TECHNICAL MANUAL FOR

AM-100

2-BOARD 16-BIT CPU

DWM-00100-00
Revision A01
December 1979

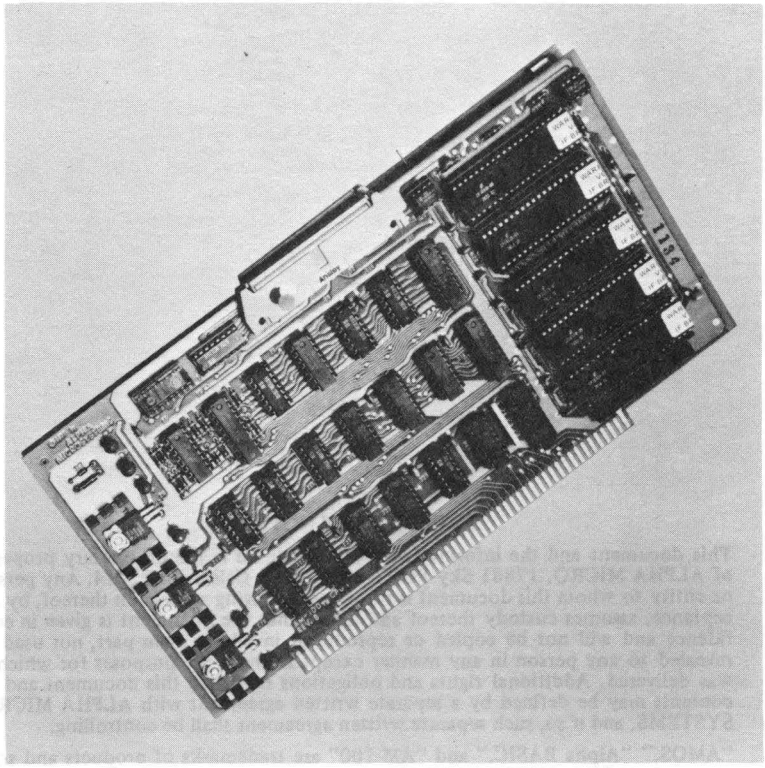
alpha micro

TECHNICAL MANUAL

FOR

AM-100

2-BOARD 16-BIT CPU



Manufactured By

ALPHA MICROSYSTEMS
17881 SKY PARK NORTH
IRVINE, CALIFORNIA 92714

USER COMMENTS

TECHNICAL PUBLICATIONS FILE REFERENCE

FROM: NAME _____

ADDRESS _____

CITY _____

STATE _____ ZIP _____

NOTE: Use this form to communicate any errors, suggested changes, or general comments about this document.

If necessary, call us at:

(714) 957-6076

DOCUMENT: TITLE/NUMBER/REVISION _____

_____

COMMENTS:

FOLD, STAPLE & MAIL

STAPLE

STAPLE

FOLD

FOLD

PLACE
STAMP
HERE

17881 Sky Park North
Irvine, California
92714

ATTN: EDUCATIONAL SERVICES DIVISION

FOLD

FOLD

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION 1
GENERAL DESCRIPTION

1.0 INTRODUCTION.

This manual provides operating and maintenance instructions for the AM-100 Central Processor Unit (CPU) circuit board set manufactured by Alpha Microsystems located in Irvine, California. Circuit board description, operating and usage instructions, programming, theory of operation, and maintenance instructions are included to provide the user with the information necessary to utilize this circuit board to its full capability.

1.1 CIRCUIT BOARD DESCRIPTION.

The AM-100 CPU circuit board set is a 16-bit microprocessor board set that is compatible with the S-100 Bus structure. The AM-100 utilizes the Western Digital WD16 chip set microprogrammed to enhance the software of the operating system. The microprocessor provides 16-bit flexibility and speed with floating point arithmetic to provide large throughput. The two board AM-100 supports most of the standard S-100 Bus peripherals including static memory, I/O facilities and video.

A simplified block diagram of the AM-100 CPU board set is shown in Figure 1-1. For a complete detailed description of CPU operation, see Section 4 of this manual.

1.2 APPLICATION.

The AM-100 is a 16-bit CPU that is fully compatible with a standard 8-bit S-100 bus system. It is fully compatible with many available peripherals from other manufacturers. A block diagram of the system capability is contained in Figure 1-2. This shows the basic S-100 Bus structure, the currently available Alpha Micro circuit cards, and the commercially available peripherals that can be used for a fully integrated system.

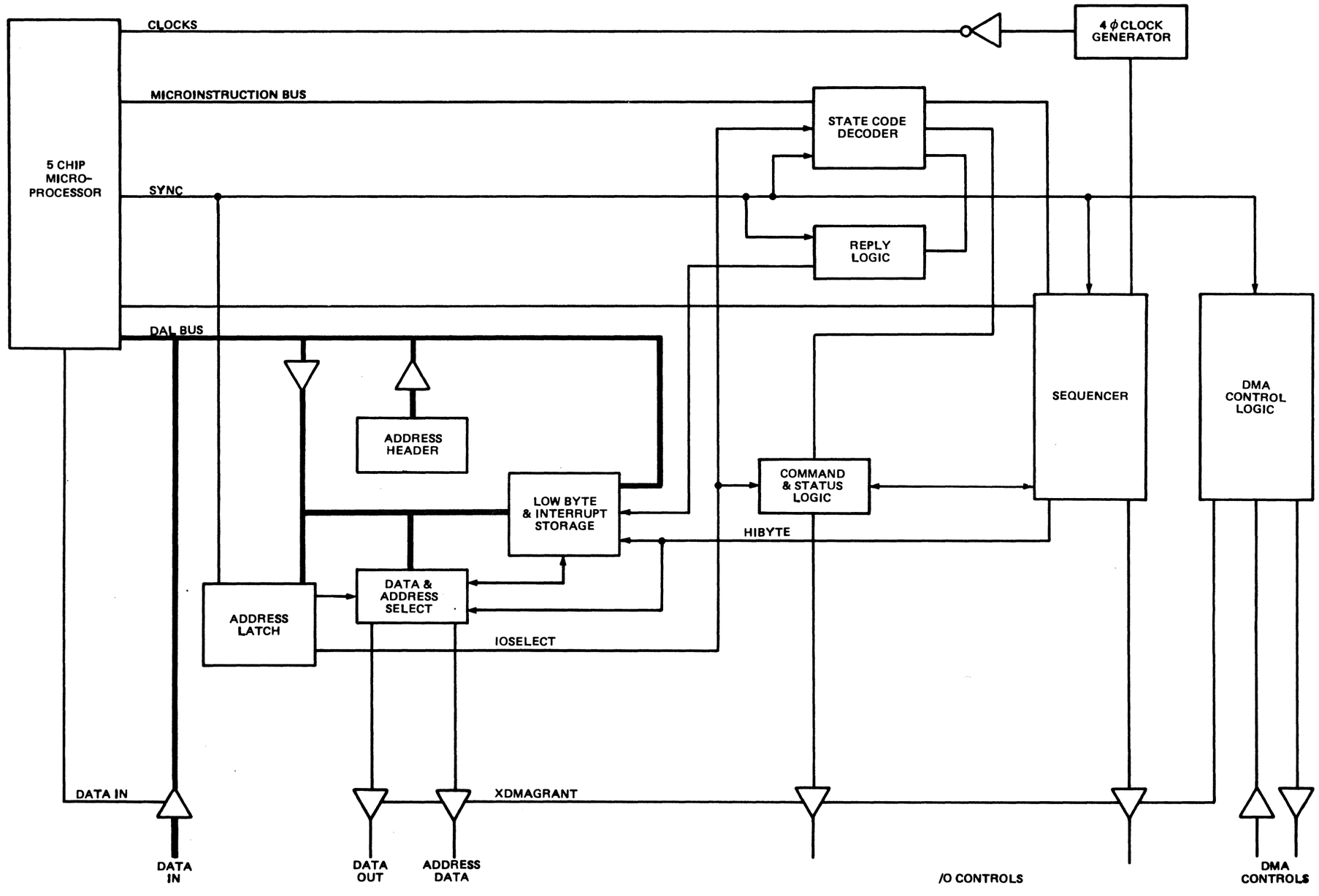
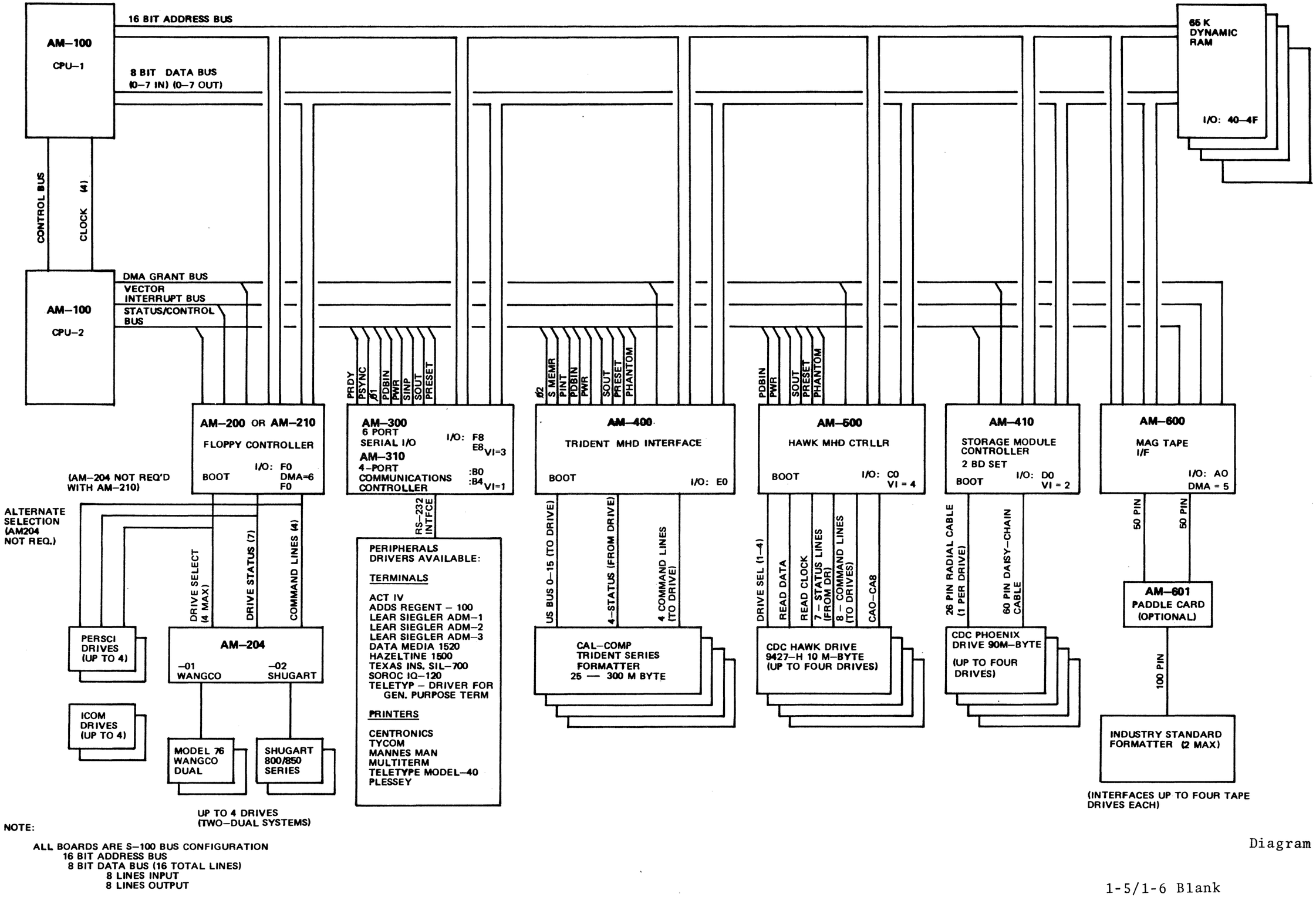


Figure 1-1. AM-100 Simplifie
Block Diagram



NOTE:

ALL BOARDS ARE S–100 BUS CONFIGURATION
16 BIT ADDRESS BUS
8 BIT DATA BUS (16 TOTAL LINES)
8 LINES INPUT
8 LINES OUTPUT

Diagram

SECTION 2
OPERATING DATA

2.0 INTRODUCTION.

This Section contains information on the use of the AM-100 CPU two board set. Capabilities, specifications, interface wiring and user option descriptions are provided for the successful integration of the board into the user's system.

2.1 CAPABILITIES AND SPECIFICATIONS.

This two board set operates from the standard S-100 Bus structure and can be integrated into a complete system. Specifications for the AM-100 are contained in Table 2-1.

Table 2-1. AM-100 Specifications

PARAMETER	SPECIFICATION
Interface type	Standard S-100 Bus, 16-Bit words, byte multiplexed for compatibility to 8-bit peripherals and memories.
Instruction Set	Over 150 standard instructions coded in unique microcode executed on the WD-16 CPU chip set.
Arithmetic Operations	Hardware floating point arithmetic to 11 significant digits.
CPU Architecture	Microprogrammed instruction set; eight 16-bit general purpose registers; floating point hardware unit; special high-speed byte multiplexing logic.

Table 2-1 (Cont.). AM-100 Specifications

PARAMETER	SPECIFICATION
Interrupt Capability	Eight vectored and one non-vectored interrupt lines.
DMA Capability	Seven DMA channels. (Included in vectored interrupt line count.)
Real Time Clock	Standard Feature.
Circuit Boards	Two board set - standard 5" x 10" with 100-pin connectors.

2.2 INTERFACE DESCRIPTION AND WIRING.

The AM-100 CPU interfaces with the standard S-100 Bus structure. All data inputs, outputs, and control signals are transferred through these lines. The S-100 bus connections are made by the bottom edge connectors and are listed in Table 2-2.

Table 2-2. AM-100 Interface Signals

SIGNAL	NAME	J1 PIN NO.
A0	Address 0	79
A1	Address 1	80
A2	Address 2	81
A3	Address 3	31
A4	Address 4	30
A5	Address 5	29
A6	Address 6	82
A7	Address 7	83
A8	Address 8	84
A9	Address 9	34
A10	Address 10	37
A11	Address 11	87
A12	Address 12	33
A13	Address 13	85
A14	Address 14	86
A15	Address 15	32
CLOC	2 MHz Clock	49
DI0	Data Input Bus	95
DI1	Bits 0-7	94
DI2		41
DI3		42
DI4		91
DI5		92
DI6		93
DI7		43

Table 2-2 (Cont.). AM-100 Interface Signals

SIGNAL	NAME	J1 PIN NO.
$\overline{\text{DMAGR0}}$	DMA Grant	63
$\overline{\text{DMAGR1}}$		62
$\overline{\text{DMAGR2}}$		61
$\overline{\text{DMAGR3}}$		60
$\overline{\text{DMAGR4}}$		59
$\overline{\text{DMAGR5}}$		58
$\overline{\text{DMAGR6}}$		57
$\overline{\text{DMAGR7}}$		56
$\overline{\text{DMARCVD}}$	DMA Received	64
D00	Data Out Bus	36
D01	Bits 0-7	35
D02		88
D03		89
D04		38
D05		39
D06		40
D07		90
$\overline{\text{C/CDSB}}$	Not Used	18
$\overline{\text{ADDDSB}}$		19
$\overline{\text{DODSB}}$		22
MWRITE	Memory Write	68
PDBIN	Data Bus In	78
PHLDA	P-Hold Acknowledge	26

Table 2-2 (Cont.). AM-100 Interface Signals

SIGNAL	NAME	J1 PIN NO.
$\overline{\text{PHOLD}}$	DMA Request Line	74
PINTE	CPU Interrupt Enable	28
$\overline{\text{POC}}$	Power On Clear	99
PRDY	Processor Ready	72
$\overline{\text{PRESET}}$	Preset	75
PSYNC	Processor Sync	76
PWAIT	CPU Wait	27
$\overline{\text{PWR}}$	Write Strobe	77
SINP	I/O Input Cycle	46
SINTA	Interrupt Acknowledge	96
SMEMR	Memory Read Cycle	47
SM1	Bus Master OP Code Fetch	44
SOUT	I/O Output Cycle	45
$\overline{\text{SW0}}$	Bus Master Output	97

Table 2-2 (Cont.). AM-100 Interface Signals

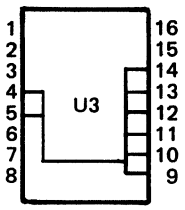
SIGNAL	NAME	J1 PIN NO.
$\overline{VI0}$	Vectored Interrupt 0	4
$\overline{VI1}$	Vectored Interrupt 1	5
$\overline{VI2}$	Vectored Interrupt 2	6
$\overline{VI3}$	Vectored Interrupt 3	7
$\overline{VI4}$	Vectored Interrupt 4	8
$\overline{VI5}$	Vectored Interrupt 5	9
$\overline{VI6}$	Vectored Interrupt 6	10
$\overline{VI7}$	Vectored Interrupt 7	11
$\overline{STVAL}$	Status and Address Valid	25
XRDY	External Ready	3
Ø2	Phase 2 Clock	24

2.3 USER OPTIONS.

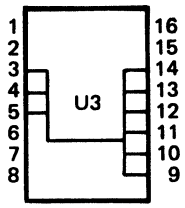
Most of the optional capabilities of the AM-100 CPU are exercised with software, and an overview of the available software capability is contained in Section 3. The hardware option that must be implemented is for the bootstrap Loader program and determines which peripheral the boot loader is accessed from.

The boot loader may be accessed from either the AM-200 circuit board (floppy disk controller), the AM-500 (hard disk controller) or the AM-400 (hard disk interface). Header U3 selects the base address for the boot program by the jumper wires as shown in Figure 2-1.

CPU1 Header U3



AM-200 Floppy Disk
or
AM-400 Trident Disk



AM-500 (Hawk) Hard Disk

Figure 2-1. Header Jumper Wiring

Header U3 on CPU1 can be jumpered for other non-standard jump addresses. The address lines that are jumper selectable are A8-A13 with A14 and A15 always high (1).

To select an address, pins 3, 4, 5, 6, 7 and 8 can be jumpered to ground or left open to generate the desired code. An open (no jumper) selects a one for that address bit and a jumper to pin 14 selects a zero.

NOTE

Pins 9, 10, 11, 12, 13 and 14 on header U3 must always be connected together.

The pins on header that correspond to the various address bits are as follows:

Address line	Pin on U3
8	5
9	4
10	7
11	3
12	8
13	6
GND	14

2.4 INTERRUPT AND DMA OPTIONS.

The AM-100 provides seven levels of DMA or eight levels of interrupt capability. The interrupt and DMA signal lines to the S-100 bus are user selected with jumper wires applied to the circuit board. These jumpers select the desired vectored interrupts or DMA grant signals.

2.5 SYSTEM CONNECTIONS.

The AM-100 board set plugs into two adjacent slots in a S-100 Bus chassis and are connected together by a 40 pin flat cable along the top of the circuit boards. An additional input must be connected to the real time clock of the CPU. This can be supplied from the 50 or 60 Hz power of the low voltage power supply in the S-100 bus chassis. Connect this 50 or 60 Hz signal to CPU board 2 at header U3 pin 12 according to the following procedure as shown in Figure 2-2. The AM-100 is then ready for use.

1. Make sure AC power is off and chassis is unplugged.

2. Locate the secondary transformer tap supplying the +8 volt rectifier. The accompanying schematic illustrates this point for the TEI MCS-122 chassis.

3. Locate the E-Z hook cable supplied with the AM-100 CPU board set.

4. Cut one end off the cable so that it is long enough to reach from the top of the AM-100 CPU set to the transformer tap connection located in Step 2. (Make sure that the cable is long enough to be conveniently routed.)

5. Solder the cut end of the cable to the transformer tap connection.

6. Before attaching the cable to the AM-100, perform the following steps:

   a. Plug in the chassis and turn on AC power.

   b. Observe the waveform at the "E-Z" hook contact. It should appear as a $\pm$ 10 volt signal at 60 or 50 Hz.

7. If the waveform at Step 6b appears correct, turn off AC power, attach the "E-Z" hook to CPU board #2 at J3 pin 4. An extended resistor lead is provided for convenience in hook-up.

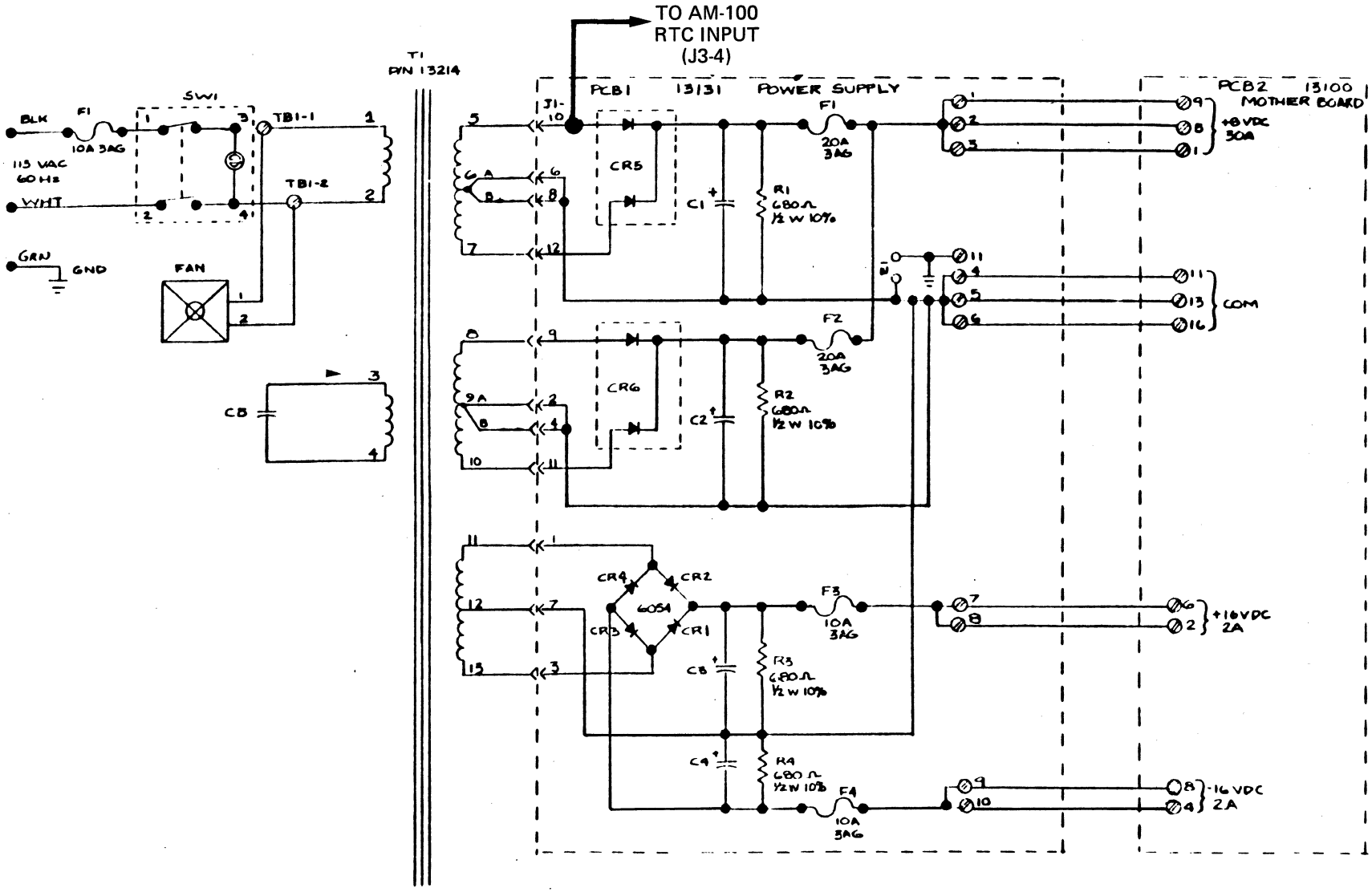


Figure 2-2. Real Time Clock Connection

SECTION 3
PROGRAMMING

3.0 INTRODUCTION.

This section provides information for programming the AM-100 CPU and integrating it into the user's system. The actual software itself is not contained in this manual. Only the requirements and descriptions of the available software programs and their capabilities are presented here.

3.1 SYSTEM CONFIGURATION.

A minimum system can consist of the AM-100 two board set, a minimum of 32K bytes of memory on a S-100 Bus memory board, a serial I/O card such as the AM-300, a disk memory, and a terminal. A PROM is needed to bootstrap the system and this can be located on one of the peripheral interface boards. Other standard Alpha Micro devices include:

1. Floppy disk subsystem (AM-200/AM-210).
2. Ten megabyte hard disk (AM-500).
3. Large storage capacity hard disk (AM-400/AM-410).
4. Tape storage subsystem (AM-600).

To utilize other devices in the system, it is necessary to write a driver program for the device and place the resulting module in a preassigned disk area. The bootstrap PROM can be located on either the floppy controller interface board (AM-200) or the hard disk controller interface board (AM-500 or AM-400). The starting address for the bootstrap program is jumper selectable on a platform header on CPU Board 1, U3.

The standard addresses used in Alpha Micro systems are listed in Table 3-1 and boot addresses are listed in Table 3-2. The standard DMA levels are listed in Table 3-4 and the standard interrupt levels are listed in Table 3-5. For further details on interfacing Alpha Micro circuit boards, consult the Technical manual for the individual board.

Table 3-1. Alpha Micro I/O Addresses

Address (Hex)	Interface
00 - 0F	Reserved
10 - 3B	Unassigned
3C - 3F	Reserved
40 - 4F	Memory Bank Switching (1 port/memory bd)
50 - 6F	Unassigned
70 - 7F	Phone Link (DC Hayes bd)
80 - 83	Imsai PIO (parallel port, Data I/O control, etc.)
84 - 9F	Unassigned
A0 - A7	AM-600 Mag Tape I/F
A8 - AF	Unassigned
B0 - BF	AM-310 4 Port Communications Bd (4 ports required/bd)
C0 - C7	ICOM Floppy Controller or AM-500 Disk Controller (4 ports/bd)
C8 - CF	Unassigned
C2	Unassigned

Table 3-1 (Cont.). Alpha Micro I/O Addresses

Address (Hex)	Interface
D0 - D3	AM-410 Disk Controller
D4 - DF	Unassigned
E0 - E7	AM-400 Trident Formatter I/F
E8 - EF	AM-300 Alternate (2nd board)
F0 - F7	AM-200 Floppy Controller or AM-210 Floppy Controller
F8 - FF	AM-300 6 Port Serial I/O Board

Table 3-2. Boot Addresses

Address (Hex)	Boot Location
F400	AM-500 Boot Address AM-410 Boot Address
FC00	AM-200 Boot Address AM-210 Boot Address AM-400 Boot address
C000	ICOM Floppy Boot Address

TABLE 3-3. DMA Levels

Level	Device
0	
1	
2	
3	
4	
5	AM-600 Mag Tape Controller
6	AM-200 Floppy Controller
7	

Table 3-4. Interrupt Levels

Level	Device
0	
1	AM-310 4 Port Communications Board
2	AM-410 Disk Controller
3	AM-300 6 Port Serial I/O
4	AM-500 Disk Controller
5 6	Not available on old systems
7	

3.2 SOFTWARE OVERVIEW.

A wide variety of software programs are available for use in the AM-100 system that have been operating in the commercial environment for many years. These programs have been written under copyright solely by Alpha Microsystems and are available in either floppy disk or hard disk form.

3.2.1 OPERATING SYSTEM.

The operating system is a commercial, full multi-tasking, timesharing system capable of supporting as many job partitions as memory capacity will allow. Each job may be controlled by its own terminal or several jobs may be controlled by the same user terminal under a unique software control system built into the monitor. One job may also control several terminals. The status of each job may be optionally displayed on a central video display interfaced through a controller.

The I/O structure is fully device independent and contained within the monitor. To incorporate a new device into the system, the user creates a software driver to interface to the device and includes it in the monitor.

The terminal service routines are also device independent and operate through terminal drivers. Each terminal driver contains a software translation routine to allow the use of special terminal functions such as cursor address and common I/O routines to all types of terminals, without being dependent on which brand of terminal is attached to the system.

3.2.2 ASSEMBLY LANGUAGE PROGRAM.

The assembly language program development system includes:

1. A multi-phase macro assembler.
2. A linking loader.
3. A symbolic debugger.

3.2.3 TEXT EDITORS.

Two text editor programs are included:

1. A character oriented editor.
2. A cursor-controlled screen editor.

3.2.4 UTILITY AND SUPPORT PROGRAMS.

Utility and support programs are available that include:

1. File maintenance programs.
2. Dump programs.
3. Memory test programs.
4. Device test programs.
5. Line printer spooler system.
6. Dynamic system status monitor programs.
7. A system generation procedure that allows custom tailoring of the monitor to individual user requirements.
8. Text formatting.

3.2.5 ALPHABASIC PROGRAM.

The AlphaBasic programming language processor is a full compiler that is disk oriented and supports:

1. Strings.
2. Multi-dimensioned arrays.
3. Disk I/O file accessing.
4. A unique variable mapping system for file manipulation in applications programs and for assembly language subroutine processing.
5. Assembly language subroutines.
6. Terminal independent cursor control.

The AM-100 utilizes floating point hardware so it typically runs several times faster than other Basic systems. It is precise to eleven decimal digits using a three-word binary floating-point format which is identical to that used by the WD16 instruction set.

3.2.6 ACCOUNTING PACKAGE.

An accounting package is available under special license that includes:

1. General ledger.
2. Accounts receivable.
3. Accounts payable.
4. Order entry-inventory control.
5. Payroll.

Each of these is a fully interactive, menu-driven, complete system by itself. However, interface is provided between all five modules to create a totally integrated accounting package.

3.2.7 ALPHALISP.

An Alpha Micro version of the programming language LISP is available. It is designed for users that require a language that is both a formal mathematical language and (with extensions) a convenient programming language.

3.2.8 ALPHAPASCAL.

ALPHAPASCAL is a systematic procedure-oriented, structured programming language that is available. Adapted from the UCSD/PASCAL system, ALPHAPASCAL utilizes a dynamic paging system that allows the system to run in a small amount of memory.

SECTION 4
FUNCTIONAL THEORY OF OPERATION

4.0 INTRODUCTION.

This section describes in detail the functional theory of operation of the AM-100 Central Processor Unit (CPU). The first part of this section provides a general description of the configuration of the CPU and its integration into an S-100 Bus system. The second part describes the details of CPU operation and how the CPU chip set operates with its associated logic elements. The third part provides a complete description of the CPU chip set and the fourth part describes the operation of the individual logic elements that support the CPU chip set.

4.1 CPU CONFIGRUATION.

The AM-100 circuit cards function as a 16-bit CPU that plugs in to an S-100 Bus system. The data processing logic is contained on a two-board set that is fully integrated into an S-100 Bus system.

4.1.1 AM-100 TWO-BOARD SET.

The AM-100 CPU is packaged on two circuit boards connected together by a 40-conductor flat ribbon cable. Both boards mate with the standard 100 pin connectors in the S-100 Bus system.

One Circuit board, CPU No. 1, contains the CPU chip set that consists of five 40-pin MOS/LSI chips. This chip set contains the necessary data processing capability for CPU operation. The second circuit board, CPU No. 2, contains the logic that interfaces the 16-bit bus of the CPU to the 8-bit S-100 bus plus the DMA and interrupt logic.

The CPU chip set contains six 16-bit accumulators/index registers, a stack pointer and program counter, eight addressing modes, hardware SAVE and RESTORE, and hardware floating point arithmetic.

4.1.2 S-100 BUS OPERATION.

The S-100 Bus system is the single bus computer architecture that supports the AM-100 CPU and its associated memory and peripherals. The I/O connectors are standard 100-pin type and each of the 100 lines are bussed together throughout the computer chassis.

The S-100 bus is an eight-bit format with eight bits of tri-state data for CPU input, eight for CPU output, 16 bits for address, and various control, status, and utility lines. There are also eight lines for vectored interrupts/DMA requests and eight for DMA Grant lines. The various devices connected to the bus are referred to as bus master and bus slave depending on the operational configuration taking place. The CPU is not the only device that can control the bus. Any DMA device has the capability of controlling the bus as a bus master.

The command and control lines determine the timing of the bus and are listed in Table 4-1. The status lines are associated with the address lines and indicate what type of bus cycle is taking place. Any bus master must generate these signals or at least ensure that the unasserted level is maintained. The AM-100 status lines are listed in Table 4-2. The utility lines are clocks, power, and initialization and are listed in Table 4-3.

Table 4-1. S-100 Bus Command/Control Lines

SIGNAL	NAME
PSYNC	Processor Sync
PDBIN	Data Bus In
$\overline{\text{PWR}}$	Write Strobe
PHLDA	P-Hold Acknowledge
$\overline{\text{PHOLD}}$	Processor Hold Request Line

Table 4-2. S-100 Bus Status Lines

SIGNAL	NAME
SM1	Bus Master OP Code Fetch
SOUT	I/O Output Cycle
SINP	I/O Input Cycle
SMEMR	Memory Read Cycle
$\overline{\text{SWO}}$	Bus Master Output
SINTA	Interrupt Acknowledge

Table 4-3. S-100 Bus Utility Lines

SIGNAL	NAME
Power	GND, +8VDC, +16VDC, -16VDC
Ø2	2 MHz Phase 2 Clock
$\overline{\text{CLOC}}$	2 MHz Clock
MWRITE	Memory Write
$\overline{\text{PRESET}}$	Preset
$\overline{\text{POC}}$	Power-On Clear

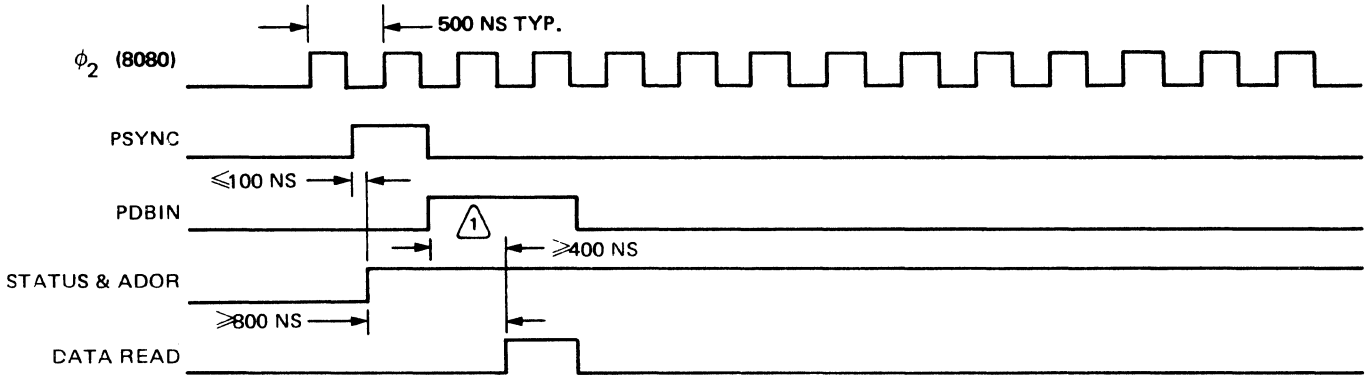
Read Timing. The timing of the signals involved with a READ BYTE sequence on the S-100 bus is shown in Figure 4-1. When an I/O operation takes place, PSYNC is generated as the first state of the bus cycle. Signal PDBIN is next to indicate that a read operation is taking place. Address lines are valid shortly after PSYNC and remain so until the next bus operation. The addressed memory or peripheral has 400ns to place its data on the bus for the read cycle. Once the CPU reads the data, PDBIN is lowered, ending the cycle. The READ WORD cycle is a similar sequence twice because two bytes are read. Timing for the READ WORD sequence is also shown in Figure 4-1.

Write Timing. The timing of the signals involved with a WRITE BYTE sequence on the S-100 bus is shown in Figure 4-2. With this sequence, PSYNC starts the bus cycle like the read timing. Signal PWR occurs next to indicate that there is valid data on the bus that is to be written into the addressed location. Status and address data is placed on the bus $\leq$ 100 nsec after PSYNC by the bus master. The WRITE WORD cycle is a similar sequence twice because two bytes are written as shown in Figure 4-2.

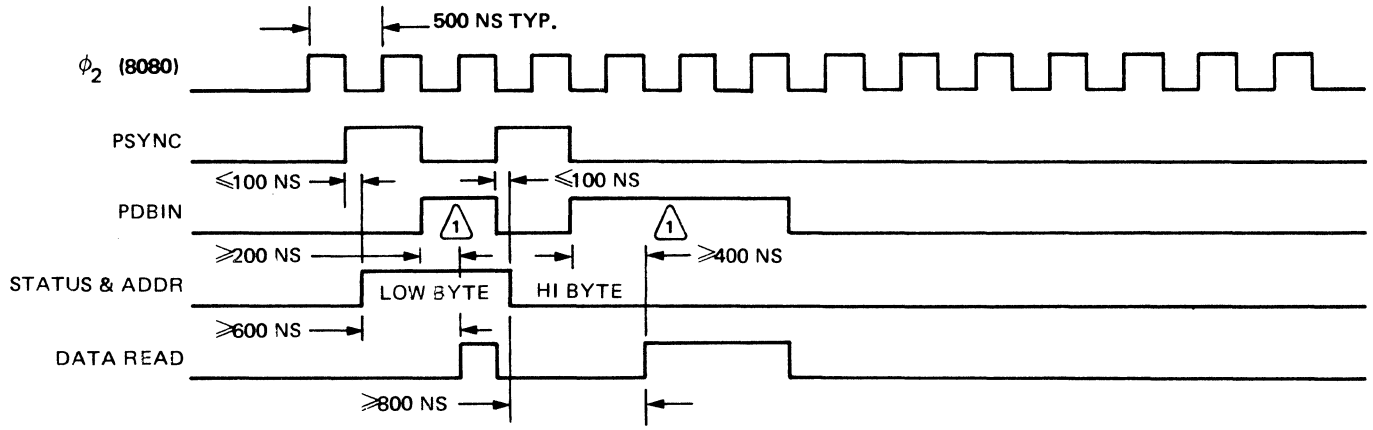
Read-Modify-Write Timing. The READ-MODIFY-WRITE cycle combines these operations into a single sequence as is shown in Figure 4-3.

DMA Timing. The timing and sequence for a DMA cycle is shown in Figure 4-4. The selected vector interrupt signal initiates this cycle and it is acknowledged by DMARCVD. The selected DMAGRANT signal indicates an active DMA cycle and the busses are active $\leq$ 50ns later.

