SYSTEM BUS (S-BUS)

Theory of Operation Manual

63-002 R00

The hardware description in this document is intended solely for use in assisting with the operation, installation, maintenance, or repair of Concurrent Computer Corporation equipment. The information in this document is subject to change without notice and should not be construed as a commitment by Concurrent Computer Corporation. Concurrent Computer Corporation assumes no responsibility for any errors that may appear in this document. Use of this document for all other purposes, without prior written approval from Concurrent Computer Corporation is prohibited.

This documentation is furnished under a license, and it can be used or copied only in a manner permitted by that license. Any copy of the described documentation must include any copyright notice, trademarks, or other legends or credits of Concurrent Computer Corporation and/or its suppliers. Title to and ownership of the described documentation and any copies thereof shall remain in Concurrent Computer Corporation and/or its suppliers.

The licensed materials described herein may contain certain encryptions or other devices which may prevent or detect unauthorized use of the licensed materials.

Any approved copy of this manual must include the Concurrent Computer Corporation copyright notice.

Warning: This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with the instructions manual, may cause interference to radio communications. It has been tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference in which case the user at his own expense will be required to take whatever measures may be required to correct the interference.

Concurrent Computer Corporation, 2 Crescent Place

Oceanport, New Jersey 07757

Printed in the United States of America

NOTICE

Before tracing signals, the technician should take the following into consideration.

On each sheet in a Functional Schematic set, signals are represented by mnemonics composed of capital letters and numbers. At the end of each signal mnemonic, either a one/zero (1/0) or nothing/negative sign (/-) is used to designate the high or low active state of the signal, respectively. Therefore,

- a one (1) or nothing ( ) indicates a high active state and,
- a zero (0) or negative sign (-) indicates a low active state.

For example:

SYSTEM CLEAR (SCLR)

	One/Zero Designation	Nothing/Negative Designation
High Active	SCLR1	SCLR
Low Active	SCLR0	SCLR-

Both are correct, but only one type of designation (either 1/0 or nothing/-) may appear on the sheets in a Functional Schematic set. The designations cannot be mixed within a Functional Schematic set. As systems documentation packages contain multiple Functional Schematic sets, it is probable that both types of designations will be present.

PREFACE

This manual describes the functional and operational capabilities of the system bus (S-bus). Chapter 1 presents an overview of the S-bus and the modules that comprise an S-bus system. Chapter 2 describes the conventions used in naming signal mnemonics and data structures. In addition, signal definitions are given. A functional description of the S-bus is provided in Chapter 3. Here, details on bus acquisition, clocks, the transmission protocol and the effects of the power clear signals are given. Chapter 4 discusses the various memory operations that are facilitated by the S-bus. Chapter 5 provides an account of direct data transfers on the S-bus including interprocessor messages, broadcast interrupts and direct input/output (I/O). Chapter 6 discusses the electrical specifications of the S-bus. Signal types and clock distribution are also discussed. Chapter 7 provides the physical specification of the S-bus backpanel.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 DESCRIPTION

The system bus (S-bus) is a high-performance synchronous bus used in the 3280 System. It interconnects major system modules: processors, input/output (I/O) controllers and memories. The S-bus conveys messages between modules on two distinct 32-bit data paths: the To-path (T-path) transfers addresses and write data to memory; the From-path (F-path) transfers read data from memory. This path is also used for interprocessor messages, direct I/O and interrupts.

The two paths are electrically identical. Each path contains 50 data lines which are divided into four fields: 5-bit function, 8-bit identification (ID), 32-bit data and 5-bit parity. These four fields comprise a data item. The parity bits help assure system integrity. Each path also contains two acknowledge and six acquisition signals. The acknowledges indicate the receipt of each data item. The acquisition signals control when each unit can use the bus. Figure 1-1 shows the modules that are connected to the S-bus in a 3280 System.

1.1.1 Control/Diagnostic System (CDS)

The CDS is used to configure, initialize and monitor the operation of the computer system. The CDS consists of a network of microprocessors linked by a full-duplex serial bus called the level 1 link. Processor and memory modules contain one of these microprocessors. All are slaves on the CDS bus (level 1 link).

The master of the CDS bus is a microcomputer in the power supply. This microcomputer communicates to the system console. It controls the power supplies and the power clear signals. It assigns and records the configuration and error status of all modules.

Each microprocessor can identify itself based on codes included in its firmware. These codes identify the module, serial number, options and revisions. The microprocessor can identify its location by reading an 8-bit slot number which is encoded into the backplane wiring. The slot number includes the chassis number and the position within that chassis. The slot number is used as an ID for all communications on the CDS bus.

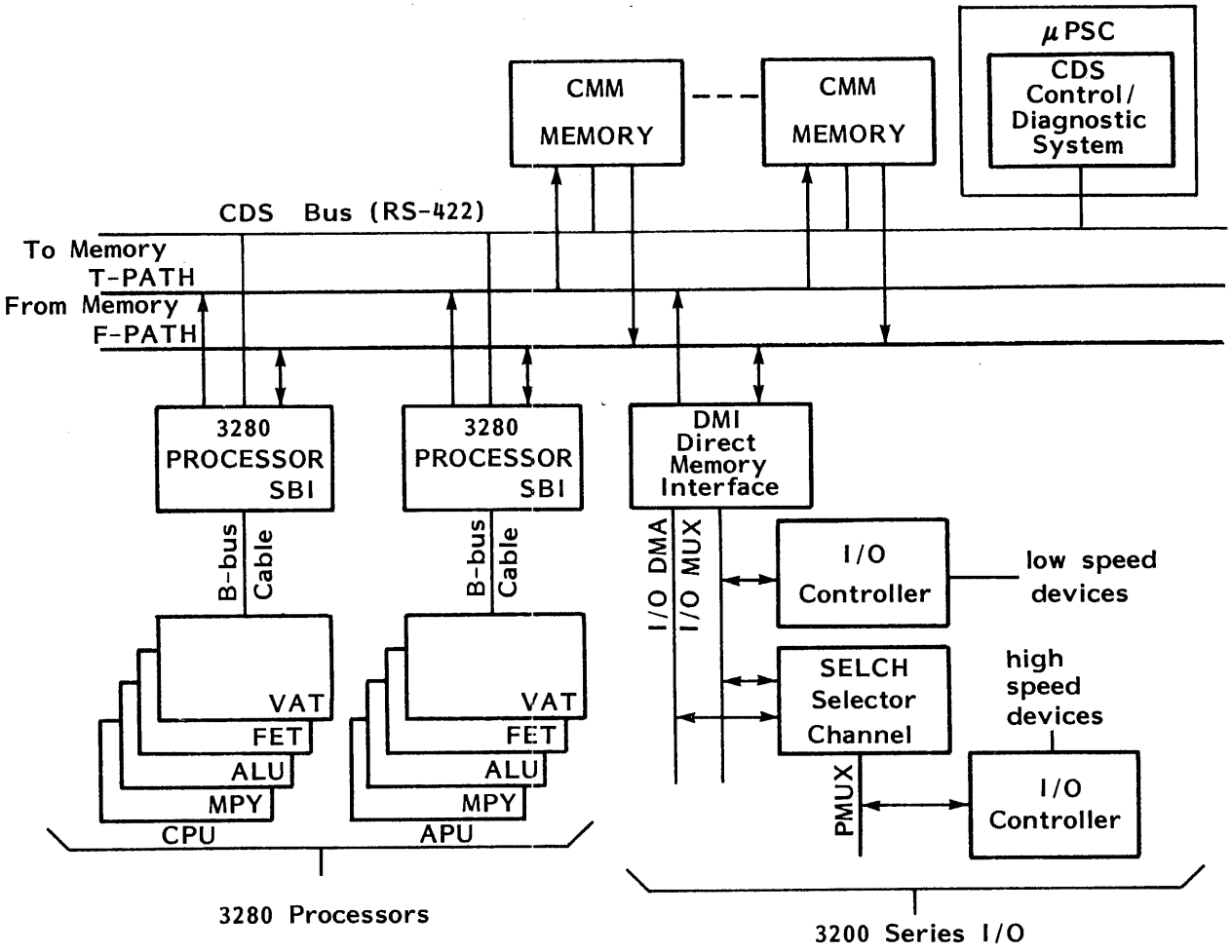
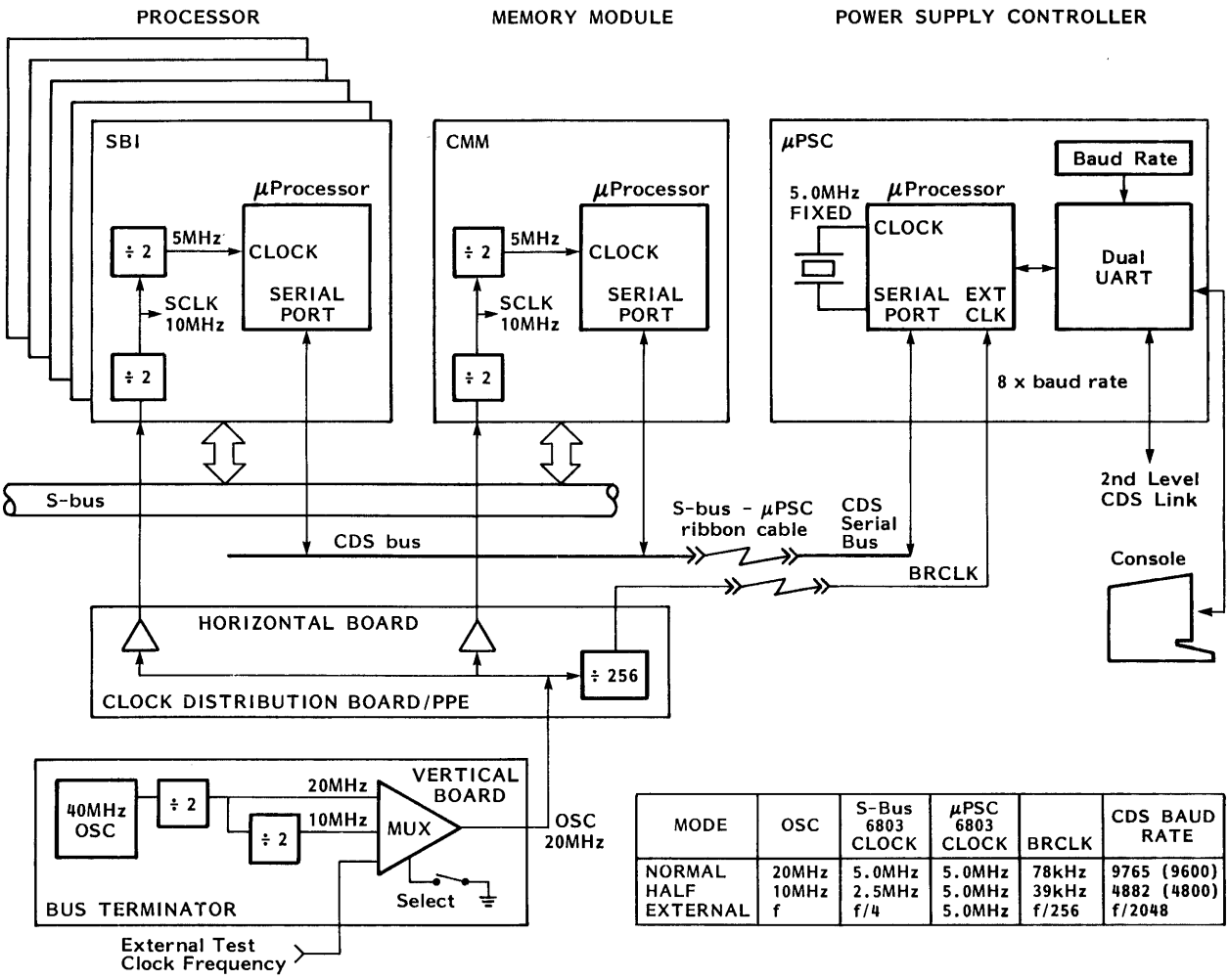


Figure 1-1 3280 System Block Diagram

1.1.2 Synchronous Operation

Bus operation is synchronous with the 10MHz system clock shown in Figure 1-2. This clock is derived from a 20MHz oscillator which is distributed to each board. Addresses and data are transferred from a master to a slave during one or more clock cycles. The master of each path has temporary control of that path. The next master is determined at the end of each operation. Bus acquisition circuitry allows positional priority, round-robin and high-priority access.



MODE	OSC	S-Bus 6803 CLOCK	μPSC 6803 CLOCK	BRCLK	CDS BAUD RATE
NORMAL	20MHz	5.0MHz	5.0MHz	78kHz	9765 (9600)
HALF	10MHz	2.5MHz	5.0MHz	39kHz	4882 (4800)
EXTERNAL	f	f/4	5.0MHz	f/256	f/2048

Figure 1-2 S-Bus and CDS Clock Distribution

1.1.3 Modules

The S-bus connects modules to form a system. Each module performs a function required in a computer system. These modules are implemented on printed circuit boards which plug into an S-bus backplane. Modules will usually be on separate boards, but several functions may be combined. Similarly, a complex module may require several boards.

Three types of modules are used to describe operations on the S-bus. Memory modules are selected by memory address. The other modules are selected by unique unit IDs, which are assigned by the CDS.

1. Memory modules consist of an array of random access memory (RAM) circuits and a controller. The controller interfaces to the S-bus, controls the RAM array and buffers data. All operations to the memory module are via the S-bus. Memory modules are not considered units.

2. Processors are programmed units which access data stored in the memory modules. The processors use cache buffers to improve performance. Processors can communicate with each other and can send commands to I/O units.

   Special purpose processors, such as array or display processors, may also be connected to the S-bus. These units contain local memories which may be modified independently of the S-bus. They can be accessed directly via the S-bus, provided that they respond with 'do not cache' codes.

3. I/O units provide an interface between the S-bus and peripheral devices. These devices are used for secondary storage (i.e., disk and tape drives), communications and interaction with the user. I/O transfers may either be direct I/O controlled by a processor or direct memory access (DMA) controlled by the device.

1.1.4 Bus Transfers

The module which has gained control of a path is called the master of that path. It can send a data item to another module, called the slave. On the T-path, any unit can be the master. The slave is a memory board; it is selected by the 32-bit memory address in the first word transferred. On the F-path, any module may be the master. The slave is a processor or direct memory interface (DMI); it is selected by the 8-bit bus ID.

Processors are identified by unit ID numbers. Each puts its ID on the T-path when it initiates a memory operation and recognizes its ID when it receives responses on the F-path.

1.1.5 Parity

All data transfers on T-path and F-path are protected using even parity. Five parity bits are used. TPAR4 and FPAR4 are parity on the function and ID fields; TPAR3:0 and FPAR3:0 are parity on the four data bytes.

Even parity means that the total number of 1-bits in the field, including the parity bit, is even. That is, if the field has an even number of ones, the parity bit is zero; otherwise, the parity bit is one. Parity is computed on logical values, not on the voltage levels on the bus; high signals are zero. When the bus is inactive (high), all bits are zero and the parity is even (correct).

1.1.6 Memory Operations

Memory is referenced solely by memory address; memory modules do not have unit IDs. Although memory modules normally contain just memory, memory within other units can be accessed via the S-bus, if that unit responds to memory commands.

Units initiate memory operations on the T-path. The function specifies a memory operation. The data field contains a memory address. Each memory module recognizes the addresses it contains. For memory writes, subsequent cycles send data to be written. For memory reads, the addressed memory module performs the operation then returns data on the F-path. Special memory operations provide test-and-set operations used for software interlocks. These operations read a word and then immediately modify that word in main memory.

1.1.7 Messages, Broadcasts and Direct Input/Output (I/O) on F-Path

Processors can communicate directly using the F-path. Messages can be sent to interrupt another unit. Broadcast interrupts are received simultaneously by all processors; this provides a means for the operating system to preempt lower priority processes and to delete old virtual address translations (VATs). Direct I/O operations allow a processor to directly access an I/O device.

