Advanced
Micro
Computers
A subsidiary of
Advanced Micro Devices

Am95/4005 MonoBoard Computer

User's Manual

REVISION RECORD	
REVISION	**DESCRIPTION**
01	Preliminary Issue
(4/27/79)	
A	Manual Released
(8/24/79)	
B	Manual updated to correct documentation errors.
(10/2/79)	
C	Manual Reprinted
(4/25/80)	

Publication No.
00680120

REVISION LETTERS I, O, Q AND X ARE NOT USED

Address comments concerning
this manual to:

ADVANCED MICRO COMPUTERS

Publications Department
3340 Scott Boulevard
Santa Clara, CA 95051

CORRECTION SHEET

The Am9555 Programmable Peripheral Interface has been replaced on the 95/4005 MonoBoard by the Am8255A Programmable Peripheral Interface. All references in this manual to the Am9555 should be considered as references to the Am8255A.

PREFACE

This manual provides general information, an installation and interface guide, programming information, principles of operation, and service information for the Advanced Micro Computers AMC 95/4005 MonoBoard Computer. Additional information can be obtained from the following documents.

AMD 8080A/9080A MOS Microprocessor Handbook

AMD Schottky and Low Power Schottky Data Book

Am8251-Am9551 Data Sheet

Am9555A/8255A Data Sheet

Am9517 Data Sheet

Am9511 Data Sheet

Am9519 Data Sheet

Am9517 Application Note

Algorithm Details for the Am9511 Arithmetic Processor Unit

Designing Interrupt Systems with the Am9519 Universal Interrupt Controller

This manual is intended for use by system designers familiar with microcomputer architecture that utilizes the Multibus†. The information presented is sufficient to support normal installation, system interface, and programming needs; a basic theory of operation and schematic diagrams are included to assist in isolating system problems.

In this manual both active-high (positive true) and active-low (negative true) signals appear in the text. To eliminate confusion, and simplify presentation, the following convention will be adhered to within this manual. Whenever a signal is active-low (negative true), its mnemonic is followed by an asterisk * (i.e., MEMR* denotes an active-low signal). For a signal that is active-high the asterisk is omitted (i.e., IORW denotes an active-high signal).

†Multibus is a trademark of Intel Corporation

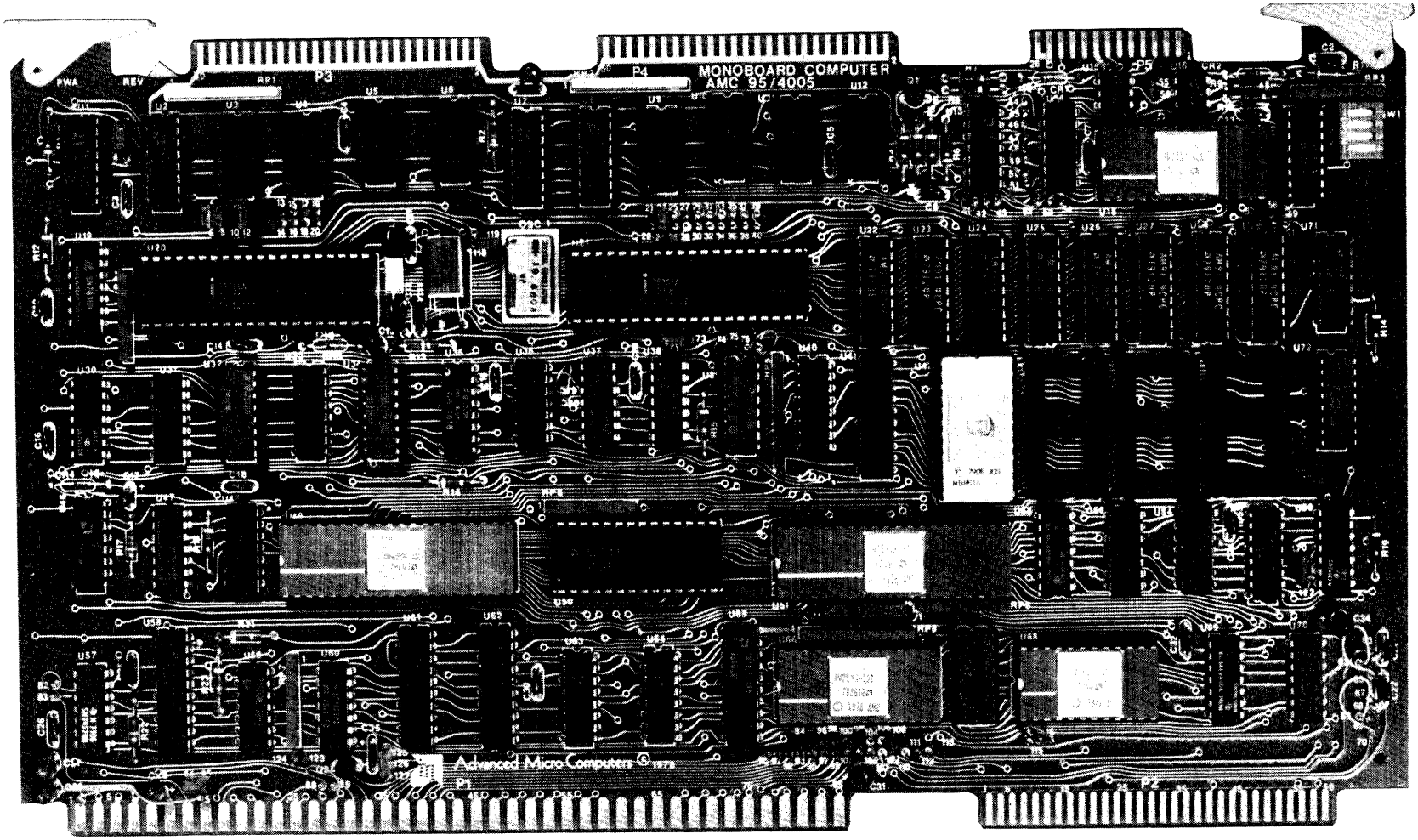
TABLE OF CONTENTS

TABLE OF CONTENTS (Cont.)

APPENDIXES

FIGURES

TABLES


MONOBOARD COMPUTER
AMC 95/4005
P3
P4
Advanced Micro Computers

CHAPTER 1
GENERAL INFORMATION

1-1. INTRODUCTION

The AMC 95/4005 MonoBoard Computer (MBC) is a complete microcomputer on a single board. It is fully compatible with Intel iSBC board products. The AMC 95/4005 offers substantially increased performance with the 3MHz option. Major functional capabilities of the AMC 95/4005 include the following:

Standard 2MHz and optional 3MHz clock rate.

Comprehensive Direct Memory Access (DMA) capability with four fully independent channels, including a built-in Memory-to-Memory transfer capability as well as software-initiated DMA requests.

Special purpose Arithmetic Processing Unit (APU) that operates concurrently with the CPU to provide both fixed-point, floating-point and transcendental computational capability.

Eight fully-programmable vectored priority interrupt channels with provisions for software-generated interrupts.

Serial priority bus master for multi-master operation.

Provision for software set override to permit MonoBoard to retain control of bus regardless of requests by other masters.

Serial Communications Interface with a 20mA current loop or RS232C capability and 13 switch selectable baud rates up to 9600 baud.

Parallel I/O Interface with 48 programmable I/O lines and sockets for drivers and receivers.

4 kilobytes of on-board read/write Random Access Memory (RAM)

Sockets for up to 16 kilobytes of on-board Read Only Memory (ROM/E-PROM)

Memory can be reconfigured by PROM programming.

Bootstrap program can be placed in on-board ROM and selected by power-on or initialization and then program disabled.

1-2. PHYSICAL DESCRIPTION

The AMC 95/4005 MonoBoard Computer (MBC) is a four layer printed circuit board with MSI TTL and LSI MOS circuits. Five edge connectors provide bus and peripheral interface capabilities. Physical characteristics of the AMC 95/4005 are:

Board Dimensions

Width	30.48 cm	(12.00 inches)
Depth	17.15 cm	(6.75 inches)
Thickness	1.50 cm	(0.60 inches)

Environmental Requirements

Operating Temperature	0°C to 55°C
Relative Humidity	Up to 90% without condensation
Storage Temperature	-40°C to +75°C

1-3. FUNCTIONAL DESCRIPTION

The AMC 95/4005 MonoBoard Computer (MBC) is a complete microcomputer on a single board. The board is fully form-factor and bus compatible with Intel iSBC 80 single board products and is designed to operate with other bus masters in a serial priority multi-master environment. However, while

maintaining mechanical and interface compatibility with the iSBC 80 series, the AMC 95/4005 offers significantly higher throughput and increased computational power over the iSBC 80 boards.

The standard board contains an Am9080A microprocessor which operates at 2MHz, a Multimode Direct Memory Access Controller, an Arithmetic Processor Unit, and a programmable eight level priority interrupt system. Additional features include: 4K bytes of RAM memory, sockets for up to 16K bytes of ROM and E-PROM memory, 48 programmable parallel I/O lines with sockets for line drivers or terminators, a programmable synchronous/asynchronous RS232C or 20mA current loop communication interface, switch selectable baud rate generator, and bus drivers for off-board memory and input/output expansion. Figure 1-1 is a block diagram of the AMC 95/4005.

A 3MHz version is available which substantially increases throughput. The 3MHz version provides all the features of the 2MHz board with an Am9080A-1 for 3MHz operation.

Direct addressing of up to 64K bytes of memory is supported by the 16-bit address bus of the Am9080. An external stack, located anywhere in RAM, can be used as a last-in/first-out stack to store the contents of the program counter, flags, accumulator, or any of the six general purpose registers. A 16-bit stack pointer controls the addressing of this external stack, which provides subroutine nesting that is bounded by memory size or 64K bytes, whichever is greater.

The Am8224 provides an oscillator, power-up controls, and two clock signals by dividing the frequency of the crystal oscillator by nine. The two clock signals (phase one and phase two) define the CPU minor cycle.

The Am8228 buffers the Data Bus and demultiplexes control signals to generate memory read/write and I/O read/write signals. The on-board memory system provides up to 4K bytes of read/write random access memory using eight Am9114 1024 by 4-bit static memory chips. Up to 16K bytes of read only memory can be installed using four Am9732 or equivalent 4K by 8-bit E-PROM chips. Alternatively, four Am9708 or 9716 equivalent memory chips, or pin compatable ROMs/PROMs, can be used if less on-board memory is required. All of the on-board memory can be disabled to permit off-board memory selection. The address to memory chip location relationship can be changed by reprogramming a single address decode PROM. This enables users to insert their existing ROM or PROM resident programs in any memory chip location. Discrete logic associated with the memory system supplies the MEMSEL* and MEMACK* signals. The MEMACK* signal indicates to the CPU that on-board memory has been selected. MEMSEL* indicates to the bus control PROM that on-board memory is selected.

An Am9517 provides four fully-independent channels of direct memory access capability. DMA transfers can be performed at up to a 1.5 megabyte per second rate. Each DMA channel may be programmed to perform I/O-to-memory or memory-to-I/O operation. Channels 0 and 1 can also be programmed for memory-to-memory block moves. Each of the DMA channels can be programmed to autoinitialize at the conclusion of the specified transfer interval. This allows repetitive DMA operations for CRT display refresh or memory buffer transfers to/from high-speed disk to be accomplished without reprogramming the DMA channel. In addition to the memory-to-memory block transfer capability on channels 0 and 1, each of the DMA channels can be programmed to operate in one of three transfer modes.

First, a block transfer mode provides for uninterrupted transfer of an entire block of data; second, a demand transfer mode provides for uninterrupted transfer as long as the demand signal is present; and third, a single transfer mode allows the I/O and CPU to

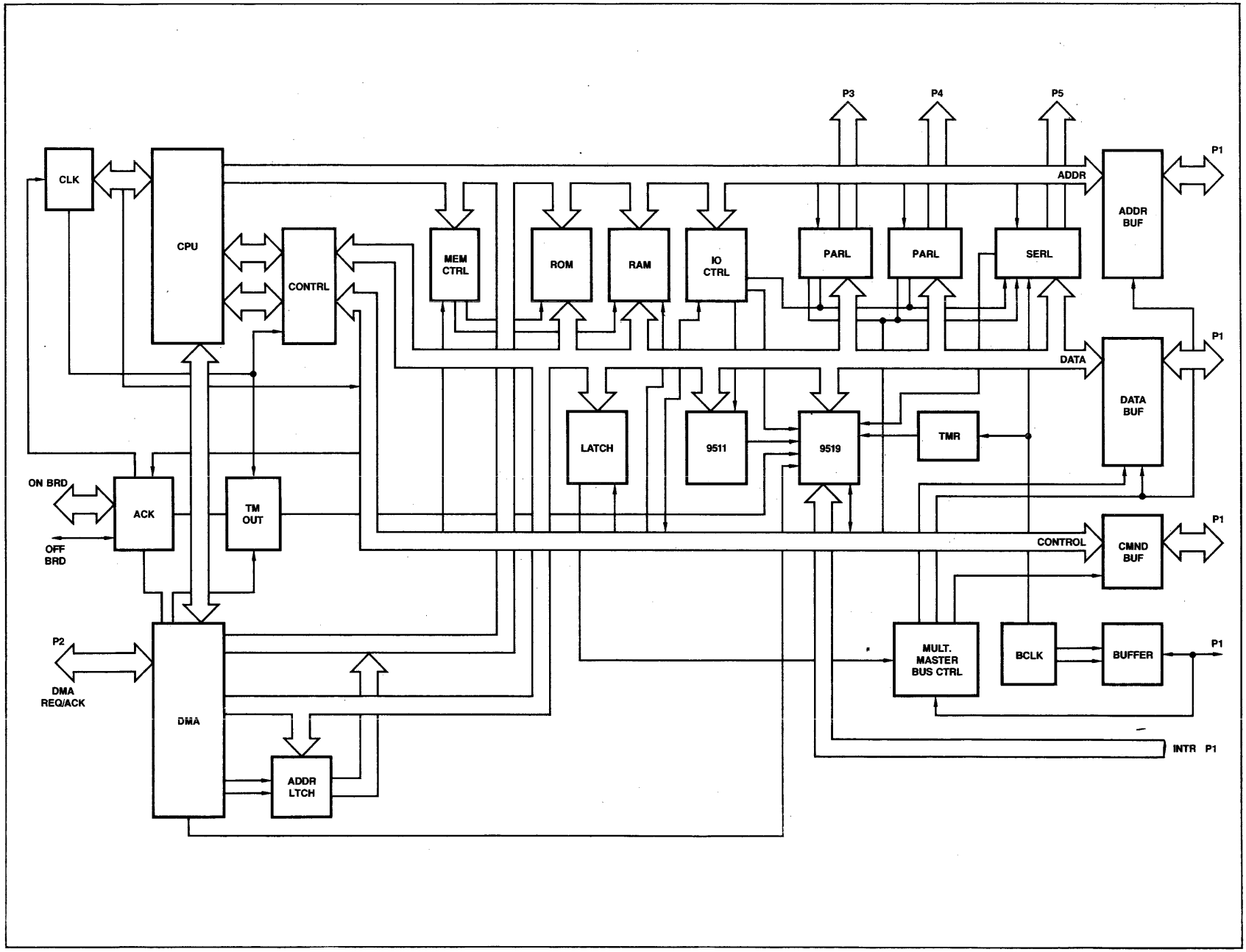


Figure 1-1. AMC 95/4005 MonoBoard Computer Block Diagram.

interleave the use of available memory cycles. DMA transfers can be initiated by hardware or software. This allows diagnostic software to be incorporated in the user's system.

An Am9519 provides the basis for an eight channel vectored priority interrupt system. Each channel is associated with its own unique four-byte location in an internal response memory. The eight channels can be programmed to perform priority resolution on either a fixed or rotating basis in either an interrupt or polled mode, allowing the user flexibility to establish interrupt service priorities based upon his unique requirements. Interrupts can also be initiated by software command, thereby permitting the hardware to resolve the priorities of software tasks and allowing maintenance and diagnostic programs to be incorporated into the user's system.

An Am9511 Arithmetic Processing Unit (APU) and its associated circuitry provide a full complement of fixed and floating point arithmetic and a variety of floating point transcendental and mathematical operations. All internal APU functions can operate concurrently with the CPU. Transfers to and from the APU are handled by the CPU. An End-of-Process signal (EOP*) is issued by the APU, and can be used as an interrupt to the CPU to help coordinate program execution. All transfers (including operand, result, status and command information) are via the data bus. Operands required for APU operations are received from the data bus and stored in an internal stack. The 8-bit bytes received during a CPU write operation are assembled into 16-bit or 32-bit operands. As each successive operand is assembled in the internal stack, each previous operand is moved down one place in the stack. The last operand entered before an Am9511 command is executed resides at the top of the stack (TOS); the next to last operand entered is next on the stack (NOS). When the Am9511 executes a command, the operands are obtained from TOS and NOS. Operation results are stored on the TOS, and the contents of the TOS are placed on the data bus during a CPU read operation. Each subsequent read operation reads NOS.

A programmable communications interface, using an Am9551 USART, provides either a 20mA current loop interface or a standard RS232C communications interface. The Am9551 provides full duplex, double buffered transmit and receive capabilities. A switch selectable baud rate generator provides the common baud rates between 50 and 9600 baud. The communications interface can be programmed to implement the desired synchronous or asynchronous serial data transmission protocol. Data format, control character format, parity and transmission rate are all under program control. Parity, overrun, and framing error detection are all incorporated on the programmable communications interface. Command and control lines, serial data lines, and signal ground lines are brought out to a 26-pin RS232C compatible connector.

The system contains 48 programmable parallel I/O lines implemented by using two Am9555 Programmable Peripheral Interface chips. System software can configure the I/O lines to sets of input, output or bidirectional input/output ports. To take full advantage of the large number of possible I/O configurations, sockets are provided for interchangeable I/O line drivers and terminators. Hence, the I/O interface is further enhanced by the capability of selecting the appropriate combination of optional line drivers and termination characteristics for each application. The programmable I/O lines and signal ground lines are brought out to two 50-pin edge connectors.

1-4. SPECIFICATIONS

Specifications for the AMC 95/4005 MonoBoard Computer are listed in table 1-1.

TABLE 1-1. SPECIFICATIONS.

Word Size	
Instruction:	8, 16 or 24 bits
Data:	8 bits
Memory Addressing	
On-Board ROM/E-PROM:	0-FFFH (using 1K devices)
On-Board RAM:	3000-3FFFH (when using 1K or 2K ROMs)
	All on-board addresses can be disabled for off-board memory expansion.
Memory Capacity	
On-Board ROM/E-PROM:	Sockets for 16K bytes (using 4K devices)
On-Board RAM:	4K bytes static (eight Am9114 chips)
Off-Board Expansion:	Up to 64K bytes
Serial I/O Address	
Control:	EDH
Data:	ECH
Parallel I/O Address	
Connector P3:	CONTROL: E7H Port A: E4H Port B: E5H Port C: E6H
Connector P4:	CONTROL: EBH Port A: E8H Port B: E9H Port C: EAH
Parallel I/O Capacity:	48 Programmable lines
Serial Communications Characteristics	
Synchronous:	5 to 8-bit characters Internal or External Character Synchronization
Asynchronous:	5 to 8-bit characters Break Character Generation 1, 1 1/2, or 2 Stop bits False start bit detector
Serial Baud Rates:	Switch Selectable (See table 2-2)
Interrupt Controller Addressing	
Control:	C3H
Data:	C2H
Latch:	EØH

TABLE 1-1. SPECIFICATIONS. (Cont.)

DMA Controller Addressing

Channel 0 Address:	B0H
Channel 0 Word Count	B1H
Channel 1 Address:	B2H
Channel 1 Word Count	B3H
Channel 2 Address:	B4H
Channel 2 Word Count:	B5H
Channel 3 Address:	B6H
Channel 3 Word Count:	B7H
Command and Status Register:	B8H
Request Register:	B9H
Mask Register (Single Bit)	BAH
Mode Register:	BBH
Clear First/Last Flip-Flop	BCH
Temporary Register:	BDH
Master Clear:	BDH
Mask Register (Four Bits):	BFH

Arithmetic Processing Unit Addressing

Control:	C1H
Data:	C0H

Power Requirements

V_{CC}	+5v ±5%
V_{DD}	+12V ±5%
V_{BB}	-5V ±5%
V_{AA}	-12V ±5%

	Without ROM Memory	
	Max	Typical
I_{CC}	3.2A	2.0A
I_{DD}	300mA	190mA
I_{BB}	1mA	1mA
I_{AA}	25mA	20mA

NOTE:

A -5 volt regulator is used to supply -5 volts to the Am9080. When 2708 E-PROMs are used, an external -5 volts must be supplied.

TABLE 1-1. SPECIFICATIONS. (Cont.)

Am9511 Command Execution Times

Command Mnemonic	Description	µSEC 95/4005/2	95/4005/3
ACOS	32-bit floating-point inverse cosine	3152-4141	2101-2761
ASIN	32-bit floating-point inverse sine	3116-3970	2077-2647
ATAN	32-bit floating-point inverse tangent	2496-3268	1664-2179
CHSD	32-bit fixed-point sign change	12-14	8.0-9.3
CHSF	32-bit floating-point sign change	8-10	5.3-6.7
CHSS	16-bit fixed-point sign change	10-12	6.7-8.0
COS	32-bit floating-point cosine	1920-2440	1280-1627
DADD	32-bit fixed-point add	10-12	6.7-8.0
DDIV	32-bit fixed-point divide	98-106	65-71
DMUL	32-bit fixed-point multiply, lower	96-106	64-71
DMUU	32-bit fixed-point multiply, upper	92-112	61-75
DSUB	32-bit fixed-point subtract	18-20	12.0-13.3
EXP	32-bit floating-point exponentiation	1898-2440	1265-1627
FADD	32-bit floating-point add	28-184	19-123
FDIV	32-bit floating-point divide	78-92	52.0-61.3
FIXD	32-bit floating-point to 32-bit fixed-point conversion	46-168	31-112
FIXS	32-bit floating-point to 16-bit fixed-point conversion	46-108	31-72
FLTD	32-bit fixed-point to 32-bit floating-point conversion	28-172	19-115
FLTS	16-bit fixed point to 32-bit floating-point conversion	32-78	21.3-52.0
FMUL	32-bit floating-point multiply	74-84	49.3-56.0
FSUB	32-bit floating-point subtraction	36-186	24-124
LOG	32-bit floating-point common logarithm	2238-3566	1492-2377
LN	32-bit floating-point natural logarithm	2140-3478	1427-2319
NOP	No operation	2	1.3
POPD	32-bit stack pop	6	4.0
POPF	32-bit stack pop	6	4.0
POPS	16-bit stack pop	5	3.3
PTOD	Push 32-bit TOS onto stack	10	6.7
PTOF	Push 32-bit TOS onto stack	10	6.7
PTOS	Push 16-bit TOS onto stack	8	5.3
PUPI	Push 32-bit floating-point π onto TOS	8	5.3
PWR	32-bit floating-point X to the Y power	4146-6016	2764-4011
SADD	16-bit fixed-point add	8-9	5.3-6.0
SDIV	16-bit fixed-point divide	42-48	28-32
SIN	32-bit floating-point sine	1898-2404	1265-1603
SMUL	16-bit fixed-point multiply, lower	42-48	28-32
SMUU	16-bit fixed-point multiply, upper	40-49	27-33
SQRT	32-bit floating-point square root	392-436	261-291
SSUB	16-bit fixed-point subtract	14-16	9.3-11
TAN	32-bit floating-point tangent	2448-2944	1632-1963
XCHD	Exchange 32-bit stack operands	13	8.6
XCHS	Exchange 16-bit stack operands	9	6.0

CHAPTER 2
INSTALLATION AND INTERFACE

2-1. INTRODUCTION

This section provides information for installing and interfaacing the AMC 95/4005 MonoBoard Computer (MBC). These instructions include unpacking and inspection, power requirements, cooling requirements, user selectable options, bus interface characteristics, and connector pin assignments.

2-2. UNPACKING AND INSPECTION

Inspect the shipping carton immediately upon receipt for evidence of mishandling during transit. If the shipping carton is severely damaged or water-stained, request the carrier's agent to be present when the carton is opened. If the carrier's agent is not present when the carton is opened and the contents of the carton are damaged, keep the carton and packing material for the agent's inspection. Shipping damages should be immediately reported to the carrier.

NOTE

Do not attempt to service the board yourself as this will void the warranty!

It is suggested that salvageable shipping cartons and packing materials be saved for use in case the product must be shipped in the future.

2-3. POWER REQUIREMENTS

The AMC 95/4005 requires +5, -5, +12V, and -12V power supply inputs. The current required from these supplies is listed in table 1-1. Ensure that the power supply has sufficient current to accomodate the AMC 95/4005 requirements.

2-4. COOLING REQUIREMENTS

The AMC 95/4005 dissipates approximately 500 gram-calories/minute (2.1 Btu/minute), and adequate air circulation must be provided to prevent a temperature rise above 55°C (130°F).

2-5. USER SELECTABLE OPTIONS

The AMC 95/4005 is designed as a general purpose microcomputer; therefore, several optional jumpers might be necessary before operation. The following paragraphs provide instructions for optional jumper configurations.

2-6. SERIAL I/O INTERFACE

The serial I/O interface is designed to interface both RS232C devices or a 20mA current loop device such as the teletype. The configuration, as shipped from the factory, is jumpered for an RS232C interface as shown in table 2-1. Jumpers must be installed between jumper pins 46 and 50 and jumper pins 47 and 48 when the Data Set Ready and Clear to Send signals are required on Connector P5. The jumper configuration for a 20mA current loop is shown in table 2-2. If an RS232C interface is used, the Am9551 can be configured to function as a data set or a data processing terminal. Connector pin assignments for connecting the P5 connector to a terminal and a modem are shown in tables 2-3 and 2-4.

2-7. BAUD RATE SELECTION

A crystal controlled baud rate generator provides an on-board baud rate clock for serial I/O interface. The baud rate is selectable via a four position dip switch. Table 2-5 shows the baud rate as a function of the switch settings.

TABLE 2-1. RS232C INTERFACE JUMPERS.

9551		Jumper	Connection
Pin	Mnemonic		
3	RxD	53-52	P5-3 (RECEIVED DATA)
19	TxD	42-41	P5-5 (TRANSMITTED DATA)
17	CTS*	49-51	Ground
9	TxC	61-59	U17-9 (CLOCK)
25	RxC	60-61	U17-9 (CLOCK)

TABLE 2-2. 20mA CURRENT LOOP INTERFACE JUMPERS.

9551		Jumper	Connection
Pin	Mnemonic		
19	TxD	42-43	P5-25 (TTY TX)
24	DTR*	46-44	P5-6 (TTY RDR CONTROL)
23	RTS*	48-49, 47-51	9551 PIN 17 (CTS)
17	CTS*	49-48	9551 PIN 23 (RTS)
9	TxC	59-61	BAUD RATE CLK
25	RxC	61-60	BAUD RATE CLK
3	RxD	53-54	P5-22 (TTY RX RET)
In the asynchronous mode, TxC and RxC can be connected to externally supplied clocks via jumpers 57-61 and 58-60.			

TABLE 2-3. CONNECTOR P5 TO TERMINAL CONNECTOR PIN ASSIGNMENTS.

Pin	Signal	DB-25 Pin No.	Pin	Signal	DB-25 Pin No.
1	CHASSIS GND	1	14	DATA TERM READY/TX CLK	20
2	Not Used	14	15	DATA CARRIER RET	8
3	TRANSMITTED DATA	2	16	TTY RDR CONTROL	21
4	Not USed	15	17	Not Used	9
5	RECEIVED DATA	3	18	Not Used	22
6	TTY RDR CONTROL	16	19	Not Used	10
7	REQUEST TO SEND	4	20	Not Used	23
8	Not Used	17	21	Not Used	11
9	CLEAR TO SEND	5	22	TTY RX RET/RX CLK	24
10	Not Used	18	23	TTY RX	12
11	DATA SET READY	6	24	TTY RX RET	25
12	Not Used	19	25	TTY TX	23
13	SIGNAL GND	7	26	SIGNAL GND	--

TABLE 2-4. CONNECTOR P5 TO MODEM CONNECTOR PIN ASSIGNMENTS.

Pin	Signal	RS232C Pin No.
1	CHASSIS GND	1
2	Not Used	--
3	TRANSMITTED DATA	3
4	Not Used	--
5	RECEIVED DATA	2
6	TTY RDR CONTROL	--
7	REQUEST TO SEND	5
8	Not Used	--
9	CLEAR TO SEND	4
10	Not Used	--
11	DATA SET READY	20
12	Not Used	--
13	SIGNAL GND	7
14	DATA TERM READY/ TX CLK	6/15
15	DATA CARRIER RET	--
16	TTY RDR CONTROL	--
17	Not Used	--
18	Not Used	--
19	Not Used	--
20	Not Used	--
21	Not Used	--
22	TTY RX RET/RX CLK	17
23	TTY RX	--
24	TTY TX RET	--
25	TTY TX	--
26	SIGNAL GND	--

2-8. PRIORITY INTERRUPT JUMPERS

The priority interrupt jumper matrix provides for eight out of sixteen possible interrupts to be jumpered to the eight interrupt controller inputs. When the .833msec real time clock interrupt is selected, the system can be interrupted at a fixed rate to service some real time event. Table 2-6 shows the possible jumper configurations for the interrupt controler inputs.

2-9. PARALLEL I/O JUMPER OPTION

The parallel I/O section is configured for Am8216/8226 bidirectional bus drivers at ports E4H and E8H. As delivered from the factory, jumpers are installed between jumper pins 2 and 3, and between jumper pins 23 and 24; this ties the Am8216/8226 DIEN* inputs to pin 6 of ports E6H and EAH, thereby configuring both ports E4H and E8H as bidirectional ports. Either or both ports can be configured as input or output ports with the following changes.

TABLE 2-5. BAUD RATE SELECTION.

BAUD	Switch Position			
	4	3	2	1
50	0	0	1	0
75	0	0	1	1
110	1	1	1	1
134.5	0	1	0	0
150	1	1	1	0
200	0	1	0	1
300	1	1	0	1
600	0	1	1	0
1200	1	0	1	1
1800	1	0	1	0
2400	0	1	1	1
4800	1	0	0	1
9600	1	0	0	0

1 = open position

CONFIGURATION	PORT	REMOVE	INSTALL
Input	E4H	2 and 3	1 and 2
Input	E8H	23 and 24	21 and 23
Output	E4H	2 and 3	2 and 4
Output	E8H	23 and 24	22 and 23

All lines for ports E6H and EAH are jumper connected to their line driver/ terminator sockets. This allows complete flexibility for signal interchange when operating in mode 2.

TABLE 2-6. PRIORITY INTERRUPT JUMPERS.

Signal	Pin No.	Column 1 Jumper Pins	Column 2 Jumper Pins	INT Input
XTT* (RTC)	On-Board	107		
ERROR* (9512)	On-Board	113		
EOP* (9517)	On-Board	105		
TIME OUT	On-Board	103		
INT 11*	On-Board	109		
INT 51A* (TxRDY)	On-Board	112	108	IRQ0*
INT 51B* (RxRDY)	On-Board	111	106	IRQ1*
SPARE	On-Board	110	104	IRQ3*
IRQ7*	P1-36	90	102	IRQ4*
IRQ6*	P1-35	91	100	IRQ4*
IRQ5*	P1-38	92	98	IRQ5*
IRQ4*	P1-37	93	96	IRQ6*
IRQ3*	P1-40	95	94	IRQ7*
IRQ2*	P1-39	97		
IRQ1*	P1-42	99		
IRQ0*	P1-41	101		
Jumper any column 1 pin to any column 2 pin.				

2-10. MEMORY SELECTION

To customize the AMC 95/4005 board for the type of memory devices being used, jumper connections must be made as shown on table 2-7. The memory address to memory device relationship is controlled by the address decoder PROM at location U39. As delivered, the PROM is programmed as shown in figure D-2. Information on how to program the PROM for unique system requirements is presented in chapter 3. A jumper must be connected between jumper pins 79 and 80 to enable on-board memory.

TABLE 2-7. MEMORY JUMPER CONNECTIONS.

Function	Memory Device			
	Am9708	Am9716	Am9732	2758
Address	76-77	76-77	---	---
Address	74-75	---	---	---
+12V to pin 19	66-68	---	---	---
GND to pin 19	---	---	---	66-69
A10	---	66-70	66-70	---
A11	---	---	62-63	---
-5V to pin 21	62-64	---	---	---
+5V to pin 21	---	62-65	---	---

2-11. Am9080A READY TIMING OPTION

The MonoBoard Computer has a set of jumper pins associated with the 9080A Ready input. When operating with the 3MHz option, one wait state is required for a ROM. A jumper is installed between jumper pins 71 and 72 to delay the MEMACK* signal and ensure the wait state. When operating with a 2MHz clock, the jumper is removed between jumper pins 71 and 72 and a jumper is installed between jumper pins 72 and 73.

2-12. APU CLOCK FOR 95/4005/3

When operating a 3MHz MonoBoard, the Am9511 APU operates on a 3MHz clock. Jumper pins 114, 115 and 116 allow the user to use the 3MHz phase two TTL clock or the APU internal clock. For APU operation with the phase two TTL clock, install a jumper between jumper pins 114 and 115. For the APU to operate on its internal clock, remove the jumper between jumper pins 114 and

115 and install a jumper between pins 115 and 116.

2-13. MULTI-MASTER CONTROL

The board, as shipped, is configured for serial bus priority resolution; a jumper is connected between jumper pins 84 and 85. Remove this jumper to configure the board for parallel priority resolution. Jumper pins 120, 121, and 122 select the conditions that cause the board to give up control of the bus. When a jumper is connected between jumper pins 120 and 121, the MonoBoard will retain control until another external bus request is received. When a jumper is connected between jumper pins 121 and 122, the MonoBoard gives up control after each bus transfer. As shipped, a jumper is connected between pins 121 and 122.

2-14. INTERFACE REQUIREMENTS

The following paragraphs identify the board external connections and bus signal characteristics and timing.

2-15. SERIAL I/O INTERFACE

The serial I/O interface communicates with an external devie via 26-pin PC edge connector P5. An external device can be connected to P5 using a 3M 3462-0001 flat cable connector or a TI H312113 or AMP 1-583715-1 solder connector. When connected to a DB-25 connector, the connector pins are numbered differently. Table 2-4 is a pin list for connector P5 and includes a cross reference to standard RS232C pin numbering.

2-16. PARALLEL I/O INTERFACE

The parallel I/O interface communicates with external I/O devices via two 50-pin double sided edge connectors P3 and P4. External devices can be attached to P3 or P4 using one of the mating connectors listed in table 2-8. Tables 2-9 and 2-10 provide a pin list for

TABLE 2-8. PARALLEL I/O MATING CONNECTORS.

Connector Type	Vendor	Part No.
Flat Cable	3M	3415-0001
	AMP	2-86792-3
Soldered	AMP	2-583715-3
	VIKING	3VH25/1JV-5
	TI	H312125
Wire-wrap	TI	H311125
	VIKING	3VH25/1JND-5
	CDC	VPB01B25D00A1
	ITT	EC4A050A1A
Crimp	AMP	1-583717-1

TABLE 2-9. PARALLEL I/O CONNECTOR P3 PIN ASSIGNMENTS.