READ BYTE TIMING



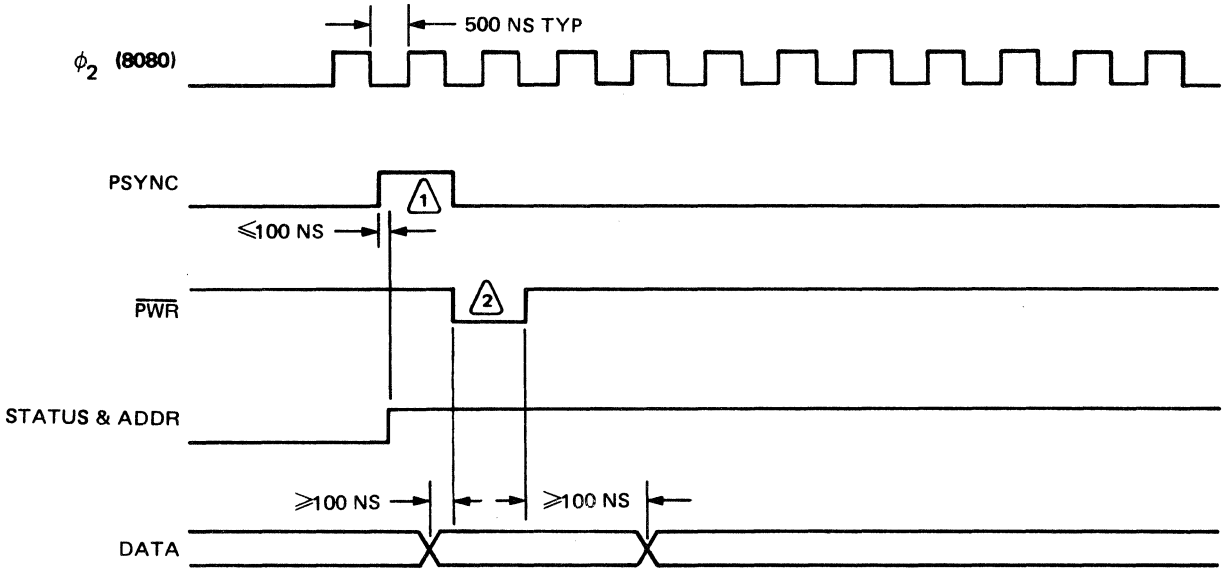
READ WORD TIMING



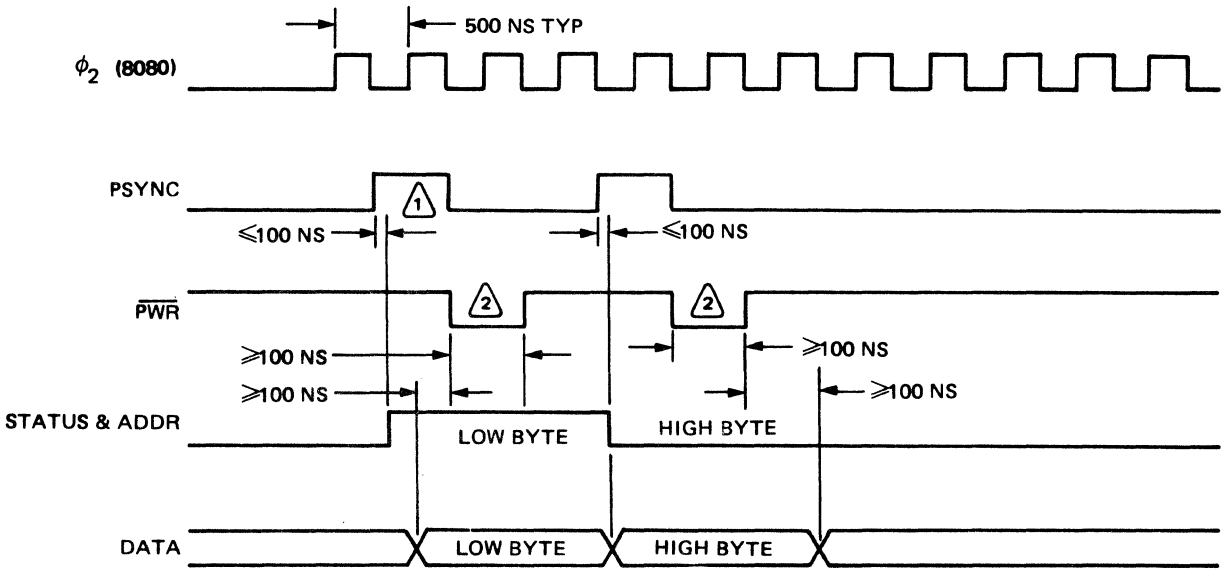
1 PDBIN MAY BE STRETCHED BY CONTROLLING PRDY. PRDY IS EXAMINED AT ϕ1 (WD) LEADING EDGE TIME DURING PDBIN.

Figure 4-1. S-100 Bus Read Timing

WRITE BYTE TIMING



WRITE WORD TIMING

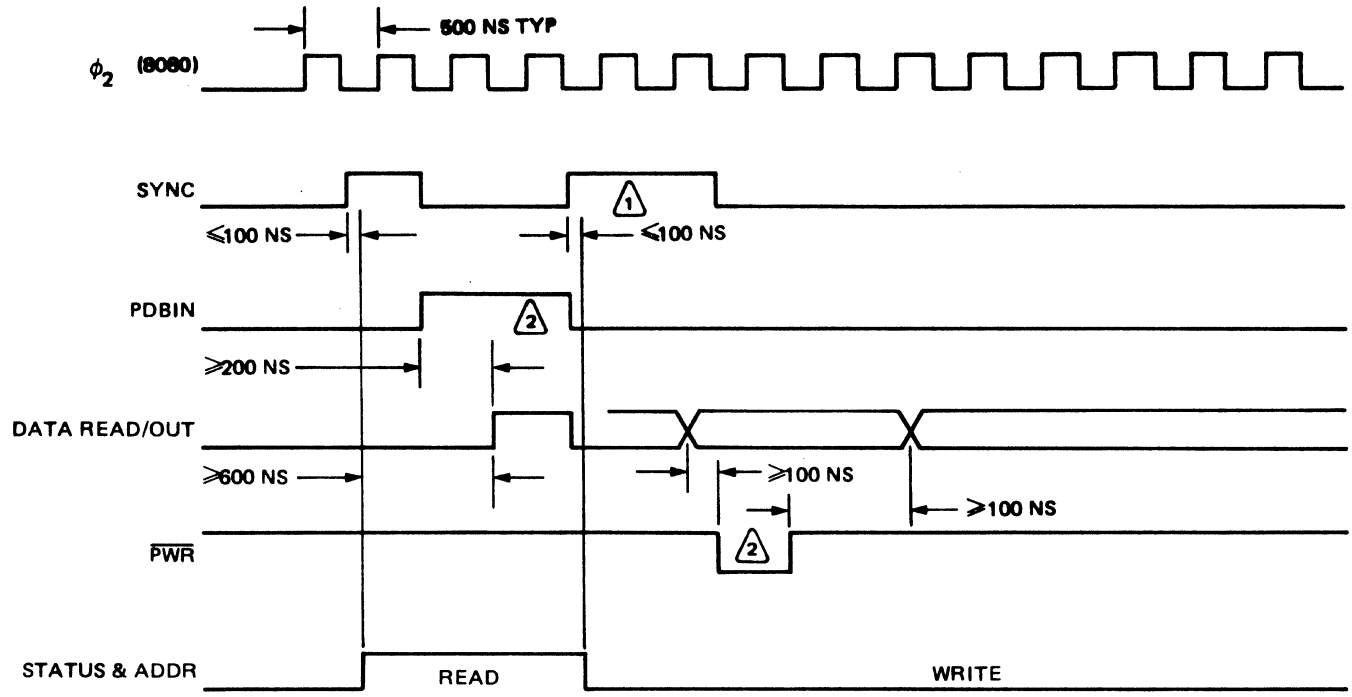


1 PSYNC MAY BE STRETCHED ONE OR MORE CYCLES DUE TO DOUT DELAY CAUSED BY MICROCODE.

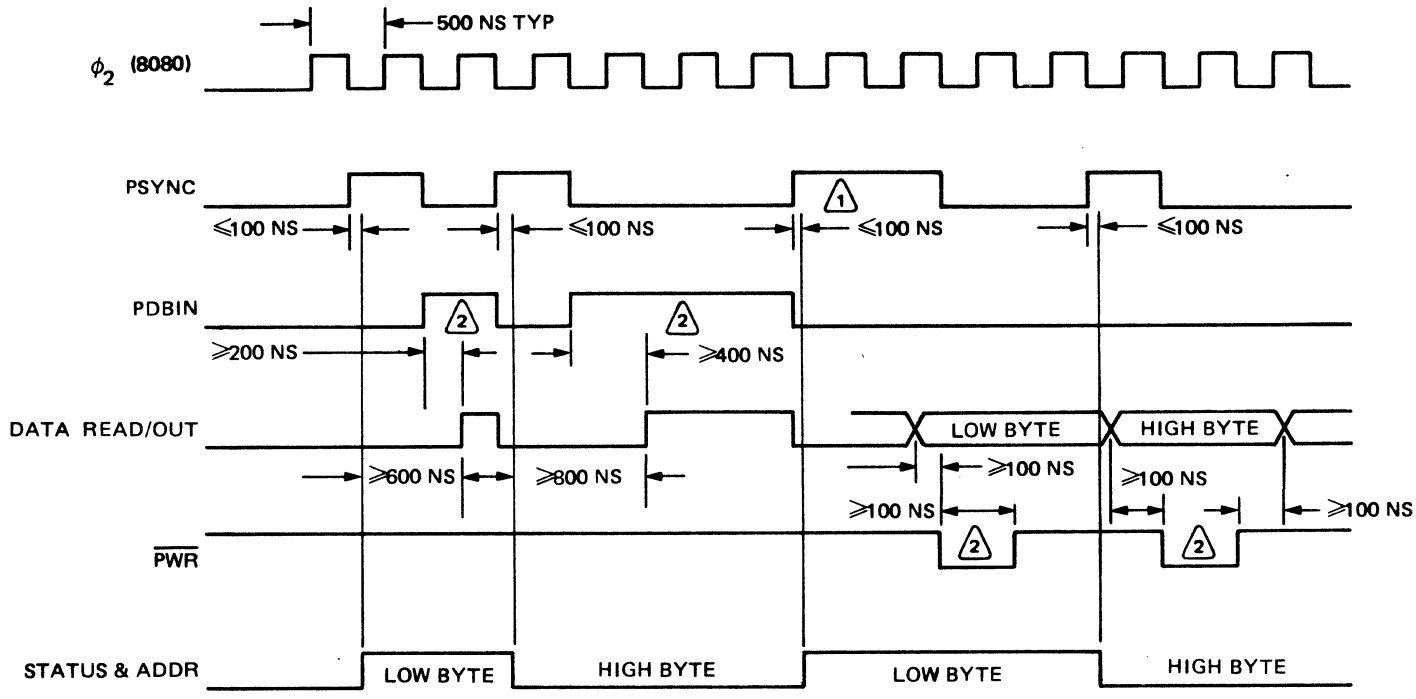
2 $\overline{\text{PWR}}$ MAY BE STRETCHED BY CONTROLLING PRDY. PRDY IS EXAMINED AT ϕ1 (WD) LEADING EDGE TIME DURING PWR.

Figure 4-2. S-100 Write Timing

READ-MODIFY-WRITE BYTE TIMING



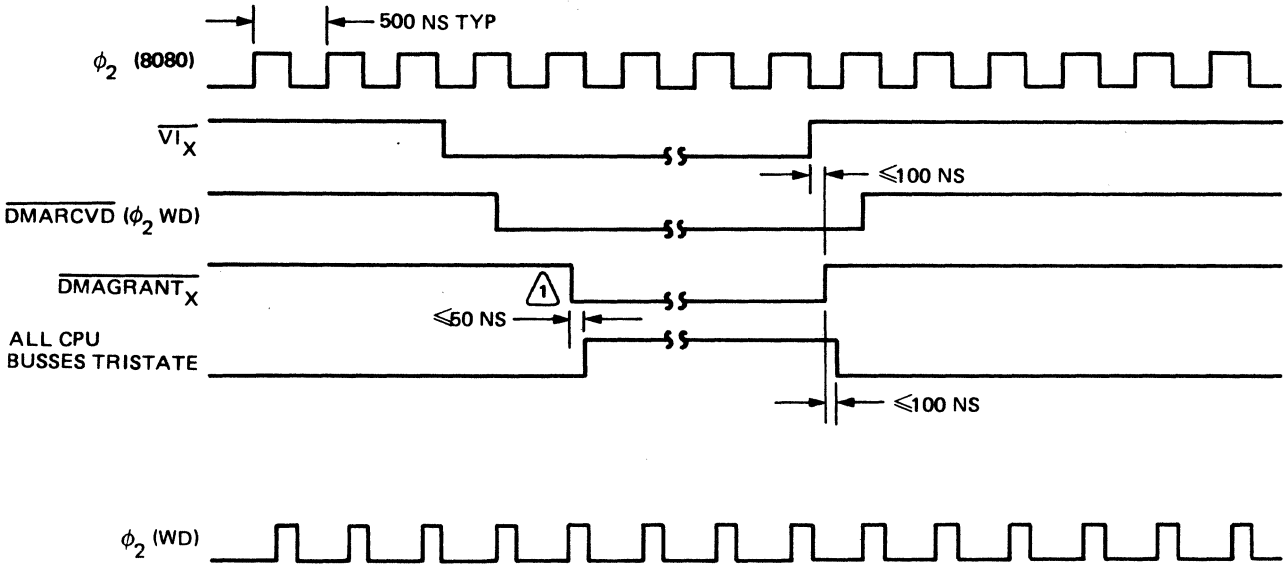
READ-MODIFY-WRITE WORD TIMING



1. PSYNC MAY BE STRETCHED ONE OR MORE CYCLES DUE TO DOUT DELAY CAUSED BY MICROCODE.
2. PDBIN AND $\overline{\text{PWR}}$ MAY BE STRETCHED BY CONTROLLING PRDY. PRDY IS EXAMINED AT $\phi 1$ (WD) LEADING EDGE TIME DURING PDBIN OR PWR.

Figure 4-3. S-100 Bus Read-Modify-Write Timing

DMA TIMING



1. $\overline{DMAGRANT}_X$ MAY BE DELAYED IF A BUS CYCLE IS IN PROGRESS WHEN A DMA REQUEST IS GENERATED. IT WILL BE ISSUED IMMEDIATELY FOLLOWING THE COMPLETION OF SYNC.
2. $\overline{PHOLD}$ IS ALWAYS FORCED LOW.
3. PHLDA IS ALWAYS HIGH.

Figure 4-4. S-100 Bus DMA Timing

For a complete description of the S-100 bus signals in the AM-100 system, refer to Table 4-4. This table lists all signals in alphabetical order with AM-100 pin numbers and also the sheet number of the schematic where they interface with the AM-100 CPU. A complete description of each signal is also given.

Table 4-4. S-100 Bus Interface Signals List

SIGNAL	NAME	J1 PIN NO.	SCHEM PAGE	DESCRIPTION
A0	Address 0	79	CPU1-9	16 bits of tri-state addressing.
A1	Address 1	80	CPU1-9	
A2	Address 2	81	CPU1-9	
A3	Address 3	31	CPU1-9	
A4	Address 4	30	CPU1-9	
A5	Address 5	29	CPU1-9	
A6	Address 6	82	CPU1-9	
A7	Address 7	83	CPU1-9	
A8	Address 8	84	CPU1-10	
A9	Address 9	34	CPU1-10	
A10	Address 10	37	CPU1-10	
A11	Address 11	87	CPU1-10	
A12	Address 12	33	CPU1-10	
A13	Address 13	85	CPU1-10	
A14	Address 14	86	CPU1-10	
A15	Address 15	32	CPU1-10	
$\overline{\text{CLOC}}$	2 MHz Clock	49	CPU2-3	2 MHz clock from same source as Ø2.

Table 4-4 (Cont.). S-100 Bus Interface Signals List

SIGNAL	NAME	J1 PIN NO.	SCHEM PAGE	DESCRIPTION
DI0	Data Input Bus	95	CPU1-6	Data input port. Eight bit tri-state data bus
DI1	Bits 0-7.	94	CPU1-6	from bus slave to bus master.
DI2		41	CPU1-6	
DI3		42	CPU1-6	
DI4		91	CPU1-6	
DI5		92	CPU1-6	
DI6		93	CPU1-6	
DI7		43	CPU1-6	
$\overline{\text{DMAGR0}}$	DMA Grant	63	CPU2-5	Grant signal issued to the highest priority
$\overline{\text{DMAGR1}}$		62	CPU2-5	controller that has requested the bus.
$\overline{\text{DMAGR2}}$		61	CPU2-5	
$\overline{\text{DMAGR3}}$		60	CPU2-5	
$\overline{\text{DMAGR4}}$		59	CPU2-5	
$\overline{\text{DMAGR5}}$		58	CPU2-5	
$\overline{\text{DMAGR6}}$		57	CPU2-5	
$\overline{\text{DMAGR7}}$		56	CPU2-5	
$\overline{\text{DMARCVD}}$	DMA Received	64	CPU2-5	CPU response to DMA request indicating that a DMA exchange is in process. No other DMA controllers may issue a DMA request while $\overline{\text{DMARCVD}}$ is asserted.

Table 4-4 (Cont.). S-100 Bus Interface Signals List

SIGNAL	NAME	J1 PIN NO.	SCHEM PAGE	DESCRIPTION
D00 D01 D02 D03 D04 D05 D06 D07	Data Out Bus Bits 0-7	36 35 88 89 38 39 40 90	CPU1-8 CPU1-8 CPU1-8 CPU1-8 CPU1-8 CPU1-8 CPU1-8 CPU1-8	Data output port. Eight bit tri-state data bus from bus master to bus slave.
ENDF		18 19 22	CPU2-2	Only used internal to the AM-100
MWRITE	Memory Write	68	CPU2-2	Gated combination of PWR and $\overline{\text{SOUT}}$.
PDBIN	Data Bus In	78	CPU2-9	Read Enable. Used by bus master to request address slave to place data on the data bus.
PHLDA	P-Hold	26	CPU2-3	When asserted, indicates that the CPU is releasing control of the bus in response to a DMA request ($\overline{\text{PHOLD}}$). (This signal is forced high by the AM-100 and not used.)

Table 4-4 (Cont.). S-100 Bus Interface Signals List

SIGNAL	NAME	J1 PIN NO.	SCHEM PAGE	DESCRIPTION
$\overline{\text{PHOLD}}$	DMA Request	74	CPU2-2	Used by DMA controllers to request bus mastership. (This signal is held low by the AM-100 and not used.)
PINTE	CPU Interrupt Enable	28	CPU2-7	Output from CPU to enable slave interrupts (not used by Alpha Micro Systems).
$\overline{\text{POC}}$	Power On Clear	99	CPU2-3	Clear signal generated by the CPU on initial turn on.
PRDY	Processor Ready	72	CPU2-8	When low, increases the duration of PDBIN or $\overline{\text{PWR}}$.
$\overline{\text{PRESET}}$	Preset	75	CPU2-2	Reset signal normally originating from front panel reset pushbutton.
PSYNC	Processor Sync	76	CPU2-7	When asserted, indicates the start of a bus cycle.
PWAIT	CPU Wait	27	CPU2-7	When asserted, indicates that the CPU is in a wait period.

Table 4-4 (Cont.). S-100 Bus Interface Signals List

SIGNAL	NAME	J1 PIN NO.	SCHEM PAGE	DESCRIPTION
$\overline{\text{PWR}}$	Write Strobe	77	CPU2-7	When asserted, is a command from the bus master for the addressed slave to accept the data on the data bus.
SINP	I/O Input Cycle	46	CPU2-7	When asserted, indicates that the current bus cycle is a bus master input from an I/O address.
SINTA	Interrupt Acknowledge	96	CPU2-7	When asserted, indicates acceptance by bus master of an interrupt request.
SMEMR	Memory Read Cycle	47	CPU2-7	When asserted, indicates that the current bus cycle is a bus master input from a memory address.
SM1	Bus Master OP Code Fetch	44	CPU2-7	When asserted, indicates that the current bus cycle is a bus master OP code fetch.
SOUT	I/O Output Cycle	45	CPU2-7	When asserted, indicates that the current bus cycle is a bus master output to an I/O address.
$\overline{\text{SWO}}$	Bus Master Output	97	CPU2-7	When asserted, indicates that the current bus cycle is a bus master output.

Table 4-4 (Cont.). S-100 Bus Interface Signals List

SIGNAL	NAME	J1 PIN NO.	SCHEM PAGE	DESCRIPTION
$\overline{VI0}$	Vectored Interrupt 0	4	CPU2-5	Vectored interrupt lines used for both interrupt requests and DMA requests.
$\overline{VI1}$	Vectored Interrupt 1	5	CPU2-5	
$\overline{VI2}$	Vectored Interrupt 2	6	CPU2-5	
$\overline{VI3}$	Vectored Interrupt 3	7	CPU2-5	
$\overline{VI4}$	Vectored Interrupt 4	8	CPU2-5	
$\overline{VI5}$	Vectored Interrupt 5	9	CPU2-5	
$\overline{VI6}$	Vectored Interrupt 6	10	CPU2-5	
$\overline{VI7}$	Vectored Interrupt 7	11	CPU2-6	
$\overline{STVAL}$	Status Valid	25	CPU2-3	Indicates Status & Address lines valid during PSYNC.
XRDY	External Ready	3	CPU2-8	When low, increases the duration of PDBIN or $\overline{PWR}$.
Ø2	Phase 2 Clock	24	CPU2-3	2 MHz clock Phase 2. Master Timing signal for the bus.
+8V	+8vdc power	1, 51	CPU1-1	System power and ground.
		1, 51	CPU2-1	
+16V	+16 vdc power	2	CPU1-1	
-16V	-16 vdc power	52	CPU1-1	
GND	Gnd	50,	CPU1-1	
		100	CPU2-1	

4.1.3 CPU CHIP SET.

The CPU chip set consists of five 40-pin MOS/LSI chips that include a data chip, a control chip, and three microms. A detailed description of the operation of the individual chips is contained in paragraph 4.3. The chip set utilizes 16-bit architecture with both word and byte operation. Since the S-100 Bus is an eight-bit system, further data processing is required for compatibility and this is described in paragraph 4.2. The primary features of the chip are as follows:

a. 16-bit architecture with both word and byte operation.
b. 16-bit data access port to memory and I/O.
c. Eight 16-bit registers.
d. Four external interrupts.
e. Three internal interrupts.
f. Eight addressing modes.

The chip set is microprogrammable by the coded data within the microms. Figure 4-5 contains a block diagram of the five chips in the CPU chip set.

4.1.3.1 DATA CHIP.

The data chip provides the interfacing capability for data and addressing. A 16-bit Data and Address bus (DAL) interfaces the CPU to external logic. It also contains eight 16-bit registers, the Arithmetic Logic Unit (ALU), microinstruction bus lines, and some microinstruction decoding logic.

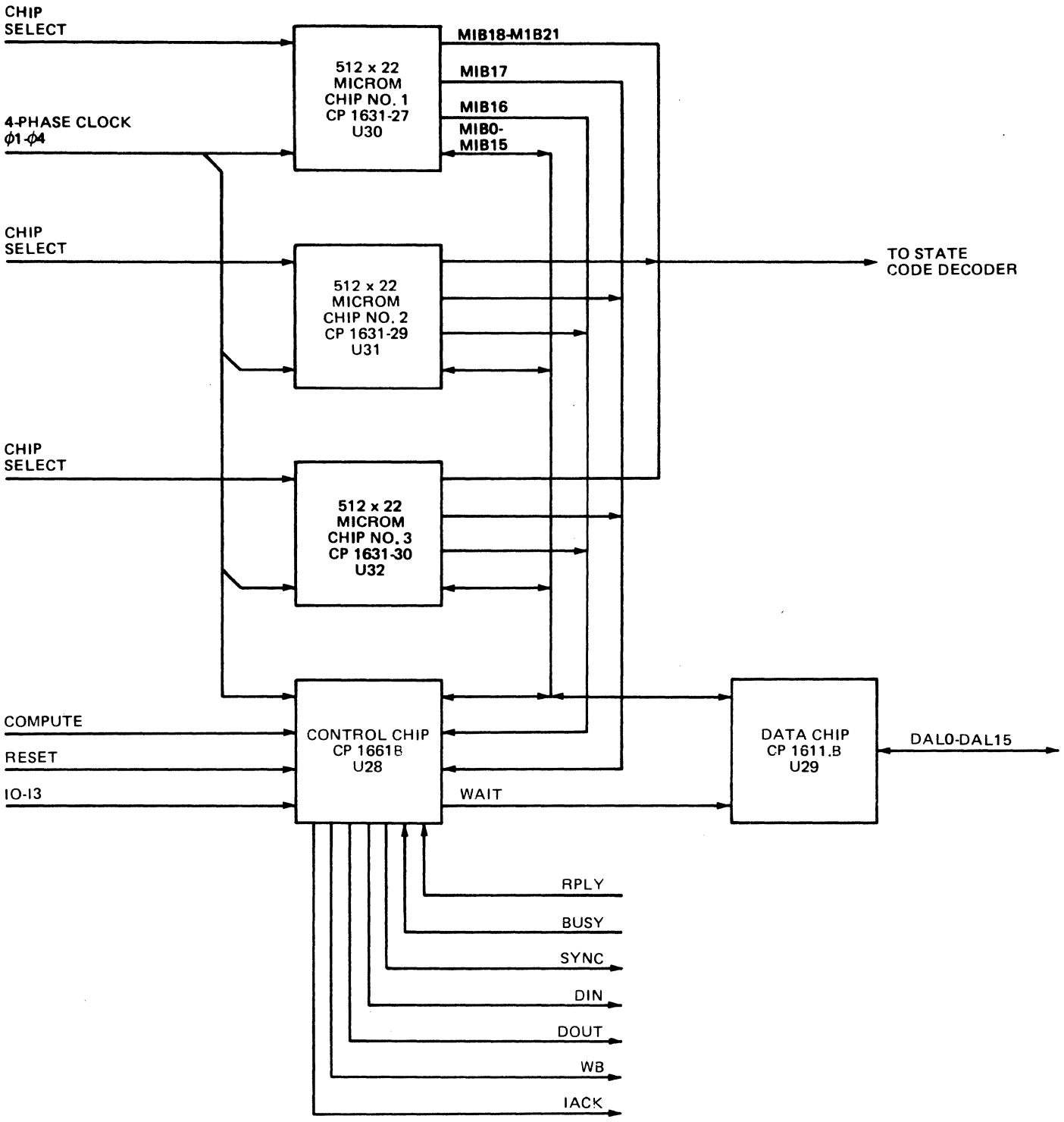


Figure 4-5. CPU Chip Set

4.1.3.2 CONTROL CHIP.

The control chip contains the microinstruction address register which controls accessing of the microinstructions. It also contains translation array logic and the interrupt logic. The master timing control is directed by the control chip which indicates when an address is valid on the DAL lines and indicates a read or write operation and a byte or word operation.

4.1.3.3 MICROM CHIPS.

The microm chips are 512 word by 22 bit ROMs which are programmed to Alpha Micro specifications. These 22-bit outputs comprise the microinstruction bus (MIB) that allows the chips in the CPU to communicate with each other (MIB00-MIB21). Each ROM location contains a 16-bit microinstruction, two control bits, and four auxiliary bits. The 16-bit microinstructions and the two control bits form a high impedance tri-state bus. Bits MIB18-MIB21 are the auxiliary bits and are the state codes that direct external logic to perform special functions.

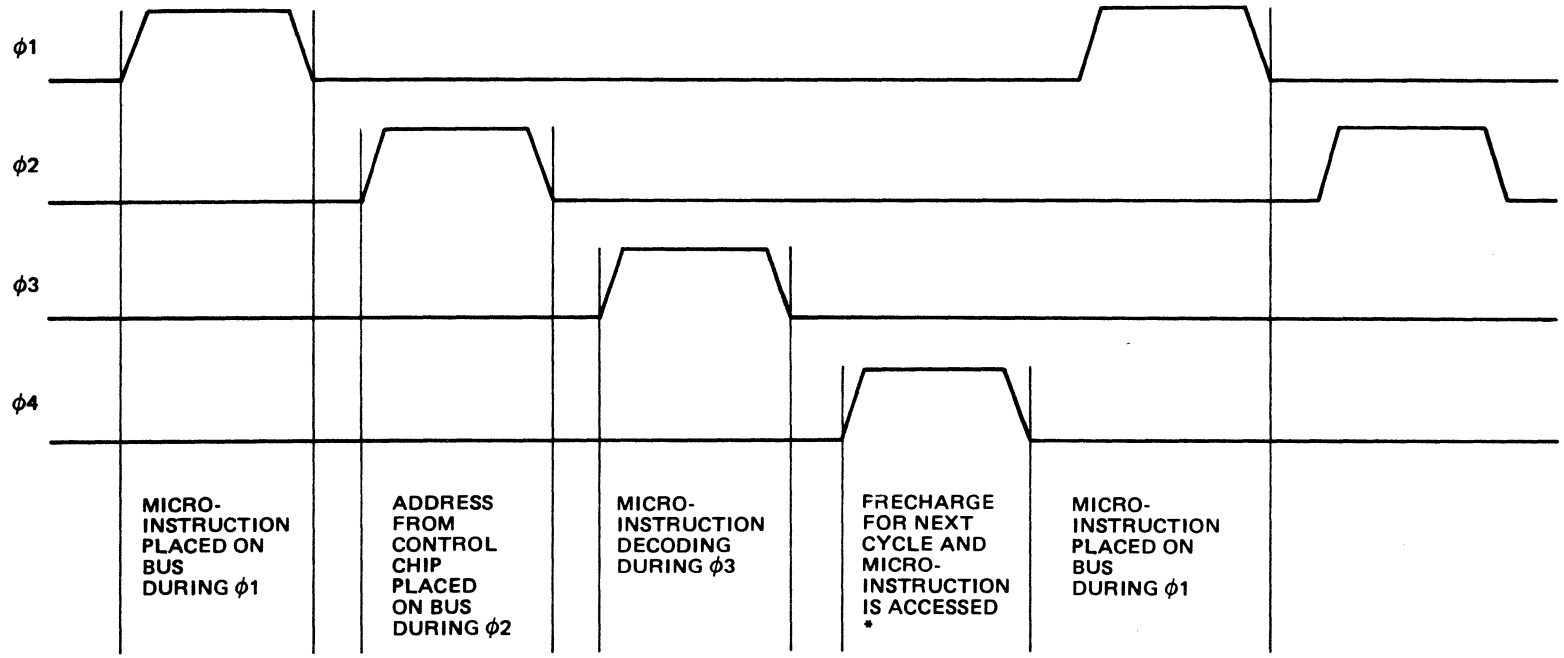
4.1.3.4 MICROINSTRUCTION BUS.

The Microinstruction Bus (MIB) is a high impedance MOS bus for data transmission between the chips in the CPU and is very sensitive to external monitoring even with oscilloscope probes. The capacitors located on CPU 1 suppress noise and optimize data transmission on the bus.

Bits MIB00-MIB10 serve a dual purpose. During the portion of the cycle that addresses the microinstruction from the microm, these bits contain the 11-bit address. During another part of the cycle, they are used to transmit the microinstruction to the other chips. Bits MIB11-MIB17 just transfer part of the microinstruction. Therefore, during the address phase, MIB00-MIB10 address the microinstruction and during another phase MIB00-MIB17 transfer the microinstruction.

Timing of the microinstruction bus is shown in Figure 4-6. The four phases are running continuously, and phase two places the address of the next required microinstruction on the MIB in bits 0-10. This address data comes from the control chip. Phase three decodes the microinstruction. Phase four provides the precharge necessary for the nodes since this is a MOS bus. At phase 1, the microinstruction is placed on the bus and the process repeats.

The function of the state codes and operation of the CPU with its associated external logic is described in paragraph 4.2.



*Except for MIB15 which is precharged at Ø3. It is used to transfer conditional jump results back to the control chip during Ø4.

MIB16 is precharged during Ø2 and Ø4. It is conditionally discharged during Ø1 to control RR register, and conditionally discharged during Ø3 to disable microm output buffers.

Figure 4-6. Microinstruction Bus Timing

4.2 CPU OPERATIONS.

The AM-100 CPU board set consists of the CPU chip set and associated logic elements that provide data processing capability for the S-100 bus system. This Section describes operation of the CPU in the system and operation of the CPU chip set with its associated logic elements.

An overall block diagram of the CPU board set is shown in Figure 4-7. Sheet 1 contains all the logic on CPU board 1, and sheet 2 contains all the logic on board 2. Table 4-5 contains a complete list of the signals in the AM-100 CPU. The list is alphabetical by signal mnemonic, and a description is given for the function of each signal and the location where it can be found on the schematic diagram.

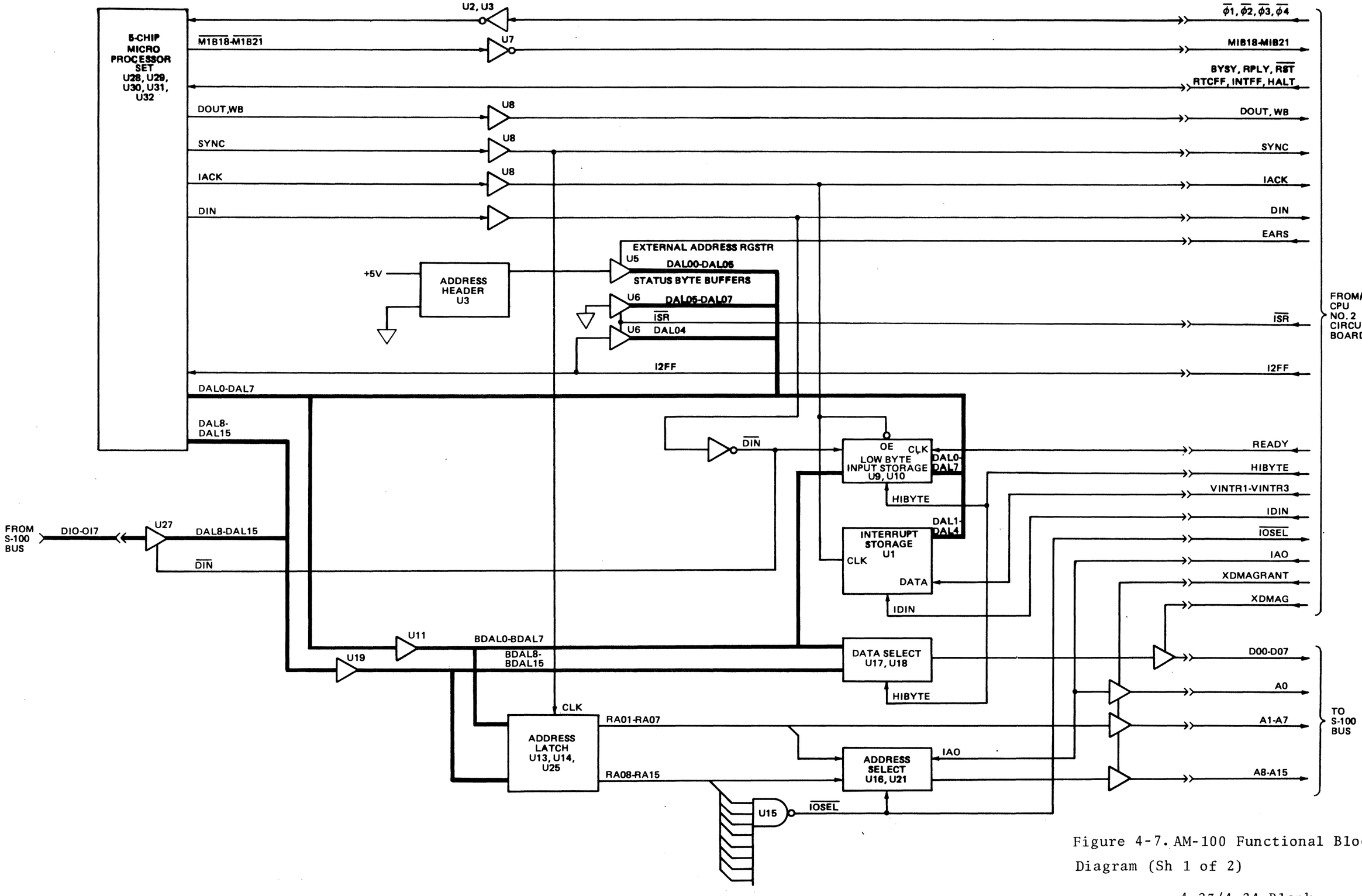


Figure 4-7. AM-100 Functional Block Diagram (Sh 1 of 2)

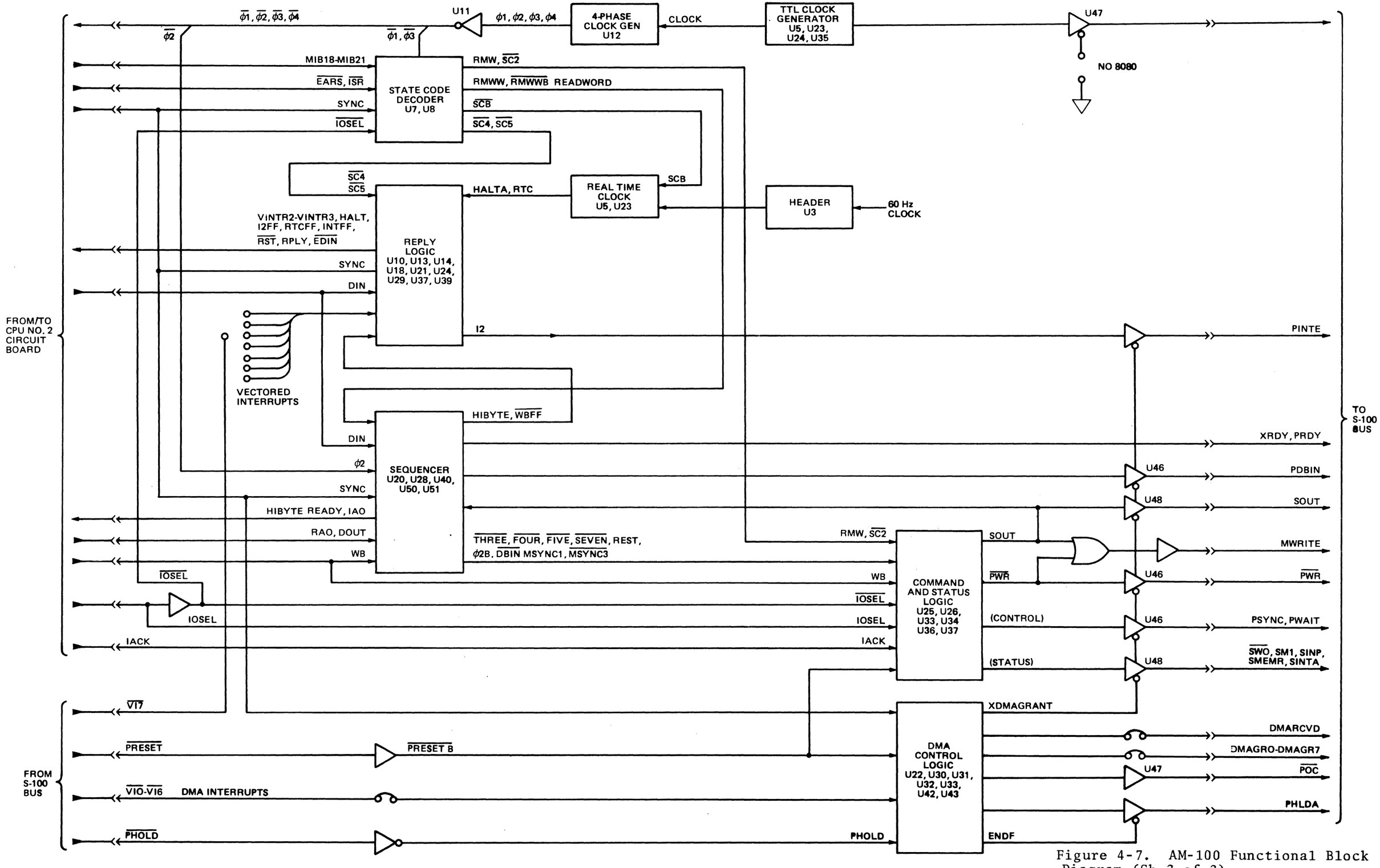


Figure 4-7. AM-100 Functional Block Diagram (Sh 2 of 2)

Table 4-5. AM-100 Signals List

SIGNAL	NAME	BOARD NO.	J2 PIN NO.	SCHEM PAGE OF SOURCE	FUNCTION
BDAL00- BDAL15	Buffered Data and Address Lines	1	-	CPU1-8	Internal tri-state bus for Data and Address Lines.
BUSY	Busy	1 2	14	CPU1-5 CPU2-5	Asserted when any of the seven vectored interrupts (VI0-VI6, PHOLD) are received. Indicates that a DMA exchange is in process.
$\overline{\text{CLOC}}$	2 MHz Clock	2	-	CPU2-3	Output of 2 MHz clock.
DALOO- DAL15	Data and Address Lines	1	-	CPU1-4, 6, 7	Tri-state bus for Data and Address Lines.
DBIN $\overline{\text{DBIN}}$	Data Bus In	2	-	CPU2-8	Read Enable equivalent to PDBIN on S-100 Bus. Enables CPU to read data from the Data In Bus.
DIN $\overline{\text{DIN}}$	Data In	1 2	5	CPU1-5 CPU2-6	Control signal from the processor to cause the address unit to gate its read data on the data lines.
DOUT	Data Out	2	9	CPU1-5 CPU2-3	Control signal from the processor which is made high at the same time that the write data is placed on the DAL bus by the processor.
$\overline{\text{EARS}}$	External Address Register Select	1 2	29	CPU1-7 CPU2-4	Output of the state code decoder that reads the starting address of the bootstrap PROM.

