact point at which normal pipeline command execution was terminated.

If the F82C5087 is executing pipeline commands when these bits are read, the firmware should perform two reads of these bits and compare the results to ensure they are equal. This is required because the bits could be changing asynchronously with respect to the MPU.

Register 20 (14h)

Register Number	Operation	Register Name
20	Write Read	Auxiliary Control 2 Auxiliary Status 2

Register 20 (14h) - Write Operation

This register provides the firmware with additional control of F82C5087 operation.

OP	7	6	5	4	3	2	1	0
W	Toggle DMA	IO_RDY Qualify	Preset UP CNT	ENB UP CNT	INF TC	OVR DISC ENB	Mode Offset Load	HLT ON PE

Bit 7 Toggle DMA Controller

When the F82C5087 is interfaced to a DMA controller that is edge sensitive on the -REQ pulse (not level sensitive), a "hung" condition can develop. The DMA controller's transfer count can expire, yet the F82C5087 can still require more bytes to complete the transfer. The F82C5087 asserts the -DMA_REQ signal and the DMA controller does not respond with the -DMA_ACK signal. The firmware should load a transfer count into the DMA controller and reenale it. Then a write to this unlatched bit forces the -DMA_REQ signal to toggle. This produces an edge and restarts the DMA transfers.

Bit 6 IO Channel Ready Qualify

When this bit is set, IO_Channel_Ready is qualified with IO_Read and IO_Write. When not set, IO_Channel_Ready is bases on the address decode of the FIFO and MCS data registers.

Bit 5 Preset Up Counter

For test purposes, the Up Counter can be preset to 7FFFFFFH by asserting this unlatched bit. Asserting this bit also presets the Select Timeout Register to 7FFFFFFH.

Bit 4 Enable Up Counter

The F82C5087 contains two 24-bit transfer counters, an Up Counter and a Down Counter. The firmware loads the Down Counter by writing to Registers 16-18. Once a data transfer command is executing, the Down Counter is decremented as each data request is made. The Down Counter is under SM control.

The Up Counter counts bytes that have left the F82C5087 chip. The value of the Up Counter, FIFO Count, and Buffer Status should account for all transferred bytes. By asserting this bit, the firmware can read the Up Count. By deasserting this bit the firmware can read Registers 16-18 for the value of the Down Counter.

Bit 3 Infinite Transfer Count

Asserting this bit forces the transfer counter to an infinite value. This bit should be used carefully when the F82C5087 is operating in target

mode and receiving data from the SCSI bus as it does not know when to stop sending requests for more data.

Bit 2 Override Disconnect Enable

When this bit is asserted, the INIT SWA SEQ (Initiator Select with Attention Sequence) will override a disconnect if the identify message allows disconnect. The command will not disconnect from the SCSI bus.

Bit 1 Mode Offset Load

When this bit is asserted, a write to Register 14 sets the current synchronous offset count. Setting this count is useful for error recovery. When this bit is deasserted, a write to Register 14 sets the maximum synchronous offset allowed count.

Bit 0 Halt on Data Parity Error

When this bit is asserted and a parity error is detected while the SCSI bus is in data phase, the pipeline command is terminated. Parity errors which occur while data is being transferred to the MCS will always terminate the pipeline command. Extended Status bit 7 is also asserted. For more information, refer to the SCSI Data Parity Bit (Mode Control Register 7's bit 5).

Register 20 (14h) - Read Operation

This register provides additional status when bit 2 of the Interrupt Status Register is asserted. It also indicates when the Down Transfer Counter equals zero.

Before a new pipeline command is issued to the F82C5087, the firmware should clear the parity error and overrun status bits (defined below) by writing a one to bit 5 of Auxiliary Control 1 (Register 19). The firmware should also clear any external errors from the -WPAR pin.

OP	7	6	5	4	3	2	1	0
R	DN TC EQ 0	FIFO Parity Error	SCSI Parity Error	EXT Error	SYNC Overrun	OFF Overrun	MCS Overrun	MPU Overrun

Bit 7 Down Transfer Counter Equal to Zero

The 24-bit Down Transfer Counter is loaded by writing to Registers 16-18. This counter is equal to zero when this bit is asserted.

Bit 6 FIFO Parity Error

When data is being read from the F82C5087 FIFO and a parity error occurs, this bit is asserted. In this case, a byte with bad parity was transferred into the FIFO.

Bit 5 SCSI Parity Error

When a parity error is detected on data received from the SCSI interface, this bit is asserted. For more information on a halt on a parity

error, refer to this chapter's explanation of bit 5 of the Mode Control Register (Register 7).

Bit 4 Parity Error on Slave F82C5087

When a parity error or overrun condition is detected on a slave F82C5087 chip, the -WPAR pin is asserted which causes this bit to be asserted. The firmware should clear this condition before a new pipeline command is issued to the F82C5087.

Bit 3 Overrun During Synchronous Data Transfer

If synchronous SCSI bus data is being sent to the F82C5087 faster than 5.3 megabytes per second (33 MHz clock), an overrun occurs and this bit is asserted.

Bit 2 Offset Counter Overrun

This bit is asserted when the Offset Counter is requested to decrement past zero or to increment greater than the maximum count specified in Register 14. The counter does not perform the illegal request. The F82C5087 latches the up/down count requests to the Offset Counter. For example, if the Offset Counter is requested to increment past the maximum offset, the up request is saved and processed after a down request has decremented the count. This provides for one level of automatic error recovery.

Bit 1 Message/Command Script Overrun

If the IORDY pin is not enabled (Register 7's bit 0) and the MPU writes or reads to the MCS Data Buffer Register faster than the F82C5087 can transfer data to the MCS, this bit is asserted.

When the IORDY pin is enabled, an overrun/underrun cannot occur because the MPU is held off until the F82C5087 is ready to process the transfer.

Bit 0 Microprocessor Overrun

If the IORDY pin is not enabled and the MPU writes or reads to the MPU FIFO Data Register faster than the F82C5087 can transfer data to the FIFO, this bit is asserted.

Register 21 (15h)

Register Number	Operation	Register Name
21	WRITE READ	Synchronous/Asynchronous ID Current SCSI Device ID

Register 21 (15h)- Write Operation

This register informs the F82C5087 whether or not a SCSI device is set up for synchronous data transfers. If the bit is deasserted, asynchronous transfer is assumed.

When the F82C5087 selects/reselects a device or is selected/reselected itself, the ID of the SCSI device is saved and compared against this register's contents. If the firmware has enabled synchronous transfers for that device, the F82C5087 is placed in synchronous data transfer mode.

Because a F82C5087 operating in Slave Mode is unaware of bus phases or bus ID's, SID0 is the default ID for placing slave F82C5087 chips into synchronous data transfer mode.

OP	7	6	5	4	3	2	1	0
W	SID 7	SID 6	SID 5	SID 4	SID 3	SID 2	SID 1	SID 0

Notes:

- If the firmware asserts the SID bit for its own SCSI bus ID, then all SCSI devices are assumed to be in synchronous data transfer mode.
- If the SID bit for a SCSI device is asserted, the F82C5087 assumes all logical units associated with that SID bit are in synchronous mode.

Register 21 (15h) - Read Operation

This register contains the last SCSI bus device ID processed by the F82C5087. If the F82C5087 is connected to the SCSI bus as an initiator, then this register contains the ID of the target. If the F82C5087 is connected as a target, then this register contains the ID of the initiator.

OP	7	6	5	4	3	2	1	0
R	CID 7	CID 6	CID 5	CID 4	CID 3	CID 2	CID 1	CID 0

Notes:

- When the F82C5087 is executing a select target or reselect initiator sequence, the information in this register must have been previously placed as the first byte in the MCS.
- When the F82C5087 is selected/reselected and a wait for select/reselect sequences is executed, this information is also available as the first byte in the MCS.

Register 22 (16h)

Register Number	Operation	Register Name
22	WRITE READ	CMD Group 6 & 7 Length SCSI Data

Register 22 (16h) - Write Operation

The SCSI specification lists commands with group codes 6 and 7 as vendor unique. Consequently, this register is used to specify their length. By defining the length of commands with group codes 6 and 7, the F82C5087 is able to handle these SCSI commands without any firmware intervention. The firmware can specify the length of these commands as 2 - 17 bytes long.

OP	7	6	5	4	3	2	1	0
W	Length of CMD Group 7				Length of CMD Group 6			

Bits 7-4 Length of Command Group 7

These bits specify the length in bytes of SCSI commands with a group code of 7.

Bits 3-0 Length of Command Group 6

These bits specify the length in bytes of SCSI commands with a group code of 6.

Note: The F82C5087 determines the command length from the 4-bit value loaded into this register as follows:

Value	SCSI Command Length in Bytes
0000	Not defined
0001	2
0010	3
0011	4
0100	5
0101	6
0110	7
0111	8
1000	9
1001	10
1010	11
1011	12
1100	13
1101	14
1110	15
1111	16

Register 22 (16h) - Read Operation

A read of this register provides the firmware with the unlatched contents of the SCSI data bus. These contents are useful to the firmware for the following reasons:

- If the firmware wants to screen SCSI devices that select/reselect the F82C5087, the firmware engineer can wait until a select/reselect interrupt occurs, then check the device ID by reading this register. If communications with the device is desired, the firmware can issue a pipeline command to process the select/reselect.
- When the F82C5087 is executing an initiator pipeline command that aborted due to an illegal message on the SCSI bus, the firmware can read this register to determine which message is on the SCSI bus. Refer to Chapter 5's discussion of the control commands for further information on messages the F82C5087 chip considers illegal.

OP		7		6		5		4		3		2		1		0
R	Unlatch SCSI Data Bus															

Register 23 (17h)

Register Number	Operation	Register Name
23	WRITE/READ	Sync ACK Out counter

This register is used only for diagnostic and error recovery purposes. As an initiator, in synchronous mode, the F82C5087 may internally attempt to generate ACK pulses faster than they can actually be issued to the SCSI bus. This counter keeps track of the number of ACK pulses which are waiting to be issued. It may be set to any value for error recovery.

OP	7	6	5	4	3	2	1	0
W	Not used			Sync ACK Out counter				

Register 24 (18h)

Register Number	Operation	Register Name
24	WRITE/READ	Sync REQ counter

In synchronous mode, with the F82C5087 operating as an initiator, a target may REQ pulses for a data transfer before the F82C5087 has been programmed to execute the transfer. This counter keeps track of these pulses so that no data will be lost. This counter may be set for diagnostic or error recovery purposes; operation is not defined if it is set to a value greater than the maximum offset programmed in Register 14.

OP		7		6		5		4		3		2		1		0
W		Not used					Sync REQ counter									

Register 25 (19h)

Register Number	Operation	Register Name
25	WRITE READ	Auxiliary Control 3 Not Used

The Auxiliary Control 3 is a write-only register that controls the power management features and provides additional status information. Bits 7-3 are reserved.

Register 25 (19h) - Write Function

OP	7	6	5	4	3	2	1	0
W	Reserved					Enable LED	Enable Term. Power	Sleep Mode

Bit 2 Enable LED

This bit enables the BUFF_OUT pin to the value of the CONNECTED to the SCSI BUS bit, Register 4. It drives BUFF_OUT low when the chip is connected to the SCSI bus.

Bit 1 Enable Termination Power

When set, this bit causes the INP1[3] pin to reflect the state of the SLEEP BIT (this register, bit 0). In the case, INP1[3] pin becomes an output and is used to drive the termination circuitry. When INP1[3] is high, sleep mode is set and the termination is disabled. When INP1[3] is low, sleep mode is not set and the termination is enabled.

Bit 0 Sleep

This bit, when set turns off all internal clocks and allows minimum power consumption. Also, if the Enable Termination Power bit (this register, bit 1) is set, it controls the termination circuitry.

The Sleep bit is cleared by writing a zero to it, RESET_CAP, processor reset, or SCSI reset (dependent on the Disable SCSI Reset bit)/

Chapter 5

Commands

This chapter explains the F82C5087's command functions to implement the SCSI interface on a host computer and/or a device controller. An operational description of each F82C5087 command is provided.