Pin		Signal	Pin	Signal
1		Bit 7	2	GND
3		Bit 6	4	
5		Bit 5	6	
7		Bit 4	8	
9	Port E5	Bit 3	10	
11		Bit 2	12	
13		Bit 1	14	
15		Bit 0	16	
17		Bit 3	18	
19		Bit 2	20	
21		Bit 1	22	
23	Port E6	Bit 0	24	
25		Bit 4	26	
27		Bit 5	28	
29		Bit 6	30	
31		Bit 7	32	
33		Bit 7	34	
35		Bit 6	36	
37		Bit 5	38	
39	Port E4	Bit 4	40	
41		Bit 3	42	
43		Bit 2	44	
45		Bit 1	46	
47		Bit 0	48	
49			50	GND

connectors P3 and P4. TTL line drivers compatible with the I/O driver sockets in the parallel I/O interface are listed in table 2-11. Parallel I/O interface lines can be terminated by either a 220Ω/330Ω divider or a 1KΩ pull-up as shown in figure 2-1. The 220Ω/330Ω divider is stocked by distributors under Intel part number iSBC-901 and National Semiconductor part number BLC-901. The 1KΩ pull-up is stocked under Intel part number iSBC-902 and National Semiconductor part number BLC-902.

2-17. AUXILIARY CONNECTOR P2

Connector P2 is a 60-pin double sided edge connector that provides interface to the DMA Controller and various other signals. Table 2-12 is a pin list for connector P2.

TABLE 2-10. PARALLEL I/O CONNECTOR P4 PIN ASSIGNMENTS.

Pin		Signal	Pin	Signal
1		Bit 7	2	GND
3		Bit 6	4	
5		Bit 5	6	
7	Port E9	Bit 4	8	
9		Bit 3	10	
11		Bit 2	12	
13		Bit 1	14	
15		Bit 0	16	
17		Bit 3	18	
19		Bit 2	20	
21		Bit 1	22	
23	Port EA	Bit 0	24	
25		Bit 4	26	
27		Bit 5	28	
29		Bit 6	30	
31		Bit 7	32	
33		Bit 7	34	
35		Bit 6	36	
37		Bit 5	38	
39	Port E8	Bit 4	40	
41		Bit 3	42	
43		Bit 2	44	
45		Bit 1	46	
47		Bit 0	48	
49		--	50	GND

2-18. BUS INTERFACE

This section describes the signals that interface the MonoBoard Computer to the external system bus. signals shown with an asterisk (*) following the signal

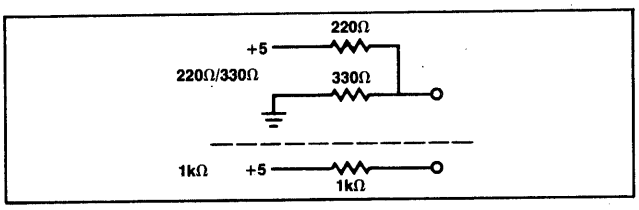


Figure 2-1. Parallel I/O Line Terminator.

TABLE 2-11. PARALLEL I/O SOCKET COMPATIBLE LINE DRIVERS.

Driver	Characteristic	Sink Current
7438	I, OC	48mA
7437	I	48mA
7432	NI	16mA
7426	I, OC	16mA
7409	NI, OC	16mA
7408	NI	16mA
7403	I, OC	16mA
7400	I	16mA

Note: I = inverting;
NI = non-inverting
OC = open collector

TABLE 2-12. CONNECTOR P2 PIN ASSIGNMENTS.

Pin	Mnemonic	Function
14	SVACK*	9511 SVACK
39	SVREQ*	9511 SVREQ
40	DRQ0	DMA Request 0
42	DACK0*	DMA Acknowledge 0
44	DRQ1	DMA Request 1
46	DACK1*	DMA Acknowledge 1
48	DRQ2	DMA Request 2
50	DACK2*	DMA Acknowledge 2
52	DRQ3	DMA Request 3
54	DACK3*	DMA Acknowledge 3
56	EOP*	External End-Of-Process
55	IOACK*	I/O Acknowledge
57	WAIT*	External ready input
58	Ø2 TTL	Ø2 TTL OUT

name are active-low signals. Connector P1 is an 86-pin double sided edge connector that provides the bus interface for the AMC 95/4005. Table 2-13 is a pin list for connector P1. DC characteristics for the bus interface are given in table 2-14. AC characteristics are given in tables 2-15 through 2-18 with timing shown in figures 2-2 through 2-4. When the MonoBoard is being used with another bus master, the BPRN* input (P1-15) to the master assigned the highest priority must be tied low. The BPRN* input to each master with the next lower priority must be connected to the BPRO* output (P1-16) of the next higher priority master.

TABLE 2-13. SYSTEM BUS CONNECTOR P1 PIN ASSIGNMENTS.

	(Component Side)			(Circuit Side)		
	Pin	Mnemonic	Description	Pin	Mnemonic	Description
Power Supplies	1	GND	Signal GND	2	GND	Signal GND
	3	+5	+5 VDC	4	+5	+5 VDC
	5	+5	+5 VDC	6	+5	+5 VDC
	7	+12	+12 VDC	8	+12	+12 VDC
	9	-5	-5 VDC	10	-5	-5 VDC
	11	GND	Signal GND	12	GND	Signal GND
Bus Controls	13	BCLK*	Bus Clock	14	INIT*	Initialize
	15	BPRN*	Bus Priority In	16	BPRO*	Bus Priority Out
	17	BUSY*	Bus Busy	18	BREQ*	Bus Request
	19	MRDC*	Mem. Read Command	20	MWTC*	Mem. Write Command
	21	IORC*	I/O Read Command	22	IOWC*	I/O Write Command
	23	XACK*	XFER Acknowledge	24	INH1*	Inhibit 1 (RAM)
	25	AACK*	Advance Acknowledge	26	INH2*	Inhibit 2 (ROM)
	27	BHEN*	Not Used	28	ADR10*	Not Used
	29	CBRQ*	Common Bus Request	30	ADR11*	Not Used
	31	CCLK*	Constant Clock	32	ADR12*	Not Used
	33	INTA*	Interrupt Acknowledge	34	ADR13*	Not Used
Interrupts	35	IRQ6*	Interrupt Requests	36	IRQ7*	Interrupt Requests
	37	IRQ4*		38	IRQ5*	
	39	IRQ2*		40	IRQ3*	
	41	IRQ0*		42	IRQ1*	
Addresses	43	ADRE*	Address Bus	44	ADRF*	Address Bus
	45	ADRC*		46	ADRD*	
	47	ADRA*		48	ADRB*	
	49	ADR8*		50	ADR9*	
	51	ADR6*		52	ADR7*	
	53	ADR4*		54	ADR5*	
	55	ADR2*		56	ADR3*	
	57	ADR0*		58	ADR1*	
Data	59	DATE*	Not Used	60	DATF*	Not Used
	61	DATC*	Not Used	62	DATD*	Not Used
	63	DATA*	Not Used	64	DATB*	Not Used
	65	DAT8*	Not Used	66	DAT9*	Not Used
	67	DAT6*		68	DAT7*	
	69	DAT4*		70	DAT5*	
	71	DAT2*	Data Bus	72	DAT3*	Data Bus
	73	DAT0*		74	DAT1*	
Power Supplies	75	GND	Signal GND	76	GND	Signal GND
	77		Reserved	78		Reserved
	79	-12	-12 VDC	80	-12	-12 VDC
	81	+5	+5 VDC	82	+5	+5 VDC
	83	+5	+5 VDC	84	+5	+5 VDC
	85	GND	Signal GND	86	GND	Signal GND

TABLE 2-14. AMC 95/4005 BUS DC CHARACTERISTICS.

Signals	Symbol	Parameter Description	Test Conditions	Min	Max	Units
ADRQ*-ADRF*	V_{OL}	Output Low Voltage	I_{OL} = 50mA		0.6	V
ADDRESS	V_{OH}	Output High Voltage	I_{OH} = -10mA	2.4		V
	V_{IL}	Input Low Voltage			0.95	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.45		-0.25	mA
	I_{IH}	Input Current at High V	V_{IN} = 5.25V		10	µA
	C_L	Capacitive Load			18	pF
MRDCI*,	V_{OL}	Output Low Voltage	I_{OL} = 32mA		0.4	V
MWTC*,	V_{OH}	Output High Voltage	I_{OH} = -5.2mA	2.4		V
IORC*,	I_{LH}	Output Leakage High	V_O = 2.4		40	µA
IOWC*	I_{LL}	Output Leakage Low	V_O = 0.4		-40	µA
	C_L	Capacitive Load			15	pF
DATA0-	V_{OL}	Output Low Voltage	I_{OL} = 50mA		0.6	V
DATA7	V_{OH}	Output High Voltage	I_{OH} = -10mA	2.4		V
	V_{IL}	Input Low Voltage			0.95	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.45		-0.25	mA
	I_{LH}	Output Leakage High	V_O = 5.25		100	µA
	I_{LL}	Output Leakage Low	V_O = 0.45		100	µA
	C_L	Capacitive Load			18	pF
INTI*	V_{IL}	Input Low Voltage			0.8	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.4V		-2.2	mA
	I_{IH}	Input Current at High V	V_{IN} = 5.5V		1	mA
	C_L	Capacitive Load			18	pF
BPRN*,	V_{IL}	Input Low Voltage			0.8	V
XACK*,	V_{IH}	Input High Voltage		2.0		V
AACK*	I_{IL}	Input Current at Low V	V_{IN} = 0.5		-2.6	mA
	I_{IH}	Input Current at High V	V_{IN} = 2.7V		0.30	mA
	C_L	Capacitive Load			18	pF
BUSY*	V_{OL}	Output Low Voltage	I_{OL} = 25 mA		0.4	V

TABLE 2-14. AMC 95/4005 BUS DC CHARACTERISTICS. (Cont.)

Signals	Symbol	Parameter Description	Test Conditions	Min	Max	Units
OPEN COLLECTOR	C_L	Capacitive Load			20	pF
INIT* (SYSTEM RESET)	V_{OL}	Output Low Voltage	I_{OL} = 32 mA		0.6	V
	V_{OH}	Output High Voltage	OPEN COLLECTOR			
	V_{IL}	Input Low Voltage			0.7	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 5.5		0.1	mA
	I_{IH}	Input Current at High V	V_{IN} = 0.3		-0.7	mA
	C_L	Capacitive Load			38	pF
BCLK*	V_{OL}	Output Low Voltage	I_{OL} = 48 mA		0.5	V
	V_{OH}	Output High Voltage	I_{OH} = -1 mA	2.7		V
	C_L	Capacitive Load			18	pF
EXT INTR*	V_{IL}	Input Low Voltage			0.8	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.4V	6.8		mA
	I_{IH}	Input Current at High V	V_{IN} = 5.5V		2	mA
	C_L	Capacitive Load			18	pF
PORT E4 and E8 BIDIREC-TIONAL DRIVERS	V_{OL}	Output Low Voltage	I_{OL} = 20 mA		.45	V
	V_{OH}	Output High Voltage	I_{OH} = -12.mA	2.4		V
	V_{IL}	Input Low Voltage			.95	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.45		5.25	mA
	I_{LH}	Output Leakage High	V_O = 5.25		.30	mA
	I_{LL}	Output Leakage Low	V_O = 0.45		5.25	mA
	C_L	Capacitive Load			18	pF
Am9555 DRIVER/ RECEIVER	V_{OL}	Output Low Voltage	I_{OL} = 1.7 mA		.45	V
	V_{OH}	Output HIgh Voltage	I_{OH} = -50 µA	2.4		V
	V_{IL}	Input Low Voltage			.8	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.45		10	µA
	I_{IH}	Input Current at High V	V_{IN} = 5.0		10	µA
	C_L	Capacitive Load			18	pF

TABLE 2-15. AMC 95/4005 BUS EXCHANGE AC CHARACTERISTICS (2MHz CLOCK).

	Overall		Read		Memory Write		
Parameter	Min (ns)	Max (ns)	Min (ns)	Max (ns)	Min (ns)	Max (ns)	Description
t_{AS}	52		52		52		Address Setup Time to Command
t_{AH}					79		Address Hold Time
t_{DS}	52				52		Data Setup Time to Command
t_{DH}	79				79		Data Hold Time
t_{RDY}	65		65				First Ready Sampling Point of Current Cycle
t_{RDY2}	565		565				Second Ready Sampling Point of Current Cycle
t_{RDY3}	1065		1065				Third Ready Sampling Point of Current Cycle
t_{CY}	500		500		500		Cycle Time
t_{WC}			715		935		Command Width
t_{ACC}			550				Read Access Time
t_{8KD}			65				9080A ACK Response Time for Minimum Delay
t_{8KO}	100		100		100		9080A ACK Turn Off Delay
t_{XKD}			50				XACK Delay From Valid Data or Write
t_{XKO}	100		100		100		XACK Turn Off Delay
t_{DBS}		3500					Bus Sample to Exchange Initiation
t_{BS}	102	110					Bus Sampling Point Delay
t_{DBY}	25	125					Bus Busy Turn On Delay

Memory and I/O access occurs with no wait states.
Assume HOLD* becomes active prior to DAT instruction.

TABLE 2-16. AMC 95/4005 CONTINUOUS BUS CONTROL AC CHARACTERISTICS (2MHz CLOCK).

	Overall		Read		Memory Write		
Parameter	Min (ns)	Max (ns)	Min (ns)	Max (ns)	Min (ns)	Max (ns)	Description
t_{AS}	52		52		52		Address Setup Time to Command
t_{AH}					79		Address Hold Time
t_{DS}	495				495		Data Setup Time to Command
t_{DH}	79				79		Data Hold Time
t_{RDY}	65		65				First Ready Sampling Point of Current Cycle
t_{RDY2}	565		565				Second Ready Sampling Point of Current Cycle
t_{RDY3}	1065		1065				Third Ready Sampling Point of Current Cycle
t_{CY}	500		500				Cycle Time
t_{WC}			915		1035		Command Width
t_{ACC}			385				Read Access Time
t_{8KD}			275				9080A ACK Response Time for Minimum Delay
t_{8KO}	100		100		100		9080A ACK Turn Off Delay
t_{XKD}			50				XACK Delay From Valid Data or Write
t_{XKO}	100		100		100		XACK Turn Off Delay
t_{BCY}	100						Bus Clock Cycly Time
t_{BW}	50						Bus Clock Low or High Periods
t_{INT}	4500						Initialization Width

MAX assumes no acknowledge delays.
Write Command to next Read Command separation.
After all voltages have stablized.

TABLE 2-17. AMC 95/4005 BUS EXCHANGE AC CHARACTERISTICS (3MHz CLOCK).

	Overall		Read		Memory Write		
Parameter	Min (ns)	Max (ns)	Min (ns)	Max (ns)	Min (ns)	Max (ns)	Description
t_{AS}	52		52		52		Address Setup Time to Command
t_{AH}					53		Address Hold Time
t_{DS}	52				52		Data Setup Time to Command
t_{DH}	53				53		Data Hold Time
t_{RDY}	43		43				First Ready Sampling Point of Current Cycle
t_{RDY2}	380		380				Second Ready Sampling Point of Current Cycle
t_{RDY3}	703		703				Third Ready Sampling Point of Current Cycle
t_{CY}	333		333		333		Cycle Time
t_{WC}			476		623		Command Width
t_{ACC}			366				Read Access Time
t_{8KD}			43				9080A ACK Response Time for Minimum Delay
t_{8KO}	100		100		100		9080A ACK Turn Off Delay
t_{XKD}			50				XACK Delay From Valid Data or Write
t_{XKO}	100		100		100		XACK Turn Off Delay
t_{DBS}		3500					Bus Sample to Exchange Initiation
t_{BS}	102	110					Bus Sampling Point Delay
t_{DBY}	25	125					Bus Busy Turn On Delay

Memory and I/O access occurs with no wait states.
Assume HOLD* becomes active prior to DAT instruction.

TABLE 2-18. AMC 95/4005 CONTINUOUS BUS CONTROL AC CHARACTERISTICS (3MHz CLOCK).

	Overall		Read		Memory Write		
Parameter	Min (ns)	Max (ns)	Min (ns)	Max (ns)	Min (ns)	Max (ns)	Description
t_{AS}	52		52		52		Address Setup Time to Command
t_{AH}					53		Address Hold Time
t_{DS}	52				52		Data Setup Time to Command
t_{DH}	53				53		Data Hold Time
t_{RDY}	43		43				First Ready Sampling Point of Current Cycle
t_{RDY2}	380		380				Second Ready Sampling Point of Current Cycle
t_{RDY3}	703		703				Third Ready Sampling Point of Current Cycle
t_{CY}	333		333		333		Cycle Time
t_{WC}			676		823		Command Width
t_{ACC}			256				Read Access Time
t_{8KD}			183				9080A ACK Response Time for Minimum Delay
t_{8KO}	100		100		100		9080A ACK Turn Off Delay
t_{XKD}			50				XACK Delay From Valid Data or Write
t_{XKO}	100		100		100		XACK Turn Off Delay
t_{BCY}	100						Bus Clock Cycly Time
t_{BW}	50						Bus Clock Low or High Periods
t_{INT}	4500						Initialization Width

MAX assumes no acknowledge delays.
Write Command to next Read Command separation.
After all voltages have stablized.

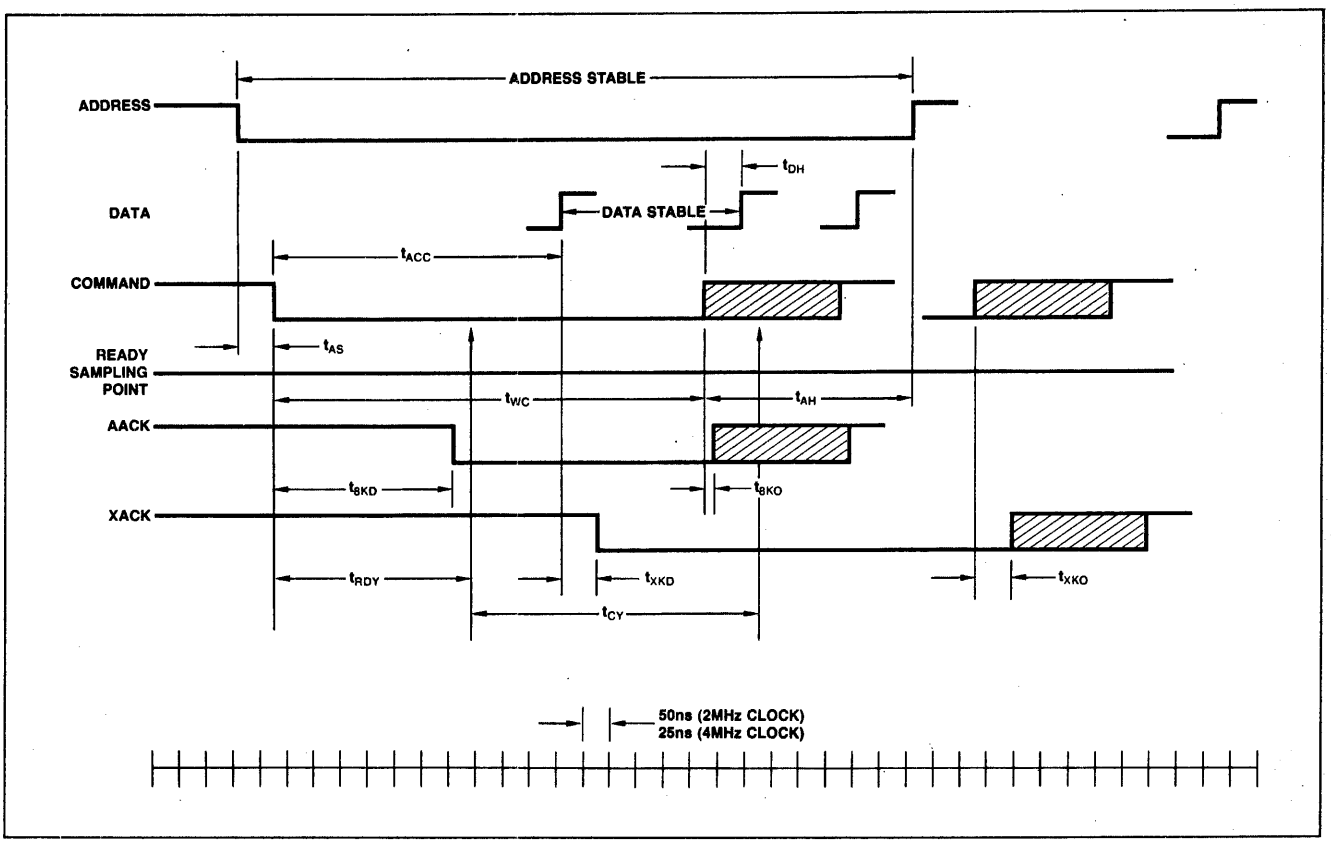
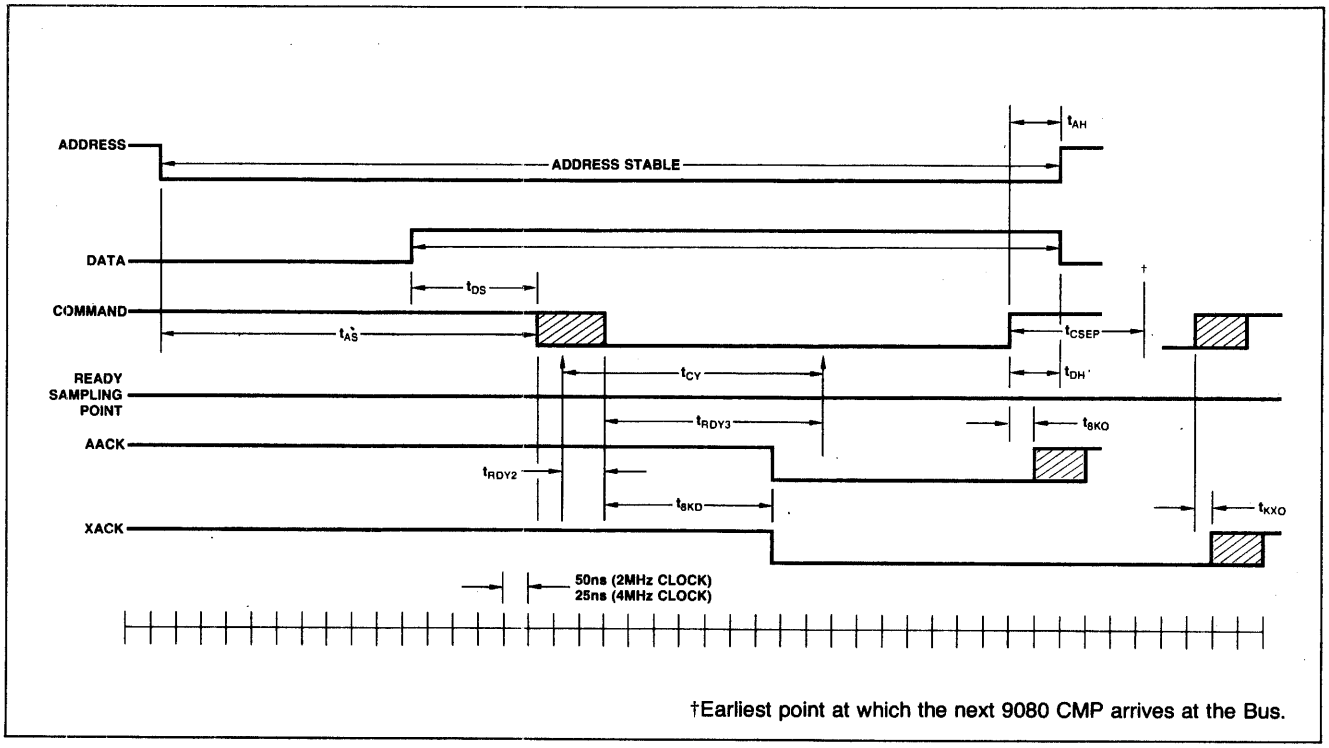


Figure 2-2. Memory and I/O Read Timing (Continuous Bus Control).



†Earliest point at which the next 9080 CMP arrives at the Bus.

Figure 2-3. Memory and I/O Write Timing (Continuous Bus Control).

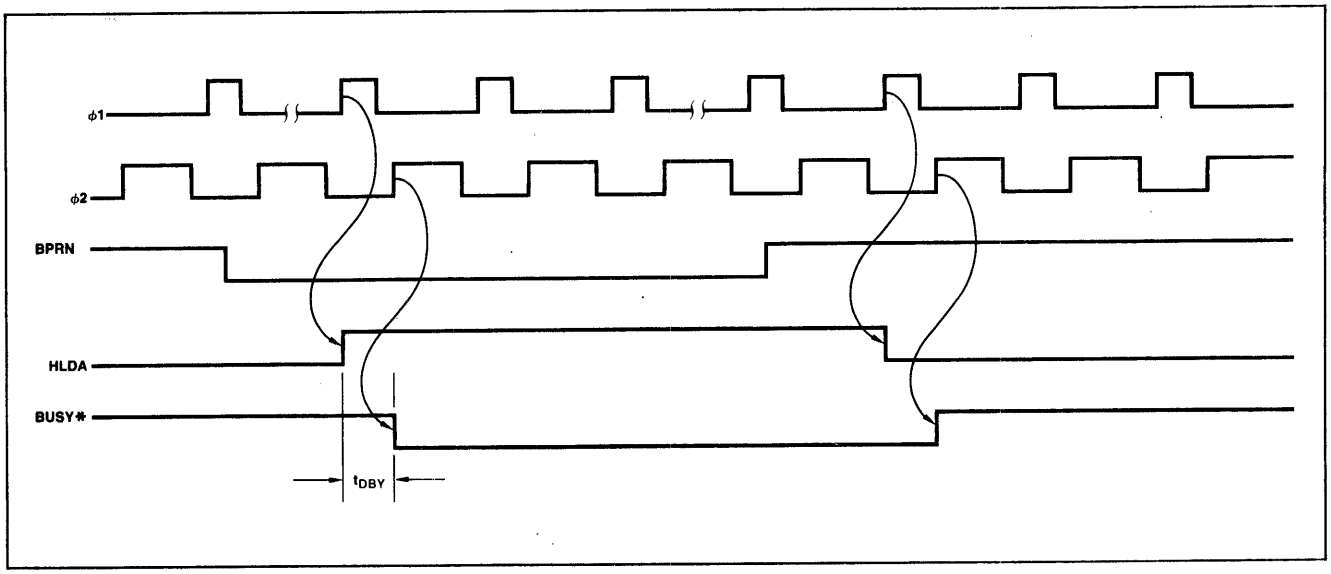


Figure 2-4. Bus Exchange Timing.

CHAPTER 3
OPERATION AND PROGRAMMING

3-1. INTRODUCTION

This section provides operating and programming information for the AMC 95/4005 MonoBoard Computer (MBC) and the programmable devices. The MBC includes five programmable devices as follows:

a An Am9551 Programmable Communications Interface chip that provides serial I/O;

b. Two Am9555 Programmable Peripheral Interface chips that control the 48 parallel I/O lines;

c. An Am9519 Universal Interrupt Controller that responds to eight interrupt lines;

d. An Am9517 Multimode DMA Controller that controls various memory and I/O data transfer operations;

e. An Am9511 Arithmetic Processing Unit that provides extended fixed and floating point arithmetic processing capabilities.

3-2. ADDRESS ASSIGNMENT

The CPU communicates with the programmable devices through a sequence of I/O read and I/O write commands. A summary of the I/O addresses as shipped from the factory is provided in table 3-1. Depending upon the application for which the board is to be used, users might have to alter the factory selected I/O addresses by programming their own I/O chip select PROM (U32). Bus address bits 9 through 15 correspond to the A1 through A7 PROM inputs respectively; PROM input A0 is driven by the 9080A Hold Acknowledge output. PROM output pins O0, O1, and O2 are connected to input pins A, B, and C respectively of the three-to-eight line decoder which generates the signals used to select the I/O chip. Therefore, the relationship of the PROM output to the I/O devices controlled can not be changed by programming; only the address at which the output pattern for selecting an I/O device is stored can be selected by PROM programming.

3-3. BUS OVERRIDE AND BOOT CONTROL

A programmable latch provides dynamic override control of bus requests and allows memory locations to be reassigned under program control. As delivered, the latch is accessed by address EØH. (This address can be changed by changing the I/O address PROM U32.) When the latch is addressed, data bit 0 controls the memory allocation and data bit 7 controls the bus request override. The latch is cleared during power up and by initialization and is input as address bit 6 to the Address Select PROM. Therefore, the low output from the bit 0 latch is normally used to select an on-board bootstrap starting in the low addresses of ROM/E-PROM. Once the bootstrap has executed, the bit 0 latch can be set high to select another group of Address Select PROM locations. Another use of the low bit 0 latch output might be to disable all on-board memory so that the bootstrap can come from off-board. The Address Select PROM determines how the bit 0 latch will be used.

The bit 7 latch is set to override a bus request made by another master; a high output prevents another master from gaining control of the bus. This latch would normally be set when it is important that the MonoBoard not loose control of the bus. When the critical condition has passed, the bit 7 latch

TABLE 3-1. I/O PORT ADDRESSES.

I/O Port Address	I/O Device	Input Function	Output Function
B0H	Am9517 DMA Controller	Channel 0 Address	Channel 0 Address
B1H		Channel 0 Word Count	Channel 0 Word Count
B2H		Channel 1 Address	Channel 1 Address
B3H		Channel 1 Word Count	Channel 1 Word Count
B4H		Channel 2 Address	Channel 2 Address
B5H		Channel 2 Word Count	Channel 2 Word Count
B6H		Channel 3 Address	Channel 3 Address
B7H		Channel 3 Word Count	Channel 3 Word Count
B8H		Status Register	Command Register
B9H		Not Used	Request Register
BAH		Not Used	Mask Register Single Bit
BBH		Not Used	Mode Register
BCH		Not Used	Clear First/Last Flip-Flop
BDH		Temporary Register	Master Clear
BEH		Not Used	Not Used
BFH		Not Used	Mask Register Four Bits
C0H	Am9511 APU	Data Byte From Stack	Data Byte Onto Stack
C1H		Read Status	Enter Command
C2H	Am9519 Interrupt Controller	Data Register	Data Register
C3H		Status Register	Command Register
EØH	74LS273 Latch	Not Used	Bus Override and Boot Control
E4H	Am9555 Parallel I/O Ports E4H-E6H	Read Port A	Write Port A
E5H		Read Port B	Write Port B
E6H		Read Port C	Write Port C
E7H		Not Used	Control Register
E8H	Am9555 Parallel I/O Ports E8H-EAH	Read Port A	Write Port A
E9H		Read Port B	Write Port B
EAH		Read Port C	Write Port C
EBH			Control Register
ECH	Am9551 Serial I/O	Receive Data Buffer	Transmit Data Buffer
EDH		Status Register	Command Register

can be cleared to permit another bus request to be recognized.

3-4. MEMORY SELECT PROM PROGRAMMING

A memory mapping PROM located at U39 is used to control the address to memory location relationship. As delivered,

the PROM is programmed as shown in the figure of Appendix D. When configured for use with Am9708s, the PROM maps addresses 0 through 0FFFH to ROM and addresses 3000H through 3FFFH to RAM; when configured for use with Am9716s, the PROM maps addresses 0 through 1FFFH to ROM and addresses 3000H through 3FFFH to RAM; and when configured for use with Am9732s, the PROM maps addresses 0 through 3FFFH to ROM and addresses 4000H through 4FFFH to RAM. Memory references are directed to ROM when input bit 5 (boot) is low. When bit 5 is high, on-board memory is disabled. To use a memory address configuration other than one supplied in the standard memory address PROM, it is necessary to replace the PROM with one customized to the specific application. A 27S21 PROM can be used as the memory mapping PROM at the expense of not being able to map memory into the upper 32K addresses.

The memory select circuitry is designed so that a low output on bit 3 of the PROM selects RAM, and a low output on bit 2 selects ROM. Address bits 10 and 11 are decoded by the 74LS139 chip at U40 to select one of four pairs of RAMs as shown in table 3-2. ROM selection is determined by PROM output bits 0 and 1 as shown in table 3-3. Note that if neither PROM output bits 0 nor 1 are low, on-board memory is not selected and the memory reference will be to off-board memory.

TABLE 3-2. RAM SELECTION.

Address Bits 10	11	Memory Pair Selected
0	0	U22, U26
0	1	U23, U27
1	0	U24, U28
1	1	U25, U29

TABLE 3-3. ROM SELECTION.

PROM Bits 0	1	ROM/E-PROM Selected
0	0	U42
0	1	U43
1	0	U44
1	1	U45

After determining what memory configurations are desired, it is necessary to determine what addresses are to be used to reference the memory locations. Address bits 10 through 14 are connected to PROM inputs 0 through 4 respectively; address bit 15 is connected to PROM input 8. PROM inputs 6 and 7 can each be tied high or tied low, as shown in table 2-7, to meet the requirements of the ROMs being used. Connect a jumper between jumper pins 74 and 75 to tie input 6 low and connect a jumper between jumper pins 76 and 77 to tie input 7 low; the inputs are high when the jumpers are removed. Input 5 to the PROM is the boot control bit already described. Based on the memory configuration selected and the input address and options desired, it is up to the user to program the PROM so that the desired PROM addres contains the data pattern required for memory chip selection.

3-5. SERIAL I/O INTERFACE PROGRAMMING

An Am9551 Programmable Communications Interface presents a parallel, 8-bit interface to the CPU via the data bus and presents an RS232C or a 20mA current loop interface to an external device via connector P5. Programmable operating modes and format options allow the Am9551 to service a wide range of communications, disciplines and applications. Operating modes are determined by a mode instruction word and a command instruction word.

3-6. Am9551 INITIALIZATION

The Am9551 chip is initialized as follows:

a. Reset the chip by writing a command word (with bit 6 set) to hex address ED.

b. Write a MODE instruction word to address ED.

NOTE

If the Am9551 is looking for a MODE when the reset command word is written, multiple reset commands must be issued.

c. If synchronous mode is selected, write one or two sync characters as required.

d. Write a command instruction word to address ED.

NOTE

After initialization, always check the status of the TxRDY bit prior to writing data or a new command word to the Am9551. The TxRDY bit must be true to prevent overwriting and subsequent loss of commands or data. The TxRDY is inactive until initialization has been completed.

Once initialized, it is not necessary for a command instruction to precede all data transactions--only those transmissions that require a change in the command instruction.

3-7. Am9551 MODE INSTRUCTION WORD FORMAT

The mode instruction word defines the general characteristic of the Am9551. Once the mode instruction has been written, sync characters or command instructions may be inserted. The mode instruction word defines the following:

a. For synchronous mode:

Character length
Parity enable/disable
Even/odd parity
Character synchronization
Single or double charcter sync

b. For asynchronous mode:

Baud rate multiplier
Character length
Parity enable/disable
Even/odd parity
Number of stop bits

The mode instruction word formats for synchronous and asynchronous modes are shown in figures 3-1 and 3-2 respectively.

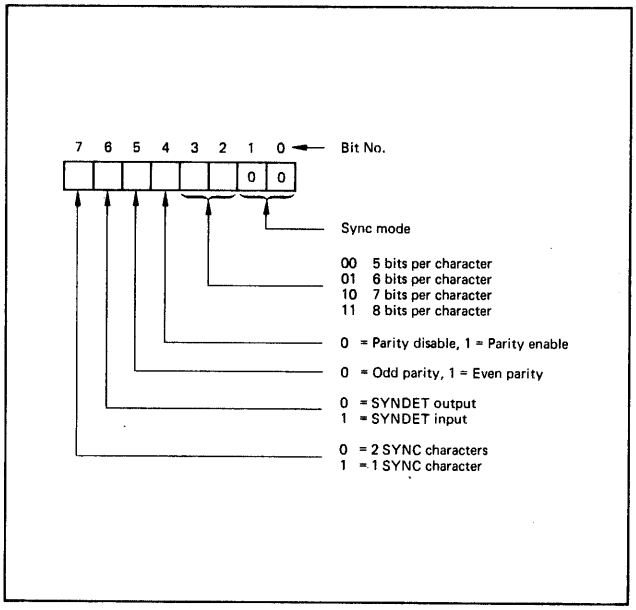


Figure 3-1. Am9551 Synchronous Mode Control Code.