Table 4-5 (Cont.). AM-100 Signals List

SIGNAL	NAME	BOARD NO.	J2 PIN NO.	SCHEM PAGE OF SOURCE	FUNCTION
ENDF		2	-	CPU2-5	Not used.
ENDR		2	-	CPU2-5	Not used.
$\overline{\text{FIVE}}$	Sequencer Count Five	2	-	CPU2-9	Sequencer output for count five.
FOUR $\overline{\text{FOUR}}$	Sequencer Count Four	2	-	CPU2-9	Sequencer output for count four.
HALT	Halt	1 2	10	CPU1-5 CPU2-6	Used for CPU interrupt I3.
HALTA	Halt-A	2	-	CPU2-2	Output of real time clock header used to generate HALT for CPU interrupt I3.
HIBYTE $\overline{\text{HIBYTE}}$	High Byte	2	3	CPU1-6 CPU2-8	When asserted, switches the upper byte (bits 8-15) to the output port from the WD16. When it is low, the lower byte (bits 0-7) is selected.
IACK $\overline{\text{IACK}}$	Interrupt Acknowledge	2	7	CPU1-5 CPU2-7	Control signal output from the CPU to indicate that the processor is responding to an interrupt.
IAO	Initiate Address AO	1	26	CPU1-10 CPU2-3	Sequencer output to generate LSB of address byte in either bit A0 or A8.
IDIN	Interrupt Data In	1 2	30	CPU1-6 CPU2-6	Enable signal to apply stored vectored interrupts VINTR1-VINTR3 to Data and Access Lines DAL1-DAL3.
INTFF	Vectored Interrupt Input	1 2	13	CPU1-5 CPU2-6	CPU vectored interrupt IO. Asserted with any of the vectored interrupts.

Table 4-5 (Cont.). AM-100 Signals List

SIGNAL	NAME	BOARD NO.	J2 PIN NO.	SCHEM PAGE OF SOURCE	FUNCTION
$\overline{\text{IOSEL}}$	I/O Selected	1,2	24	CPU1-9 CPU2-7	Asserted for an I/O data transfer. False for a memory transfer.
$\overline{\text{ISR}}$	Input Status Register	1 2	28	CPU1-7 CPU2-4	Generated by State Code 6 ($\overline{\text{SC6}}$) to apply the status data to the DAL bus bits DAL0-DAL7.
I2	Interrupt 2	2	-	CPU2-6	Interrupt signal set by State Code 4 ($\overline{\text{SC4}}$) and cleared by State Code 5 ($\overline{\text{SC5}}$). Serves as an Interrupt enable for vectored interrupts.
I2FF	CPU Interrupt I2	1 2	11	CPU1-7 CPU2-6	CPU vectored interrupt I2 generated by any vectored interrupt.
JFIVE	Set Sequence 5	2	-	CPU2-9	J input to sequencer flip-flop 5.
JFOUR $\overline{\text{JFOUR}}$	Set Sequence 4	2	-	CPU2-9	J input to sequencer flip-flop 4.
$\overline{\text{JREST}}$	Set Sequence Rest	2	-	CPU2-8	J input to sequencer flip-flop zero (REST).
$\overline{\text{JSEVEN}}$	Set Sequence 7	2	-	CPU2-9	J input to sequencer flip-flop 7.
JTHREE	Set Sequence 3	2	-	CPU2-9	J input to sequencer flip-flop 3.
MIB00- MIB21	Microinstruction Bus Bits 0-21	1 2	21, 22, 23, 25	CPU1-2,3,4,5 CPU2-4	Data bus to transmit data between the five chips at the CPU chip set.
MSYNC1 $\overline{\text{MSYNC1}}$	M Sync One	2	-	CPU2-8	State one of the sequencer.
PHLDA	P-Hold Acknowledge	2	-	CPU2-3	When asserted, indicates that present bus master is releasing control of the bus in response to a DMA request (PHOLD) (forced high and not used).

Table 4-5 (Cont.). AM-100 Signals List

SIGNAL	NAME	BOARD NO.	J2 PIN NO.	SCHEM PAGE OF SOURCE	FUNCTION
PHOLD	DMA Request Line	2	-	CPU2-2	Used by DMA controllers to request bus mastership (forced low and not used).
$\overline{\text{POC}}$	Power On Clear	2	-	CPU2-3	Pulse signal asserted on initial power-up and by PRESETB signal.
$\overline{\text{POS}}$	Power On Set	2	-	CPU2-3	Inverse of Power On Clear.
$\overline{\text{PRESETB}}$	Preset	2	-	CPU2-2	Buffered preset signal from S-100 Bus.
PSYNCI	Processor Sync Internal	2	-	CPU2-7	Indicates the start of a bus cycle, used to generate PSYNC on S-100 BUS.
RAO	Stored Address Zero	1 2	27	CPU1-9 CPU2-3	Address data bit zero stored from the DAL bus clocked into the register by SYNC.
RA01- RA15	Stored Address	1	-	CPU1-9	Address Data stored from the DAL bus clocked into the register by SYNC.
RB $\overline{\text{RB}}$	Read Byte	2	-	CPU2-4	State Code decoder signal for Read Byte operation.
READWORD	Read Word	2	-	CPU2-4	State code decoder signal for Read Word operation.
READY $\overline{\text{READY}}$	Ready	1 2	4	CPU1-6 CPU2-8	Sequencer output indicating that the CPU chip set is ready to accept data.
REST	Rest	2	-	CPU2-8	The state of the sequencer when no instruction or bus operations are taking place (state 0 of the sequencer).
RMW	Read-Modify-Write	2	-	CPU2-4	State code decoder signal for Read-Modify-Write either word or byte.

Table 4-5 (Cont.). AM-100 Signals List

SIGNAL	NAME	BOARD NO.	J2 PIN NO.	SCHEM PAGE OF SOURCE	FUNCTION
RMWB $\overline{\text{RMWB}}$	Read-Modify-Write-Byte	2	-	CPU2-4	Generated during an INPUT BYTE micro op code with RMW active to indicate a Read-Modify-Write byte sequence (1010 Microcode).
RMWW $\overline{\text{RMWW}}$	Read-Modify-Write-Word	2	-	CPU2-4	Generated during an INPUT WORD micro op code with RMW active to indicate a Read-Modify-Write word sequence (1001 Microcode).
RPLY	Reply	1 2	16	CPU1-5 CPU2-6	Control signal used by the addressed unit to respond to the processors data access signals.
$\overline{\text{RST}}$	CPU Reset	1 2	15	CPU1-5 CPU2-6	Resets CPU chip set on initial power-up.
RTC	Real Time Clock	2	-	CPU2-2	
RTCFF	CPU Real Time Clock	1 2	12	CPU1-5 CPU2-6	Real time clock input to CPU.
SC1-SCF	State Codes 1-F (Hex)	2	-	CPU2-4	Outputs of state code decoder.
$\overline{\text{SEVEN}}$	Sequencer Count Seven	2	-	CPU2-9	Sequencer output for count seven.
SIX	Sequencer Count Six	2	-	CPU2-9	Sequencer output for count six.
SOUT $\overline{\text{SOUT}}$	I/O Output Cycle	2	-	CPU2-7	When asserted indicates that the current bus cycle is a bus master output to an I/O address.
SYNC $\overline{\text{SYNC}}$	Sync	1 2	6	CPU1-5 CPU2-4	Output of the CPU chip set indicating the start of an I/O operation.

Table 4-5 (Cont.). AM-100 Signals List

SIGNAL	NAME	BOARD NO.	J2 PIN NO.	SCHEM PAGE OF SOURCE	FUNCTION
THREE $\overline{\text{THREE}}$	Sequencer Count Three	2	-	CPU2-9	Sequencer output for count three.
VINTR 1	Vectored Interrupt 1	1 2	31	CPU1-6 CPU2-6	Vectored interrupt level one signal for the DAL bus.
VINTR 2	Vectored Interrupt 2	1 2	32	CPU1-6 CPU2-6	Vectored interrupt level two signal for the DAL bus.
VINTR 3	Vectored Interrupt 3	1 2	33	CPU1-6 CPU2-6	Vectored interrupt level three signal for the DAL bus.
WO $\overline{\text{WO}}$	Write Operation	2	-	CPU2-7	Indicates that the current process is a write operation.
WAIT	Wait	1	-	CPU1-4	Signal from CPU data chip to control chip to establish whether the data chip is in the Run or Wait mode. Low = Run, and the microinstruction is loaded and executed.
WB	Write Byte	1 2	8	CPU1-5 CPU2-3	Control signal from the processor which is high when the address is on the bus to signify a WRITE rather than a READ and high during Data Out to signify BYTE rather than WORD.
WBFF $\overline{\text{WBFF}}$	Write Byte Output	2		CPU2-3	Write Byte signal clocked by DOUT.
XDMAG	XDMA Grant	-	2	CPU1-8 CPU2-5	Controls the buffers that interface the CPU control signals with the S-100 bus to allow them to be removed when another master is controlling the bus.

Table 4-5 (Cont.). AM-100 Signals List

SIGNAL	NAME	BOARD NO.	J2 PIN NO.	SCHEM PAGE OF SOURCE	FUNCTION
XDMAGRANT	XDMA Grant	1 2	1	CPU1-10 CPU2-5	Same as XDMAG.
$\overline{\emptyset 1}$	Phase 1 Clock	1 2	18	CPU1-1 CPU2-2	Phase 1 of 2 MHz 4-phase clock see Figure 4-8.
ø1M	Phase 1 Clock, MOS Levels	1	-	CPU1-1	Phase 1 of 2 MHz 4-phase clock buffered for 10.5 volt operation of CPU chip set.
$\overline{\emptyset 2}$	Phase 2 Clock	1 2	17	CPU1-1 CPU2-2	Phase 2 of 2 MHz 4-phase clock see Figure 4-8.
ø2B	Phase 2 Clock	2	-	CPU2-8	Phase 2 of 2 MHz 4-phase clock buffered for operation sequencer flip-flops.
ø2M	Phase 2 Clock, MOS Levels	1	-	CPU1-1	Phase 2 of 2 MHz 4-phase clock buffered for 10.5 volt operation of CPU chip set.
$\overline{\emptyset 3}$	Phase 3 Clock	1 2	20	CPU1-1 CPU2-2	Phase 3 of 2 MHz 4-phase clock see Figure 4-8.
ø3M	Phase 3 Clock, MOS Levels	1	-	CPU1-1	Phase 3 of 2 MHz 4-phase clock buffered for operation of CPU chip set.
$\overline{\emptyset 4}$	Phase 4 Clock	1 2	19	CPU1-1 CPU2-2	Phase 4 of 2 MHz 4-phase clock see Figure 4-8.
ø4M	Phase 4 Clock, MOS Levels	1	-	CPU1-1	Phase 4 of 2 MHz 4-phase clock buffered for operation of CPU chip set.

4.2.1 CLOCK GENERATOR.

The basic clock signals originate at the 4 MHz crystal oscillator U5. Its output is divided-by-two by the J-k flip-flop U23, gated and buffered to generate clock signals Ø1, Ø2 and $\overline{\text{CLOC}}$. Clock Ø2 provides the S-100 Bus clock and $\overline{\text{CLOC}}$ drives the four-phase clock generator U12. The four phase clock generator provides the internal clock signals to drive the CPU chip set and to synchronize its associated logic. Operation of U12 is described in paragraph 4.4.1. The timing relationship of the S-100 bus clocks and the four-phase clocks is described in Figure 4-8.

The TTL level clocks from CPU 2 are sent to CPU 1 where they are buffered through U2 and U3 to a level of at least 10.5 volts to drive the CPU chip set.

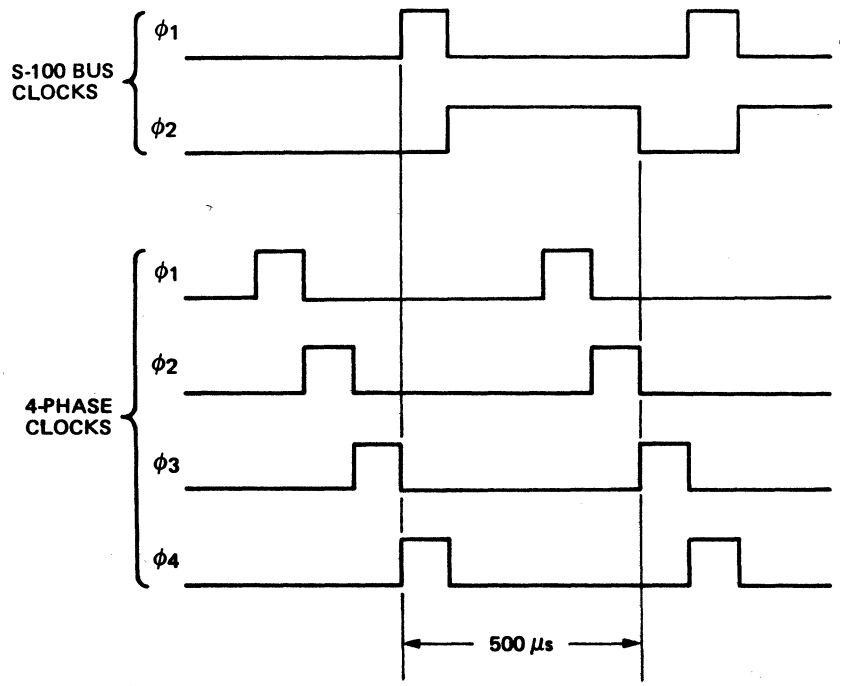


Figure 4-8. AM-100 System Clocks

4.2.2 STATE CODE DECODER LOGIC.

The microinstruction state codes are contained in microinstruction bits MIB18-MIB21 from the microinstruction bus in the CPU chip set located on CPU 1. These bits are buffered on CPU 1 and then sent to decoders U7 and U8 on CPU 2. The information contained in these bits controls the type of processing to be performed. The decoder logic generates signals to control the input of the status register and control the sequencer to perform the required processing operations (write, read, read/modify/write). The state codes and their functions are listed in Table 4-6.

4.2.3 SEQUENCER OPERATION.

The sequencer on the AM-100 CPU provides the timing and synchronizing required to interface the 16 bit CPU chip set to the 8-bit S-100 Bus. There are ten states in this sequencer that provide the various operations required for CPU data processing.

State	Function
0	Rest
1	MSYNC1 - The sync signal for memory or I/O.
2, 3	DBIN - Input functions. Once around for byte, twice for word.
4	Read-Modify-Write chain.
5, 7	Output functions. Once around for byte, twice for word.
HIBYTE	LSB Address Control.
READY	Replies to the CPU and latches the upper byte of the DAL to the lower byte of the DAL during READ WORD operations.

Table 4-6. Microm State Code Functions

CODE	MNEMONIC	NAME	FUNCTION
0001	PMSK	Priority Mask Out.	The state code is generated on an OUTPUT WORD instruction when a new mask is written into location 2E. It signals the I/O devices that a new interrupt mask is on the DAL. (Not used on AM-100.)
0010	RUN	Macro Instruction Fetch	Generated during macro instruction fetch from memory for a run light.
0011	IORST	I/O Reset	Not Used.
0100	INTEN	I2 Set	Enables the interrupt enable line I2. Sets I2 flip-flop.
0101	INTDS	I2 Reset	Disables the interrupt enable line I2. Resets I2 flip-flop.
0110	ESRR	External Status Register Request	Generated during an INPUT STATUS BYTE micro op code to indicate that the external status register is being requested.
0111	SRS	System Reset	Not used.
1000	BYTE	Read Byte Operation	Generated during an INPUT BYTE micro op code to indicate a read byte operation without a read/modify/write. (Not used on AM-100.)
1001	RMWW	Read/Modify/Write Word	Generated during an INPUT WORD micro op code with RMW active to indicate a read/modify/write word sequence. (Not used on AM-100.)
1010	RMWB	Read/Modify/Write Byte	Generated during an INPUT BYTE micro op code with RMW active to indicate a read/modify/write byte sequence. (Not used on AM-100.)

Table 4-6. Microm State Code Functions

CODE	MNEMONIC	NAME	FUNCTION
1011	RLCI	Reset Line Clock Interrupt	
1100	EARR	External Address Register Request	Controls where the CPU looks for the bootstrap PROM upon power up.
1101	-	Duplicate of BYTE	Same as BYTE except one bit sooner.
1110	-	Duplicate of RMWW	Same as RMWW except one bit sooner.
1111	-	Duplicate of RMWB	Same as RMWB except one bit sooner.

The flow diagram for sequencer operations is shown in Figure 4-9 and the associated logic equations are shown in Figure 4-10. Each block on the flow diagram represents a flip-flop (CPU2 schematic sheets 8 and 9). The J equations set the flip-flop on the next clock, and the K equations reset the flip-flop on the next clock.

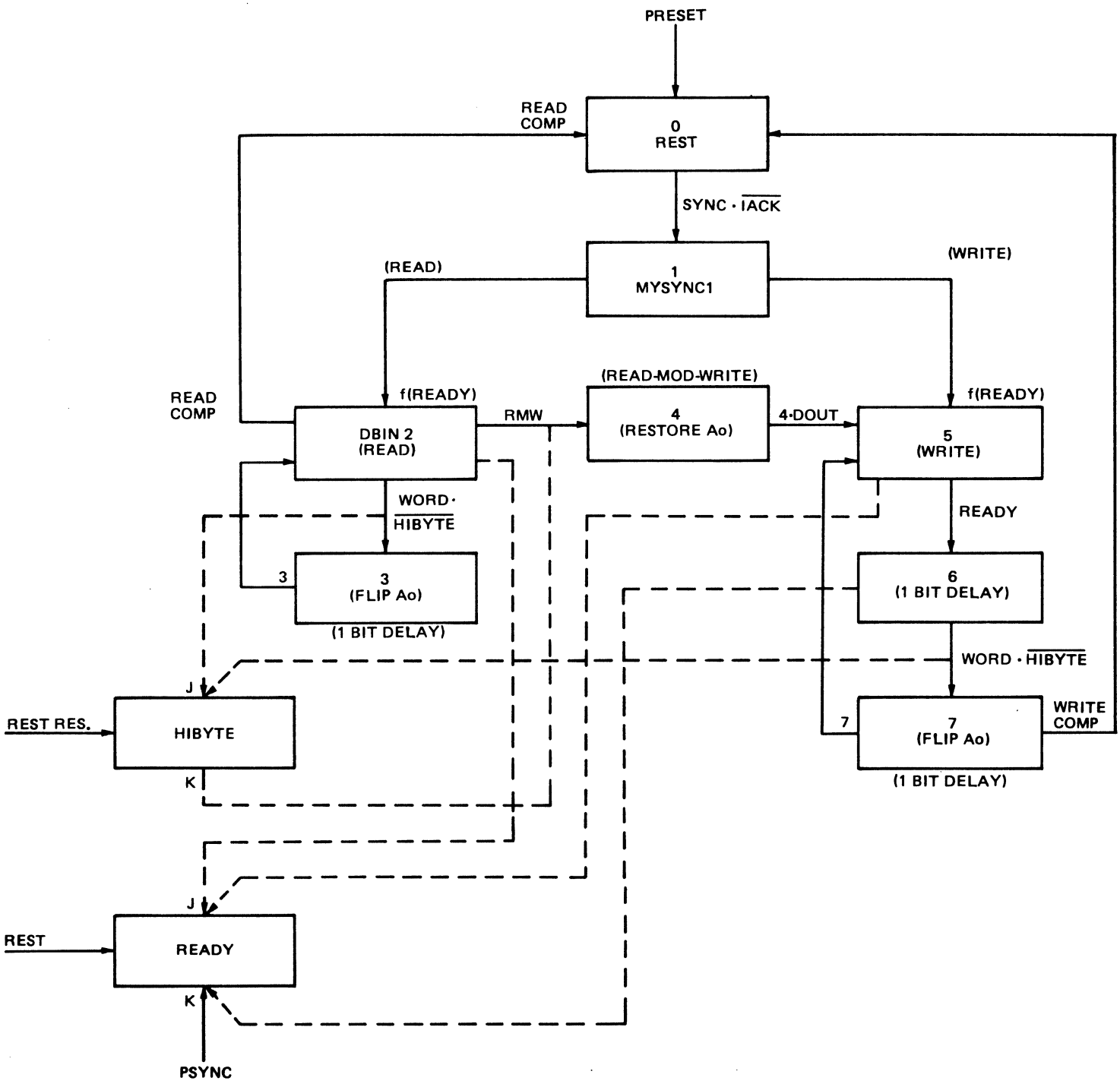
Note

The sequencer clock is Ø2 from the four-phase clock generator.

Timing diagrams for sequencer operations are contained in Figure 4-11 through 4-16. The timing relationships given show the S-100 Bus timing on the top part of the diagram with the processor signals necessary to produce the bus signals drawn underneath. Only one state of the sequencer is active at a time.

On initial power up, the reset line (PRESETB) is forced low initializing the sequencer to the rest or zero state. The sequencer is in this state until a bus operation is required.

When the CPU chip set is ready for an I/O operation, SYNC is asserted to generate MSYNC1 which is state one of the sequencer. When the Interrupt Acknowledge (IACK) line is true, the processor is not allowed to perform a bus operation. Input signals to the sequencer from the state code decoder and CPU then determine what type of sequence is required. Eight different sequences are possible as shown on Figure 4-9. The sequencer then proceeds through the required combinations of states as shown on the flow diagram.



SEQUENCE OF OPERATION

MODE	STATES	REPLY LOGIC*
1. READY BYTE (MEM OR I/O)	0-1-2-0	READY · RB · DIN
2. READY WORD (MEM OR I/O)	0-1-2-3-2-0	READY · RW · HYBITE · DIN
3. WRITE BYTE (MEM OR I/O)	0-1-5-6-0	READY · WB · 5
4. WRITE WORD (MEM OR I/O)	0-1-5-6-7-5-6-0	READY · WB · 5 · HIBYTE
5. RMW BYTE (MEM)	0-1-2-4-5-6-0	1+3 ABOVE
6. RWM WORD (MEM)	0-1-2-3-2-4-5-6-7-5-6-0	2+4 ABOVE
7. RMW BYTE (I/O)	0-1-5-6-0	DIN+3 ABOVE
8. RMW WORD (I/O)	0-1-5-6-7-5-6-0	DIN+4 ABOVE

*STROBED AT ϕ1 TRAILING EDGE AND GATED WITH SYNC. REPLY LOGIC GENERATES SIGNALS TO TERMINATE CPU CYCLE.

Figure 4-9. Sequencer Flow Diagram

REST

$JREST = DBIN \cdot \overline{SYNC} \cdot \overline{DIN} + 6 \cdot \overline{SYNC}$

$KREST = JMSYNC_1$

$MSYNC_1$

$JMSYNC_1 = REST \cdot SYNC \cdot \overline{IACK}$

$KMSYNC_1 = JDBIN + J5$

DBIN

$DBIN = 3 + MSYNC_1 \cdot DIN \cdot \overline{(IOSEL\ [RMWB + RMWW])}$

$KDBIN = JREST + J3 + J4$

4

$J4 = DBIN \cdot \overline{DIN} \cdot (RMWB \cdot READY + RHWW \cdot READY \cdot HIBYTE)$

$K4 = 4 \cdot DOUT$

5

$J5 = MSYNC_1 \cdot DOUT + K4 + 7$

$K5 = J6$

3

$J3 = READY \cdot \overline{HIBYTE} \cdot DBIN \cdot \overline{(RB + RMWB)}$

$K3 = 3$

6

$J6 = 5 \cdot READY$

$K6 = JREST + J7$

7

$J7 = 6 \cdot WBFF \cdot \overline{HIBYTE}$

$K7 = 7$

ADDRESS LSB LOGIC

$A_o = A_{oI} \cdot \overline{(READWORD + W_8 \cdot DOUT)} + HIBYTE \cdot \overline{(READWORD + W_8 \cdot DOUT)}$

HIBYTE

$JHIBYTE - J3 + J7$

$KHIBYTE = J4$

DIR. RESET = REST

READY

$JREADY = PRDY \cdot XRDY \cdot (DBIN + 5)$

$KREADY = PSYNC + 6$

DIR RESET = REST

BUS SIGNALS

$PSYNC = MSYNC_1 + 4 + 7 + 3$

$PWR = 5$

$PDBIN = DBIN$

$SMEMR = \overline{IOWEL} \cdot \overline{SW_o}$

$SINP = IOSEL \cdot \overline{SW_o}$

$SOUT = IOSEL \cdot SW_o$

$MWRITE = PWR \cdot \overline{SOUT}$

Figure 4-10. Sequencer Logic Equations

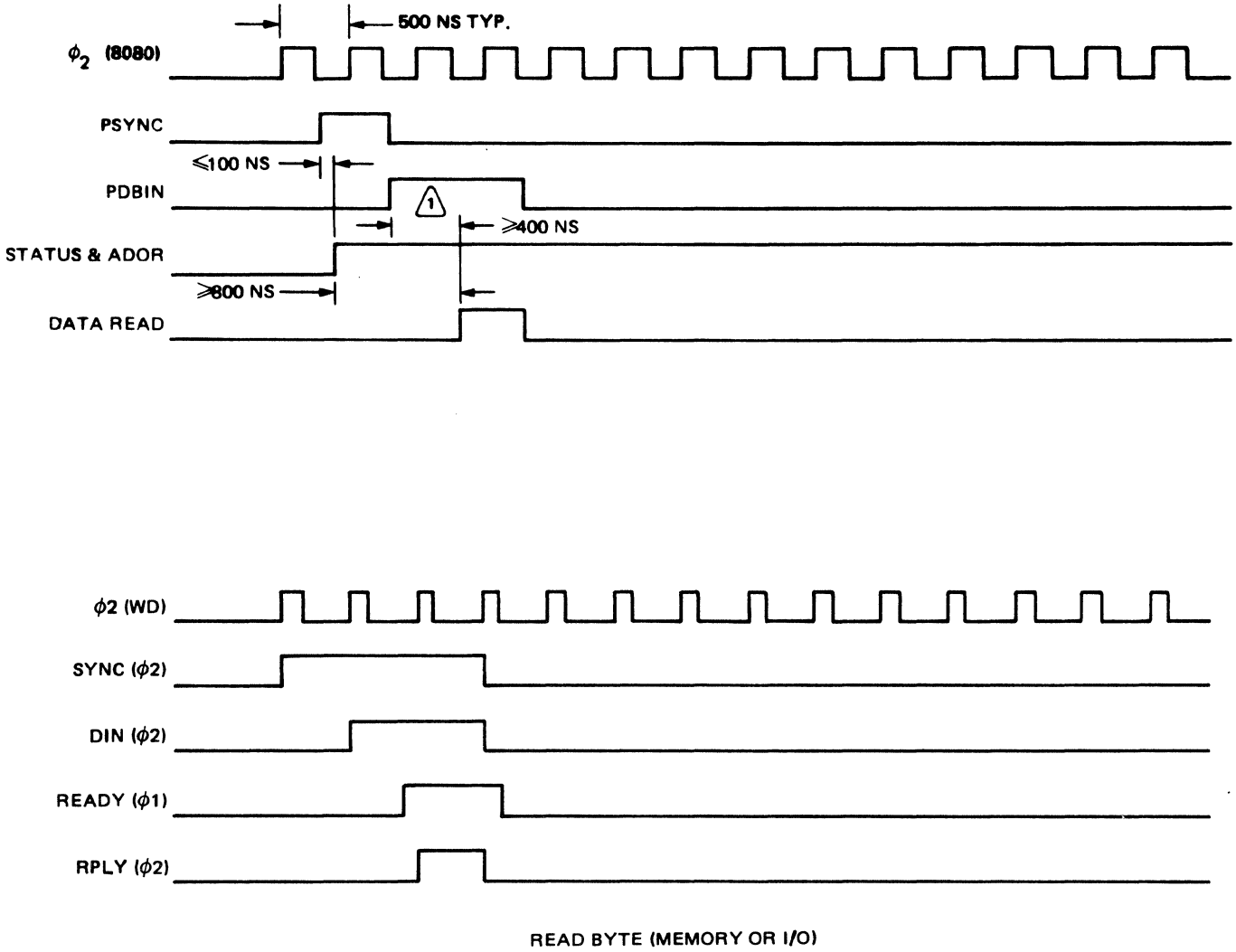
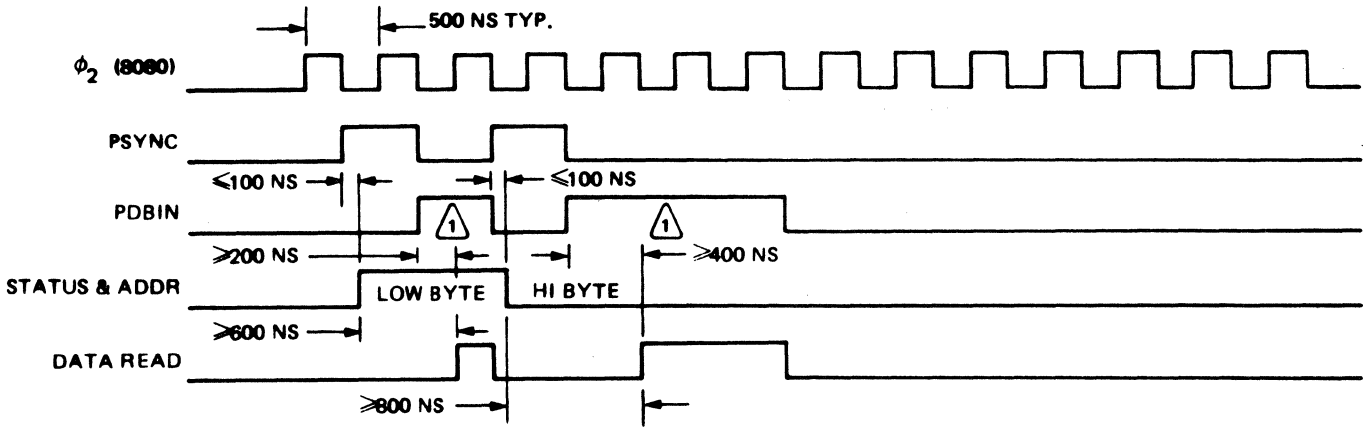
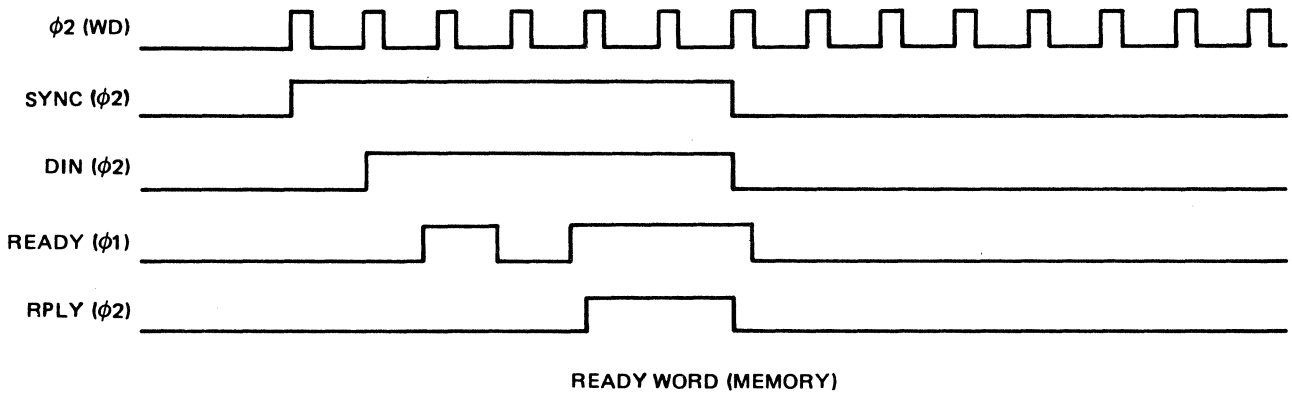


Figure 4-11. CPU Read Byte Timing

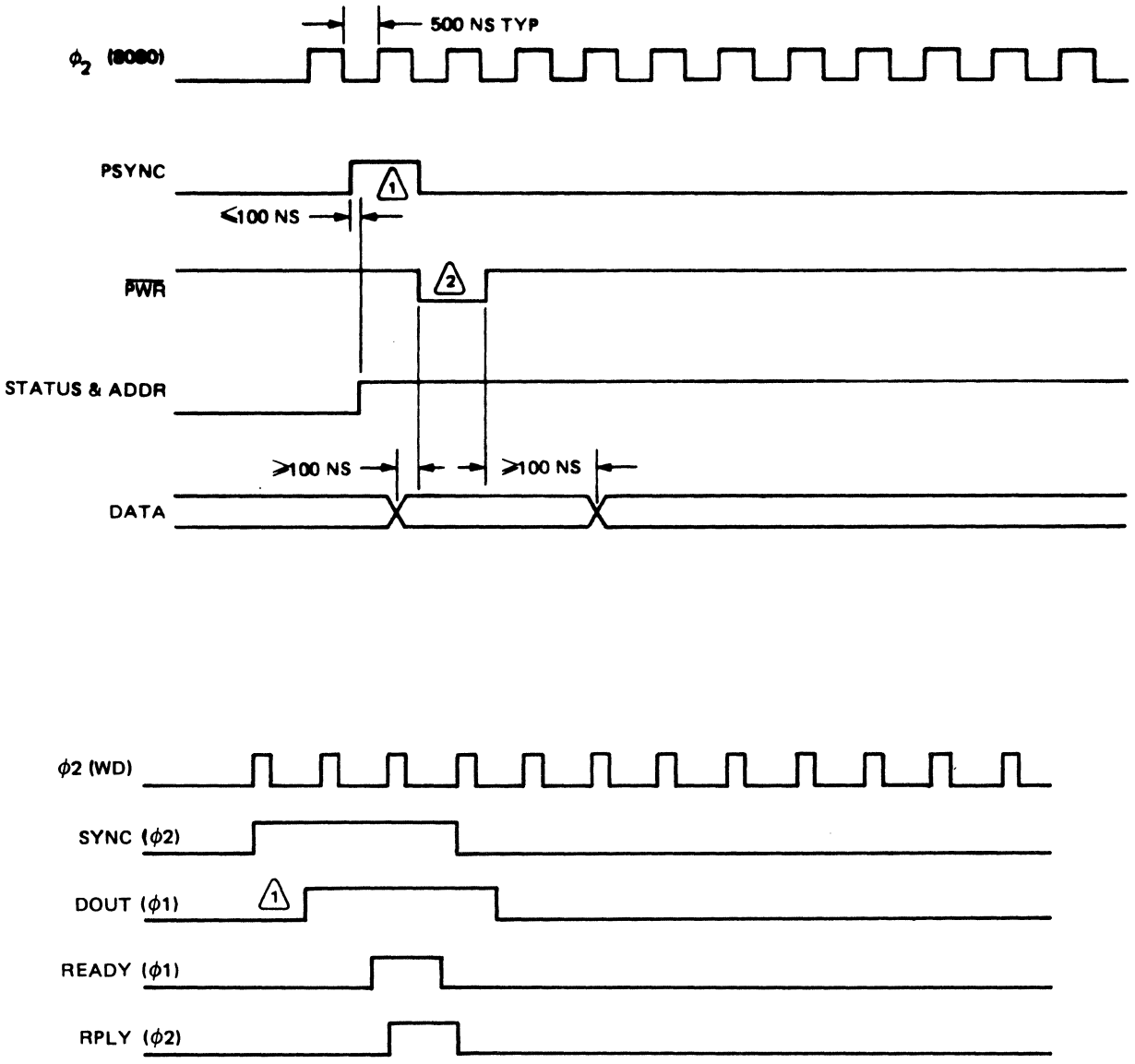


1 PDBIN MAY BE STRETCHED BY CONTROLLING PRDY. PRDY IS EXAMINED AT ϕ1 (WD) LEADING EDGE TIME DURING PDBIN.



1 PDBIN MAY BE STRETCHED BY CONTROLLING PRDY. PRDY IS EXAMINED AT ϕ1 (WD) LEADING EDGE TIME DURING PDBIN.

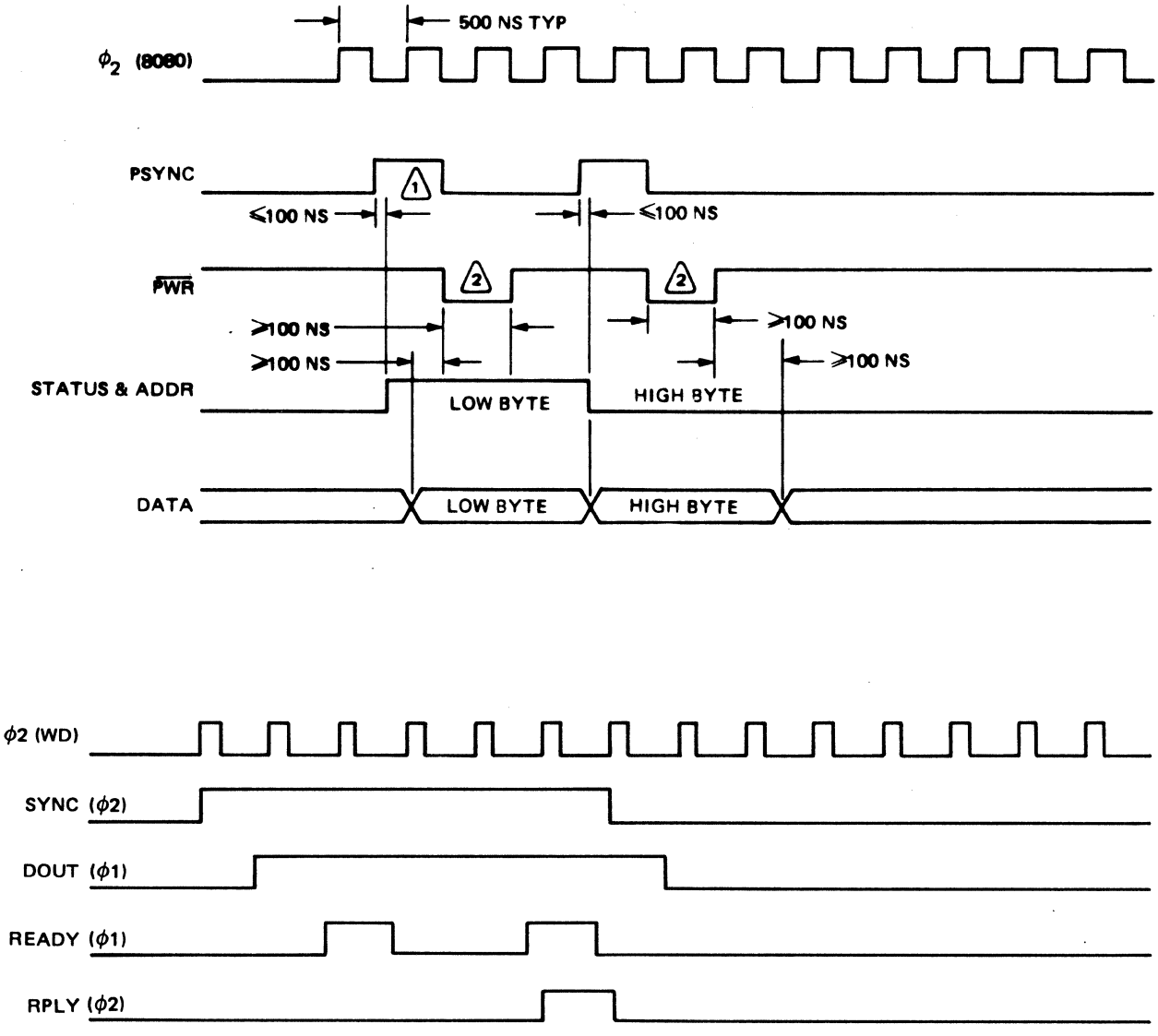
Figure 4-12. CPU Read Word Timing



WRITE BYTE (MEMORY OR I/O)

1. PSYNC MAY BE STRETCHED ONE OR MORE CYCLES DUE TO DOUT DELAY CAUSED BY MICROCODE.
2. $\overline{\text{PWR}}$ MAY BE STRETCHED BY CONTROLLING PRDY. PRDY IS EXAMINED AT φ1 (WD) LEADING EDGE TIME DURING PWR.