Command Functions

Functions needed to implement the SCSI interface are provided by the F82C5087 pipeline commands. The 46 pipeline commands of the F82C5087 can be used to manage SCSI data, command, message, and status flow.

These pipeline commands can be classified into three categories:

- Control commands
- Initiator commands
- Target commands.

Table 1-1 provides a complete list of the 46 commands and a brief description of their function. SCSI bus conditions (e.g., enable a F82C5087 target device to respond to a SCSI bus -SEL signal) required before specific initiator and/or target commands (e.g., wait for select) can be issued through F82C5087 control commands.

F82C5087 control commands can also be used to establish a data path between the memory data bus and the MPU's data bus. For instance, the CNTL DATA MTP command (Control Data from Memory Data Bus to MPU) enables data to be transferred from the memory data bus to the MPU data bus.

Certain initiator and target commands perform single tasks (e.g., send a message to the SCSI bus), while others execute a series of tasks. For example, the INIT SWA SEQ (Initiator Select with Attention Sequence) performs eight tasks. Refer to this command sequence's description later in this chapter for a description of the eight tasks performed. Thus, sequences of tasks can be performed simply by issuing a single F82C5087 pipeline command sequence.

In this chapter's command descriptions, directional references (e.g., DATA OUT phase) are in reference to the initiator. For instance, when a target goes to the DATA OUT phase it indicates it is ready to receive data from the initiator.

Command Description

The commands are arranged into the control, initiator, and target groups. Within each of these groupings, commands are ordered sequentially according to command code. For each command the following information is provided: command code, bus condition, memory accessed, execution time, and operational description. Following is a description of the notations and conventions used within the chapter's command descriptions.

Command Code

As mentioned previously, up to four pipeline commands or command sequences can be queued into the F82C5087's pipeline registers (Registers 0-3). When performing a write operation to issue a command to a pipeline register, the following conventions must be used:

OP	7	6	5	4	3	2	1	0
W	Pipeline Command Code				INIT Mode	TARGET Mode	MOD Bit	INT

Bits 7-4 Pipeline Command Code

These bits specify the command code for the particular pipeline command the firmware wants to queue into the pipeline register. For instance, the control NOP command has a command code of 0000. For the command code of any pipeline command refer to the command's description.

Bits 3 and 2 Initiator and Target Mode

These bits specify whether the F82C5087 is operating in initiator or target mode, respectively. When bit 3 is asserted, the F82C5087 assumes the initiator role on the SCSI bus. Control commands do not have bit 3 or bit 2 asserted.

Bit 1 Modifier Bit

This bit is a command modifier bit. For example, commands that transfer data between the MPU and the F82C5087 use this bit to enable 16-bit transfers over the MPU bus. Bits 0-7 are transferred via the MPU's data bus. Simultaneously, bits 8-15 are transferred via the memory data bus.

Bit 0 Interrupt When COMMAND COMPLETE Bit

When this bit is asserted, and the firmware has enabled the interrupting condition (by enabling its bit in Interrupt Mask Register 5), the F82C5087 interrupts the MPU upon command completion. Refer to the NOP command description for an example of this bit's use.

Note:

When a bit is specified as "X" in a command code, it indicates that the bit can be functional in both asserted and deasserted forms.

Bus Condition

The F82C5087 can be disconnected or connected from the SCSI bus. For each of the commands the required SCSI bus condition prior to command execution is noted. "D" denotes a disconnected bus; "C" denotes a connected bus; and "C/D" indicates the command can be issued under either bus condition. Note that the control commands can be issued under either condition.

Execution Time

Stated execution time for the specific command represents a typical execution time (33 MHz operation is assumed). The symbol "us" is used as an abbreviation for microseconds.

Operational Description

A description of how the particular command operates is provided. On several of the initiator and target commands, there are notes concerning the set up of the MCS script. Before the firmware issues the F82C5087 a command, it must have initialized the MCS. For example, if the firmware wants the F82C5087 to send a message to a SCSI device that message must have been built in one of the six MCS scripts before the Send Message Command can be executed. After the firmware performs consecutive write operations to the MPU MCS Buffer Register (Register 11) to build the message in the MCS, it must load the SM MCS Address Register (Register 9) to point to the first byte of that message. After the MPU has completed this MCS set up, the Send Message Command could be issued to one of the four pipeline command registers. Note that if the auto MCS advance feature is enabled, the SM MCS address pointer is checked before command execution begins. If it points to any location within an MCS other than zero, it is advanced automatically to location zero of the next MCS. Therefore, in the previous example the firmware would not have to load the SM MCS Address Register if it were pointing to a location within the previous MCS.

Possible Errors

When the execution of a command can cause any error conditions, it is noted. The execution of the control commands cannot cause any error conditions. Error conditions noted (e.g., parity error) can be grouped into one of the following categories:

- Error conditions indicated by Extended Status Register 6.
- Error conditions indicated by Auxiliary Status Register 20.

When an unexpected condition is detected, the Extended Status Mask Bit (bit 1) of Interrupt Mask Register 5 is asserted. When this bit is asserted, it indicates Extended Status Register 6 contains information on why the command was terminated prematurely (e.g., SM error, timeout error).

When a parity error or overflow condition is detected, the Parity Error/Overflow Mask Bit (Bit 1) of Interrupt Mask Register 5 is asserted. When this bit is asserted, it indicates Auxiliary Status Register 20 contains additional

information (e.g., identifies error as a FIFO parity error or a SCSI parity error) concerning the detected error condition.

Control Command Descriptions

Table 5-1 lists the F82C5087 control commands. Following this table each control command is described in detail.

Table 5-1.

F82C5087 Control Commands

Command Code	Command	Function
0000 000X	CNTL NOP	No operation.
0001 000X	CNTL RST FIFO	Reset F82C5087 data FIFO.
0010 000X	CNTL RST ON	Assert reset signal to SCSI bus.
0011 000X	CNTL RST OFF	Deassert reset signal to SCSI bus.
0100 000X	CNTL ATN ON	Assert -ATN signal to SCSI bus.
0101 000X	CNTL ATN OFF	Deassert -ATN signal to SCSI bus.
0110 000X	CNTL ENB SEL	Enable selection.
0111 000X	CNTL DIS SEL	Disable selection.
1000 000X	CNTL ENB RESEL	Enable reselection.
1001 000X	CNTL DIS RESEL	Disable reselection.
1010 000X	CNTL ENB ADV	Enable MCS auto advance.
1011 000X	CNTL DIS ADV	Disable MCS auto advance.
1110 000X	CNTL DATA PTM	Receive MPU data and send it to the memory data bus.
1111 000X	CNTL DATA MTP	Receive data from memory data bus and send it to MPU.

CNTL NOP - No Operation

Command Code	Bus	Memory Access	Execution Time	Function
0000 000X	C/D	No	.5 us	No operation.

Operational Description

The firmware can use this command to check that the F82C5087 pipeline/status registers 0-3 are operating correctly. For example, the firmware can send the following four NOP commands to pipeline registers 0-3.

Command Pipeline Register	Contents of Command Pipeline Register
0	0000 0000
1	0000 0000
2	0000 0000
3	0000 0001

The first three NOP commands have bit 0 (Interrupt when Command Complete bit) deasserted; however, this bit is set in the fourth NOP command.

Consequently, this sequence of pipeline commands should terminate with an interrupt to the MPU under the following conditions:

- The Command Complete Interrupt Bit (bit 0) of Interrupt Mask Register 5 is asserted, and
- All four pipeline commands have executed.

Notes:

- Once this command has started executing, it cannot be soft aborted.
- The execution of this command cannot cause any error conditions.

CNTL RST FIFO - Control Reset Data FIFO

Command Code	Bus	Memory Access	Execution Time	Function
0001 000X	C/D	No	.5 us	Reset F82C5087's data FIFO.

Operational Description

Before executing any data transfer commands, the F82C5087 normally resets the F82C5087 data FIFO. For error handling and diagnostic purposes, the firmware may disable the automatic resetting of the FIFO. If it has been disabled by asserting the FIFO Reset Bit (bit 1) of Mode Control Register 7, the firmware must issue this command to reset the F82C5087's internal FIFO.

Notes:

- Once this command has started executing, it cannot be soft aborted.
- The execution of this command cannot cause any error conditions.

CNTL RST ON - Control RESET On

Command Code	Bus	Memory Access	Execution Time	Function
0010 000X	C/D	No	.5 us	Assert the reset signal to SCSI bus.

Operational Description

The firmware can use this command to assert the reset signal on the SCSI control bus. The F82C5087 pin asserted to indicate this reset is dependent on the F82C5087's mode of operation. In non-differential (single-ended drivers/receivers) mode, the -RESET_IN pin is asserted.

The reset signal remains asserted until the firmware issues the CNTL RST OFF command.

Notes:

- SCSI specification requires the reset signal be asserted for a minimum of 25 microseconds.
- Once this command has started executing, it cannot be soft aborted.
- The execution of this command cannot cause any error conditions.

CNTL RST OFF - Control RESET

Command Code	Bus	Memory Access	Execution Time	Function
0011 000X	C/D	No	.5 us	Deassert the reset signal to SCSI bus.

Operational Description

Reset on the SCSI bus is asserted by executing the CNTL RST ON command. To remove this condition, the firmware can issue this command to the pipeline registers.

CNTL ATN ON - Control Attention On

Command Code	Bus	Memory Access	Execution Time	Function
0100 000X	C/D	No	.5 us	Asserts -ATN signal to the SCSI bus.

Operational Description

This command asserts the -ATN signal on the SCSI bus. It remains asserted until the firmware issues a CNTL ATN OFF pipeline command, or until the last byte of a MESSAGE OUT phase is transferred.

When the following two initiator pipeline commands select a SCSI device, the -ATN signal is automatically asserted: (1) INIT SWA SEQ (Initiator Select with ATN Sequence) and (2) INIT SWA CMD (Initiator Select with ATN Command).

Notes:

- Automatic deassertion of -ATN is disabled during the transfer of the MESSAGE OUT phase's last byte if the INIT SEND MSG command's modifier bit is asserted.
- Once this command has started executing, it cannot be soft aborted.
- The execution of this command cannot cause any error conditions.

CNTL ATN OFF - Control Attention Off

Command Code	Bus	Memory Access	Execution Time	Function
0101 000X	C/D	No	.5 s	Deassert -ATN signal to SCSI bus.

Operational Description

The firmware can use this pipeline command to deassert the SCSI bus -ATN signal. It remains deasserted until the CNTL ATN ON pipeline command is issued, or an INIT SWA SEQ (Initiator Select with ATN Sequence) or INIT SWA CMD (Initiator Select with ATN Command) is executed.

Notes:

- Once this command has started executing, it cannot be soft aborted.
- The execution of this command cannot cause any error conditions.

CNTL ENB SEL - Control Enable Select

Command Code	Bus	Memory Access	Execution Time	Function
0110 000X	C/D	No	.5 us	Enable selection.

Operational Description

This command enables the F82C5087 controller to respond to a SCSI bus SELECT phase. When the F82C5087 device is selected, Select Bit (bit 5) of Interrupt Status Register 5 is asserted.

The firmware needs to execute this command in order for the TGT WFS CMD (Target Wait for Select Command) and the TGT WFS SEQ (Target Wait for Select then Receive Sequence) to detect a select.

Notes:

- By executing the CNTL DIS SEL (Control Disable Select) pipeline command, the F82C5087 is disabled from responding to selects.
- After a reset, the F82C5087 does not respond to selects until this command is executed.
- Once this command has started executing, it cannot be soft aborted.
- The execution of this command cannot cause any error conditions.

CNTL DIS SEL - Control Disable Select

Command Code	Bus	Memory Access	Execution Time	Function
0111 000X	C/D	No	.5 us	Disable selection.