3-8. Am9551 SYNC CHARACTERS

In the synchronous mode, one or two sync characters must be written to address EDH. The format of the sync characters is at the option of the programmer.

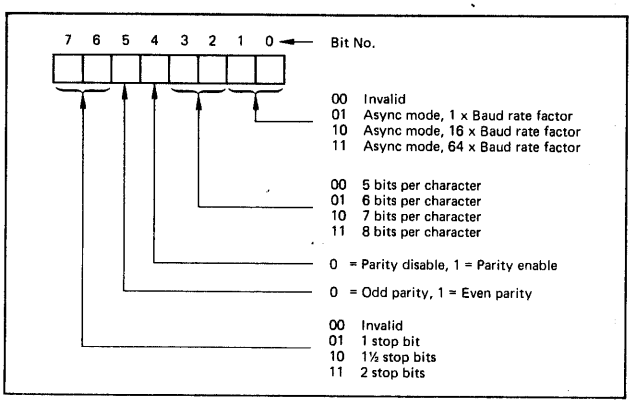


Figure 3-2. Am9551 Asynchronous Mode Control Code.

3-9. COMMAND INSTRUCTION WORD FORMAT

The command instruction word must follow the mode and/or sync words. Once the command word is written, data can be transmitted or received by the Am9551. The format of the command word is shown in figure 3-3.

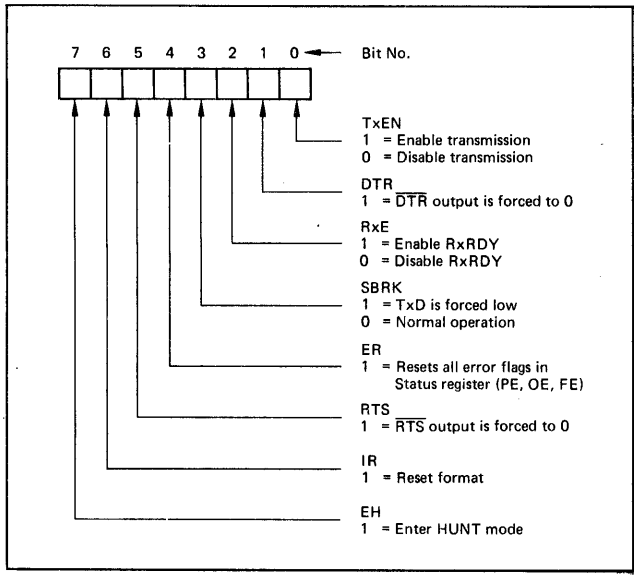


Figure 3-3. Am9551 Command Instruction Word Format.

3-10. Am9551 STATUS READ

The CPU can determine the status of the Am9551 any time by issuing an I/O input to address EDH. The format of the status byte is shown in figure 3-4.

The definition of the status bits is as follows:

TxRDY — Transmitter Ready indicates the Am9551 is ready to accept a data character or command.

RxRDY — Receiver Ready indicates the Am9551 has received a character on its serial input and is ready to transfer it to the CPU.

TxE — Transmitter Empty signals the processor that the transmit register is empty.

PE — Parity Error indicates the character stored in the receive character buffer was received with an incorrect number of binary 1 bits.

OE — Overrun flag is set when a byte stored in the receiver character register is overwritten with a new byte before being transferred to the processor.

FE — Framing Error indicates the asynchronous mode byte stored in the receiver character buffer was received with incorrect character bit format.

SYNDET — When Sync Detect is set for internal sync detect, this bit indicates character sync has been achieved and the Am9551 is ready for data.

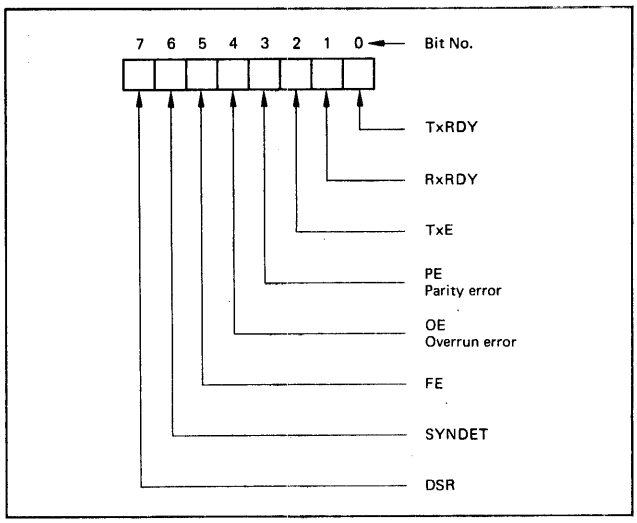


Figure 3-4. Am9551 Status Word Format.

DSR Data Set Ready is set by the external Data Set Ready Signal to indicate the communications data set is ready

3-11. PARALLEL I/O INTERFACE PROGRAMMING

Two Am9555 Programmable Peripheral Interface chips provide 48 parallel signal lines for the transfer and control of data to and from peripheral devices.

Each chip provides three 8-bit ports (A, B, and C). Each port can be configured as either input or output, and port C on each chip is used as control lines for ports A and B in some modes. The operating modes of the ports are controlled by outputting either an operation control word or a bit set/reset control word. Table 3-4 is a complete configuration guide for the Am9555s.

3-12. Am9555 ADDRESSING

Each chip uses four consecutive addresses (E4-E7H and E8-EBH) for control, data transfer, and status read. See table 3-1 for the port addresses and their functions.

3-13. Am9555 INITIALIZATION

The Am9555 chips are initialized by writing an operation control word to address E7H and EBH to define the mode and by writing a bit set/reset control word for Port C control if required.

3-14. Am9555 OPERATION CONTROL WORD FORMAT

The operation control word (bit 0=1) defines three basic modes of operation

Mode 0 = Basic Input/Output

Mode 1 = Strobed Input/Output

Mode 2 = Bidirectional Bus

The modes for port A and B can be separately defined, while port C is divided into two 4-bit ports as required by the port A and port B definitions. Table 3-4 provides a summary of all mode definitions and port restrictions. The mode control word format is shown in figure 3-5.

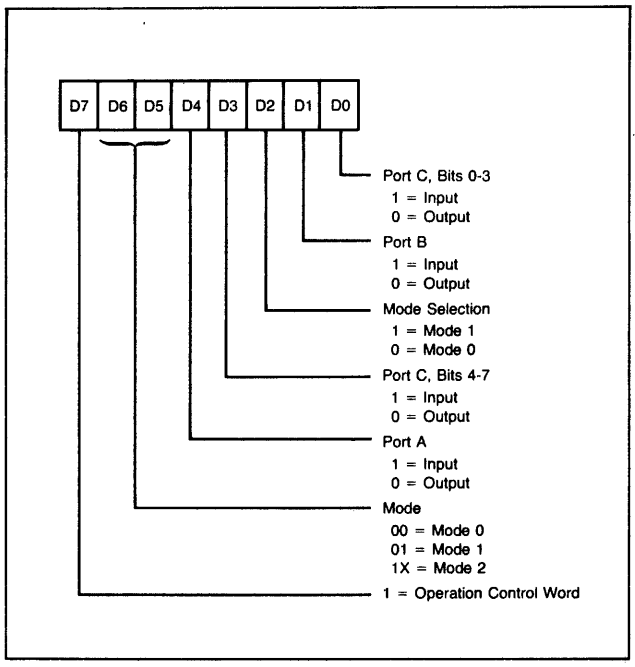


Figure 3-5. Am9555 Operation Control Word Format.

3-15. Am9555 BIT SET/RESET CONTROL WORD

When operating in mode 1 or 2, the bits of port C can be set or reset using the bit set/reset control word. The functions of some port C bits are defined by port A and B operations in modes 1 and 2. Refer to table 3-4 for port C bit definitions in modes 1 and 2. Figure 3-6 shows the bit set/reset control word format.

3-16. Am9555 PORT C STATUS READ

The status of port C can be read at any time by an I/O read to address E6H or EAH. The definition of port C bits are determined by the operating modes of port A and B. Refer to table 3-4 for port C bit definitions.

TABLE 3-4. PARALLEL I/O PORT CONFIGURATION SUMMARY.

Port and Mode	Control Word	Connector Polarity	Driver/Terminator Network	Jumper Action			Port Restrictions
				Delete	Add	Effect	
E4H 0 Input	1001XXXX	Negative True	8226s at U1 and U2	1-3	1-2	Enable input at U1 and U2	Port E5H: None. Port E6H: None unless port E5H is in mode 1.
E4H 0 Output (latched)	1000XXXX	Negative True	8226s at U1 and U2	1-2	1-3	Enable input at U1 and U2	Port E5H: None. Port E6H: None unless port E5H is in mode 1.
E4H 1 Input (strobed)	1011XXXX	Negative True	8226s at U1 and U2. Termination Network at U3.	1-3	1-2	Enable input at U1 and U2.	Port E5H: None. Port E6H: Performs the following dedicated functions: Bit 0, 1, 2: None unless port E5H is in mode 1. Bit 3: INTR (Interrupt Request) output for port E4H. Bit 4: STB* (Strobe) input for port E4H. Bit 5: IBF (Input Buffer Full) output for port E4H. Bits 6 and 7: Can be used for input or output. Both have same direction.
E4H 1 Output (latched)	1010XXXX	Negative True	8226s at U1 and U2. Termination Network at U4	1-2	1-3	Enable Outputs at 8226s	Port E5H: None Port E6H: Performs the following dedicated functions: Bit 0, 1, 2: None unless port E5H is in mode 1. Bit 3: INTR (Interrupt Request) output for port E4.

TABLE 3-4. PARALLEL I/O PORT CONFIGURATION SUMMARY. (Cont.)

Port and Mode	Control Word	Connector Polarity	Driver/Terminator Network	Jumper Action			Port Restrictions
				Delete	Add	Effect	
E4H 1 Output (latched) (continued)							Bits 4 and 5: Can be used for input or output; both have same direction. Bit 6: ACK* (Acknowledge) input for port E4H. Bit 7: OBF* (Output Buffer Full) Output for port E4H.
E4H 2 Bidirectional	11XXXXXX	Negative True	8226s at U1 and U2	1-2 1-3	1-4	Allows ACK_A* output of port E6H to control 8226 direction of data flow.	Port E5H: None Port E6H: Performs the following dedicated functions: Bit 0: Cannot be used. Bits 1 and 2: Can be used as input or output if port E5H is in mode 0. Bit 3: INTR (Interrupt Request) output for port E4H. Bit 4: STR* (Strobe) input for port E4H. Bit 5: IBF (Input Buffer Full) output for port E4H. Bit 6: ACK* (Acknowledge) input for port E4H. Data flow direction control for 8226 via jumper 57-52.† Bit 7: OBF* (Output Buffer Full) output for port E4H.
E5H 0 Input	1XXXX01X	Positive True	Termination Networks at U5 and U6.	None	None		Port E4H: None. Port E6H: None unless port E4H is in mode 1 or 2.

† Jumper use is optional.

TABLE 3-4. PARALLEL I/O PORT CONFIGURATION SUMMARY. (Cont.)

Port and Mode	Control Word	Connector Polarity	Driver/Terminator Network	Jumper Action			Port Restrictions
				Delete	Add	Effect	
E5H 0 Output (latched)	1XXXX00X	Negative True	Driver Networks at U5 and U6.	None	None		Port E4H: None. Port E6H: None unless port E4H is in mode 1 or 2.
E5H 1 Input (strobed)	1XXXX11X	Positive True	Termination Networks at U4, U5, and U6.	Opt.	Opt.		Port E4H: None. Port E6H: Performs the following dedicated functions: Bit 0: INTR (Interrupt Request) output for port E5H. Bit 1: IBF (Input Buffer Full) output for port E5H. Bit 2: STB* (Strobe) input for port E5H. Bit 3: Can be used as input or output if port E4H is in mode 0. Bits 4 to 7: Can be used as input or output if port E4H is in mode 0 or in some combinations where port E4H is in mode 1. These bits are always dedicated when port E4H is in mode 2.

TABLE 3-4. PARALLEL I/O PORT CONFIGURATION SUMMARY. (Cont.)

Port and Mode	Control Word	Connector Polarity	Driver/Terminator Network	Jumper Action			Port Restrictions
				Delete	Add	Effect	
E5H 1 Output (latched)	1XXXX10X	Negative True	Driver Networks at U4, U5 and U6	Opt.	Opt.		Port E4H: None. Port E6H: Performs the following dedicated functions: Bit 0: INTR (Interrupt Request) output for port E5H. Bit 1: OBF* (Output Buffer Full) output for port E5H. Bit 2: ACK* (Acknowledge) input for port E5H. Bit 3: Can be used as input or output if port E8H is in mode 0. Bits 4 to 7: Can be used as input or output if port E4H is in mode 0 in some combinations where port 1 is in mode 1. These bits are always dedicated when port E4H is in mode 2.
E6H High Order Bits 0 Input	100X10XX	Positive True	Termination Network at U3	Opt.	Opt.		Port E4H: Port E4H must be in mode 0 for all four bits to be available. Port E5H: Port E5H must be in mode 0 for all four bits to be available.
E6H Low Order Bits 0 Input	100XX0X1	Positive True	Termination Network at U4	Opt.	Opt.		Port E4H: Port E4H must be in mode 0 for all four bits to be available. Port E5H: Port E5H must be in mode 0 for all four bits to be available.

TABLE 3-4. PARALLEL I/O PORT CONFIGURATION SUMMARY. (Cont.)

Port and Mode	Control Word	Connector Polarity	Driver/Terminator Network	Jumper Action			Port Restrictions
				Delete	Add	Effect	
E6H High Order Bits 0 Output (latched)	100X00XX	Negative True	Driver Network at U3	Opt.	Opt.		Port E4H: Port E4H must be in mode 0 for all four bits to be available. Port E5H: Port E5H must be in mode 0 for all four bits to be available.
E6H Low Order Bits 0 Output (latched)	100XX0X0	Negative True	Driver Network at U3	Opt.	Opt.		Port E4H: Port E4H must be in mode 0 for all four bits to be available. Port E5H: Port E5H must be in mode 0 for all four bits to be available.
E8H 0 Input	1001XXXX	Negative True	8226s at U7 and U8	22-23	21-23	Enable inputs at U7 and U8	Port E9H: None. Port EAH: None unless port E9H is in mode 1.
E8H 0 Output (latched)	1000XXXX	Negative True	8226s at U7 and U8	21-23	22-23	Enable outputs at U7 and U8	Port E9H: None. Port EAH: None unless port E9H is in mode 1.
E8H 1 Input (strobed)	1011XXXX	Negative True	8226s at U7 and U8 Termination Network at U9	22-23	21-23	Enable inputs at U7 U8	Port E9H: None. Port EAH: Performs the following dedicated functions: Bits 0, 1, 2: None unless port E9 is in mode 1. Bit 3: INTR (Interrupt Request) output for port E8H. Bit 4: STR* (Strobe) input for port E8H.

TABLE 3-4. PARALLEL I/O PORT CONFIGURATION SUMMARY. (Cont.)

Port and Mode	Control Word	Connector Polarity	Driver/Terminator Network	Jumper Action			Port Restrictions
				Delete	Add	Effect	
E8H 1 Input (strobed) (continued)							Bit 5: IBF (Input Buffer Full) output for port E8H. Bits 6 and 7: Can be used for input or output. Both have same direction.
E8H 1 Output (latched)	1010XXXX	Negative True	8226s at U7 and U8. Termination Network at U10.	21-23	22-23	Enable outputs at U7 and U8.	Port E9H: None. Port EAH: Performs the following dedicated functions: Bits 0, 1, 2: None unless port E9H is in mode 1. Bit 3: INTR (Interrupt (Request) output for port E84. Bits 4 and 5: Can be used for input or output; both have same direction. Bit 6: ACK* (Acknowledge) input for E8H. Bit 7: OBF* (Output Buffer Full) output for port E4H.

TABLE 3-4. PARALLEL I/O PORT CONFIGURATION SUMMARY. (Cont.)

Port and Mode	Control Word	Connector Polarity	Driver/Terminator Network	Jumper Action			Port Restrictions
				Delete	Add	Effect	
E8H 2 Bidirectional	11XXXXXX	Negative True	8226s at U7 and U8. Termination Network at U9. Driver Network at U10.	21-23 22-23	23-24	Allows ACK_A* output of port EAH to control 8226 direction of data flow.	Port E9H: None. Port EAH: Performs the following dedicated functions: Bit 0: Cannot be used. Bits 1 and 2: Can be used as input or output if port E9H is in mode 0. Bit 3: INTR (Interrupt Request) output for port E8H. Bit 4: STB* (Strobe) input for port E8H. Bit 5: IBF (Input Buffer Full) output for port E8H.
E9H 0 Output	1XXXX01X	Positive True	Termination Network at U11 and U12	None	None		Port E8H: None. Port EAH: None unless port E8H is in mode 1 or 2.
E9H 0 Output (latched)	1XXXX00X	Negative True	Driver Network at U11 and U12	None	None		Port E8H: None. Port EAH: None unless port E8H is in mode 1 or 2.
E9H 1 Input (strobed)	1XXXX11X	Positive True	Termination Network at U10, U11 and U12	Opt.	Opt.		Port E8H: None. Port EAH: Performs the following dedicated functions: Bit 0: INTR (Interrupt Request) output for port E9H. Bit 1: IBF (Input Buffer Full) output for port E9H.

TABLE 3-4. PARALLEL I/O PORT CONFIGURATION SUMMARY. (Cont.)

Port and Mode	Control Word	Connector Polarity	Driver/Terminator Network	Jumper Action			Port Restrictions
				Delete	Add	Effect	
E9H 1 Input (strobed) (continued)							Bit 2: STB* (Strobe) input for port E9H. Bit 3: Can be used as input or output if port E8H is in mode 0. Bits 4 through 7: Can be used as input or output if port E8H is in mode 0 or in some combinations where port E8H is in mode 1. These bits are always dedicated when port E8H is in mode 2.
E9H 1 Output (latched)	1XXXX1OX	Netative True	Driver Network at U10, U11, and U12.	Opt.	Opt.		Port E8H: None. Port EAH: Performs the following dedicated functions: Bit 0: INTR (Interrupt Request) output for port E9H. Bit 1: OBF* (Output Buffer Full) output for port E9H. Bit 2: ACK* (Acknowledge) input for port E9H. Bit 3: Can be used as input or output if port E8H is in mode 0 or some combinations of mode 1. These bits are always reserved when port E8H is in mode 2.

TABLE 3-4. PARALLEL I/O PORT CONFIGURATION SUMMARY. (Cont.)

Port and Mode	Control Word	Connector Polarity	Driver/Terminator Network	Jumper Action			Port Restrictions
				Delete	Add	Effect	
EAH High Order Bits O Input	100X10XX	Positive True	Termination Network at U9.	Opt.	Opt.		Port E8H: Port E8H must be in mode 0 for all four bits to be available. Port E9H: Port E9H must be in mode 0 for all four bits to be available.
EAH Low Order Bits O Input	100XX0X1	Positive True	Termination Network at U10.	Opt.	Opt.		Port E8H: Port E8H must be in mode 0 for all four bits to be available. Port E9H: Port E9H must be in mode 0 for all four bits to be available.
EAH High Order Bits O Output (latched)	100X00XX	Negative True	Driver Network at U9.	Opt.	Opt.		Port E8H: Port E8H must be in mode 0 for all four bits to be available. Port E9H: Port E9H must in mode 0 for all four bits to be available.
EAH Low Order Bits O Output (latched)	100XX0X0	Negative True	Driver Network at U10.	Opt.	Opt.		Port E8H: Port E8H must be in mode 0 for all four bits to be available. Port E9H: Port E9H must be in mode 0 for all four bits to be available.

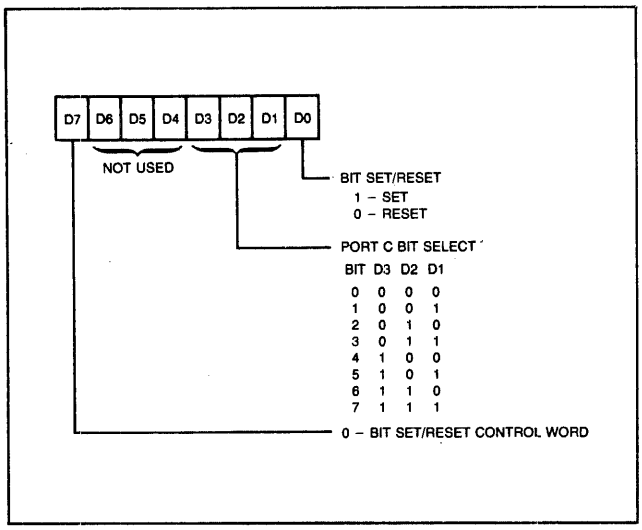


Figure 3-6. Bit Set/Reset Control Word Format.

3-17. INTERRUPT CONTROL PROGRAMMING

An Am9519 Universal Interrupt Controller chip provides priority resolution for eight interrupt levels. A jumper pad provides for connecting the eight interrupt lines to 16 possible interrupt requests. When one or more of the eight interrupt inputs are true, the Am9519 determines the following.

Which unmasked input has the highest priority.

Whether the input has a higher priority than the input presently being serviced. If so, the interrupt being serviced is interrupted; if not, the input signal is held for later service.

Commands entered are loaded into the comand register. Depending upon the specific command that is entered, an immediate internal activity could be initiated or the chip might be preconditioned for subsequent data transfers (as in the preselect commands). Commands controlling the operating modes or commands that preselect other registers for loading from the data bus are loaded into the mode register from the command register. The bit assignments of the mode register are shown in figure 3-7.

3-18. Am9519 ADDRESSING

The Am9519 uses two consecutive I/O addresses (C2H and C3H) for writing commands and vector data and reading status. The addresses and their functions are shown in table 3-1.

3-19. Am9519 INITIALIZATION

Before the Am9519 can operate, it must be initialized for a specific application and loaded with appropriate response data. Operating modes, controlling bits, and response data can be modified while the controller is operating. The following are the basic steps required to initialize the Am9519.

a. Disable the CPU interrupts.

b. Issue a reset command to address C3H.

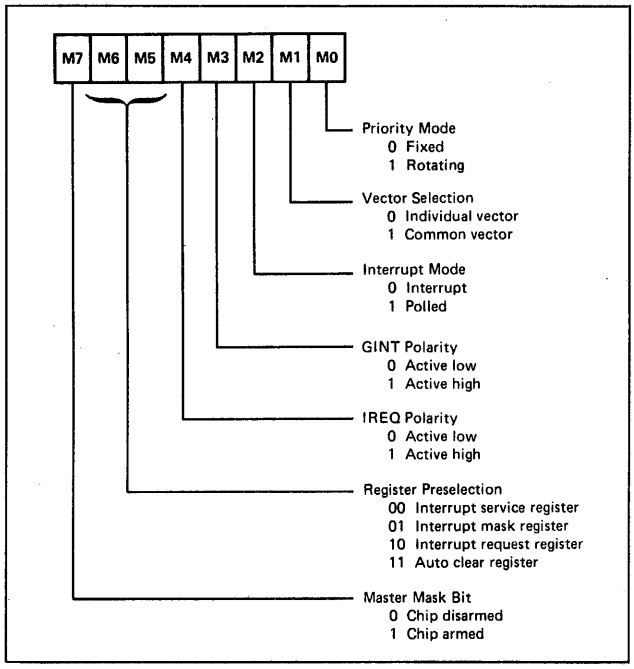


Figure 3-7. Am9519 Mode Register Bit Assignments.

c. Issue the appropriate commands to address C3H defining the operating mode and setting any mask bits required.

d. Load the data register (address C2H) with the appropriate response data.

e. Enable CPU interrupts.

3-20. Am9519 Command Descriptions

The Am9519 command set allows the programmer to customize and alter the interrupt operating modes, initialize and update the response data registers, and manipulate the internal controlling bit set during interrupt servicing. All the available commands are described in the following paragraphs.

RESET

C7	C6	C5	C4	C3	C2	C1	C0
0	0	0	0	0	0	0	0

Description: The Reset command allows the host processor to establish a known internal condition. The response memory and byte count registers are not affected by the software reset. The IMR is set to all ones. The ISR, IRR, ACR and Mode registers are cleared to all zeros.

CLEAR IRR AND IMR

C7	C6	C5	C4	C3	C2	C1	C0
0	0	0	1	0	X	X	X

Description: All bits in the IMR and all bits in the IRR are cleared at the same time. Thus, all interrupts are enabled and the previous history of all IREQ transitions is forgotten. If GINT was active when the command was entered, it will go inactive.

CLEAR SINGLE IMR AND IRR BIT

C7	C6	C5	C4	C3	C2	C1	C0
0	0	0	1	1	B2	B1	B0

Description: The same single bit position is cleared in both the IMR and the IRR. Other bits are not changed. If the specified Irr bit was generating an active interrupt output, GINT may go inactive upon entry of the command. The bit position cleared is specified by the octal number contained in the B2, B1, B0 field.

CLEAR IMR

C7	C6	C5	C4	C3	C2	C1	C0
0	0	1	0	0	X	X	X

Description: All bits in the IMR are cleared to zeros. All IRR bits will therefore be unmasked and any IRR bits that had been set will be able to cause an active GINT output after the command is entered.

CLEAR SINGLE IMR BIT

C7	C6	C6	C5	C4	C3	C2	C1
0	0	1	0	1	B2	B1	B0

Description: A single bit in the IMR is cleared. Other bits are not changed. If the corresponding bit in the IRR was set, it will be unmasked and will be able to cause an active GINT after entry of the command. The IMR bit cleared is specified by the octal number expressed by the B2, B1, B0 field.

SET IMR

C7	C6	C5	C4	C3	C2	C1	C0
0	0	1	1	0	X	X	X

Description: All bis in the IMR are set to ones. All IRR bits will therefore be masked and unable to generate an active GINT. If GINT had been active, it will go inactive after the command is entered.

SET SINGLE IMR BIT

C7	C6	C5	C4	C3	C2	C1	C0
0	0	1	1	1	B2	B1	B0

Description: A single bit in the IMR is set. Other bits are not changed. If the corresponding bit in the IRR was active and generating a GINT output, GINT will become inactive after the command is entered. The IMR bit set is specified by the octal number contained in the B2, B1, B0 field.

CLEAR IRR

C7	C6	C5	C4	C3	C2	C1	C0
0	1	0	0	0	X	X	X

Description: All bits in the IRR are cleared to zeros. GINT will become inactive. New transitions on the IREQ inputs will be necessary to cause an interrupt.

CLEAR SINGLE IRR BIT

C7	C6	C5	C4	C3	C2	C1	C0
0	1	0	0	1	B"	B1	B0

Description: A single bit in the IRR is cleared to zero. It will not cause an active GINT until it is set. The IRR bit cleared is specified by the octal number contained in the B2, B1, B0 field.

SET IRR

C7	C6	C5	C4	C3	C2	C1	C0
0	1	0	1	0	X	X	X

Description: All bits in the IRR are set to ones. Any that are unmasked will be able to cause an active GINT output. This command allows the host CPU to initiate eight interrupts in parallel.

SET SINGLE IRR BIT

C7	C6	C5	C4	C3	C2	C1	C0
0	1	0	1	1	B2	B1	B0

Description: A single bit in the IRR is set to a logical one. If it is unmasked, it will be able to generate an active GINT. This command allows the host processor to simulate with software the arrival of a hardware interrupt request. It also gives the software access to the hardware priority resolution, masking and control features of the Am9519. The bit set is specified by the octal number contained in the B2, B1, B0 field.

CLEAR HIGHEST PRIORITY ISR BIT

C7	C6	C5	C4	C3	C2	C1	C0
0	1	1	0	X	X	X	X

Description: A single bit in the ISR is cleared to zero. If only one bit was set, that is the one cleared. If more than one bit was set, this command clears the one with the highest priority. This command is useful in software contexts where the service routine does not know which device is being serviced. It should be used with caution since the highest priority

ISR bit may not really be the bit intended. When using the auto clear option on some interrupts and/or when a subroutine nesting hierarchy is not priority driven, the highest priority ISR bit may not correspond to the one being serviced.

CLEAR ISR

C7	C6	C5	C4	C3	C2	C1	C0
0	1	1	1	0	X	X	X

Description: All bits in the ISR are cleared to zeros. Mask fencing is elimated.

CLEAR SINGLE ISR BIT

C7	C6	C5	C4	C3	C2	C1	C0
0	1	1	1	0	X	X	X

Description: A single bit in the ISR is cleared to zero. If the bit was already cleared, no effective operation takes place. The bit cleared is specified by the octal number contained in the B2, B1, B0 field. This will be the most useful command for service routines to use in managing the ISR without the help of the auto-clear option.

LOAD MODE BITS M0 THROUGH M4

C7	C6	C5	C4	C3	C2	C1	C0
1	0	0	M4	M3	M2	M1	M0

Description: The five low order bits of the command register are transferred into the five low order bits of the Mode register. This command controls all of the Mode options except the master mask and the register preselection,

CONTROL MODE BITS M5, M6, M7

C7	C6	C5	C4	C3	C2	C1	C0
1	0	1	0	M6	M5	N1	N0

Description: The M6, M5 field in the command is loaded into the M6, M5 locations in the Mode register. This field controls the register preselection bits in the Mode register. The N1, N0 field in the command controls Mode bit M7 (Master Mask) and is decoded as follows:

N1	N0	
0	0	No change to M7
0	1	Set M7
1	0	Clear M7
1	1	(illegal)

Thus, this command may be considered as three distinct commands, depending on the coding of N1 and N0.

1. Load M5, M6 only
2. Load M5, M6 and set M7
3. Load M5, M6 and clear M7

PRESELECT IMR FOR WRITING

C7	C6	C5	C4	C3	C2	C1	C0
1	0	1	1	X	X	X	X

Description: The IMR is targeted to be loaded from the data bus when the next write operation occurs at the data port. All subsequent data write operatons will also load the IMR until a different command is entered. Read operations may be successfully inserted between the entry of this command and the sub-

sequent writing of data into the IMR. The Mode register is not affected by this command.

PRESELECT ACR FOR WRITING

C7	C6	C5	C4	C3	C2	C1	C0
1	1	0	0	X	X	X	X

Description: The ACR is targeted to be loaded from the data bus when the next write operation occurs at the data port. All subsequent data write operations will also load the ACR until a different command is entered. Read operations may be successfully inserted between the entry of this command and the subsequent writing of data into the ACR. The Mode register is not affected by this command.

PRESELECT RESPONSE MEMORY FOR WRITING

C7	C6	C5	C4	C3	C2	C1	C0
1	1	1	BY1	BY0	L2	L1	L0

Description: One level in the response memory is targeted for loading from the data bus by subsequent data write operations. The byte count register for that level is loaded from the BY1, BY0 field in the command. The L2, L1, L0 field specifies which of the eight response levels is being selected. This command should be followed by one to four data write operations to load response bytes. Field coding:

BY1	BY0	Count
0	0	1
0	1	2
1	0	3
1	1	4

L2	L1	L0	Level
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

The byte count value does not control the number of bytes entered into the response memory. It does control the number of bytes read from the memory by IACK* pulses. Response bytes are output by the Am9519 in the same order they were entered.

3-21. Am9519 STATUS READ

The status register is eight bits wide and contains information describing the internal state of the Am9519 chip. The status register is read directly by executing an I/O read to port C3H. The status bit assignments are shown in figure 3-8.

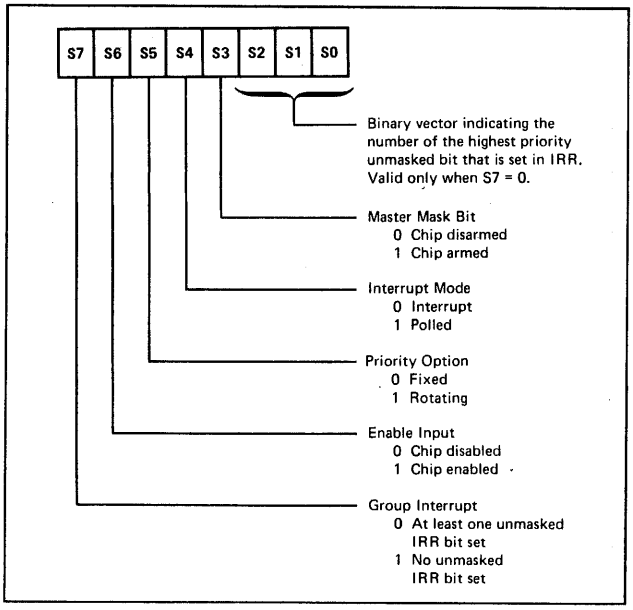


Figure 3-8. Am9519 Status Word Definition.

3-22. MULTIMODE DIRECT MEMORY ACCESS (DMA) CONTROLLER PROGRAMMING

An Am9517 Multimode DMA Controller provides the capability for external devices to directly transfer information to or from memory. The Am9517 contains four separate channels. Each channel can be programmed for one of three different transfer modes, and each transfer mode can perform three different types of transfers. The transfer modes include single transfer mode, block transfer mode, and demand transfer mode. Within the modes, the three types of transfers are read, write, and verify. Memory-to-memory block moves can be performed. The operation of the DMA Controller is programmed by writing control words to an internal register; address and word count values are sent to the four channel address and word count registers.

3-23. Am9517 ADDRESSING

The Am9517 uses 16 consecutive addresses (B0H through BFH) for reading and writing to the internal registers. See table 3-1 for the port addresses and their function.

3-24. Am9517 Command Register

Am9517 operation is controlled by the 8-bit command register. The register is programmed by a I/O Write to address B8H. Command register control word format is shown in figure 3-9.

3-25. Am9517 Mode Register

Each channel has a 6-bit mode register. The mode register is programmed by an I/O write to address BBH. Bits 0 and 1 of the mode control word define the channel mode register to be programmed. The format of the mode register control word is shown in figure 3-10.

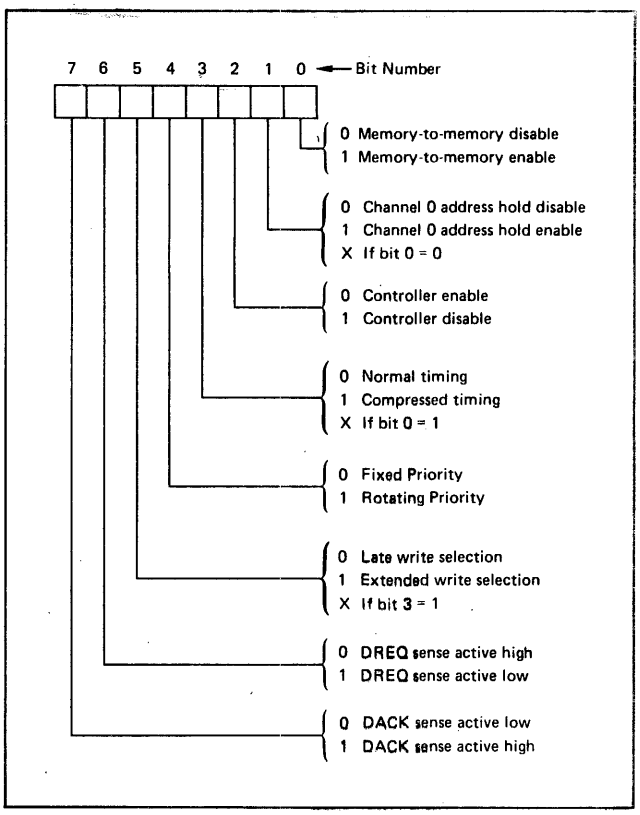


Figure 3-9. Am9517 Command Register Control Word Format.