1.1.8 Series 3200 Input/Output (I/O)

The Series 3200 I/O subsystem is supported in S-bus systems using the DMI board. This board generates the Series 3200 I/O multiplexer (MUX) and DMA buses. The MUX bus communicates with an assigned processor using special codes on the S-bus. The DMA bus accesses memory using standard S-bus operations. The bus is limited to 24-bit addresses, but the DMI board can translate these to 32-bit addresses for use in extended memory systems.

1.2 CACHE OPERATION

Processors use cache buffers to reduce memory operations. A cache is a small high-speed memory which is used to retain recently used data. This data is likely to be used again soon. When it is, the cache provides the data, thus avoiding the delay and contention of using main memory. Caches are a hardware feature; their operation is transparent to the running program.

1.2.1 Quadword Blocks

Cache block size is a quadword. A quadword is a block of four consecutive words aligned on a quadword boundary (any multiple of 16 bytes). These blocks can be efficiently loaded using the quadword read (MR4) function. The desired (addressed) word is

transferred first. Larger blocks may be used, but the bus cannot transfer more than a quadword in a single operation.

1.2.2 Cache Consistency

Caches significantly improve speed and reduce contention for bus cycles. However, each processor must ensure that the data in its cache is current. To do so, it monitors the activity of other units on the T-path for memory writes and special memory operations. Whenever a write affects a block stored in the cache, that block must be invalidated. The cache contents cannot be updated from the data on the bus.

All S-bus operations which modify memory use at least two cycles on the T-path. Thus, two cycles are available for each address that needs to be checked.

The need to ensure that caches are kept current puts constraints on cache design of processors on the S-bus. Caches must use write through organization. That is, when a processor writes to memory, both that cache and main memory are updated immediately. Otherwise, other processors could not guarantee that their data was current.

New blocks may be added to a cache only by loading them with a quadword read (MR4). Existing cache blocks are updated by any memory write, but new blocks may not be created by a write even if a complete quadword is written.

1.3 REFERENCE DOCUMENTATION

Additional documentation which describes other aspects of the 3280 System is available. Those manuals listed below provide either module-level theory or information on the 3280 System on a system level.

- 3280 System Control/Diagnostic System (CDS) Maintenance Manual
- 3280 System Operator Guide
- Model 34-043 Power Subsystem Theory of Operation Manual
- Composite Memory Module (CMM) Theory of Operation Manual
- Direct Memory Interface (DMI) Theory of Operation Manual
- 3280 System Central Processing Unit (CPU) Theory of Operation Manual
- 3280 System Installation and Configuration Manual

CHAPTER 2
SIGNAL DEFINITIONS AND CODES

2.1 INTRODUCTION

To-path (T-path) and From-path (F-path) signals are identical. All T-path signals have T prefixes and all F-path signals have F prefixes. The clock and control signals serve both paths. The oscillator and clock signals are distributed radially. Separate buffers for OSC- exist on each slot. Buffers for CLK- are present on every three slots. Each slot has individual request and grant signals for each path. See Table 2-1 for a summary of the system bus (S-bus) signal lines and data fields.

TABLE 2-1 S-BUS FIELDS AND SIGNAL DEFINITIONS

T-PATH SIGNALS	F-PATH SIGNALS	SIGNAL DEFINITION
		DATA FIELDS
T31:00-	F31:00-	32-bit address/data field
TFN4:0-	FFN4:0-	5-bit function select
TID7:0-	FID7:0-	8-bit unit identification number
		PARITY BITS ON MESSAGE
TPAR4-	FPAR4-	even parity bit for ID and FN fields
TPAR3-	FPAR3-	even parity bit for data bits 31:24
TPAR2-	FPAR2-	even parity bit for data bits 23:16
TPAR1-	FPAR1-	even parity bit for data bits 15:08
TPAR0-	FPAR0-	even parity bit for data bits 07:00
		RESPONSES (FROM PATH RECEIVER)
TAK1:0-	FAK1:0-	acknowledge: here, busy, fault
		BUS ACQUISITION SIGNALS
TRQn-	FRQn-	positional priority request (per slot)
TGRn-	FGRn-	positional priority grant (per slot)
TKEEP-	FKEEP-	keep bus for next cycle (open-collector)
TREQ-	FREQ-	request bus (open-collector)
TRREQ-	FRREQ-	round-robin request (open-collector)
THREQ-	FHREQ-	high-priority request (open-collector)

TABLE 2-1 S-BUS FIELDS AND SIGNAL DEFINITIONS (Continued)

T-PATH SIGNALS	F-PATH SIGNALS	SIGNAL DEFINITION
		CLOCKS
OSC-		20.00MHz oscillator
CLK-		2.000MHz clock
TXLF-		Twice line frequency
		CONTROL
MCLR-		Memory clear (sets memory nonvalid)
PCLR-		Processor clear (forces processors into reset)
SCLR-		System clear (forces system into reset state)
SLOT7:0		Slot ID number (for CDS address)
TOPSC, TOPSC-		CDS serial line "To Power Supply Controller"
FRPSC, FRPSC-		CDS serial line "From Power Supply Controller"

2.2 NOTATION

The following sections describe the conventions used to name signal mnemonics and data structures.

2.2.1 Mnemonic Naming Conventions

Signals are represented by mnemonics composed of capital letters and numbers. A signal is logic 1 (active) when the condition named by its mnemonic is true; otherwise, it is logic 0 (inactive). Postfix numbers identify individual signals within a field. A field is labelled with its bit range using a colon (:), for example, T31:00 represents T31 through T00 (32 signals). The most significant bit is listed first. A postfix minus indicates the complement of a signal, which is active when at logic low. Otherwise, the signal is assumed active when at logic high. The true definition of mnemonics is assumed in all circuit descriptions; the minus is used only when the polarity of the signal is significant. See Table 2-2 for additional information on signal mnemonics.

TABLE 2-2 SIGNAL MNEMONIC CONVENTION

LOGIC	TRUE SIGNAL	COMPLEMENT
Signal active '1'	Logic high	Logic low
Signal inactive '0'	Logic low	Logic high

Signal voltages are assigned transistor-to-transistor logic (TTL) levels as shown in Table 2-3.

TABLE 2-3 TTL LEVELS

LOGIC	VOLTAGE LEVELS
Logic low	Low voltage (0.0V to 0.8V)
Logic high	High voltage (2.0V to 5.5V)

2.2.2 Logic Equations and Tables

The operation of circuitry is described in equations and tables. In these, the mnemonics indicate the true meaning of the signal. The logic polarity of the actual signal is ignored. Instead, the postfix minus sign is used to indicate the logical complement.

Mnemonic SIG	Complement SIG-	
0	1	Mnemonic false, complement true
1	0	Mnemonic true, complement false

2.2.3 Data Words, Halfwords, Bytes

A word is 32 bits wide; a halfword is 16 bits wide; a byte is 8 bits wide. Binary values are represented in hexadecimal (hex) notation; each 4-bit nibble has a value of 0 to 15. The values 10 through 15 are represented by characters A through F respectively.

2.2.4 Power-of-Two Notation and Memory Addressing

Power-of-two notation is used to number bits within a field. Bits T31, TFN4, TID7, etc., are the most significant bits; bit 0 is the least significant. The weight of each bit 'n' equals two raised to the power 'n'. Therefore, bits 0, 1, 2, 3 through 'n' have weights 1, 2, 4, 8 through 2^n.

To maintain functional compatibility with earlier machines, memory bytes and halfwords are numbered left to right. Byte 0 is the most significant and byte 3 is the least significant. Similarly, halfword 0 is the most significant and halfword 1 is the least significant. See the following table and Figure 2-1 for a graphic representation of these data structures.

Bytes (8 bits)	Halfwords (16 bits)
Byte 0 = bits 31:24	HW 0 = bits 31:16 (bytes 0 and 1)
Byte 1 = bits 23:16	
Byte 2 = bits 15:08	HW 1 = bits 15:00 (bytes 2 and 3)
Byte 3 = bits 07:00	

002-3

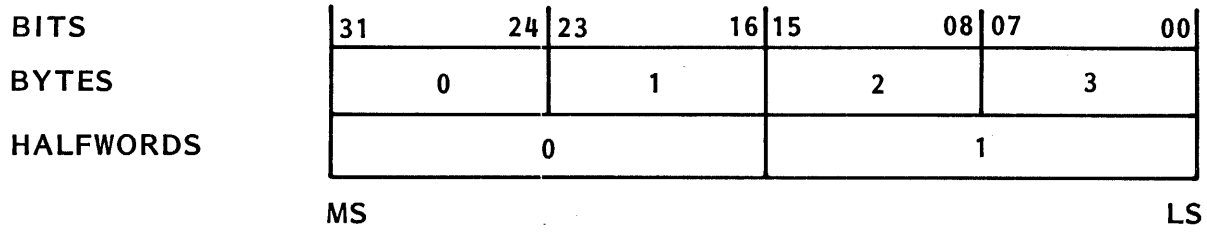


Figure 2-1 Conventions for Naming Data Structures

2.3 FUNCTION CODES (FN)

On each path, the 5-bit function code specifies the type of data that is on the path during each cycle. For memory write, I/O write and messages, the data cycle(s) immediately follows the address cycle. Other data cycles (memory read, I/O read) are separate.

Function 0 specifies an idle cycle during which there is no transfer. The other fields, including parity, are ignored. The master must drive the function code to zero during idle cycles; the other fields need not be driven.

2.3.1 T-Path Function Codes

The T-path is used solely to initiate memory operations. Each T-path operation begins with an address cycle and with TFN4 set. TFN4 is zero for data and idle cycles.

The four memory read functions cause the memory module to read one to four words from the same quadword. This data is returned to the requesting unit via the F-path. The contents of memory are not modified.

The four memory write functions modify one to four words within a quadword in a memory module. The first and last words may be written partially, where only selected bytes are modified. This facilitates writing byte strings into memory. When the entire word is written, the memory module computes an error check and correction (ECC) code and writes into its array. When only part of a word is written, the module must first read and check the previous contents of that word and then calculate a new ECC on the changed data. There is no response on the F-path.

The six special memory functions perform read-modify-write operations as an indivisible sequence. These operations are useful for coordinating tasks in a multiprocessor system. Special functions operate as a combination of memory read and memory writes: one data word must follow each address, the previous contents of the addressed word are returned on the F-path.

The four data functions accompany the data items following memory write or special memory functions. The code identifies the last (least significant) byte which is to be modified. DW3 must be used when the entire word is written and for all special functions. See Table 2-4 for the T-path function codes.

TABLE 2-4 T-PATH FUNCTION CODES

HEX	TFN4:0	MNEMONIC	OPERATION	
00	00000	IDL	Idle bus cycle	
01-03			Reserved	
04	00100	DW0	Data word, end at byte 0	D
05	00101	DW1	Data word, end at byte 1	A
06	00110	DW2	Data word, end at byte 2	T
07	00111	DW3	Data word, end at byte 3	A
08-0F			Reserved	
10	10000	MR1	Memory Read: 1 word	R
11	10001	MR2	Memory Read: 2 words	E
12	10010	MR3	Memory Read: 3 words	A
13	10011	MR4	Memory Read: 4 words	D
14	10100	MW1	Memory Write: 1 word	W
15	10101	MW2	Memory Write: 2 words	R
16	10110	MW3	Memory Write: 3 words	I
17	10111	MW4	Memory Write: 4 words	T
				E
18	11000	MRS	Memory Read and Set	
19	11001	MRR	Memory Read and Reset	S
1A	11010	MRI	Memory Read and Increment	P
1B	11011	MRD	Memory Read and Decrement	E
1C	11100	MRW	Memory Read and Write word (Exchange word with memory)	C I
1D	11101	MWD	Memory Write Diagnostic	A
1E			Reserved	L
1F			Reserved	

2.3.2 F-Path Function Codes

The F-path is used for memory read data, direct I/O, broadcast interrupts and messages between units. The four broadcast functions are received and decoded by all units. Memory read data, direct I/O and the four message functions are decoded only by the unit addressed by FID7:0.

IOW and FDAT must be sent as a pair of consecutive items; each MSGn must be followed consecutively by two FDAT items. The other transfers may be sent singly. Compatible I/O and messages use subfunction commands on F31:28. See Table 2-5 for the F-path function codes.

TABLE 2-5 F-PATH FUNCTION CODES

HEX	FFN4:0	MNEMONIC	OPERATION
00	00000	IDL	Idle bus cycle
		DIRECT I/O RESPONSE	
01	00001	ATN	Attention (I/O interrupt lines)
02	00010	IOK	I/O acknowledge, data okay
03	00011	IER	I/O acknowledge, data error
		MEMORY RESPONSE	
04	00100	MDC0	Memory read, word 0, disable cache.
05	00101	MDC1	" word 1, "
06	00110	MDC2	" word 2, "
07	00111	MDC3	" word 3, "
08	01000	MOK0	Memory read, word 0, data correct.
09	01001	MOK1	" word 1, "
0A	01010	MOK2	" word 2, "
0B	01011	MOK3	" word 3, "
0C	01100	MER0	Memory read, word 0, data error.
0D	01101	MER1	" word 1, "
0E	01110	MER2	" word 2, "
0F	01111	MER3	" word 3, "
		DIRECT I/O	
10	10000	IOR	I/O read (see I/O subfunction)
11	10001	*	reserved
12	10010	IOW	I/O write (see I/O subfunction)
13	10011	FDAT	F-path data (follows IOW or MSGn)
		BROADCASTS	
14	10100	PRE	Preempt interrupt
15	10101	DVA	Delete virtual address interrupt
16	10110	SYNC	Start real-time clock interrupt
17	10111	*	reserved broadcast interrupt
		MESSAGES	
18	11000	MSG0	Message #0
19	11001	MSG1	Message #1
1A	11010	MSG2	Message #2
1B	11011	MSG3	Message #3
1C-1F		*	reserved

2.3.3 F-Path Response Codes

IOK and IER are responses to IOR and IOW commands. (See Section 5.4 for details.) MOKn, MERn and MDCn are responses for memory read operations. The responses for a multiple word read operation will usually occur on successive cycles, but this is not required. (See Chapter 4 for additional details.)

The responses (IOK, IER, MOK, MER, MDC) send one item to indicate completion of a direct I/O or memory operation. All direct I/O is acknowleged; this signals the processor to proceed. For IOR responses, read data is returned on F31:00. For IOW responses, the data on F31:00 is not used. Memory reads, including special operations, are responded to with the read data returned on F31:00. Each word of a quadword memory read is sent (MOK, MER or MDC) separately. The memory module may retain the bus for all four words. For memory writes, there is no response on the F-path; the processor does not wait.

2.3.4 Summary of F-Path Transfers

All F-path transfers are illustrated in Table 2-6. Each line represents the item sent during one S-bus cycle. When a transfer requires two or three words, these must be sent on consecutive cycles.

TABLE 2-6 SUMMARY OF F-PATH TRANSFERS

MNEM	WORD	FFN	FID	F31 ^ F23 ^ F15 ^ F07 ^ F00
ATN:	1:	01	unit-d	.....P3210
IOK,	1:	02-03	unit-d	|<--data (if read operation)--->|
IER	1:	03	unit-d	|<--data (if read operation)--->|
MDC	1:	04-07	unit-d	|<----data read from memory---->|
MOK	1:	08-0B	unit-d	|<----data read from memory---->|
MER	1:	0C-0F	unit-d	|<----data read from memory---->|
IOR:	1:	10	unit-d	|SFN|PR |<-SID->|<---device---->|
IOW:	1:	12	unit-d	|SFN|PR |<-SID->|<---device---->|
FDAT:	2:	13	unit-d	|<--------write data---------->|
PRE:	1:	14	sender	|<level>|<---unused--->|<-mask->|
DVA:	1:	15	sender	|<virtual address to be deleted>|
SYNC:	1:	16	sender	|<---------not used---------->|
(reserved)		17	sender	
MSG0:	1:	18	unit-d	|<level>|<-ID-->|<-Parameter#1->|
FDAT:	2:	13	unit-d	|<----Parameter#2 (address)---->|
FDAT:	3:	13	unit-d	|<----Parameter#3-------------->|
MSG1:	1:	19	unit-d	|<level>|<-ID-->|<-Parameter#1->|
FDAT:	2:	13	unit-d	|<----Parameter#2 (address)---->|
FDAT:	3:	13	unit-d	|<----Parameter#3-------------->|
MSG2:	1:	1A	unit-d	|<level>|<-ID-->|<-Parameter#1->|
FDAT:	2:	13	unit-d	|<----Parameter#2 (address)---->|
FDAT:	3:	13	unit-d	|<----Parameter#3-------------->|
MSG3:	1:	1B	unit-d	|<level>|<-ID-->|<-Parameter#1->|
FDAT:	2:	13	unit-d	|<----Parameter#2 (address)---->|
FDAT:	3:	13	unit-d	|<----Parameter#3-------------->|

Where:

FFN — lists the F-path function codes used for each transfer.