Operational Description

The firmware can use this pipeline command to prevent the F82C5087 from responding to a SCSI bus SELECT phase. After this command is executed, a TGT WFS CMD (Target Wait for Select Command) and a TGT WFS SEQ (Target Wait for Select and Receive Sequence) will not respond to a -SEL signal. Also Interrupt Status Register 5's Select Bit (bit 5) will no longer be asserted when a select occurs.

Notes:

- By executing the CNTL ENB SEL command, the firmware engineer can enable the F82C5087 to respond to selects.
- After a reset, the F82C5087 does not respond to selects until the CNTL ENB SEL command is executed.
- Once execution has started, this command cannot be aborted.
- The execution of this command cannot cause any error conditions.

CNTL ENB RESEL - Control Enable Reselect

Command Code	Bus Access	Memory Access	Execution Time	Function
1000 000X	C/D	No	.5 us	Enable reselection.

Operational Description

This command enables the F82C5087 to respond to a SCSI bus RESELECT phase. When the F82C5087 initiator device is reselected, the Select Bit (bit 5) of Interrupt Status Register 5 is asserted.

The firmware needs to execute this command in order for INIT WFR CMD (Initiator Wait for Reselect Command) and INIT WFR SEQ (Initiator Wait for Reselect then Receive Sequence) to be able to detect a reselect.

Notes:

- By executing the CNTL DIS RESEL (Control Disable Reselect) pipeline command, the F82C5087 is disabled from responding to the reselect phase.
- After a reset, the F82C5087 does not respond to reselects until this command is executed.
- Once this command has started executing, it cannot be soft aborted.
- The execution of this command cannot cause any error conditions.

CNTL DIS RESEL - Control Disable Reselect

Command Code	Bus	Memory Access	Execution Time	Function
1001 000X	C/D	No	.5 us	Disable reselection.

Operational Description

The firmware can issue this command to the pipeline registers to prevent the F82C5087 from responding to a SCSI bus RESELECT phase. After this command is executed, INIT WFR CMD Initiator Wait for Reselect Command) and INIT WFR SEQ (Initiator Wait for Reselect and Receive Sequence) will not respond to a reselect phase. When a reselect occurs, the Select Bit (bit 5) of Interrupt Status Register 5 will no longer be asserted.

Notes:

- By executing the CNTL ENB RESEL (Control Enable Reselect) pipeline command, the firmware can enable the F82C5087 to respond to reselects.
- After a reset, the F82C5087 does not respond to reselects until a CNTL ENB RESEL command is executed.
- Once this command has started executing, it cannot be soft aborted.
- The execution of this command cannot cause any error conditions.

CNTL ENB ADV - Control Enable MCS Auto Advance

Command Code	Bus Access	Memory Access	Execution Time	Function
1010 000X	C/D	No	.5 us	Enable auto advance to next MCS script.

Operational Description

Normally the F82C5087 advances the SM address pointer to the beginning of the next MCS script at the start of a new pipeline command that involves MCS access. If this auto advance feature is disabled, all F82C5087 state machine (SM) accesses to the MCS will be to contiguous memory locations. The firmware can disable this MCS auto advance by executing a CNTL DIS ADV (Disable MCS Auto Advance) pipeline command.

Notes:

- Once execution has started, this command cannot be soft aborted.
- The execution of this command cannot cause any error conditions.

CNTL DIS ADV - Control Disable MCS Auto Advance

Command Code	Bus	Memory Access	Execution Time	Function
1011 000X	C/D	No	.5 us	Disable auto advance to next MCS script.

Operational Description

This command prevents the F82C5087 from automatically advancing the SM address pointer to the beginning of the next MCS script at the start of a new pipeline command that involves MCS access. The firmware should realize that when a command that will require MCS access is issued and the SM address pointer is already pointing to location zero of an MCS script, the pointer will not be automatically incremented at the start of command execution.

The firmware may decide to disable auto advance to make more efficient use of MCS space. For instance, the firmware may decide to place all commonly used messages contiguously in one MCS script. After the message has been placed in the MCS, the firmware must complete the following steps to send out that message:

- Execute the CNTL DIS ADV command to disable the MCS auto advance function.
- Load SM MCS Address Register 9 to point to the first byte of the message.
- Execute an INIT SEND MSG or TGT SEND MSG command.

Notes:

- Once execution has started, this command cannot be aborted.
- The execution of this command cannot cause any error conditions.

CNTL DATA PTM - Control Data from MPU to Memory Data Bus

Command Code	Bus	Memory Access	Execution Time	Function
1110 000X	C/D	FIFO	1 us plus .3 us/byte	Receive data from MPU bus and send it to memory data bus.

Operational Description

The firmware can use this command to provide a data path between the MPU data bus and the memory data bus. It transfers data from the MPU to the memory data bus. When the Down Transfer Counter equals zero and the FIFO is empty, this transfer is completed.

Notes:

- A soft abort of this command terminates command execution immediately.
- The only possible error with this command is a parity error.
- SCSI bus conditions are not pertinent to this command's execution.
- Refer to Figure 3-2 for an illustration of the MPU's use of MPU FIFO Buffer Register 12 and the F82C5087's internal FIFO to transfer data between these two buses.

CNTL DATA MTP - Control Data from Memory Data Bus to MPU

Command Code	Bus	Memory Access	Execution Time	Function
1111 000X	C/D	FIFO	1 us plus .3 us/byte	Receive data from memory data bus and send it to the MPU.

Operational Description

This command provides a data path between the MPU data bus and the memory data bus. The firmware can issue this command to transfer data from the memory data bus to the MPU bus. When the Down Transfer Counter equals zero and the FIFO is empty, the data transfer is completed.

Notes:

- A soft abort of this command terminates command execution immediately.
- SCSI bus conditions are not pertinent to this command's execution.
- The only possible error with this command is a parity error.

Initiator Command Descriptions

Refer to Table 5-2 for a list of the F82C5087 initiator commands. Following this table, each initiator pipeline command is described in detail.

Table 5-2.

Initiator Commands

Command Code	Command Initiator	Function
0000 100X	INIT S ^W OA SEQ	Select target without attention sequence.
0001 10XX	INIT SWA SEQ	Select target with attention sequence.
0010 100X	INIT S ^W OA CMD	Select target without attention command.
0011 100X	INIT SWA CMD	Select target with attention command.
0100 100X	INIT REC INFO	Receive information.
0101 100X	INIT DATA MTS	Receive data from memory data bus and send it to SCSI data bus.
0110 10XX	INIT DATA PTS	Receive data from MPU and send it to the SCSI data bus.
0111 100X	INIT DATA STM	Receive data from SCSI bus and send it to memory data bus.
1000 10XX	INIT DATA STP	Receive data from SCSI bus and send it to the MPU's data bus.
1001 10XX	INIT SEND MSG	Send message.
1010 100X	INIT SEND CMD	Send multi-byte command.
1011 100X	INIT XFER PAD	Transfer filler data.
1100 10XX	INIT XFER BYTE	Transfer a byte.
1101 100X	INIT CMDC SEQ	Perform COMMAND COMPLETE/DISCONNECT sequence.
1110 100X	INIT WFR CMD	Wait for the RESELECT phase.
1111 100X	INIT WFR SEQ	Wait to be reselected, then reconnect.

INIT SWOA SEQ - Initiator Select without Attention Sequence

Command Code	Bus	Memory Access	Execution Time	Function
0000 100X	D	MCS	6 us	Initiator select target without attention sequence.

Operational Description

This initiator command sequence performs the following tasks:

- Arbitrates for the SCSI bus.
- Selects a target.
- Sends a multi-byte command.

The firmware should issue this pipeline command sequence when the initiator does not want to send the target an identify message. The identify message would not be sent to the target when the system is operating in single-initiator mode.

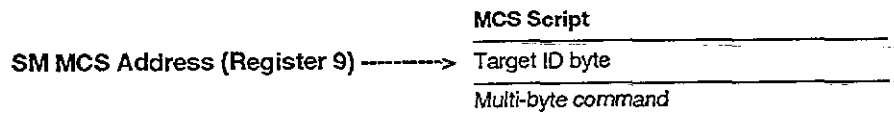
Possible Errors

The following errors can cause this command sequence to abort prematurely:

- A SELECT/RESELECT signal is detected during the SCSI bus ARBITRATION phase.
- A SELECT/RESELECT timeout occurs.
- An illegal SCSI bus phase is detected.
- SM error (illegal command group code was encountered) occurs.
- The F82C5087 is already connected to the SCSI bus when this command sequence is issued.
- The target disconnects before this command is completed.

Notes:

- A soft abort of this command sequence halts execution at one of the following points:
 - Before the SCSI bus ARBITRATION phase
 - Before the SCSI bus SELECTION phase
 - During the SCSI bus COMMAND phase.
- Before issuing this command sequence, the firmware engineer should ensure the MCS has been set up as follows:



INIT SWA SEQ - Initiator Select with Attention Sequence

Command Code	Bus	Memory Access	Execution Time	Function
0001 10XX	D	MCS	20 us	Initiator select target with attention sequence.

Operational Description

The firmware can use this initiator command sequence to perform the following series of tasks:

- Arbitrate for the SCSI bus.
- Select a target.
- Send a one-byte identify message to the target.
- Optionally send the target a two-byte tag message (if modifier bit equals 1).
- Send the target a multi-byte command.
- Terminate sequence at this point if:
 - the identify message disallows a disconnect, or
 - the Override Disconnect Enable Bit (bit 2 of Auxiliary Status Register 20) is asserted.
- If a disconnect is allowed, receive one of the following from the target:
 - one status byte and one message
 - one or two messages.
- Wait for the SCSI bus BUS FREE phase.

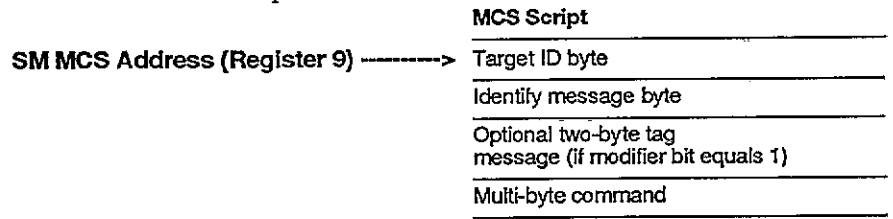
Possible Error

- This command sequence could be terminated early for any of the conditions listed under the following commands: INIT SWA CMD, INIT SEND MSG, INIT SEND CMD, and INIT CMDC SEQ.

Notes:

- A soft abort of this command halts execution at one of the following points:
 - At the beginning of the ARBITRATION phase, or
 - At the beginning of the SELECTION phase, or
 - After completion of the SELECTION phase.

- Before the firmware issues this command sequence, it should ensure the MCS has been set up as follows:



Any information received from the SCSI bus (e.g., a DISCONNECT message) will be placed in the MCS after the multi-byte command.

INIT SWOA CMD - Initiator Select without Attention Command

Command Code	Bus	Memory Access	Execution Time	Function
0010 100X	D	MCS	6 us	Initiator select target without attention command.

Operational Description

This initiator command arbitrates for the SCSI bus and selects a target. It is terminated when the target responds with the -BSY signal, or an error occurs.

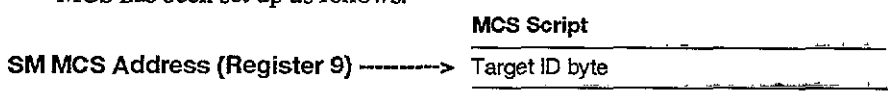
Possible Errors

The following errors are possible:

- A SELECT or RESELECT occurs during the ARBITRATION phase.
- A timeout occurs while the initiator is selecting the target.

Notes:

- A soft abort of this command sequence halts execution as follows: before the SCSI bus ARBITRATION phase, or before target SELECTION phase.
- Before issuing this initiator command, the firmware should ensure that the MCS has been set up as follows:



INIT SWA CMD - Initiator Select with Attention Command

Command Code	Bus	Memory Access	Execution Time	Function
0011 100X	D	MCS	6 us	Initiator select target with attention command.