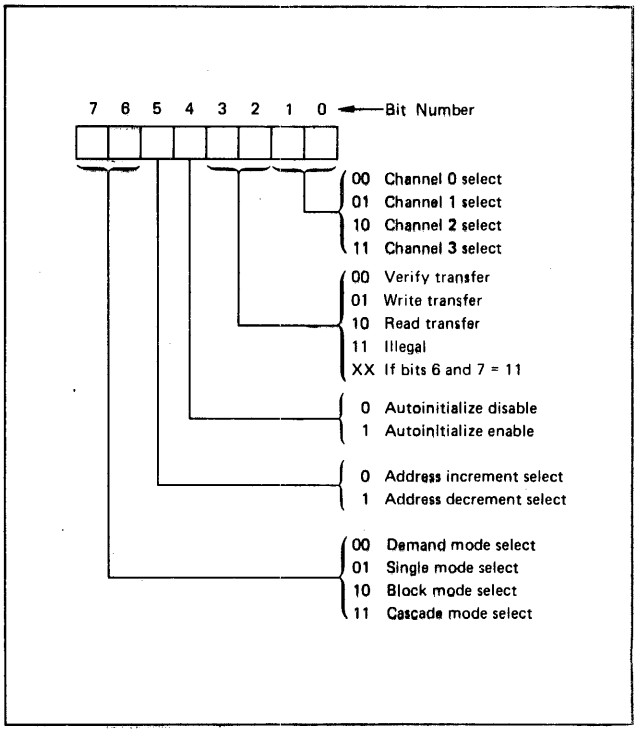


Figure 3-10. Am9517 Mode Register Control Word.

3-26. Am9517 Request Register

The Am9517 provides a software service request. Each channel has a request bit associated with it in the 4-bit request register. These bits are non-maskable and are subject to prioritization by the priority encoder. To set or reset a request register bit, write a data word to I/O address B9. The format of the request register data word is shown in figure 3-11.

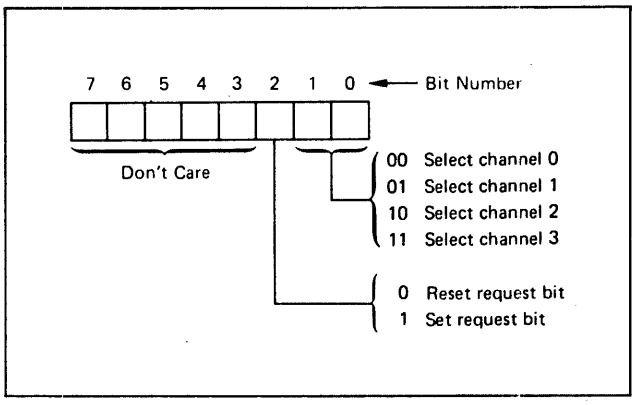


Figure 3-11. Am9517 Request Register Data Word.

NOTE

Software requests will be serviced only if the channel is in block mode. When initiating a memory-to-memory transfer, the software request for channel 0 should be set.

3-27. Am9517 Mask Register

A 4-bit mask register provides a mask bit for each of the four DMA channels. These mask bits select which channels will respond to the incoming DMA Request (DREQ) signal. A mask bit is set when its associated channel produces an EOP* signal and the channel is not programmed to autoinitialize. The entire register is set by an external reset. Setting the entire register disables all DMA requests until a clear mask register instruction allows them to occur. The instruction used to separately set and clear the mask bits is shown in figure 3-12. All four bits can be set or reset with a single command as shown in figure 3-13.

3-28. Am9517 Temporary Register

A temporary register is used to hold data during memory-to-memory transfers. Following a memory-to-memory transfer, the last word moved can be read by the CPU with an I/O read to address BDH. The temporary register always contains the last byte transferred in the previous memory-to-memory operation, unless cleared by a reset.

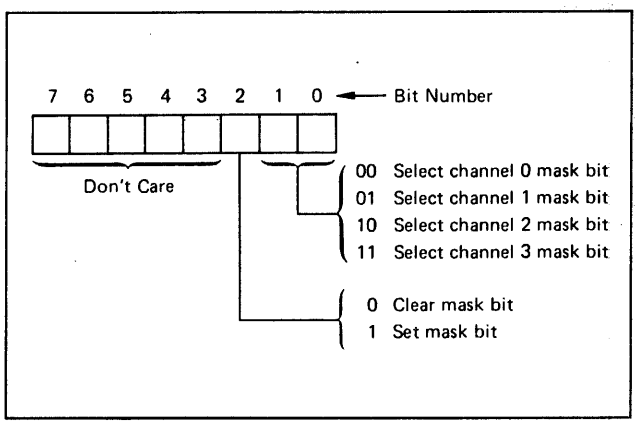


Figure 3-12. Am9517 Mask Register Single-Bit Instruction.

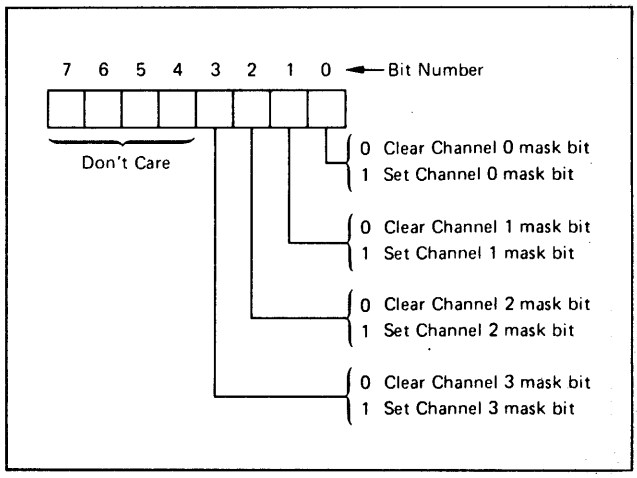


Figure 3-13. Am9517 Mask Register Four-Bit Instruction.

3-29. Am9517 Clear First/Last Flip/Flop Command

Prior to writing or reading new address or word count information to the Am9517, a clear first/last flip/flop command to address BCH must be executed. This initializes the flip/flop to a known state so that subsequent CPU accesses to the register contents will address upper and lower bytes in the correct sequence. The state of the data bus does not affect this command.

3-30. Am9517 Master Clear

The Am9517 chip can be reset at any time with a master clear command to address BDH. The master clear command has the same effect as a hardware reset. The master clear command clears all internal registers and sets the mask register. The state of the data bus does not affect this command.

3-31. Am9517 STATUS READ

The status register is available to be read by the CPU at address B8H. It contains information about the current status of the chip. This status information includes which channels have reached a terminal count (TC) and which channels have a pending DMA request. The format of the status word is shown in figure 3-14.

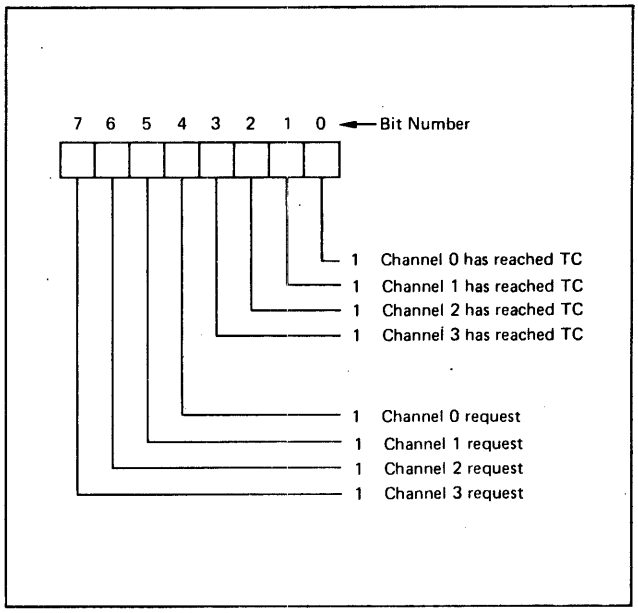


Figure 3-14. Am9517 Status Word Format.

3-32. ARITHMETIC PROCESSING UNIT PROGRAMMING

The Am9511 Arithmetic Processing Unit (APU) provides high performance fixed and floating point arithmetic and floating point transcendental and mathematical operations.

All transfers (including operands, results, and commands) take place on the data bus. Operands are pushed onto an internal 8-level, 16-bit wide data stack and a command is issued to perform operations on the data in the stack. Results are then retrieved from the stack, or additional commands may be entered.

Transfers to and from the APU can be handled by the CPU by I/O read and writes or by the Am9517 Multimode DMA Controller. Upon completion of each command, the APU issues an end of execution signal that can be used as an interrupt to the CPU.

3-33. Am9511 ADDRESSING

The APU uses two consecutive addresses (C0H and C1H) for commands, data transfers, and status read. See table 3-1 for the port addresses and their functions.

3-34. Am9511 INITIALIZATION

The APU does not require special initialization. After a power-on or system reset operation, the status register is clear and the APU is in the idle state.

3-35. Am9511 DATA FORMATS

The Am9511 APU handles operands in both fixed-point and floating-point formats. Within the internal stack, data is logically organized as 16-bit or 32-bit

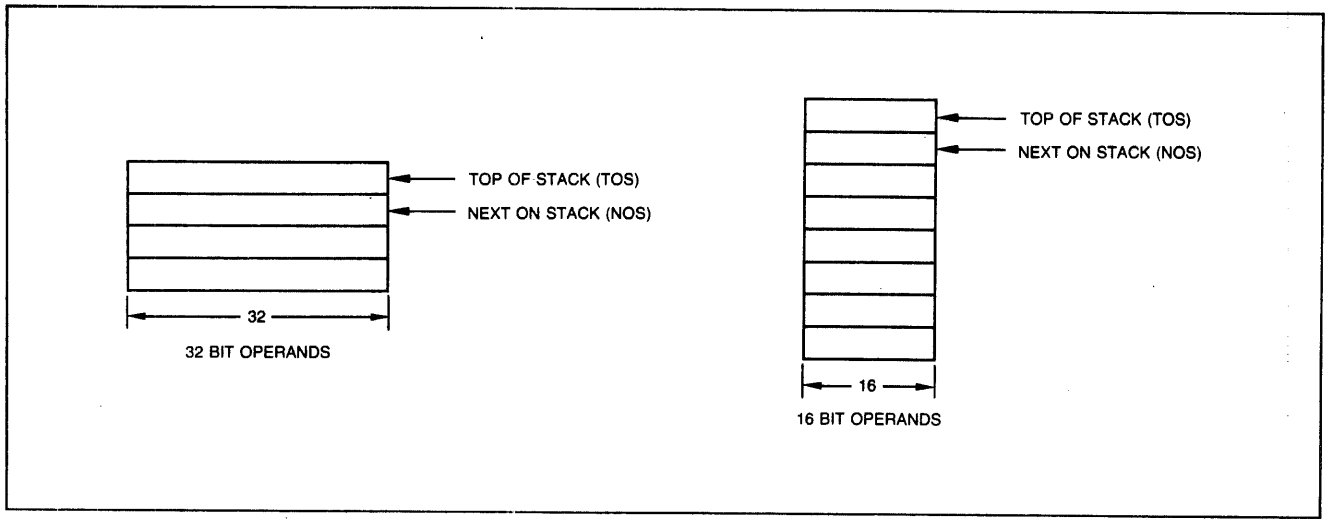


Figure 3-15. Am9511 Data Stack Configurations.

operands as shown in figure 3-15. The data stack operates as a true push-down first-in/last-out (FILO) stack; the data first written in is the last data read out. Within each stack entry the least significant byte is entered first and retrieved last. Since all words are entered as 8-bit bytes, data must be entered into the stack in multiples of the number of bytes appropriate to the chosen data format.

3-36. Fixed-Point

Fixed-point operands can be represented in either single (16-bit) or double (32-bit) precision formats. They are always represented as binary, twos complement values. The single precision and double precision fixed-point word formats are shown in figure 3-16. The sign of the operand is located in the most significant bit position. Positive values are represented by a sign bit of 0; negative values are represented by a sign bit of 1. The range of values that can be expressed by the single precision format is -32,768 to +32,767. The double precision value range is from -2,147,483,648 to +2,147,483,647.

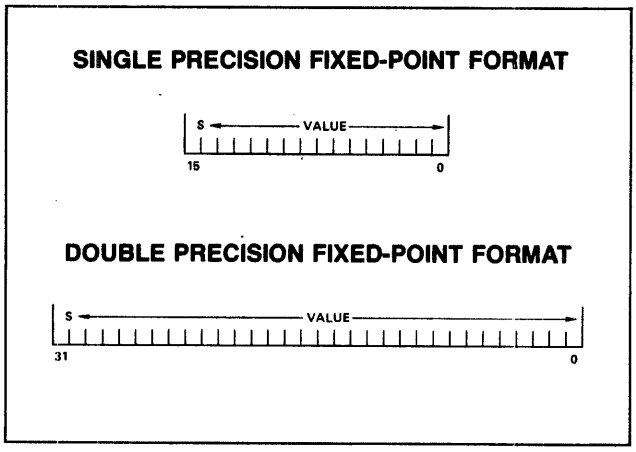


Figure 3-16. Fixed-Point Word Formats.

3-37. Floating-Point

The 32-bit floating-point format is shown in figure 3-17. Bit 31 indicates the sign of the mantissa. The next seven bits form the exponent, with bit 30 representing the exponent sign. Bits 0 through 23 form the mantissa value.

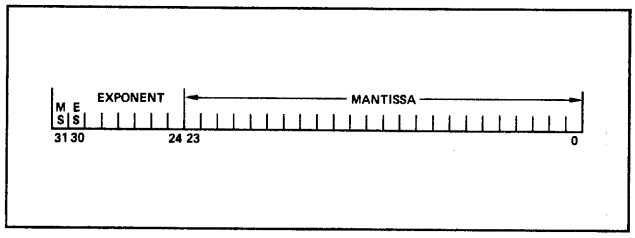


Figure 3-17. Floating-Point Word Format.

The mantissa is a sign-magnitude number with an assumed binary point just to the left of the most significant mantissa bit (bit 23). The exponent is interpreted as a power of two and is

expressed as a twos complement value having a range of from -64 to +63 (2^{-64} to 2^{+63}).

All floating-point values must be normalized, which makes bit 23 always equal to 1 except when representing a value of zero. The number zero is represented with binary zeros in all 32 bit positions.

3-38. Am9511 COMMAND DESCRIPTIONS

The following detailed description of the Am9511 commands are presented in alphabetical order by command mnemonic. In the descriptions, TOS means Top of Stack and NOS means Next on Stack. Figure 3-18 shows the command format.

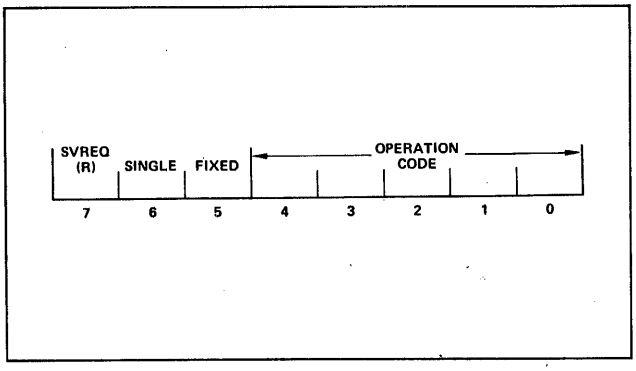


Figure 3-18. Am9511 Command Format.

All derived functions except square root use Chebyshev polynomial approximating algorithms. This approach is used to minimize the maximum error values and to provide a relatively even distribution of errors over the data range. The basic arithmetic operations are used by the derived functions to compute the various Chebyshev terms. The basic operations can produce error codes in the status register as a result.

Execution times are listed in terms of clock cycles and can be converted into time values by multiplying by the clock period used. For example, an execution time of 44 clock cycles when running at a 2MHz rate translates to 22 microseconds (44 x .5µs = 22µs); the same 44 clock cycles translate to 14.5 microseconds when running at a 3MHz rate (44 x .33µs = 14.5µs). Variations in execution cycles reflect the data dependency of the algorithms.

In some operations, exponent overflow or underflow is possible. When this occurs, the exponent returned in the result will be 128 greater or smaller than its true value.

Many of the functions use portions of the data stack as scratch storage during development of the results. Thus, previous values in those stack locations will be lost. Scratch locations destroyed are listed in the command descriptions and shown with the crossed-out locations in the Stack Contents After diagram.

3-39. Am9511 STATUS READ

The APU status register is read by executing an I/O read to port C1H. When the status busy bit (bit 7) is high, the APU is processing a previously entered command and the balance of the status register is not valid. The definition of the status bits is given in figure 3-19.

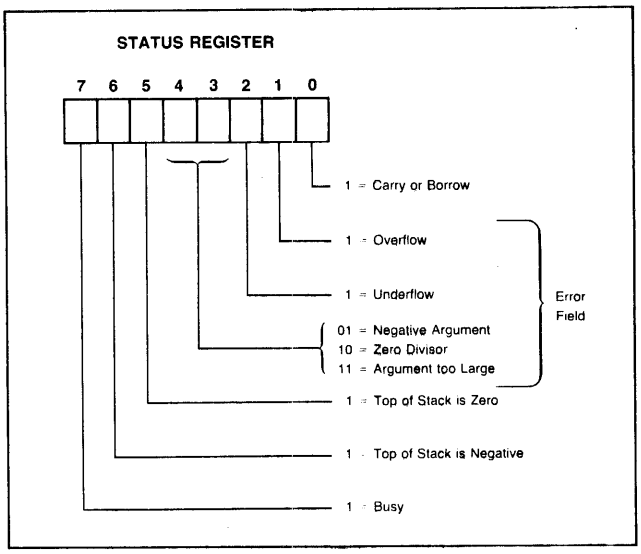


Figure 3-19. Am9511 Status Register Bit Definitions.

ACOS

32-BIT FLOATING-POINT INVERSE COSINE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	0	1	1	0

Hex Coding: 86 with sr = 1
06 with sr = 0

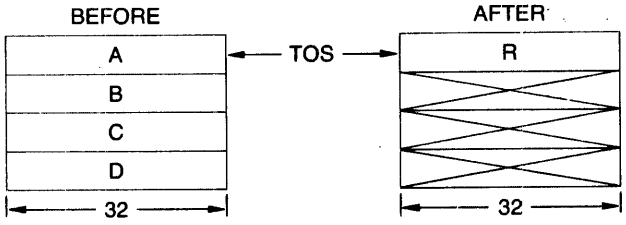
Execution Time: 6304 to 8284 clock cycles

Description:
The 32-bit floating-point operand A at the TOS is replaced by the 32-bit floating-point inverse cosine of A. The result R is a value in radians between 0 and π. Initial operands A, B, C and D are lost.
ACOS will accept all input data values within the range of −1.0 to +1.0. Values outside this range will return an error code of 1100 in the status register.

Accuracy: ACOS exhibits a maximum relative error of 2.0 x 10^{-7} over the valid input data range.

Status Affected: Sign, Zero, Error Field

STACK CONTENTS



ASIN

32-BIT FLOATING-POINT INVERSE SINE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	0	1	0	1

Hex Coding: 85 with sr = 1
05 with sr = 0

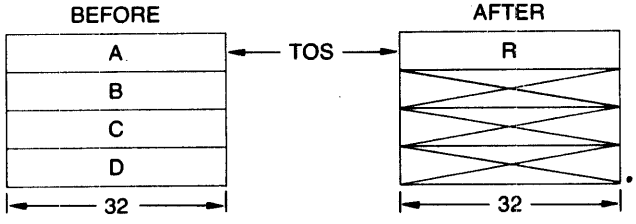
Execution Time: 6230 to 7938 clock cycles

Description:
The 32-bit floating-point operand A at the TOS is replaced by the 32-bit floating-point inverse sine of A. The result R is a value in radians between $-\pi/2$ and $+\pi/2$. Initial operands A, B, C and D are lost.
ASIN will accept all input data values within the range of −1.0 to +1.0. Values outside this range will return an error code of 1100 in the status register.

Accuracy: ASIN exhibits a maximum relative error of 4.0 x 10^{-7} over the valid input data range.

Status Affected: Sign, Zero, Error Field

STACK CONTENTS



ATAN

32-BIT FLOATING-POINT INVERSE TANGENT

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	0	1	1	1

Hex Coding: 87 with sr = 1
07 with sr = 0

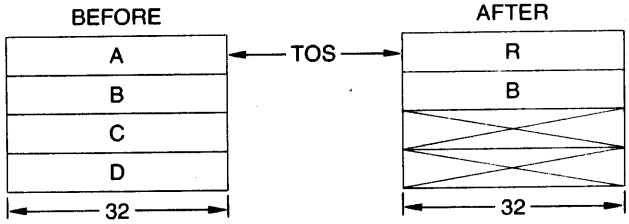
Execution Time: 4992 to 6536 clock cycles

Description:
The 32-bit floating-point operand A at the TOS is replaced by the 32-bit floating-point inverse tangent of A. The result R is a value in radians between $-\pi/2$ and $+\pi/2$. Initial operands A, C and D are lost. Operand B is unchanged.
ATAN will accept all input data values that can be represented in the floating point format.

Accuracy: ATAN exhibits a maximum relative error of 3.0 x 10^{-7} over the input data range.

Status Affected: Sign, Zero

STACK CONTENTS



CHSD

32-BIT FIXED-POINT SIGN CHANGE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	1	1	0	1	0	0

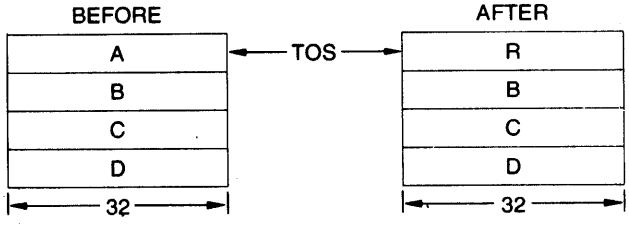
Hex Coding: B4 with sr = 1
34 with sr = 0

Execution Time: 26 to 28 clock cycles

Description:
The 32-bit fixed-point two's complement integer operand A at the TOS is subtracted from zero. The result R replaces A at the TOS. Other entries in the stack are not disturbed.
Overflow status will be set and the TOS will be returned unchanged when A is input as the most negative value possible in the format since no positive equivalent exists.

Status Affected: Sign, Zero, Error Field (overflow)

STACK CONTENTS



CHSF

32-BIT FLOATING-POINT SIGN CHANGE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	0	1	0	1

Hex Coding: 95 with sr = 1
15 with sr = 0

Execution Time: 16 to 20 clock cycles

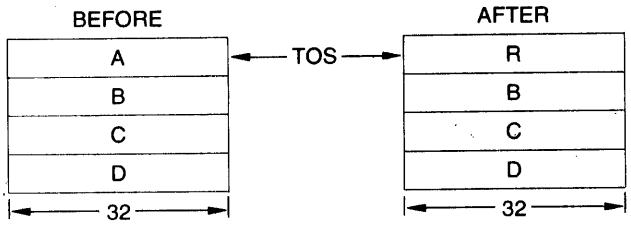
Description:

The sign of the mantissa of the 32-bit floating-point operand A at the TOS is inverted. The result R replaces A at the TOS. Other stack entries are unchanged.

If A is input as zero (mantissa MSB = 0), no change is made.

Status Affected: Sign, Zero

STACK CONTENTS



CHSS

16-BIT FIXED-POINT SIGN CHANGE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	1	1	1	0	1	0	0

Hex Coding: F4 with sr = 1
74 with sr = 0

Execution Time: 22 to 24 clock cycles

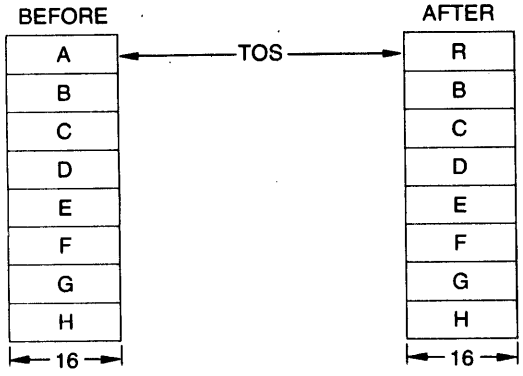
Description:

16-bit fixed-point two's complement integer operand A at the TOS is subtracted from zero. The result R replaces A at the TOS. All other operands are unchanged.

Overflow status will be set and the TOS will be returned unchanged when A is input as the most negative value possible in the format since no positive equivalent exists.

Status Affected: Sign, Zero, Overflow

STACK CONTENTS



COS

32-BIT FLOATING-POINT COSINE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	0	0	1	1

Hex Coding: 83 with sr = 1
03 with sr = 0

Execution Time: 3840 to 4878 clock cycles

Description:

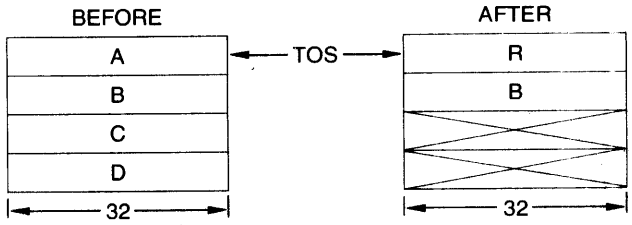
The 32-bit floating-point operand A at the TOS is replaced by R, the 32-bit floating-point cosine of A. A is assumed to be in radians. Operands A, C and D are lost. B is unchanged.

The COS function can accept any input data value that can be represented in the data format. All input values are range reduced to fall within an interval of $-\pi/2$ to $+\pi/2$ radians.

Accuracy: COS exhibits a maximum relative error of 5.0 x 10^{-7} for all input data values in the range of -2π to $+2\pi$ radians.

Status Affected: Sign, Zero

STACK CONTENTS



DADD

32-BIT FIXED-POINT ADD

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	1	0	1	1	0	0

Hex Coding: AC with sr = 1
2C with sr = 0

Execution Time: 20 to 22 clock cycles

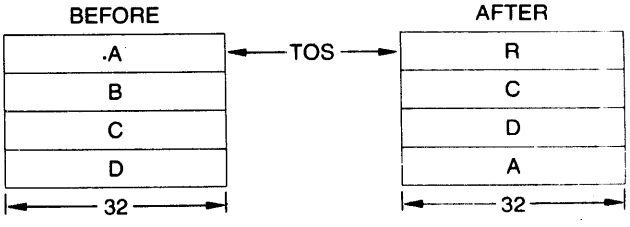
Description:

The 32-bit fixed-point two's complement integer operand A at the TOS is added to the 32-bit fixed-point two's complement integer operand B at the NOS. The result R replaces operand B and the Stack is moved up so that R occupies the TOS. Operand B is lost. Operands A, C and D are unchanged. If the addition generates a carry it is reported in the status register.

If the result is too large to be represented by the data format, the least significant 32 bits of the result are returned and overflow status is reported.

Status Affected: Sign, Zero, Carry, Error Field

STACK CONTENTS



DDIV

32-BIT FIXED-POINT DIVIDE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	1	0	1	1	1	1

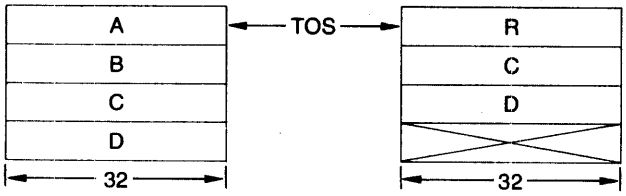
Hex Coding: AF with sr = 1
2F with sr = 0

Execution Time: 196 to 210 clock cycles when A ≠ 0
18 clock cycles when A = 0.

Description:
The 32-bit fixed-point two's complement integer operand B at NOS is divided by the 32-bit fixed-point two's complement integer operand A at the TOS. The 32-bit integer quotient R replaces B and the stack is moved up so that R occupies the TOS. No remainder is generated. Operands A and B are lost. Operands C and D are unchanged.
If A is zero, R is set equal to B and the divide-by-zero error status will be reported. If either A or B is the most negative value possible in the format, R will be meaningless and the overflow error status will be reported.

Status Affected: Sign, Zero, Error Field

STACK CONTENTS



DMUL

32-BIT FIXED-POINT MULTIPLY, LOWER

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	1	0	1	1	1	0

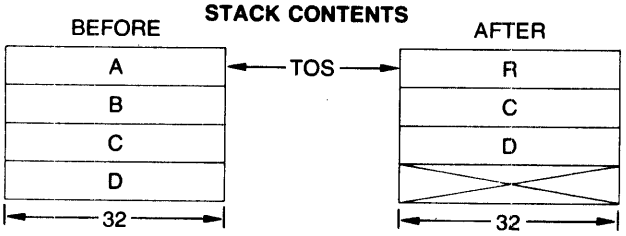
Hex Coding: AE with sr = 1
2E with sr = 0

Execution Time: 194 to 210 clock cycles

Description:
The 32-bit fixed-point two's complement integer operand A at the TOS is multiplied by the 32-bit fixed-point two's complement integer operand B at the NOS. The 32-bit least significant half of the product R replaces B and the stack is moved up so that R occupies the TOS. The most significant half of the product is lost. Operands A and B are lost. Operands C and D are unchanged.
The overflow status bit is set if the discarded upper half was non-zero. If either A or B is the most negative value that can be represented in the format, that value is returned as R and the overflow status is set.

Status Affected: Sign, Zero, Overflow

STACK CONTENTS



DMUU

32-BIT FIXED-POINT MULTIPLY, UPPER

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	1	1	0	1	1	0

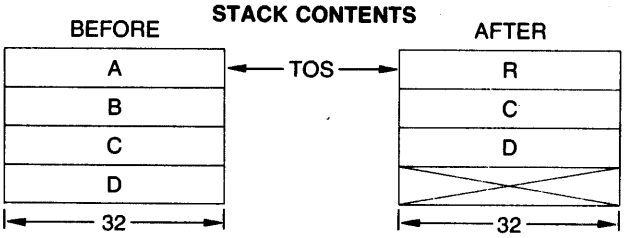
Hex Coding: B6 with sr = 1
36 with sr = 0

Execution Time: 182 to 218 clock cycles

Description:
The 32-bit fixed-point two's complement integer operand A at the TOS is multiplied by the 32-bit fixed-point two's complement integer operand B at the NOS. The 32-bit most significant half of the product R replaces B and the stack is moved up so that R occupies the TOS. The least significant half of the product is lost. Operands A and B are lost. Operands C and D are unchanged.
If A or B was the most negative value possible in the format, overflow status is set and R is meaningless.

Status Affected: Sign, Zero, Overflow

STACK CONTENTS



DSUB

32-BIT FIXED-POINT SUBTRACT

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	1	0	1	1	0	1

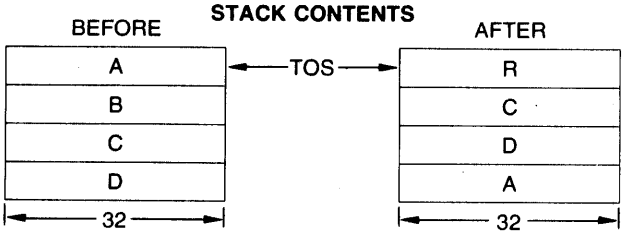
Hex Coding: AD with sr = 1
2D with sr = 0

Execution Time: 38 to 40 clock cycles

Description:
The 32-bit fixed-point two's complement operand A at the TOS is subtracted from the 32-bit fixed-point two's complement operand B at the NOS. The difference R replaces operand B and the stack is moved up so that R occupies the TOS. Operand B is lost. Operands A, C and D are unchanged.
If the subtraction generates a borrow it is reported in the carry status bit. If A is the most negative value that can be represented in the format the overflow status is set. If the result cannot be represented in the data format range, the overflow bit is set and the 32 least significant bits of the result are returned as R.

Status Affected: Sign, Zero, Carry, Overflow

STACK CONTENTS



EXP

32-BIT FLOATING-POINT e^X

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	1	0	1	0

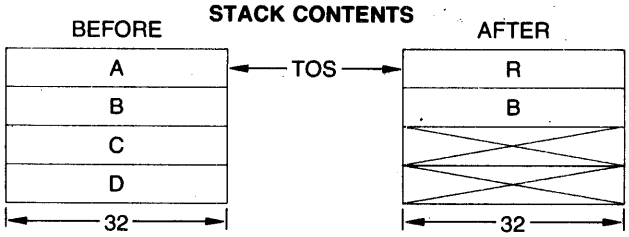
Hex Coding: 8A with sr = 1
0A with sr = 0

Execution Time: 3794 to 4878 clock cycles for $|A| \leq 1.0 \times 2^5$
34 clock cycles for $|A| > 1.0 \times 2^5$

Description:
The base of natural logarithms, e, is raised to an exponent value specified by the 32-bit floating-point operand A at the TOS. The result R of e^A replaces A. Operands A, C and D are lost. Operand B is unchanged.
EXP accepts all input data values within the range of $-1.0 \times 2^{+5}$ to $+1.0 \times 2^{+5}$. Input values outside this range will return a code of 1100 in the error field of the status register.

Accuracy: EXP exhibits a maximum relative error of 5.0×10^{-7} over the valid input data range.

Status Affected: Sign, Zero, Error Field



FADD

32-BIT FLOATING-POINT ADD

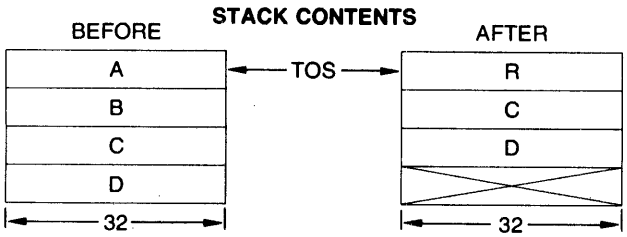
	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	0	0	0	0

Hex Coding: 90 with sr = 1
10 with sr = 0

Execution Time: 54 to 368 clock cycles for A ≠ 0
24 clock cycles for A = 0

Description:
32-bit floating-point operand A at the TOS is added to 32-bit floating-point operand B at the NOS. The result R replaces B and the stack is moved up so that R occupies the TOS. Operands A and B are lost. Operands C and D are unchanged.
Exponent alignment before the addition and normalization of the result accounts for the variation in execution time. Exponent overflow and underflow are reported in the status register, in which case the mantissa is correct and the exponent is offset by 128.

Status Affected: Sign, Zero, Error Field



FDIV

32-BIT FLOATING-POINT DIVIDE

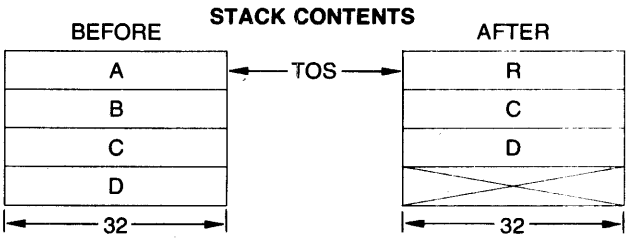
	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	0	0	1	1

Hex Coding: 93 with sr = 1
13 with sr = 0

Execution Time: 154 to 184 clock cycles for A ≠ 0
22 clock cycles for A = 0

Description:
32-bit floating-point operand B at NOS is divided by 32-bit floating-point operand A at the TOS. The result R replaces B and the stack is moved up so that R occupies the TOS. Operands A and B are lost. Operands C and D are unchanged.
If operand A is zero, R is set equal to B and the divide-by-zero error is reported in the status register. Exponent overflow or underflow is reported in the status register, in which case the mantissa portion of the result is correct and the exponent portion is offset by 128.

Status Affected: Sign, Zero, Error Field



FIXD

32-BIT FLOATING-POINT TO 32-BIT FIXED-POINT CONVERSION

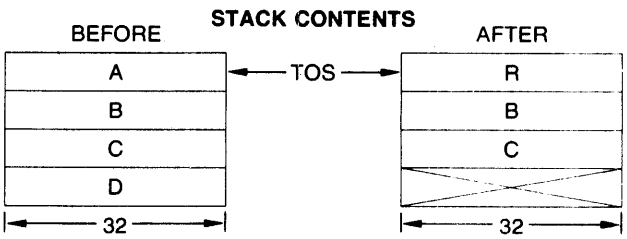
	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	1	1	1	0

Hex Coding: 9E with sr = 1
1E with sr = 0

Execution Time: 90 to 336 clock cycles

Description:
32-bit floating-point operand A at the TOS is converted to a 32-bit fixed-point two's complement integer. The result R replaces A. Operands A and D are lost. Operands B and C are unchanged.
If the integer portion of A is larger than 31 bits when converted, the overflow status will be set and A will not be changed. Operand D, however, will still be lost.