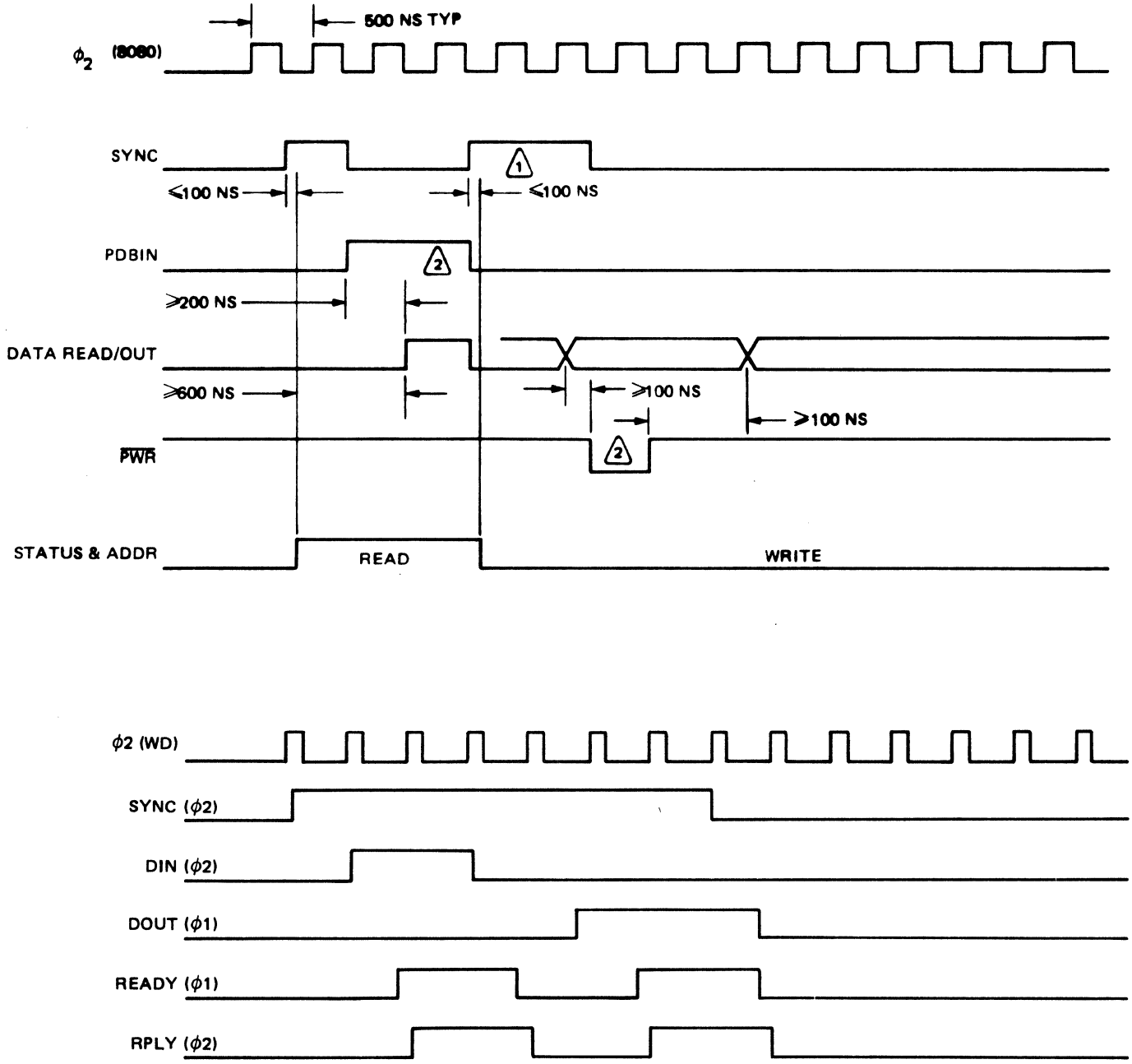
Figure 4-13. CPU Write Byte Timing



WRITE WORD (MEMORY)

1. PSYNC MAY BE STRETCHED ONE OR MORE CYCLES DUE TO DOUT DELAY CAUSED BY MICROCODE.
2. $\overline{PWR}$ MAY BE STRETCHED BY CONTROLLING PRDY. PRDY IS EXAMINED AT ϕ1 (WD) LEADING EDGE TIME DURING PWR.

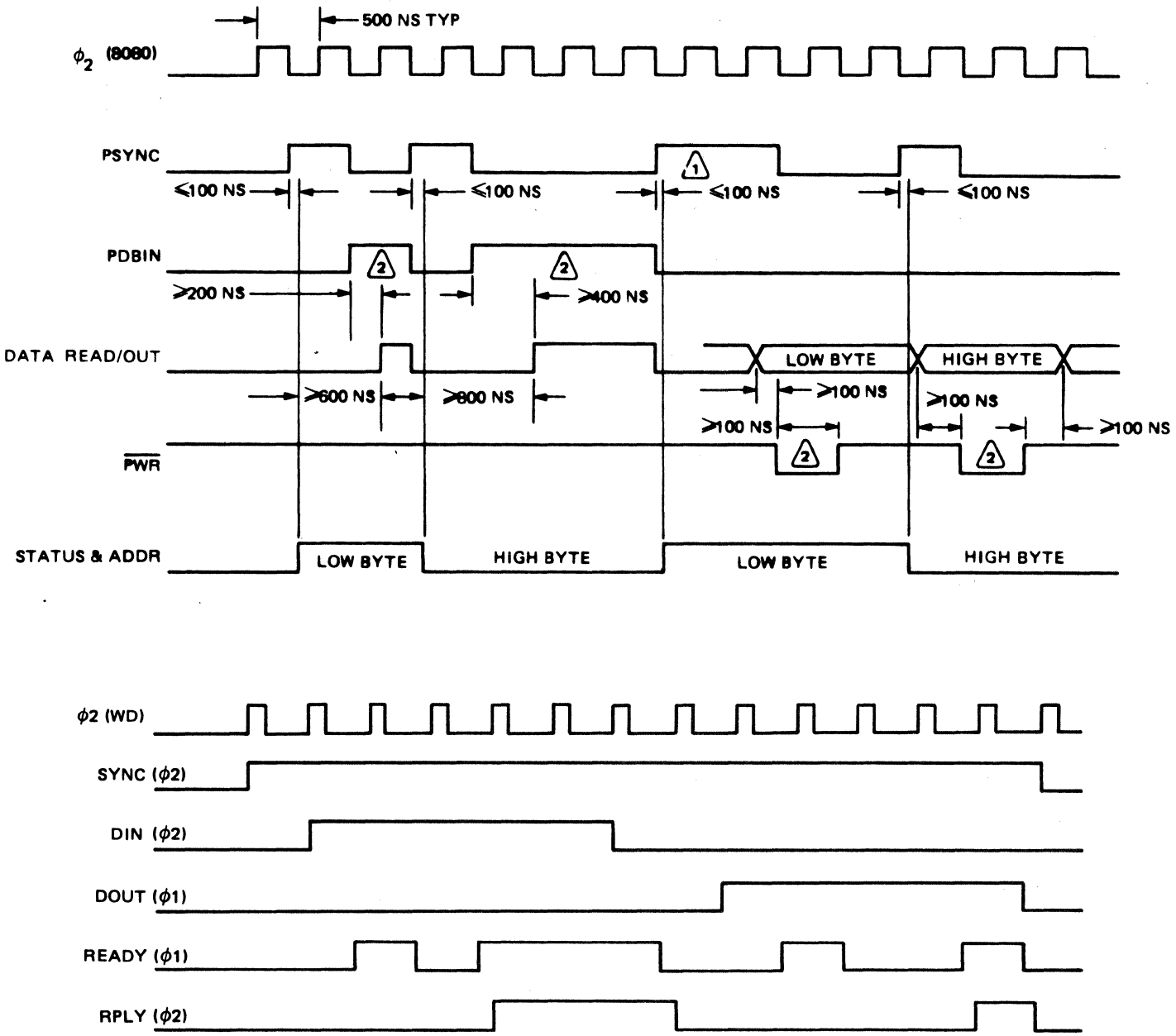
Figure 4-14. CPU Write Word Timing



READ-WRITE-MODIFY-WRITE BYTE

1. PSYNC MAY BE STRETCHED ONE OR MORE CYCLES DUE TO DOUT DELAY CAUSED BY MICROCODE.
2. PDBIN AND $\overline{\text{PWR}}$ MAY BE STRETCHED BY CONTROLLING PRDY. PRDY IS EXAMINED AT ϕ1 (WD) LEADING EDGE TIME DURING PDBIN OR PWR.

Figure 4-15. CPU Read-Modify-Write Byte Timing



READ-MODIFY-WRITE WORD

1. PSYNC MAY BE STRETCHED ONE OR MORE CYCLES DUE TO DOUT DELAY CAUSED BY MICROCODE.
1. PDBIN AND $\overline{\text{PWR}}$ MAY BE STRETCHED BY CONTROLLING PRDY. PRDY IS EXAMINED AT ϕ1 (WD) LEADING EDGE TIME DURING PDBIN OR PWR.

Figure 4-16. CPU Read-Modify-Write Word Timing

4.2.4 INITIALIZATION AND STATUS.

Initialization of the AM-100 occurs when the board set receives its initial power. Initial reset is accomplished by Power-On-Clear signal POC which goes low on initial power-up or for the PRESET signal that actuates the $\overline{POC}$ one-shot U22. Signal $\overline{POC}$ provides an output to the S-100 Bus and also clears register U10 in the reply logic.

The logic that controls initialization of the CPU chip set is located on CPU1 and consists of two buffers: the status Byte Register buffer and the External Address Register Select buffer.

The Status Byte buffer is used by the CPU chip set during power-up to determine the initialization sequence. There are eight bits in this register that function as defined in Table 4-7. The CPU chip set checks these bits as it proceeds through its initialization.

Table 4-7. Status Byte Buffer Bits

BIT	NAME	FUNCTION
0 1	Power-Up Option Jumpers	00 Used for AM-100
2 3	Halt Option Jumpers	Not Used
4	Interrupt Enable	Vectored Interrupt Enable
5	Parity Error Status	Wired to ground
6	Bus Error Status	Wired to ground
7	Power Fail Status	Wired to ground

The external address register is used by the CPU to determine the base address of the bootstrap ROM. The base address is six bits that are jumper selected by the header in U3. During the power-up sequence, these six bits are added to C000 to determine the bootstrap starting address.

Note

See Section 2 of this manual for header U3 wiring for the various peripherals used with the AM-100.

The initialization process for the AM-100 must proceed through the sequence described below so that the CPU can access the bootstrap load program and begin executing. When the RESET pushbutton is pressed, the CPU sends out an external request to read the Status Register. This signal is $\overline{\text{ISR}}$ that accesses the eight bits of data as defined in Table 4-7. The CPU then checks each bit as described in Figure 4-17. When the CPU has checked the eight bits of the Status Register, signal $\overline{\text{EARS}}$ is asserted from the state code decoder and reads the base address from header U3.

4.2.5 DATA ACCESS.

Data is processed from the eight-bit S-100 bus system to the 16-bit CPU through the supporting logic on both CPU boards. The data bus is addressed and then the CPU performs either read or write operations. Read/Modify/Write is a combination of both in one sequence.

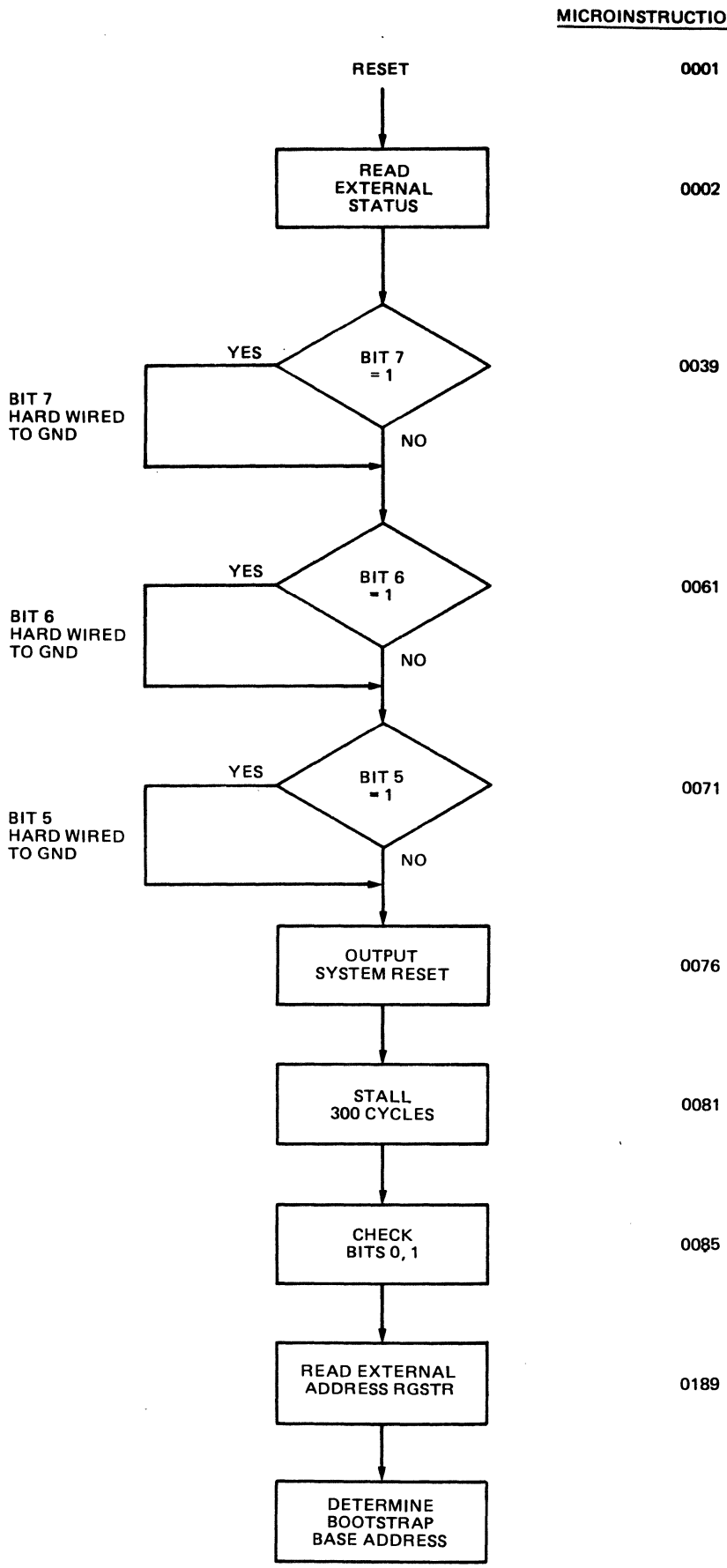


Figure 4-17. AM-100 Power Up Sequence

4.2.5.1 ADDRESSING OPERATIONS.

The CPU chip set utilizes a data bus that contains both the data and address multiplexed onto the same lines. When the CPU chip set has address data on the DAL, the SYNC signal goes true. This is applied to the clock input of the address latches U13, U14 and U25 to store the address data. Bits A1-A7 are applied directly to the S-100 bus through buffers, but the other bits require additional processing.

There are two basic bus operations where addressing is concerned: memory and I/O. If we are performing memory read or write, then the lower byte contains A0-A7 and the upper byte contains A8-A15. When I/O operations are taking place, the CPU sends out Hex address FFXX. This generates signal IOSEL from the gate on U15 to distinguish between I/O and memory operations. Signal IOSEL switches the Data Select MUX to place A0-A7 on both the upper and lower bytes of the address bus.

Address line A0 is generated by the sequencer logic on CPU2 to provide for byte/word data operations. With word operations, two sequential addresses are fetched so A0 is sequenced first with 0, then with 1. With byte operations, address line A0 is unaffected by the sequencer.

4.2.5.2 WRITE OPERATIONS.

The DAL bus that processed the address data also contains the data for write operations. The buffered data lines are applied to the data select MUX on U17 and U18. This selects the appropriate byte for S-100 bus transfer to the selected memory or I/O controller. Signal HIBYTE selects either the upper byte or lower byte for transfer to the output bus of the S-100 Bus.

If a write byte operation is taking place, only eight bits of data are required. The address is first transmitted and then the lower byte of data is placed on both DAL0-DAL7 and DAL8-DAL15. If a write word operation is taking place, a two-stage operation transmits the lower byte first to the eight S-100 lines and then switches to the upper byte. Signal HIBYTE makes this selection.

4.2.5.3 READ OPERATIONS.

The data input lines are buffered and are applied to the internal DAL bus on DAL8-DAL15. This input data is then stored in the Low Byte Input Storage register on U9 and U10. For the read byte sequence, the S-100 bus signals request that the addressed memory or I/O port place its data on S-100 bus lines DI0-DI7; signal $\overline{\text{DIN}}$ gates it on to the internal data bus to the lower byte storage. Signal READY from the sequencer indicates that the CPU chip set is ready to accept the data so it is clocked into the Low Byte Input Storage register. The lower byte of data is on both DAL0-DAL7 and DAL8-DAL15, and the CPU can read from either one.

For the read word sequence, the lower byte is transmitted exactly like the read byte sequence. The sequencer then starts another cycle and increments the LSB of the address to fetch the data from an odd address in memory. The data is applied to the input port as before and is gated to DAL8-DAL15. The Low Byte Input Storage register is not clocked this time so it still contains the low byte and the high byte is on DAL8-DAL15. The operation is terminated and the CPU chip set reads a 16-bit word from DAL0-DAL15.

4.2.6 DMA AND INTERRUPT OPERATIONS.

The AM-100 provides seven levels of DMA and eight levels of interrupt capability. This data access processing is integrated with the interrupt operations of the CPU chip set.

4.2.6.1 CPU CHIP SET INTERRUPTS.

The CPU chip set provides capability for four interrupts as listed in Table 4-8. These interrupts are inputs to the control chip U28 on CPU1.

Table 4-8. CPU Chip Set Interrupts

INTERRUPT	PIN	SIGNAL	FUNCTION
I0	6	INTFF	Vectored interrupt
I1	5	RTCFF	Non-vectored interrupt - real time clock.
I2	4	I2FF	Enable for I0 and I1, controlled by user state codes.
I3	3	HALT	Halt switch (not used).

Interrupt I0 is the vectored interrupt and is the normal interrupt procedure. It is used by the peripheral controllers to signal the CPU when they have finished a task. Signal INTFF provides the I0 input from a signal generated on CPU2 when any of the eight vectored interrupts have been asserted. This signal is generated by decoder U29, inverted, and stored in a D flip-flop on U10. Interrupt I1 is the non-vectored interrupt and the real time clock signal RTCFF supplies this input. Interrupt I2 is an enable for I0 and I1 and is supplied by the user state codes. This is generated by flip-flop U18 on CPU2 that is preset by $\overline{SC4}$ and cleared by $\overline{SC5}$. Interrupt I3 is a halt switch input that is not implemented on the AM-100.

When a non-vectored interrupt is received by the CPU chip set (real time clock), it checks the contents of the status register to examine bit 7. This is the power fail bit which is not implemented in the S-100 bus structure so it is wired to ground. The CPU then determines that bit 7 is false and then fetches the program counter address from Location 2A.

4.2.6.2 VECTORED INTERRUPTS.

When a vectored interrupt is received by the CPU chip set (I0), the resulting process is similar to a read instruction. When Vectored Interrupt is asserted, jumper wires on CPU2 connect it to any of the eight pads VI0-VI7. This produces two outputs at the decoder: VINTR1-VINTR3 produce the interrupt code and the other output from pin 14 is asserted any time there is an input to the decoder. This produces INTFF which is applied to CPU interrupt I0. At the same time, VINTR1-VINTR3 are sent to CPU1 to the interrupt storage register U1 where they are clocked in by IACK. When Data In (DIN) is received from the CPU, $\overline{\text{IDIN}}$ is asserted which applies VINTR1-VINTR3 to the data bus DAL1-DAL3. The CPU then reads the interrupts from the bus.

4.2.6.3 DMA OPERATIONS.

The AM-100 provides seven levels of DMA capability. Levels VI0-VI6 are jumper selectable and level 7 is wired directly to PHOLD.

The other DMA inputs are jumper selectable and provide corresponding DMAGRANT ($\overline{\text{DMAGR0}}$-$\overline{\text{DMAGR6}}$) outputs to the S-100 Bus. A DMAG output ($\overline{\text{DMARCVD}}$) is generated one clock cycle before the selected DMAGRANT to signal all the DMA boards that a new DMA grant is about to be issued. No DMA device should issue a DMA request during the time $\overline{\text{DMARCVD}}$ is asserted.

4.3 CPU MICROPROCESSOR CHIP SET DESCRIPTION.

The Western Digital MCP1600 microprocessor is an 8-bit microprogrammable computer implemented with 3 MOS/LSI chips using UDC's N-channel silicon gate process. The chip set consists of the CP1611B Data chip, the CP1621B Control Chip, and the CP1631B microinstruction ROM (MICROM) chip.

These chips are interconnected by the 22 bit Microinstruction Bus (MIB) which provides bi-directional communication between the chips for addresses and instructions. A Data Access Bus provides a 16 bit port for communicating with other system components such as memory and I/O.

The list below describes the pertinent aspects of the set:

- 8-bit Internal Organization
- 16-bit Data Access Port to Memory and I/O
- 26 Registers
- Extensive Microinstruction Set Including Decimal Operations
- Single and Double Byte Operations
- Micro and Macro Level Condition Flags
- 512 Word x 22-bit Control ROM
- Control ROM Expandable to 2048 words
- Micro Level Subroutine Capability
- Programmable Translation Array for Macroinstruction Interpretation
- Four External Interrupts and Three Internal Interrupts
- External Instruction Set
- Power Supplies +12V, +5V, and -5V
- TTL Compatible 3 State Interface

Figure 4-18 describes the interconnection of the required components of the MCP 1600 microprocessor set. The DATA CHIP, (CP1611B) contains the arithmetic logic unit, the microinstruc-

tions decode and the register file. Additionally, it contains paths to control the operation of the processor.

The CONTROL CHIP (CP1621B) contains the program translation array, portions of the control circuitry to control operation of the processor set, the microinstruction counter and the I/O control system.

The MICROM CHIP (CP1631B) contains the microinstruction ROM. The MP1600 microprocessor set may be expanded up to four MICROMs giving the user a total of 2,048 22 bit microinstructions. In addition to the three parts comprising the Microprocessor Set, twelve other available standard TTL parts are required. These parts serve to:

- Generate the clocks
- Latch and gate input signals
- Latch and gate output signals

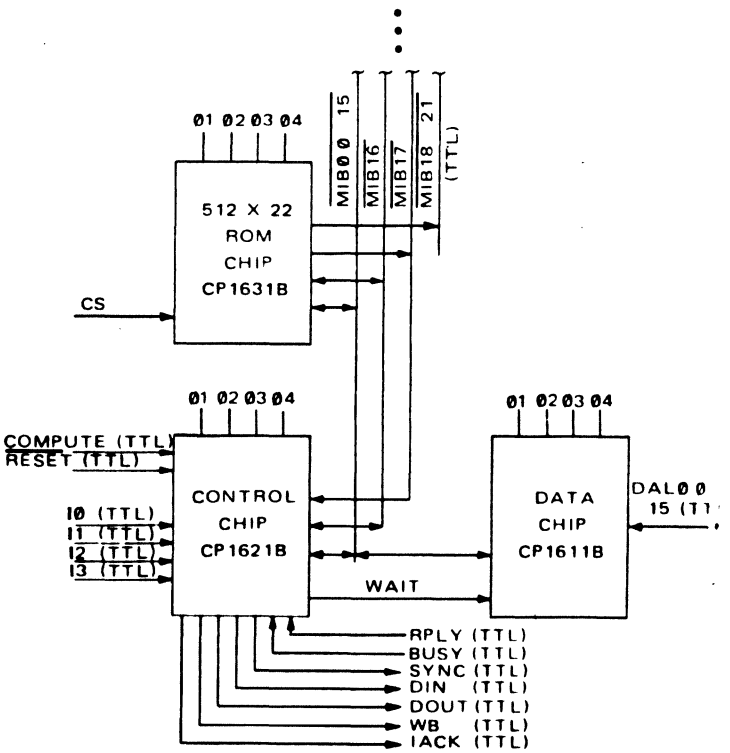


Figure 4-18. MCP1600 Microprocessor Block Diagram

The three chips that make up the MP1600 microprocessor set are contained in standard 40-pin DIPS with pin assignments as listed in Tables 4-9, 4-10, and 4-11.

Table 4-9. Data Chip (CP1611B) Pin Assignments

PIN NO.	SIGNAL	PIN NO.	SIGNAL	PIN NO.	SIGNAL	PIN NO.	SIGNAL
1	Ø3	11	DALØ8	21	Ø2	31	$\overline{\text{MIBØ7}}$
2	VBB	12	DALØ9	22	WAIT	32	$\overline{\text{MIBØ6}}$
3	DALØØ	13	DAL1Ø	23	$\overline{\text{MIB15}}$	33	$\overline{\text{MIBØ5}}$
4	DALØ1	14	DAL11	24	$\overline{\text{MIB14}}$	34	$\overline{\text{MIBØ4}}$
5	DALØ2	15	DAL12	25	$\overline{\text{MIB13}}$	35	$\overline{\text{MIBØ3}}$
6	DALØ3	16	DAL13	26	$\overline{\text{MIB12}}$	36	$\overline{\text{MIBØ2}}$
7	DALØ4	17	DAL14	27	$\overline{\text{MIB11}}$	37	$\overline{\text{MIBØ1}}$
8	DALØ5	18	DAL15	28	$\overline{\text{MIB1Ø}}$	38	$\overline{\text{MIBØØ}}$
9	DALØ6	19	VSS	29	$\overline{\text{MIBØ9}}$	39	VDD
10	DALØ7	20	Ø4	30	$\overline{\text{MIBØ8}}$	40	Ø1

Table 4-10. Control Chip (CP1621B) Pin Assignments

PIN NO.	SIGNAL	PIN NO.	SIGNAL	PIN NO.	SIGNAL	PIN NO.	SIGNAL
1	Ø3	11	$\overline{\text{MIB16}}$	21	Ø2	31	$\overline{\text{MIBØ7}}$
2	VBB	12	REPLY	22	VCC	32	$\overline{\text{MIBØ6}}$
3	I3	13	WAIT	23	$\overline{\text{MIB15}}$	33	$\overline{\text{MIBØ5}}$
4	I2	14	DOUT	24	$\overline{\text{MIB14}}$	34	$\overline{\text{MIBØ4}}$
5	I1	15	WB	25	$\overline{\text{MIB13}}$	35	$\overline{\text{MIBØ3}}$
6	IØ	16	IACK	26	$\overline{\text{MIB12}}$	36	$\overline{\text{MIBØ2}}$
7	$\overline{\text{MIB17}}$	17	SYNC	27	$\overline{\text{MIB11}}$	37	$\overline{\text{MIBØ1}}$
8	BUSY	18	DIN	28	$\overline{\text{MIB1Ø}}$	38	$\overline{\text{MIBØØ}}$
9	COMPUTE	19	VSS	29	$\overline{\text{MIBØ9}}$	39	VDD
10	$\overline{\text{RESET}}$	20	Ø4	30	$\overline{\text{MIBØ8}}$	40	Ø1

Table 4-11. Microm Chip (CP1631B) Pin Assignments

PIN NO.	SIGNAL	PIN NO.	SIGNAL	PIN NO.	SIGNAL	PIN NO.	SIGNAL
1	Ø3	11	$\overline{\text{MIB16}}$	21	Ø2	31	$\overline{\text{MIBØ6}}$
2	VBB	12	$\overline{\text{MIB17}}$	22	VCC	32	$\overline{\text{MIBØ5}}$
3	NC	13	$\overline{\text{MIB18}}$	23	CHIP SELECT	33	$\overline{\text{MIBØ4}}$
4	NC	14	$\overline{\text{MIB19}}$	24	NC	34	$\overline{\text{MIBØ3}}$
5	NC	15	$\overline{\text{MIB2Ø}}$	25	NC	35	$\overline{\text{MIBØ2}}$
6	NC	16	$\overline{\text{MIB21}}$	26	$\overline{\text{MIB11}}$	36	NC
7	$\overline{\text{MIB15}}$	17	NC	27	$\overline{\text{MIB1Ø}}$	37	$\overline{\text{MIBØ1}}$
8	$\overline{\text{MIB14}}$	18	NC	28	$\overline{\text{MIBØ9}}$	38	$\overline{\text{MIBØØ}}$
9	$\overline{\text{MIB13}}$	19	VSS	29	$\overline{\text{MIBØ8}}$	39	VDD
10	$\overline{\text{MIB12}}$	20	Ø4	30	$\overline{\text{MIBØ7}}$	40	Ø1

4.3.1 SYSTEM COMPONENTS.

The main functional components of the CPU microprocessor are physically partitioned into three kinds of devices. The logical partitioning of the system, however, encompasses five areas as follows:

1. Processing and Data Handling. This is generally handled on the CP1611B Data Chip.

2. Next Instruction Address Generation. This is generally handled on the CP1621B Control Chip.

3. Microinstruction Storage. These are contained on one or more CP1631B Microinstruction ROMs.

4. Microinstruction Bus. This bus connects the three types of devices together and provides a path for the microinstructions to flow from the microinstruction ROM to both the Control and the Data Chip.

5. Data Access Bus. This bus provides access from the MCP1600 microprocessor set to the outside world. Lines comprising this data access bus come from all three of the chip types. Lines containing the address and data come from the Data Chip, control lines defining the state of the data access come from the Control Chip and user programmable control lines come from the Microinstruction ROM.

The MCP 1600 operates on a four phase clock system. While there are some variations, the general use of each phase period is:

- Ø1 Instruction Access
- Ø2 Data Access
- Ø3 Execution
- Ø4 Data Update

4.3.1.1 REGISTERS.
The various registers in the CPU microprocessor are described in the following paragraphs.

Register File. The Register File consists of 26, 8 bit registers which provide RAM data storage for the MCP1600 processor set. The register file has 2 output ports and 1 input port. Fourteen of the registers of the file can be directly addressed by the A and/or B register designators of a microinstruction. Additionally, the 16 top-most registers of the register file may be considered as register pairs and can be addressed by the G register to permit operation on full words of data. The Register File is on the Data Chip.

The A and the B output ports of the register file feed into the ALU.

G Register. The G register is a pointer register on the Data Chip that describes the currently accessed linked consecutive pair of registers in the register file. Note that, when the G register is being used, access to the register file is from the top down. This is opposed to the access to the register file when only the A and B fields are being used as designators; in which case, it is from the bottom up. This register is loaded by IW and LGL instructions. "Input Word" instruction loads the G Register from the DAL bus as specified by the "b" field of the instruction. "Load G Low" instruction loads the G Register Ra.

The first (or lowest) 14 registers of the Register File are addressable only from the MIR register. The top 12 registers are addressable only from the G Register. The middle 4 registers are addressable from either the G or the MIR registers. Figure 4-19 describes the addressing conventions of this file. It is helpful to note that if the a or b fields of the MIR are 1 or 0, then this enables G Register addressing.

Some Examples

Assume G = 4, a = 0, b = B
Then one operand (A Port) will be from G'8'
The other operand (B Port) will be from R'B'

Assume G = 0, a = 7, b = 1
Then one operand (A Port) will be from R'7'
The other operand (B Port) will be from G'1'

Assume G = b, a = 1, b = 0
Then one operand (A Port) will be from G'D'
The other operand (B Port) will be from R'D'

Assume G = 3, a = 6, b = A
Then one operand (A Port) will be from R'6'
The other operand (B Port) will be from R'A'

Assume G = 5, a = 0, b = 1
Then one operand (A Port) will be from G'5'
The other operand (B Port) will be from G'6'

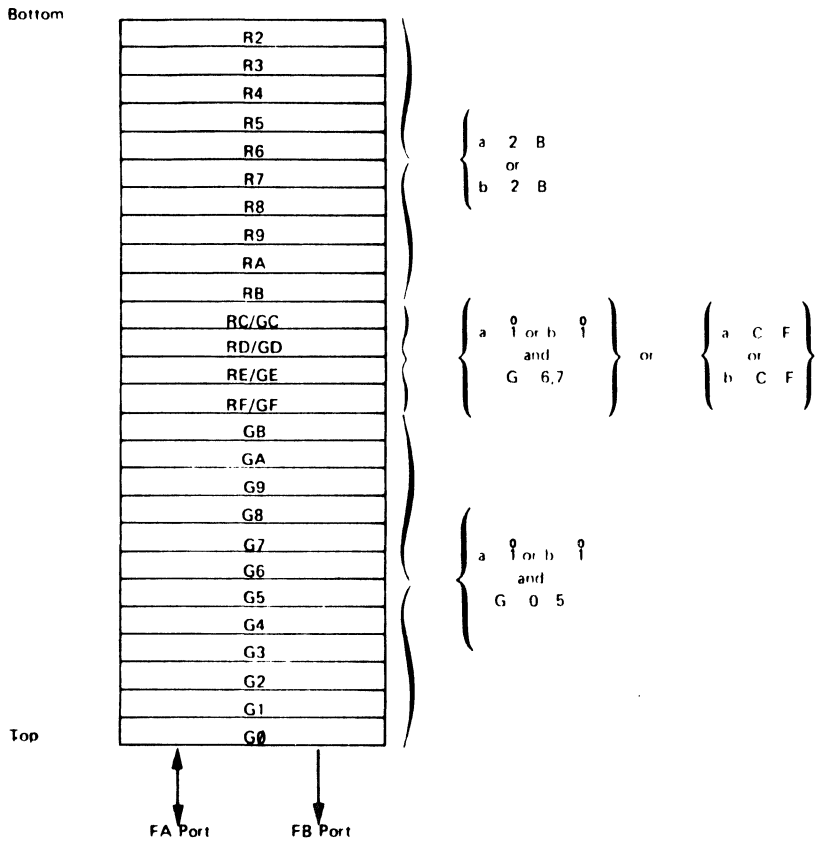


Figure 4-19. Register File

ALU Status Bits. The ALU provides 4 status bit outputs which reflect the result of each 8 bit output which in turn reflects the result of each 8 bit operation. The ZB and NB bits are updated after every ALU operation. C4 and C8 are updated only during Arithmetic or Shift operations. These bits may be tested by Jump, but their primary function is to pass result data from the first cycle of a word instruction to the second cycle. The available status bits are:

ZB: Set if the result of a Byte or Word operation is zero; cleared otherwise.