Operational Description

This initiator command arbitrates for the SCSI bus and selects a target with the -ATN signal asserted. This command is terminated when the target responds with the -BSY signal, or an error occurs.

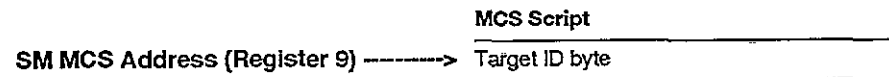
Possible Errors

The following errors are possible:

- A connect error occurs (e.g., F82C5087 is already connected to the SCSI bus when this command begins execution).
- A SELECT/RESELECT occurs during the ARBITRATION phase.
- A SELECT/RESELECT timeout occurs.

Notes:

- A soft abort of this command halts execution as follows: before the beginning of the SCSI bus ARBITRATION phase, or before the target SELECTION phase.
- Before the firmware issues this initiator command, it should ensure the MCS script has been set up as follows:



INIT REC INFO - Initiator Receive Information

Command Code	Bus	Memory Access	Execution Time	Function
0100 100X	C	MCS	2 us	Initiator receive information.

Operational Description

This initiator command waits for the target to go to MESSAGE IN phase or STATUS phase and then it receives the target's message or status into the MCS. At this point, the command is completed.

Possible Errors

The following errors can cause this pipeline command to abort prematurely:

- SM error occurs (if target goes to a SCSI bus phase other than the MESSAGE IN or STATUS phase).
- The initiator detects a wrong phase (if this command starts to process a multi-byte message and target unexpectedly changes the SCSI bus phase).

Notes:

- A long message can begin in one MCS and end in another MCS. If a message exceeds 65 bytes, the F82C5087 automatically transfers the first 65 bytes. After this initial transfer, the firmware engineer must handshake each byte by issuing the INIT XFER BYTE or TGT XFER BYTE command. When this condition develops, an error condition is not generated.
- A soft abort of this command terminates execution immediately.
- Once the command is successfully completed, the MCS script will contain the status byte or the message received from the target.

	MCS Script
SM MCS Address (Register 9) ----->	Status byte or message

INIT DATA MTS - Initiator Receive Data from Memory Data Bus to SCSI

Command Code	Bus	Memory Access	Execution Time	Function
0101 100X	C	FIFO	1 us plus .2 s/byte	Initiator receive data from the memory data bus and send it to the SCSI bus.

Operational Description

This initiator command waits for the target to go to DATA OUT phase then transfers data from the memory data bus to the SCSI bus. While the F82C5087 initiator is waiting for the target to change to DATA OUT phase, the F82C5087 starts to fill the internal FIFO with the data from the memory data bus.

When the Down Transfer Counter (set with the transfer count in Registers 16-18) equals zero and the FIFO is empty, this command is completed.

Possible Errors

The following errors can cause this command to abort prematurely:

- The target changes to a SCSI bus phase other than the DATA OUT phase, and asserts the -REQ signal; this causes a wrong phase error.
- The F82C5087 is not connected to the SCSI bus during this command's execution.

Notes:

- Because the F82C5087 is filling the FIFO while waiting for the target to switch to DATA OUT phase, the firmware must be prepared to backup the DMA controller pointers if the target changes to an illegal SCSI bus phase.
- A soft abort of this command terminates execution immediately.

INIT DATA PTS - Initiator Receive Data from MPU to SCSI Bus

Command Code	Bus	Memory Access	Execution Time	Function
0110 10XX	C	FIFO	1 us plus .2 us/data byte	Initiator receive data from MPU and send it to the SCSI bus.

Operational Description

This pipeline command waits for the target to go to the DATA OUT phase then transfers data from the MPU to the SCSI bus. SCSI bus data can be asynchronous or synchronous. Refer to Chapter 4's description of Sync/Async ID Register 21 for information on how synchronous or asynchronous transfer mode is specified.

While the F82C5087 is waiting for the target to change to the DATA OUT phase, the F82C5087 starts to fill the FIFO with data from the MPU. MPU data transfers can be 8 or 16 bits wide. If the modifier bit (bit 1) of this command code is set, 16-bit transfers are enabled. With 16-bit MPU data transfers, bits 0-7 are transferred via the MPU bus; bits 8-15 are transferred simultaneously via the memory data bus. Data transfers from the MPU to the F82C5087 must be written to MPU FIFO Data Register 12. For more details refer to Register 12's description in Chapter 4.

When the Down Transfer Counter (set with the transfer count in Registers 16-18) equals zero and the FIFO is empty, this command is completed.

Possible Errors

The following errors can cause this command to abort prematurely:

- The target changes to a SCSI bus phase other than the DATA OUT phase, and asserts the -REQ signal; this causes a wrong phase error.
- The F82C5087 is not connected to the SCSI bus during this command's execution.

Notes:

- Because the F82C5087 is filling the FIFO while waiting for the target to switch to DATA OUT phase, the firmware must be prepared to backup the DMA controller pointers if the target changes to an illegal SCSI bus phase.
- A soft abort of this command terminates execution immediately.

INIT DATA STM - Initiator Receive Data from SCSI to Memory Data Bus

Command Code	Bus	Memory Access	Execution Time	Function
0111 100X	C	FIFO	1 us plus .2 us/byte	Initiator receive data from SCSI bus and send it to the memory data bus.

Operational Description

This initiator command waits for the target to switch to the DATA IN phase then transfers data from the SCSI data bus to the memory data bus. SCSI bus data can be asynchronous or synchronous. Refer to Chapter 4's description of Sync/Async ID Register 21 for information on how synchronous or asynchronous transfer mode is specified.

When the Down Transfer Counter equals zero and the FIFO is empty, this command is completed.

Possible Errors

The following errors can cause this command to abort prematurely:

- The target changes to a SCSI bus phase other than the DATA IN phase, and asserts the -REQ signal; this causes a wrong phase error.
- The F82C5087 is not connected to the SCSI bus during this command's execution.

Note:

- If the firmware soft aborts this command, command execution is terminated immediately.

INIT DATA STP - Initiator Receive Data from SCSI to MPU Bus

Command Code	Bus	Memory Access	Execution Time	Function
1000 10XX	C	FIFO	1 us plus .2 us/byte	Initiator receive data from the SCSI bus and send it to the MPU's data bus.

Operational Description

This initiator command waits for the SCSI bus to change to the DATA IN phase then transfers asynchronous or synchronous data from the SCSI data bus to the MPU. The MPU must receive the data by reading the MPU FIFO Buffer Register 12. The MPU can read 8 or 16 bits per cycle. If the modifier bit (bit 1) of this command code is asserted, 16-bit transfers are enabled.

When the Down Transfer Counter equals zero and the FIFO is empty, this command is completed.

Possible Errors

The following errors can cause this command to abort prematurely:

- The target changes to a SCSI bus phase other than the DATA IN phase, and asserts the -REQ signal; this causes a wrong phase error.
- The F82C5087 is not connected to the SCSI bus during this command's execution.

Note:

- If the firmware soft aborts this command, execution terminates immediately.

INIT SEND MSG - Initiator Send Message

Command Code	Bus	Memory Access	Execution Time	Function
1001 10XX	C	MCS	1 us plus .5 us/byte	Initiator send message to target.

Operational Description

This pipeline command has the initiator wait for the target to go to the MESSAGE OUT phase then transfers a single or multi-byte message from the MCS to the target. If the F82C5087 does not have the -ATN signal asserted, this command does the following:

- Asserts the -ATN signal before transferring the first byte, and
- Deasserts -ATN on the last message byte. However, if the modifier bit (bit 1) of this command code is asserted, -ATN is not deasserted on the last message byte. This gives the firmware the opportunity to send multiple messages to the target.

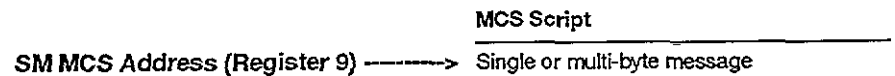
Possible Errors

The following errors can cause this initiator command to abort prematurely:

- A connect error occurs (if the target disconnects during execution of this pipeline command).
- The initiator detected a wrong SCSI bus phase (e.g., target goes to phase other than the MESSAGE OUT phase).

Notes:

- A long message can begin in one MCS and end in another MCS. If a message exceeds 65 bytes, the F82C5087 automatically transfers the first 65 bytes. After this initial transfer, the firmware engineer must handshake each byte by issuing the INIT XFER BYTE or TGT XFER BYTE command. When this condition develops, an error condition is not generated.
- A soft abort of this command terminates command execution immediately.
- Before the firmware issue this command, it should ensure the MCS has been set-up as follows:



INIT SEND CMD - Initiator Send Command

Command Code	Bus	Memory Access	Execution Time	Function
1010 100X	C	MCS	1 us plus .5 us/byte	Initiator send a multi-byte command.

Operational Description

This initiator command waits for the target to go to the SCSI bus COMMAND phase then transfers a multi-byte command from the MCS to the target. Bits 7-5 of the first pipeline command byte are decoded by the F82C5087 to determine the command length. Refer to Table 5-3 for list of command length codes.

Table 5-3.

Command Length Codes

Command Byte Bits			Command Length
7	6	5	
0	0	0	6 bytes
0	0	1	10 bytes
0	1	0	SM (state machine) error
0	1	1	SM error
1	0	0	SM error
1	0	1	12 bytes
1	1	0	Register 22's bits 0-3
1	1	1	Register 22's bits 4-7

Refer to the operational description of the Target Receive Multi-byte Command (TGT REC CMD) later in this chapter for information on command lengths that cause SM errors. For details on command length specified by Register 22's bits refer to that register's description in Chapter 4.

Possible Errors

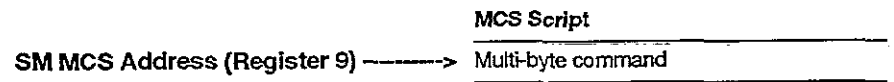
The following errors are possible with the execution of this command:

- A connect error occurs.
- The initiator detects a wrong SCSI bus phase; the target changed to a phase other than the COMMAND phase.
- SM error (illegal group code) occurs. Byte 0, bits 7-5, contains a group code of 2, 3, or 4. If the firmware wants to transfer a command with one of these group codes to the target, it must back up the SM MCS address pointer one location, and issue the INIT XFER BYTE command for each byte sent to the target.

Notes:

- A soft abort of this command terminates execution immediately.

- Before the firmware issues this command, it should ensure the MCS has been set up as follows:



INIT XFER PAD - Initiator Transfer Pad

Command Code	Bus	Memory Access	Execution Time	Function
1011 100X	C	No	1 us plus .5u s/byte	Initiator transfer filler data.

Operational Description

This transfer pad command enables the initiator either to receive or send data, depending on the state of the SCSI bus $-I_O$ signal. Data is transferred until the target changes SCSI bus phase. If this command is receiving data, the data is discarded; parity is not checked. If this command is sending data, it sends a zero value without a parity error.

The Transfer Counter that is set with Registers 16-18 is not effected by this command. This command cannot be used to pad synchronous data bytes. The firmware must do an actual data transfer count when synchronous data padding is required. Except for the $-I_O$ signal, SCSI bus phase is not pertinent to this command's execution; only a phase change is pertinent. This command does not set any bits in the pipeline status registers.

Possible Error

The following error can cause this command to abort prematurely:

- A connect error occurs. The F82C5087 is not connected to the SCSI bus during this command's execution.

Note:

- A soft abort of this command terminates execution immediately.

INIT XFER BYTE - Initiator Transfer Byte

Command Code	Bus	Memory Access	Execution Time	Function
1100 10XX	C	MCS	1 us plus .5 us/byte	Initiator transfer a byte.

Operational Description

This initiator command transfers a byte from the SCSI bus to the MCS, or transfers the next MCS byte to the SCSI bus. If this command's modifier bit is asserted, this command asserts the -ATN signal while the -REQ signal is true on the SCSI bus. This protocol ensures that the target detects the asserted -ATN signal.