Status Affected: Sign, Zero Overflow



FIXS

32-BIT FLOATING-POINT TO 16-BIT FIXED-POINT CONVERSION

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	1	1	1	1

Hex Coding: 9F with sr = 1
1F with sr = 0

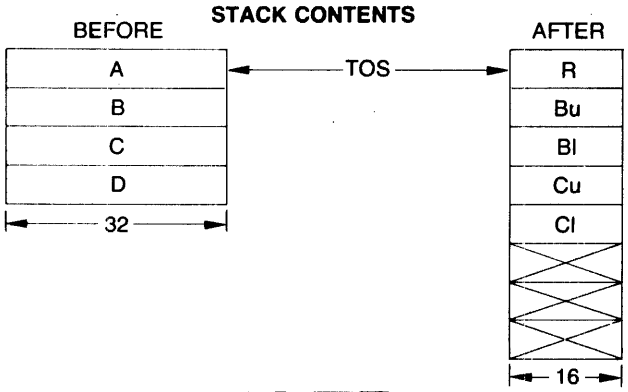
Execution Time: 90 to 214 clock cycles

Description:

32-bit floating-point operand A at the TOS is converted to a 16-bit fixed-point two's complement integer. The result R replaces the lower half of A and the stack is moved up by two bytes so that R occupies the TOS. Operands A and D are lost. Operands B and C are unchanged, but appear as upper (u) and lower (l) halves on the 16-bit wide stack if they are 32-bit operands.

If the integer portion of A is larger than 15 bits when converted, the overflow status will be set and A will not be changed. Operand D, however, will still be lost.

Status Affected: Sign, Zero, Overflow



FLTD

32-BIT FIXED-POINT TO 32-BIT FLOATING-POINT CONVERSION

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	1	1	0	0

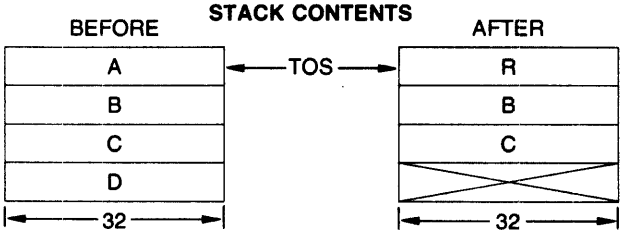
Hex Coding: 9C with sr = 1
1C with sr = 0

Execution Time: 56 to 342 clock cycles

Description:

32-bit fixed-point two's complement integer operand A at the TOS is converted to a 32-bit floating-point number. The result R replaces A at the TOS. Operands A and D are lost. Operands B and C are unchanged.

Status Affected: Sign, Zero



FLTS

16-BIT FIXED-POINT TO 32-BIT FLOATING-POINT CONVERSION

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	1	1	0	1

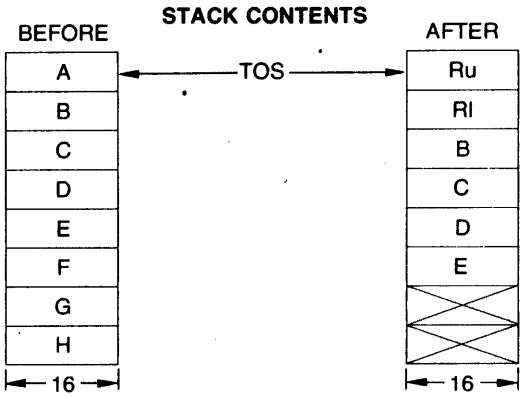
Hex Coding: 9D with sr = 1
1D with sr = 0

Execution Time: 62 to 156 clock cycles

Description:

16-bit fixed-point two's complement integer A at the TOS is converted to a 32-bit floating-point number. The lower half of the result R (Rl) replaces A, the upper half (Ru) replaces H and the stack is moved down so that Ru occupies the TOS. Operands A, F, G and H are lost. Operands B, C, D and E are unchanged.

Status Affected: Sign, Zero



FMUL

32-BIT FLOATING-POINT MULTIPLY

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	0	0	1	0

Hex Coding: 92 with sr = 1
12 with sr = 0

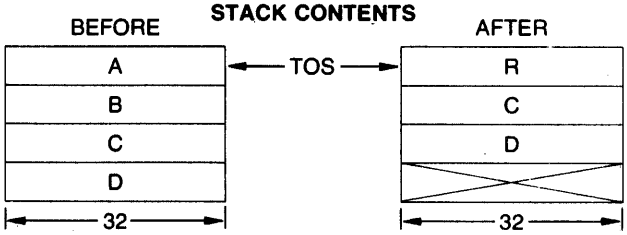
Execution Time: 146 to 168 clock cycles

Description:

32-bit floating-point operand A at the TOS is multiplied by the 32-bit floating-point operand B at the NOS. The normalized result R replaces B and the stack is moved up so that R occupies the TOS. Operands A and B are lost. Operands C and D are unchanged.

Exponent overflow or underflow is reported in the status register, in which case the mantissa portion of the result is correct and the exponent portion is offset by 128.

Status Affected: Sign, Zero, Error Field



FSUB

32-BIT FLOATING-POINT SUBTRACTION

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	0	0	0	1

Hex Coding: 91 with sr = 1
11 with sr = 0

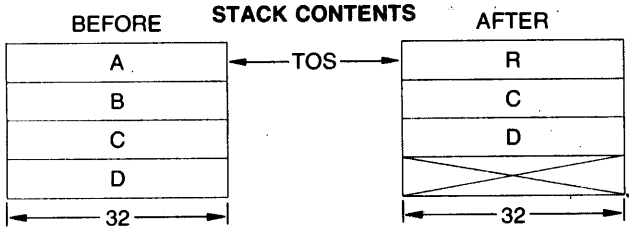
Execution Time: 70 to 370 clock cycles for A ≠ 0
26 clock cycles for A = 0

Description:
32-bit floating-point operand A at the TOS is subtracted from 32-bit floating-point operand B at the NOS. The normalized difference R replaces B and the stack is moved up so that R occupies the TOS. Operands A and B are lost. Operands C and D are unchanged.

Exponent alignment before the subtraction and normalization of the result account for the variation in execution time.

Exponent overflow or underflow is reported in the status register in which case the mantissa portion of the result is correct and the exponent portion is offset by 128.

Status Affected: Sign, Zero, Error Field (overflow)



LOG

32-BIT FLOATING-POINT COMMON LOGARITHM

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	1	0	0	0

Hex Coding: 88 with sr = 1
08 with sr = 0

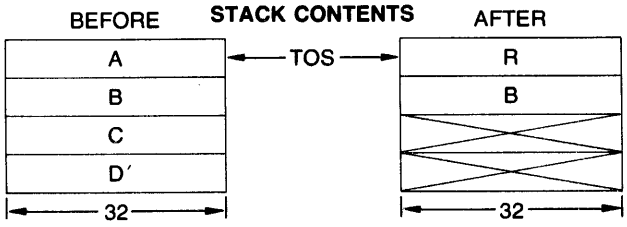
Execution Time: 4474 to 7132 clock cycles for A > 0
20 clock cycles for A ≤ 0

Description:
The 32-bit floating-point operand A at the TOS is replaced by R, the 32-bit floating-point common logarithm (base 10) of A. Operands A, C and D are lost. Operand B is unchanged.

The LOG function accepts any positive input data value that can be represented by the data format. If LOG of a non-positive value is attempted an error status of 0100 is returned.

Accuracy: LOG exhibits a maximum absolute error of 2.0×10^{-7} for the input range from 0.1 to 10, and a maximum relative error of 2.0×10^{-7} for positive values less than 0.1 or greater than 10.

Status Affected: Sign, Zero, Error Field



LN

32-BIT FLOATING-POINT NATURAL LOGARITHM

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	1	0	0	1

Hex Coding: 89 with sr = 1
09 with sr = 0

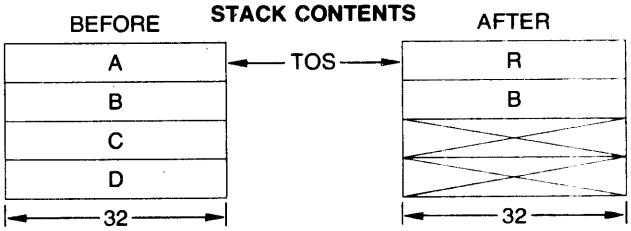
Execution Time: 4298 to 6956 clock cycles for A > 0
20 clock cycles for A ≤ 0

Description:
The 32-bit floating-point operand A at the TOS is replaced by R, the 32-bit floating-point natural logarithm (base e) of A. Operands A, C and D are lost. Operand B is unchanged.

The LN function accepts all positive input data values that can be represented by the data format. If LN of a non-positive number is attempted an error status of 0100 is returned.

Accuracy: LN exhibits a maximum absolute error of 2×10^{-7} for the input range from e^{-1} to e, and a maximum relative error of 2.0×10^{-7} for positive values less than e^{-1} or greater than e.

Status Affected: Sign, Zero, Error Field



NOP

NO OPERATION

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	0	0	0	0

Hex Coding: 80 with sr = 1
00 with sr = 0

Execution Time: 4 clock cycles

Description:
The NOP command performs no internal data manipulations. It may be used to set or clear the service request interface line without changing the contents of the stack.

Status Affected: The status byte is cleared to all zeroes.

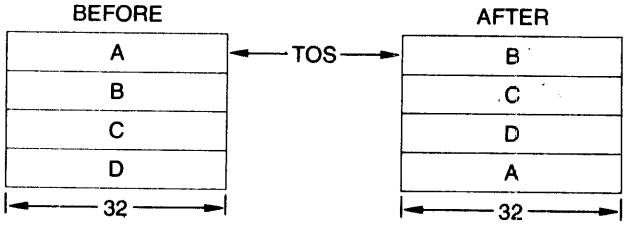
POPD

32-BIT
STACK POP

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	1	1	1	0	0	0

Hex Coding: B8 with sr = 1
38 with sr = 0
Execution Time: 12 clock cycles
Description:
The 32-bit stack is moved up so that the old NOS becomes the new TOS. The previous TOS rotates to the bottom of the stack. All operand values are unchanged. POPD and POPF execute the same operation.
Status Affected: Sign, Zero

STACK CONTENTS



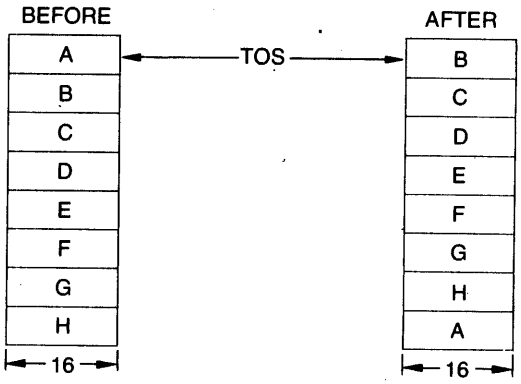
POPS

16-BIT
STACK POP

	7	6	5	4	3	2	1	0
Binary Coding:	sr	1	1	1	1	0	0	0

Hex Coding: F8 with sr = 1
78 with sr = 0
Execution Time: 10 clock cycles
Description:
The 16-bit stack is moved up so that the old NOS becomes the new TOS. The previous TOS rotates to the bottom of the stack. All operand values are unchanged.
Status Affected: Sign, Zero

STACK CONTENTS



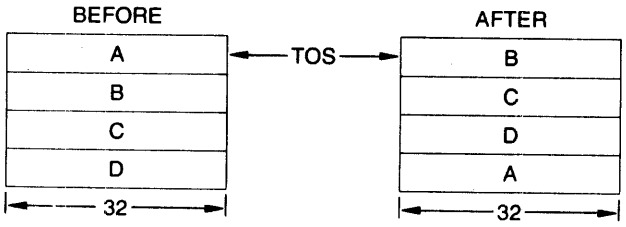
POPF

32-BIT
STACK POP

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	1	0	0	0

Hex Coding: 98 with sr = 1
18 with sr = 0
Execution Time: 12 clock cycles
Description:
The 32-bit stack is moved up so that the old NOS becomes the new TOS. The old TOS rotates to the bottom of the stack. All operand values are unchanged. POPF and POPD execute the same operation.
Status Affected: Sign, Zero

STACK CONTENTS



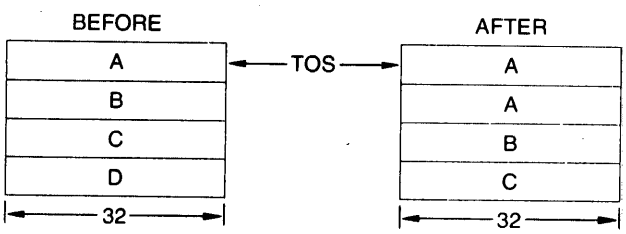
PTOD

PUSH 32-BIT
TOS ONTO STACK

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	1	1	0	1	1	1

Hex Coding: B7 with sr = 1
37 with sr = 0
Execution Time: 20 clock cycles
Description:
The 32-bit stack is moved down and the previous TOS is copied into the new TOS location. Operand D is lost. All other operand values are unchanged. PTOD and PTOF execute the same operation.
Status Affected: Sign, Zero

STACK CONTENTS



PTOF

PUSH 32-BIT TOS ONTO STACK

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	0	1	1	1

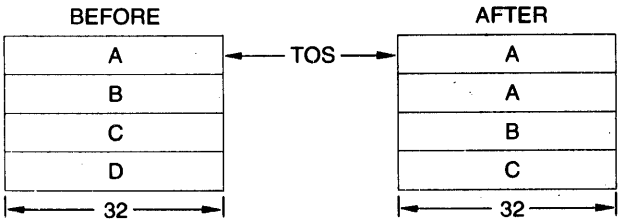
Hex Coding: 97 with sr = 1
17 with sr = 0

Execution Time: 20 clock cycles

Description:
The 32-bit stack is moved down and the previous TOS is copied into the new TOS location. Operand D is lost. All other operand values are unchanged. PTOF and PTOD execute the same operation.

Status Affected: Sign, Zero

STACK CONTENTS



PTOS

PUSH 16-BIT TOS ONTO STACK

	7	6	5	4	3	2	1	0
Binary Coding:	sr	1	1	1	0	1	1	1

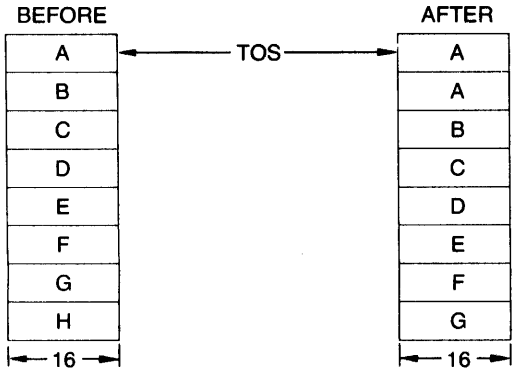
Hex Coding: F7 with sr = 1
77 with sr = 0

Execution Time: 16 clock cycles

Description:
The 16-bit stack is moved down and the previous TOS is copied into the new TOS location. Operand H is lost and all other operand values are unchanged.

Status Affected: Sign, Zero

STACK CONTENTS



PUPI

PUSH 32-BIT FLOATING-POINT π

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	1	0	1	0

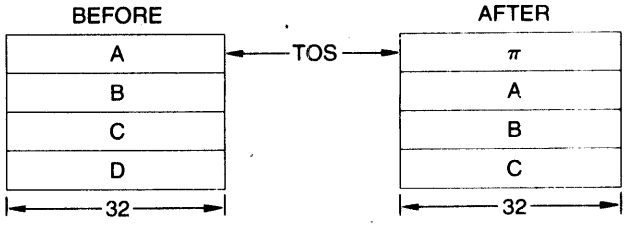
Hex Coding: 9A with sr = 1
1A with sr = 0

Execution Time: 16 clock cycles

Description:
The 32-bit stack is moved down so that the previous TOS occupies the new NOS location. 32-bit floating-point constant π is entered into the new TOS location. Operand D is lost. Operands A, B and C are unchanged.

Status Affected: Sign, Zero

STACK CONTENTS



PWR

32-BIT FLOATING-POINT X^Y

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	1	0	1	1

Hex Coding: 8B with sr = 1
0B with sr = 0

Execution Time: 8290 to 12032 clock cycles

Description:

32-bit floating-point operand B at the NOS is raised to the power specified by the 32-bit floating-point operand A at the TOS. The result R of B^A replaces B and the stack is moved up so that R occupies the TOS. Operands A, B, and D are lost. Operand C is unchanged.

The PWR function accepts all input data values that can be represented in the data format for operand A and all positive values for operand B. If operand B is non-positive an error status of 0100 will be returned. The EXP and LN functions are used to implement PWR using the relationship $B^A = EXP\ [A(LN\ B)]$. Thus if the term $[A(LN\ B)]$ is outside the range of $-1.0 \times 2^{+5}$ to $+1.0 \times 2^{+5}$ an error status of 1100 will be returned. Underflow and overflow conditions can occur.

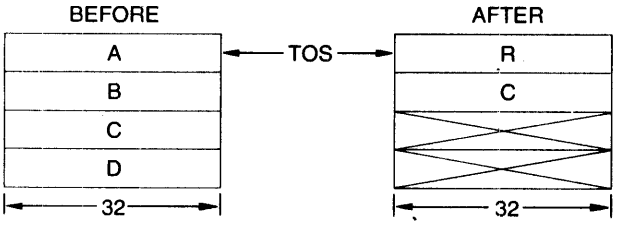
Accuracy: The error performance for PWR is a function of the LN and EXP performance as expressed by:

$|(\text{Relative Error})_{PWR}| = |(\text{Relative Error})_{EXP} + |A(\text{Absolute Error})_{LN}|$

The maximum relative error for PWR occurs when A is at its maximum value while $[A(LN\ B)]$ is near 1.0×2^5 and the EXP error is also at its maximum. For most practical applications the relative error for PWR will be less than 7.0×10^{-7}.

Status Affected: Sign, Zero, Error Field

STACK CONTENTS



SADD

16-BIT FIXED-POINT ADD

	7	6	5	4	3	2	1	0
Binary Coding:	sr	1	1	0	1	1	0	0

Hex Coding: EC with sr = 1
6C with sr = 0

Execution Time: 16 to 18 clock cycles

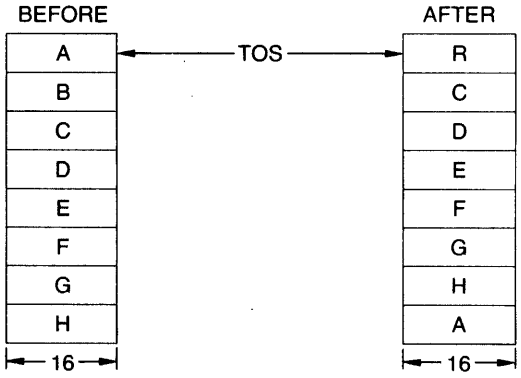
Description:

16-bit fixed-point two's complement integer operand A at the TOS is added to 16-bit fixed-point two's complement integer operand B at the NOS. The result R replaces B and the stack is moved up so that R occupies the TOS. Operand B is lost. All other operands are unchanged.

If the addition generates a carry bit it is reported in the status register. If an overflow occurs it is reported in the status register and the 16 least significant bits of the result are returned.

Status Affected: Sign, Zero, Carry, Error Field

STACK CONTENTS



SDIV

16-BIT FIXED-POINT DIVIDE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	1	1	0	1	1	1	1

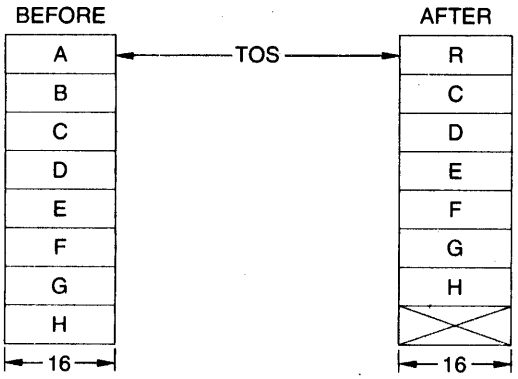
Hex Coding: EF with sr = 1
6F with sr = 0

Execution Time: 84 to 94 clock cycles for A ≠ 0
14 clock cycles for A = 0

Description:
16-bit fixed-point two's complement integer operand B at the NOS is divided by 16-bit fixed-point two's complement integer operand A at the TOS. The 16-bit integer quotient R replaces B and the stack is moved up so that R occupies the TOS. No remainder is generated. Operands A and B are lost. All other operands are unchanged.
If A is zero, R will be set equal to B and the divide-by-zero error status will be reported.

Status Affected: Sign, Zero, Error Field

STACK CONTENTS



SIN

32-BIT FLOATING-POINT SINE

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	0	0	1	0

Hex Coding: 82 with sr = 1
02 with sr = 0

Execution Time: 3796 to 4808 clock cycles for $|A| > 2^{-12}$ radians
30 clock cycles for $|A| \leq 2^{-12}$ radians

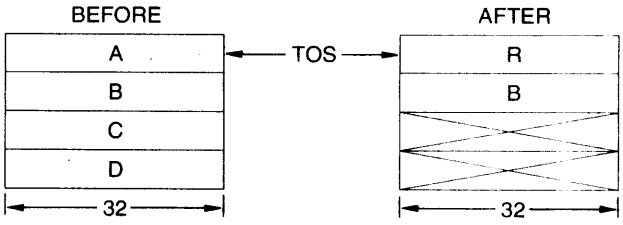
Description:
The 32-bit floating-point operand A at the TOS is replaced by R, the 32-bit floating-point sine of A. A is assumed to be in radians. Operands A, C and D are lost. Operand B is unchanged.

The SIN function will accept any input data value that can be represented by the data format. All input values are range reduced to fall within the interval $-\pi/2$ to $+\pi/2$ radians.

Accuracy: SIN exhibits a maximum relative error of 5.0×10^{-7} for input values in the range of -2π to $+2\pi$ radians.

Status Affected: Sign, Zero

STACK CONTENTS



SMUL

16-BIT FIXED-POINT MULTIPLY, LOWER

	7	6	5	4	3	2	1	0
Binary Coding:	sr	1	1	0	1	1	1	0

Hex Coding: EE with sr = 1
6E with sr = 0

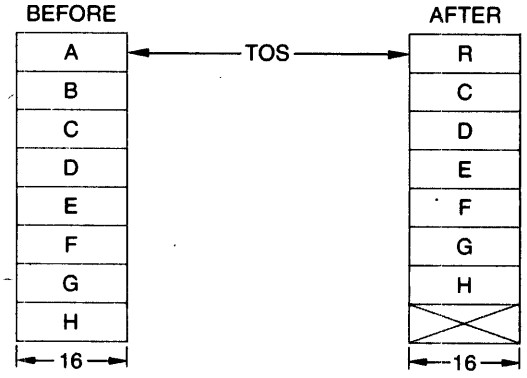
Execution Time: 84 to 94 clock cycles

Description:

16-bit fixed-point two's complement integer operand A at the TOS is multiplied by the 16-bit fixed-point two's complement integer operand B at the NOS. The 16-bit least significant half of the product R replaces B and the stack is moved up so that R occupies the TOS. The most significant half of the product is lost. Operands A and B are lost. All other operands are unchanged. The overflow status bit is set if the discarded upper half was non-zero. If either A or B is the most negative value that can be represented in the format, that value is returned as R and the overflow status is set.

Status Affected: Sign, Zero, Error Field

STACK CONTENTS



SMUU

16-BIT FIXED-POINT MULTIPLY, UPPER

	7	6	5	4	3	2	1	0
Binary Coding:	sr	1	1	1	0	1	1	0

Hex Coding: F6 with sr = 1
76 with sr = 0

Execution Time: 80 to 98 clock cycles

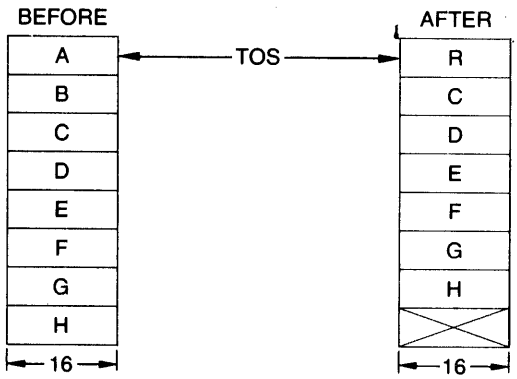
Description:

16-bit fixed-point two's complement integer operand A at the TOS is multiplied by the 16-bit fixed-point two's complement integer operand B at the NOS. The 16-bit most significant half of the product R replaces B and the stack is moved up so that R occupies the TOS. The least significant half of the product is lost. Operands A and B are lost. All other operands are unchanged.

If either A or B is the most negative value that can be represented in the format, that value is returned as R and the overflow status is set.

Status Affected: Sign, Zero, Error Field

STACK CONTENTS



SQRT

32-BIT FLOATING-POINT SQUARE ROOT

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	0	0	0	1

Hex Coding: 81 with sr = 1
01 with sr = 0

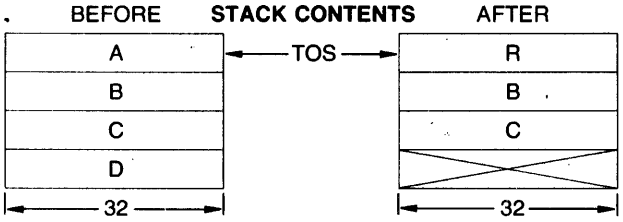
Execution Time: 782 to 870 clock cycles

Description:

32-bit floating-point operand A at the TOS is replaced by R, the 32-bit floating-point square root of A. Operands A and D are lost. Operands B and C are not changed.

SQRT will accept any non-negative input data value that can be represented by the data format. If A is negative an error code of 0100 will be returned in the status register.

Status Affected: Sign, Zero, Error Field



SSUB

16-BIT FIXED-POINT SUBTRACT

	7	6	5	4	3	2	1	0
Binary Coding:	sr	1	1	0	1	1	0	1

Hex Coding: ED with sr = 1
6D with sr = 0

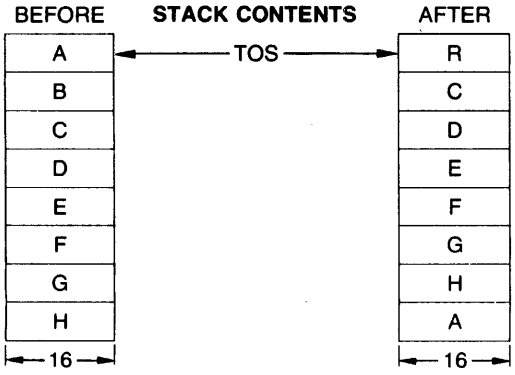
Execution Time: 30 to 32 clock cycles

Description:

16-bit fixed-point two's complement integer operand A at the TOS is subtracted from 16-bit fixed-point two's complement integer operand B at the NOS. The result R replaces B and the stack is moved up so that R occupies the TOS. Operand B is lost. All other operands are unchanged.

If the subtraction generates a borrow it is reported in the carry status bit. If A is the most negative value that can be represented in the format the overflow status is set. If the result cannot be represented in the format range, the overflow status is set and the 16 least significant bits of the result are returned as R.

Status Affected: Sign, Zero, Carry, Error Field



TAN

32-BIT FLOATING-POINT TANGENT

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	0	0	1	0	0

Hex Coding: 84 with sr = 1
04 with sr = 0

Execution Time: 4894 to 5886 clock cycles for $|A| \geq 2^{12}$ radians
30 clock cycles for $|A| \leq 2^{-12}$ radians

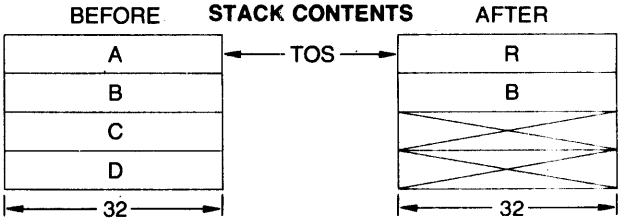
Description:

The 32-bit floating-point operand A at the TOS is replaced by the 32-bit floating-point tangent of A. Operand A is assumed to be in radians. A, C and D are lost. B is unchanged.

The TAN function will accept any input data value that can be represented in the data format. All input data values are range-reduced to fall within $-\pi/4$ to $+\pi/4$ radians. TAN is unbounded for input values near odd multiples of $\pi/2$ and in such cases the overflow bit is set in the status register. For angles smaller than 2^{-12} radians, TAN returns A as the tangent of A.

Accuracy: TAN exhibits a maximum relative error of 5.0×10^{-7} for input data values in the range of -2π to $+2\pi$ radians except for data values near odd multiples of $\pi/2$.

Status Affected: Sign, Zero, Error Field (overflow)



XCHD

EXCHANGE 32-BIT STACK OPERANDS

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	1	1	1	0	0	1

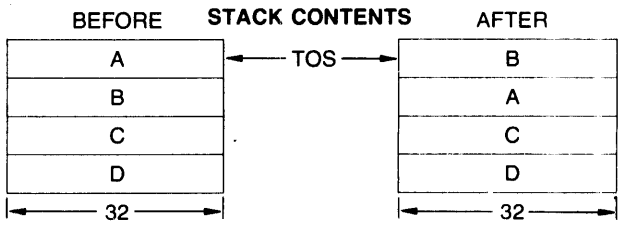
Hex Coding: B9 with sr = 1
39 with sr = 0

Execution Time: 26 clock cycles

Description:

32-bit operand A at the TOS and 32-bit operand B at the NOS are exchanged. After execution, B is at the TOS and A is at the NOS. All operands are unchanged. XCHD and XCHF execute the same operation.

Status Affected: Sign, Zero



XCHF

EXCHANGE 32-BIT STACK OPERANDS

	7	6	5	4	3	2	1	0
Binary Coding:	sr	0	0	1	1	0	0	1

Hex Coding: 99 with sr = 1
19 with sr = 0

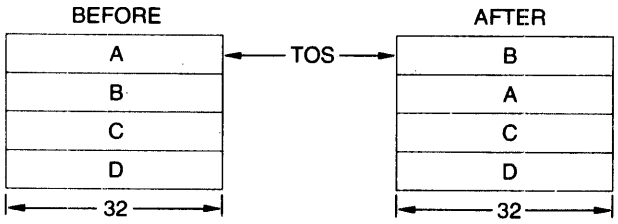
Execution Time: 26 clock cycles

Description:

32-bit operand A at the TOS and 32-bit operand B at the NOS are exchanged. After execution, B is at the TOS and A is at the NOS. All operands are unchanged. XCHD and XCHF execute the same operation.

Status Affected: Sign, Zero

STACK CONTENTS



XCHS

EXCHANGE 16-BIT STACK OPERANDS

	7	6	5	4	3	2	1	0
Binary Coding:	sr	1	1	1	1	0	0	1

Hex Coding: F9 with sr = 1
79 with sr = 0

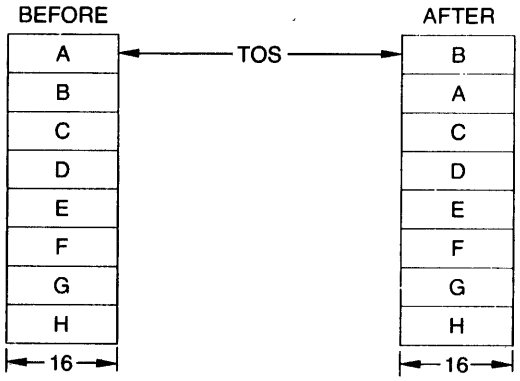
Execution Time: 18 clock cycles

Description:

16-bit operand A at the TOS and 16-bit operand B at the NOS are exchanged. After execution, B is at the TOS and A is at the NOS. All operand values are unchanged.

Status Affected: Sign, Zero

STACK CONTENTS



CHAPTER 4
THEORY OF OPERATION

4-1. INTRODUCTION

This chapter provides a detailed functional description of the AMC 95/4005 MonoBoard Computer. The MonoBoard Computer consists of the functional blocks shown in figure 4-1.

Both active-high (positive true) and active-low (negative true) signals appear on the schematics. To eliminate confusion, and simplify presentation, the following convention is adhered to within this manual: whenever a signal is active-low (negative true), its mnemonic is followed by an asterisk (*). For example, MEMR* denotes an active-low signal. When a signal is active-high, the asterisk is omitted.

4-2. CENTRAL PROCESSING UNIT (CPU) GROUP

The CPU group consists of an Am9080A CPU, an Am8224 Clock Generator, an Am8228 High Speed System Controller and associated control ciruits.

The Am8224 Clock Generator (U34) and an 18MHz crystal provide the primary timing reference for board operation. The 18MHz crystal establishes the frequency of oscillation for the Am8224 via a 33pF capacitor. The Am8224 divides the crystal frequency by nine to produce the 2MHz phase one and phase two timing inputs to the CPU. A TTL level phase two TTL signal is also derived and made available for TTL gating. All processing activities are referenced to the phase one and phase two clocks.

The 3MHz MonoBoard contains a 27MHz crystal which is divided by nine to produce the 3MHz phase one and phase two timing inputs for the CPU.

A power up reset circuit provides a reset to the Am8224 upon power turn-on. The Am8224 in turn provides a reset (RST) output to the CPU and other programmable devices on the board. The reset output also generates the INIT* signal that is used to initialize other boards.

A Ready In (RDYIN) input to the Am8224 is gated from the on-board and off-board ready and acknowledge outputs. When an addressed device responds with its ready or acknowledge signal, it is gated to the RDYIN input of the Am8224. The RDYIN input provides the D input for an internal flip-flop. After RDYIN goes high, the leading edge of the next phase two clock pulse clocks the flip-flop to produce a high READY signal to the CPU. A failsafe timer circuit produces a TIME OUT* signal that is also gated to the Am8224 RDYIN input. The failsafe timer is designed to prevent hanging up the CPU if memory or a device does not respond within approximately 25ms.

NOTE

It is recommended that the failsafe timer be used to generate a timeout interrupt to inform the program that the operation has not completed. Otherwise, invalid data could cause the program to go to unknown locations.

The failsafe timer signal goes high when a one-shot is triggered by the Status Strobe signal (STSTB*) from the Am8224 at the beginning of each machine cycle. If the one-shot is not retriggered within approximately 25ms, its output goes low and is gated through to the RDYIN pin on the Am8224, thereby driving the READY line high and allowing the CPU to exit the wait state.