NB: Set if the high-order bit of a Byte or Word operation is a one; cleared otherwise (except for SRW and SRWC).

C4: Set if Carry Out of Position 3 is a one; cleared otherwise. Updated only on Arithmetic operations. This status bit is used mainly for decimal arithmetic corrections.

C8: Set if Carry Out of Position 7 is a one; cleared otherwise. (Note that this status bit is not set to borrow for subtract as is the case with the C Flag.) Also set if the shifted off bit of a Shift operation is a one; cleared otherwise.

Condition Flags. The Condition Flags consist of 4 latches which can reflect the status of the previous ALU results. The updating of these flags can be selectively enabled or disabled at the discretion of the microprogrammer. The condition flags are updated with odd-numbered instruction opcodes in the range of 8Ø-EF.

Z Flag: Set if the result of a Byte or Word operation is zero, cleared otherwise.

N Flag: Set if the high-order bit of the result of Byte or Word operation (except for SRW and SRWC) is a one; cleared otherwise. (Note that this is the complement of the sign of the result if overflow occurs.)

C Flag: Monitors the carry, borrow and shifted off bits as follows:

Add and Increment: Set if there is a carry from the most significant bit of the Byte or Word result; cleared otherwise.

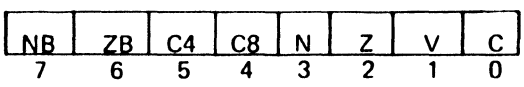
Subtract and Decrement: Set if there is a borrow (complement of carry) from the most significant bit of the Byte or Word result; cleared otherwise.

Shift: Set if the bit shifted off in a left or right shift is a one; cleared otherwise.

The C Flag is not affected for operations (other than those listed above) even if the other flags are updated.

V Flag: Set if there is an arithmetic overflow on Arithmetic operations, cleared if there is no overflow and on Non-arithmetic operations. On Add operations, overflow occurs when the sign of the two operands are the same and the sign of the result is different. On Subtract operations, overflow occurs when the signs of the two operands are different and the sign of the result is different from the operand in Ra (V=C7 ⊕ C8).

The format of the ALU status and flag register word is shown below:



MIR Register. The 16 bit MIR register holds the least significant 16 bits of the 22 bit microinstruction currently being executed. The MIR register is loaded every Ø1 unless the currently executing microinstruction is of the 2 cycle variety. In this case, the register is not clocked on the second occurrence of Ø1.

4.3.1.2 INSTRUCTION ADDRESS GENERATION.
This paragraph describes the registers and arrays used in generating the next microinstruction address.

Location Counter Register. The Location Counter (LC) is an 11 bit register which holds the address of the next microinstruction to be accessed. Under normal conditions, the LC is incremented by one after each access of a microinstruction. The LC can be altered by other than one through execution of a Jump instruction or a Return from Sub-routine (RFS) instruction or by the invocation of a PTA translation.

Return Register. The Return Register (RR) holds an 11 bit address and permits a subroutine depth of one in the microprogram. When a subroutine jump is indicated (MIB 16 = 1), the return register stores the incremented contents of the LC. Contents of the return register can be transferred to the LC under the control of a RFS - (Return From Subroutine).

Translation State Register. The Translation Register (TR) is a 16 bit register which holds the data presently being input to the PTA. Data on the Microinstruction Bus (MIB) is used to load the Translation Register. Note that only the upper or lower half of the Translation Register can be fed into Array 3 at one time.

Translation Arrays. There are four arrays on the control chip that control the generation of the next microinstruction address. They generate new inputs into the LC register, as a function of the LC register contents, the contents of the translation register, interrupts, and other miscellaneous inputs. The detailed operation of these arrays is discussed in Section 4.3.3, PROGRAMMABLE TRANSLATION ARRAY (PTA).

4.3.1.3 MICROINSTRUCTION BUS.

The Microinstruction Bus serves to interconnect the three required parts of the MCP1600 Microprocessor set. The bus is organized into 4 distinct sections.

MIB00 - MIB15 carry proper microinstruction data from the microinstruction ROM chip to both the control chip and the data chip. These lines may also carry data between the data chip and the control chip under certain conditions.

MIB 16 controls the subroutine jump operation. When MIB16 is set on a jump instruction, it causes the incremented contents of the LC register to be placed into the Return Register.

MIB17, if set, causes the PTA on the Control Chip to perform a Read Next Instruction translation.

MIB18 - MIB21 are TTL level outputs that can be programmed by the user as his needs dictate. They are not used by the MCP1600 Microprocessor set to control its operations. They are provided for the convenience of the user in order that he may interface to the MIB bus or control devices directly from the microprogram level.

4.3.1.4 DATA ACCESS BUS.
The data access consists of three sections:

DAL00 - DAL15 carry address and data between the data chip and the outside world.

Outside Control Lines. There are five TTL level Outgoing Control Lines that inform the outside world of the present state of the MCP1600 set. These signals include Sync, Input Instruction, Output Instruction, Wait and Interrupt Acknowledge.

Incoming Control Lines. There are 8 TTL level Incoming Control Lines. These incoming control lines inform the MCP1600 Microprocessor set of the state of affairs in the outside world. There are four interrupt lines, a Compute line, a Reset line, a Reply line and a Busy line.

With these lines, the MCP1600 can control a wide variety of peripheral devices.

4.3.1.5 MICROINSTRUCTION STORAGE.
The microinstruction ROM (MICROM) is a 512 x 22-bit word, high speed ROM which stores the instructions of the microprogram. The transfer of addresses into the chip and the microinstruction out of the chip are performed over the MIB. Address is received from push-pull drivers in the Control Chip on Ø2. The decoding takes place on Ø3. On Ø4 the selected microinstruction is internally accessed and the MIB is precharged. The accessed microinstruction is placed on MIB15-MIBØ for transfer to the Data Chip and Control Chip during Ø1.

4.3.2 CPU OPERATIONS.
This section describes the internal workings of the Data Chip, the CPU of the MCP1600 system. There are two important and distinct partitions to the processor. The first partition,

called the Data Chip, provides classical stored program processing. The Data Chip consists of:

- A register file
- An arithmetic logic unit (ALU)
- A Microinstruction register
- A register file address decoder
- A control signal generation function
- Condition flags
- Jump decoding
- Input/Output gating

The data chip is responsible for data manipulation as a result of instruction execution.

The second partition, the Control Chip, generates address data that directs the accessing of the next microinstruction to be executed. This address generation mechanism provides the MCP1600 with its unique emulation capabilities. While the Data Chip is executing the presently fetched microinstruction, the Control Chip is performing a transformation upon the presently executing macroinstruction to determine the address from which to fetch the next microinstruction to be executed.

The description of the processor proceeds by first outlining the operation of the Data Chip (CPU) and then in the following section outlining the operation of the Control Chip. It is important to note that the two chips are not synchronized except by a common clock, and the fact that the contents of the MI register on the Control Chip and the contents of the MIR register on the Data Chip are the same. Very few control lines are passed between the Data Chip and the Control Chip.

4.3.2.1 CPU SEQUENCING.

First assume that the machine has been properly reset and has just entered the compute mode. The Control Chip causes a

microinstruction to be fetched from the MICROM to be placed on the MIB bus. (See Figure 4-20.) Assuming that the clock is now on the leading edge of Ø1, the contents of the MIB bus are gated into the MIR register on the Data Chip and the MI register on the Control Chip.

At the end of Ø1 (trailing edge of the clock), the contents of the MIR register have settled. The contents on the MIR register are held fixed until the next occurrence of Ø1 (four phase times).

At the beginning of Ø2, the contents of the MIR register are presented to the various different functions of the CPU. The A and B fields are presented to the address decoder, as is the G register. The contents of the MIR register's C field and OP field are presented to the master control function of the Data Chip. During Ø2 and Ø3, the master control function decodes the C field and generates signals which control the gating of data to the various functions of the Data Chip. In Ø2, the address decoder is interpreting the contents of A field and the B field and accessing the appropriate register in the register file. Also during Ø2, the condition codes reflecting the result of the last ALU operation become valid. They are presented to the FLAGS register which retains them for examination on the next occurrence of Ø4. Figure 4-21 illustrates Ø2 data paths.

Phase 3 is the register access phase. During Ø3, data is presented to the ALU by the A and the B output ports of the register file. Data may also be presented to the ALU B port by means of the literal path from the MIR register. Note the literal path (whose contents are the A and B field of the MIR) bypasses the address decoding logic and the register file. Also during Ø3, the contents of the A and B register file output ports may be presented to the Data Access on line DALOO

-DAL15. During this phase time, the contents of the register file A port may be presented to the flag register. Finally, during this phase time, the contents of the register files A and B ports may be presented to the gating which controls access to the MIB bus. Figure 4-22 illustrates Ø3 data paths.

During Ø4, the output of the ALU becomes valid. Input to the ALU may occur from the data bus in Ø4. That is, data presented on lines DALOO-DAL15 may be presented to the gating structure which controls the output of the ALU. Finally, during Ø4, the data available from the Data Access may be presented to the gating structure that controls access to the MIB bus. Figure 4-23 illustrates the Ø4 data paths. Note that, during Ø4, the jump control takes place. If the jump decoder indicates that all the proper jump conditions are met and if JXX is asserted, then MIB15 will cause the Control Chip to effect a jump on the next clock cycle.

As the next Ø1 clock occurs to start the next cycle of the instruction/execution, the data that was presented to the gating structure that controls access to the A input port of the register file is gated or not gated, depending upon a signal from the Master Control, into the register file. Figure 4-24 illustrates this path. If the instruction contained in the MIR register is one that requires only one clock cycle for execution, then a new instruction is fetched from the MICROM according to the address generated by the Control Chip and execution begins as in Figure 4-20.

If however, the instruction requires two cycles for execution, things proceed differently. At the beginning of Ø1 of the second cycle, the contents of the MIR are preserved and no new data is gated in. The low order bits of both the 'A' and 'B' fields are complemented to access adjacent slots in the register

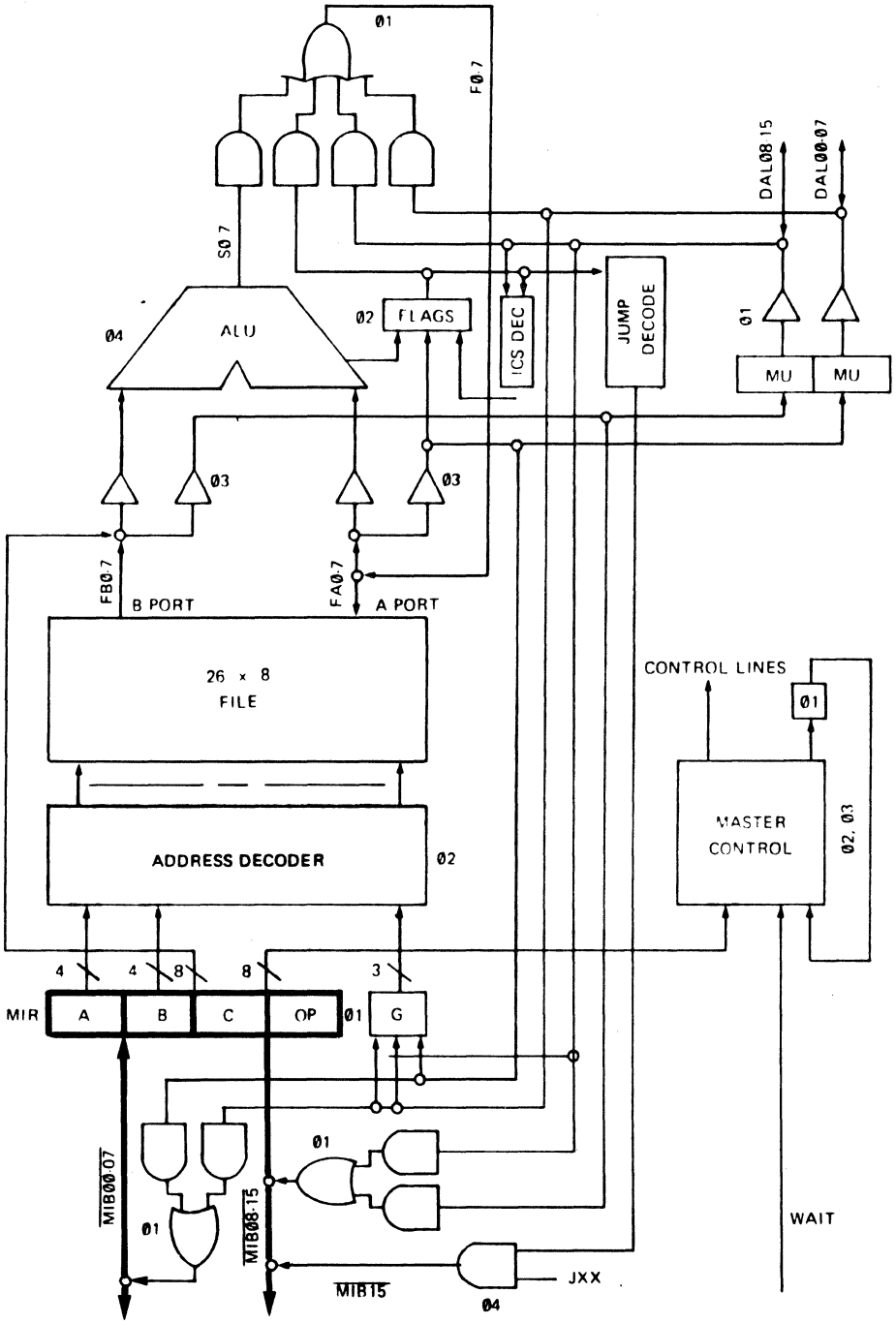


Figure 4-20. Ø1 Data Paths

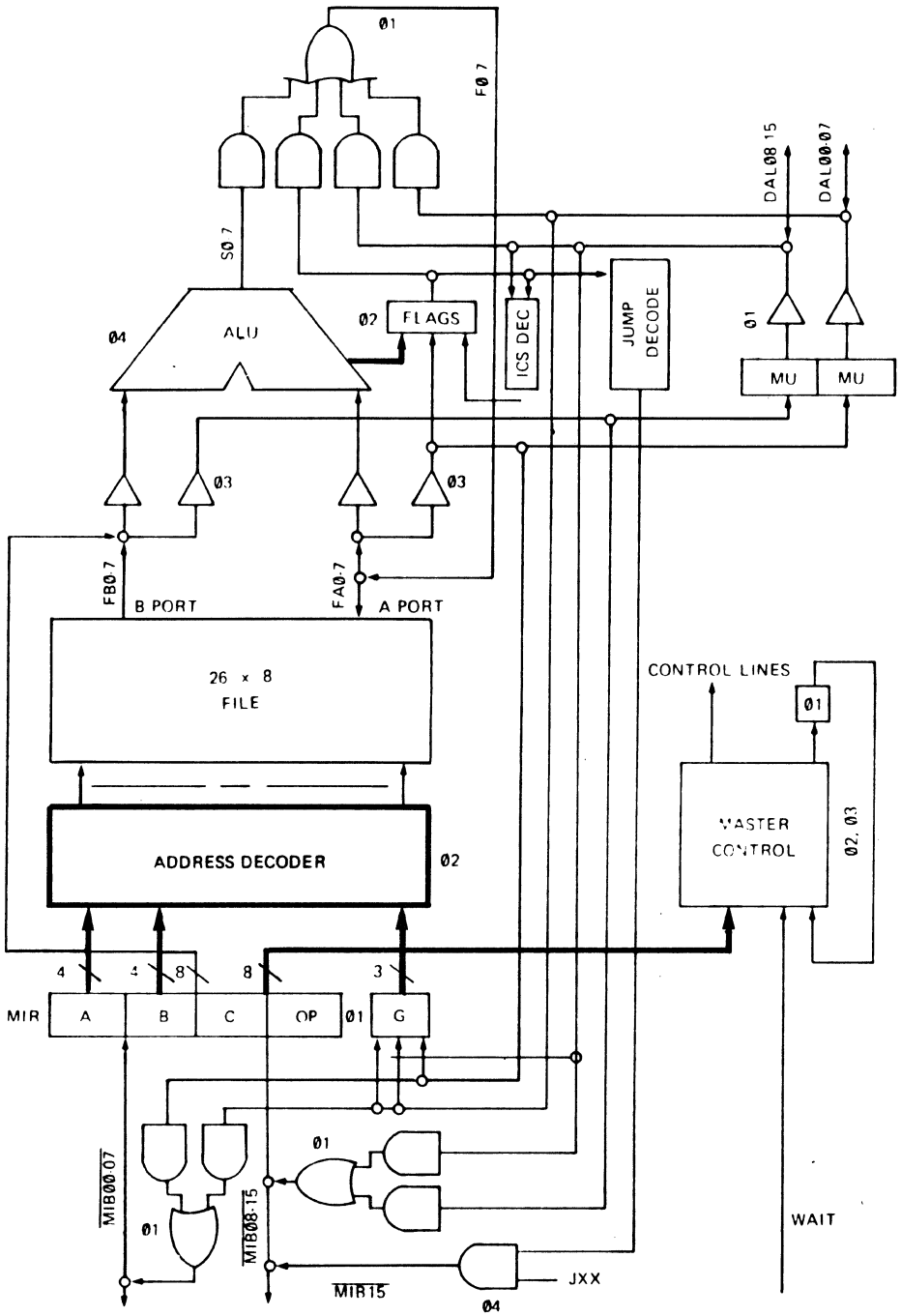


Figure 4-21. Ø2 Data Paths

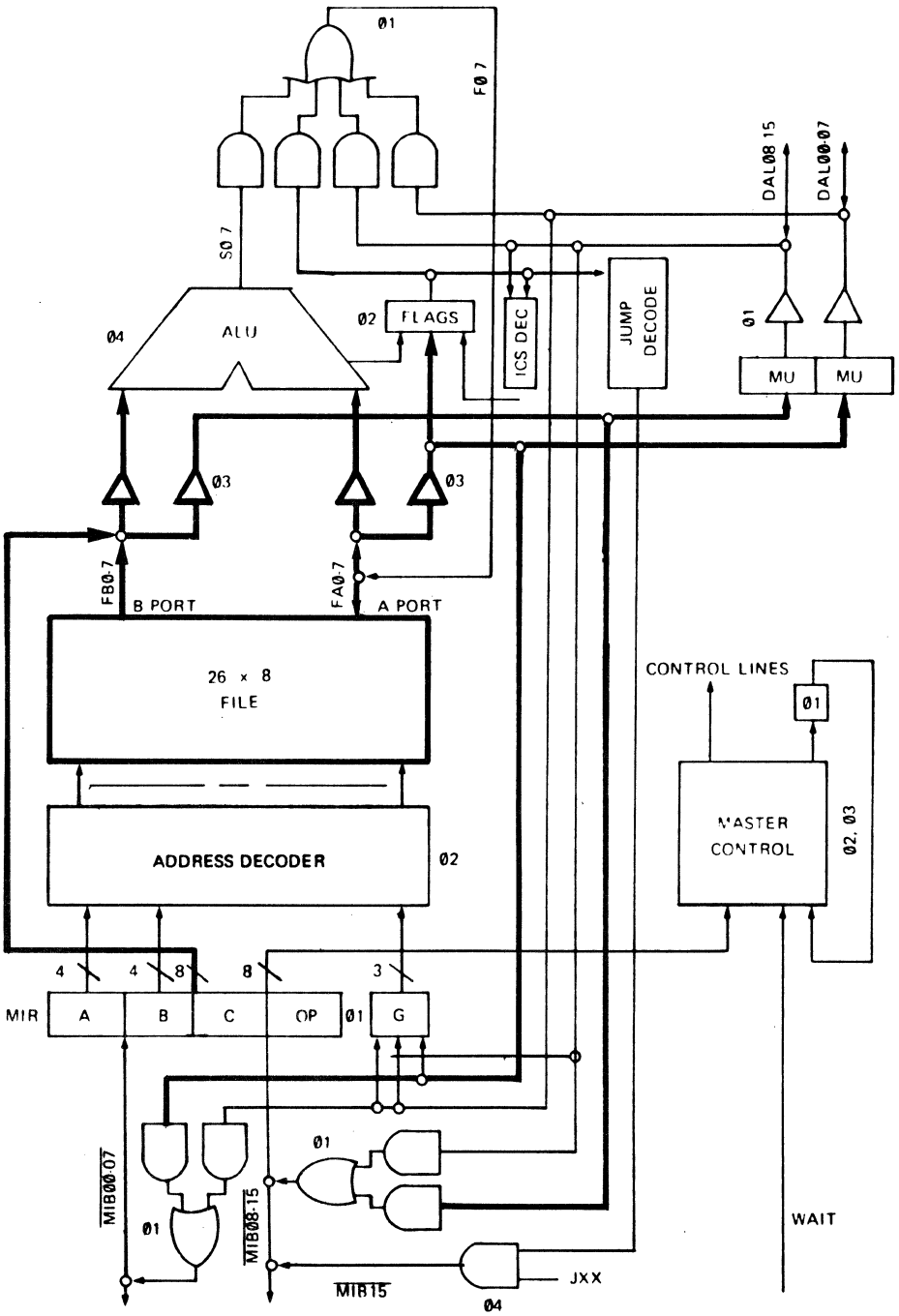


Figure 4-22. Ø3 Data Paths

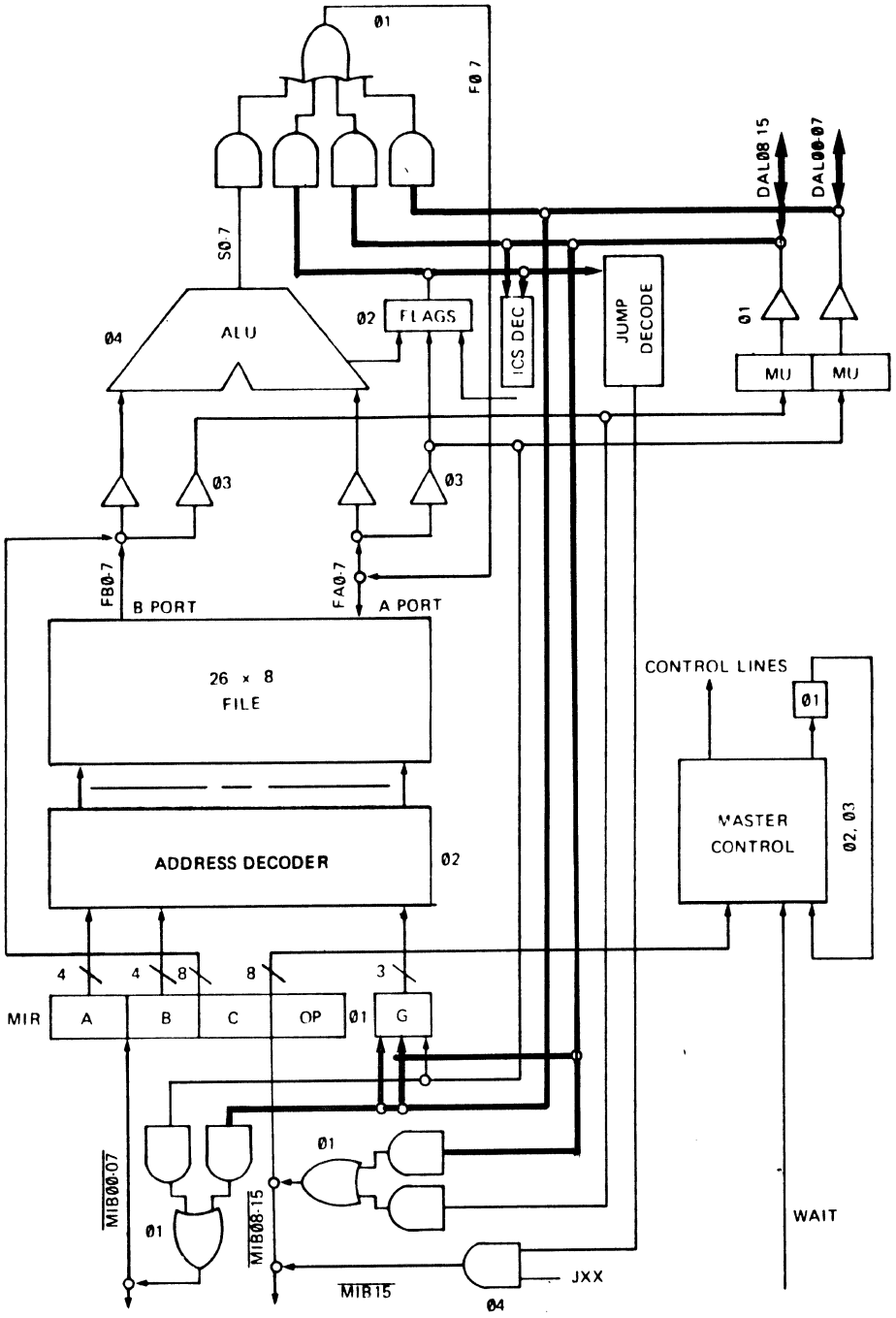


Figure 4-23. Ø4 Data Paths

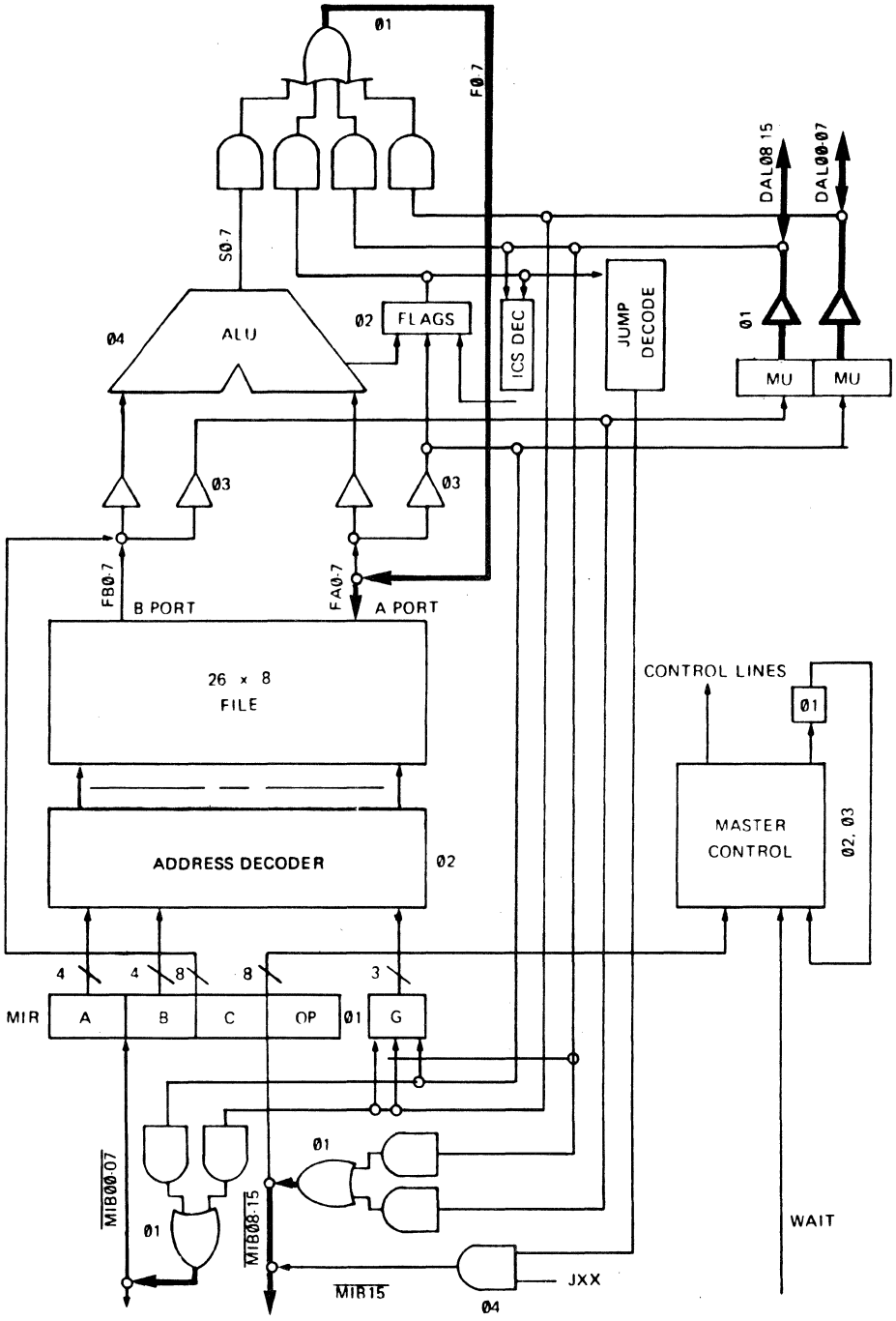


Figure 4-24. Ø4 Data Paths Second Cycle

file to those previously addressed in the first cycle. Also during this phase, the CPU may output data to the MIB bus. Figure 4-24 illustrates this path.

Other than these exceptions, two cycle instructions proceed as do one cycle instructions.

4.3.2.2 CONTROL LINES.

During Ø2 and Ø3, the Master Control function is decoding the new contents of the MIR register in order to generate signals to control the data transfers described above. These control signals are generated on the basis of the contents of the C and OP fields of the MIR register. All of the control lines generated by the Master Control function are not described here because some of them serve timing and sequencing functions that are not necessary to the understanding of the CPU operation and how it affects the Data Access and Microinstruction Bus. Several are described in detail in the following paragraphs and in Figure 4- 25.

Register Load. This control line determines whether or not the output of the ALU will be gated back into the register file input port. It is invoked by register to register format instructions.

Load Flags. This line controls whether or not the output of the FLAGS register will be gated into the register file input port. It is generated by the Load Flags and Copy Flags instruction.

Load Hi Byte. This control line determines whether or not the contents of the DAL08-15 will be gated into the register file input port. This line is controlled primarily by the Input class of instructions.

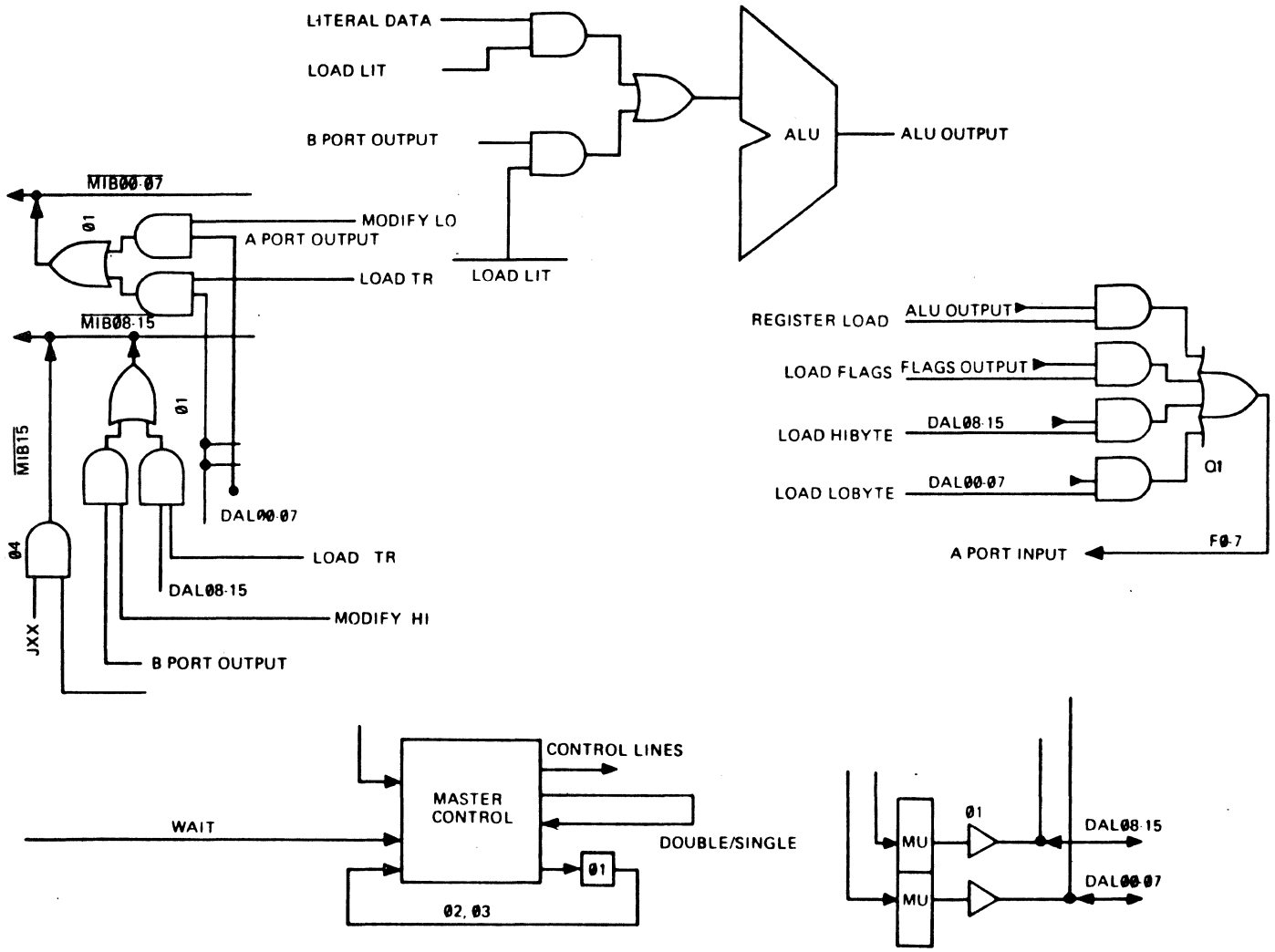


Figure 4-25. Major Control Lines

Load Lo Byte. This control line determines whether the contents of DAL-07 will be gated into the register file input port. It is generated by input class instructions. Note that this signal and preceeding signal are generated by the specification in the B field of the appropriate input instruction. Note also that one can input the low order byte, the hi order byte, or both bytes. This latter case is an input Word Instruction.

Modify Lo. This control line is invoked by the Modify instruction and causes the output of the A register file port to be gated on to the low order 7 bits of the microinstruction bus. This control line is also invoked by the Load Translation Register instruction.

Modify Hi. This control line is the high order analog of the previously described control line.

JXX. This control line is invoked by the jump instruction. It is made hi during the first phase four of the jump instruction, if the jump instructions have been met. It causes the contents of the MI register on the control chip to be placed into the LC register.

Load LIT. This instruction controls the multiplexer which selects the B input to the ALU. The ALU B input port may be fed from either the literal field of the MIR register or the B port output of the register file. This control signal is set by the literal class of instructions and causes the B input to the ALU to be selected from the literal path.

Double/Single. This control line is generated by the Master Control for use by itself. It is invoked by instructions that require two cycles for execution. In the main, it controls

whether or not a new instruction is gated into the MIR register at the next occurrence of Ø1.

Load TR. This control line is invoked by the Input Word instruction if either bits 4 or 5 of the B field are set. These bits being set will cause the data on the Data Access to be placed into the translation register on the Control Chip. It is in this fashion that new macroinstructions are fetched from the user memory for decoding.

4.3.3 PROGRAMMABLE TRANSLATION ARRAY (PTA).

This section describes the operation of the MCP1600 Microprocessor set's most unique feature, the Programmable Translation Array. The Programmable Translation array serves to generate new microinstruction fetch addresses as a function of several parameters. These parameters are those which are normally considered during the decoding of a macroinstruction. While the Programmable Translation Array was designed specifically to eliminate most of the overhead of macroinstruction translation, it is useful for other purposes. For example, a data driven processor could determine the next operation to be performed on a data structure as a function of the present data and the present operation.

Briefly, the Programmable Translation Array consists of eight elements. They are:

- Location Counter
- Array 1
- Array 2
- Array 3
- Translation Register
- Array 4
- Translation State Register
- Return Register

These elements give the capability of calculating micro-instruction addresses other than sequentially, or as specified by a jump instruction. Figure 4-26 depicts their interconnection. The mapping (or new Location Counter value) that the PTA generates, when invoked, is a function of the value of the Location Counter, the contents of the Translation Register, the interrupts, and the state of the RNI line.

4.3.3.1 PROGRAMMABLE TRANSLATION ARRAY COMPONENTS.
This Section describes the components of the Programmable Translation Array and their functions.

Location Counter. It is the function of the Location Counter to hold the address of the next microinstruction. The Location Counter is 11 bits wide and has outputs to MIB00-MIB10 and to Array No. 1. It has inputs from the Master Control function, Array No. 4 and the Return Register. There is also an input to the Location Counter from an incrementer.

Array No. 1. Array No. 1 is an 88 element array with 23 inputs. There are 11 true inputs and 11 complemented inputs from the Location Counter and the RNI bit (MIB17). Figure 4-27 illustrates the organization of Array No. 1. Figure 4-28 illustrates the concept, as opposed to implementation of a typical gate, one of 88, that make up Array No. 1. The outputs of these gates make up the bulk of the inputs of Array No. 2.