This protocol with the modifier bit asserted is normally used to request the target to perform a MESSAGE OUT phase (possibly to reject a message, etc). Except for the -I_O signal, SCSI bus phase is not pertinent to this command. This command does not set any bits in the pipeline status registers.

Possible Error

The following error is possible:

- The F82C5087 is not connected to the SCSI bus during this command's execution.

Notes:

- A soft abort of this command terminates execution immediately.
- The firmware should ensure that the SM address pointer (Register 9) points to the location in the MCS where the bytes should be sent/received. Generally, the firmware will want to disable the MCS automatic advance feature before executing this command. This feature can be disabled by issuing the CNTL DIS ADV pipeline command to one of the pipeline registers.

INIT CMDC SEQ - Initiator COMMAND COMPLETE/DISCONNECT Sequence

Command Code	Bus	Memory Access	Execution Time	Function
1101 100X	C	MCS	4 us	Initiator COMMAND COMPLETE /DISCONNECT sequence.

Operational Description

This initiator command sequence receives one status byte and one message, or two messages; and places the information in the MCS. If one of the messages is the DISCONNECT message or COMMAND COMPLETE message, this command sequence waits for the BUS FREE phase to occur before it is completed.

A state machine (SM) error is generated and the sequence is terminated if the F82C5087 receives a message other than one of the following: (1) DISCONNECT message (2) COMMAND COMPLETE message (3) LINKED COMMAND COMPLETE (4) LINKED COMMAND COMPLETE with flags (5) SAVE DATA POINTERS (6) RESTORE DATA POINTERS or (7) the identify message.

Possible Errors

The following errors can cause this command sequence to abort prematurely:

- A connect error occurs.
- Any of the following four SM errors occur:
 - **ILLEGAL PHASE.** The target changes the SCSI bus phase to a phase other than the MESSAGE IN or STATUS phase.
 - **ILLEGAL MESSAGE.** The message byte is not one of the seven legal messages (e.g., DISCONNECT message) listed in this command's "Operational Description" section.
 - **ILLEGAL DISCONNECT.** The target disconnects without first sending a COMMAND COMPLETE or DISCONNECT message to the initiator.
 - The target asserts the -REQ signal when it should have disconnected.

Notes:

- A soft abort of this command sequence terminates execution immediately.
- This initiator command sequence will not receive illegal messages; they are left on the SCSI bus. The firmware can examine the contents of the SCSI bus by reading SCSI Data Register 22.
- When this command is completed successfully, the MCS will contain one status and one message byte, or two message bytes. The SM MCS address pointer (Register 9) will be pointing to the last byte received, plus one byte.

INIT WFR CMD - Initiator Wait for Reselect

Command Code	Bus	Memory Access	Execution Time	Function
1110 100X	D	MCS	1.5 us plus wait time	Initiator wait for reselect.

Operational Description

The firmware needs to execute the CNTL ENB RESEL command for the INIT WRF CMD and INIT WFR SEQ to be able to detect a reselect. The INIT WRF CMD waits for the F82C5087 to be selected or reselected, then it saves the ID of the device that selected or reselected the F82C5087 in the MCS. The command asserts the -BSY signal and waits for the -SEL signal to be deasserted.

This command can also change the initiator or target mode of the F82C5087. If the F82C5087 is reselected, the initiator mode is maintained; however, the F82C5087 is switched to target mode if it is selected. If a reselection occurs, this command deasserts the -BSY signal after the -SEL signal is deasserted.

The F82C5087 tags the ID byte to indicate whether a select or reselect occurred. When the MPU MCS Buffer Register contains the ID byte, the firmware should read Register 19's bit 6 before reading the MPU MCS Buffer Register. If bit 6 of Auxiliary Status Register 19 is asserted, a reselection occurred.

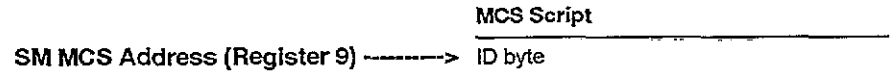
Possible Error

The following error can cause this command to abort prematurely:

- A connect error occurs. The F82C5087 is connected to the SCSI bus during this command's execution.

Notes:

- A soft abort of this command terminates execution before a select or reselect occurs.
- After this command is successfully executed, the ID of the device that selected or reselected the F82C5087 will have been received from the SCSI bus and placed in the MCS:



INIT WFR SEQ - Initiator Wait for Reselect Sequence

Command Code	Bus	Memory Access	Execution Time	Function
1111 100X	D	MCS	4 us	Initiator wait for RECONNECT condition then begin RECONNECT sequence.

Operational Description

If the F82C5087 is reselected and the firmware has specified this command, this command sequence is executed. If the F82C5087 is selected, it switches to target mode and executes the Target Wait for Select Sequence (TGT WFS SEQ).

When this sequence is completed, the firmware should read Auxiliary Status Register 19 to check if the F82C5087 is in target or initiator mode.

If the reconnected target is set up for synchronous data transfers, this command sequence is terminated after the device ID is saved in the MCS. For details on setting up devices for synchronous data transfers refer to Sync/Async ID Register 21's description in Chapter 4.

If the reconnected target is set up for asynchronous data transfers, this command sequence continues to receive status and the following legal messages: (1) DISCONNECT message (2) COMMAND COMPLETE message (3) LINKED COMMAND COMPLETE (4) LINKED COMMAND COMPLETE with flags (5) SAVE DATA POINTERS (6) RESTORE DATA POINTERS, or (7) the identify message. It continues to receive status and/or these legal messages until the target changes to a SCSI bus phase other than the STATUS or MESSAGE IN phase.

The F82C5087 tags the target ID in the MCS to indicate the occurrence of a reselection. When the MPU MCS Buffer Register contains the ID byte, the firmware should read Register 19's bit 5 before reading the MPU MCS Buffer Register. If bit 5 of Auxiliary Status Register 19 is asserted, a reselection occurred.

Possible Errors

The following errors are possible:

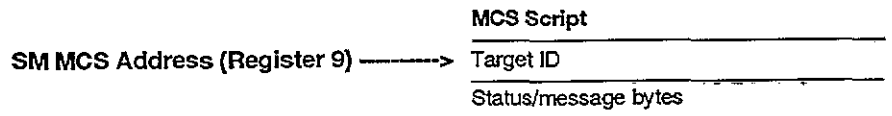
- A connect/disconnect error occurs. The F82C5087 is connected when sequence is started, or an unexpected disconnect occurs.
- An SM error occurs. The target sends an illegal message. The message byte is not one of the seven legal messages listed in this command's "Operational Description" section.

Notes:

- Many times the firmware will not know if the next SCSI bus access will be a selection or reselection. Because the F82C5087 will execute the appropriate command (INIT WFR SEQ or TGT WFS SEQ) regardless of which sequence was specified, the firmware can use these two pipeline commands interchangeably. However, it is recommended that target/initiator mode be maintained whenever possible. For example, if the

F82C5087 is operating as an initiator, the firmware should issue this command to one of the pipeline registers.

- A soft abort of this sequence terminates execution before the reselect occurs. After the F82C5087 has been reselected, the command will run to completion and ignore the soft abort.
- This initiator sequence will not receive illegal messages; they are left on the SCSI bus for the firmware to handle. The firmware may want to receive the message by issuing an INIT REC INFO command, or may want to begin a message reject process by asserting the -ATN signal.
- After this command is successfully completed, normally the MCS will contain the target ID followed by the status and message bytes.



Target Command Descriptions

Refer to Table 5-4 for a list of the F82C5087 target commands. Following this table each command is described in detail.

Table 5-4.

F82C5087 Target Commands

Command Code	Command	Target Function
0000 010X	TGT SEND MSG	Send message.
0001 010X	TGT SEND STAT	Send status to initiator.
0010 010X	TGT DATA MTS	Receive data from memory data bus and send it to the SCSI bus.
0011 01XX	TGT DATA PTS	Receive data from MPU bus and send it to SCSI bus.
0100 010X	TGT REC CMD	Receive a multi-byte command.
0101 010X	TGT REC MSG	Receive a single-byte or multi-byte message.
0110 010X	TGT DATA STM	Receive data from SCSI bus and send it to the memory data bus.
0111 01XX	TGT DATA STP	Receive data from SCSI bus and send it to the MPU's data bus.
1000 010X	TGT CMDC SEQ	Perform COMMAND COMPLETE sequence.
1001 010X	TGT XFER BYTE	Transfer byte.
1010 010X	TGT DISC CMD	Disconnect from the SCSI bus.
1011 010X	TGT DISC SEQ	Perform DISCONNECT sequence.
1100 01XX	TGT RCNT SEQ	Perform RECONNECT sequence.
1101 01XX	TGT SEND BSY	Wait for select, then send busy response and disconnect.
1110 010X	TGT WFS CMD	Wait for select.
1111 01XX	TGT WFS SEQ	Perform Wait for Select sequence.

TGT SEND MSG - Target Send Message

Command Code	Bus	Memory Access	Execution Time	Function
0000 010X	C	MCS	2u s first byte plus 1 us/additional byte	Target send single or multi-byte message.

Operational Description

This pipeline command switches the SCSI bus to the MESSAGE IN phase, and sends a single or multi-byte message to the initiator.

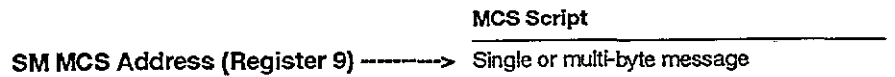
Possible Error

The following errors are possible when this command is executing:

- The initiator asserts the -ATN signal unexpectedly.
- Not connected to SCSI when command is issued.

Notes:

- A soft abort of this command terminates execution immediately.
- A long message can begin in one MCS and end in another MCS. If a message exceeds 65 bytes, the F82C5087 automatically transfers the first 65 bytes. After this initial transfer, the firmware must handshake each byte by issuing the TGT XFER BYTE command. When this condition develops, an *error condition is not generated*.
- Before the firmware issues this target command, it should ensure the MCS script has been set up as follows:



TGT SEND STAT - Target Send Status

Command Code	Bus	Memory Access	Execution Time	Function
0001 010X	C	MCS	2 us	Target send status.

Operational Description

This target command switches the SCSI bus to the STATUS phase, and transfers one status byte from the MCS to the initiator.

Possible Errors

The following errors can cause this target command to abort prematurely:

- A connect error occurs. The F82C5087 is not connected to the SCSI bus when this command is issued.
- The initiator asserts the -ATN signal.

Notes:

- A soft abort of this command terminates execution immediately.
- Before the firmware issues this target command, it should ensure the MCS has been set up as follows:

SM MCS Address (Register 9)	MCS Script
----->	Status byte

TGT DATA MTS - Target Receive Data from Memory Data Bus to SCSI Bus

Command Code	Bus	Memory Access	Execution Time	Function
0010 010X	C	FIFO	1 us plus .2 us/byte	Target receive data from memory data bus and send it to SCSI data bus.

Operational Description

This pipeline command switches the SCSI bus to the DATA IN phase, and transfers data from the memory data bus to the SCSI bus. Data transfers over the SCSI bus can be synchronous or asynchronous. Refer to the description of the Sync/Async ID Register (Register 21) in Chapter 4 for information on how synchronous or asynchronous mode for data transfers is specified.

When the Down Transfer Counter (set with the transfer count in Registers 16-18) equals zero and the FIFO is empty, this command is completed.

Possible Errors

The following errors can cause this command to abort prematurely:

- The F82C5087 is not connected to the SCSI bus during this command's execution.
- The initiator asserts the -ATN signal while this command is executing, and the -ATN pin is not disabled. The -ATN pin is disabled by asserting the Disable ATN Bit (bit 2) of Auxiliary Control Register 19.

Note:

- A soft abort of this command terminates execution immediately.