FID — specifies whether the master transmits its own 8-bit unit ID (sender) or the ID of the destination unit (unit-d).

F31:00 illustrates the type of information sent in each word. For attentions (ATN), P3210 represents the status of the DMI's external power fail and interrupts levels 3, 2, 1 and 0.

The first word of IOR, IOW and MSG operations is divided into the following subfields:

F31:28	SFN	Subfunction code	(4 bits)
F27:24	PR	Priority code	(4 bits)
F23:16	SID	Sender unit ID number	(8 bits)
F15:00	DEV	Device select	(16 bits)

See Chapter 5 for additional information.

The sender inserts its unit ID into the items it sends. For I/O operations and messages, this allows the receiving unit to respond to the sender's ID. For broadcasts, the sender puts its own ID onto FID7:0, so that it may be monitored for diagnostics.

CHAPTER 3
FUNCTIONAL DESCRIPTION

3.1 CLOCKS

Bus timing is synchronous with a 10.00MHz system clock (SCLK). SCLK is generated on each board using two timing signals which are distributed radially on the system bus (S-bus) backplane:

OSC - is a 20.00 ± 0.001%MHz oscillator. OSC has a 50ns period with less than 2ns jitter. It has a 50% ± 10% duty cycle at the connector.

CLK - is a 2.00MHz signal used to synchronize SCLKs. CLK is derived from OSC and is active for one OSC period (50ns) every 500ns.

Bus timing is relative to the falling edge of OSC- on the backpanel. Timing skew from OSC- to SCLK must be minimized. The rising edge of OSC- is imprecise; hence, it is not used for any bus operation.

The bus distributes the double frequency (20MHz) OSC clock for use internally on each board. Bus activity occurs only at 10MHz.

3.2 POWER CLEAR SIGNALS

The three clear signals indicate the status of the power supplied to the system. These signals are used to reset circuitry.

MCLR- Memory Clear (P5U powered circuitry)
PCLR- Processor Clear
SCLR- System Clear

3.2.1 Power Supplies

S-bus systems operate from two 5V power supplies. These power supplies are controlled by the power supply controller. This includes a microprocessor which is also the master node of the control/diagnostic system (CDS). The microprocessor provides the three power status signals to the S-bus. The microprocessor operates from its own power supply, so it can switch the main power supplies on and off.

P5 (Power, 5V) is the main logic power for the system, derived from AC line power. P5U (Power, 5V, Uninterrupted) is auxiliary logic power which operates from a battery when AC line power is not available. P5U uses AC line power when available. P5 and P5U are independently switched on and off by console commands.

P5U powers the volatile semiconductor random access memory (RAM) chips and associated control circuits on the memory modules. Battery power retains data during short power outages. Battery back-up is optional. However, P5U is still distinct from P5 when there is no battery. P5U is active whenever AC power is present, even if P5 power is switched off by console command.

3.2.2 Memory Clear (MCLR)

The MCLR indicates that P5U power is not available and main memory data is invalid. MCLR resets circuitry powered by P5U on memory modules. After MCLR, the contents of the memory array are assumed not valid. During the next initialize period, the memory module must write zeros (or other values) into all memory words. This guarantees that each location has a valid error check and correction (ECC) code and will not create spurious memory failure errors. The memory has at least one second to initialize before the processor initialization begins. Power clear (PCLR) and system clear (SCLR) are always active whenever MCLR is active.

When MCLR is inactive and both PCLR and SCLR are active, the memory modules operate in a standby mode. In this mode, the memory array is powered and refresh cycles are performed, keeping the contents of memory valid.

3.2.3 Logic Power Sequencing

Logic power-on/-off cycles occur when the AC power is lost or upon commands from the system console to turn the P5 supply on or off. PCLR and SCLR encode four steps in each power cycle. This sequence is repeated each time power is cycled on and off. P5U power and MCLR are not affected by power cycles as long as the battery has enough stored energy to provide P5U power. For test purposes, the power supply controller must be able to issue this sequence without actually turning power off.

PCLR	SCLR	
1	1	Reset all circuits powered by P5.
0	1	Initialization.
0	0	Operation. System is powered and running.
1	0	Early power-fail (EPF) warning; save state in memory.

```
Normal Sequence:   11 ->  01 ->  00 ->  10 ->  11  ...
Aborted Init.:     11 ->  01 ---------------->  11  ...
                  Reset   Init   Run    EPF    Reset
```

The power supply controller determines the timing of each period. The reset period is at least one second long. The initialization period is normally at least one second long. It is aborted if there is a power failure during initialization; PCLR/SCLR changes immediately from "01" to "11"; the operation and EPF states are omitted.

Subsystems must be able to abort their initialization cycle without causing errors. Initialization lasts far longer than the 3ms ride-out time of the power supply. Therefore, power may fail again during initialization. When this happens, PCLR is reasserted without SCLR ever going inactive. See Figure 3-1 for a timing diagram of the three S-bus power clear signals.

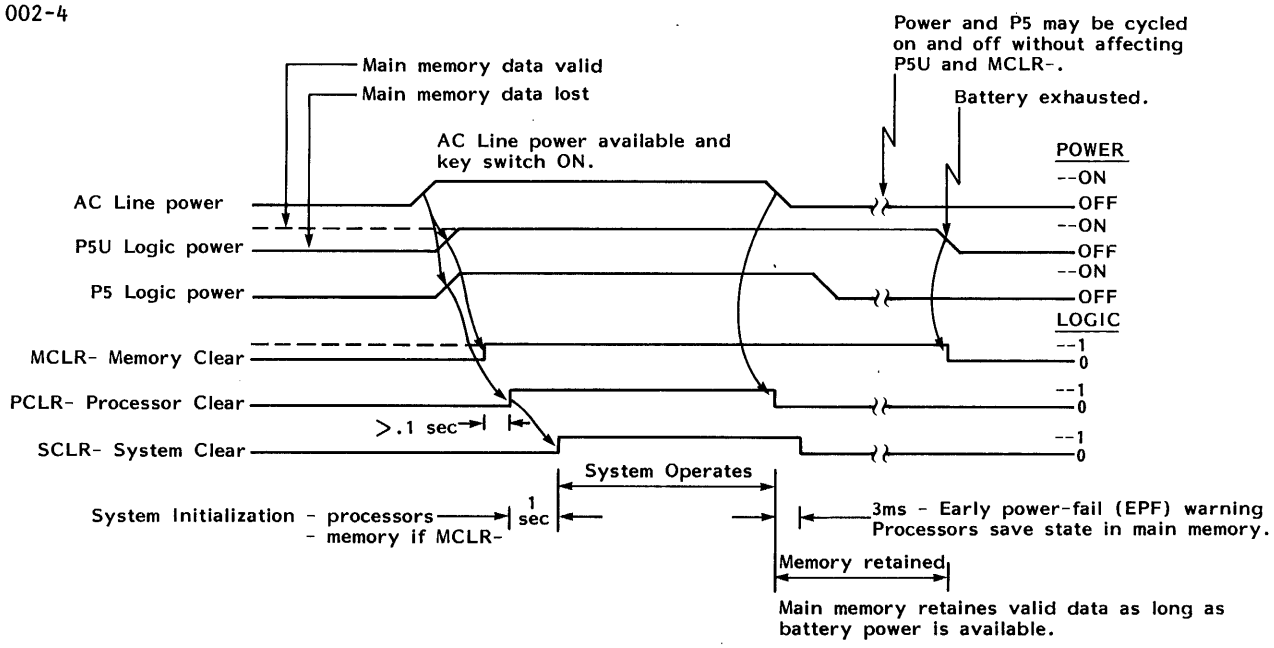


Figure 3-1 Power Clear Signal Timing

3.2.3.1 Reset

P5 power is unavailable. All circuits powered by P5 must be reset. Reset lasts at least one second.

3.2.3.2 Initialization on Power-On

The power subsystem turns P5 on when AC line power is available. When P5 is fully on and within regulation limits, the power system deactivates PCLR. SCLR is deactivated one second later, unless a power failure occurs. During this interval, all processors perform power-on diagnostics and initialize all registers. Furthermore, memory modules reset and test their controllers. If MCLR was active at any time since the last initialization step, each memory module initializes all words in its memory array. Each word is set to zero with a correct ECC code. Otherwise, memory data is valid and must be retained.

3.2.3.3 Operation

The system operates when both PCLR and SCLR are inactive.

3.2.3.4 Early Power-Fail (EPF) Warning

PCLR goes active to indicate an EPF warning. This informs the processors that power will fail, and SCLR will go active 3ms later. This warning is issued when either the power supply is turned off using a console command or when the AC line fails to provide enough power to operate the computer. The P5 and P5U power supplies remain within regulation during this interval. Meanwhile, processors store their registers into main memory so that they can resume processing when power is restored. Main memory remains valid during short power outages (with battery option).

The EPF warning is issued whether or not the power supply includes battery back-up. The system may perform some I/O or other emergency operations, within the 3ms limit.

3.3 BUS ACQUISITON AND PRIORITY

The To-path (T-path) and From-path (F-path) have identical but independent circuitry which allocates bus cycles between boards. This includes acquisition circuits on each board and a positional priority encoder mounted on the backplane. These circuits determine the sequence in which requests are granted; more urgent requests are given priority. Each board slot has positional priority over slots with higher location numbers; slot 0 has the highest priority.

Bus requests have three levels of priority:

High-priority	(highest priority level)
Round-robin	
Simple	(lowest priority level)

All requests at a higher level are granted before any request at a lower level. Within any level, positional priority determines which request is granted first. Within the round-robin level, requests are selectively enabled to provide rotating priority.

To use a path, a board must request and acquire the path. Bus allocation operates in parallel with transfers on the S-bus. Each cycle determines which board gets each path on the following cycle. The board which has a path is known as the path's master and has its MINE flipflop set. At any cycle, there is only one master of each path. For all other boards, MINE is reset.

3.3.1 Bus Request and Grant

For each path, every S-bus slot is allotted one bus request signal, RQn-, and one bus grant signal, GRn- ('n' is the slot number). The requests and grants are active low; pull-up resistors assure that the request lines of empty slots remain inactive. Unused request lines may be left unconnected.

A board requests a path by asserting its RQn-. The positional priority encoder determines which board has the highest priority request pending and grants it control of the path for the next cycle (GRn-). All other grant signals remain inactive (high); the other boards must wait.

The positional priority is modified by four open-collector bus signals. These operate by disabling selected categories of requests, which allow the more urgent requests from other slots to be granted.

KEEP-	Master retains the bus to complete the current operation.
REQ-	Any unit, other than the master, requests use of the bus.
RREQ-	Round-robin priority request.
HREQ-	High-priority request.

3.3.1.1 Bus KEEP

The master asserts KEEP during operations that require two or more cycles. This prevents loss of the bus in the middle of an operation. KEEP is asserted when the master needs the bus on the following cycle to complete the current operation; normally, KEEP is inactive on the last cycle of any operation. However, KEEP is not used to retain the bus for consecutive operations. KEEP is inactive during idle cycles.

Any board may use KEEP to retain the T-path for retrying a memory operation, if it has already gotten two busy acknowledges on that operation. (See Section 3.5.1.1 for further details.)

3.3.1.2 Bus Request Pending

REQ (bus request pending) is asserted by any board that needs to acquire the bus. This enables transferring control. Otherwise, the master retains the bus, even if it is idle. The master should not assert REQ. The bus is transferred only when KEEP=0 and REQ=1. The priority circuits determine which board is granted use of the bus; the master will retain the bus if it has the highest enabled request.

3.3.1.3 Priority Requests

HREQ or RREQ can be asserted with REQ for high-priority or round-robin requests. HREQs have priority over RREQs; both have priority over other simple requests. These signals operate by inhibiting (and thus delaying) requests from boards with lower level of priority. When HREQ is active, only high-priority requests are enabled. When RREQ is active, simple requests are disabled.

Processors should have high-positional priority since they are most sensitive to memory access time. (System throughput degrades when a processor must wait for the bus.) Consequently, processors should be placed in slots with higher priority (low number slots). To avoid direct memory access (DMA) lock-out, processors do not assert HREQ.

3.3.1.4 Round-Robin Allocation

Round-robin priority is used to allocate use of the bus equally to all processors. Otherwise, a high-performance processor could monopolize the bus during periods of high-memory usage. Lower priority processors might stall since they could not acquire the bus. The round-robin control line RREQ permits processors to share the bus more equally. Processors in lower slots still have priority for a single cycle, but each may do only one operation until all have had access.

Processors request their first operation with both REQ and RREQ. This gives them priority over simple requests. When several processors have pending requests, RREQ stays active until each gets one operation. A processor which requests a second cycle while RREQ is still active must assert only REQ. This is a low-priority request; it can be granted only if no round-robin requests are asserted. When RREQ goes inactive, all processors may again use RREQ.

Memory modules use RREQ to share the F-path equally. When possible, the four words of a quadword are sent on consecutive cycles without relinquishing the F-path. Processor messages and direct I/O also use RREQ.

3.3.1.5 High-Priority Allocation

High-priority requests are serviced first; the HREQ signal inhibits round-robin and simple requests. Thus, the high-priority request will get the next access. If several high-priority requests are pending, the positional priority encoder determines which board gets the bus.

HREQ is not used indiscriminately, otherwise it would lose its effectivness due to heavy bus contention. Boards which routinely require rapid access should be given high-positional priority.

The direct memory interface (DMI) generates a Series 3200 DMA bus. Some controllers designed for this bus require prompt response because they do not include sizable buffers. Thus, the DMI can be configured to request high-priority for DMA operations.