The purpose of Array No. 1 is to perform a selection function. When the inputs to Array No. 1 match one (or more) of the word values (addresses), the output associated with that word is asserted. This array is programmed with 6 hexidecimal digits as shown in Figure 4-29. Transistors are placed in the array for 1 bits in the data specification. To program a group of addresses, the bits of the address to be left out are specified

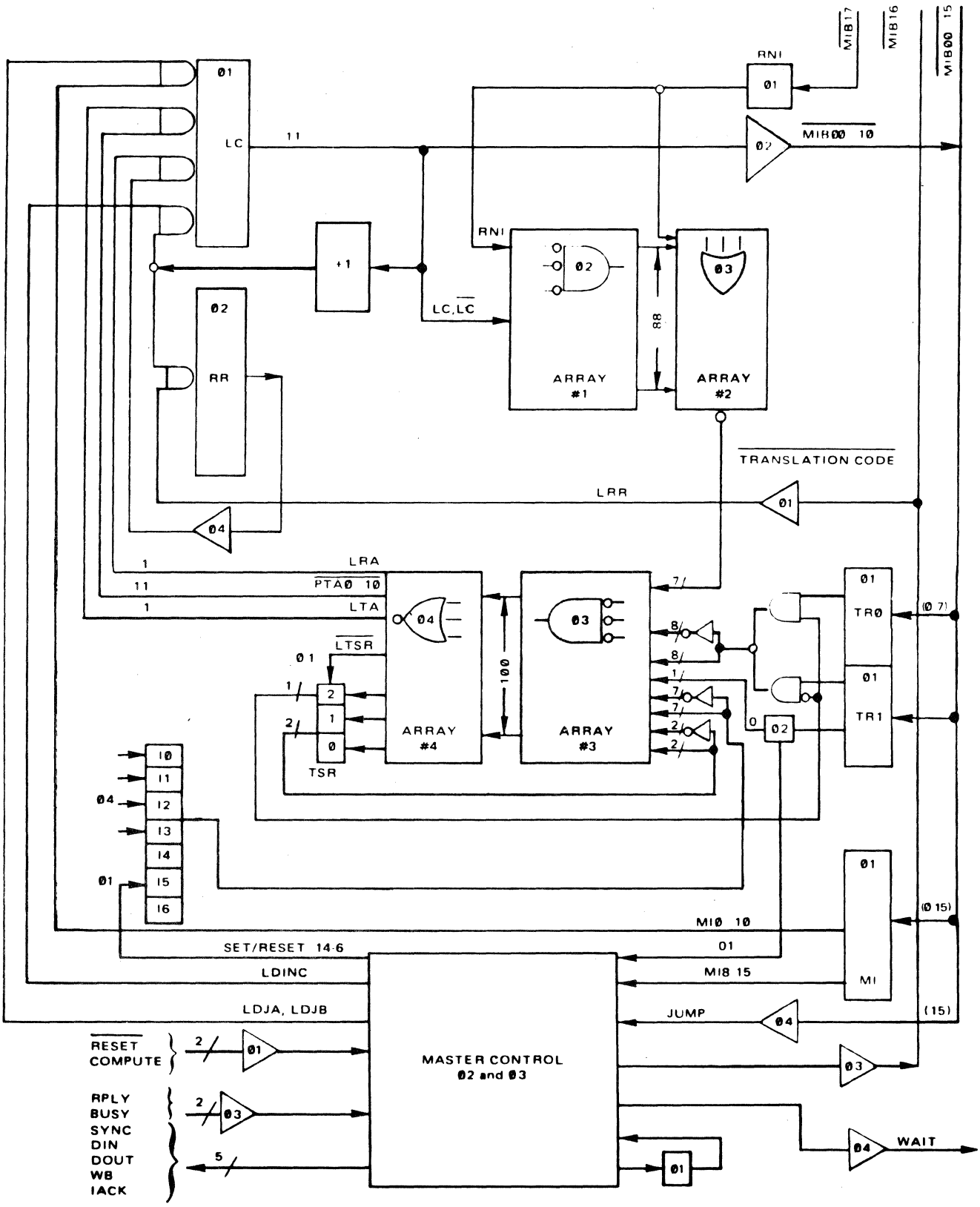


Figure 4-26. PTA Component Interconnections

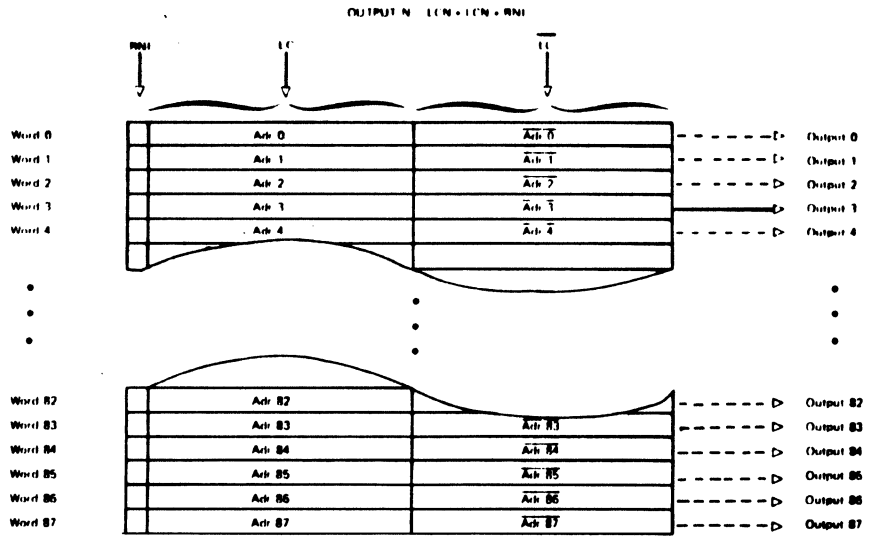


Figure 4-27. Array No. 1 Organization

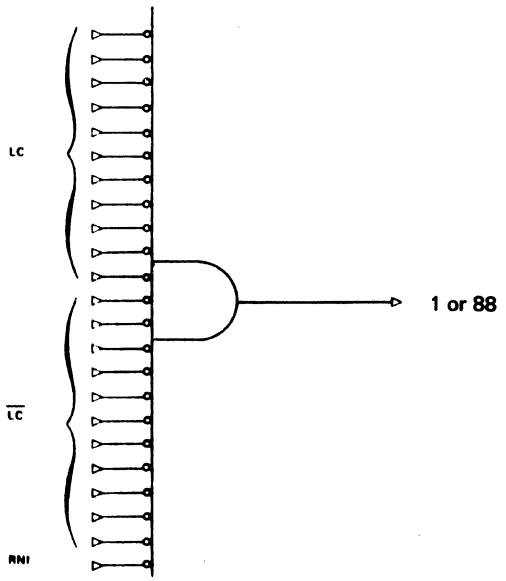


Figure 4-28. Typical Gate - Array No. 1

as Ø bits for the true and complement address inputs. This produces a 'DON'T CARE' effect for those bits and the result will always be a match. This lets the user match on modulo address.

The RNI serves to inhibit the output of Array No. 1. The reason for this is described later.

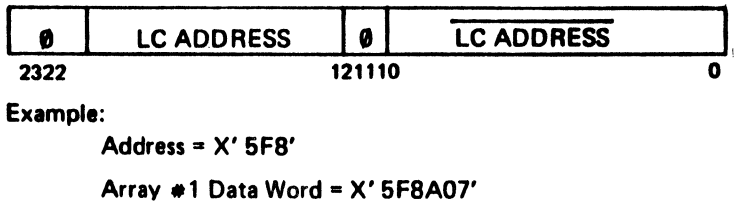


Figure 4-29. Array No. 1 Data Specification

Array No. 2. Array No. 2 is an ORed array whose outputs are negated. The array has 89 inputs, 88 from Array No. 1 and one from MIB17, the RNI line. The seven outputs from Array No. 2 represent a translation state code and become inputs into Array No. 3.

Figure 4-30 illustrates the organization of Array No. 2. Figure 4-31 illustrates in some detail, the conceptual structure of the gate and interconnect structure of Array No. 2. The small circles present where the lines would normally cross represent connections that can or cannot be made. It is via the making or not making of the connections with transistors that this array is programmed.

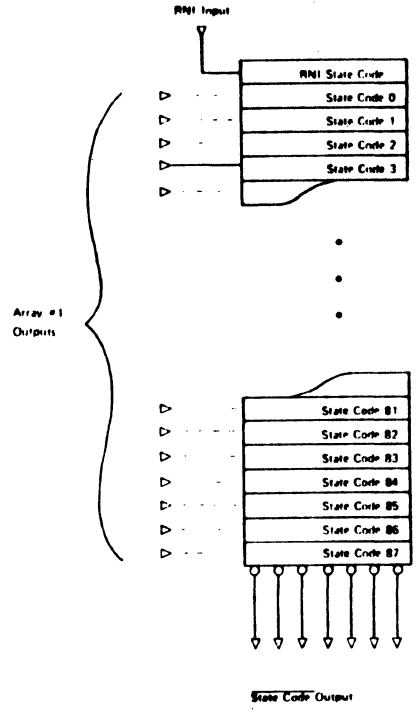
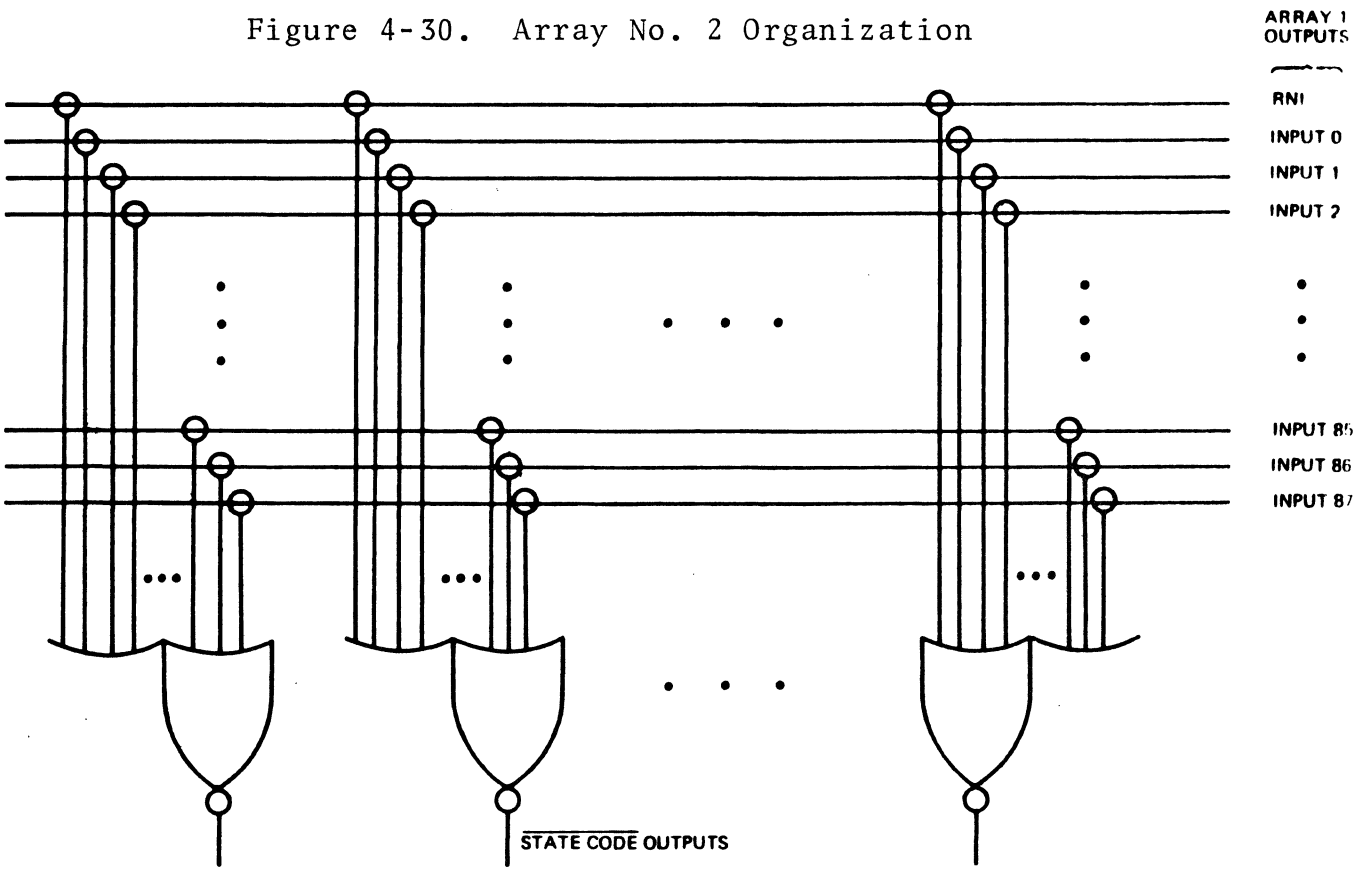


Figure 4-30. Array No. 2 Organization



Array #2 is driven by the inputs from Array #1. When one or more outputs of Array #1 are asserted, Array #2 will make the corresponding word available as its outputs.

The purpose of Arrays #1 and #2 is to provide a mapping between the present value of the Location Counter and a preselected set of translation state codes.

Figure 4-31. Array No. 2 Gate and Interconnect Structure

Due to electrical considerations, there are restrictions placed on the configuration of Array No. 2. Normally this array is programmed in a 4 of 7 code (4 of 7 outputs may be selected to be high, the other 3 must be low), or more ideally, to reduce capacitive loading, a 5 of 7 code. The tables 4-12 and 4-13 describe the various state code outputs available in both the 4 of 7 and the 5 of 7 schemes.

Table 4-12. Translation State Code Outputs--4 of 7

0F	2D	3C	56	69
17	2E	47	59	6A
1B	33	4B	5A	6C
1D	35	4D	5C	71
1E	36	4E	63	72
27	39	53	65	74
2B	3A	55	66	78

Table 4-13. Translation State Code Outputs--5 of 7

1F	3D	5B	6B	75
2F	3E	5D	6D	76
37	4F	5E	6E	79
3B	57	67	73	7A
				7C

The outputs of Array No. 2 represent some of the inputs to Array No. 3.

The RNI line (MIB17) deserves mention at this point. It is the purpose of the RNI line to force a particular user defined state code from the mapping represented by Arrays No. 1 and No. 2. This state code will be independent of the contents of the Location Counter.

Accordingly, RNI inhibits the outputs of Array No. 1 and invokes that state code selected by the user in Array No. 2 and makes it present at the output of Array No. 2.

The programming of Array No. 2 is illustrated in Figure 4-32. Transistors are put in the arrays for zero bits as per the data specifications.

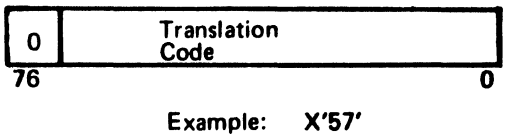


Figure 4-32. Data Specification

Array No. 3. Array No. 3 is a NANDed array, and has 42 inputs. In contrast to Arrays No. 1 and No. 2, the inputs come from a variety of sources. The inputs to this array constitute a direct interface to the real world at the macromachine level, as one of the inputs is the translation register. The inputs to Array No. 3 are:

- Seven from Array No. 2. (Translation Code)
- Sixteen from the Translation register. Eight are true data and eight are complemented data.
- Fourteen inputs from the interrupt latches. Again, both true and complemented data are present.
- Two from the translation state register. Again, both true and complemented data are used.
- One input, called the Q signal, not generally usable.

It is worth noting, that while there are 42 inputs in Array No. 3, they are not all present at the same time. The array is broken into two partitions. The first partition consists of words 0 through 15, and has as its inputs, the 14 bits from the interrupt latches. The second partition of the array, words 16 through 99 have in the same slots the 16 bits of inputs from the Translation Register.

The Figure 4-33 illustrates the structure of the first partition of the array. Figure 4-34 illustrates the second partition of the array. Figure 4-35 shows a conceptual picture of a typical gate in the array.

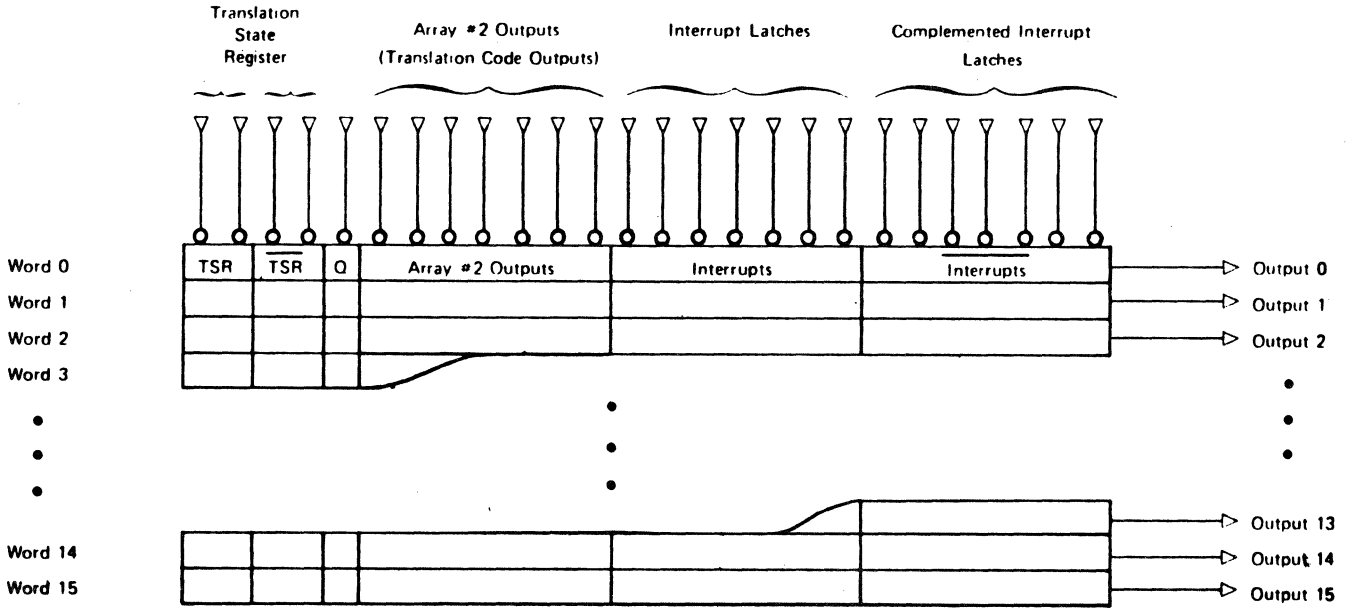


Figure 4-33. Array No. 3 Interrupt Organization

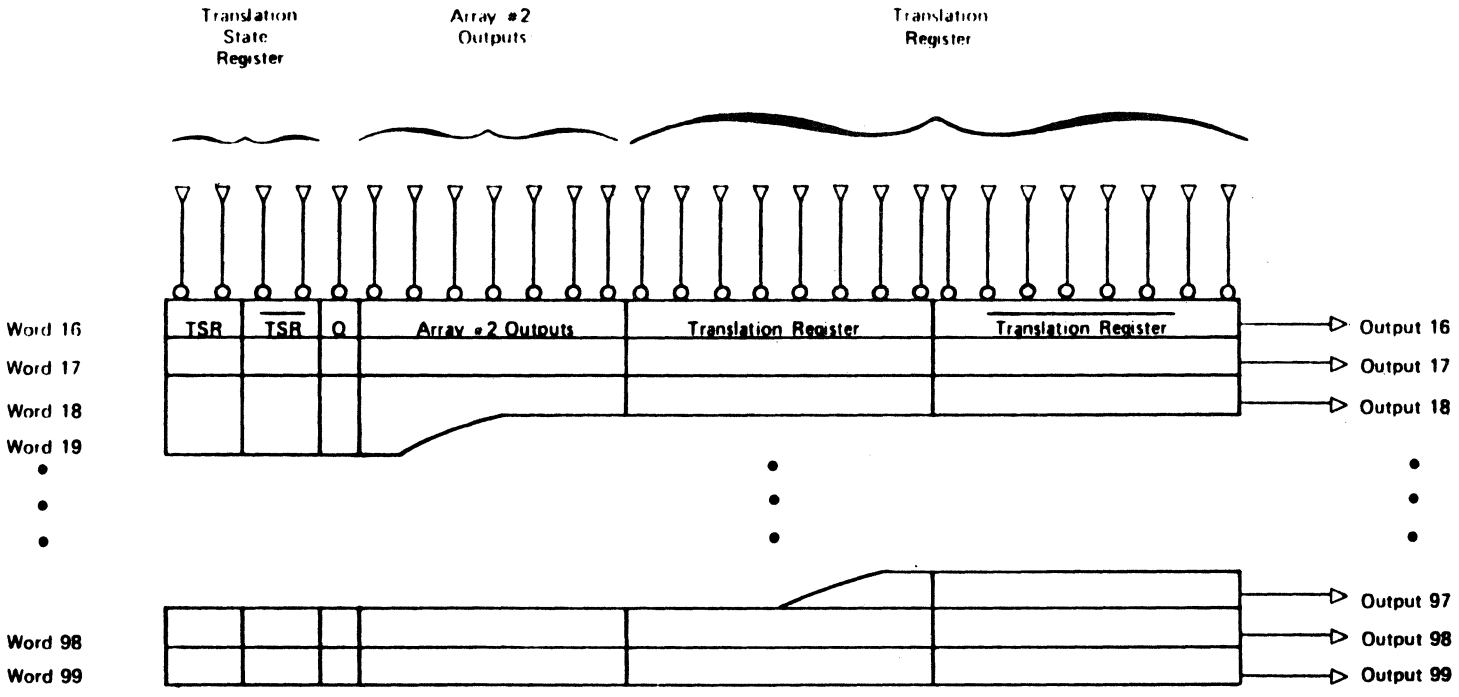


Figure 4-34. Array No. 3 Translation Register Organization

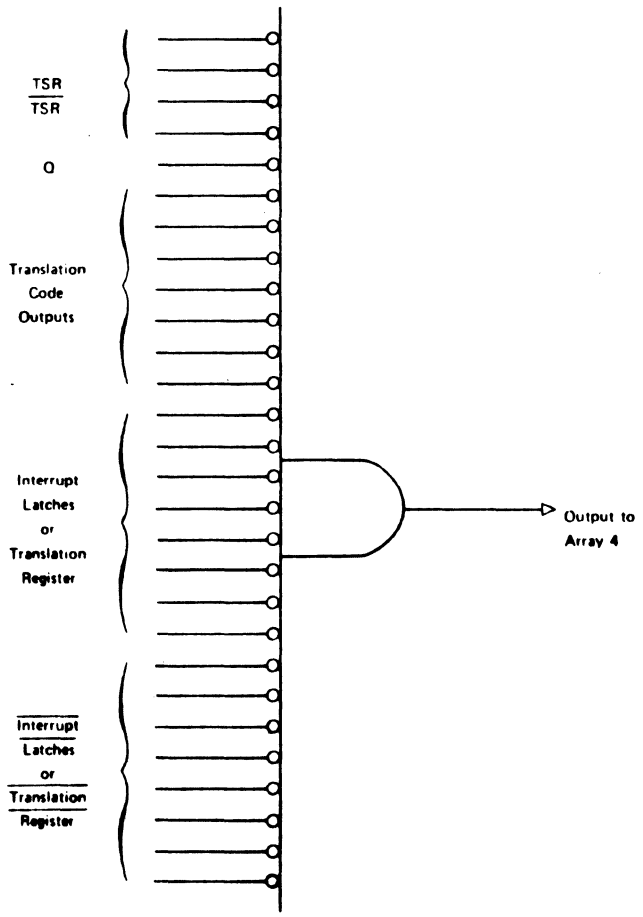
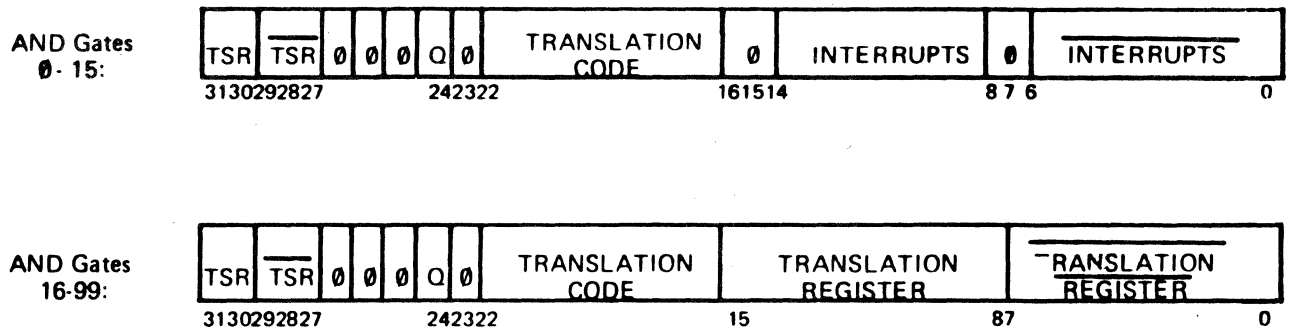


Figure 4-35. Typical Gate of Array No. 3

Array No. 3, being an ANDed array, will try to match the configuration of its inputs against one of the words that make up the array. If it finds a match, the output associated with that word will be asserted.

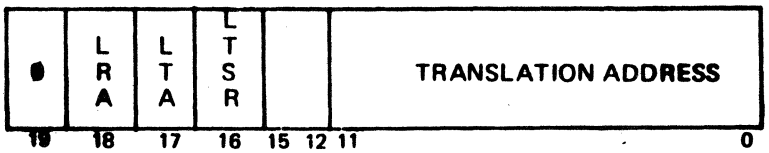
Array No. 3 is programmed with true data as eight hexadecimal digits. DON'T CAREs, which always result in matches, are created by placing both true and complemented bits to zero. Below is a data specification for the programming of Array No. 3.



Array No. 4. Array No. 4 is a NORed array with 100 inputs which come from Array No. 3. The outputs generated by Array No. 4 include address data to be fed into the Location Counter, data to be fed into the Translation State Register, and three control lines. Two control lines determine whether the Location Counter will be loaded from the Return Register or the output of the translation array. The third control line determines whether or not the value presented to the Translation State Register will in fact be loaded into the Translation State Register. Figure 4-36 illustrates the organization of this array. Note that not all the outputs are complemented. The two control signals LRA and LTA are true data.

Array No. 4 will determine if any of its inputs are active. If they are, the word associated with the active input is placed on the output lines.

The array is programmed into terms of true data. Five hexadecimal digits are used to program each word. Transistors are placed in the array for 1 bits in the data specification. More than one active AND gate in Array 3 causes the outputs to OR.



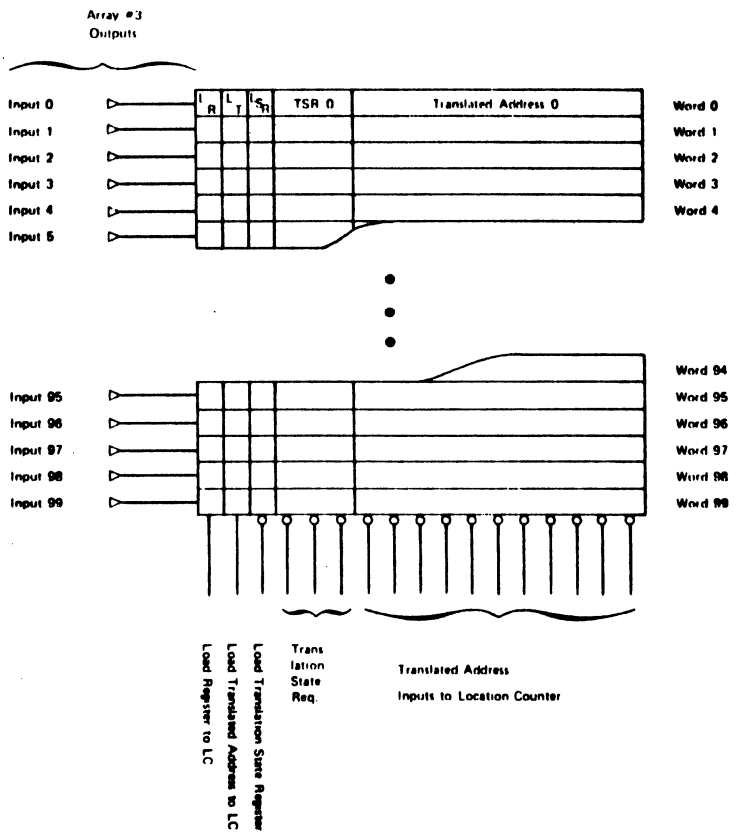


Figure 4-36. Array No. 4 Organization

Translation State Register. The Translation State Register serves two functions. One is to determine whether the Hi or the Lo order byte of the Translation Register is to be selected for input to Array No. 3. The other is to provide a four state feedback as an input of Array No. 3. The loading of the Translation State Register is controlled by an output (LTSR) of Array No. 4.

Translation Register. The Translation Register holds the data presently serving as input to the Programmable Translation Array. It receives its inputs from the Microinstruction Bus (MIB00 - MIB15) as a result of an LTR or IW instruction. Because of the fact that its output destination is 8 bits wide, only half of the Translation Register can be examined at any one time. The

Translation State Register controls which half of the Translation Register is gated into the input of Array No. 3, its destination. It serves as a feedback term of information to the second half of the word.

Bit 2 of the TSR controls the selection of translation input from the TR. When this bit is a zero, the upper byte of the TR is selected; when it is a one, the lower byte is selected. TSR2 does not serve as an input to Array No. 3.

4.3.3.2 PROGRAMMABLE TRANSLATION ARRAY OPERATIONS.
This section describes the operation of the Programmable Translation Array. The Programmable Translation Array operates as a loop. The beginning and end points of the loop can be considered as the Location Counter.

This loop is completed within one machine cycle and the main components are:

The Location Counter
Arrays No. 1 and No. 2, which taken together perform a mapping of the Location Counter to a translation code.
The translation register, which holds the macroinstruction undergoing the translation process.
Arrays No. 3 and No. 4 which take the translation code and the macroinstruction presently undergoing translation and map them into an address.

This newly generated address can then point to the beginning of a macroroutine to interpret the macroinstruction presently accessed.

The operation of the Programmable Translation Array is a function of phase time clocks and begins at Ø1. During Ø1, the location counter is loaded with its new value. The new value can come from any one of four sources as follows:

From Array No. 4 of the Programmable Translation Array
From the MI register
From the incrementer
From the Return Register

Figure 4-37 assumes, for the purposes of instruction, that the Location Counter is loaded from the incrementer. Concurrently, with the loading of the Location Counter at Ø1, the Translation Register is also being loaded if either a LTR or an IW instruction was executed.

At the beginning of Ø2, (see Figure 4-38) the contents of the Location Counter are clocked out to the MIB bus (MIB00-10) and to Array No. 1. Both the true and complemented forms of the Location Counter are input to Array No. 1. Also input is the RNI line. Array No. 1 takes the input data word and determines whether or not it has a match. If it has a match, then the appropriate output is set at the end of Ø2. If there is no match, the translation proceeds no farther because no outputs are set.

At the beginning of Ø3, (see Figure 4-39), Array No. 2 samples the outputs from Array No. 1 and determines whether any are active. If there is a match, then the translation code is generated and fed to Array No. 3, also during Ø3.

During Ø3, the contents of the selected byte of the Translation Register are fed to Array No. 3 along with the contents of the interrupt register, the Translation State Register, and the Q bit. During the later portions of Ø3, Array No. 3 determines if there is a match and, if a match is found, at the end of Ø3 an output is generated.

This output is fed into Array No. 4 at the beginning of Ø4. (See Figure 4-40.) During Ø4, Array No. 4 determines whether there is an active input to it. If there is an active input,

the contents of the word associated with the input are sent out at the end of Ø4. The Array No. 4 word may or may not cause the Translation State Register to be loaded, and may or may not cause the Location Counter to be loaded at the next occurrence of Ø1.