TGT DATA PTS - Target Receive Data from MPU to SCSI Bus

Command Code	Bus	Memory Access	Execution Time	Function
0011 01XX	C	FIFO	1u s plus .2u s/byte	Target receive data from MPU bus and send it to SCSI bus.

Operational Description

This pipeline command switches the SCSI bus to the DATA IN phase and transfers 8 or 16 bits of data from the MPU bus to the SCSI bus. SCSI bus data transfers can be asynchronous or synchronous. Refer to Chapter 4's description of Sync/Async ID Register 21 for information on how synchronous or asynchronous data transfers are specified.

MPU data transfers can be 8 or 16 bits wide. If the modifier bit (bit 1) of this command is asserted, 16-bit transfers are enabled. With 16-bit MPU data transfers, bits 0-7 are transferred from the MPU bus to MPU FIFO Data Register 12. Bits 8-15 are simultaneously transferred over the memory data bus. For more details on Register 12 refer to Chapter 4.

When the Down Transfer Counter (set with the transfer count in Registers 16-18) equals zero and the FIFO is empty, this command is completed.

Possible Errors

The following errors can cause this command to abort prematurely:

- The F82C5087 is not connected to the SCSI bus when this command is executing.
- The initiator asserts the -ATN signal while this command is executing, and the -ATN pin is not disabled. The -ATN pin is disabled by asserting the Disable ATN Bit (bit 2) of Auxiliary Control Register 19.

Note:

- A soft abort of this command terminates execution immediately.

TGT REC CMD - Target Receive Command

Command Code	Bus	Memory Access	Execution Time	Function
0100 010X	C	MCS	1 us plus .5 us/byte	Target receive a multi-byte command.

Operational Description

This pipeline command switches the SCSI bus to the COMMAND phase and transfers a multi-byte command from the SCSI bus to the MCS.

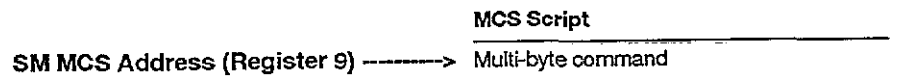
Possible Errors

The following errors can cause this target command to abort prematurely:

- The F82C5087 is not connected to the SCSI bus when this command is executed.
- The initiator asserts the -ATN signal.
- An SM error (illegal group code) occurs. A command with a group code of 2, 3, or 4 was received. No command bytes have been transferred into the MCS. The firmware must issue the TGT XFER BYTE command to receive each command byte into the MCS.

Note:

- A soft abort of this command terminates execution immediately.
- After this command is successfully executed, the MCS will contain a multi-byte command:



TGT REC MSG - Target Receive Message

Command Code	Bus	Memory Access	Execution Time	Function
0101 010X	C	MCS	1 us plus .5 us/byte	Target receive single or multi-byte message.

Operational Description

This command switches the SCSI bus to the MESSAGE OUT phase and receives a single or multi-byte message into the MCS.

Possible Errors

The following errors are possible:

- The F82C5087 is not connected to the SCSI bus when this command is executed.

Notes:

- A long message can begin in one MCS and end in another MCS. If a message exceeds 65 bytes, the F82C5087 automatically transfers the first 65 bytes. After this initial transfer, the firmware must handshake each byte by issuing the TGT XFER BYTE command. When this condition develops, an error condition is not generated.
- A soft abort of this command terminates execution immediately.
- After this command is successfully executed, the MCS will contain the following:

	<u>MCS Script</u>
SM MCS Address (Register 9) ----->	Single or multi-byte message

TGT DATA STM - Target Receive Data from SCSI Bus to Memory Data Bus

Command Code	Bus	Memory Access	Execution Time	Function
0110 010X	C	FIFO	1 us plus .2 us/byte	Target receive data from the SCSI bus and send to memory data bus.

Operational Description

This target command switches the SCSI bus to DATA OUT phase, receives data from the SCSI bus, and sends it to the memory data bus. Data transfers over the SCSI bus can be synchronous or asynchronous. Refer to Chapter 4's description of Sync/Async ID Register 21 for information on how synchronous or asynchronous data transfer mode is specified.

When the Down Transfer Counter (set with the transfer count in Registers 16-18) equals zero and the FIFO is empty, this command is completed.

Possible Errors

The following errors can cause this command to abort prematurely:

- The F82C5087 is not connected to the SCSI bus when this command is executing.
- The initiator asserts the -ATN signal while this command is executing, and the -ATN pin is not disabled. The -ATN pin is disabled by asserting the Disable ATN Bit (bit 2) of Auxiliary Control Register 19.

Note:

- A soft abort will terminate execution of this command immediately.

TGT DATA STP - Target Receive Data from SCSI to MPU Bus

Command Code	Bus	Memory Access	Execution Time	Function
0111 01XX	C	FIFO	1 μ s plus .2 us/byte	Target receive data from SCSI bus and send it to memory data bus.

Operational Description

This target command switches the SCSI bus to the DATA OUT phase and transfers data from the SCSI data bus to the MPU data bus. Data transferred can be synchronous or asynchronous. Refer to Chapter 4's description of Sync/Async ID Register 21 for details on how synchronous or asynchronous transfer mode can be specified.

If this command's modifier bit (bit 1) is asserted, 16 bits of data will be transferred to the MPU when MPU FIFO Data Register 12 is read. Bits 0-7 are transferred over the MPU data bus while bits 8-15 are being transferred simultaneously over the memory data bus.

Possible Errors

The following errors can cause this command to abort prematurely:

- The F82C5087 is not connected to the SCSI bus when this command is executing.
- The initiator asserts the -ATN signal while this command is executing, and the -ATN pin is not disabled. The -ATN pin is disabled by asserting the Disable ATN Bit (bit 2) of Auxiliary Control Register 19.

Note:

- A soft abort of this command terminates execution immediately.

TGT CMDC SEQ - Target COMMAND COMPLETE Sequence

Command Code	Bus	Memory Access	Execution Time	Function
1000 010X	C	MCS	3 us	Target perform COMMAND COMPLETE sequence.

Operational Description

This target command sequence switches the SCSI bus to the STATUS phase and transfers one status byte from the MCS to the initiator. This sequence then switches the SCSI bus to the MESSAGE IN phase and transfers a single or multi-byte message from the MCS to the initiator.

If the transferred message was a COMMAND COMPLETE, the F82C5087 goes to the SCSI bus phase BUS FREE. Otherwise, the F82C5087 remains connected to the SCSI bus.

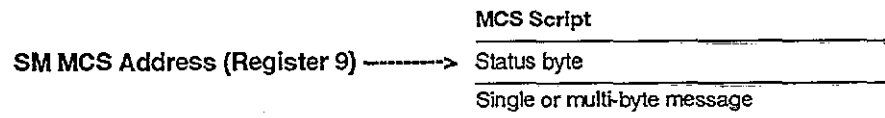
Possible Errors

The following errors can cause this sequence to abort prematurely:

- The F82C5087 is not connected to the SCSI bus when this command sequence is executed.
- The initiator asserted an -ATN signal unexpectedly.

Notes:

- When this sequence is soft aborted, command execution is terminated immediately.
- Before issuing this target command sequence, the firmware should ensure that the MCS has been set up as follows:



TGT XFER BYTE - Target Transfer Byte

Command Code	Bus	Memory Access	Execution Time	Function
1001 010X	C	MCS	1.5 us	Target transfer byte.

Operational Description

This command transfers a byte from the MCS to the SCSI bus, or from the SCSI bus to the MCS. It is normally used to perform clean-up when a command sequence has failed. For example, if TGT REC CMD is executing and an illegal group code caused it to abort prematurely, the firmware engineer can receive the SCSI command by executing TGT XFER BYTE. This pipeline command needs to be executed one time for each byte received into the MCS.

Except for the -I_O signal, SCSI bus phase is not pertinent to this command. It will not change the phase of the SCSI bus, or set any bits in the pipeline status registers.

Possible Error

The following error can cause this command to abort prematurely:

- A connect error occurs. The F82C5087 is not connected to the SCSI bus when this command started to execute.

Notes:

- A soft abort terminates the execution of this command immediately.
- Parity checking is performed by this command.
- Before issuing this command, the firmware should ensure that the MCS has been set up as follows:

SM MCS Address (Register 9)	MCS Script
Write Operation ----->	One byte transfer to SCSI bus or one byte received from SCSI bus.
Read Operation ----->	

TGT DISC CMD - Target Disconnect Command

Command Code	Bus	Memory Access	Execution Time	Function
1010 010X	C	N/A.	5 us	Target disconnect from SCSI bus.

Operational Description

This command disconnects the F82C5087 from the SCSI bus. The SCSI bus is allowed to go to the BUS FREE phase.

Note:

- Once this command has started executing, it cannot be soft aborted.

TGT DISC SEQ - Target DISCONNECT Sequence

Command Code	Bus	Memory Access	Execution Time	Function
1011 010X	C	MCS	1 us plus .5 us/byte	Target perform DISCONNECT sequence.

Operational Description

This sequence changes the SCSI bus to the MESSAGE IN phase, and transfers message(s) to the initiator until the DISCONNECT message is received. Once the DISCONNECT message is transferred to the initiator, this sequence will set the SCSI bus to the BUS FREE phase and terminate execution. The firmware engineer should exercise care when using this sequence because an infinite loop is possible if the F82C5087 does not find a DISCONNECT message.

Possible Error

The following error can cause this command to abort prematurely:

- The F82C5087 is not connected to the SCSI bus when this command is executed.

Notes:

- A soft abort terminates the execution of this command immediately.
- Before issuing this command, the firmware should ensure that the MCS has been set up as follows:

MCS Script	
SM MCS Address (Register 9) ----->	Multiple, single-byte messages or multi-byte messages
	DISCONNECT message

TGT RCNT SEQ - Target Reconnect Sequence

Command Code	Bus	Memory Access	Execution Time	Function
1100 01XX	D	MCS	6 us	Target perform RECONNECT sequence.

Operational Description

This sequence arbitrates for the SCSI bus and reselects an initiator. If the modifier bit (bit 1) is set, this sequence changes the SCSI bus to the MESSAGE IN phase, and sends a single or multi-byte message to the initiator.

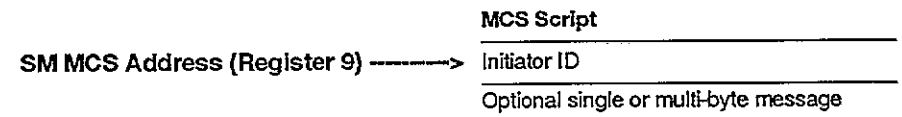
Possible Errors

The following errors can cause this command to abort prematurely:

- A select or reselect occurs during the ARBITRATION phase.
- A timeout occurs while reselecting the initiator.
- A connect error occurs. The F82C5087 is connected to the SCSI bus when this command is issued.

Notes:

- A soft abort of this command halts execution at one of the following points:
 - Before the SCSI bus ARBITRATION phase, or
 - Before the initiator RESELECTION phase, or
 - Any time during the MESSAGE TRANSFER phase.
- Before issuing this command, the firmware should ensure that the MCS is set up as follows:



TGT SEND BSY - Target Send Busy Command

Command Code	Bus	Memory Access	Execution Time	Function
1101 01XX	D	N/A	5 us plus wait time	Target wait for select then send busy response and disconnect.

Operational Description

This sequence should be executed when the firmware wants to disregard any further selects. The sequence will perform the following tasks:

- Wait for a select to occur.
- Assert the -BSY signal on the SCSI bus.
- Receive one message from the initiator if the -ATN signal is asserted; however, the message is not saved.
- Send BUSY status byte to initiator.
- Send COMMAND COMPLETE message to initiator.
- Deassert the -BSY signal and allow the SCSI bus to go to the BUS FREE phase.

If the modifier bit is asserted, this command will not be cleared from the pipeline register when normal execution is completed. If no other commands exist in the other pipeline registers, then this command will be executed repeatedly until the firmware soft aborts the command.