3.3.1.6 Bus Acquisition Signals

Figure 3-2 illustrates bus acquisition circuitry. All priority modes are shown; RREN and/or HPEN requests may be deleted if they are not needed.

```
  MINE flipflop: board is master of the path.
  RQ   flipflop: board requests the bus.
  RREN flipflop: round-robin priority enable.
  HPEN flipflop: high-priority enable.

  PASS=KEEP-*REQ     pass the bus to a new master.
  GETBUS =PASS*GRn   gain use of bus (sets MINE, resets RREN).
  LOSEBUS=PASS*GRn-  lose use of bus (resets MINE).

  assert KEEP if MINE* (next cycle is needed for this operation)
  assert REQ  if RQ*MINE-
  assert RREQ if RQ*RREN  round-robin
  assert HREQ if RQ*HPEN  high-priority

bus request:
  RQn = RQ*HPEN              high-priority request
      + RQ*RREN*HREQ-        round-robin, unless inhibited
      + RQ*RREQ-*HREQ-       simple, unless inhibited
```

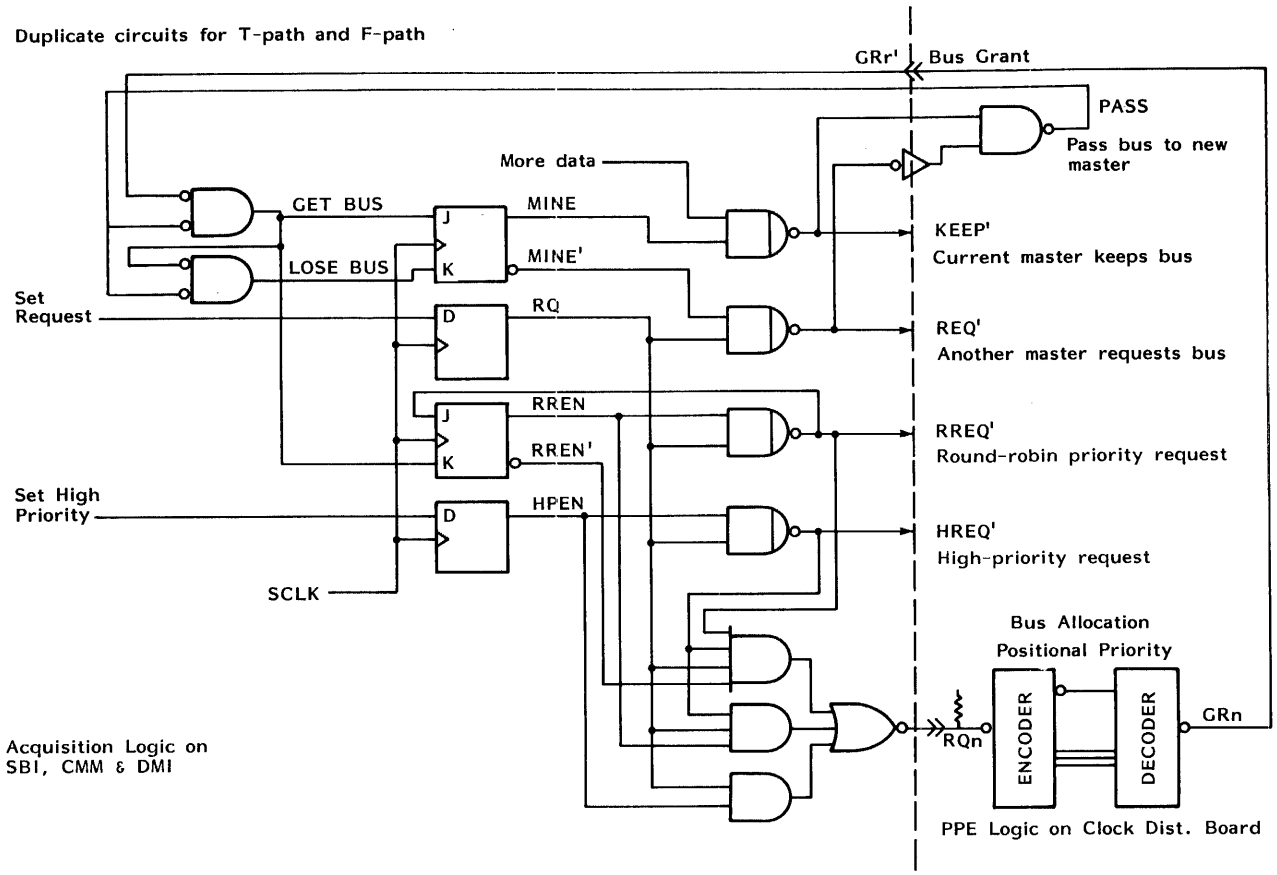


Figure 3-2 S-Bus Acquisition Logic

3.4 UNIT IDENTIFICATION (ID)

Processors and channels use 8-bit ID codes. On the T-path, TID7:0 identifies the requesting unit. Memory modules latch TID and echo it on FID when responding with read data. On the F-path, FID7:0 addresses the destination unit. Each unit compares FID with its unit address and latches all messages which match.

Broadcast interrupts must be received by all processors regardless of the ID match. No acknowledge is sent. For any other F-path function code, only the addressed unit may respond. The FID field must identify the unit which sent the broadcast; this ID is required to allow diagnostic hardware to monitor activity on the S-bus; it is not used for normal operation.

3.4.1 Multiple IDs Per Unit

Bus ID codes must be unique within an S-bus system. However, units may use several IDs to keep track of overlapped memory read operations. Due to memory interleaving, the data sequence may differ from the request sequence. High-performance processors, like the 3280, can use concurrent memory reads to reduce interference between instruction and data fetches.

3.5 TRANSMISSION PROTOCOL

The master of each path can send one item during each clock cycle. The master enables its bus drivers when it gains control and disables them when it relinquishes control to another module.

Every slave receives every item. The item is decoded during the following cycle; no logic delays are allowed during the transfer itself. The slave determines whether or not it is addressed. If it is not, or if the function code is idle, it takes no further action. The slave checks the parity of each item it receives. This check must be performed after the item has been latched in a register, to ensure that all bits are stable. Parity of idle cycles is ignored.

3.5.1 Bus Acknowledge

On each path, the selected slave acknowledges each item it receives two cycles after that item was sent. This pipelining of the acknowledges permits high transfer rates.

These lines acknowledge only receipt of the message. The Here response does not imply that the requested operation can be performed.

The acknowledge lines are driven with open-collector gates. The response is the logical-OR of all slaves responding. Normally only one slave will respond, but a parity fault or improper configuration can cause two boards to respond to the same message. The fault code can be aliased if one of these boards is busy.

AK1:0	Meaning
0 0	No response. (Idle cycle or device not present.)
0 1	Here. (Slave is present and accepted the item.)
1 0	Busy. (Slave is present but did not accept item.)
1 1	Fault. (Slave detected parity error.)

3.5.1.1 Bus Busy Acknowledge

The slave responds as busy if its input buffer is full and cannot accept the operation. Except for the memory module, the master reacquires the bus and repeats the operation. The slave empties the buffer expeditiously.

Memory and other slaves assert busy only on address cycles. They assert here only when it can accept the entire operation. On the T-path, the function code bit, TFN4, is 1 for an address cycle, 0 for a data cycle.

3.5.1.2 Holding the T-Path After Busy

The master repeats an operation if it receives a busy acknowledge. Normally, the unit reacquires the path using the same priority while other units may use the path. The unit must retry the operation as long as it receives busy acknowledges. In rare cases, this could continue indefinitely. Although the slave cannot remain busy for long, other units can refill its input queue while the unit waits to reacquire the path. To avoid indeterminate delays, the unit may hold the bus after it has received two consecutive busys. It holds the bus by asserting KEEP continuously. It continues to repeat the operation until it is accepted.

Holding the path in this manner assures that priority requests will be completed promptly. Bus bandwidth may be momentarily reduced because of the idle cycles. In this case, however, the system's performance is limited by the memory's bandwidth rather than the bus's.

3.5.2 Bus Fault Acknowledge

The fault acknowledge is sent when a slave detects a parity error in any field of an item addressed to it. This represents a hardware malfunction. Both the master and slave should inform the CDS of all faults. Processors take machine malfunction interrupts. Error recovery is determined by software. The S-bus interface circuits should not automatically retry operations which cause a fault.

When a fault is detected on an address word, the slave aborts the operation. If a memory module detects a fault on a memory write data word, that word is not written, but any preceding word is written. Any subsequent data words are ignored.

3.5.3 Parity Bits

Parity bits accompany the data, function and ID fields. Parity detects errors which alter an odd number of bits; it cannot affect errors which alter an even number of bits. These bits are generated by the master and checked by all slaves. A parity

error is detected if the parity check bit differs from the bus parity bit.

If the addressed slave detects a parity error, it ignores the data item and sends a 'fault' acknowledge to the master. If the error alters the ID field or memory address, the intended may not respond. If the altered field selects another slave, that slave will report the error.

Parity bits are numbered corresponding to power-of-two notation, which is the reverse of the memory byte numbering convention. For instance, parity bit 3 covers bits 31:24 (byte 0).

3.5.4 Bus Time-Out

All operations on the S-bus are deferred response. The master releases the bus after issuing a command; the immediate acknowledge assures that the command was received. The master waits for a response after the operation is done. However, the master does not hang indefinitely if the response never occurred due to a hardware fault. The master contains time-out logic which handles missing responses.

Memory operations should complete within 25 s. Longer delays are faults. Compatible I/O operations should allow 50 s; this allows time for the DMI to detect a false-sync after 30 s. False-sync is a time-out on the multiplexor (MUX) bus when a nonexistent device is addressed; it is signalled by an input/output error response (IER). False-sync does not indicate a hardware fault and it does not cause an S-bus time-out.

CHAPTER 4
MEMORY OPERATIONS

4.1 INTRODUCTION

System bus (S-bus) memory comprises one or more memory modules, each containing a controller and an array of semiconductor memory chips. Modules are selected by memory address using decoders on each board. Different size and speed modules can be mixed on the same bus.

There are three types of memory operations: read, write and special. The special operations combine a read-modify-write operation into a single primitive command.

4.2 INTERLEAVING

Burst transfers are divided into quadword block transfers. Each module interleaves four ranks of memory to match bus bandwidth for a quadword transfer. However, one module cannot match the bandwidth of both buses. Byte, halfword and special operations are slower than bus speed.

Improved bandwidth is possible when there are two or more modules to share the work. These modules share the bus most effectively when bus operations alternate between modules. Thus, two or four modules can be interleaved by quadwords. Burst transfers access each module cyclically, with successive quadwords sent to different modules in turn. Module interleaving is determined by the address decoders. Memory is configured using the control/diagnostic system (CDS).

4.3 MEMORY MODULE QUEUES

Each module contains an input first-in/first-out (FIFO) buffer. This queue stores addresses and data received while the module is busy performing previous operations. The queue improves performance by allowing more effective sharing. The buffer size is limited to queue four read or write operations. This provides adequate access time for high-priority read operations. (Operations already in the buffer are completed first.)

Each memory module may also contain an output queue for data sent on the From-path (F-path). Memory read data is loaded into this queue queue transmission on the F-path as soon as the module gains control of the bus. Meanwhile, the memory can continue

with subsequent operations. This improves performance especially for intermixed reads and writes.

4.3.1 Preserving Memory Operation Sequence

Memory operations are logically performed in the order they appear on the To-path (T-path). The input buffers on each memory module must preserve the sequence of operations to any given memory location. Otherwise, memory values could be uncertain, and cache invalidate logic could not operate precisely.

Operations to different addresses may proceed in any order; the queue on any module is independent of the other modules. In particular, when a high-bandwidth unit performs concurrent quadword reads to overlap access times, the data may come back out of order from interleaved memory. The unit must use unique IDs for concurrent read operations in order to identify the data on the F-path.

4.3.2 Address Space

32-bit memory addresses are transferred on the T-path. This provides 4GB of real address space.

For compatibility, some processors and input/output (I/O) units use 24-bit real memory addresses. The high 8 bits of their addresses will be zero. These units are limited to the first 16MB of memory.

4.4 T-PATH AND F-PATH USAGE FOR MEMORY

Processors initiate memory operations by sending addresses and data on the T-path. This is the only function of the T-path. Read data is returned by memory on the F-path. The F-path is also used for communication between processors.

4.5 MEMORY READ OPERATIONS

The four memory read commands (MR1, MR2, MR3, MR4) read from one to four words from memory. MR1 reads one fullword; it can be used for byte and halfword reads as well. Bytes and halfwords have the same alignment on the bus as in memory. The byte-select bits of the address (A01:00) are ignored by the memory modules. Memory reads do not alter the contents of memory.

MR4 reads an entire quadword. MR2 and MR3 read two or three words from the same quadword. Words are selected by memory address bits 03:02. The addressed word is transferred first; the remaining words are transferred in cyclic order: W0, W1, W2, W3, etc. (W0 is the addressed word.) The 2-bit word address is returned with each word as part of the F-path function code.

Processors use MR4 to fill quadword blocks in cache buffers. Normally, the processor waits for the addressed word when it suffers a cache miss. This word is transferred first. The processor accepts this word and resumes operation while the quadword is written into cache.

4.5.1 Memory Read Sequence

Figure 4-1 depicts the sequence of a read single-word (MR1) operation.

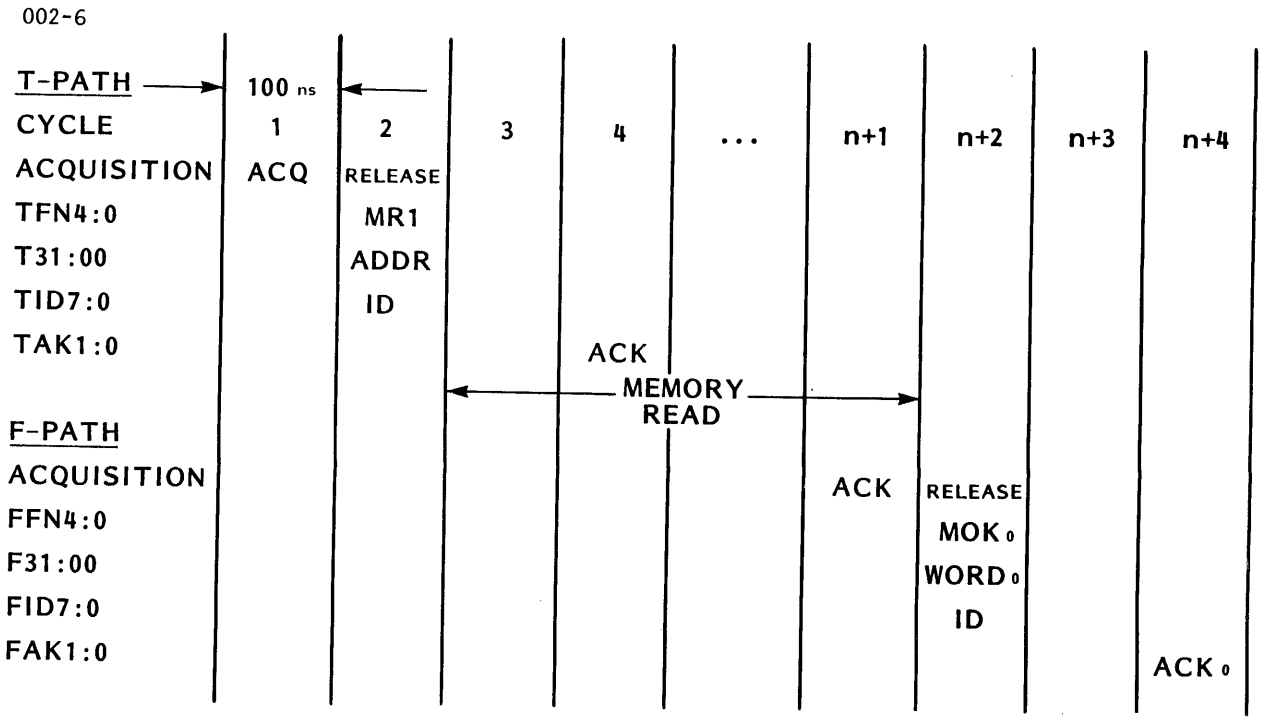


Figure 4-1 Read Singleword (MR1)

CYCLE

1. The unit acquires the T-path as described in Section 3.3.

2. ADDRESS CYCLE: The unit
   - puts a read function on TFN4:0 (MR1, MR2, MR3, MR4)
   - puts its unit ID onto TID7:0
   - puts the memory address onto T31:00

   All memory modules latch T-path data during every cycle. The unit releases the bus; only the address is sent.

3. Each module decodes the memory address to determine if it contains the addressed word. If not, it takes no further action. Parity is checked. The selected module begins the memory operation.

4. The selected memory module acknowledges the message. The module reads the requested data from the memory array.

n+1 The memory module requests the F-path.

n+2 DATA CYCLE: The module
- puts its response function on FFN4:0
  MOKn or MDCn indicates the data is valid
  MERn indicates an uncorrectable memory error
  'n' is the word select (memory address bits 03:02)
- puts the unit ID latched in step 3 onto FID7:0
  (This addresses the unit which requested the read.)
- puts a data word on F31:00

The module releases the F-path during the last transferred data word. Usually, all requested words are transferred on consecutive cycles. However, memory data errors can impede or interrupt the transfer. The memory controller can request the F-path before it has checked its validity. If there is an error, the next cycle is wasted; zero FFN4:0 selects an idle bus cycle. The corrected data word is sent on a later cycle.

Each unit latches data on the F-path during every cycle.

n+3 Each unit matches FID to its own ID. If unequal, the unit takes no further action. The selected unit checks the data for proper parity. (It now has the data it requested.)

n+4 The unit acknowledges the item with either Here or Fault. Units do not respond busy (or not present) to the memory operations they request because the memory module cannot repeat the data. Any response other than Here is a malfunction.

For each additional word accessed, cycles n+2 through n+4 are repeated. For example, an MR4 transfers a quadword on the F-path as shown in Figure 4-2.