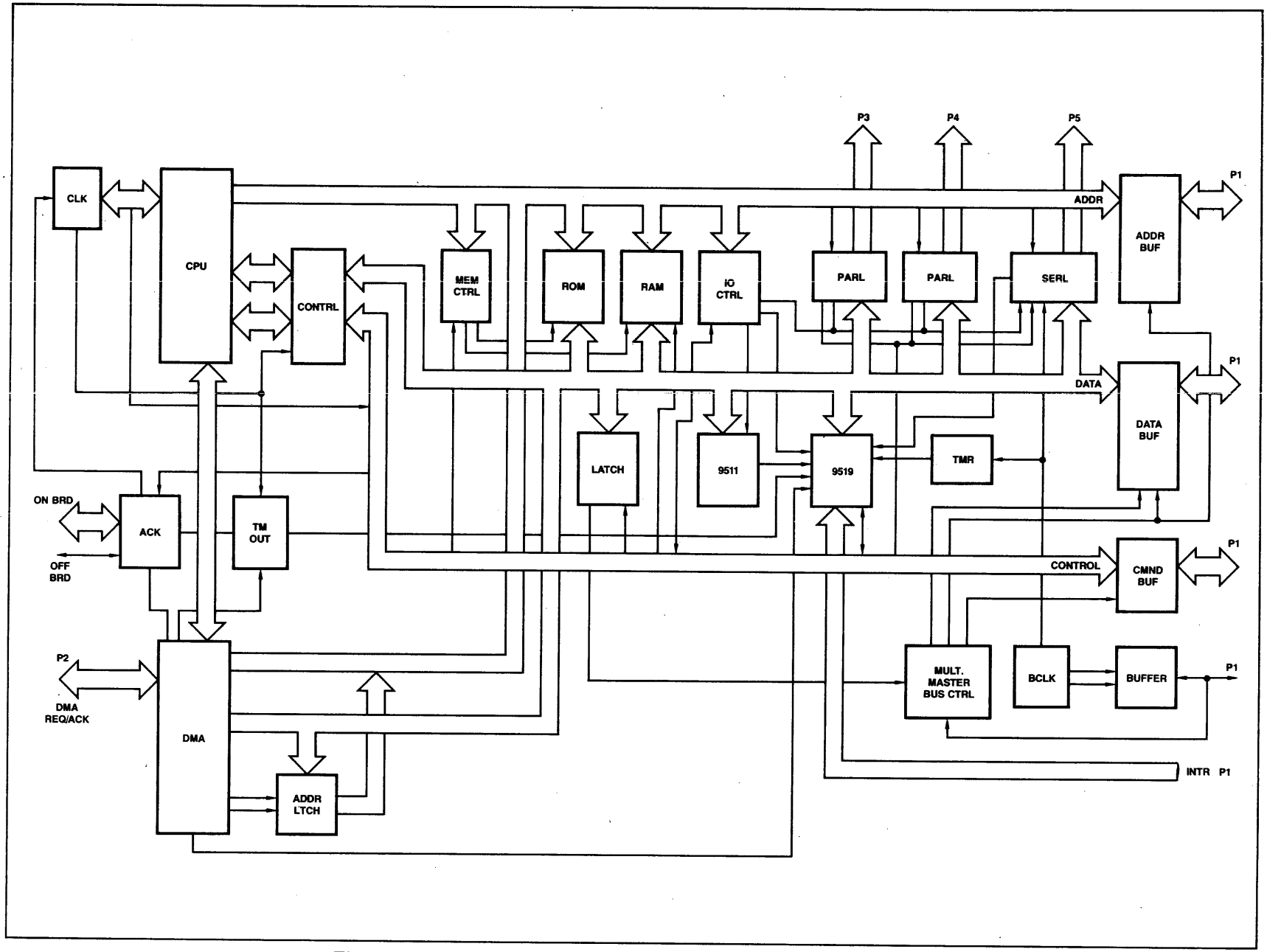


Figure 4-1. AMC 95/4005 MonoBoard Computer Block Diagram.

Three groups of signals are provided by the Am9080A CPU to transmit data and controls between the CPU, memory, and peripheral devices. An 8-bit parallel data bus transmits instructions, data, and status to and from the CPU. A 16-bit address bus identifies any of 65,536 (64K) bytes of memory location during a memory access operation. The address bus also addresses peripheral devices during I/O read or write instructions. The third group of signals includes control signals to synchronize CPU operations with memory and peripheral devices.

Within the CPU, an instruction cycle is defined as the time required to fetch and execute an instruction. During the fetch, a selected instruction is extracted from memory and stored in the CPU operating registers. During the execution part, the instruction is decoded and translated into specific processing activities.

The instruction cycle consists of from one to five machine cycles. A machine cycle is required each time the CPU accesses memory or an I/O port. The fetch portion of an instruction cycle requires one machine cycle for each byte to be fetched. The number of machine cycles required for instruction execution depends on the kind of instruction that has been fetched. Some instructions do not require additional machine cycles for execution; others require additional cycles to write or read data to/from memory or I/O devices. An instruction could require from one to five machine cycles.

Each machine cycle consists of from three to five clock pulses. The clock pulses are identified as states (T1 through T5). A state is identified as the interval between two succesive positive-going transitions of the phase one clock.

In summary, a clock period defines a state. Three to five states constitute a machine cycle, and one to five machine cycles comprise an instruction cycle.

There are three exceptions to the defined duration of a state. These are the Wait state, the Hold state, and the Halt state. Because the duration of these states depends upon external events, they are of indeterminate length. These states must be synchronized with the phase one clock pulses. Their duration is therefore an integral multiple of the phase one clock.

At the beginning of each machine cycle (in state T1) the CPU activates its SYNC output and issues status information on its data bus. The Am8224 accepts SYNC and generates an active-low status strobe (STSTB*) signal to the Am8238 and the failsafe timer. The status information indicates the type of machine cycle in progress.

The rising edge of the phase two clock during T1 loads the address lines of the processor. These lines become stable during the phase two clock and remain stable until the first phase two pulse after state T3. This gives the CPU time to read the data returned from memory. Once the processor has sent an address to memory, the processor samples the READY input. If the READY input is low, the processor will idle. This gives the memory time to respond to the addressed data request. The CPU remains in the idle condition until its READY line again goes high. When the READY line goes high, the cycle proceeds, beginning with the rising edge of the next phase one clock.

The events that occur during the T3 through T5 states are determined by the machine cycle in progress. During a fetch memory cycle the CPU interprets the data on the data bus as an instruction. During a memory read the CPU interprets the data as a data word. The CPU outputs data on the data bus during a memory write. During an I/O operation the CPU can either receive or

transmit data, depending upon whether the machine cycle is an input or output operation.

An Am8238 eight bit, bidirectional bus driver buffers the data bus from memory and devices. At the beginning of each machine cycle the CPU issues status on the data bus that indicates the type of function that will occur during the cycle. The Am8238 stores this information in a status latch when the STSTB input from the Am8224 goes low. The outputs of the status latch are connected to a gating array which is used for control signal generation. The gating array generates control signals MEMW*, MEMR*, IOW*, IOR*, and INTA* by gating the status latch with the DBIN, WR*, and HLDA signals from the CPU.

4-3. MULTI-MASTER BUS CONTROL LOGIC

Multi-Master bus operation is made possible by the bus control logic shown on sheet 6 of the schematics. The AMC 95/4005 MonoBoard Computer can be configured for either serial or parallel bus priority resolution. As shipped, a jumper is connected between jumper pins 84 and 85 to configure the board for serial priority operation. Therefore, the BPRO* signal is low when no higher priority master (including this master) is using the bus.

When an off-board memory or I/O operation is to take place, the XFER REQ signal is generated by the PROM at U67 to initiate a Bus Request. The flip-flop at U55 pin 2 is set, which in turn generates the common Bus Request (BREQ*) signals. The BREQ* signal is used only in parallel priority resolution application to indicate a request for control of the bus.

When no higher priority master is requesting the bus (BPRN* high) and no master has control of the bus (BUSY* high), the flip-flop at U55 pin 14 is set, which causes the CBRQ* signal to go high. One B clock cycle later, the flip-flop at U57 pin 2 is set to turn on the bus command drivers. Jumper pins 120, 121, and 122 are used to select the conditions under which this master will release the bus. Depending on the connection of the jumpers, the MonoBoard will either retain control of the bus until another master requests the bus (by pulling CBRQ* low) or relinquish control of the bus after each bus transfer.

When this master is in control of the bus, the OVERRIDE signal from the programmable latch going high causes the bus to be held busy. All bus requests are ignored and this master retains control of the bus until the override bit is cleared.

4-4. ON-BOARD MEMORY

MonoBoard memory addressing is controlled by a mapping PROM (U39) and the 74LS139 decoder at U40. Address bits 0 through 9 specify a single storage location within each memory device. The decoder output determines which memory chips are enabled.

Inputs to the memory select PROM can be program and jumper selected, as described in chapters 2 and 3. A customized PROM might also be inserted to satisfy unique application requirements. PROM output bit 3 is used to enable the decoder to select a pair of RAM memory chips. PROM output bit 2 is used to enable the ROM memory chip select decoder. When either the RAM or ROM is selected, the resultant RAMSEL* or ROMSEL* is used to generate the MEMSEL* to indicate that an on-board memory read or write operation is in progress.

When this master is already in control of the bus, data and addresses are placed on the bus during an on-board or off-board device selection. When the MEMSEL* signal is not present during a memory read operation or the IORDY* signal is not present during an I/O read operation, an off-board memory read or off-board I/O read is taking place; the DBOUT* signal causes the bus

drivers to read into the board. During a read or write to an on-board device, bus master logic determines whether data and address information is placed onto the bus, but read and write control signals are not placed on the bus.

When this master is not already in control of the bus during an on-board memory reference or on-board I/O operation, the Am8226 address and data bus drivers are held in the high impedance state by the high BUSEN* signal. The control drivers are placed in the high impedence state by the CMEN* signal. Therefore, although the direction control for the data bus drivers is out, the data bus driver/receivers are disabled during an on-board memory reference.

The memory acknowledge (MEMACK*) signal is input to the Am8224 control circuit, which in turn generates the READY signal to the Am9080 CPU to indicate that the memory access is taking place. One wait state is inserted in the on-board memory acknowledge timing by the circuitry at U47 when the board is configured for 3MHz operation.

4-5. READ ONLY MEMORY (ROM/E-PROM)

The Read Only Memory (ROM/E-PROM) section provides for installing up to 16K bytes of ROM using four Am9208, Am9218, or Am9233 compatable ROMs or four Am9708, Am9716, Am9732, or 2758 compatable E-PROMs. Jumpers are used to customize the memory socket voltages to the ROM/E-PROM being used as shown in table 2-7.

Memory addressing is controlled by a mapping PROM at U39 and the 74LS139 decoder at U40. Address bits 0 through 9 specify a single storage location within each of the four ROM/E-PROM devices; the decoder output selects one of the four chip locations.

4-6. RANDOM ACCESS MEMORY (RAM)

The Random Access Memory (RAM) section provides the user with 4096 (4K) by 8-bits of read/write storage. The RAM memory consists of eight Am9114 static RAM chips which store 1024 by 4-bits each. Address decoding for chip selection is accomplished by using an Am25LS139 one-of-four decoder and assorted control gates.

Each Am9114 chip has 10 address inputs (0 through 9) and four common data input/output pins I/01 through I/04, as well as active low chip select CS* and write enable WE* inputs. The data input/output pins I/01 through I/04 from two Am9114 chips are connected to the data bus as bits DB0 through DB7.

When on-board RAM is accessed, the ten least significant bits (0 through 9) of the address bus specify the 4-bit segment to be accessed on the selected RAMs. Address bits 10 and 11 specify which pair of RAMs are to be selected. Access to the RAMs is controlled by bit 3 from the memory select PROM at U39. RAM read/write control is provided by the ADVMW* signal.

4-7. I/O ADDRESS DECODING

The MBC includes six programmable devices and one 8-bit latch that are accessed by I/O commands.

Two Am9555 Programmable Peripheral Interfaces

An Am9551 Programmable Communications Interface

An Am9519 Universal Interrupt Controller

An Am9517 Multimode Direct Memory Access Controller

An Am9511 Arithmetic Processing Unit

The I/O Address Decode logic consists of 256 by 4-bit PROM and an Am25LS138 one-of-eight decoder. The PROM is programmed with the bit pattern required by the Am25LS138 to select one of eight active low outputs. Seven of these outputs select the six programmable devices and the latch. One output is not used and is never selected. Table D-2 in appendix D shows the bit pattern to memory location relationship for the I/O addresses.

Output bit 3 of the PROM is gated to the RDYIN pin of the Am8224 chip at U32 to provide an input ready acknowledge signal to the CPU. During a DMA operation, the PROM is disabled; however, another signal, IORW (U50 pin 3), is generated by the IOR* and ADVIOW* signals to provide (in conjunction with the IORDY* or EXTIOACK* signals) the Ready input (U28 pin 8) to the DMA controller.

4-8. SERIAL I/O INTERFACE

The Serial I/O Interface provides the MonoBoard Computer with a serial data communications channel that can be programmed to operate with most of the current serial data transmission protocols: synchronous or asynchronous. Character length, number of stop bits, and even/odd parity are program selectable. Baud rate is switch selectable. The serial I/O Interface consists primarily of an Am9551 Programmable Communications Interface, a 4702 Baud Rate Generator, and driver receiver circuits.

The least significant address bit is applied to the C/$\overline{D}$ input of the Am9551. An output instruction (IOW* true, CS51* true, and C/$\overline{D}$ high) causes the Am9551 to accept a control byte through its data bus pins. The control byte can be either a mode instruction or a command instruction, depending on the sequence in which it is sent. The mode instruction specifies the baud rate multiplier, character length, parity, and the number of stop bits. The actual baud rate selection is determined by the baud rate input from the 4702. The command word instructs the Am9551 to enable/disable the receiver and transmitter, to reset errors, to return to Idle mode, and to set/clear the Data Terminal Ready Signal output. An output instruction also causes the Am9551 to accept a data byte through its data bus pins. Bit 0 is the least significant bit and bit 7 is the most significant bit. The Am9551 will subsequently transmit serial data to an external device if the transmitter is enabled.

An input instruction (IOR* true, CS51* true and C/$\overline{D}$ high) causes the Am9551 to place a status byte on the data bus. The status bits are the result of status and error checking functions performed within the Am9551. An input instruction also causes the Am9551 to output a data byte onto the data bus. Bit 0 is the least significant bit and bit 7 is the most significant bit.

Timing for the Am9551 internal functions is provided by the CO pin of the 4702 baud rate generator. The baud rate generator uses a derived 2.4576MHz crystal controlled frequency as the basic timing reference. A four position DIP switch is used to selct baud rates. A second output pin (Z) on the baud rate generator provides an output that can be jumpered to the transmit clock (TXC) and receive clock (RXC). The TXC and RXC signals can also be supplied externally.

A high on the Am9551 reset (RST) line forces the Am9551 into an idle mode. After a high RST input, the device remains idle until a new set of control words are written into the Am9551 to define its function.

In addition to the above control lines, the Am9551 also has a set of control inputs and outputs that can be used to simplify the interface to almost any

serial data device. These control signals are general purpose in nature. The following paragraphs describe the interface controls.

Receiver Data (RxD) Serial data is received from the communication line of the RxD input.

Receiver Clock (RxC*) The serial data on input RxD is clocked into the Am9551 by the RxC* clock signal. In the synchronous mode, RxC* is determined by the baud rate and supplied by the modem. In the asychronous mode, RxC* is 1, 16, or 64 times the baud rate selected by the mode control instruction. Data is sampled by the Am9551 on the rising edge of RxC*.

Receiver Ready (RxRDY) The RxRDY output signal indicates to the processor that data has been shifted from the receiver section into the receiver buffer and can be read. The active high RxRDY signal is reset when the buffer is read by the processor. RxRDY can be activated only if the receiver enable (RxE) has been set in the command register, even though the receiver might be running. If the processor does not read the receiver buffer before the next character is shifted from the receiver section, an overrun error is indicated in the status buffer.

Sync Detect (SYNDET) This signal is used only in the synchronous mode. It can be either an output or input depending on whether the program is set for internal or external synchronization. As an output, a high level indicates when the sync character has been detected in the received data stream after the internal synchronization mode has been programmed. If the Am9551 is programmed to utilize two sync characters, SYNDET will go to a high level when the last bit of the second sync character is received. SYNDET is reset when the status buffer is read or when a reset signal is activated. SYNDET will perform as an input when the external synchronization mode is programmed. External logic can supply a positive-going signal to indicate to the Am9551 that synchronization has been achieved. This will cause it to initiate the assembly of characters on the next falling edge of RxC*. To successfully achieve synchronization, the SYNDET signal should be maintained in a high condition for at least one full RxC period.

Transmit Data (TxD) Serial data is transmitted to the communication line on the TxD output.

Transmitter Clock (TxC) The serial data on TxD is clocked out with the TxC* signal. The relationship between clock rate and baud rate is similar to that for RxC*. Data is shifted out of the Am9551 on the falling edge of TxC*.

Transmitter Ready (TxRDY) The TxRDY output signal goes high when data in the transmit data buffer has been shifted into the transmitter section allowing the transmit data buffer to accept the next byte from the processor. TxRDY is reset when information is written into the transmit data buffer. Loading the command register also resets TxRDY. TxRDY is available only when the Am9551 is enabled to transmit (CTS* = 0, TxEN = 1). However, the TxRDY bit in the status buffer will always be set when the transmit data buffer is empty regardless of the state of TxEN and CTS*.

Transmitter Empty (TxE) The TxE output signal goes high when the transmitter section has transmitted its data and is empty. The signal remains high until a new data byte is shifted from the transmit data buffer to the transmitter section. In the synchronous mode, when the processor does not load a new byte into the buffer in time, TxE will momentarily go to a high level as SYNC characters are loaded into the

transmitter section. TxE going high is independent of the status of the TxEN bit in the command register.

Data Terminal Ready (DTR*) This signal is a general purpose output which reflects the state of bit 1 in the command instruction. It is commonly connected to an associated modem to indicate that the Am9551 is ready.

Data Set Ready (DSR*) This is a general purpose input signal and forms part of the status byte that can be read by the processor. The DSR* signal is generally generated by the modem as a response to the DTR* signal; it indicates that the modem is ready. The signal acts only as a flag and does not control any internal logic.

Request to Send (RTS*) This is a general purpose output, similar to DTR*; it reflects the state of bit 5 in a command instruction. RTS* is normally used to initiate a data transmission by requesting the modem to prepare to send.

Clear to Send (CTS*) This is a general purpose input signal used to enable the Am9551 to transmit data if the TxEN bit in the command byte is a one. CTS* is generally used as a response to RTS* by a modem to indicate that transmission can begin. Designers not using CTS* in their systems should remember to tie it low so that Am9551 data transmission will not be disabled.

4-9. PARALLEL I/O INTERFACE

The Parallel I/O Interface on the Mono-Board provides 48 lines for the transfer and control of data to and from peripheral devices. Sixteen lines have bidirectional driver/terminator networks installed. The remaining 32 lines are uncommitted. Sockets are provided for installing driver/terminator networks in groups of four lines per network.

All forty-eight signal lines emanate from the I/O ports on two Am9555 Programmable Peripheral Interface Chips. The two Am9555s each provide 24 I/O lines and are operationally identical except for addressing. For this discussion only one Am9555 will be covered.

Each Am9555 contains three 8-bit ports (A, B and C). All ports can be configured by software for a variety of functions. Each port has the following special characteristics.

Port A: Provides an output latch, an input latch, an input buffer or a bidirectional bus for eight data bits.

Port B: Provides an output latch, an input latch or an input input buffer for eight data bits.

Port C: Operates as an output latch for eight bits, an input buffer for eight bits, or as two four bit control ports for ports A and B.

Communication between the CPU and the Am9551 is via the data bus and six control lines. Control bytes and data bytes are transmitted to the Am9551; status bytes and data bytes are transmitted from the Am9551 on the data bus. The six control lines provide the necessary controls for all data bus operations.

The chip select (CS55A* and CS55B*) to the CS* inputs allow the chips to be individually addressed. When CS* is true (logical 0), the Am9555s accept or output data or control bytes from the data bus. The direction of data flow is determined by the RD and WR (IOR* and IOW*) inputs to the Am9555. A low (true) IOR* allows the CPU to read data or status from the Am9555. A low (true) on the IOW* allows the CPU to write data or control words to the Am9555.

Address lines 0 and 1 allow selection, by the CPU, of a specific port or control word register. When the chip select (CS*) is true and IOR* or IOW* is true, the CPU can read or write to the ports or control registers identified by 0 and 1 as follows:

A1	A0	SELECTION
0	0	Port A
0	1	Port B
1	0	Port C
1	1	Control Register

All internal registers are cleared and the ports are set to the high impedance input mode when a high level is presented to the Reset input.

4-10. INTERRUPT CONTROLLER

The interrupt controller logic consists of an Am9519 Interrupt Controller and a jumper pad that allows the user to connect any of 16 possible interrupt requests to the Am9519 eight interrupt priority inputs.

The Interrupt Controller resolves priorities among the eight levels. The priority resolution algorithm can be changed dynamically at any time. This means that the complete interrupt structure can be modified as required.

The operation of the Interrupt Controller is controllrd via five control lines and the data bus. The five control lines are decoded to provide controls for programming and reading status. Control words and status information are transferred through the data bus. The functions of the control lines are as follows:

The Chip Select (CS) input conditions the chip to read or write on the data bus. Read/write transfers occur when the CS input is low. The chip select does not condition Interrupt Acknowledge Operations.

The RD (IOR*) input is an active low signal that indicates information is to be transferred from the Am9519 to the data bus.

The WR (ADVIOW*) input is an active low signal that indicates information is to be transferred from the data bus to the Am9519.

The Control/Data (C/$\overline{D}$) input is connected to the low order address bit (A0). A0 identifies the data bus information as control or data. A low on A0 identifies data bus information as data. A high on A0 causes data bus information to be written into the control registers when WR is low; the status register is read when RD is low.

The Interrupt Acknowledge (IACK*) is supplied by the Am8238 INTA* signal. INTA* is used to request interrupt-response information. One response byte is transferred by the Am9519 for each INTA* pulse received. As many as four bytes can be transferred during each interrupt acknowledge sequence. The first INTA* pulse following an interrupt (INT*) to the CPU initiates the internal selection of the highest priority unmasked interrupt.

Two outputs from the Am9519 provide the interrupt to the CPU and also provide coordination of activities between the Am9519 and the CPU. The GINT output from the Am9519 (INT*) indicates to the CPU that at least one bit is set in the Interrupt Request Register. The PAUSE* output signal is used during INTA* cycles to indicate to the CPU that the Am9519 has not completed the data bus transfer. PAUSE* is connected to the CPU RDY input. The INTA* pulse is held true by the CPU until the PAUSE* output goes high.

4-11. ARITHMETIC PROCESSING UNIT (APU)

An optional Am9511 Arithmetic Processing Unit (APU) performs both single (16-bit) and double (32-bit) precision fixed-point arithmetic as well as floating-point addition, subtraction, multiplication, and division. Transcendental derived functions and data manipulation and conversion commands can also be executed by the APU. The following paragraphs are intended to familiarize the user with the basic theory of the Am9511 operation.

Operands and commands are written into the APU, and results and status are read from the APU via the data bus. Four control signals identify these operations to the APU. The Chip Select (CS) input to the APU selects a read or write operation; the RD and WR inputs define the direction of data flow to (ADVIOW* true) or from (IOR* true) the APU. Data bus information is defined as data (C/$\overline{D}$ low and ADVIOW* low) command (C/$\overline{D}$ low and IOR* low), or status (C/$\overline{D}$ high and IOR* low) by the C/D input.

Data (operands) is stored in the Am9551 in an eight level 16-bit wide data stack. Since single precision fixed-point operations are 16-bits wide, eight such values can be concurrently maintained in the stack. When using either double precision fixed-point or floating-point formats, four values can be concurrently stored. Data is written into the stack eight bits at a time. The least significant byte is written first. Data is removed from the stack in the reverse byte order. Data must be entered onto the stack in multiples of the number of bytes appropriate to the chosen format.

Data is removed from the stack in the reverse order of entry. That is, the first byte in is the last byte out. The removal of each data word redefines the top of stack (TOS) so that the next successive byte to be removed becomes TOS. Data removed from the stack rotates to the bottom of the stack.

During data bus operations, an active low output (PAUSE) from the APU to the CPU indicates the APU has not completed its information transfer over the data bus. Whenever a data read or status read operation is requested, PAUSE goes low. It returns high only after the data bus contains valid data. When an existing command is in the process of execution, any read or write request will cause the PAUSE output to go low for the remaining duration of the command plus any time needed for initiating a data bus operation.

When any command has completed execution, an active low (open drain) output (END) indicates that execution of the previous entered command is complete. The END output is applied to the interrupt controller jumper matrix as INT11*.

4-12. DIRECT MEMORY ACCESS CONTROLLER

The Direct Memory Access (DMA) controller allows external devices to directly transfer information to or from the MonoBoard memory. Memory-to-memory transfer capability is also provided. The DMA controller logic consists of an Am9517 Programmable DMA Controller and an Am74LS374 8-bit address register.

The DMA controller contains four fully independent channels. Each channel can be programmed to autoinitialize to its original condition following an End-Of-Process (EOP*) condition. The DMA controller is programmed by writing data words to its various internal registers. When CS and IOW (ADVIOW*) are true, the DMA controller accepts data bytes to program the internal registers. Table 3-1 gives the I/O addresses and the registers programmed. Conversely, when CS and IOR are true, the internal registers can be read by the CPU.

After programming, the DMA controller responds to a DMA request (DREQ) with a hold request (HREQ) to the CPU. The CPU responds with a DHLDA to the DMA hold acknowledge (HACK) input. The DMA then pulls the DMA acknowledge (DACK*) low to notify the requesting device a DMA request has been issued. Once the DMA request has been granted, the DMA controller provides the necessary address and control signals to complete the operation. The input/output controls and their functions are described in the following paragraphs.

CS (Chip Select, Input) Chip Select is an active low input signal used to select the Am9517 as an I/O device during the idle cycle. This allows CPU communication on the data bus.

CLK (Clock, Input) This input controls the internal operations of the Am9517 and its rate of data transfer. CLK is supplied by the phase two TTL from the Am8224 Clock Driver.

RESET (Reset, Input) Reset is an asynchronous active high input which clears the Command, Status, Request and Temporary registers. It also clears the first/last flip-flop and sets the Mask register. Following a reset, the device is in the idle cycle.

READY (Ready, Input) Ready is an asynchronous input used to extend the memory read and write pulses from the Am9517 to accomodate slow memories or I/O peripheral devices.

HACK (Hold Acknowledge, Input) This active high from the CPU indicates that control of the address and data bus has been relinquished.

DREQ0-DREQ3 (DMA Request, Input) The DMA Request lines are individual asynchronous channel request inputs used by peripheral circuits to obtain DMA service. In fixed priority, DREQ0 has the highest priority and DREQ3 has the lowest priority. A request is generated by activating the DREQ line of a channel. DACK will acknowledge the recognition of the DREQ signal. Polarity of DREQ is programmable. Reset initializes these lines to active high.

DB0-DB7 (Data Bus, Input/Output) The Data Bus lines are bidirectional three-state signals. The outputs are enabled in the program condition during the I/O read to output the contents of an address register, a status register, the temporary register or a Word Count register to the CPU. The outputs are disabled and the inputs are read during an I/O write cycle when the CPU is programming the Am9517 control registers. During DMA cycles, the most significant eight bits of the address are output onto the data bus to be strobed into an external latch by ADSTB. In memory-to-memory operations, data from the memory comes into the Am9517 on the data bus during the read-from-memory transfer. In the write-to-memory transfer the data bus outputs place the data into the new memory location.

IOR (I/O Read, Input/Output) I/O Read is a bidirectional active low three-state line. In the idle cycle, it is an input control signal used by the CPU to read the control registers. In the active cycle, it is an output control signal used by the Am9517 to access data from a peripheral during a DMA write transfer.

IOW (I/O Write, Input/Output) I/O Write is a bidirectional active low three-state line. In the idle cycle, it is an input control signal used by the CPU to load information into the Am9517. In the active cycle, it is an output control signal used by the Am9517 to load data to the peripheral during a DMA read transfer.

EOP (End of Process, Input/Output) EOP is an active low bidirectional signal. Information concerning the completion of DMA services is available at the bidirectional End of Process

(EOP) pin. The Am9517 allows an external signal to terminate an active DMA service. This is accomplished by pulling the EOP input low with an external signal. The Am9517 also generates a pulse when the terminal count (TC) for any channel is reached. This generates an EOP* signal which is output through the EOP* line. The reception of EOP*, either internal or external, causes the Am9517 to terminate the service, reset the request and (if autoinitialize is enabled) to write the base registers to the Current Address registers of that channel. The mask bit and TC bit in the status word will be set for the currently active channel by EOP* unless the channel is programmed to autoinitialize, in which case the mask bit remains clear. During memory-to-memory transfers, EOP* will be output when the TC for channel 1 occurs.

A0-A3 (Address, Input/Output) The four least significant address lines are bidirectional three-state signals. In the idle cycle they are inputs and are used by the Am9517 to address the control register to be loaded or read. In the active cycle they are outputs and provide the lower four bits of the output address.

A4-A7 (Address, Output) The four most significant address lines are three-state outputs and provide four bits of address. These lines are enabled only during the DMA service.

HREQ (Hold Request, Output) This is the Hold Request to the CPU and is used to request control of the system bus. If the corresponding mask bit is clear, the presence of any valid DREQ causes the Am9517 to issue the HREQ.

DACK0-DACK3 (DMA Acknowledge, Output) DMA Acknowledge is used to notify the individual peripherals when one has been granted a DMA cycle. These lines can be programmed to be active high or active low; however, reset initializes them to active low.

AEN (Address Enable, Output) The Address Enable is an active high level used to enable the output of the Am74LS374 external latch (which holds the upper byte of address) and to disable the system bus during the DMA cycle. Note that during DMA transfers HACK and AEN should be used to deselect all other I/O peripherals which might erroneously be accessed as programmed I/O during the DMA operation. The Am9517 automatically deselects itself during DMA transfer.

ADSTSB (Address Strobe, Output) The active high Address Strobe is used to strobe the upper address byte into the Am74LS374 external latch.

MEMR (Memory Read, Output) The memory Read output is an active low three-state signal used to access data from the selected memory location during a DMA Read or memory-to-memory transfer.

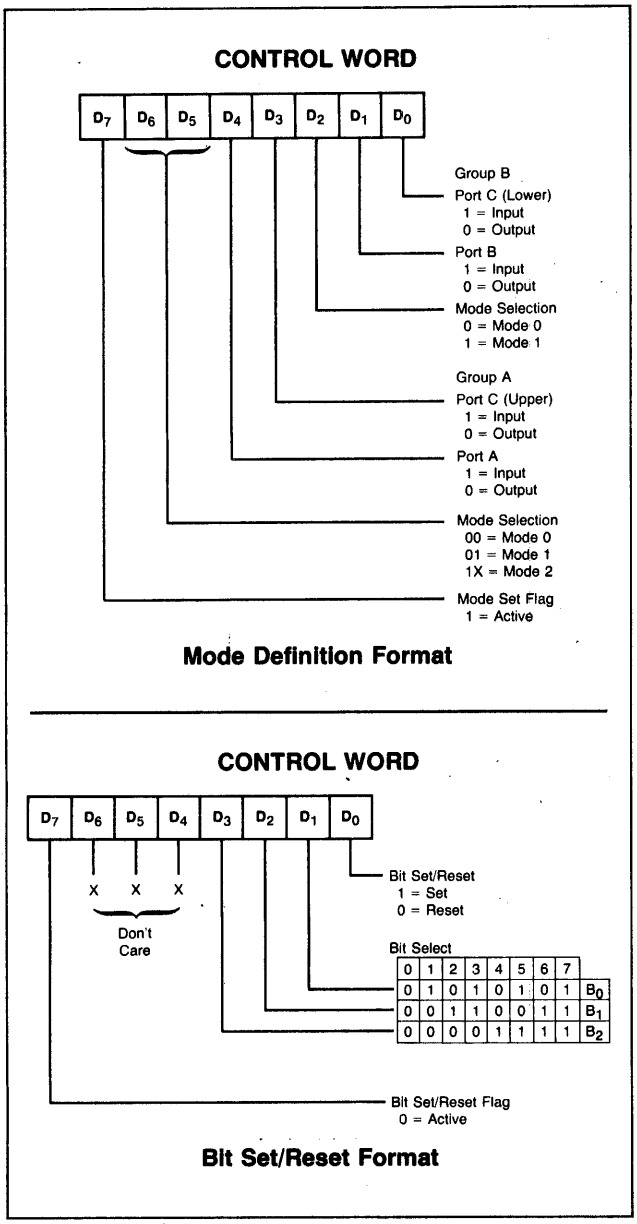
MEMW (Memory Write, Output) The Memory Write output is an active low three-state signal used to write data to the selected memory location during a DMA Write or a memory-to-memory transfer.

APPENDIX A
COMMAND SUMMARIES

A-1. INTRODUCTION

The following are command summaries for the programmable devices in the AMC 95/4005 MonoBoard Computer.

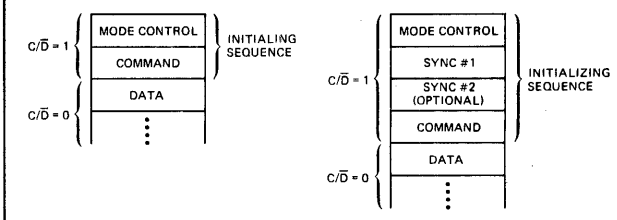
A-2. Am9555 PROGRAMMABLE PERIPHERAL INTERFACE



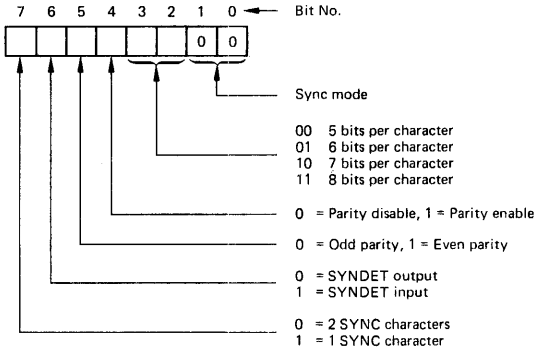
Mode Definition Format

Bit Set/Reset Format

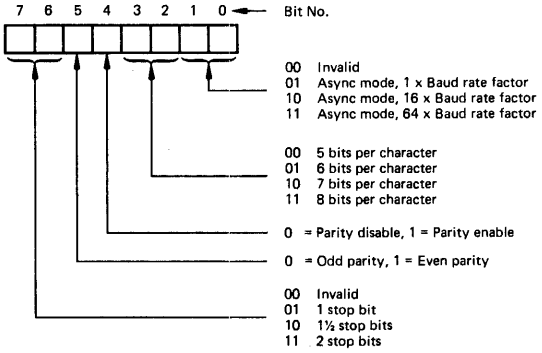
A-3 Am9551 PROGRAMMABLE COMMUNICATIONS INTERFACE



Control Word Sequence for Initialization



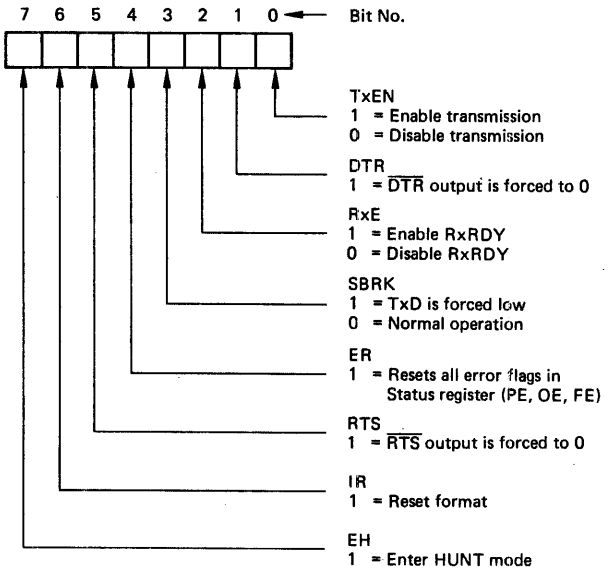
Synchronous Mode Control Code



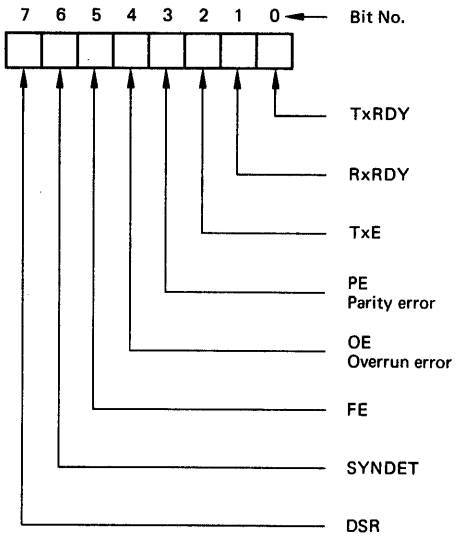
Asynchronous Mode Control Code

TxEN	TxE	TxRDY	
1	1	1	Transmit Output Register and Transmit Character Buffer empty. TxD continues to mark if Am9551 is in the asynchronous mode. TxD will send Sync pattern if Am9551 is in the Synchronous Mode. Data can be entered into Buffer.
1	0	1	Transmit Output Register is shifting a character. Transmit Character Buffer from the processor.
1	1	0	Transmitt Register had finished sending. A new character is waiting for transmission. This is a transient condition.
1	0	0	Transmit Register is currently sending and an additional character is stored in the Transmit Character Buffer for Transmission.
0	0/1	0/1	Transmitter is disabled.