Possible Errors

The following errors can cause this command to abort prematurely:

- The F82C5087 is not connected to the SCSI bus when this command is executed.
- An SM error occurs. A reselection occurs when the sequence is expecting a selection.

Notes:

- A soft abort of this command terminates command execution before the F82C5087 has been selected.
- The firmware will normally terminate this command with a soft abort when it is ready to process selects, or ready to reconnect to an initiator.

TGT WFS CMD - Target Wait for Select Command

Command Code	Bus	Memory Access	Execution Time	Function
1110 010X	D	MCS	1.5 us plus wait time	Target wait for select.

Operational Description

The firmware needs to execute the CNTL ENB SEL command for TGT WFS CMD and TGT WFS SEQ to be able to detect a select. The TGT WFS CMD waits for the F82C5087 to be selected or reselected, then it saves the ID of the device that selected or reselected the F82C5087 in the MCS. This command asserts the -BSY signal, and waits for the -SEL signal to be deasserted.

This command can also change the initiator or target mode of the F82C5087. If the F82C5087 is selected, the target mode is maintained; however, if the F82C5087 is reselected, the F82C5087 is changed to initiator mode. If a reselection occurs, this command deasserts the -BSY signal after the -SEL signal is deasserted.

The F82C5087 tags the ID byte to indicate whether a select or reselect occurred. When the MPU MCS Buffer Register contains the ID byte, the firmware should read Register 19's bit 5 before reading the MPU MCS Buffer Register. If bit 5 of Auxiliary Status Register 19 is asserted, a reselection occurred.

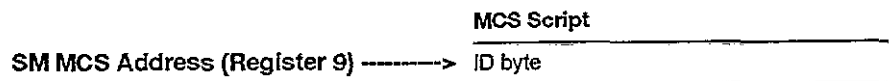
Possible Error

The following error can cause this command to abort prematurely:

- A connect error occurs. The F82C5087 is connected to the SCSI bus when this command is executed.

Notes:

- A soft abort of this command terminates execution before a select or reselect occurs.
- After this command is successfully executed, the ID of the device that selected or reselected the F82C5087 will have been received from the SCSI bus and placed in the MCS:



TGT WFS SEQ - Target Wait for Select Sequence

Command Code	Bus	Memory Access	Execution Time	Function
1111 01XX	D	MCS	10-20 us	Target perform wait for select sequence.

Operational Description

If the firmware issues this command to one of the pipeline registers and the F82C5087 is reselected, the mode is changed to initiator and INIT WFR SEQ is executed. If the F82C5087 is selected, the following sequence is executed:

- The ID of the initiator is saved in the MCS and the -BSY signal is asserted.
- If the -ATN signal is asserted, the identify message is received and saved in the MCS.
- If the -ATN signal is still asserted and the modifier bit is asserted, then the two-byte tag message is received and saved in the MCS.
- A multi-byte command is received and saved in the MCS.
- The first command byte is decoded.
- If the identify message allows a disconnect and the Override Disconnect Enable Bit is deasserted, the DISCONNECT message is sent to the initiator. If the identify message disallows a disconnect, the sequence is terminated at this point.
- The -BSY signal is deasserted and the SCSI bus is allowed to go to the BUS FREE phase.

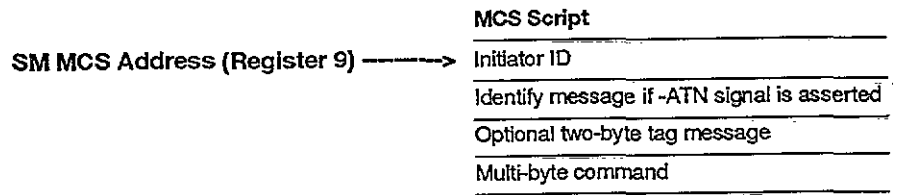
Possible Errors

The following errors can cause this command to abort prematurely:

- The F82C5087 is connected to the SCSI bus when this command starts to execute.
- An unexpected -ATN signal is asserted when the F82C5087 is receiving the SCSI command or sending the DISCONNECT message.
- An SM error (illegal message) occurs. If the -ATN signal is asserted, the F82C5087 must receive the ID as the first message, and the optional tag message as the next message. If any other message is received, the sequence is aborted with this error. The illegal message is transferred into the MCS.

Notes:

- A soft abort of this sequence terminates command execution before the select occurs, or after the sequence is completed.
- This sequence automatically calls the following pipeline commands: TGT WFS CMD, TGT REC MSG, TGT REC CMD, TGT SEND MSG, and TGT DISC CMD. For more information on any of these commands refer to their description in this chapter.
- After this command is successfully executed, the MCS should contain the information received from the initiator:



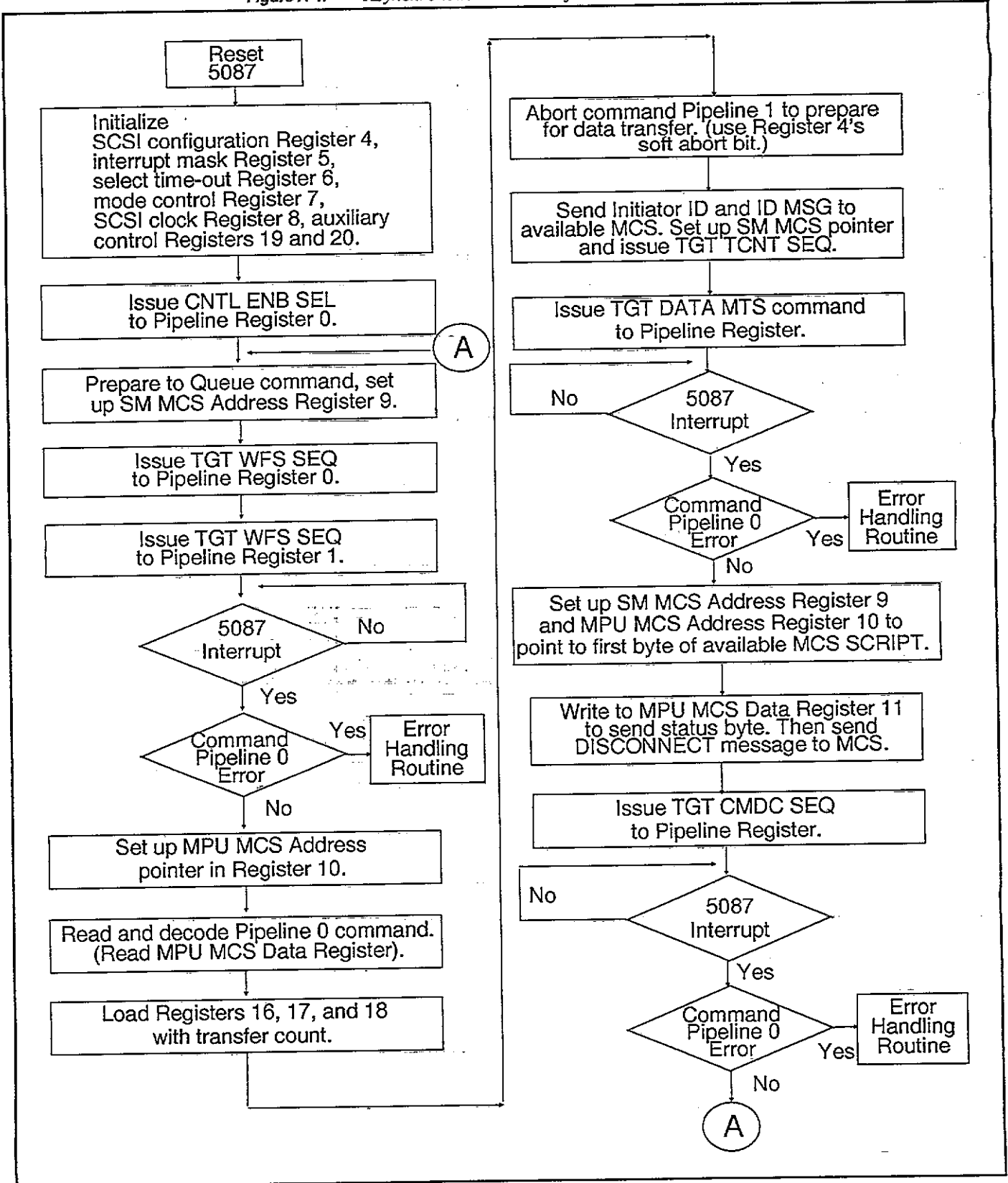


Appendix A

The following chart illustrates the steps involved in completing an asynchronous data transfer to a target device that allows a disconnect. The firmware must initialize certain F82C5087 registers, then queue the commands into the pipeline registers. The F82C5087 target device controller must receive the command from the initiator, decode it, and send status back to the initiator. Commands with vendor unique codes 6 and 7 are not used. For more details on command group codes refer to the description of Register 22 in Chapter 4. Figure A-1 illustrates the following steps:

1. The firmware sets up the necessary F82C5087 internal registers.
2. The firmware sets up the F82C5087 pipeline registers for command queuing and places two pipeline commands in these registers.
3. The SCSI target device receives the command.
4. The data transfer is completed and the target sends the initiator status.
5. Control loops back to the command queuing process.

Figure A-1. Asynchronous Data Transfer Flow Chart



Glossary

A_D	address/data
ANSI	American National Standard for Information Systems
ARBITRATION phase	the SCSI bus phase in which one SCSI device can gain bus control in order to assume the initiator or target role.
ATTENTION condition	the SCSI bus condition in which the initiator can signal the target to indicate it has a message ready to transfer out. The target prepares to receive this message by changing the SCSI bus to the MESSAGE OUT phase. The initiator can create the ATTENTION condition at any point other than during the SCSI bus ARBITRATION or BUS FREE phase by asserting the -ATN signal.
BUS FREE phase	the SCSI bus phase that indicates bus availability because there are no SCSI devices actively using it.
C	centigrade
Cascade	a method of connecting up to four 5087s (80- or 84-pin packages) to support wide SCSI and memory data bus transfers. Cascading four 5087s extend these two buses to 32 bits.
CHAM	chamfer
CMOS	complementary metal-oxide semiconductor
COMMAND phase	the SCSI bus phase in which the target asks the initiator for command information.
CONNECT sequence	the occurrence of an initiator selecting a target in order to start an operation.
CP	clock period
DATA IN phase	the SCSI bus phase in which the target requests that data be transferred to the initiator.
DATA OUT phase	the SCSI bus phase in which the target requests that data from the initiator be transferred to the target.

DC	digital circuit or computer
Differential mode	the mode of operation that supports a system configured with multiple-ended drivers/receivers. This mode supports operation over cables up to 25 meters long.
DISCONNECT sequence	the occurrence of the target giving up control of the SCSI bus, and allowing the SCSI bus to switch to the BUS FREE phase.
DMA	direct memory access
EEPROM	electrically erasable programmable read-only memory
FIFO	(first in, first out) the 64 byte portion of F82C5087 memory used for data transfers over the SCSI bus, the memory data bus, and the MPU's data bus.
Hard reset	the event that disconnects the target from the SCSI bus, and clears the contents of the 5087's internal registers.
Host adapter	the use of the F82C5087 as a bridge between the host computer bus and the SCSI bus.
Host MPU	the microprocessor that controls the host computer.
I/O	input/output
ID	identifier
Identify message	a single-byte message that specifies if a target is allowed to disconnect. It also specifies the initiator's LUN (logical unit number). This optional message is sent by the initiator, and is stored in the MCS.
Information transfers	a term used to encompass transfers of commands, data, status, and messages through the system.
Initiator	the SCSI device that initiates an operation with another SCSI device (the target). Usually, the initiator is the host adapter.
LUN	logical unit number
mA	milliamp
Master mode	one of the operational modes of a cascaded F82C5087: The F82C5087 operating in master mode is responsible for synchronizing the data transfers of the slave chip(s).
MAX	maximum

MCS script	(Message/Command Space script). The portion of F82C5087 memory that the F82C5087 state machine and the MPU use as a depository for SCSI commands, status and/or messages that will be routed through the system. It consists of six MCS scripts each 32 bytes long.
MESSAGE IN phase	the SCSI bus phase in which the target asks that the message(s) be sent to the initiator.
MESSAGE OUT phase	the SCSI bus phase in which the target asks that it receive the message(s) from the initiator.
MESSAGE phase	the term that encompasses the SCSI bus MESSAGE IN phase or the MESSAGE OUT phase.
MHz	megahertz
MIN	minimum
MPU	microprocessor
ms	millisecond
Non-differential mode	mode of operation that supports a system configured with single-ended drivers/receivers. This mode supports operation over cables up to six meters long.
NRZ	non-return to zero
ns	nanosecond
Peripheral device	a peripheral (e.g., printer, optical disk) capable of being attached to a SCSI device.
PF	the 80-pin version of the F82C5087 is available in a PFP (plastic flat package).
PLCC	the 68 and 84-pin versions of the F82C5087 are available in PLCCs (plastic leaded chip carrier).
RAM	random access memory
RECONNECT sequence	the occurrence of the target selecting an initiator to continue an operation after the target has disconnected.
RESELECTION phase	the SCSI bus phase in which the target reconnects to continue an incomplete operation after having disconnected from the initiator.
RESET condition	the SCSI bus condition in which all SCSI device connections are cleared from the SCSI bus.
SC	SCSI clock
SCSI	Small Computer System Interface
SCSI device	the host adapter or device controller that is capable of being attached to a SCSI bus.