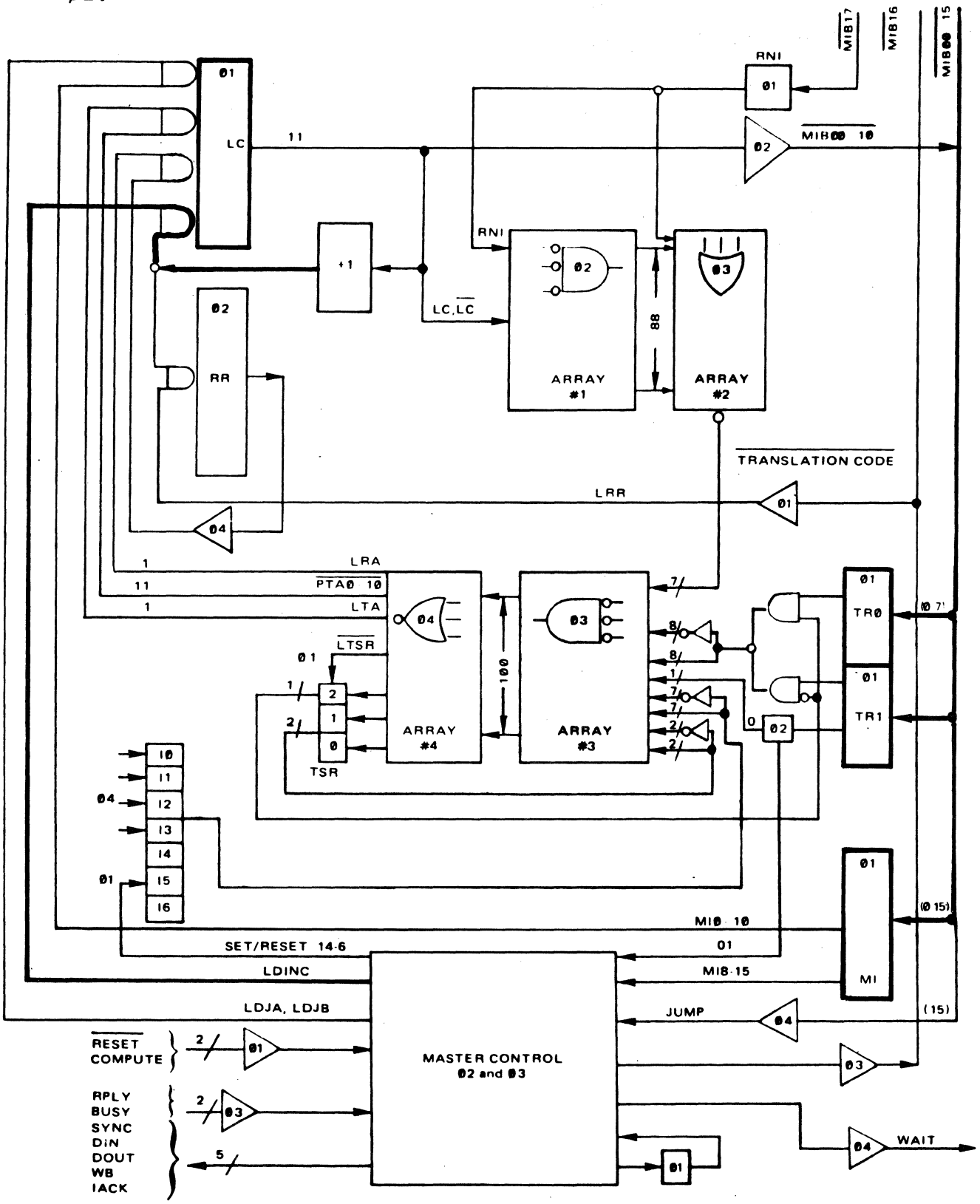


Figure 4-37. Ø1 Data Flow

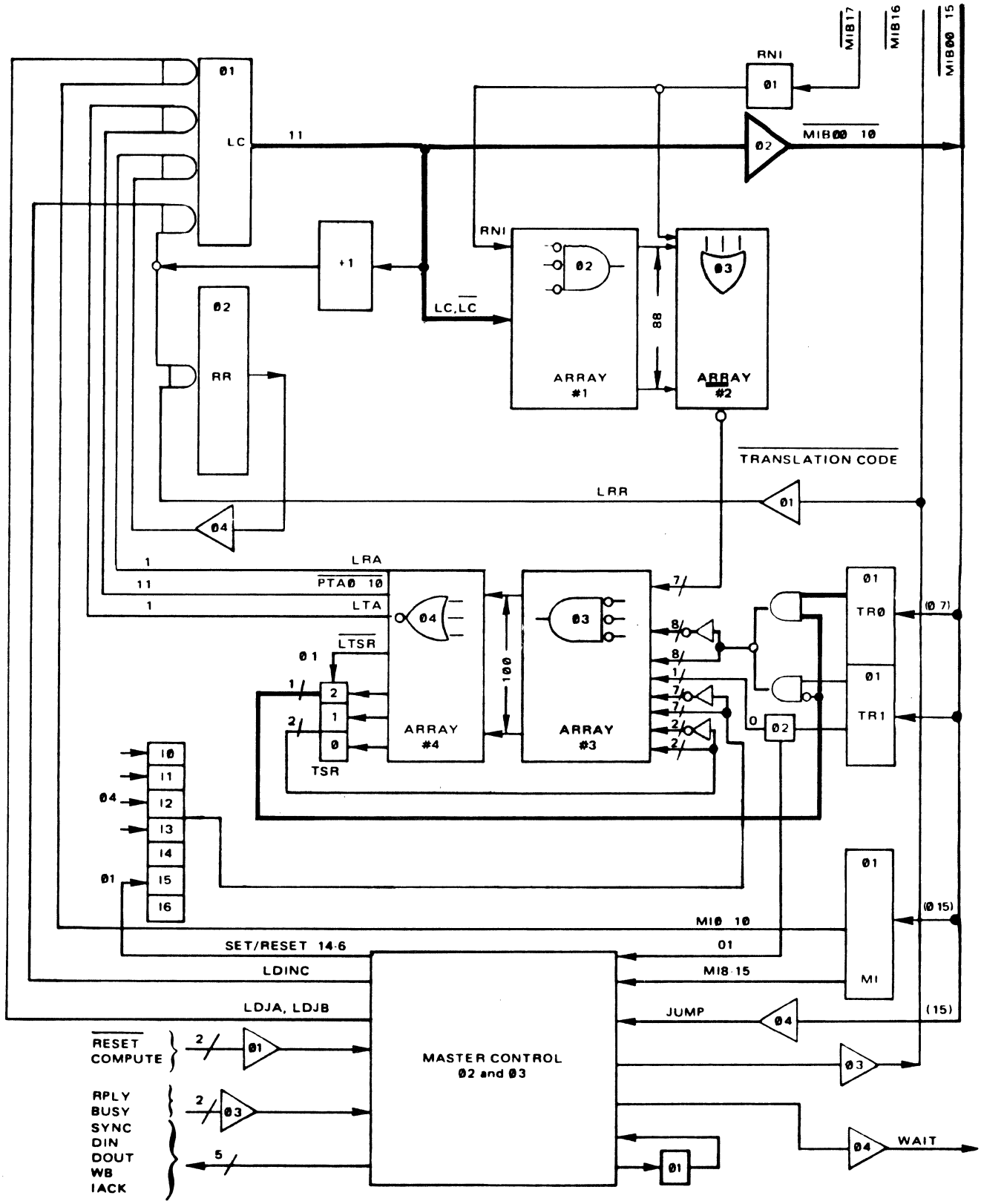


Figure 4-38. Ø2 Data Flow

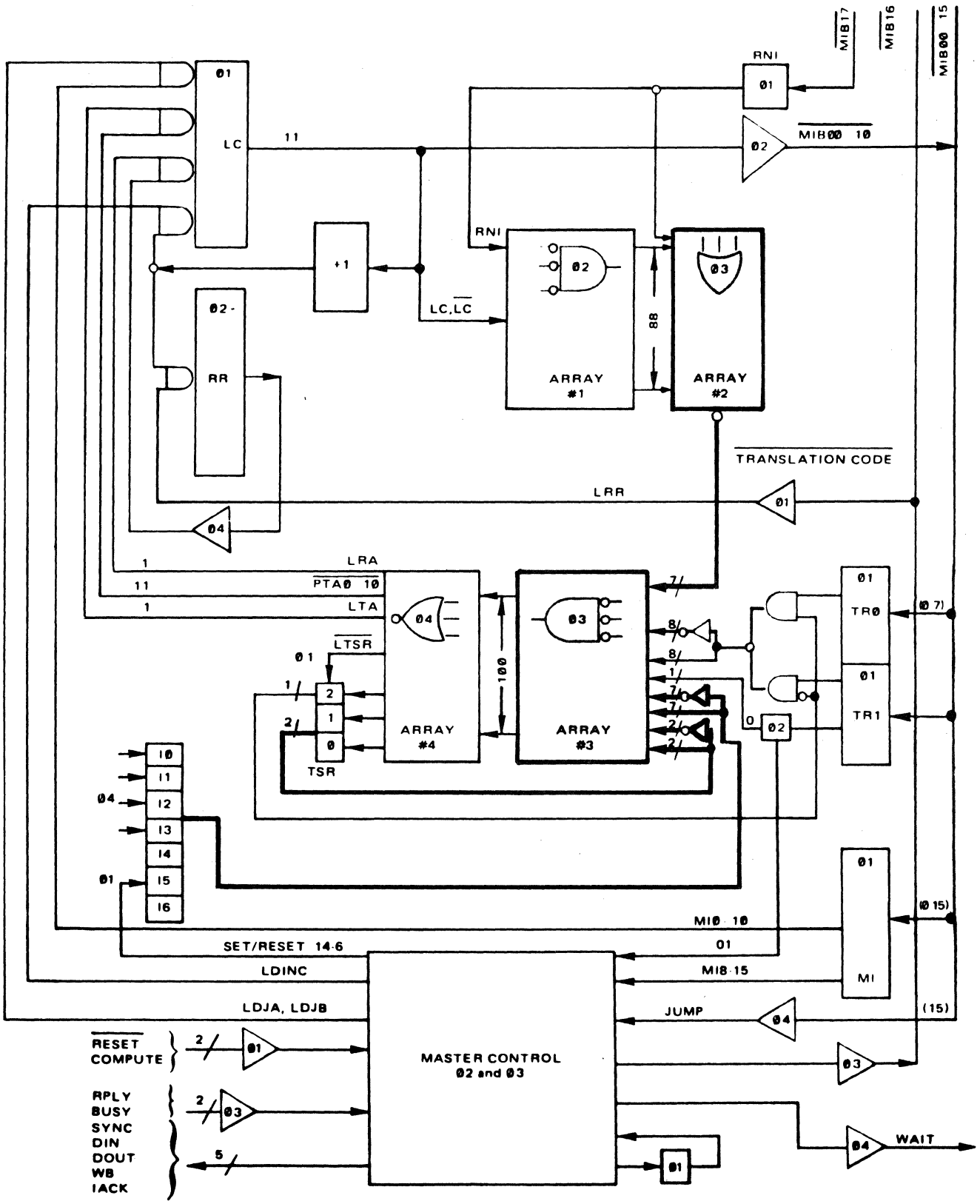


Figure 4-39. Ø3 Data Flow

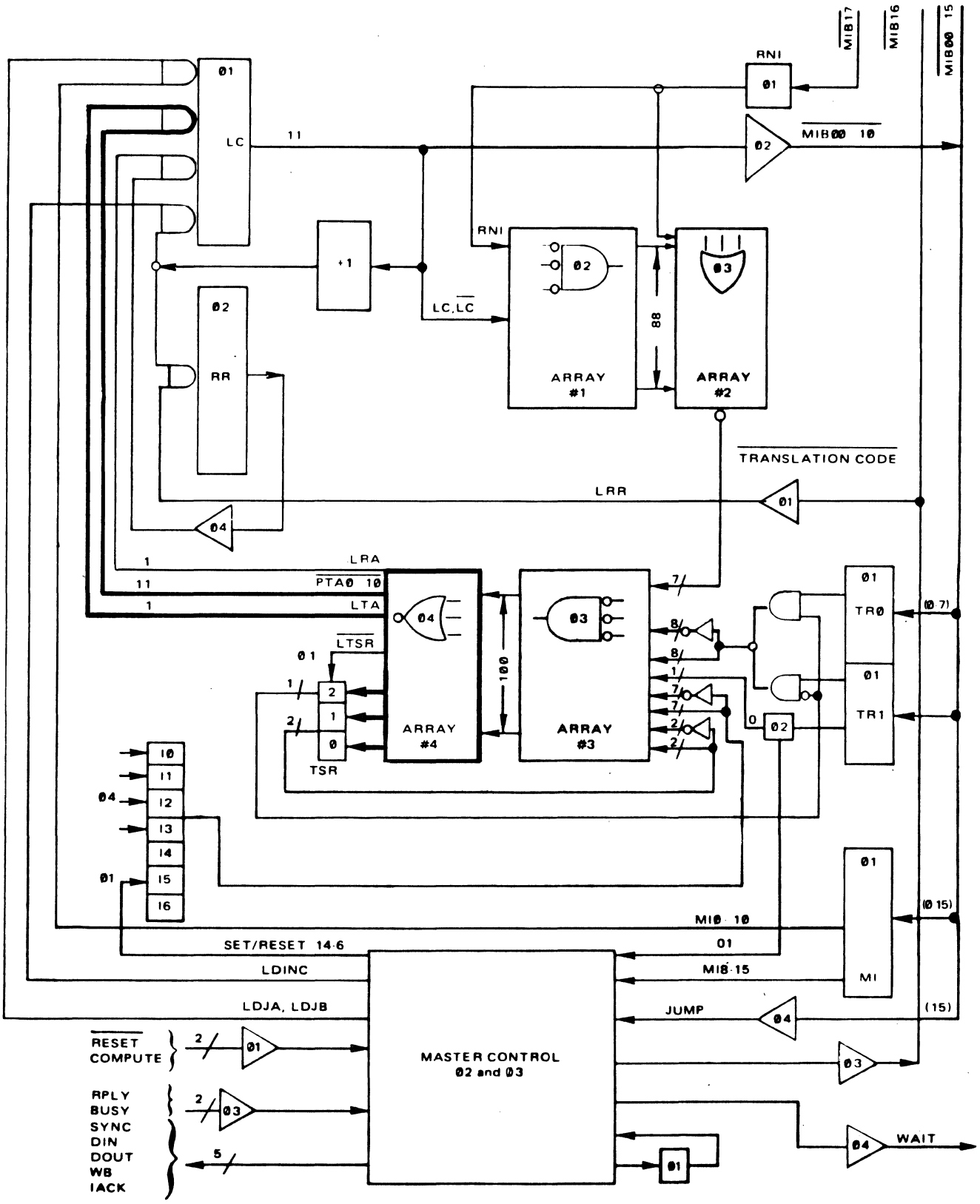


Figure 4-40. Ø4 Data Flow

4.3.4 MICROINSTRUCTIONS.

This section contains descriptions of the microinstructions. With each description is a diagram showing the format of the instruction and its operation code given in hexadecimal. Above each diagram is the instruction mnemonic operation code, the argument needed by the assembler, the instruction, and the name of the instruction. Under each diagram is a description of the command and its timing in clock cycles.

While the formats presented here describe 16 bits of instruction word, be aware that the word is, in fact, 22 bits wide. The portion of the instruction described here controls the operation of the processor. The other 6 bits are:

- bits 16 and 17 are involved with control of the Location Counter. Bit 16 determines whether or not the contents of the Return Register will be loaded into the Location Counter. Bit 17 determines whether or not a Read Next Instruction translation will be invoked.

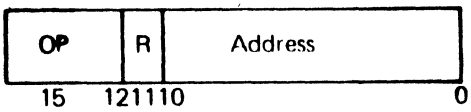
  These two options are available on all instructions.

- bits 18 through 21 have nothing to do with control of the MCP1600 system. They are available for the user to program as TTL levels. These bits are available on all instructions.

Instruction Types. All of the MCP1600 microinstructions are 22 bits long. Of the 22 bits, the 16 least significant effect the operation of the CPU. The purposes of the other 6 bits are covered in either the Programmed Translation Array description or the Micro Instruction Bus description.

The instructions for the MCP1600 are divided into four classes as follows:

JUMP FORMAT

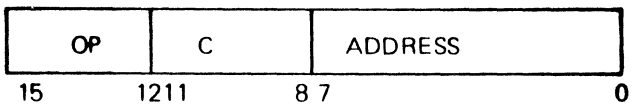


This format provides an 11-bit Control ROM address. It causes an unconditional jump to any location in the permissable MICROM address space by always asserting the jump control line (JXX on the Data Chip or MIB 15 on the Micro Instruction Bus). The unconditional jump instruction is the only one that utilizes this format.

Unconditional jump instructions always take two clock cycles; one to decode the instruction, the second to load the Location Counter on the Control Chip.

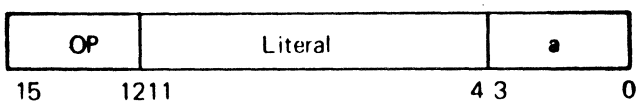
A special case of this format is the Return From Subroutine instruction, covered later.

CONDITIONAL JUMP FORMAT



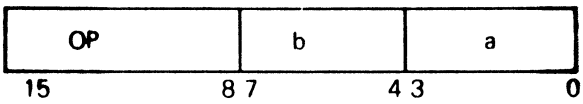
This format provides a Jump address within a page. This instruction is substantially the same as the Unconditional Jump format with the exception of the reduced address space. It is also a two cycle instruction with the jump decision (based upon the contents of the C Field) being made by the jump decode logic on the Data Chip during the first cycle. This decision is communicated to the Control Chip via MIB15 (JXX on the Data Chip). During the second cycle the jump is affected.

LITERAL FORMAT



This format provides 8-bit literal data. The "a" field usually specifies a file register. These instructions invoke the "literal path" to fetch data around the register file for the "b" input to the ALU. All of these instructions execute in one clock cycle.

REGISTER FORMAT



This format is by far the predominant one. In it, "b" and "a" are register file designators. Generally, depending upon the operation, there can be two meanings to the register designators:

"b" $\xrightarrow{\text{(operation)}}$ "a"

as, for instance, in a shift or move instruction

"b" (operation) "a" $\xrightarrow{\qquad\qquad}$ "a"

as, for instance, in an arithmetic instruction.

"b" and "a" can specify either a single byte data quantum or a word (double byte) data quantum. In this latter case, the designators usually point to the even addressed member of a register pair. While they may both point to an odd addressed pair, note carefully the instructions description for the effect.

If the designators apply to a word data quantum, the instruction takes two cycles to execute. The following sections describe MCP1600 instructions by classes of function. See Table 4-14 for the microinstruction set, Table 4-15 for a list of the micro-instructions, and Table 4-16 for a summary of microinstruction and status flags.

Table 4-14. Microinstruction Set

Symbols	Meaning
→	is transferred to
( )	contents of location or register
Λ	Logic Product (AND)
V	Inclusive "or"
⊻	Exclusive "or"
Rx:Ry	Forms extended register Ry = LSB, Rx = MSB
Ra	The register specified by the micro-instruction "a" field
Rb	Designates no catagory
Flag Setting	The register specified by the micro-instruction "b" field
X	Don't care condition
Ø	Flag cleared (set to Ø)
1	Flag set
–	Flag not affected
*	Set according to function

Table 4-15. Microinstructions

Instruction	OP Code	Mnemonic	Microcycles	Description of Operation
Jump	Ø(Ø)	JMP	2	(MIR11-ØØ)→(LC) Jump unconditionally to microm location specified by MIR11-ØØ
Return from Subroutine	Ø(1)	RFS	2	(RR)→(LC) Restore return address from subroutines
Jump if ZB False	1Ø	JZBF	2	If ZB=Ø, (MIR ØØØ)→(LC7-Ø) jump conditionally within microm page
Jump if ZB True	11	JZBT	2	If ZB=1, (MIRØ7-ØØ)→(LC7-Ø)
Jump if C8 False	12	JC8F	2	If C8=0, (MIRØ7-ØØ)→(LC7-Ø)
Jump if C8 True	13	JC8T	2	If C8=1, (MIRØ7-ØØ)→(LC7-Ø)
Jump if ICS False	14	JIF	2	If ICS=Ø, (MIRØ7-ØØ)→(LC7-Ø)
Jump if ICS True	15	JIT	2	If ICS=1, (MIRØ7-ØØ)→(LC7-Ø)
Jump if NB False	16	JNBF	2	If NB=0, (MIRØ7-ØØ)→(LC7-Ø)
Jump if NB True	17	JNBT	2	If NB-1, (MIRØ7-ØØ)→(LC7-Ø)
Jump if Z False	18	JZF	2	If Z=0, (MIRØ7-ØØ)→(LC7-Ø)
Jump if Z True	19	JZT	2	If Z=1, (MIRØ7-ØØ)→(LC7-Ø)
Jump if C False	1A	JCF	2	If C=0, (MIRØ7-ØØ)→(LC7-Ø)
Jump if C True	1B	JCT	2	If C=1, (MIRØ7-ØØ)→(LC7-Ø)
Jump if V False	1C	JVF	2	If V=0, (MIRØ7-ØØ)→(LC7-Ø)
Jump if V True	1D	JVT	2	If V=1, (MIRØ7-ØØ)→(LC7-Ø)
Jump if N False	1E	JNF	2	If N=0, (MIRØ7-ØØ)→(LC7-Ø)
Jump if N True	1F	JNT	2	If N=1, (MIRØ7-ØØ)→(LC7-Ø)
Add Literal	2	AL	1	(Ra)+Literal→Ra the 8 bit constant is added to Ra
Compare Literal	3	CL	1	(Ra)-Literal. The 8 bit result of the literal is compared against Ra and the approprate flags set.
And Literal	4	NL	1	(Ra)∧Literal→(Ra) The 8 bit result of a logical product of Ra and the literal are coded into Ra.
Test Literal	5	TL	1	(Ra)∧Literal The 8 bit result of the logical product of Ra and the literal set condition Flags. The contents of Ra are unaffected.
Load Literal	6	LL	1	Literal→(Ra). The 8 bit literal is loaded into Ra.
Reset Interrupts	7Ø	RI	1	The three software interrupts 14, 15, and 16 are reset as indicated in the instruction b field.
Set Interrupts	71	SI	1	The three software interrupts 14, 15, and are set as indicated in the instructions b field.
Copy Condition Flags	72	CCF	1	Flags→(Ra) The four condition flags are four ALU status flags are loaded in Ra
Load Condition Flags	73	LCF	1	(Ra)→Flags. The contents of Ra are transferred to the ALU status flags unconditionally and to the condition flags under control of the b field.
Reset TSR	74	RTSR	1	0→(TSR). The three TSR bits in the control chip are reset.
Load G Low	75	LGL	1	(Ra) 2-0→(G). The three least significant bits of Ra are loaded into the G register.
Conditionally Increment Byte	76	CIB	1	If C8 is set, (Ra)=I→(Ra)
Conditionally Decrement Byte	77	CDB	I	If C8 is set, (Ra)-I→(Ra)

Table 4-15 (Cont.). Microinstructions

Instruction	OP Code	Mnemonic	Microcycles	Description of Operation
Move Byte	80/81	MB	1	(Rb)→(Ra). The 8 bit contents of Rb are transferred to Ra. Rb is unaffected.
Move Word	82/83	MW	2	(Rb)→(Ra). The 16 bit contents of Rb+1:Rb are transferred to Ra+1:Ra. Rb+1 and Rb are unaffected.
Conditionally Move Byte	84/85	CMB	1	(Rb)→(Ra), if C=1. The 8 bit contents of Rb are transferred to Ra if the carry flags is set from a previous operation.
Conditionally Move Word	86/87	CMW	2	(Rb)→(Ra), if C=1. The 16 bit contents of Rb=1:Rb are transferred into Ra+1:Ra if the carry flag is set from a previous operation.
Shift Left Byte with Carry	88/89	SLBC	1	(Rbm)→(Ram+1), (carry)→(Ra0). The 8 bit contents of Rb are shifted left one bit and loaded into Ra. The contents of the carry flag is inserted into the vacated low position of Ra.
Shift Left Word with Carry	8A/8B	SLWC	2	(Rbm)→(Ram+1), (carry)→(Ra0). The 16 bit contents of Rb+1:Rb are shifted left one bit and loaded into Ra+1:Ra. The contents of the carry flag is inserted into the vacated low position of Ra.
Shift Left Byte	8C/8D	SLB	1	(Rbm)→(Ram+1) The 8 bit contents of Rb are shifted left one bit and loaded into Ra.
Increment Byte by 1	90/91	ICBI	1	(Rb)+1→(Ra). The 8 bit contents of Rb are incremented by one and the result is transferred to Ra.
Increment Word by 1	92/93	ICWI	2	(Rb)+1→(Ra). The 16 bit contents of Rb+1:Rb are incremented by one and the result transferred to Ra+1:Ra.
Increment Byte by 2	94/95	ICB2	1	(Rb)+2→(Ra). The 8 bit contents of Rb are incremented by two and the result transferred to Ra.
Increment Word by 2	96/97	ICW2	2	(Rb)+2→(Ra). The 16 bit contents of Rb+1:Rb are incremented by two and the result transferred to Ra+1:Ra.
Twos Complement Byte	98/99	TCB	1	$(\overline{Rb})$+1→Ra. The 8 bit contents of Rb are two's complemented and transferred to Ra.
Twos Complement Word	9A/9B	TCW	2	$(\overline{Rb})$+1→Ra. The 16 bit contents of Rb+1 Rb are two's complemented and transferred to Ra+1:Ra.
Ones Complement Byte	9C/9D	OCB	1	$(\overline{Rb})$→Ra. The 8 bit contents of Rb are one's complemented and transferred to Ra.
One's Complement Word	9E/9F	OCW	2	$(\overline{Rb})$→Ra. The 16 bit contents of Rb+1:Rb are one's complemented and transferred to Ra+1:Ra.
Add Byte	A0/A1	AB	1	(Rb)+(Ra)→(Ra). The 8 bit contents of Rb are added to Ra and loaded into Ra.
Add Word	A2/A3	AW	2	(Rb)+(Ra)→(Ra). The 16 bit contents of Rb+1:Rb are added to Ra+1:Ra and loaded into Ra+1:Ra.
Conditionally Add Byte	A4/A5	CAB	1	(Rb)+(Ra)→(Ra), if C=1. The 8 bit contents of Rb are added to Ra and the result is transferred to Ra if the carry flag is set from a previous operation.
Conditionally Add Word	A6/A7	CAW	2	(Rb)+(Ra)→(Ra), if C=1. The 16 bit contents of Rb+1:Rb are added to Ra+1:Ra and the result is transferred to Ra+1;Ra if the carry flag is set from a previous operation.
Add Byte with Carry	A8/A9	ABC	1	(Rb)+(Ra)+C→(Ra). The 8 bit sum of the contents of Rb plus the carry flag are added to Ra and the result is transferred to Ra.

Table 4-15 (Cont.). Microinstructions

Instruction	OP Code	Mnemonic	Microcycles	Description of Operation
Add Word with Carry	AA/AB	AWC	2	(Rb)+(Ra)+C→(Ra). The 16 bit sum of the contents of Rb+1:Rb plus the carry flag are added to Ra+1:Ra and the result is transferred to Ra+1:Ra.
Conditionally Add Digits	AC	CAD	1	$(Rb)_{3-0}+(Ra)_{3-0}\rightarrow(Ra)_{3-0}$, if C4=0 $(Rb)_{7-4}+(Ra)_{7-0}\rightarrow(Ra)_{7-4}$, if C8=0 The 4 least significant and/or 4 most significant bits of Rb are added to their corresponding bits in Ra if C4 or C8 and reset respectively from a previous operation. This allows decimal arithmetic corrections.
Conditionally Add Word on Indirect Condition Status	AE/AF	CAWI	2	(Rb)+(Ra)→(Ra), if ICS=1. The 16 bit contents of Rb+1:Rb are added to Ra+1:Ra and the results transferred to Ra+1:Ra if the ICS code is set from a prior operation. This instruction allows address displacement addition on Branch conditions.
Subtract Byte	BØ/BI	SB	1	(Ra)-(Rb)→(Ra). The 8 bit contents of Rb are subtracted from Ra and the result loaded into Ra.
Subtract Word	B2/B3	SW	2	(Ra)-(Rb)→(Ra). The 16 bit contents of Rb+1:Rb are subtracted from Ra+1:Ra and the results loaded into Ra+1:Ra.
Compare Byte	B4/B5	CB	1	(Ra)-(Rb). The 8 bit difference between the contents of Ra and Rb is used to set the status flags at all times and the condition flags if OP code B5 is selected Registers Ra and Rb are not changed.
Compare Word	B6/B7	CW	2	(Ra)-(Rb). The 16 bit difference between the contents of Ra+1:Ra and Rb+1:Rb are used to set the status flags at all times and the condition flags if OP Code B7 is selected. Registers Ra+1:Ra and Rb+1:Rb are not changed.
Subtract Byte with Carry	B8/B9	SBC	1	(Ra)-(Rb)-C→(Ra). The difference of the 8 bit contents of Rb subtracted from Ra minus the contents of C flag is loaded into Ra.
Subtract Word with Carry	BA/BB	SWC	2	(Ra)-Rb)-C→(Ra). The difference of the 16 bit contents of Rb+1:Rb subtracted from Ra+1:Ra minus the contents of the C flag is loaded into Ra+1:Ra.
Decrement Byte by 1	BC/BD	DBI	1	(Rb)-1→Ra. The 8 bit contents of Rb minus one are loaded into Ra.
Decrement Word by 1	BE/BF	DWI	2	(Rb)-1→Ra. The 16 bit contents of Rb+1:Rb minus one are loaded into Ra+1:Ra.
And Byte	CØ/CI	NB	1	(Rb)∧(Ra)→(Ra). The 8 bit logical product of Rb and Ra is loaded into Ra.
And Word	C2/C3	NW	2	(Rb)∧(Ra)→(Ra). The 16 bit logical product of Rb+1:Rb and Ra+1:Ra is loaded into Ra+1:Ra.
Test Byte	C4/C5	TB	1	(Rb)∧(Ra). The 8 bit logical product of Rb and Ra sets the status flags at all times and the condition flags if OP code C5 is selected. Ra and Rb are unchanged.
Test Word	C6/C7	TW	2	(Rb)∧(Ra). The 16 bit logical product of Rb+1:Rb and Ra+1:Ra sets the status flags at all times and the condition flags if OP code C7 is selected. Ra and Rb are unchanged.
Or Byte	C8/C9	OB	1	(Rb)∨(Ra)→(Ra). The 8 bit logical OR operation is performed between the contents of Ra and Rb and the result are transferred to Ra.

Table 4-15 (Cont.). Microinstructions

Instruction	OP Code	Mnemonie	Microcycles	Description of Operation
Or Word	CA/CB	OW	2	(Rb)V(Ra)→(Ra). The 16 bit logical OR operation is performed between the contents of Ra+1:Ra and Rb+1:Rb and the results are transferred to Ra+1:Ra.
Exclusive-Or Byte	CC/CD	XB	1	(Rb)⊻(Ra)→Ra. The 8 bit logical exclusive OR operation is performed between the contents of Rb and Ra and the result is transferred to Ra.
Exclusive-Or Word	CE/CF	XW	2	(Rb)⊻(Ra)→Ra. The 16 bit logical exclusive OR operation is performed between the contents of Rb+1:Rb and Ra+1:Ra and the result is transferred to Ra+1:Ra.
And Complement Byte	DØ/DI	NCB	1	$(\overline{Rb})\wedge(Ra)\rightarrow(Ra)$. The 8 bit logical product of the inverse of Rb and the contents of Ra is loaded into Ra.
And Complement Word	D2/D3	NCW	2	$(\overline{Rb})\wedge(Ra)\rightarrow(Ra)$. The 16 bit logical product of the inverse of Rb+1:Rb and the contents of Ra+1:Ra is loaded into Ra+1:Ra.
Shift Right Byte with Carry	D8/D9	SRBC	1	(Rbm+1)→(Ram), (carry)→(Ra_7). The 8 bit contents of Rb are shifted right one bit and loaded into Ra. The carry flag is inserted into the high order position of Ra.
Shift Right Word with Carry	DA/DB	SRWC	2	(Rbm+1)→(Ram), (carry)→(Ra_{15}). The 16 bit contents of Rb+1:Rb are shifted right one bit and loaded into Ra+1:Ra. The carry flag is inserted into the high order position of Ra+1.
Shift Right Byte	DC/DD	SRB	1	(Rbm+1)→(Ram). The 8 bit contents of Rb are shifted right one bit and loaded into Ra.
Shift Right Word	DE/DF	SRW	2	(Rbm+1)→(Ram). The 16 bit contents of Rb+1:Rb are shifted right one bit and loaded into Ra+1:Ra.
Input Byte	EØ/EI	IB	1 (min)	(DAL)→(Ra). An 8 bit byte on the DAL is loaded into the specified Ra. The b field in this instruction selects read or read-modify-Write operation and selects upper (Bit 15-8) or lower (Bits 7-0) as the byte to be input from the Dal lines.
Input Word	E2/E3	IW	2 (min)	(DAL)→(Ra). The 16 bit word contained on the DAL is loaded into Ra+1:Ra. The b field in this instruction selects read or read-modify-write operation, selectively updates the G register, and selectively sets the ICS bit.
Input Status Byte	E4/E5	ISB	1	(DAL)→(Ra). The 8 bit byte from the DAL line, as specified by the b fold is input to register Ra, regardless of the state of reply or busy signal.
Input Status Word	E6/E7	ISW	2	(DAL)→(Ra). The 16 bit word from the DAL line is loaded into Ra+1:Ra regardless of the state of the Reply or Busy signal.
Modify Instruction	EC/ED	MI	1	(MIB)V(Rb:Ra). The 16 bit contents of registers Rb:Ra are ORed with the contents of next microm micro-instruction on the MIB lines to modify any or all of the next micro-instruction. This instruction can be used to make on-line changes to micro-program flow.

Table 4-15 (Cont.). Microinstructions

Instruction	OP Code	Mnemonic	Microcycles	Description of Operation
Load Translation Register	EE/EF	LTR	2	(Rb:Ra)→(TR). The 16 bit contents of registers Rb:Ra are transferred to the Translation Register on the chip. This allows a translation of resultant data into a micro-instruction.
Read and Increment Byte by 1	FØ	RIBI	1	(Rb:Ra)→DAL (Ra)+1→(Ra). The 16 bit address located in Rb:Ra is transferred to the DAL lines and a DATA READ operation is initiated. The contents of Ra are incremented by one. Rb is unchanged.
Write and Increment Byte by 1	F1	WIBI	1	(Rb:Ra)→DAL, (Rb)+1→(Ra). The 16 bit address located in Rb;Ra is transferred to the DAL lines and a DATA WRITE operation is initiated. The contents of Ra are incremented by one. Rb is unchanged.
Read and Increment Word by 1	F2	RIWI	2	(Rb:Ra)→DAL, (Ra+1:Ra)+1→(Ra+1:Ra). The 16 bit address located in Rb;Ra is transferred to the DAL lines and a DATA READ operation is initiated. The contents of Ra+1:Ra are incremented by one.
Write and Increment Word by 1	F3	WIWI	2	(Rb:Ra)→DAL, (Ra+1:Ra)+1→(Ra+1:Ra). The 16 bit address located in Rb:Ra is transferred to the DAL lines and a DATA WRITE operation is initiated. The contents of Ra+1:Ra are incremented by one.
Read and Increment Byte by 2	F4	RIB2	1	(Rb:Ra)→DAL, (Ra)+2→(Ra). The 16 bit address located in Rb:Ra is transferred to the DAL lines and a DATA READ operation is initiated. The contents of Ra are incremented by 2. Rb remains unchanged.
Write and Increment Byte by 2	F5	WIB2	1	(Rb:Ra)→DAL, (Ra)+2→(Ra). The 16 bit address located in Rb:Ra is transferred to the DAL line and a DATA WRITE operation is initiated. The contents of Ra are incremented by 2. Rb is not changed.
Read and Increment Word by 2	F6	RIW2	2	(Rb:Ra)→DAL, (Ra+1:Ra)+2→(Ra+1:Ra). The 16 bit address located in Rb:Ra is transferred to the DAL lines and a DATA READ operation is initiated. The content of Ra+1:Ra are incremented by 2.
Write and Increment Word by 2	F7	WIW2	2	(Rb:Ra)→DAL, (Ra+1:Ra)+2→(Ra+1:Ra). The 16 bit address located in Rb:Ra is transferred to the DAL lines and a DATA WRITE operation is initiated. The contents of Ra+1:Ra are incremented by 2.
Read	F8	R	1	(Rb:Ra)→DAL. The 16 bit address located in Rb:Ra is transferred to the DAL line and a DATA READ operation is initiated.
Write	F9	W	1	(Rb:Ra)→DAL. The 16 bit address located in Rb:Ra is transferred to the DAL lines and a DATA WRITE operation is initiated.
Read Acknowledge	FA	RA	1	(Rb:Ra)→(DAL, 1→IACK. The 16 bit address located in Rb:Ra is transferred to the DAL lines and a DATA READ operation is initiated. The interrupt acknowledge line is set high.
Write Acknowledge	FB	WA	1	(Rb:Ra)→DAL, 1→IACK. The 16 bit address located in Rb:Ra is transferred to the DAL lines and a DATA WRITE operation is initiated. The Interrupt Acknowledge line is set high.

Table 4-15 (Cont.). Microinstructions

Instruction	OP Code	Mnemonic	Microcycles	Description of Operation
Output Byte	FC	OB	1 (min.)	(Rb:Ra)→DAL, 1→DOUT. The 16 bit data contents of Rb Ra are transferred to the DAL lines. Registers Ra and Rb are unchanged. The DATA OUT signal is activated. This instruction is completed when the REPLY signal is received from the addressed unit. For byte operations Rb must equal Ra so that the same byte is placed in both 8 bit positions on the DAL.
Output Word	FD	OW	1 (min.)	(Rb:Ra)→DAL, 1→DOUT. The 16 bit data contents of Rb:Ra are transferred to the DAL lines. The DATA OUT signal is activated. Registers Ra and Rb are unchanged. This instruction is complete when the REPLY signal is received from the addressed unit.
Output Status	FE	OS	1	(Rb:Ra)→DAL. The 16 bit contents of registers Rb:Ra is transferred to the DAL lines and takes place regardless of the state of the Reply or Busy signal. DOUT is not activated and Sync is not terminated. Registers Rb and Ra are unchanged.
No Operation	FF	NOP	1	This instruction causes no operation.

Table 4-16. Summary of Microinstruction and Status Flags

Mnemonic	OP Code	Microcycles	ALU Status Flags NB	ZB	C4	C8	Condition Flags N	Z	V	C
A. Jump Format	Bit 15-12 11 10-0									
JMP	0 0 ADDR	2	–	–	–	–	–	–	–	–
RFS	0 1 ---------	2	–	–	–	–	–	–	–	–
B. Conditional Jump Format										
	Bit 15-12 11-8 7-0									
JZBF	1 0 ADDR	2	–	–	–	–	–	–	–	–
JZBT	1 1 ADDR	2	–	–	–	–	–	–	–	–
JC8F	1 2 ADDR	2	–	–	–	–	–	–	–	–
JC8T	1 3 ADDR	2	–	–	–	–	–	–	–	–
JIF	1 4 ADDR	2	–	–	–	–	–	–	–	–
JIT	1 5 ADDR	2	–	–	–	–	–	–	–	–
JNBF	1 6 ADDR	2	–	–	–	–	–	–	–	–
JNBT	1 7 ADDR	2	–	–	–	–	–	–	–	–
JZF	1 8 ADDR	2	–	–	–	–	–	–	–	–
JZT	1 9 ADDR	2	–	–	–	–	–	–	–	–
JCF	1 A ADDR	2	–	–	–	–	–	–	–	–
JCT	1 B ADDR	2	–	–	–	–	–	–	–	–
JVF	1 C ADDR	2	–	–	–	–	–	–	–	–
JVT	1 D ADDR	2	–	–	–	–	–	–	–	–
JNF	1 E ADDR	2	–	–	–	–	–	–	–	–
JNT	1 F ADDR	2	–	–	–	–	–	–	–	–
C. Literal Format										
	Bit 15-12 11-4 3-0									
AL	2 Lit a	1	*	*	*	*	–	–	–	–
CL	3 Lit a	1	*	*	–	–	–	–	–	–
NL	4 Lit a	1	*	*	–	–	–	–	–	–
TL	5 Lit a	1	*	*	–	–	–	–	–	–
LL	6 Lit a	1	*	*	–	–	–	–	–	–

Table 4-16. (Cont.). Summary of Microinstruction and Status Flags

Mnemonic	OP Code			Microcycles	ALU Status Flags				Conditon Flags			
					NB	ZB	C4	C8	N	Z	V	C
). Register Format	Bit 15-8	7-4	3-0									
RI	70	b	x	1	-	-	-	-	-	-	-	-
SI	71	b	x	1	-	-	-	-	-	-	-	-
CCF	72	x	a	1	-	-	-	-	-	-	-	-
LCF	73	b	a	1	-	-	-	-	-	-	-	-
RTSR	74	x	x	1	-	-	-	-	-	-	-	-
LGL	75	x	a	1	-	-	-	-	-	-	-	-
CIB	76	x	a	1	*	*	*	*	-	-	-	-
CDB	77	x	a	1	*	*	*	*	-	-	-	-
MB	80/81	b	a	1	*	*	-	-	*	*	0	-
MW	82/83	b	a	2	*	*	-	-	*	*	0	-
CMB	84/85	b	a	1	(*	*	-	-	*	*	0	-).C
CMW	86/87	b	a	2	(*	*	-	-	*	*	0	-).C
SLBC	88/89	b	a	1	*	*	*	*	*	*	*	*
SLWC	8A/8B	b	a	2	*	*	*	*	*	*	*	*
SLB	8C/8D	b	a	1	*	*	*	*	*	*	*	*
SLW	8E/8F	b	a	2	*	*	*	*	*	*	*	*
ICB	90/91	b	a	1	*	*	*	*	*	*	*	*
ICWI	92/93	b	a	2	*	*	*	*	*	*	*	*
ICBZ	94/95	b	a	1	*	*	*	*	*	*	*	*
ICWZ	96/97	b	a	2	*	*	*	*	*	*	*	*
TCB	98/99	b	a	1	*	*	*	*	*	*	*	*
TCW	9A/9B	b	a	2	*	*	*	*	*	*	*	*
OCB	9C/9D	b	a	1	*	*	0	0	*	*	0	1
OCW	9E/9F	b	a	2	*	*	0	0	*	*	0	1
AB	A0/A1	b	a	1	*	*	*	*	*	*	*	*
AW	A2/A3	b	a	2	*	*	*	*	*	*	*	*
CAB	A4/A5	b	a	1	(*	*	*	*	*	*	*	*).C
CAW	A6/A7	b	a	2	(*	*	*	*	*	*	*	*).C
ABC	A8/A9	b	a	1	*	*	*	*	*	*	*	*
AWC	AA/AB	b	a	2	*	*	*	*	*	*	*	*
CAD	AC	b	a	1	*	*	*	*	*	*	*	*
CAWI	AE/AF	b	a	2	(*	*	*	*	*	*	*	*).C
SB	B0/B1	b	a	1	*	*	*	*	*	*	*	*
SW	B2/B3	b	a	2	*	*	*	*	*	*	*	*
CB	B4/B5	b	a	1	*	*	*	*	*	*	*	*
SW	B6/B7	b	a	2	*	*	*	*	*	*	*	*
SBC	B8/B9	b	a	1	*	*	*	*	*	*	*	*
SWC	BA/BB	b	a	2	*	*	*	*	*	*	*	*
DBI	BC/BD	b	a	1	*	*	*	*	*	*	*	*
DWI	BE/BF	b	a	2	*	*	*	*	*	*	*	*
NB	C0/C1	b	a	1	*	*	-	-	*	*	0	-
NW	C2/C3	b	a	2	*	*	-	-	*	*	0	-
TB	C4/C5	b	a	1	*	*	-	-	*	*	0	-
TW	C6/C7	b	a	2	*	*	-	-	*	*	0	-
ORB	C8/C9	b	a	1	*	*	-	-	*	*	0	-
ORW	CA/CB	b	a	2	*	*	-	-	*	*	0	-
XB	CC/CD	b	a	1	*	*	-	-	*	*	0	-
XW	CE/CF	b	a	2	*	*	-	-	*	*	0	-

Table 4-16. (Cont.). Summary of Microinstruction and Status Flags

Mnemonic	OP Code			Microcycles	ALU Status Flags				Condition Flags			
					NB	ZB	C4	C8	N	Z	V	C
D. Register Format	**Bit 15-8**	**7-4**	**3-0**									
NCB	D0/D1	b	a	1	*	*	–	-	*	*	0	–
NCW	D2/D3	b	a	2	*	*	–	–	*	*	0	–
SRBC	D8/D9	b	a	1	*	*	0	*	*	*	0	*
SRWC	DA/DB	b	a	2	*	*	0	*	**	*	0	*
SRB	DC/DD	b	a	1	*	*	0	*	*	*	0	*
SRW	DE/DF	b	a	2	*	*	0	*	*	*	0	*
IB	E0/E1	b	a	1	*	*	–	–	*	*	0	–
IW	E2/E3	b	a	2	*	*	–	–	*	*	0	–
ISB	E4/E5	b	a	1	*	*	–	–	*	*	0	–
ISW	E6/E7	x	a	2	*	*	–	–	*	*	0	–
MI	EC/ED	b	a	1	–	–	–	–	–	–	–	–
LTR	EE/EF	b	a	2	–	–	–	–	–	–	–	–
RIBI	F0	b	a	1	*	*	*	*	–	–	–	–
WIBI	F1	b	a	1	*	*	*	*	–	–	–	–
RIWI	F2	b	a	2	*	*	*	*	–	–	–	–
WIWI	F3	b	a	2	*	*	*	*	–	–	–	–
RIBZ	F4	b	a	1	*	*	*	*	–	–	–	–
WIBZ	F5	b	a	1	*	*	*	*	–	–	–	–
RIWZ	F6	b	a	2	*	*	*	*	–	–	–	–
WIWZ	F7	b	aa	2	*	*	*	*	–	–	–	–
R	F8	b	a	1	–	–	–	–	–	–	–	–
W	F9	b	a	1	–	–	–	–	–	–	–	–
RA	FA	b	a	1	–	–	–	–	–	–	–	–
WA	FB	b	a	1	–	–	–	–	–	–	–	–
OB	FC	b	a	1	–	–	–	–	–	–	–	–
OW	FD	b	a	1	–	–	–	–	–	–	–	–
OS	FE	b	a	1	–	–	–	–	–	–	–	–
NOP	FF	x	x	1	–	–	–	–	–	–	–	–

4.3.5 MICROINSTRUCTION BUS OPERATION.

The Microinstruction Bus interconnects the three different circuits that make up a MCP1600 Microprocessor system. Connected to the Microinstruction Bus can be one CP1611B Data Chip, one CP1621B Control Chip, and as many as four CP1631B Microinstruction ROM Chips. There may also be user supplied TTL logic connected to the Microinstruction Bus in a fashion to be defined below.