Operation of the Transmitter Section as a Function of TxE, TxRDY and TxEN



The Am9551 Status Register

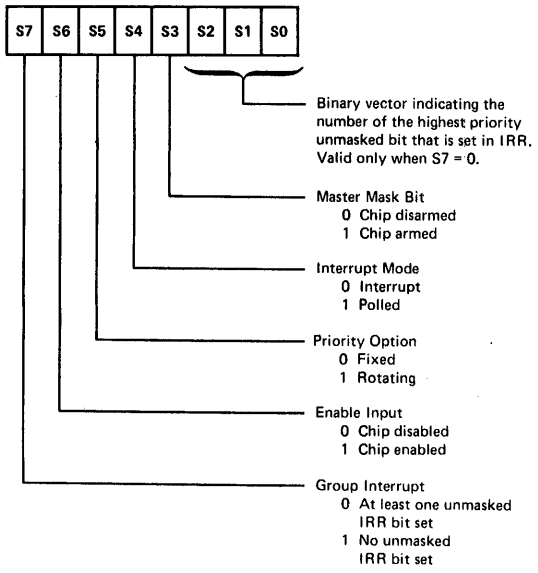


Am9551 Control Command

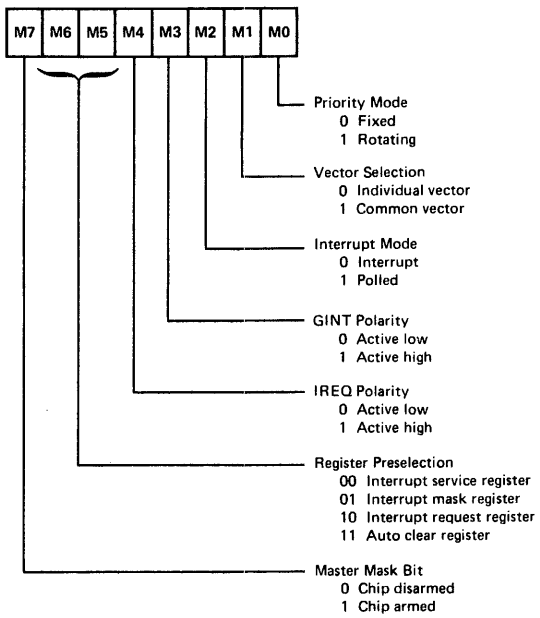
A-4. Am9519 UNIVERSAL INTERRUPT CONTROLLER

COMMAND CODE								COMMAND DESCRIPTION
7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	Reset
0	0	0	1	0	X	X	X	Clear all IRR and all IMR bits
0	0	0	1	1	B2	B1	B0	Clear IRR and IMR bit specified by B2, B1, B0
0	0	1	0	0	X	X	X	Clear all IMR bits
0	0	1	0	1	B2	B1	B0	Clear IMR bit specified by B2, B1, B0
0	0	1	1	0	X	X	X	Set all IMR bits
0	0	1	1	1	B2	B1	B0	Set IMR bit specified by B2, B1, B0
0	1	0	0	0	X	X	X	Clear all IRR bits
0	1	0	0	1	B2	B1	B0	Clear IRR bit specified by B2, B1, B0
0	1	0	1	0	X	X	X	Set all IRR bits
0	1	0	1	1	B2	B1	B0	Set IRR bit specified by B2, B1, B0
0	1	1	0	X	X	X	X	Clear highest priority ISR bit
0	1	1	1	0	X	X	X	Clear all ISR bits
0	1	1	1	1	B2	B1	B0	Clear ISR bit specified by B2, B1, B0
1	0	0	M4	M3	M2	M1	M0	Load Mode register bits 0–4 with specified pattern
1	0	1	0	M6	M5	0	0	Load Mode register bits 5, 6 with specified pattern
1	0	1	0	M6	M5	0	1	Load Mode register bits 5, 6 and set mode bit 7
1	0	1	0	M6	M5	1	0	Load Mode register bits 5, 6 and clear mode bit 7
1	0	1	1	X	X	X	X	Load IMR from data bus
1	1	0	0	X	X	X	X	Load auto-clear register from data bus
1	1	1	BY1	BY0	L2	L1	L0	Load BY1, BY0 into byte count and load data bus into response memory of level specified by L2, L1, L0

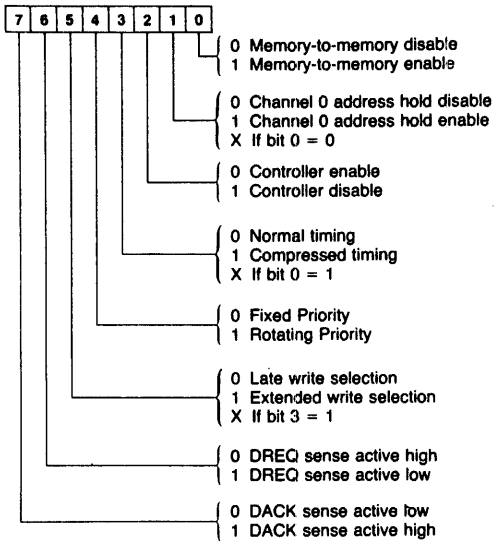
STATUS REGISTER



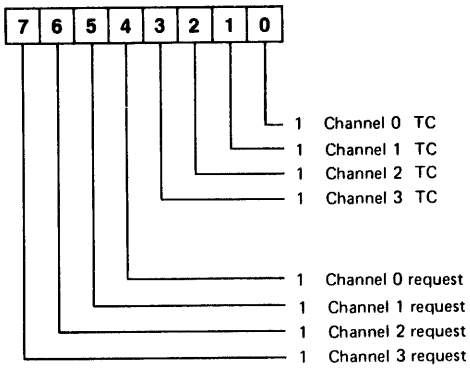
MODE REGISTER



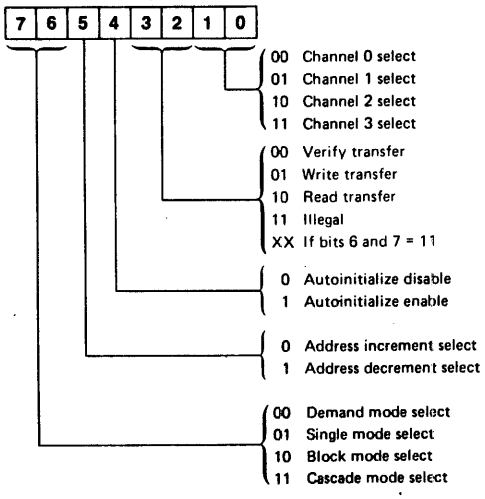
COMMAND REGISTER



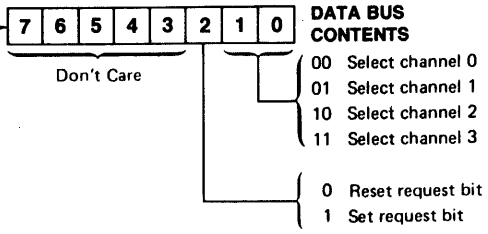
STATUS REGISTER



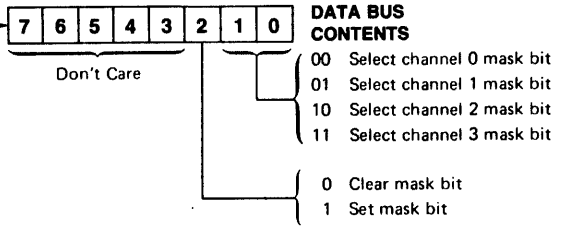
MODE REGISTER



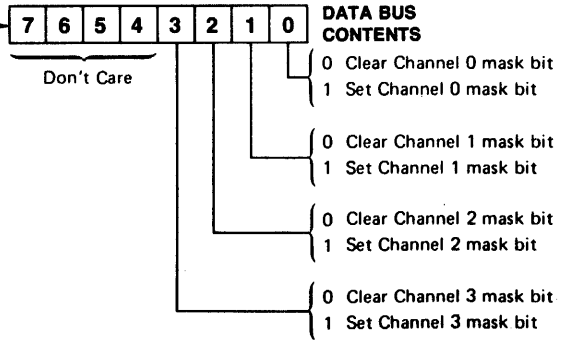
DATA BUS CONTENTS



DATA BUS CONTENTS



DATA BUS CONTENTS



REGISTER ADDRESSING

A3	A2	A1	A0	Operation	Register
0	0	0	0	R/W	Channel 0 Address
0	0	0	1	R/W	Channel 0 Word Count
0	0	1	0	R/W	Channel 1 Address
0	0	1	1	R/W	Channel 1 Word Count
0	1	0	0	R/W	Channel 2 Address
0	1	0	1	R/W	Channel 2 Word Count
0	1	1	0	R/W	Channel 3 Address
0	1	1	1	R/W	Channel 3 Word Count
1	0	0	0	Read	Status Register
1	0	0	0	Write	Command Register
1	0	0	1	Write	Request Register
1	0	1	0	Write	Mask Register Single Bit
1	0	1	1	Write	Mode Register
1	1	0	0	Write	First/Last Bit Cleared
1	1	0	1	Read	Temporary Register
1	1	0	1	Write	Master Reset
1	1	1	1	Write	Mask Register Four Bits

A-5. Am9511 ARITHMETIC PROCESSING UNIT

COMMAND CODE

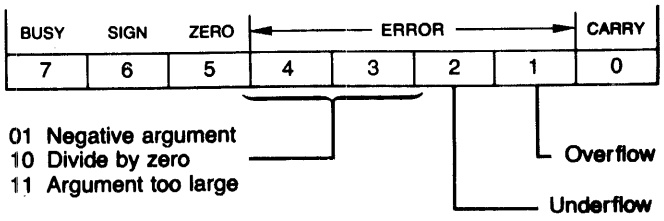
MNEMONIC	HEX OPCODE: SERVICE REQUEST	HEX OPCODE: NO SERVICE REQUEST	CYCLES
FIXED POINT 16-BIT			
SADD	EC	6C	17
SSUB	ED	6D	31
SMUL	EE	6E	88
SDIV	EF	6F	89
FIXED POINT 32-BIT			
DADD	AC	2C	21
DSUB	AD	2D	40
DMUL	AE	2E	194-210
DMUU	B6	36	182-218
DDIV	AF	2F	196-210
FLOATING POINT 32-BIT			
FADD	90	10	54-368
FSUB	91	11	70-370
FMUL	92	12	146-168
FDIV	93	13	154-184
SQRT	81	01	800 *
SIN	82	02	4464 *
COS	83	03	4118 *
TAN	84	04	5754 *
ASIN	85	05	7668 *
ACOS	86	06	7734 *
ATAN	87	07	6006 *
LOG	88	08	4490 *
LN	89	09	4478 *
EXP	8A	0A	4616 *
PWR	8B	0B	9292 *
DATA MANIPULATION			
NOP	80	00	4
FIXS	9F	1F	90-214
FIXD	9E	1E	90-336
FLTS	9D	1D	62-156
FLTD	9C	1C	56-342
CHSS	F4	74	23 *
CHSD	B4	34	27 *
CHSF	95	15	18 *
PTOS	F7	77	16
PTOD	B7	37	20
PTOF	97	17	20
POPS	F8	78	10
POPD	B8	38	12
POPF	98	18	12
XCHS	F9	79	18
XCHD	B9	39	26
XCHF	99	19	26
PUPI	9A	1A	16

*Weighted average execution cycle

PORT ADDRESSING

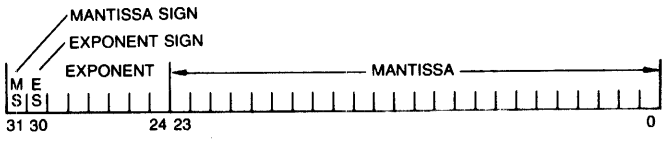
C/$\overline{D}$	$\overline{RD}$	$\overline{WR}$	Operation
0	1	0	Enter data byte into stack
0	0	1	Read data byte from stack
1	1	0	Enter command
1	0	1	Read status

STATUS REGISTER



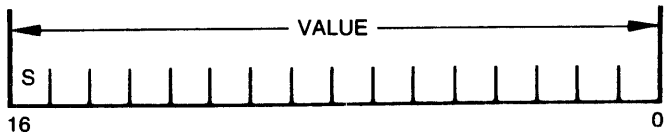
DATA FORMATS

32-BIT FLOATING POINT



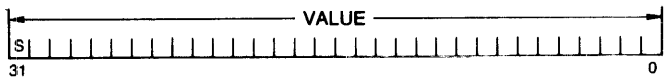
SIGN-MAGNITUDE MANTISSA
UNBIASED TWO'S COMPLEMENT EXPONENT

16-BIT FIXED POINT



TWO'S COMPLEMENT

32-BIT FIXED POINT



TWO'S COMPLEMENT

APPENDIX B
INSTRUCTION SET SUMMARY

B-1. INTRODUCTION

This appendix summarizes all the instruction opcodes for the Am9080A Microprocessor. Instructions are listed by functional categories, in ascending order of hex opcode value, and alphabetically by mnemonic.

B-2. INSTRUCTION SUMMARY BY FUNCTIONAL GROUP

Data Transfer

Hex	Mnemonic	Hex	Mnemonic	Hex	Mnemonic	Hex	Mnemonic
40	MOV B,B	58	MOV E,B	70	MOV M,B	1A	LDAX D
41	MOV B,C	59	MOV E,C	71	MOV M,C	2A	1HLD
42	MOV B,D	5A	MOV E,D	72	MOV M,D	3A	LDA
43	MOV B,E	5B	MOV E,E	73	MOV M,E	02	STAX B
44	MOV B,H	5C	MOV E,H	74	MOV M,H	12	STAX D
45	MOV B,L	5D	MOV E,L	75	MOV M,L	22	SHLD
46	MOV B,M	5E	MOV E,M	77	MOV M,A	32	STA
47	MOV B,A	5F	MOV E,A	78	MOV A,B	01	LXI B
48	MOV C,B	60	MOV H,B	79	MOV A,E	11	LXI D
49	MOV C,C	61	MOV H,C	7A	MOV A,D	21	LXI H
4A	MOV C,D	62	MOV H,D	7B	MOV A,E	31	LXI SP
4B	MOV C,E	63	MOV H,E	7C	MOV A,H	F9	SPHL
4C	MOV C,H	64	MOV H,H	7D	MOV A,L	E3	XTHL
4D	MOV C,L	65	MOV H,L	7E	MOV A,M	EB	XCHG
4E	MOV C,M	66	MOV H,M	7F	MOV A,A	D3	OUT
4F	MOV C,A	67	MOV H,A	06	MVI B	DB	IN
50	MOV D,B	68	MOV L,B	0E	MVI C	C5	PUSH B
51	MOV D,C	69	MOV L,C	16	MVI D	D5	PUSH D
52	MOV D,D	6A	MOV L,D	IE	MVI E	E5	PUSH H
53	MOV D,E	6B	MOV L,E	26	MVI H	F5	PUSH PSW
54	MOV D,H	60	MOV L,H	2E	MVI L	C1	POP B
55	MOV D,L	6D	MOV L,L	36	MVI M	D1	POP D
56	MOV D,M	6E	MOV L,M	3E	MVI A	E1	POP H
57	MOV D,A	6F	MOV L,A	0A	LDAX B	F1	POP PSW

Arithmetic

Hex	Mnemonic	Hex	Mnemonic	Hex	Mnemonic	Hex	Mnemonic
80	ADD B	C6	ADI	9E	SBB M	3C	INR A
81	ADD C	CE	ACI	9F	SBB A	03	INX B
82	ADD D	90	SUB B	D6	SUI	13	INX D
83	ADD E	91	SUB C	DE	SBI	23	INX H
84	ADD H	92	SUB D	09	DAD B	33	INX SP
85	ADD L	93	SUB E	19	DAD D	05	DCR B
86	ADD M	94	SUB H	29	DAD H	0D	DCR C
87	ADD A	95	SUB L	39	DAD SP	15	DCR D
88	ADC B	96	SUB M	27	DAA	ID	DCR E
89	ADC C	97	SUB A	04	INR B	25	DCR H
8A	ADC D	98	SBB B	0C	INR C	2D	DCR L
8B	ADC E	99	SBB C	14	INR D	35	DCR M
8C	ADC H	9A	SBB D	1C	INR E	3D	DCR A
8D	ADC L	9B	SBB E	24	INR H	0B	DCX B
8E	ADC M	9C	SBB H	2C	INR L	1B	DCX D
8F	ADC A	9D	SBB L	34	INR M	2B	DCX H
						3B	DCX SP

Logical

Hex	Mnemonic	Hex	Mnemonic	Hex	Mnemonic	Hex	Mnemonic
A0	ANA B	A9	XRA C	B2	ORA D	BB	CMP E
A1	ANA C	AA	XRA D	B3	ORA E	BC	CMP H
A2	ANA D	AB	XRA E	B4	ORA H	BD	CMP L
A3	ANA E	AC	XRA H	B5	ORA L	BE	CMP M
A4	ANA H	AD	XRA L	B6	ORA M	BF	CMP A
A5	ANA L	AE	XRA M	B7	ORA A	FE	CPI
A6	ANA M	AF	XRA A	F6	ORI	07	RLC
A7	ANA A	EE	XRI	BB	CMP B	0F	RRC
E6	ANI	B0	ORA B	B9	CMP C	17	RAL
A8	XRA B	B1	ORA C	BA	CMP D	1F	RAR
						2F	CMA

Branching

Hex	Mnemonic	Hex	Mnemonic	Hex	Mnemonic
C3	JMP	D7	RST 2	EC	CPE
C2	JNZ	DF	RST 3	F4	CP
CA	JZ	E7	RST 4	FC	CM
D2	JNC	EF	RST 5	C9	RET
DA	JC	F7	RST 6	C0	RNZ
E2	JPO	FF	RST 7	C8	RZ
EA	JPE	CD	CALL	D0	RNC
F2	JP	C4	CNZ	D8	RC
FA	JM	CC	CZ	E0	RPO
E9	PCHL	D4	CNC	E8	RPE
C7	RST 0	DC	CC	F0	RP
CF	RST 1	E4	CPO	F8	RM

Control

Hex	Mnemonic
00	NOP
76	HLT
F3	DI
FB	EI
37	STC
3F	CMC

B-3. INSTRUCTION SUMMARY IN HEXADECIMAL ORDER

Hex	Mnemonic	Hex	Mnemonic	Hex	Mnemonic	Hex	Mnemonic
00	NOP	37	STC	69	MOV L,C	9A	SBB D
01	LXI B	39	DAD SP	6A	MOV L,D	9B	SBB E
02	STAX B	3A	LDA	6B	MOV L,E	9C	SBB H
03	INX B	3B	DCX SP	6C	MOV L,H	9D	SBB L
04	INR B	3C	INR A	6D	MOV L,L	9E	SBB M
05	DCR B	3D	DCR A	6E	MOV L,M	9F	SBB A
06	MVI B	3E	MVI A	6F	MOV L,A	A0	ANA B
07	RLC	3F	CMC	70	MOV M,B	A1	ANA C
09	DAD B	40	MOV B,B	71	MOV M,C	A2	ANA D
0A	LDAX B	41	MOV B,C	72	MOV M,D	A3	ANA E
0B	DCX B	42	MOV B,D	73	MOV M,E	A4	ANA F
0C	INR C	43	MOV B,E	74	MOV M,H	A5	ANA L
0D	DCR C	44	MOV B,H	75	MOV M,L	A6	ANA M
0E	MVI C	45	MOV B,L	76	HLT	A7	ANA A
0F	RRC	46	MOV B,M	77	MOV M,A	A8	XRA B
11	LXI D	47	MOV B,A	78	MOV A,B	A9	XRA C
12	STAX D	48	MOV C,B	79	MOV A,C	AA	XRA D
13	INX D	49	MOV C,C	7A	MOV A,D	AB	XRA E
14	INR D	4A	MOV C,D	7B	MOV A,E	AC	XRA H
15	DCR D	4B	MOV C,E	7C	MOV A,H	AD	XRA L
16	MVI D	4C	MOV C,H	7D	MOV A,L	AE	XRA M
17	RAL	4D	MOV C,L	7E	MOV A,M	AF	XRA A
19	DAD D	4E	MOV C,M	7F	MOV A,A	B0	ORA B
1A	LDAX D	4F	MOV C,A	80	ADD B	B1	ORA C
1B	DCX D	50	MOV D,B	81	ADD C	B2	ORA D
1C	INR E	51	MOV D,C	82	ADD D	B3	ORA E
1D	DCR E	52	MOV D,D	83	ADD E	B4	ORA H
1E	MVI E	53	MOV D,E	84	ADD H	B5	ORA L
1F	RAR	54	MOV D,H	85	ADD L	B6	ORA M
21	LXI H	55	MOV D,L	86	ADD M	B7	ORA A
22	SHLD	56	MOV D,M	87	ADD A	B8	CMP B
23	INX H	57	MOV D,A	88	ADC B	B9	CMP C
24	INR H	58	MOV E,B	89	ADC C	BA	CMP D
25	DCR H	59	MOV E,C	8A	ADC D	BB	CMP E
26	MVI H	5A	MOV E,D	8B	ADC E	BC	CMP H
27	DAA	5B	MOV E,E	8C	ADC H	BD	CMP L
29	DAD H	5C	MOV E,H	8D	ADC L	BE	CMP M
2A	LHLD	5D	MOV E,L	8E	ADC M	BF	CMP A
2B	DCX H	5E	MOV E,M	8F	ADC A	C0	RNZ
2C	INR L	5F	MOV E,A	90	SUB B	C1	POP B
2D	DCR L	60	MOV H,B	91	SUB C	C2	JNZ
2E	MVI L	61	MOV H,C	92	SUB D	C3	JMP
2F	CMA	62	MOV H,D	93	SUB E	C4	CNZ
31	LXI SP	63	MOV H,E	94	SUB H	C5	PUSH B
32	STA	64	MOV H,H	95	SUB L	C6	ADI
33	INX SP	65	MOV H,L	96	SUB M	C7	RST 0
34	INR M	66	MOV H,M	97	SUB A	C8	RZ
35	DCR M	67	MOV H,A	98	SBB B	C9	RET
36	MVI M	68	MOV L,B	99	SBB C	CA	JZ

B-3. INSTRUCTION SUMMARY IN HEXADECIMAL ORDER (Cont.)

Hex	Mnemonic	Hex	Mnemonic	Hex	Mnemonic	Hex	Mnemonic
CC	CZ	D8	RC	E6	ANI	F3	DI
CD	CALL	DA	JC	E7	RST 4	F4	CP
CE	ACI	DB	IN	E8	RPE	F5	PUSH PSW
CF	RST 1	DC	CC	E9	PCHL	F6	ORI
DO	RNC	DE	SBI	EA	JPE	F7	RST 6
D1	POP D	DF	RST 3	EB	XCHG	F8	RM
D2	JUN	EO	RPO	EC	CPE	F9	SPHL
D3	OUT	E1	POP H	EE	XRI	FA	JM
D4	CNC	E2	JPO	EF	RST 5	FB	EI
D5	PUSH D	E3	XTHL	FO	RP	FC	CM
D6	SUI	E4	CPO	F1	POP PSW	FE	CPI
D7	RST 2	E5	PUSH H	F2	JP	FF	RST 7

B-4. INSTRUCTION SUMMARY IN ALPHABETICAL ORDER

Hex	Mnemonic	Hex	Mnemonic	Hex	Mnemonic	Hex	Mnemonic
CE	ACI	19	DAD D	78	MOV A,B	69	MOV L,C
8F	ADC A	29	DAD H	79	MOV A,C	6A	MOV L,D
88	ADC B	39	DAD SP	7A	MOV A,D	6B	MOV L,E
89	ADC C	3D	DCR A	7B	MOV A,E	6C	MOV L,H
8A	ADC D	05	DCR B	7C	MOV A,H	6D	MOV L,L
8B	ADC E	0D	DCR C	7D	MOV A,L	6E	MOV L,M
8C	ADC H	15	DCR D	7E	MOV A,M	77	MOV M,A
8D	ADC L	1D	DCR E	47	MOV B,A	70	MOV M,B
8E	ADC M	25	DCR H	40	MOV B,B	71	MOV M,C
87	ADD A	2D	DCR L	41	MOV B,C	72	MOV M,D
80	ADD B	35	DCR M	42	MOV B,D	73	MOV M,E
81	ADD C	05	DCR B	43	MOV B,E	74	MOV M,H
82	ADD D	1B	DCX D	44	MOV B,H	75	MOV M,L
83	ADD E	2B	DCX H	45	MOV B,L	3E	MVI A
84	ADD H	3B	DCX SP	46	MOV B,M	06	MVI B
85	ADD L	F3	DI	4F	MOV C,A	0E	MVI C
86	ADD M	FB	EI	48	MOV C,B	16	MVI D
C6	ADI	76	HLT	49	MOV C,C	IE	MVI E
A7	ANA A	DB	IN	4A	MOV C,D	26	MVI H
A0	ANA B	3C	INR A	4B	MOV C,E	2E	MVI L
A1	ANA C	04	INR B	4C	MOV C,H	36	MVI M
A2	ANA D	0C	INR C	4D	MOV C,L	00	NOP
A3	ANA E	14	INR D	4E	MOV C,M	B7	ORA A
A4	ANA H	1C	INR E	57	MOV D,A	B0	ORA B
A5	ANA L	24	INR H	50	MOV D,B	B1	ORA C
A6	ANA M	2C	INR L	51	MOV D,C	B2	ORA D
E6	ANI	34	INR M	52	MOV D,D	B3	ORA E
CD	CALL	03	INX B	53	MOV D,E	B4	ORA H
DC	CC	13	INX D	54	MOV D,H	B5	ORA L
FC	CM	23	INX H	55	MOV D,L	B6	ORA M
2F	CMA	33	INX SP	56	MOV D,M	F6	ORI
3F	CMC	DA	JC	5F	MOV E,A	D3	OUT
BF	CMP A	FA	JM	58	MOV E,B	E9	PCHL
B8	CMP B	C3	JMP	59	MOV E,C	C1	POP B
B9	CMP C	D2	JNC	5A	MOV E,D	D1	POP D
BA	CMP D	C2	JNZ	5B	MOV E,E	E1	POP H
BB	CMP E	F2	JP	5C	MOV E,H	F1	POP PSW
BC	CMP H	EA	JPE	5D	MOV E,L	C5	PUSH B
BD	CMP L	E2	JPO	5E	MOV E,M	D5	PUSH D
BE	CMP M	CA	JZ	67	MOV H,A	E5	PUSH H
D4	CNC	3A	LDA	60	MOV H,B	F5	PUSH PSW
C4	CNZ	0A	LDAX B	61	MOV H,C	17	RAL
F4	CP	1A	LDAX D	62	MOV H,D	1F	RAR
EC	CPE	2A	LHLD	63	MOV H,E	D8	RC
FE	CPI	01	LXI B	64	MOV H,H	C9	RET
E4	CPO	11	LXI D	65	MOV H,L	07	RLC
CC	CZ	21	LXI H	66	MOV H,M	F8	RM
27	DAA	31	LXI SP	6F	MOV L,A	D0	RNC
09	DAD B	7F	MOV A,A	68	MOV L,B	C0	RNZ

B-4. INSTRUCTION SUMMARY IN ALPHABETICAL ORDER (Cont.)

Hex	Mnemonic	Hex	Mnemonic	Hex	Mnemonic	Hex	Mnemonic
F0	RP	C8	RZ	32	STA	D6	SUI
E8	RPE	9F	SBB A	02	STAX B	EB	XCHG
E0	RPO	98	SBB B	12	STAX D	AF	XRA A
0F	RRC	99	SBB C	37	STC	A8	XRA B
C7	RST 0	9A	SBB D	97	SUB A	A9	XRA C
CF	RST 1	9B	SBB E	90	SUB B	AA	XRA D
D7	RST 2	9C	SBB H	91	SUB C	AB	XRA E
DF	RST 3	9D	SBB L	92	SUB D	AC	XRA H
E7	RST 4	9E	SBB M	93	SUB E	AD	XRA L
EF	RST 5	DE	SBI	94	SUB H	AE	XRA M
F7	RST 6	22	SHLD	95	SUB L	EE	XRI
FF	RST 7	F9	SPHL	96	SUB M	E3	XTHL

APPENDIX C
ASCII CHARACTER SET

C-1. INTRODUCTION

The ASCII internal character set used with the AMC 95/4005 is the ANSI X3.4 1968 Version. The following is a summary of the ANSI X3.4 1968.

ASCII CODES

GRAPHIC OR CONTROL	ASCII (HEXADECIMAL)	GRAPHIC OR CONTROL	ASCII (HEXADECIMAL)	GRAPHIC OR CONTROL	ASCII (HEXADECIMAL)
NUL	00	+	2B	V	56
SOH	01	,	2C	W	57
STX	02	–	2D	X	58
ETX	03	.	2E	Y	59
EOT	04	/	2F	Z	5A
ENQ	05	0	30	[	5B
ACK	06	1	31	\	5C
BEL	07	2	32	]	5D
BS	08	3	33	^ (↑)	5E
HT	09	4	34	–(←)	5F
LF	0A	5	35	`	60
VT	0B	6	36	a	61
FF	0C	7	37	b	62
CR	0D	8	38	c	63
SO	0E	9	39	d	64
SI	0F	:	3A	e	65
DLE	10	;	3B	f	66
DC1 (X-ON)	11	<	3C	g	67
DC2 (TAPE)	12	=	3D	h	68
DC3 (X-OFF)	13	>	3E	i	69
DC4 (TAPE)	14	?	3F	j	6A
NAK	15	@	40	k	6B
SYN	16	A	41	l	6C
ETB	17	B	42	m	6D
CAN	18	C	43	n	6E
EM	19	D	44	o	6F
SUB	1A	E	45	p	70
ESC	1B	F	46	q	71
FS	1C	G	47	r	72
GS	1D	H	48	s	73
RS	1E	I	49	t	74
US	1F	J	4A	u	75
SP	20	K	4B	v	76
!	21	L	4C	w	77
"	22	M	4D	x	78
#	23	N	4E	y	79
$	24	O	4F	z	7A
%	25	P	50	{	7B
&	26	Q	51	|	7C
'	27	R	52	} (ALT MODE)	7D
(	28	S	53	~	7E
)	29	T	54	DEL (RUB OUT)	7F
*	2A	U	55		

APPENDIX D
SERVICE INFORMATION

D-1. INTRODUCTION

This appendix provides information on service and repair assistance, user replaceable parts and service diagrams for the AMC 95/4005 MonoBoard Computer.

D-2. SERVICE AND REPAIR ASSISTANCE

If it is necessary to return the AMC 95/4005 MonoBoard Computer to Advanced Micro Computers for service or repair, contact the Service Manager for OEM Products at the telephone number listed below. A Return Material Authorization number must be obtained prior to shipment. When reshipment is due to the board being damaged during shipment from AMC or the board is out of warranty, a purchase order is required to complete the repair.

Repackage the board in the original packing material, or an equivalent substitute, and enclose in a corrugated carton suitable for shipping. Seal the shipping carton securely, mark it FRAGILE, and address to:

Advanced Micro Computers
Service Manager, OEM Products
3340 Scott Boulevard
Santa Clara, California 95051

TELEPHONE: (408) 988-7777

TOLL FREE:
800-672-3548 California
800-538-9791 U.S.A. (Except California)

D-3. USER-REPLACEABLE PARTS

A listing of all user replaceable parts is provided in table D-1. Figure D-1 is a component location diagram.

D-4. SERVICE DIAGRAMS

The MonoBoard Computer component locations are shown on the assembly drawing, figure D-1. Part numbers for the components shown on the assembly drawing are listed in table D-1.

The programmed bit patterns for the MonoBoard Memory Mapping PROM and the I/O Mapping PROM are shown in figures D-2 and D-3 respectively.

Schematic diagrams of the MonoBoard Computer are shown in figures D-4 through D-9. Active-low (logical 0) signals are indicated by an asterisk (*) following the signal name.

TABLE D-1. USER REPLACEABLE PARTS

AMC Part Number	Description	Location
340035	Capacitor, 33 PF, 100V, 5%	C13, 25
340044	Capacitor, 330 PF, 200V, 5%	C7
340011	Capacitor, .1MF, 100V	C1-6, 10, 14, 16, 18-22, 24, 26, 30, 32, 33
340038	Capacitor, 1MF, 35V, 20%	C17
340046	Capacitor, 10MF, 16V, 20%	C15
340042	Capacitor, 22MF, 15V, 20%	C27-29, 31, 34
340012	Capacitor, 82PF (3MHz)	C12
340040	Capacitor, 1000PF, 100V, 5% (3MHz)	C11
470003	Card Extractor	
650002	Diode, IN 914	CR1, 2, 3, 4
380004	Inductor, .33MH	L1
280007	Integrated circuit, Am1488	U13
280001	Integrated circuit, Am1489	U14
260003	Integrated circuit, Am4702	U17
210003	Integrated circuit, Am26123	U46
240017	Integrated circuit, Am27S13	U39
240004	Integrated circuit, Am27S21	U32,67
240016	Integrated circuit, Am2708PC	U42-45
220010	Integrated circuit, Am8224PC	U34
220014	Integrated circuit, Am8226PC (2MHz)	U63, 64
220015	Integrated circuit, Am8226PC (3MHz)	U63, 64
280002	Integrated circuit, Am8228PC	U50
260004	Integrated circuit, Am9080APC (2MHz)	U49
260023	Integrated circuit, Am9080A-1PC (3MHz)	U49
230005	Integrated circuit, Am9114CPC	U22-29
260008	Integrated circuit, Am9511DC (2MHz)	U68
260019	Integrated circuit, Am9511-1DC (3MHz)	U68
260009	Integrated circuit, Am9517PC (2MHz)	U51
260024	Integrated circuit, Am9517-1DC (3MHz)	U51
260011	Integrated circuit, Am9519PC (2MHz)	U66
260020	Integrated circuit, Am9519-1DC (3MHz)	U66
260001	Integrated circuit, Am9551PC (2MHz)	U18
260013	Integrated circuit, Am9551-4PC (3MHz)	U18
260025	Integrated circuit, Am9555APC (2MHz)	U20, 21
260026	Integrated circuit, Am9555A-1PC (3MHz)	U20, 21
210005	Integrated circuit, SN7438	U19
310008	Integrated circuit, TIL 113	U15, 16
260030	Integrated circuit, 4020	U72
220002	Integrated circuit, 74S00	U59
220024	Integrated circuit, 74S03	U48
220017	Integrated circuit, 74S04	U54
220004	Integrated circuit, 74S11	U56
220013	Integrated circuit, 74S74	U57
220025	Integrated circuit, 74S109	U55
220007	Integrated circuit, 74S241	U58
200045	Integrated circuit, 74LS00	U31, 53

TABLE D-1. USER REPLACEABLE PARTS (Cont.)