SCSI ID	a single bit on the SCSI data bus that corresponds to the SCSI device's unique SCSI address.
SELECTION phase	the SCSI bus phase in which the initiator can select a target in order to initiate an operation.
Signal assertion	the driving of a signal to the true state.
Signal deassertion	the driving a signal to the false state.
Slave mode	one of the operational modes of a F82C5087 cascaded to support wide data transfers over the SCSI and memory data bus. When in slave mode, the chip's data transfers are synchronized by the F82C5087 chip operating in master mode.
SM	state machine. The F82C5087 controller has an internal state machine. This hardware component manages the flow of SCSI information through the system. Consequently, the firmware is free to manage higher-level system tasks (e.g., resource allocation).
Soft reset	the occurrence that disconnects the target from the SCSI bus; however, contents of the 5087's internal registers remain intact.
SQ	square
Status	after a command is completed the target sends one byte of status information to the initiator.
STATUS phase	the SCSI bus phase in which the target requests the initiator to transfer status to it.
TA	temperature ambient
Tag message	a two-byte, optional message that enables multiple commands to be queued in the MCS from a single initiator.
Target	the SCSI device that responds to the initiator's requests to perform operation(s).
TYP	typical
V	voltage
Vdd	drain DC voltage
uA	microamp
us	microsecond

CHIPS Regional Sales Offices

UNITED STATES

California, Irvine
Chips and Technologies, Inc.
(714) 852-8721

California, San Jose
Chips and Technologies, Inc.
(408) 437-3300

Florida, Boca Raton
Chips and Technologies, Inc.
(407) 241-3162

Georgia, Norcross
Chips and Technologies, Inc.
(404) 662-5098

Illinois, Schaumburg
Chips and Technologies, Inc.
(708) 397-4300

Massachusetts, Andover
Chips and Technologies, Inc.
(508) 688-4600

Texas, Dallas
Chips and Technologies, Inc.
(214) 702-9855

INTERNATIONAL

Germany
Munich
Chips and Technologies
GmbH
49-89-463-074

Japan
Tokyo
Chips and Technologies
Japan, K.K.
813-3-797-7242

Korea
Seoul
Chips and Technologies, Inc.
82-2-558-5559

Switzerland
Marin
Chips and Technologies S.A.
41-38-336379

Taiwan
Taipei
Chips and Technologies, Inc.
88-62-717-5595

CHIPS Sales Representatives

UNITED STATES

Arizona, Scottsdale
AzTECH Component Sales
(602) 991-6300

California, Santa Clara
Magna Sales
(408) 727-8753

California, San Diego
S. C. Cubed
(619) 458-5808

California, Thousand Oaks
S. C. Cubed
(805) 496-7307

California, Tustin
S. C. Cubed
(714) 731-9206

Colorado, Wheat Ridge
Wescom Marketing, Inc.
(303) 422-8957

Connecticut, Guilford
DataMark, Inc.
(203) 453-0575

Florida, Casselberry
Dyne-A-Mark Corp.
(407) 831-2822

Florida, Clearwater
Dyne-A-Mark Corp.
(813) 441-4702

Florida, Ft. Lauderdale
Dyne-A-Mark Corp.
(305) 771-6501

Idaho, Boise
Wescom Marketing, Inc.
(208) 336-6654

Illinois, Hoffman Estates
Micro-Tex, Inc.
(312) 382-3001

Indiana, Carmel
Giesting & Associates
(317) 844-5222

Kentucky, Versailles
Giesting & Associates
(606) 873-2330

Maryland, Annapolis
EES
(301) 269-4234

Massachusetts, Woburn
Mill-Bern Associates
(617) 932-3311

Michigan, Coloma
Giesting & Associates
(616) 468-3308

Michigan, Comstock Park
Giesting & Associates
(616) 784-9437

Michigan, Livonia
Giesting & Associates
(313) 478-8106

Minnesota, Eden Prairie
High Tech Sales Associates
(612) 944-7274

Missouri, Bridgeton
CENTECH
(314) 291-4230

Missouri, Raytown
CENTECH
(816) 358-8100

New Jersey, Moorestown
T.A.I.
(609) 778-5353

New York, Commack
ERA, Inc.
(516) 543-0510

New York, Pleasant Valley
Pitronics
(914) 635-3233

New York, Syracuse
Pitronics
(315) 455-7346

New York, Williamsville
Pitronics
(716) 689-2378

Ohio, Cincinnati
Giesting & Associates
(513) 385-1105

Ohio, Cleveland
Giesting & Associates
(216) 261-9705

Oregon, Beaverton
L-Squared Limited
(503) 629-8555

Pennsylvania, Pittsburg
Giesting & Associates
(412) 828-3553

Texas, Austin
OM Associates, Inc.
(512) 794-9971

Texas, Houston
OM Associates, Inc.
(713) 789-4426

Texas, Richardson
OM Associates, Inc.
(214) 690-6746

Utah, Salt Lake City
Wescom Marketing Inc.
(801) 269-0419

Washington, Kirkland
L-Squared Limited
(206) 827-8555

Wisconsin, Waukesha
Micro-Tex, Inc.
(414) 542-5352

CANADA

Ontario, Kanata
Electro Source, Inc.
(613) 592-3214

Ontario, Rexdale
Electro Source, Inc.
(416) 675-4490

Quebec, Pointe Claire
Electro Source, Inc.
(514) 630-7846

British Columbia, Vancouver
Electro-Source
(604) 435-8066

CHIPS Distributors

UNITED STATES

Alabama, Huntsville
Reptron Electronics, Inc.
(205) 722-9500

Arizona, Tempe
Anthem
(602) 966-6600

California, Chatsworth
Anthem
(818) 775-1333

California, Fountain Valley
Bell Microproducts
(714) 963-0667

California, Irvine
Anthem
(714) 768-4444

California, Milpitas
Bell Microproducts
(408) 434-1150

California, Rocklin
Anthem
(916) 624-9744

California, San Diego
Anthem
(619) 453-9005

California, San Jose
Anthem
(408) 453-1200

Colorado, Englewood
Anthem
(303) 790-4500

Connecticut, Waterbury
Anthem
(203) 237-2282

Florida, Ft. Lauderdale
Reptron Electronics, Inc.
(305) 735-1112

Florida, Tampa
Reptron Electronics, Inc.
(813) 854-2351

Georgia, Norcross
Reptron Electronics, Inc.
(404) 446-1300

Georgia, Norcross
JACO/QC
(404) 449-9508

Illinois, Elk Grove
Anthem
(708) 884-0200

Illinois, Schaumburg
Reptron Electronics, Inc.
(312) 832-1700

Massachusetts, Wilmington
Anthem
(508) 657-5170

Massachusetts, Wilmington
Bell Microproducts
(508) 658-0222

Maryland, Columbia
Anthem
(301) 995-6640

Michigan, Livonia
Reptron Electronics, Inc.
(313) 525-2700

Minnesota, Eden Prairie
Anthem
(612) 944-5454

Minnesota, Minnetonka
Reptron Electronics, Inc.
(612) 938-0000

New Jersey, Fairfield
Anthem
(201) 227-7960

New York, Commack
Anthem
(516) 864-6600

North Carolina, Raleigh
Reptron Electronics, Inc.
(919) 870-5189

North Carolina, Raleigh
JACO/QC
(919) 876-7767

Ohio, Solon
Reptron Electronics, Inc.
(216) 349-1415

Ohio, Worthington
Reptron Electronics, Inc.
(614) 436-6675

Oklahoma, Tulsa
JACO/QC
(918) 664-8812

Oregon, Beaverton
Anthem
(503) 643-1114

Pennsylvania, Horsham
Anthem
(215) 443-5150

Texas, Addison
JACO/QC
(214) 733-4300

Texas, Austin
JACO/QC
(512) 835-0220

Texas, Richardson
Anthem
(214) 238-7100

Texas, Sugarland
JACO/QC
(713) 240-2255

Utah, Salt Lake City
Anthem
(801) 973-8555

Washington, Bothell
Anthem
(206) 483-1700

CANADA

Ontario, Woodbridge
Valtrie Marketing
(416) 798-2555

CHIPS Distributors

INTERNATIONAL ASIA/PACIFIC

Australia
Victoria, Burwood
R&D Electronics
613-808-8911

Hong Kong
K wung Tong
Wong's Kong King Ltd.
85-2-345-0121

India
U.S.A.
Silicon Electronics
(408) 988-4408

India
New Delhi
Ajay Jain
91-11-686-3044

Israel
Tel Aviv
CVS
97-23-544-7475

Japan
Tokyo
ASCII Corporation
81-3-486-9188

Japan
Kawasaki
CTC Components Systems
Co. Ltd.
81-44-852-1825

Korea
Seoul
Nae Wae Semiconductor
82-2-704-0103

Malaysia
Penang
Dynamar
60-4-637-292

Singapore
Singapore
Technology Distribution
PTE, Ltd.
65-368-6065

Taiwan
Taipei
World Peace Industrial Co.,
Ltd.
88-62-786-5311

Taiwan
Taipei
ALLY, Inc.
886-2-788-6270

Thailand
Bangkok
Grawinner
66-2-215-8742

EUROPE

Belgium
Zaventem
ACAL Auriema Belgium
32-2-720-5983

Denmark
Herlev
Nordisk Elektronik A/S
454-284-2000

Finland
Helsinki
OY Fintronic AB
35-806-926022

France
Le Chesnay
A2M
33-1-395-49113

Germany
Netetal
Rein Elektronik GmbH
49-215-37330

Italy
Milano
Moxel S.r.L.
39-2-612-90521

Netherlands
Eindhoven
ACAL Auriema
Nederland B.V.
31-40-816-565

Norway
Hvalstad
Nordisk Elektronik A/S
47-284-6210

Spain
Madrid
Compania Electronica
de Tecnicas Aplicadas S.A.
34-1-754-3001

Spain
Barcelona
Compania Electronica
de Tecnicas Aplicadas S.A.
34-3-300-7712

Sweden
Kista
Nordisk Elektronik A.B.
46-8-703-4630

Switzerland
Dieikon
DataComp AG
41-1-740-5140

United Kingdom
Berkshire, Maidenhead
Katakana Limited
44-628-778622

United Kingdom
Berkshire
Magna Technology
44-734-880211

AMERICAS

Brazil
Florida
Etek Electronics Corp.
(305) 593-1188

Brazil
Sao Paulo
Brastek
55-11-531-2632

Mexico
Guadalajara
Sonika
52-364-74250