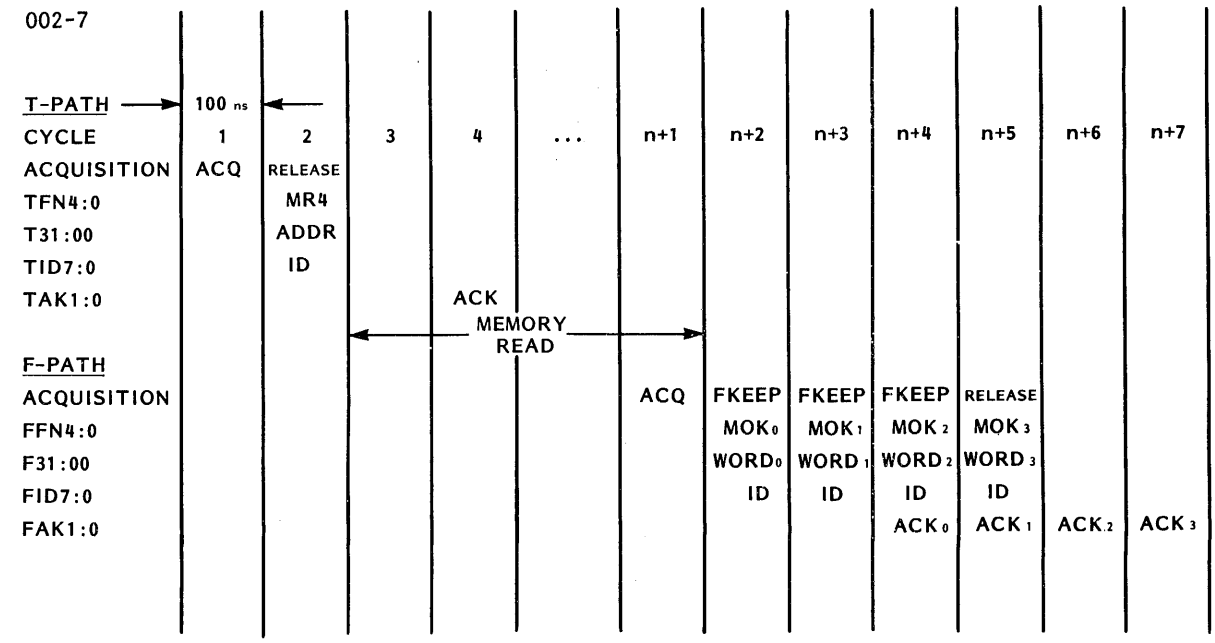


Figure 4-2 Read Quadword (MR4)

4.5.2 Memory Read Responses

Memory sends read data back to the requesting unit on the F-path. Three sets of four function codes are used: (See the previous description for other details.)

MOKn Memory OK - memory data was read correctly. It may be used and retained in cache.

MDCn Memory Disable Cache - memory data was read correctly. It may be used but must not be retained in cache.

MERn Memory Error - memory data was read with a permanent noncorrectable error. The data is not used or retained in a cache.

'n'=0, 1, 2 or 3 indicates the word within a quadword block. 'n' is bits T03:02 of the word's address on the bus.

4.6 MEMORY WRITE OPERATIONS

Memory write operations write bytes, word or multiple-word data into main memory. The data word(s) immediately follows the address on the T-path. The F-path is not used.

4.6.1 Memory Write Sequence

The sequence of a write operation is shown in Figure 4-3.

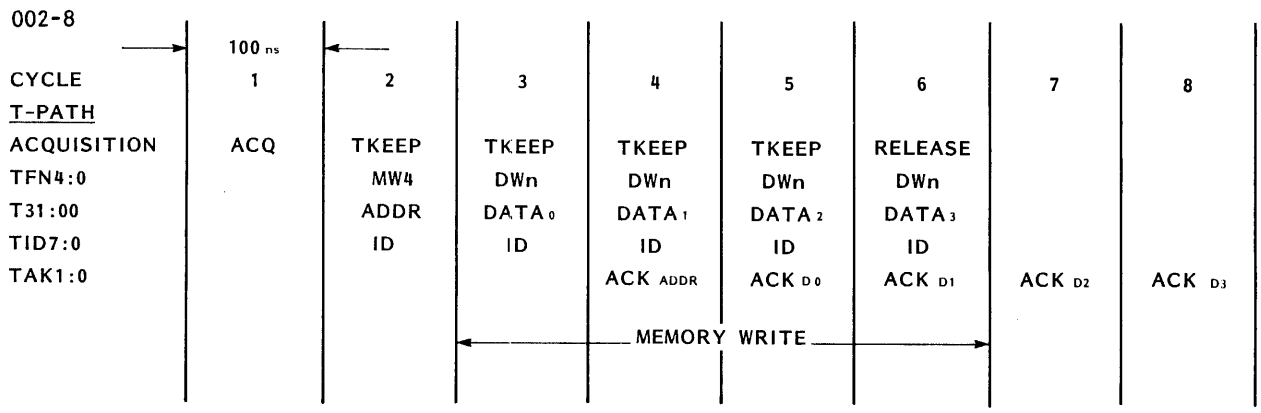


Figure 4-3 Quadword Write (WR4)

D2, D3 and D4 are used only in quadword operations.

CYCLE

1. The unit acquires the T-path as described in Section 3.3.

2. ADDRESS CYCLE: This is the same as in a read operation, but the function code specifies a write: MW1, MW2, MW3, or MW4. The unit retains the bus by asserting TKEEP.

3-6 DATA CYCLES: The unit
- puts write data onto T31:00
- puts its unit ID onto TID7:0
- puts a data code on TFN4:0 (DW0, DW1, DW2 or DW3)

The selected module acknowledges the address and data words.

During the last data cycle of normal memory writes, function DWn specifies that writing ends on byte 'n'. This is used for byte, halfword or other partial-word writes. (See Table 4-1.)

Function code DW3 must be used for other write data cycles and for all special memory data on T-path.

Meanwhile, each module decodes the memory address to determine if it contains the addressed word. If not, it takes no further action. Parity is checked. The selected module begins the memory operation.

7-8 The selected module acknowledges the last two data words.

For MW1, MW2 and MW3, only 1, 2 and 3 (respectively) data cycles are used to write data into memory. Each item is checked and decoded on the cycle after it is sent, then acknowleged on the following cycle: 5, 6 and 7.

The diagnostic write (MWD) operates as a 1-word write (MW1), but the 7-bit error correction bits (ECC) are not altered. Only the 32-bit data field of the 39-bit memory word is written. Therefore, single and multiple bit errors can be simulated to test the ability of each memory module to detect and correct memory errors.

4.6.2 Multiple-Word Write Operations

The multiple-word operations (MW1-MW4) write a string of data into a memory quadword. The string begins at the addressed byte. Lower bytes (to the left) in this word are not written. During the last word, the function code TFN4:0 selects the last byte to be written.

Words are cyclic within a quadword. Writing begins at the addressed word and continues in cycle order: 0, 1, 2, 3, 0, 1, 2, etc. The entire quadword can be written starting with any word.

4.6.3 Partial-Quadword Write Operations

Partial-quadword writes are useful for string operations and for burst transfers which begin or end on nonaligned addresses. Data bytes are aligned on the bus as in memory. Unused bytes are ignored except for parity checking. When only one word is transferred, both the address and TFN limit which bytes are written. See Table 4-1 for additional details.

TABLE 4-1 FIRST AND LAST WORD CONTROL

FIRST WORD CONTROL		LAST WORD CONTROL	
ADDRESS	BYTES WRITTEN	TFN40:00	BYTES WRITTEN
T010:000 = 00	0,1,2,3	DW0 00100	0
01	1,2,3	DW1 00101	0,1
10	2,3	DW2 00110	0,1,2
11	3	DW3 00111	0,1,2,3

Table 4-2 combines the two tables above for a single-word.

TABLE 4-2 BYTES WRITTEN FOR SINGLE-WORD WRITE (MW1)

ADDRESS (T1:0)	FUNCTION CODE WITH DATA WORD DW0	DW1	DW2	DW3
...00	0	0,1	0,1,2	0,1,2,3
...01	none	1	1,2	1,2,3
...10	none	none	2	2,3
...11	none	none	none	3

4.6.4 Byte and Halfword Operations

Byte and halfword writes are performed as writes with one data word (MW1). They are special cases of partial-word writes, as described previously. The memory address specifies the first byte written; DWn specifies the last byte. This is shown in Table 4-3. The other bytes are not altered. Bus timing is the same as a fullword write; memory timing uses longer read-modify-write cycles because the error correction codes are computed on fullwords. Each byte to be written must be properly aligned within the data field; the memory controller does not shift data bytes. Unused bytes are ignored.

TABLE 4-3 BYTE AND HALFWORD OPERATIONS

BYTE OPERATIONS			HALFWORD OPERATIONS			
BYTE	T01:00	TFN4:0	HALFWORD	BYTES	T01:00	TFN4:0
0	00	00100 DW0	0	0,1	00	00101 DW1
1	01	00101 DW1				
2	10	00110 DW2	1	2,3	10	00111 DW3
3	11	00111 DW3				

4.7 SPECIAL MEMORY OPERATIONS

Special memory operations facilitate multiprocessor interaction. These operations are indivisible primitives which operate as read-modify-write cycles. They combine write and read data transfers on the S-bus. The original unmodified memory value is read and sent on the F-path. Several types are supported:

1. Bit operations (MRS, MRR) set or reset any one bit within a data word. The number of this bit is selected by the write data value, T04:00. For instance, MRS is used to set reference and dirty bits when accessing the virtual address translation (VAT) tables.

2. Exchange words (MRW) swaps a word with memory. The addressed word is read and returned on the F-path; then the T-path data word is written.

3. Semaphore operations (MRI, MRD) increment or decrement the addressed word. The T-path data word is not used. (By convention, all write messages are at least two words long. Write invalidation hardware within cache memory buffers may need two cycles to test each address and invalidate entries.)

4.7.1 Special Memory Sequences

The sequence of any special operation is shown in Figure 4-4.

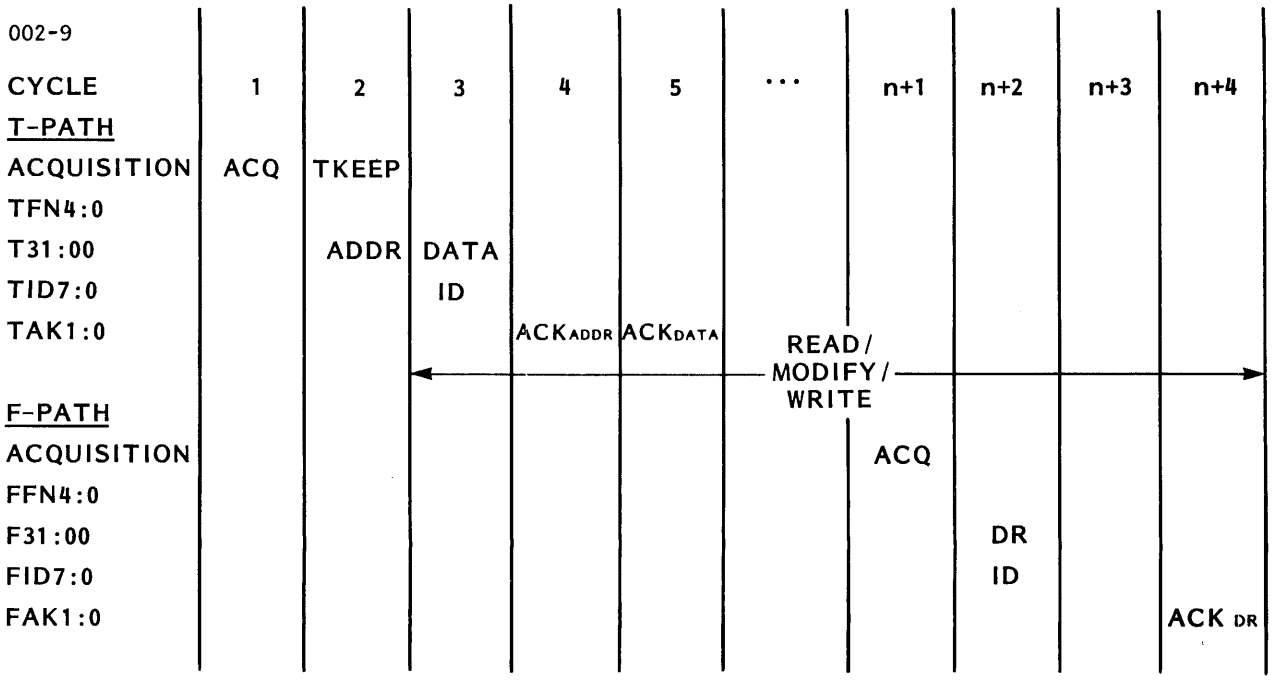


Figure 4-4 Special Memory Operations

See memory read and write sections for detailed descriptions. The T-path operates as a memory write; the F-path operates as a memory read.

4.7.2 Test-and-Set Operations

Test-and-set memory operations are used to coordinate activity within multitasking systems. These provide race-proof interlocks for allocating access to devices, tables or other resources. The memory system performs an indivisible operation which: reads the interlock word, tests the specified bit, then sets it and writes the word. This function is performed by the MRS command.

However, test-and-set should not simply use MRS. This may heavily load the S-bus and memory system. Frequently, routines are programmed to spin on an interlock; the processor repeats test-and-set until the interlock opens. Although spinning is usually limited to short delays, this can result in many unnecessary bus operations.

Instead, processors should first read the interlock word using their cache. If the tested bit is already set, there is no reason to do an MRS operation. If the processor spins, it uses only its cache and does not waste bus cycles. The interlock is later opened by a memory-modify operation which invalidates the cache block. The processor must access the new contents to refill its cache. If the bit is reset, the processor then does the MRS operation. Usually, the bit is still reset, and the test-and-set loop is finished.

CHAPTER 5
DIRECT TRANSFERS ON F-PATH

5.1 INTRODUCTION

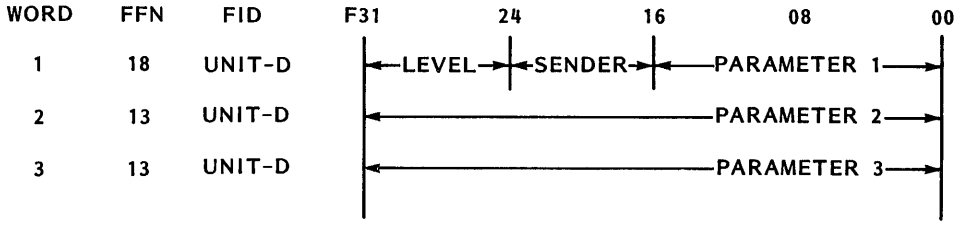
In addition to memory read data, the From-path (F-path) is used for interprocessor messages, broadcast interrupts and direct input/output (I/O). The system bus (S-bus) message allows a unit to send an interrupt or a few data words to a specific unit. Broadcasts are sent simultaneously to all units, including the unit which sends it. Direct I/O can be used for transferring commands from a processor to a channel or a simple I/O device.

Direct I/O is also used to implement the Series 3200 I/O subsystem. This allows a processor to communicate with a direct memory interface (DMI) board.

5.2 INTERPROCESSOR MESSAGES

Interprocessor messages transfer interrupts and three data words from any processor or channel to another designated unit. Messages are used to coordinate processors in multiprocessor systems, to initiate channel operation and to reschedule tasks when I/O operations are completed. Channels perform I/O tasks according to channel control blocks (CCBs) stored in main memory. CCBs select devices, functions, memory buffers, etc. The processor sends a message to initiate operation of a CCB. The message includes the CCB's (real) address in main memory. The channel accesses the CCB using memory operations and performs the required functions. Data is transferred directly to or from memory. The channel can chain data buffers and commands; it reports its status to the processor by sending messages like the following.

002-10



Start I/O commands a channel to initiate a CCB.
Halt I/O commands a channel to halt execution of a CCB.
End I/O interrupts a processor after a CCB completes.
Processor interrupt sends a message between processors.

FID7:0 contains the ID of the intended destination unit. Sender and unit-d are source and destination units. Device is the device address. (See Section 2.3.2 for further details.)