AMC Part Number	Description	Location
200032	Integrated circuit, 74LS04	U36, 69
200001	Integrated circuit, 74LS08	U37
200057	Integrated circuit, 74LS11	U30
200058	Integrated circuit, 74LS21	U60
200003	Integrated circuit, 74LS32	U38
200004	Integrated circuit, 74LS74	U47, 52
200005	Integrated circuit, 74LS138	U33
200006	Integrated circuit, 74LS139	U40
200042	Integrated circuit, 74LS193	U71
200016	Integrated circuit, 74LS240	U61, 62
200078	Integrated circuit, 74LS273	U41
200087	Integrated circuit, 74LS368	U35
200026	Integrated circuit, 74LS374	U65
850101	MonoBoard Sub-Assembly	
450016	14 Pin Socket	U3, 4, 5, 6, 9, 10, 11, 12
450017	16 Pin Socket	U1, 2, 7, 8, 17, 32, 39, 67
450018	18 Pin Socket	U22-29
450019	24 Pin Socket	U42-45, 68
450020	28 Pin Socket	U18, 50, 66
450021	40 Pin Socket	U20, 21, 49, 51
750036	PWB MonoBoard	
630040	Resistor, 47 ohm, 1/4W, 5%	R3
630058	Resistor, 100 ohm, 1/4W, 5%	R10
630046	Resistor, 470 ohm, 1/4W, 5%	R5, 11, 16
630029	Resistor, 680 ohm, 1/4W, 5%	R8
630029	Resistor, 1K, 1/4W,1 5%	R1, 2
630043	Resistor, 2.2K, 1/4W, 5%	R6, 9
630019	Resistor, 2.7K, 1/4W, 5%	R7
630017	Resistor, 4.7K 1/4W, 5%	R4, 12, 15, 18, 19, 21
630033	Resistor, 10K 1/4W, 5%	R22
630060	Resistor, 68K, 1/4W, 5%	R17
630061	Resistor, 220K, 1/4W, 5%	R13
630012	Resistor Network, 1K SIP, 10 Pin	RP1, 2
630014	Resistor Network, 4.7K SIP, 8 Pin	RP3, 4, 6, 7, 9
630056	Resistor Network, 4.7K SIP, 10 Pin	RP5, 8
960036	Schematic	
690007	Switch, 4 Pos. DIP	SW1
310502	Transistor, 2N4403	Q1
430011	Wire Wrap, Jumper Pin	
480003	Xtal, 18MHz (2MHz)	Y1
480016	Xtal OSC 19.6608MHz	OSC-1
480012	Xtal, 27MHz (3MHz)	Y1

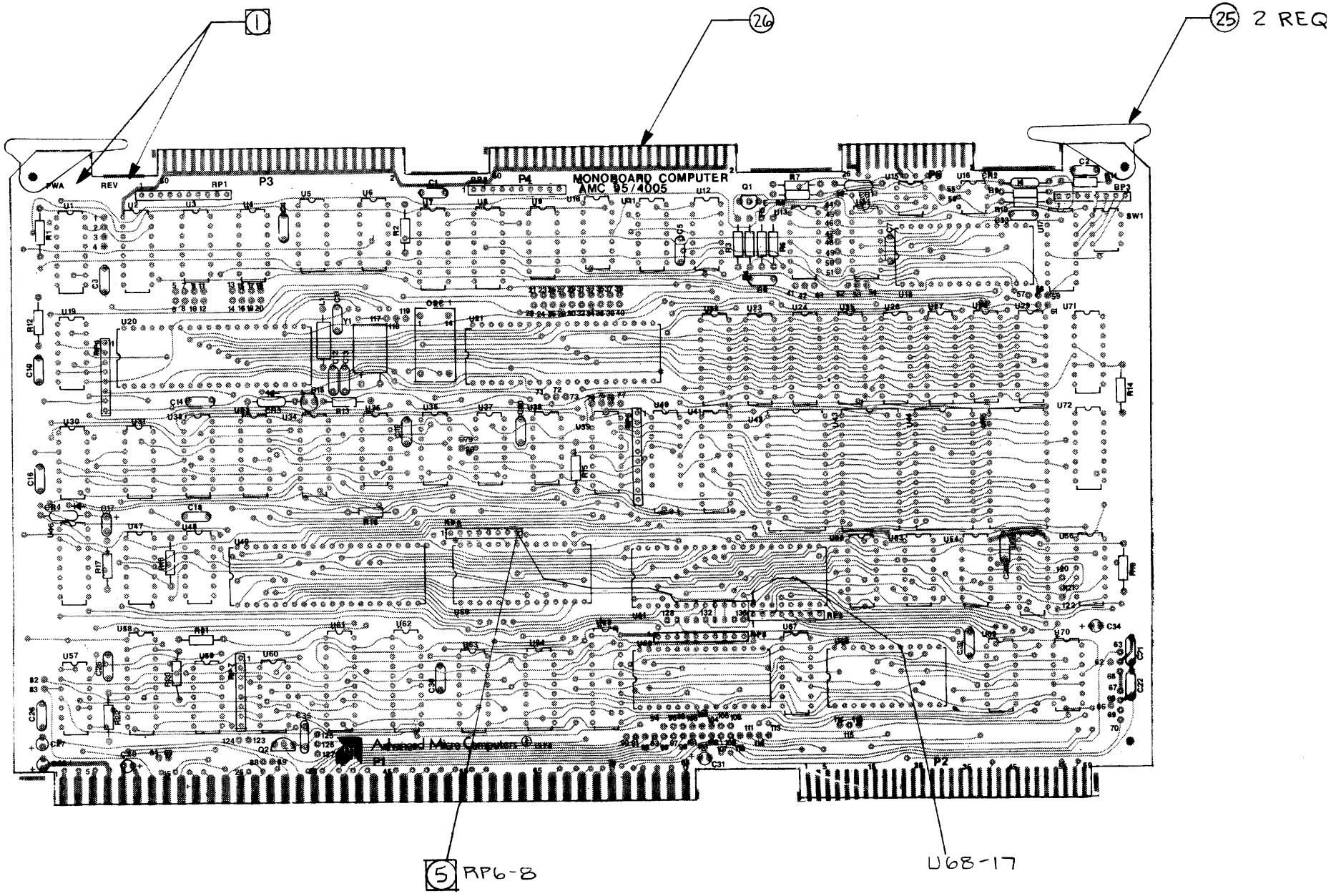


Figure D-1. AMC 95/4005 Components Location Diagram.

00: 8	20: F	40: 8	60: F	80: 8	A0: F	C0: F	E0: F
01: 9	21: ↑	41: 8	61: ↑	81: 8	A1: ↑	C1: ↑	E1: ↑
02: A	22:	42: 9	62:	82: 8	A2:	C2:	E2:
03: B	23:	43: 9	63:	83: 8	A3:	C3:	E3:
04: F	24:	44: A	64:	84: 9	A4:	C4:	E4:
05: F	25:	45: A	65:	85: 9	A5:	C5:	E5:
06: F	26:	46: B	66:	86: 9	A6:	C6:	E6:
07: F	27:	47: F	67:	87: 9	A7:	C7:	E7:
08: F	28:	48: F	68:	88: A	A8:	C8:	E8:
09: F	29:	49: F	69:	89: A	A9:	C9:	E9:
0A: F	2A:	4A: F	6A:	8A: A	AA:	CA:	EA:
0B: F	2B:	4B: F	6B:	8B: A	AB:	CB:	EB:
0C: 4	2C:	4C: 4	6C:	8C: B	AC:	CC:	EC:
0D: 5	2D:	4D: 5	6D:	8D: B	AD:	CD:	ED:
0E: 6	2E:	4E: 6	6E:	8E: B	AE:	CE:	EE:
0F: 7	2F:	4F: 7	6F:	8F: B	AF:	CF:	EF:
10: F	30:	50: F	70:	90: 4	B0:	D0:	F0:
11: F	31:	51: F	71:	91: 5	B1:	D1:	F1:
12: F	32:	52: F	72:	92: 6	B2:	D2:	F2:
13: F	33:	53: F	73:	93: 7	B3:	D3:	F3:
14: ↑	34:	54: F	74:	94: F	B4:	D4:	F4:
15:	35:	55: ↑	75:	95: F	B5:	D5:	F5:
16:	36:	56:	76:	96: F	B6:	D6:	F6:
17:	37:	57:	77:	97: F	B7:	D7:	F7:
18:	38:	58:	78:	98: ↑	B8:	D8:	F8:
19:	39:	59:	79:	99:	B9:	D9:	F9:
1A:	3A:	5A:	7A:	9A:	BA:	DA:	FA:
1B:	3B:	5B:	7B:	9B:	BB:	DB:	FB:
1C:	3C:	5C:	7C:	9C:	BC:	DC:	FC:
1D:	3D:	5D:	7D:	9D:	BD:	DD:	FD:
1E: ↓	3E: ↓	5E: ↓	7E: ↓	9E: ↓	BE: ↓	DE: ↓	FE: ↓
1F: F	3F: F	5F: F	7F: F	9F: F	BF: F	DF: F	FF: F

Address: MSB = A7
LSB = A0

Data: MSB = 03
LSB = 00

APPROVALS	DATE
DRAWN A. HAYWOOD	6-20-79
CHECKED BILL ORR	6-20-79
ISSUED	

Advanced Micro Computers
3340 SCOTT BLVD, SANTA CLARA CA. 95051

256 X 4 PROM PROGRAM

Memory Map Prom for OEM Board

SIZE		DWG. NO.	REV.
A		290005	A_1
SCALE		SHEET 1 OF 1	

Figure D-2. Memory Mapping PROM Pattern.

00: F	20: F	40: F	60: F	80: F	A0: F	C0: 8	E0: 6
01:	21:	41:	61:	81:	A1:	C1: F	E1: F
02:	22:	42:	62:	82:	A2:	C2: 2	E2: F
03:	23:	43:	63:	83:	A3:	C3: F	E3: F
04:	24:	44:	64:	84:	A4:	C4:	E4: 4
05:	25:	45:	65:	85:	A5:	C5:	E5: F
06:	26:	46:	66:	86:	A6:	C6:	E6: 4
07:	27:	47:	67:	87:	A7:	C7:	E7: F
08:	28:	48:	68:	88:	A8:	C8:	E8: 5
09:	29:	49:	69:	89:	A9:	C9:	E9: F
0A:	2A:	4A:	6A:	8A:	AA:	CA:	EA: 5
0B:	2B:	4B:	6B:	8B:	AB:	CB:	EB: F
0C:	2C:	4C:	6C:	8C:	AC:	CC:	EC: 3
0D:	2D:	4D:	6D:	8D:	AD:	CD:	ED: F
0E:	2E:	4E:	6E:	8E:	AE:	CE:	EE:
0F:	2F:	4F:	6F:	8F:	AF: F	CF:	EF:
10:	30:	50:	70:	90:	B0: 1	D0:	F0:
11:	31:	51:	71:	91:	B1: F	D1:	F1:
12:	32:	52:	72:	92:	B2: 1	D2:	F2:
13:	33:	53:	73:	93:	B3: F	D3:	F3:
14:	34:	54:	74:	94:	B4: 1	D4:	F4:
15:	35:	55:	75:	95:	B5: F	D5:	F5:
16:	36:	56:	76:	96:	B6: 1	D6:	F6:
17:	37:	57:	77:	97:	B7: F	D7:	F7:
18:	38:	58:	78:	98:	B8: 1	D8:	F8:
19:	39:	59:	79:	99:	B9: F	D9:	F9:
1A:	3A:	5A:	7A:	9A:	BA: 1	DA:	FA:
1B:	3B:	5B:	7B:	9B:	BB: F	DB:	FB:
1C:	3C:	5C:	7C:	9C:	BC: 1	DC:	FC:
1D:	3D:	5D:	7D:	9D:	BD: F	DD:	FD:
1E:	3E:	5E:	7E:	9E:	BE: 1	DE:	FE:
1F: F	3F: F	5F: F	7F: F	9F: F	BF: F	DF: F	FF: F

Address: MSB = A7
LSB = A0

Data: MSB = 03
LSB = 00

APPROVALS	DATE
DRAWN A. HAYWOOD	6-20-79
CHECKED BILL ORR	6-20-79
ISSUED	

Advanced Micro Computers
3340 SCOTT BLVD, SANTA CLARA CA. 95051

256 X 4 PROM PROGRAM

U32-2 I/O Mapping Prom

SIZE		DWG. NO.	REV.
A		290001	A1

SCALE | SHEET 1 OF 1

Figure D-3. I/O Mapping PROM Pattern.

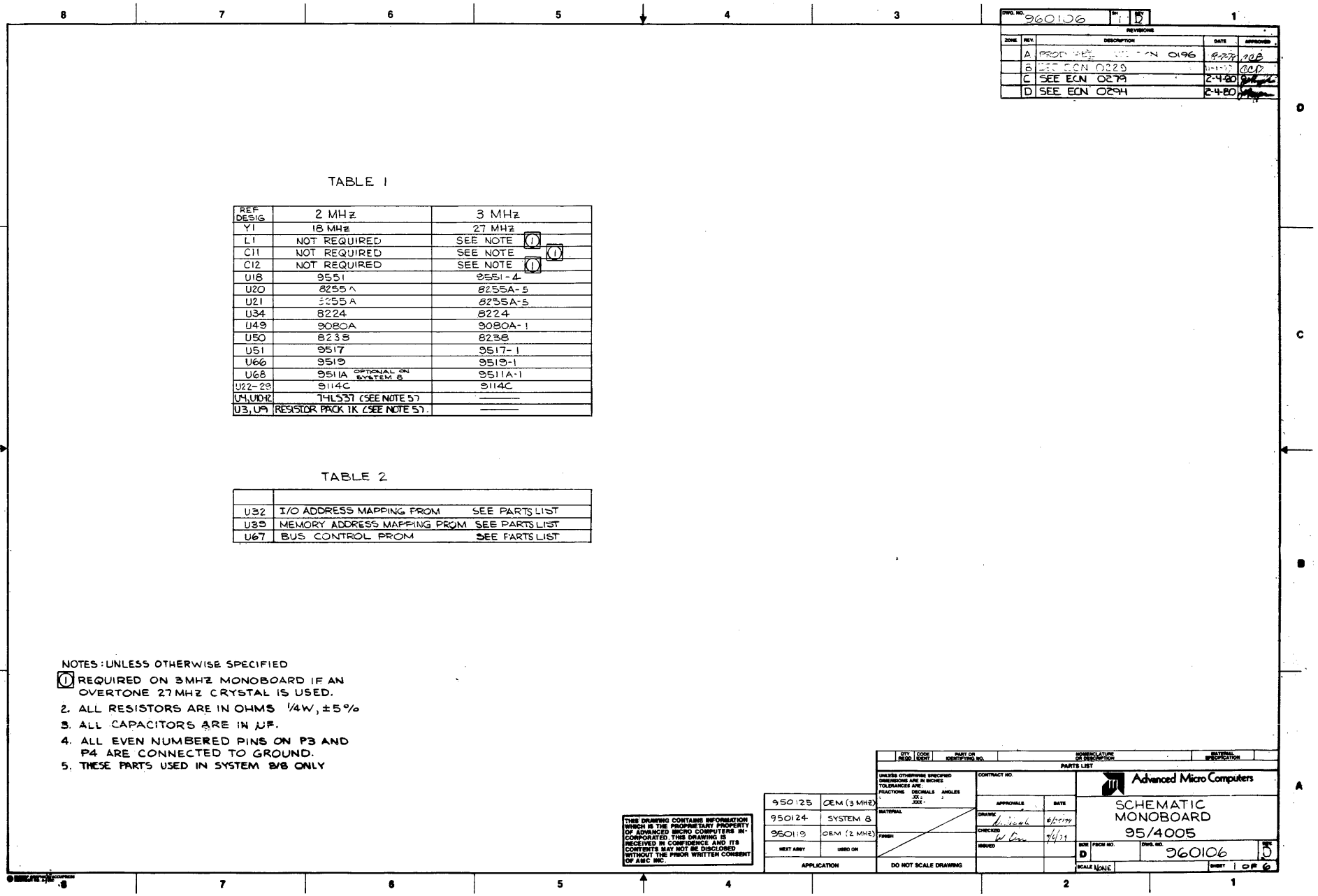


Figure D-4. AMC 95/4005 Schematic Sheet 1.

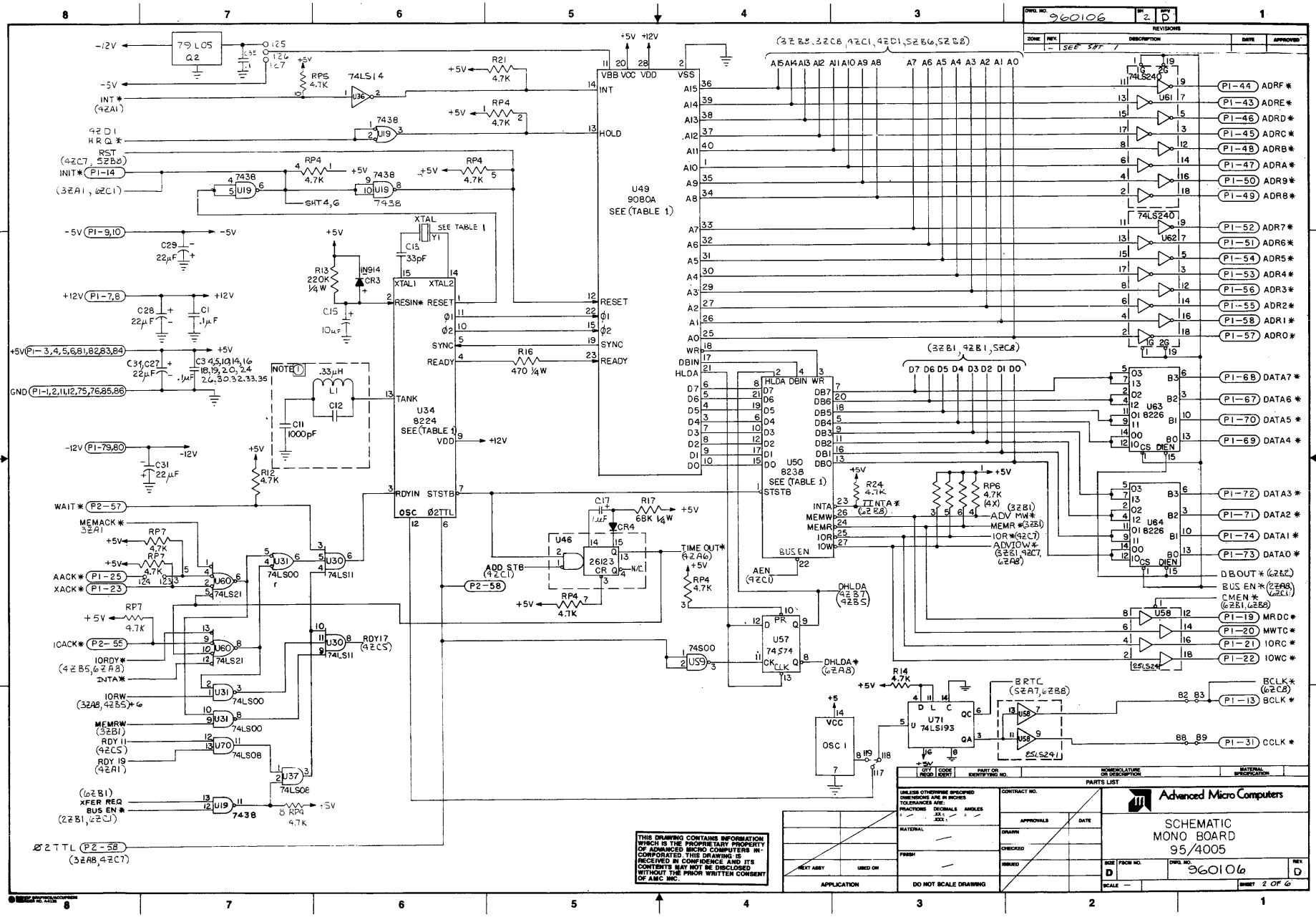


Figure D-5. AMC 95/4005 Schematic Sheet 2.

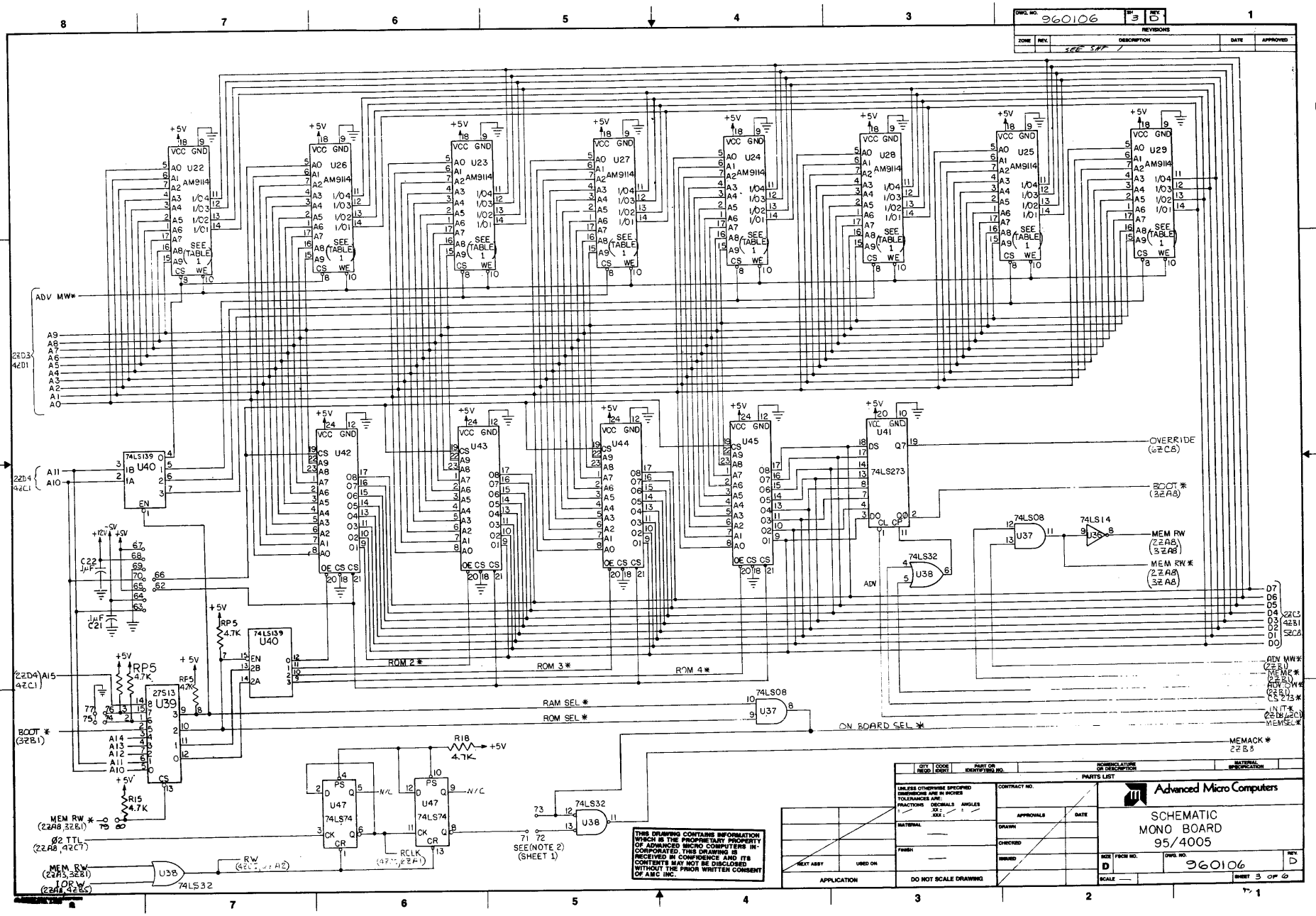


Figure D-6. AMC 95/4005 Schematic Sheet 3.

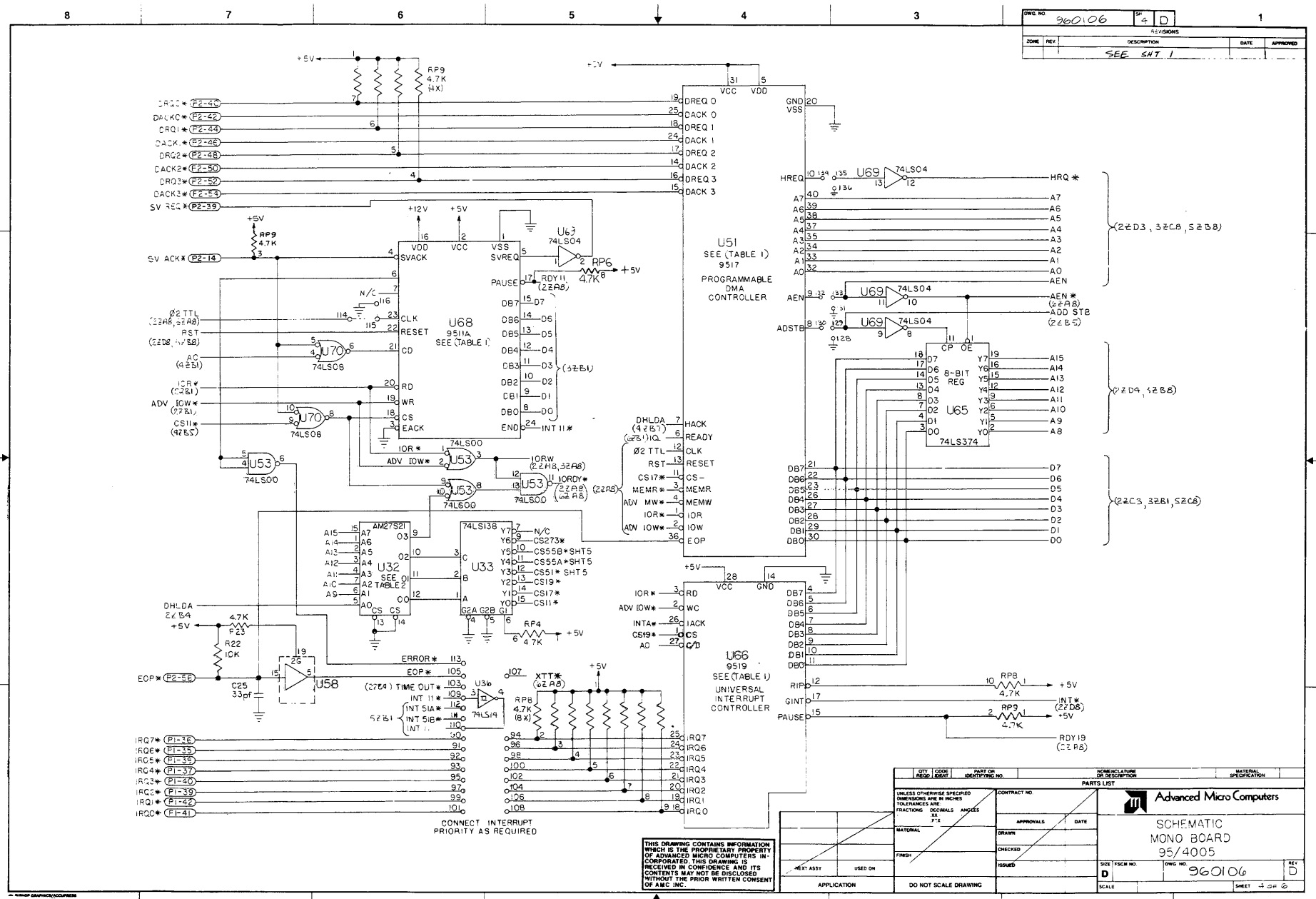


Figure D-7. AMC 95/4005 Schematic Sheet 4.

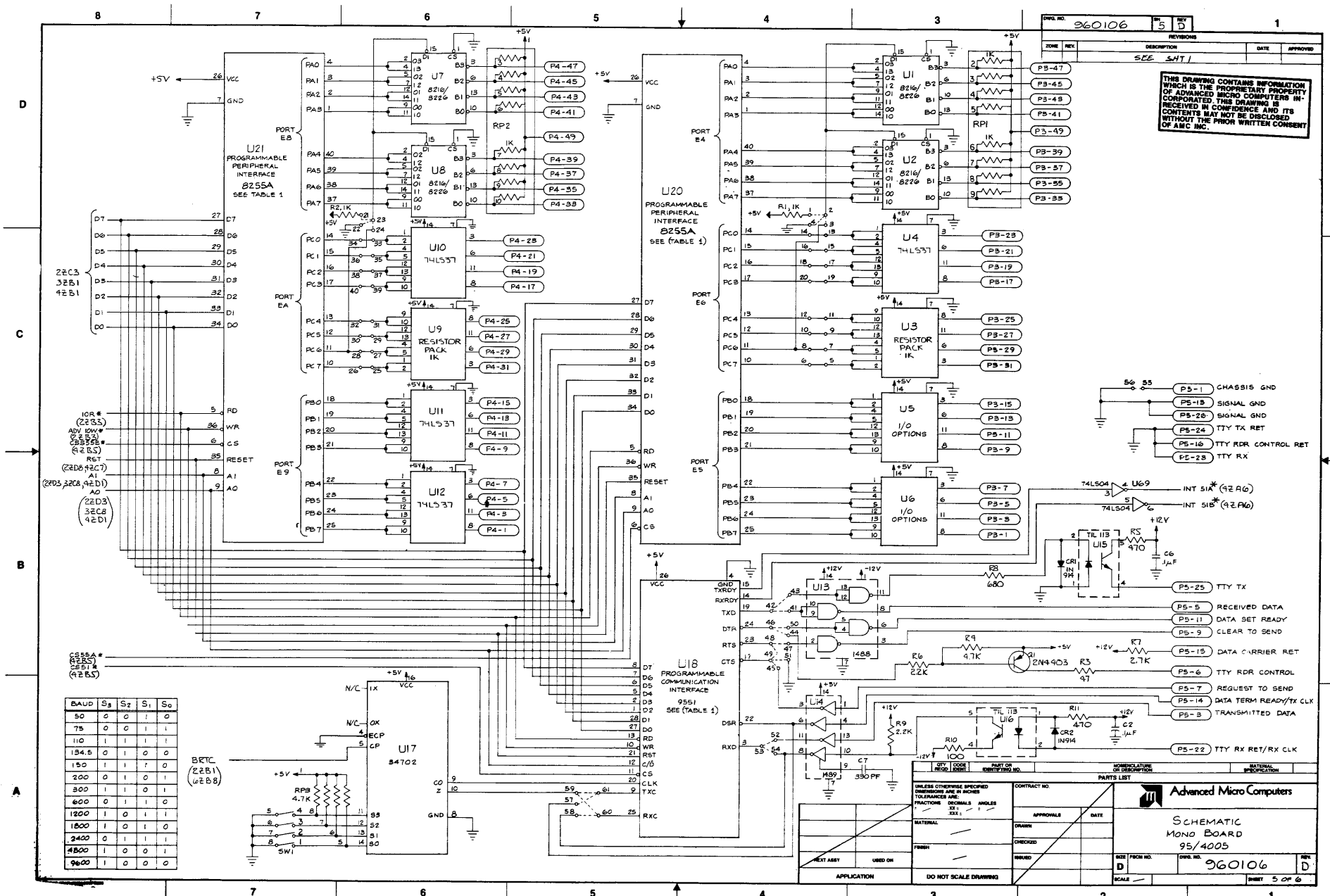


Figure D-8. AMC 95/4005 Schematic Sheet 5.

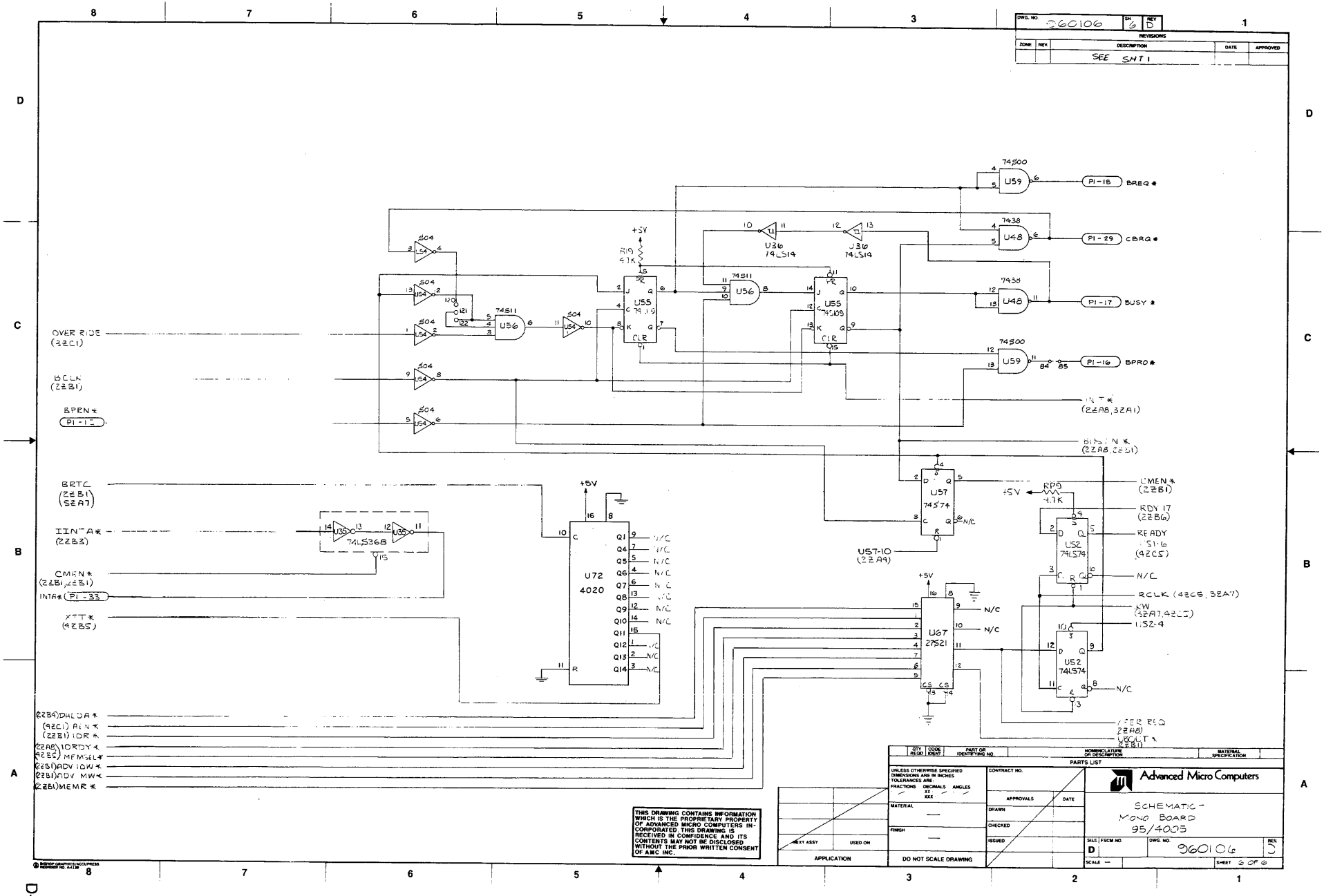


Figure D-9. AMC 95/4005 Schematic Sheet 6.

COMMENT SHEET

Address comments to:

Advanced Micro Computers
Publications Department
3340 Scott Boulevard
Santa Clara, CA 95051

TITLE: MONOBOARD COMPUTER
PUBLICATION NO: 0680120C

COMMENTS: (Describe errors, suggested additions or deletions, and include page numbers, etc.)

From: Name: _____ Position: _____

Company: _____

Address: _____

Advanced Micro Computers

A subsidiary of
Advanced Micro Devices
3340 Scott Boulevard
Santa Clara,
California 95051
(408) 988-7777
TELEX: 171 142
Distributed by
Advanced Micro Devices