Both source and destination units are processors.

The priority level indicates the relative urgency of the CCB operation or interrupt. (0 is highest priority.) Parameter 1 is a halfword; parameters 2 and 3 are fullwords. Their meaning is defined by software convention.

5.2.1 Message Sequence

Figure 5-1 shows the sequence of a message.

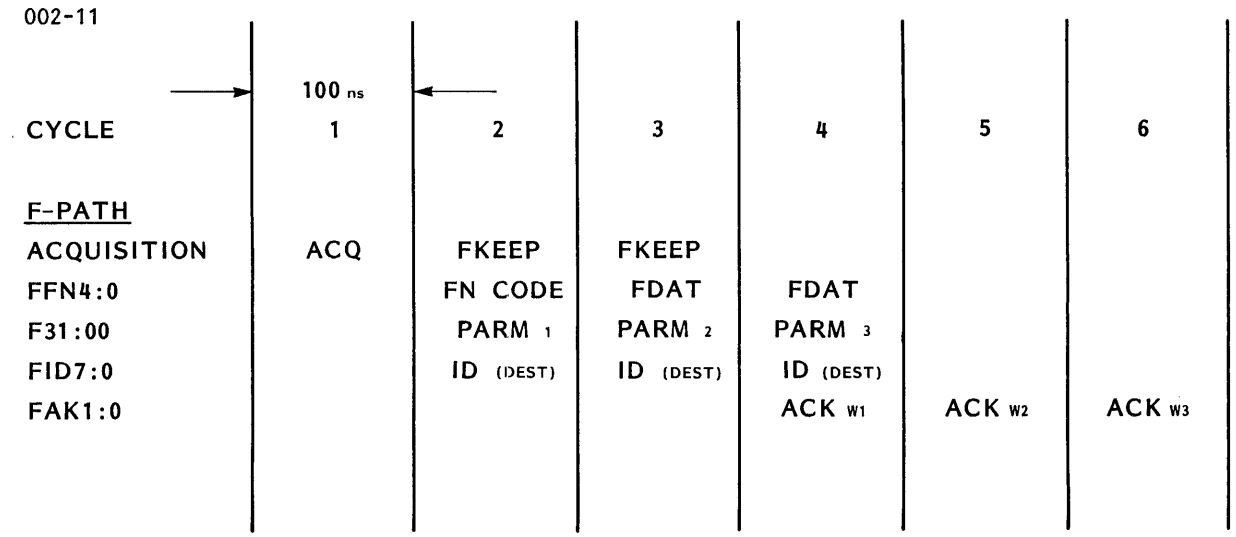


Figure 5-1 Message Sequence

1. The source unit acquires the F-path as described in Section 3.3.

2. WORD 1 (address): The unit
   - puts function code (MSGn) on FFN4:0
   - puts the destination unit ID on FID7:0
   - puts priority, its ID and parameter 1 on F31:00

   All units latch F-path data on every cycle.

3. WORD 2 (data): The source unit
   - puts function code (FDAT) on FFN4:0 (data word)
   - puts the destination unit ID on FID7:0
   - puts parameter 2 on F31:00

4. WORD 3 (data): The unit
   - puts function code (FDAT) on FFN4:0 (data word)
   - puts the destination unit ID on FID7:0
   - puts parameter 3 on F31:00

   The destination unit acknowledges word 1.

5. The destination unit acknowledges word 2.

6. The destination unit acknowledges word 3.

5.3 BROADCAST INTERRUPTS

Broadcasts are interprocessor interrupts which are sent simultaneously to all processors within an S-bus system. Processors receive all broadcasts regardless of their ID. Three broadcast interrupts are defined: preempt (PRE), delete virtual address (DVA) and start real-time clock (SYNC). A fourth broadcast code is reserved.

Broadcasts are not acknowledged on the F-path. Thus, processors must use the broadcast when it is received. They cannot respond with a busy acknowledge. Since DVA can require several cycles, the operating system guarantees that no subsequent DVA occurs for several cycles. There is no problem if only one process can issue DVAs.

The unit which sends a broadcast should identify itself in the FID field. This field is not used for the broadcast but can be monitored by test equipment.

Figure 5-2 shows the sequence of a broadcast interrupt.

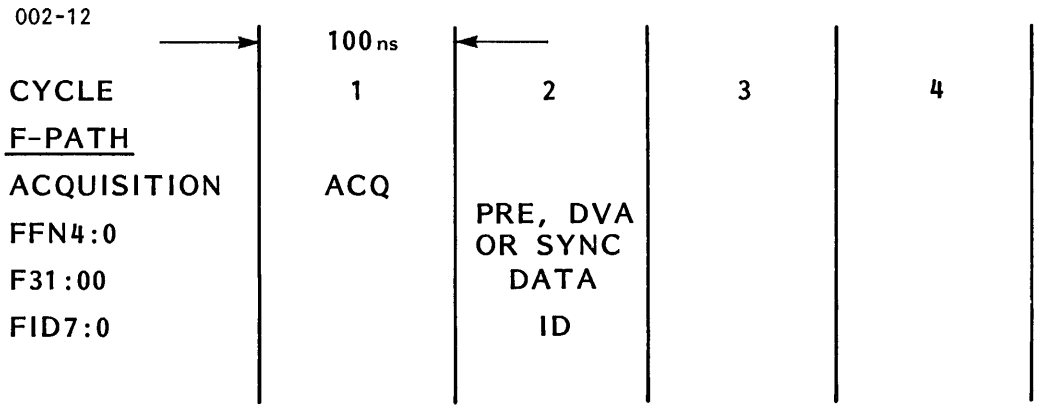


Figure 5-2 Broadcast Interrupt

1. The source unit acquires the F-path.

2. BROADCAST: The unit
   - puts function code (PRE, DVA or SYNC) on FFN4:0
   - puts its unit ID on FID7:0
   - puts broadcast messaage on F31:00 (see below)
   All units latch F-path data on every cycle.

3. (All processors decode the broadcast message.)

4. (Broadcasts are not acknowledged.)

5.3.1 Broadcast Preempt (PRE) Interrupt

The preempt interrupt is an F-path broadcast which is sent from any unit to all processors within an S-bus system. A preempt is issued by a unit after it has added a process to a previously empty ready-queue. The interrupt signals each processor to inspect its queue. A processor which finds a higher priority process will preempt its current process and perform a context switch to the new process.

Preempt is a 1-word message whose format is:

```
FFN      FID      F31...24......16......08.....F00
x14    sender   Priority|   unused     | Mask  |
```

Where:

FFN x'14	indicates a preempt interrupt.
Sender ID	identifies the unit which sent the interrupt. This ID is monitored only for diagnostic purposes.
Priority	F31:24.
Mask	F07:00.

5.3.2 Preempt (PRE) Priority and Mask

The priority and mask fields are decoded by the processor's bus interface hardware when the message is received. The interface either accepts or ignores the interrupt. If ignored, no action is taken; no software overhead is incurred.

The priority indicates the relative urgency of the process added to the ready-queue. Priority 0 is the most urgent. The processor compares this field to its extended status word (ESW) priority. The interrupt is ignored unless F31:24 < ESW 31:24.

The processor mask contains a bit for each processor group 0-7. The interface inspects the bit selected by its group number ESW 02:00. The interrupt is ignored if that bit is 0.

If the preempt is accepted, the preempt interrupt pending (PIP) bit is set. No other information is conveyed. The processor responds to this interrupt when there are no higher priority interrupts in progress. The processor inspects its ready-queue. If required, a context switch is substituted for the normal return from interrupt.

An idle processor responds more quickly than one which must interrupt a process. Otherwise, unnecessary context switches may increase overhead losses. The hardware does not attempt to order responses by priority, however.

No information is associated with a preempt interrupt. The priority and mask fields are discarded after the initial comparisons are made. Only one preempt interrupt is queued; any new preempts are ignored while PIP is set.

5.3.3 Broadcast Delete Virtual Address (DVA)

DVA is an F-path message which is broadcast to all processors in an S-bus system. DVA is issued by the page handling routine whenever it modifies a virtual address translator (VAT) table entry whose reference bit is set. Any outdated copy of this entry must be deleted from the VAT buffers of all processors. Every processor must check its VAT buffer.

DVAs are not recognized by DMI boards. The operating system insures that all required memory pages are present in main memory before an I/O operation is initiated. These pages cannot be discarded or paged to secondary memory until the operation has been completed.

DVA is a 1-word message whose format is:

```
FFN      FID      F31...24......16......08.....F00
x15      sender   |Virtual address to be deleted |
```

Where:

FFN x'15'	indicates a DVA.
Sender ID	identifies the unit which sent the interrupt. This ID is monitored only for diagnostic purposes.

A processor must retranslate the virtual page address if it uses a virtual page after its entry has been deleted. This procedure sets the reference bit in the table entry.

The operating system provides each user task with its own virtual memory address space. Virtual addresses are not unique. However, the virtual address is deleted from all buffers without checking whether the processor is using the affected domain. Such checking would considerably complicate how DVA is handled. As a consequence, some entries may be unnecessarily deleted from a processor's buffer and must be reloaded.

5.3.4 Start Real-Time Clock (SYNC)

SYNC is used to synchronize clocks within a multiprocessor system. Every processor includes a 64-bit real-time clock which keeps precise time by counting system bus cycles. Once initialized, all clocks within the system will keep exactly the same time. SYNC does not send any data; bits F31:00 are ignored.

The following procedure is used to initialize clocks:

1. One processor sends messages to all processors instructing them to stop their real-time clocks and to reload them with the 64-bit value contained in parameters 2 and 3.

2. That processor waits. For a system using 3280 processors, a delay of 100 microseconds is adequate. This provides time for each processor to acknowledge its message interrupt and to reload its real-time clock.

3. That processor issues SYNC on the F-path. This broadcast message is decoded by hardware within each processor, and all processors start their clocks within two cycles. This step provides precise synchronization.

SYNC is a 1-word message whose format is:

```
FFN      FID      F31...24......16......08.....F00
x16     sender    ................................
```

Where:

FFN x'16'	indicates SYNC.
Sender ID	identifies the unit which sent the interrupt. It is required only for diagnostic purposes.

5.4 DIRECT INPUT/OUTPUT (I/O)

Direct I/O allows a processor to read or write a word from a selected device. Since the processor must wait for a response, these devices must be fast, otherwise processor performance is

degraded. Direct I/O can also be used for communicating with test equipment.

The processor can either read or write data to a selected device. The read command, IOR, sends one word containing a subfunction and a device address. The write command, IOW, sends two words; the second contains data. The destination unit responds with either an IOK (operation was completed successfully) or an input/output error response (IER). For read operations, the data is returned with the IOK.

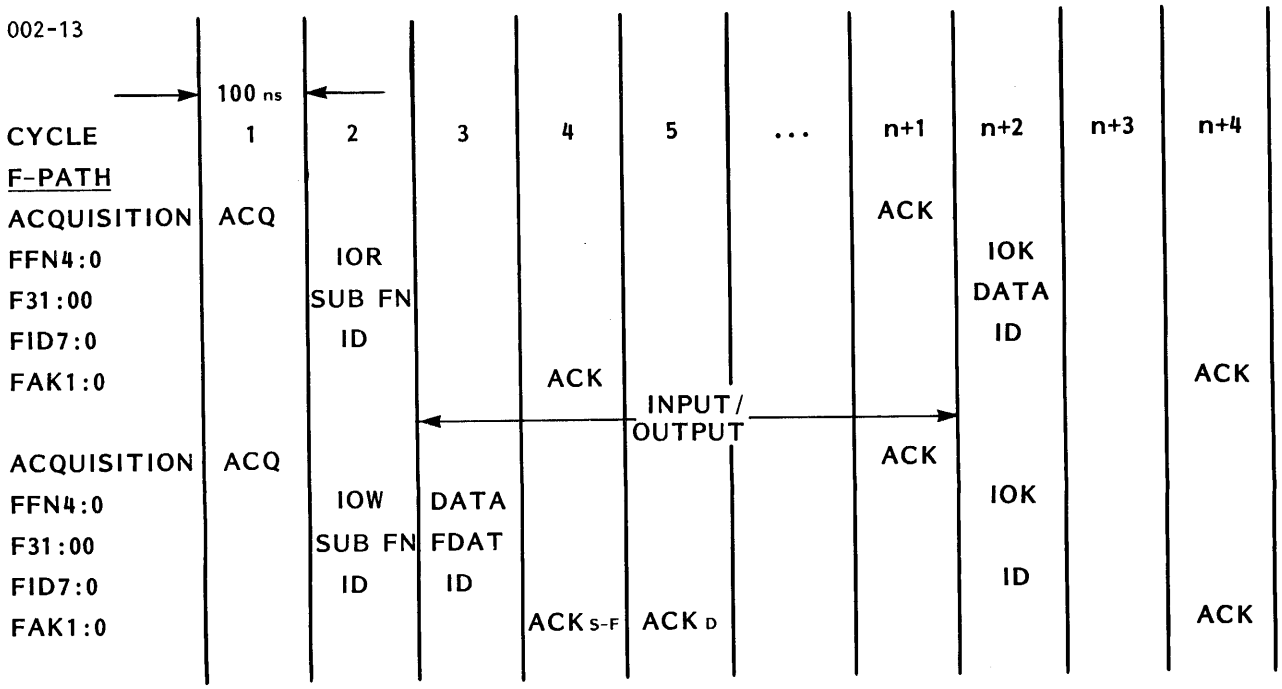


Figure 5-3 Direct I/O

1. Processor acquires the F-path.

2. SELECT CYCLE: the processor
   - puts IOR or IOW on FFN4:0
   - puts destination unit ID on FID7:0
   - puts subfunction/device on F31:00
     F31:28 (4 bits) subfunction
     F27:24 (4 bits) not used
     F23:16 (8 bits) sender unit ID number
     F15:00 (16 bits) device number

3. DATA WRITE CYCLE: (IOW only) the processor
   - puts FDAT on FFN4:0 "data word"
   - puts destination unit ID on FID7:0
   - puts write data on F31:00

4. Destination unit acknowledges select cycle.

5. Destination unit acknowledges data write cycle (IOW only).

   (I/O unit does a data transfer.)

n+1 DMI acquires the F-path.

n+2 IOK or IER CYCLE: the device unit
- puts IOK or IER on FFN4:0
- puts processor unit ID on FID7:0
- puts data on F31:00 (response to IOR only)

n+3 (Processor decodes and checks the item received.)

n+4 Processor acknowledges.

The I/O operation may simply read or write a register on the interface board. In this case, the operation is completed quickly. The device must not send the IOK or IER function until after it has sent the acknowledge to the original IOR or IOW function. That is, cycle n+2 must occur later than cycle 4 for reads or later than cycle 5 for writes.

5.5 SERIES 3200 INPUT/OUTPUT (I/O)

The Series 3200 I/O subsystem can be attached to an S-bus system using a DMI board. This board generates a Series 3200 multiplexer (MUX) I/O bus and a Direct Memory Access (DMA) bus. The DMI board is included in a configuration only if these buses are required.

The processor uses direct I/O operations to access the MUX bus through the DMI board. In previous machines, each processor generated a MUX bus directly. In S-bus systems, the processors communicate to DMI boards, which generate MUX buses for them. The MUX buses are assigned to processors by the control/diagnostic system (CDS). This has several advantages in a multiprocessor system. Because there are no hardwired connections, the system can be reconfigured without moving cables or changing straps; and the assignment of processors can easily be altered to adapt the system if some components fail.

5.5.1 Assigning Direct Memory Interface (DMI) Boards to Processors

The MUX bus of each DMI board must be assigned for the exclusive use of a single processor. This limitation arises primarily from the interrupt structure of the MUX bus. The DMI monitors four attention lines; these are echoed to the assigned processor whenever any line changes; the processor determines when to acknowledge each level. Devices are assigned attention levels based on priority.

A processor may control more than one MUX bus, if it includes separate registers for each set of attention lines. The 3280 processor can use two buses. An extra bit is appended to the device address (bit 10) to select either MUX bus 0 or 1. This bit is tested to select the bus ID of the desired DMI.

The processor must be able to distinguish between the attention interrupts from the two DMI boards. To do this, the processor uses a pair of IDs. The even ID is used with MUX bus 0; the odd ID with MUX bus 1. The DMIs direct their attention interrupts to the ID used; the low ID bit (FID0) steers the status to the corresponding register. (This bit is not decoded for other DMI responses.)

5.5.2 Series 3200 Input/Output (I/O) Subfunctions

The DMI generates a Series 3200 MUX I/O bus for use by an assigned processor. That processor controls the MUX bus using direct I/O operations on the S-bus.

The DMI board is assigned a pair of S-bus IDs. Direct I/O to the even ID implements compatible I/O; direct I/O to the odd ID is used to configure the hardware virtual address translation (HVAT). The HVAT is used for DMA bus operations.

TABLE 5-1 I/O SUBFUNCTIONS FOR IOR AND IOW OPERATIONS TO EVEN DMI ID

WITHOUT T31:28	WITH ADDRESS T31:28	R/W	SUBFUNCTION COMMAND
0000	1000		No operation
0001 SS	1001 SSA	IOR	Sense status (8 bits)
0010 OC	1010 OCA	IOW	Output command (8 bits)
0011 ACK	1011 *	IOR	Acknowledge interrupt
0100 WH	1100 WHA	IOW	Write halfword (16 bits)
0101 RH	1101 RHA	IOR	Read halfword (16 bits)
0110 WD	1110 WDA	IOW	Write data (8 bits)
0111 RD	1111 RDA	IOR	Read data (8 bits)

5.5.2.1 Multiplexor (MUX) Bus Subfunctions

For compatible I/O, each subfunction command is equivalent to an existing MUX bus operation. There are two versions of each command. If SFN3 is set (postfix A on mnemonic), a MUX bus address cycle (ADRS0 pulse) is sent to select a device. Otherwise, it is assumed that the device is already selected, and

the ADRS0 cycle is omitted. Address cycles are needed on all user instructions but can be omitted during some autodriver sequences.

The MUX bus uses a bidirectional 16-bit data path. Device addresses are limited to 10 bits. Individual device controllers transfer either 8 bits (byte devices) or 16 bits (halfword devices) for each data strobe. Bytes are right-aligned on the bus. The device asserts a halfword status control line HW0 to select 16-bit transfers.

The status, command, read and write functions result in SR0, CMD0, DR0 and DA0 pulses on the MUX bus. The status and command functions transfer one byte from or to the device. The DMI inserts the halfword status of the device in bit 15 of the status word returned on the F-path. The DMI issues two DR0 or DA0 pulses when a read or write halfword operation accesses a byte device. The more significant byte is transferred first. If a halfword is written to a byte device, the high byte is ignored. If a halfword is read from a byte device, the high byte is zero. On the F-path, data is right-aligned in an IOK word or in word 2 of an IOW operation.

5.5.3 Direct Memory Access (DMA) Bus Operations

The DMA bus performs memory transfers under the control of I/O selector channels (SELCHs). Up to eight SELCHs can share a DMA bus. The DMA bus has a 10MB per second bandwidth. The SELCHs and DMA bus use 24-bit memory addresses.

The DMA bus operates independently of the MUX bus or any processor. The S-bus is used only for memory access. Multiple DMA buses may be required for high I/O bandwidth systems; the extra MUX buses need not be connected.

5.5.3.1 Direct Memory Interface (DMI) Virtual Address Translation (VAT)

The 3280 processor implements VAT and 32-bit real memory addresses. Its paging hardware lets it dynamically load 4096-byte pages from disk. Because these pages may be loaded in any free page in real memory, they are likely to be scattered.

When the processor performs an I/O operation, it specifies a block of virtual addresses. The real memory addresses which correspond to this block may be broken into many separate pages. To access this block efficiently, the DMI includes an HVAT unit.

The DMI provides three address modes: direct, extended and paged. In direct mode, the DMA addresses are used unmodified. The data block must be contiguous and in the low 16MB of real memory. In extended mode, each 24-bit address from the DMA bus is added to a 32-bit offset. This mode allows access to addresses above the first 16MB region. The offset must be a

multiple of 4096. The data block must be contiguous in real memory, but there are no boundary or alignment restrictions. The maximum transfer length is 16MB. The offset points to the first page of the block. This offset is subtracted from the begin and end addresses which are loaded into the SELCH; the modified begin address is the block address modulo 4096.

In paged mode, the HVAT translates bits 20:12 of each DMA bus virtual address into a real address page (bits 31:12). The real pages may be anywhere in main memory. The SELCH is initialized the same as in extended mode, but the maximum transfer length is 2MB.

The HVAT has separate translation registers for each of the eight SELCHs supported on a DMA bus. All must operate in the same mode.

TABLE 5-2 I/O SUBFUNCTIONS FOR IOR AND IOW OPERATIONS TO ODD DMI ID

SFN3:0	R/W	SUBFUNCTION COMMAND
0100 IOWV	IOW	Write real address into HVAT RAM
0101 IORV	IOR	Read real address from HVAT RAM
1000 CNFV	IOW	Write DMI configuration
1011 IOCLR	IOW	Software reset of I/O subsystem

5.5.3.2 Hardware Virtual Address Translation (HVAT) Configuration Subfunctions

Three subfunctions are used to initialize the VAT random access memory (RAM) on the DMI. The fourth subfunction (IOCLR) allows the software to reset all I/O controllers connected to the DMI.

The configuration command (CNFV) selects one of three modes for the HVAT using bits 01:00 of the first word.

Mode	Addresses (DMA-to-S-bus)
00 Direct	Untranslated.
01 Extended	A 32-bit offset is added to each address.
10 Paged	Translation for each 4K-byte block.

The read and write commands, IOWV and IORV, access the 4096-word translation RAM. Word #1 selects the RAM location in bits 11:00. Bits 11:09 correspond to the eight DMA bus IDs; bits 08:00 select

a virtual page address; only page 0 is used in extended mode. The data word selects a 20-bit real page address in bits 19:00; the corresponding real address equals 4096 times the page number.

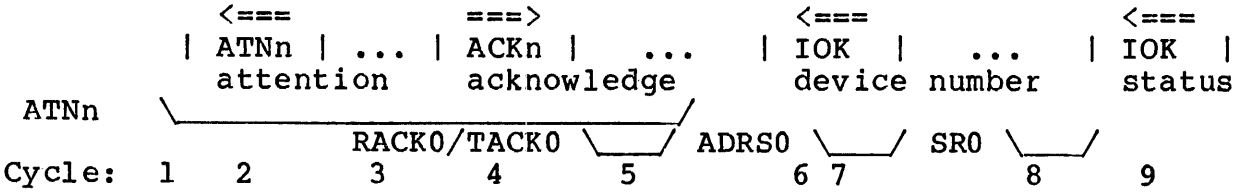
5.5.4 Attention Interrupt

The F-path attention (ATN) function is used by the DMI to signal external power-fail (XPF) or I/O attention interrupts (ATN3:0) to its processor. Each ATNn represents a separate interrupt priority line. ATN operates as IOK except that it occurs spontaneously whenever XPF or any ATNn line changes. The state of ATN3:0 is transmitted on F03:00.

The I/O system may include circuit boards which are powered independently of the S-bus. These external power supplies may be turned on or off or may lose AC power separately from the main S-bus power. This is signalled using an external power-fail signal, XPF0, on the MUX bus. This condition does not affect S-bus power or the three S-bus power clear signals. Instead, the state of XPF is sent to the processor on bit F04 of the attention item.

The processor acknowledges ATNn interrupts in a fashion that is hardware and program compatible with existing machines. The processor's program status word (PSW) determines which interrupt levels (i.e., which ATNn) are enabled. Other interrupts are left pending until the PSW is changed.

The processor begins an interrupt acknowledge sequence whenever it receives an ATN with an enabled interrupt or when the PSW is changed to enable a pending interrupt. This sequence is:



1. A MUX bus device asserts ATNn to request an interrupt.

2. DMI sends an ATN message to alert the processor.

3. The processor masks the ATNn lines with its current PSW. Masked interrupts are left pending.

   If any pending interrupt is enabled by the processor, then:

4. The processor sends ACKn to the DMI (as an IOR message). ('n' is sent in the device field: word 1, bits F01:00.)

5. The DMI sends a pulse down the RACK0/TACK0 daisy chain 'n' to acknowledge the device with the highest priority pending

interrupt. This device supplies its device address on the MUX bus, then resets its interrupt. ATNn will go inactive unless another device is requesting it. This change is signalled by another ATN message (not shown).

6. DMI sends this device address to the processor as an IOK message.

7. DMI sends an ADRS0 to address that device on the MUX bus.

8. DMI sends an SR0 to read the device's status.

9. DMI sends status to the processor as an IOK message.

The processor may continue running after issuing the ACKn. This reduces the time lost due to interrupts. However, the processor may not load a new PSW which disables the interrupt it just acknowledged. The processor can begin processing the interrupt as soon as it has the device address.

The DMI acknowledges the attention line specified by the processor even if a higher priority attention has become active. If the ATNn has been removed, the DMI will detect false-sync on the MUX bus and signal IER to the processor. IER ends the sequence.

CHAPTER 6
ELECTRICAL SPECIFICATIONS

6.1 SIGNAL TYPES

The system bus (S-bus) uses four types of signals. The logic signals are driven (OSC-, CLK-, TGRn-, FGRn-) or received (TRQn-, FRQn-) by circuits on the backpanel. Bus signals connect to all boards on the S-bus. The backpanel terminates the bus lines but has no logical connection to them.

Logic

1. OSC- and CLK- clocks are radially distributed.

2. Bus request TRQn-, FRQn- and bus grant TGRn-, FGRn- are individual advance Schottky TTL signals to each board.

Bus

3. Tristate bus signals:
   T31:00-, TFN4:0-, TID7:0-, TPAR5:0-
   F31:00-, FFN4:0-, FID7:0-, FPAR5:0-

4. Open-collector bus control signals:
   TKEEP-, TREQ-, TRREQ-, THREQ-, TAK1:0-
   FKEEP-, FREQ-, FRREQ-, FHREQ-, FAK1:0-

5. Power clear signals:
   MCLR-, PCLR-, SCLR-

Logic must conform to advanced Schottky TTL levels and loading:

Receivers:	Input-High	> 2.0V with I < 0.04ma
	-Low	< 0.8V with I < 0.6ma
Drivers:	Output-High	> 2.7V @ 1ma
	-Low	< 0.5V @ 20ma

6.1.1 Radial Clock Distribution

All S-bus timing is derived from a single 40MHz crystal oscillator as shown in Figure 6-1. This signal is divided by two to form OSC-, a 20MHz clock. OSC- is buffered and radially distributed to each board on the S-bus. Radial cables are used to minimize clock skew and waveform degradation. All signals are referenced to the falling edge of OSC- at the backpanel pin. Skew is less than ± 6ns between boards in a chassis and less than ± 10ns between chassis.

Onboard clock gating increases skew further. S-bus cycles allow for 10ns skew; only 90ns are available for data transfers.

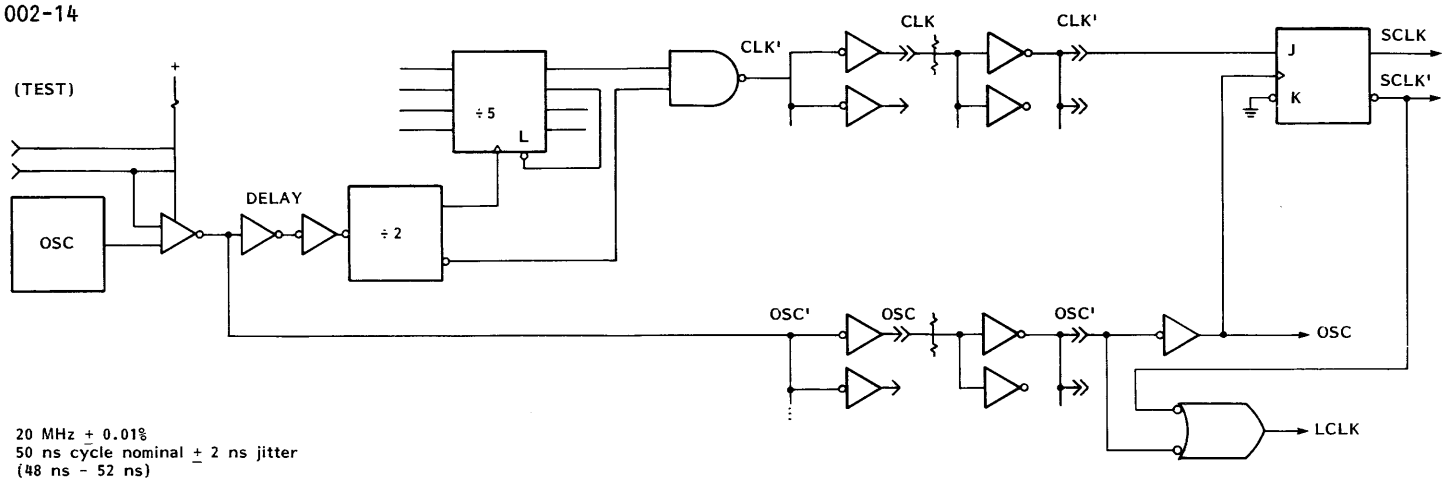
The oscillator on the S-bus terminator card can provide clocks to six chassis using coaxial cables and 4-pin connectors. To minimize skew, the electrical length of these cables must be equal.

The oscillator (OSC) is divided by 10 to create a 2MHz CLK signal. CLK is used to synchronize the bus at 100ns cycles (10MHz). CLK is a 50ns pulse, equal to one period of OSC.

OSC and CLK are buffered on the terminator and then on the backpanel to increase fanout. The backpanel receives the clocks with a 220/330 ohm termination. The signals are buffered with quad driver chips. Each gate drives OSC- to one board or CLK- to three boards. Traces are less than 6" on the distribution board. OSC- may be continued 10" to 15" onto the circuit board, but skew inside the board must be minimized. OSC- must be a continuous trace terminated with a 100-ohm pull-down resistor at its far end.

Each S-bus board may put eight loads on each clock. Trace stub length to the buffers must be minimized.

The clock (CLK) is wired to three boards. Each board must minimize its stub length, with a maximum of 1.5in on each board. Each board is allowed two loads.



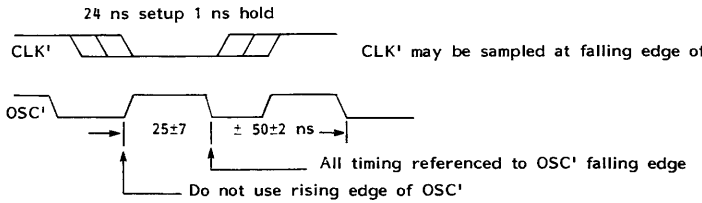
20 MHz ± 0.01%
50 ns cycle nominal ± 2 ns jitter
(48 ns - 52 ns)

		TERMINATOR CARD			BACKPANEL		LOGIC BOARDS				
CLK		'F04		'F112	'F00	'F04	CABLE	'F04	WIRE	F00	TOTAL (ns)
	MIN	2	2	3	2	2	4	2	1		18
	MAX	5	5	8	6	6	4	6	1		41
CLK-OSC skew	MIN	+9			-1	0	-1	0	-6		
	MAX	+24			+1	0	+1	0	-2	24	
OSC	MIN	2			4		2	1	2		
	MAX	6			4		6	1	6		
OSC-OSC skew		±4					±2		±4	±10	
							±2		±4	±6	
									±4	±4	

Times in ns are worst case over loading and temperature.

1 ns minimum hold time
24 ns - 48-24 minimum setup time

skew between chassis
skew between boards in same chassis
skew between adjacent boards
(assumes same OSC buffer.)



NOTES:
1. Clock cable and wire lengths are matched within 3" to minimize clock skews. (Not included in these calculations.)
2. Skew is difference in timing between separate clock signals, usually the difference between thin and max values. However, skew between circuits within a single package with similar loading is assumed less since there are fewer variations.
3. When two packages are used on a background or board, parts with the same date code should be used.
4. The 4 processor 'slots' should be driven by the same OSC' driver.

Figure 6-1 S-Bus Clock Distribution - Timing and Skew

6.1.2 Positional Priority Signals

Each positional priority encoder has one input (TRQn-, FRQn-) and one output (TGRn-, FGRn-) for each S-bus slot. Each circuit consists of three priority encoders and three decoders. Maximum delay is 20.5ns from any RQn- to any GRn-. Each RQn- is one unit load plus a 1K-ohm pull-up. Each GRn- can drive 20ma at 0.4V.

1K-ohm pull-up resistors hold unused inputs inactive (high). No jumper changes are needed when boards are inserted or removed. Request and grant timing is:

Delay SCLK to RREQ-, HREQ-	15ns max
Settling time	30ns max
Delay RREQ-, HREQ- to RQn-	10ns max
Delay RQn- to GRn-	20ns max
Setup time GRn- to LCLK	15ns max
Clock skew	10ns max
Total	100ns

6.1.3 Tristate Bus Signals

Each path uses 50 tristate bus signals. These are driven by the current master, synchronously with SCLK at 100ns cycles. These signals should be driven and received with edge-triggered D-type registers. Each slave receives every cycle and decodes its address or ID.

The master must drive the function (TFN or FFN) during every cycle that it has the bus. If it does not use the bus, it must drive the function code to zero (high); tristated lines do not rise fast enough. The other fields are ignored during idle cycles. They are allowed to float high to save power; the bus termination dissipates significant power only for low signals. They float high too slowly to guarantee an adequate logic high; parity is ignored. See Figure 6-2 for the timing of various S-bus signals.

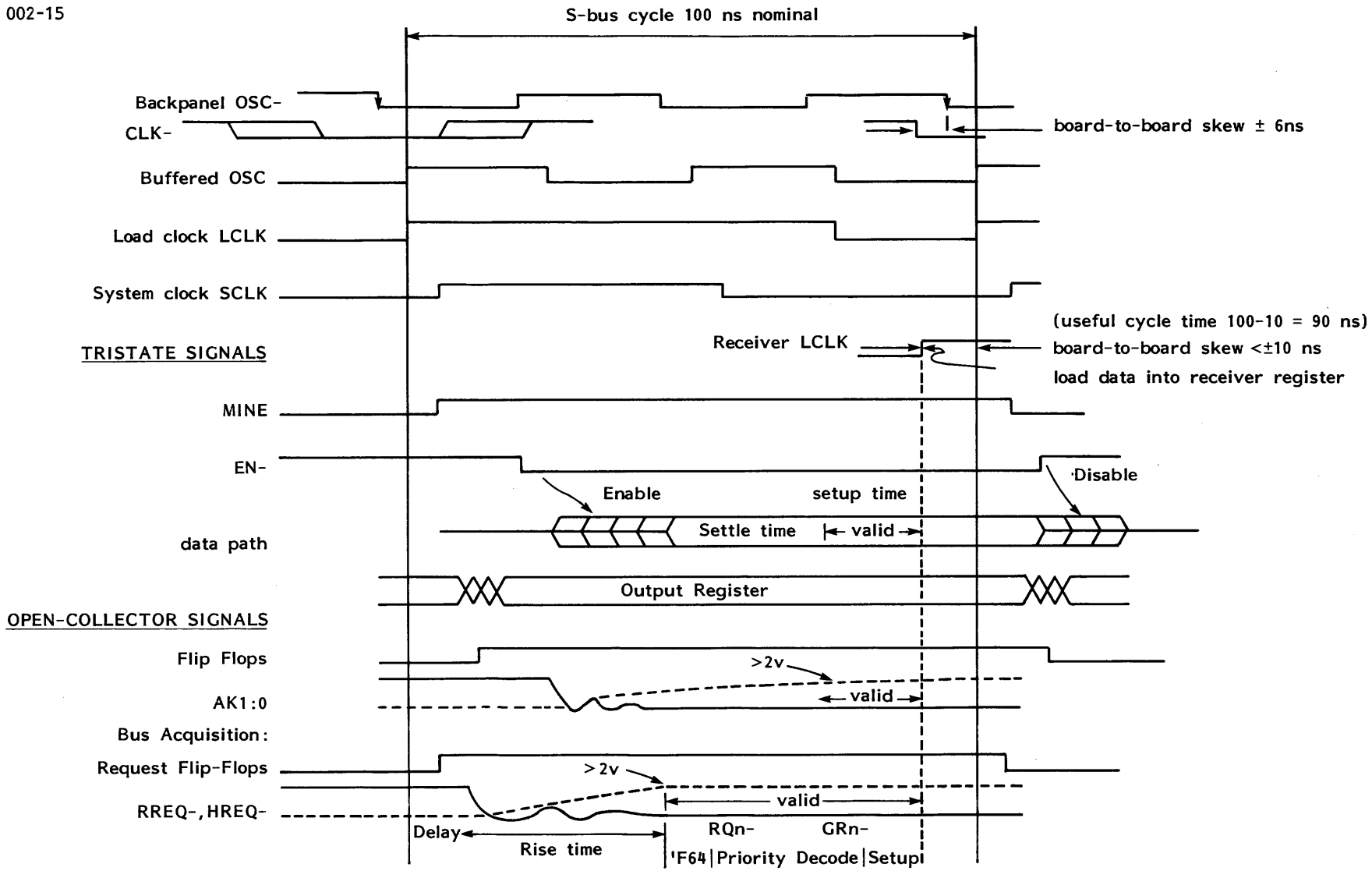


Figure 6-2 S-Bus Electrical Timing

6.1.3.1 Bus Termination

These signals are terminated through 150 ohms to 3V on the S-bus terminator card. The terminator uses a series-pass voltage regulator to produce 3V. This termination has AC response similar to a split resistor terminator, but dissipates little power for inactive (high) signals. On average, this halves current requirements to 580ma.

Terminator power for T-path or F-path:

50	tristate	x 20ma	=	1000 max	500 average
2	fast O.C.	x 35ma	=	70	35
4	slow O.C.	x 22ma	=	90	45
			Total	1158 max	580 average

In addition, a diode clamp to 0.6V reduces negative excursions. The bus is terminated at one end only. (Double-ended termination requires too much drive current and dissipates substantially more power.) Signal edges are absorbed by the terminator, but reflect at the far end. Thus, bus timing allows 20ns (> 2ns/ft x 1.5ft x 4) for settling.

Delay SCLK to drivers disabled	(15ns typ)
"	(25ns max)
Delay SCLK to drivers enabled	(30ns typ)
Delay SCLK to drivers enabled	45ns max
Bus settling time	20ns max
Bus setup time	25ns max
Clock skew	10ns max
Total	100ns

6.1.4 Open-Collector Bus Signals

Each path uses six open-collector signals: four for acquisition control, two for acknowledge. The control signals are driven by any board requesting the bus. Their outputs are logically wire-OR-ed; a signal is active low if any board asserts it. The acknowledges are usually driven only by the addressed slave. However, parity errors or improper configuration can result in several boards responding at once.

AK1:0, KEEP and REQ are terminated with 220 ohms to 5V on the S-bus terminator card. The 5V pull-up improves rise time to above the 1.6V logic threshold. A diode clamp to 0.6V reduces negative overshoots. The pull-up requires a maximum of 5.5-0.4V/220 = 25ma.

Delay SCLK to bus	30ns max
Bus rise time	40ns max
Bus setup time	20ns max
Clock skew	10ns max
Total	100ns

RREQ and HREQ control signals use 150 ohm 2% pull-ups for faster rise times. Buffer gates are required. The pull-ups require 5.5-0.5V/150 ohm-2% = 34ma maximum. These signals are faster since they control inputs to the priority encoder. The gates and the encoder require 28ns extra. (See Section 6.1.2 for timing.)

Each board is allowed 0.6ma load. 22 boards require 13ma maximum. HREQ drives two gates in the bus acquisition circuit. However, only one I-IL load is required since these gates have complementary enables: RREN and RREN-.

6.1.5 Power Clear Signals

The three power clear signals (MCLR-, PCLR- and SCLR-) are driven by the power supply controller (PSC). These signals are asynchronous to the system clocks and must be resynchronized on each board.

The PSC must be able to sink 24ma on each signal. Each board is allowed one load (0.6ma) per signal. Thus, 22 boards require a maximum of 13.2ma. These signals are terminated to 5V through 680 ohms and are diode clamped to ground on the backpanel terminator. The pull-up resistor uses 5.5-0.4V/680 ohm = 7.5ma. The pull-up resistors must be connected to P5U since the P5 may not be powered during battery backup. (The PSC requires a 5V high signal level since it monitors these signals using HCMOS logic.)

Bus Capacitance:

Bus connections have high-distributed capacitance. This affects signal transmission by lowering the characteristic impedance and slows the rise time on the open-collector signals.

Per board:	multiwire 3pf/in	x 3in =	10pf	x 1.5in =	5pf
	input	74F	5	74F	5
	output	74F	5	74F38	10
	connector, pins, etc.		5		5
	Total per board		25pf		25pf
	x 22 boards =		550pf		550pf
Backpanel:	multilayer 2pf/in x 17in =		35pf		
	Total per line		585pf		585pf

6.2 DESIGN NOTES

Synchronous buses provide simple reliable signal timing. Still, some pitfalls must be avoided.

6.2.1 Tristate Conflict

Tristate drivers provide cleaner, faster switching speeds than open-collector. However, severe noise problems can result whenever two opposing drivers are enabled simultaneously. A slight overlap of a few nanoseconds is acceptable if the drivers are properly decoupled. (Every driver should have an adjacent 0.1μf decoupling capacitor.) Any overlap increases system noise, but comparable current spikes occur when switching a high-capacitance bus. However, consistent overlaps greater than 5ns must be avoided.

This specification gives minimum and maximum values for most characteristics. However, delays are not predictable enough to use absolutes here. Instead, circuit timing eliminates tristate conflict only for typical delays.

The delay given for enabling a driver onto the bus is the specified delay for a 50pf load. The bus capacitance is much larger. Extra bus delay results from charging this capacitance. This time includes time for damping of the reflections which result from the imperfect impedence match at the terminator.

Device data sheets list enable delays which are usually longer than disable delays. However, this enable time measures when the output attains a valid logic level. By this time, the driver is already sourcing or sinking substantial current; its output is fully on. Furthermore, variations between devices and the timing differences between boards are significant. Each board must reduce tristate conflict by delaying bus enable vs. bus disable by 10ns. (The following table assumes a flipflop, a gate, a 10 $\pm$ 2ns delay and a registered bus driver.)

TABLE 6-1 BUS ENABLING AND DISABLING TIMING DELAYS

	FLIPFLOP	GATE	DELAY	ENABLE	TOTAL	
TURN-ON	8ns	6	12	16	42ns	max
	5ns	3	10	12	30ns	typ
	3ns	2	8	8	21ns	min
TURN-OFF	10ns	6	-	8	24ns	max
	5ns	3	-	6	14ns	typ
	3ns	2	-	4	9ns	min

6.2.2 Hold Times

Data is transferred from a source register to a destination register during the bus cycle. The worst case delays and necessary setup time determine the cycle time (100ns nominal). In effect, clock skew shortens the cycle to 100-10 = 90ns.

Hold times are more difficult since the clock skew is comparable to the minimum delay path. Extra timing margin is desirable. Input registers are clocked on every cycle. The input register clock, LCLK, is generated with a single gate delay. Output clocks are gated and may have an extra gate delay.

6.2.3 Clock Skew

OSC- is radially distributed to each board in the system. It has a nominal 50% duty cycle, ± 10% or 5ns. The high and low periods can nominally vary between 20 and 30ns. However, the pulse width becomes less certain with each buffer. Only the falling edge of OSC- may be used for timing.

CHAPTER 7
MECHANICAL SPECIFICATION

7.1 GENERAL DESCRIPTION

The system bus (S-bus) uses a 22-slot chassis of 16.62" x 15.4" boards. Boards are mounted vertically for optimal air flow. Each board has three 92-pin backpanel connectors. The To-path (T-path) is on connector0 (CONN0) and the From-path (F-path) is on CONN2. CONN1 is available for two 40-pin ribbon cables; four pins are deleted from each slot to allow mechanical tolerances for cable headers. See Figure 7-1 for the T-path and F-path pin-outs.

002-16 T-Path

	Connector0 row1	Connector0 row2	Connector1 row1	Connector1 row2		F-Path Connector2 row1	F-Path Connector2 row2
00	gnd	gnd	gnd	gnd		gnd	gnd
01	gnd	gnd	gnd	gnd		gnd	gnd
02	TRQn-	FRQn-				FRPSC	FRPSC-
03	TGRn-	FGRn-			c	TOPSC	TOPSC-
04	OSC-	gnd			a	SLOT0	SLOT1
05	CLK-	gnd			b	SLOT2	SLOT3
06	TKEEP-	FKEEP-			l	SLOT4	SLOT5
07	TREQ-	FREQ-			e	SLOT6	SLOT7
08	TRREQ-	FRREQ-				MCLR-	PCLR-
09	THREQ-	FHREQ-			A	SCLR-	TLFC-
10	gnd		gnd			gnd	
11		gnd		gnd			gnd
12	TPAR0-	TPAR1-				FPAR0-	FPAR1-
13	TPAR2-	TPAR3-				FPAR2-	FPAR3-
14	TPAR4-					FPAR4-	
15	TAK0-	TAK1-				FAK0-	FAK1-
16	TFN0-	TFN1-				FFN0-	FFN1-
17	TFN2-	TFN3-				FFN2-	FFN3-
18	TFN4-					FFN4-	
19							
20	gnd	gnd	gnd	gnd		gnd	gnd
21	TID0-	TID1-	***	***		FID0-	FID1-
22	TID2-	TID3-				FID2-	FID3-
32	TID4-	TID5-				FID4-	FID5-
24	TID6-	TID7-				FID6-	FID7-
25	gnd	gnd	***	***		gnd	gnd
26	T00-	T01-	gnd	gnd		F00-	F01-
27	T02-	T03-			c	F02-	F03-
28	T04-	T05-			a	F04-	F05-
29	T06-	T07-			b	F06-	F07-
30	T08-	T09-			l	F08-	F09-
31	T10-	T11-			e	F10-	F11-
32	T12-	T13-				F12-	F13-
33	T14-	T15-			B	F14-	F15-
34	gnd		gnd			gnd	
35		gnd		gnd			gnd
36	T16-	T17-				F16-	F17-
37	T18-	T19-				F18-	F19-
38	T20-	T21-				F20-	F21-
39	T22-	T23-				F22-	F23-
40	T24-	T25-				F24-	F25-
41	T26-	T27-				F26-	F27-
42	T28-	T29-				F28-	F29-
43	T30-	T31-				F30-	F31-
44	gnd	gnd	gnd	gnd		gnd	gnd
45	gnd	gnd	gnd	gnd		gnd	gnd

'***' These pins are deleted to make room for the cable connector.

'gnd' indicates signal grounds. Power supply connections for P5, P5U and ground are made using 23-amp bullet connectors. Pins are numbered: "Row Pin - Connector". (For example, OSC- is on 104-0.)

Figure 7-1 Pin-Outs for T-Path and F-Path

INDEX

Concurrent Computer Corporation

PUBLICATION COMMENT FORM

We try to make our publications easy to understand and free of errors. Our users are an integral source of information for improving future revisions. Please use this postage paid form to send us comments, corrections, suggestions, etc.

1. Publication number _____

2. Title of publication _____

3. Describe, providing page numbers, any technical errors you found. Attach additional sheet if necessary. _____

4. Was the publication easy to understand? If no, why not? _____

5. Were illustrations adequate? _____

6. What additions or deletions would you suggest? _____

7. Other comments: _____

From _____ Date _____

Position/Title _____

Company _____

Address _____

FOLD

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 22 OCEANPORT, N.J.

POSTAGE WILL BE PAID BY ADDRESSEE

Concurrent Computer Corporation
2 Crescent Place
Oceanport, NJ 07757

ATTN:
TECHNICAL SYSTEMS PUBLICATIONS DEPT.

FOLD

FOLD

STAPLE

STAPLE

9410