The discussion below describes the 22 different lines on the Microinstruction Bus from the standpoint of each of the various types of devices attached to it and from the standpoint of user attached devices. Figure 4-41 illustrates the system interconnections.

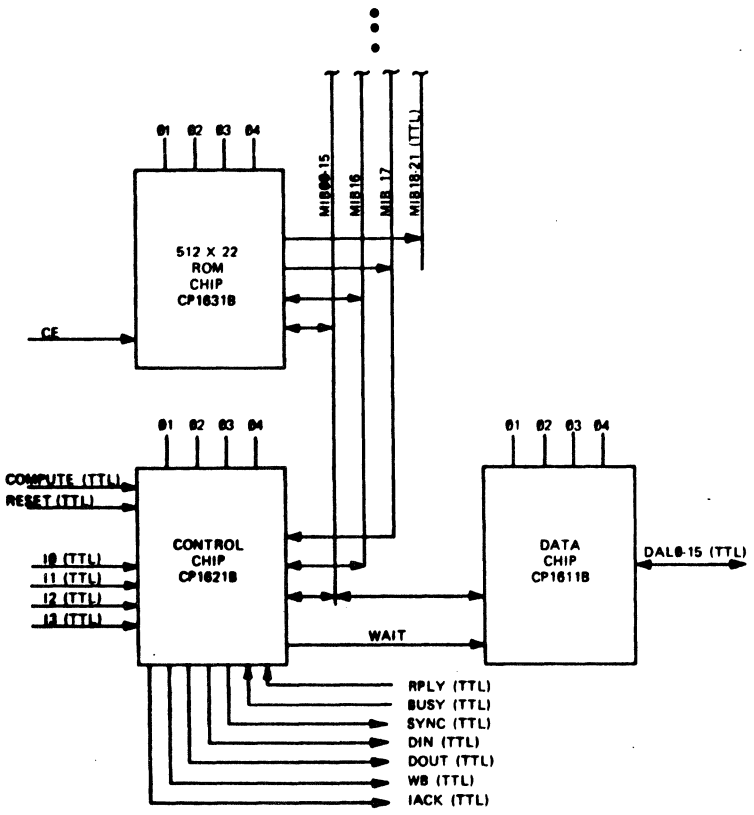


Figure 4-41. System Interconnections Microprocessor Set

4.3.5.1 BUS OPERATION.

The Microinstruction Bus consists of 22 lines. These lines are precharged by each of the microinstruction ROMs attached to the bus. Most generally, the lines are precharged high during Ø4 but MIB15 is precharged high during Ø3 and MIB16 is charged high during both Ø2 and Ø4. The microinstruction bus conveys its information by conditional discharge at the appropriate phase times. The microinstruction bus is a MOS compatible 4 phase bidirectional bus, and data on the bus is in logical complement form.

Table 4-17 defines the meaning of each of the lines on the bus at each phase time on the bus. Note that some lines have more than one meaning, depending upon the phase time.

4.3.5.2 MICROINSTRUCTION BUS ELEMENT DESCRIPTION.

The Microinstruction Bus is divided into seven partitions. These partitions can carry data in both directions. They assume different meanings at different points in the clock cycle of the processor set. This section describes each element and its meaning as a function of the phase time in each clock cycle.

MIB00-MIB10. These lines serve to carry data bidirectionally between microinstruction ROMs and the Data and Control Chips. They are unconditionally precharged Hi at Ø4 by the MICROMS attached to the bus. At the following Ø1, they may have two different meanings depending upon whether the Microinstruction presently being executed is a one cycle or a two cycle instruction.

If the Microinstruction executes one cycle, then the next Ø1 conveys the next microinstruction from the Microinstruction ROM to the Data Chip and Control Chip. If the Microinstruction takes two cycles, the second occurrence of Ø1 may result in data being transferred from the Data Chip to the Control Chip.

Table 4-17. Microbus Timing

	Ø1	Ø2	Ø3	Ø4
MIB00-MIB10	– Microinstruction from ROM to Data Chip and Control – Second Cycle of 2-Cycle Instruction: Data from Data Chip to Control Chip.	– Address Data to MICROM	– Address must remain valid	– Precharge by MICROM
MIB11-MIB14	– Microinstruction from ROM to Data Chip and Control Chip – Second Cycle of 2-Cycle Instruction: Data from Data Chip to Control Chip.			– Precharge by MICROM
MIB15	– Microinstruction from ROM to Data Chip and Control Chip. – Second Cycle of 2-Cycle Instruction: Data from Data Chip to Control Chip.		– Precharge by MICROM	– Conditional Jump Results
MIB16	– Load Return Register	– Precharge by MICROM.	– Disable MICROM outputs at next Ø1	– Precharge by MICROM
MIB17	– RNI			– Precharge by MICROM
MIB18-MIB21	– TTL Outputs Valid			– Precharge by MICROM

At Ø2, address data is transferred from the Location Counter in the Control Chip to the MICROM. At Ø3, the address data remains valid on the bus. The processor cycle is completed by an unconditional precharge of these lines by the MICROM at Ø4.

MIB11-MIB14. These lines serve to convey Microinstruction data from the Microinstruction ROM to the Data and Control Chips in much the same fashion as the preceeding lines do. These lines are unconditionally precharged Hi by the MICROM at Ø4. At Ø1

of a single cycle, they are conditionally discharged to represent Microinstruction Data from the Microinstruction ROM to the Data Chip and the Control Chip.

If the preceeding Microinstruction was of the 2-cycle variety, the second occurrence of Ø1 on these lines can carry data from the Data Chip to the Control Chip. These lines have no other significance during Ø2 and Ø3.

MIB15. MIB 15 is used to carry Microinstruction data from the ROM to the Data Chip and Control Chip and also to transfer the results of conditional jump tests from the Data Chip to the Control Chip.

This line is precharged Hi unconditionally by the Microinstruction ROM at Ø3. At Ø4, it can be conditionally discharged by the results of a conditional jump test.

At Ø1 this line conveys the microinstruction data from the ROM to the Data and Control Chip. In the case of a microinstruction whose execution takes two cycles, the second occurrence of Ø1 may serve to convey data from the Data Chip to the Control Chip. The contents of MIB15 are not significant at Ø2. The cycle is completed by an unconditional precharge of MIB15 at Ø3.

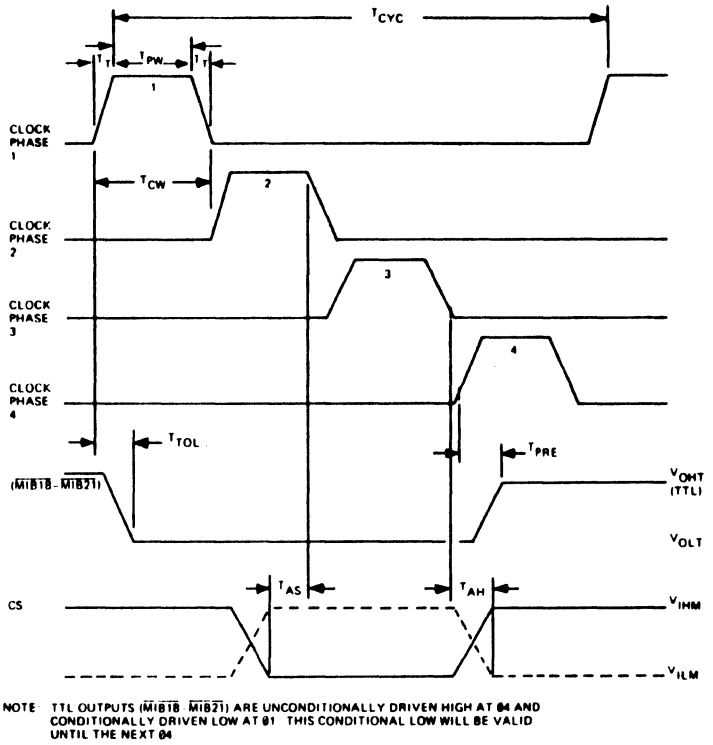
MIB16. This control line conveys data from the ROM to the Control and Data Chips and from the Control Chip to the Microinstruction ROM. It is unconditionally precharged HI at both Ø2 and Ø4. At Ø1 following Ø4, it may conditionally discharge low by the MICROM; in which case, the signal is interpreted by the Control Chip as a command to load the subroutine Return Register with the incremented content of the Location Counter. On Ø2, the lines are unconditionally precharged Hi. At Ø3, the Control Chip may conditionally discharge these lines. If it does so, this is an instruction to the selected MICROM (Microinstruction ROM) to disable its outputs at the next Ø1. In this fashion,

2 cycle instructions inhibit the transfer of new microinstructions from the MICROM to the Control Chip and the Data Chip.

MIB17. MIB17 has but one purpose. This is to convey the READ NEXT INSTRUCTION imperative from the MICROM to the Control Chip. This line is unconditionally precharged by the MICROMs attached to the Microinstruction Bus and conditionally discharged at Ø1. A discharge indicates that the RNI imperative is required.

MIB18-MIB21. These lines are not part of the Microinstruction Bus in that they do not transfer data between the Microinstruction ROM and the Data Chip and the Control Chip. They are TTL level outputs capable of driving one TTL load per line. They represent four bits in the Microinstruction word and are made valid at the same time as the other outputs of the MICROM, Ø1. The lines are unconditionally precharged Hi by the MICROM at Ø4 and conditionally discharged Low according to the contents of the word at Ø1. The user may set these outputs in any fashion he chooses. They remain valid in the inclusive interval Ø1 to Ø3. Figure 4-42 illustrates their timing.

WAIT Line. The WAIT Line from the Control Chip to the Data Chip establishes whether or not the Data Chip is in the RUN or WAIT mode. Whenever the WAIT control line is in the Low state, the Data Chip is in the RUN mode and the Microinstruction will be loaded into the MIR register and executed. This line is normally Low and must be driven Hi during Ø4 to cause the Data Chip to enter the WAIT state. It always returns to Low when the beginning edge of the Ø1 clock appears.



NOTE: TTL OUTPUTS ($\overline{MIB18}$-$\overline{MIB21}$) ARE UNCONDITIONALLY DRIVEN HIGH AT Ø4 AND CONDITIONALLY DRIVEN LOW AT Ø1. THIS CONDITIONAL LOW WILL BE VALID UNTIL THE NEXT Ø4

Figure 4-42. TTL Output Timing CP1631B

4.3.5.3 DATA CHIP.

The CP1611B Data Chip interacts with a subset of the Microinstruction Bus. Lines MIB00-MIB15 are present at the Data Chip. They convey Microinstructions from the Microinstruction ROM to the Data Chip, and they convey data from the Data Chip to the Control Chip. MIB15 also has a use as a control line conveying the results of conditional jump sets.

During the first Ø1 of a two cycle instruction, data is transferred from the microinstruction ROM into the MIR register of the Data Chip. The microinstruction bus is not sampled during Ø2 and Ø3. At Ø4, the Data Chip may conditionally discharge MIB15 to indicate the results of a conditional jump test. At the second occurrence of Ø1 of a two cycle instruction, the Data Chip may or may not transfer data to the Control Chip. If the two cycle instruction was of the Jump class or word operation class, the Data Chip will not transfer data to the Control Chip. If the two cycle instruction was a LTR or IW instruction (with the appropriate bits in the control field properly set), then the Data Chip transfers 16 bits of data into the Control Chip. This data is gated into the Translation Register on the Control Chip.

During the execution of a one cycle microinstruction, data is only transferred from the microinstruction ROM to the Data Chip. This transfer occurs at Ø1. Figure 4-43 illustrates the timing of the Microinstruction bus interface as seen by the Data Chip.

4.3.5.4 CONTROL CHIP.

The CP1621B Control Chip interfaces with the MIB00-MIB17 as the Data bus. Data can come to the Control Chip from both the Data Chip and the Microinstruction ROM. From the Microinstruction ROM, the Control Chip can receive microinstructions into

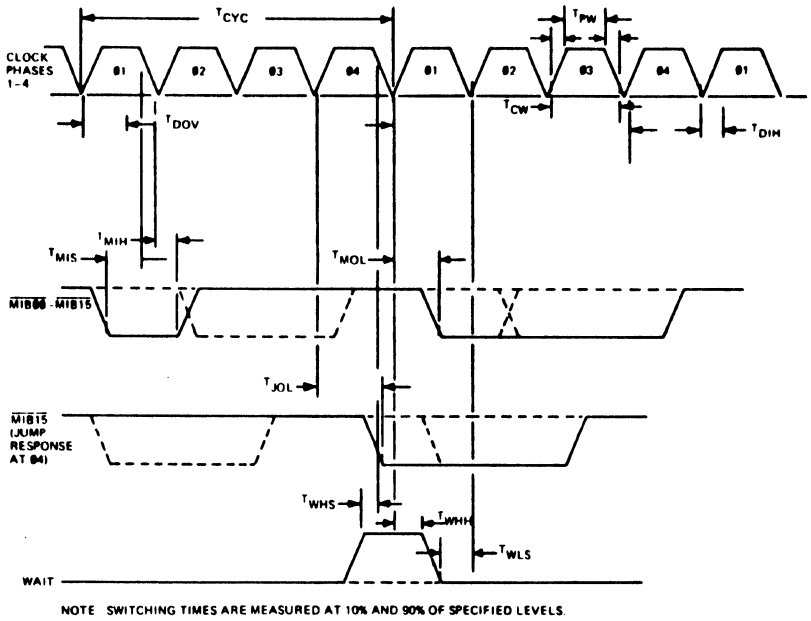


Figure 4-43. Microinstruction Bus Interface Timing

the MI register. There these microinstructions are used to drive the Master Control function and are also present in the event that the Microinstruction is of the jump variety and a jump is required. Data from the microinstruction ROM also is presented on MIB16, which controls whether or not the Return Register will be loaded with the incremented contents of the Location Counter. MIB17 from the Microinstruction ROM, controls whether or not the RNI translation will be invoked. Data from the Data Chip to the Control Chip can be passed over MIB00-MIB15 as a result of an LTR instruction or an IW instruction. MIB15 is also used by the Data Chip to send the results to the Control Chip's Master Control function informing it of the results of a conditional jump test.

The Control Chip can generate a WAIT signal which goes to the Data Chip, and causes Data Chip operations to be suspended, pending the completion of an I/O operation. The Control Chip also generates a signal placed on MIB16 which enables or disables the outputs of selected microinstruction ROM. This is used

during the second cycle of the execution of two cycle instructions to avoid conflicts on the microinstruction bus. The timing of the microinstruction bus as seen from the Control Chip is illustrated in Figure 4-44.

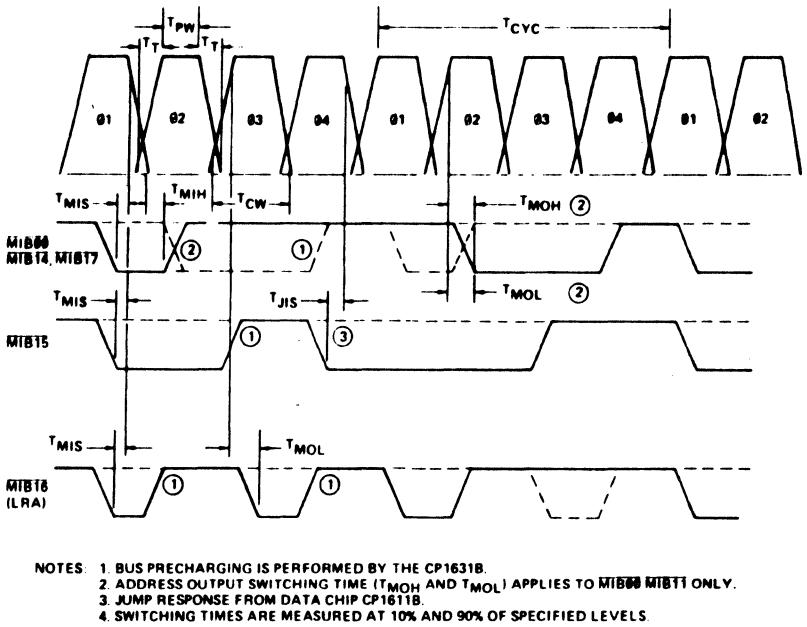


Figure 4-44. Control Chip Microinstruction Bus Timing

4.3.5.5 MICROINSTRUCTION ROM.

The Microinstruction ROM interfaces with all 22 lines of the Microinstruction bus. It receives Microinstruction addresses on MIB00-MIB10 from the Location Counter in the Control Chip. It also receives an ENABLE/DISABLE signal on MIB16 from the Control Chip. It sends data on MIB00-MIB17 to the Control Chip and the Data Chip. MIB00-MIB15 are presented to both the Control Chip and the Data Chip. These 16 bits comprise microinstruction data for the Data Chip and the Control Chip to interpret. MIB16 is properly part of the microinstruction and controls whether or not the Return Register will be loaded with the incremented contents of the Location Counter. MIB17, also properly part of the microinstruction, controls whether or not the matrix represented by Arrays 1 and 2 will force a

user specified State code that will cause the "READ NEXT INSTRUCTIONS" translation to be performed. The interface of the microinstruction ROM to the microinstruction bus is illustrated in Figure 4-45.

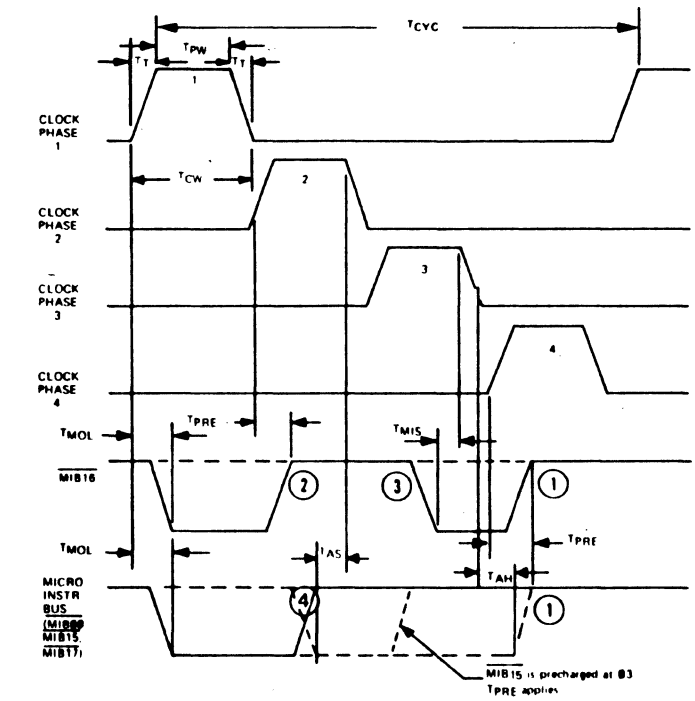


Figure 4-45. Microinstruction Bus Timing CP1631B

4.3.6 DATA ACCESS.

Communications between the MCP1600 system and attached peripheral devices or memory is conducted via a path called the Data Access Bus. The Data Access Bus consists of 16 data lines (DAL00-DAL15), 7 control lines and 4 interrupt lines. The operation of these elements is controlled by the Input/Output class of instructions.

The Data Access provides a single 16 bit bidirectional path to and from the processor for transfer of data and addresses.

A number of different types of bus arrangements are possible with the data access, such as:

- Common data and address
- Separate data and address
- Separate input and output busses

The Data Access Bus provides for 16 bit address and either 8 or 16 bit data (addresses are conventionally deemed to be byte addresses) and provide a maximum addressing capability of 65k bytes.

In accessing data, no distinction is made between memory and peripheral units or between instructions, data, control, or status. Each byte or word of information, regardless of function, is assigned an address and is referenced by means of this address.

In essence, then the Data Access Bus can be conceived to consist of the following elements:

- Data Lines (DAL00-DAL15)
- Control Lines (SYNC, REPLY, DATA-OUT or DOUT, DATA-IN or DIN, WRITE/BYTE, IACK, BUSY)
- Processor Control Lines (10, 11, 12, 13, COMPUTE, RESET)

The remainder of this section describes each of these elements of the data bus and their interactions and timings.

4.3.6.1 INPUT/OUTPUT INSTRUCTIONS.

This section breaks the I/O Instruction class into 5 classes. These classes are:

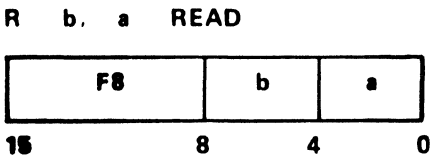
- Control Instruction consisting of the READ and WRITE INSTRUCTIONS that serve mainly to operate the Control Lines on the Data Access and to provide addresses.

- Data Transfer Instruction, consisting of the INPUT and OUTPUT instructions which serve to pass data along the Data Access lines.

- ACKNOWLEDGE instructions which serve to respond to interrupts.

Figure 4-46 illustrates the condition testing performed prior to executing the Input/Output instructions. The execute function is described in the following sections.

READ Instructions. The READ instructions take a 16 bit address from a designated register pair and transfer it to the M register. When the address becomes valid on the bus, (during the following Ø1) the SYNC line is made high. The variations on the READ instruction are primarily for address manipulation and easing the coding of I/O routines. They cause the address source registers to be modified in some fashion. As far as the system designer is concerned, the pertinent operation of this instruction class is that it causes the selected address to be placed on the Data Access lines (DAL00-DAL15) and the SYNC line to be raised. The addressed device asserts the REPLY Line when ready for the data transfer. A flow diagram of the READ instruction operation is shown in Figure 4-47.



The 16-bit address in Registers Rb:Ra is transferred to the M Register and a DATA READ operation is initiated. Registers Rb and Ra are not changed.

Timing: 1 cycle

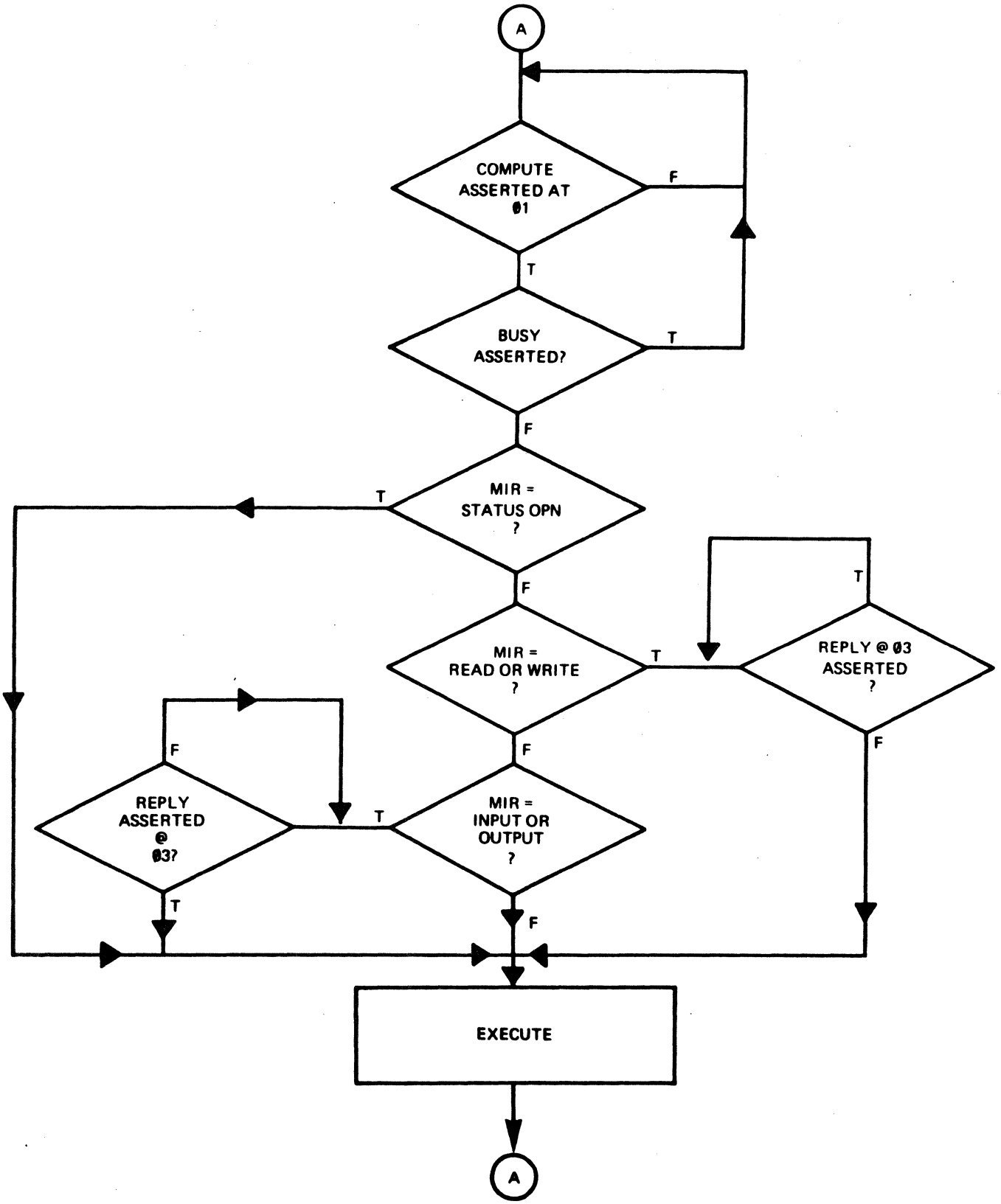


Figure 4-46. I/O Instruction Condition Testing

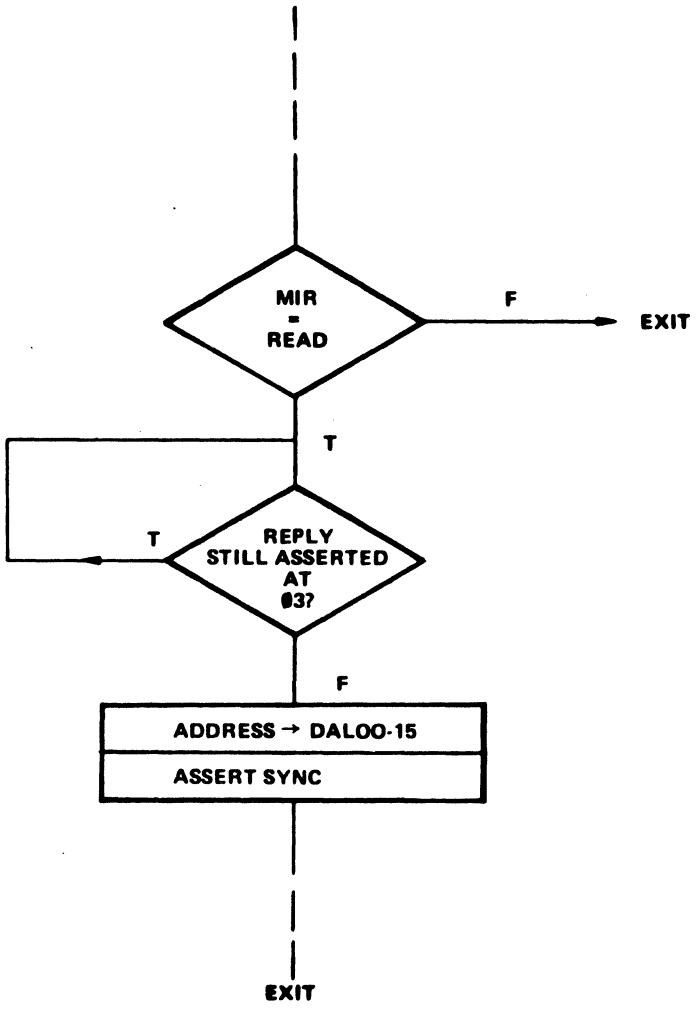
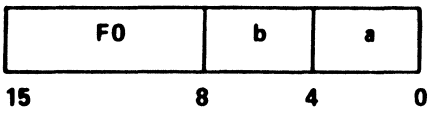


Figure 4-47. Execution of Read Instruction

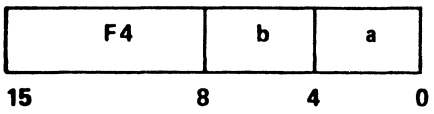
RIB1 b, a READ AND INCREMENT BYTE BY 1



The 16-bit address in Registers Rb:Ra is transferred to the M Register and a DATA READ operation is initiated. The contents of Register Ra are incremented by 1. Register Rb is not changed.

Timing: 1 cycle

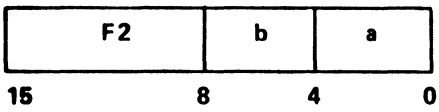
RIB2 b, a READ AND INCREMENT BYTE BY 2



The 16-bit address in Register Rb:Ra is transferred to the M Register and a DATA READ operation is initiated. The contents of Register Ra are incremented by 2. Register Rb is not changed.

Timing: 1 cycle

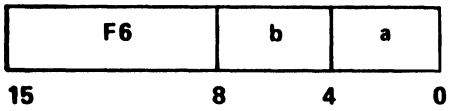
RIW1 b, a READ AND INCREMENT WORD BY 1



The 16-bit address in Registers Rb:Ra is transferred to the M Register and a DATA READ operation is initiated. The word in Ra+1:Ra is incremented by 1.

Timing: 2 cycles

RIW2 b, a READ AND INCREMENT WORD BY 2



The 16-bit address in Registers Rb:Ra is transferred to the M Register and a DATA READ operation is initiated. The word in Ra+1:Ra is incremented by 2.

Timing: 2 cycles

Write Instructions. This class of instructions causes an address in a designated register pair to be placed on the Data Access lines DAL00-DAL15. The SYNC line is raised as the address becomes valid at the next occurrence of Ø1. Also the WRITE-BYTE line is raised during the next occurrence of Ø1. When the addressed device is ready to transfer data, it asserts the REPLY line.

The comments about address source register manipulation pertain to this class of instructions as they do to the READ class of instructions described above.

A DATA WRITE operation is distinguished from a DATA READ operation by the assertion of WRITE/BYTE at the same time the address becomes valid on the bus. Note carefully that this signal later is used to describe data length when the data becomes valid on the bus.

A flow diagram of the WRITE instruction operation is shown in Figure 4-48.

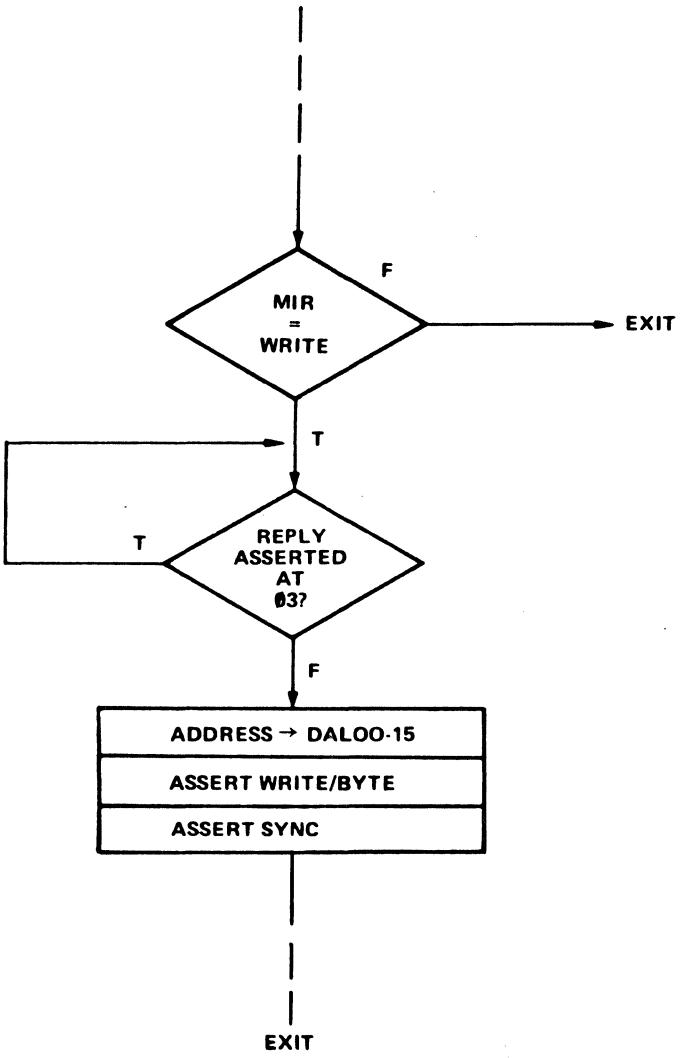
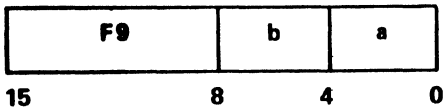


Figure 4-48. Execution Write Instruction

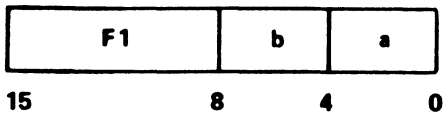
W b, a WRITE



The 16-bit address in Registers Rb:Ra is transferred to the M Register and a DATA WRITE operation is initiated. Registers Rb and Ra are not changed.

Timing: 1 cycle

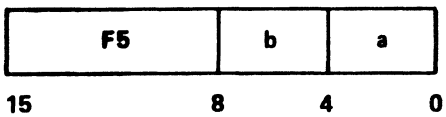
WIB1 b, a WRITE AND INCREMENT BYTE BY 1



The 16-bit address in Registers Rb:Ra is transferred to the M Register and a DATA WRITE operation is initiated. The contents of Register Ra are incremented by 1. Register Rb is not changed.

Timing: 1 cycle

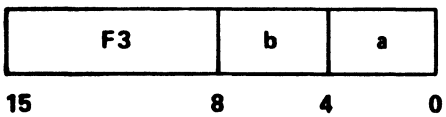
WIB2 b, a WRITE AND INCREMENT BYTE BY 2



The 16-bit address in Registers Rb:Ra is transferred to the M Register and a DATA WRITE operation is initiated. The contents of Register Ra are incremented by 2. Register Rb is not changed.

Timing: 1 cycle

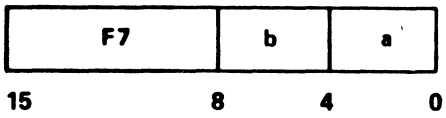
WIW1 b, a WRITE AND INCREMENT WORD BY 1



The 16-bit address in Registers Rb:Ra is transferred to the M Register and a DATA WRITE operation is initiated. The word in Ra+1:Ra is incremented by a.

Timing: 2 cycles

WIW2 b, a WRITE AND INCREMENT WORD BY 2



The 16-bit address in Registers Rb:Ra is transferred to the M Register and a DATA WRITE operation is initiated. The word in Ra+1:Ra is incremented by 2.

Timing: 2 cycles

INPUT Instructions. The INPUT Instructions control transfer of data from external devices on the Data Access to the processor. The data present on the data access is input by the instruction to the specified register or register pair. The INPUT class of instructions will not execute until a REPLY signal has been received from the device addressed by the previous READ instructions. Refer to a description of the REPLY signal below for the timing required. When this instruction is executed, it sets DATA-IN high to inform the addressed device that it should place its data on the bus. This instruction terminates by making SYNC and DATA-IN Low on the Ø2 after completion.

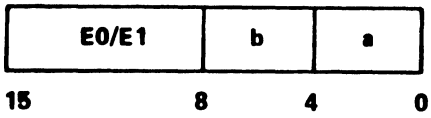
Note that the instructions INPUT STATUS BYTE and INPUT STATUS WORD execute regardless of the state of the REPLY signal. Other

than this exception, these two instructions are identical to the rest of the INPUT class instructions.

Note also that there are no timing restrictions after the selection of a device. As long as the operations are executed in the proper sequence, no timeouts or arbitrary disciplines bother the system designer.

The flow diagram in Figure 4-49 illustrates the operation of the INPUT instruction.

IB b, a INPUT BYTE



The 8-byte from the Data Lines, as specified by b, is placed in Register Ra. Code E1 causes the condition flags, except C, to be updated. The Read Data Access operation is terminated unless Bit 6 is a one which allows a Read-Modify Write (RMW) requiring termination by an output instruction. The instruction will not execute until after a Reply signal has been received from the addressed unit. The optional inputs are listed below:

```
b = 0   Upper Byte (Bits 15-8)
b = 1   Lower Byte (Bits 7-0)
b = 2   Upper Byte if M(0) = 1; Lower Byte if M(0) = 0
b = 3   Lower Byte if M(0) = 1; Upper Byte if M(0) = 0
b = 4   Upper Byte (Bits 15-8); RMW
b = 5   Lower Byte (Bits 7-0); RMW
b = 6   Upper Byte if M(0) = 1; Lower Byte if M(0) = 0; RMW
b = 7   Lower Byte if M(0) = 1; Upper Byte if M(0) = 0; RMW
```

Timing: 1 cycle (minimum)

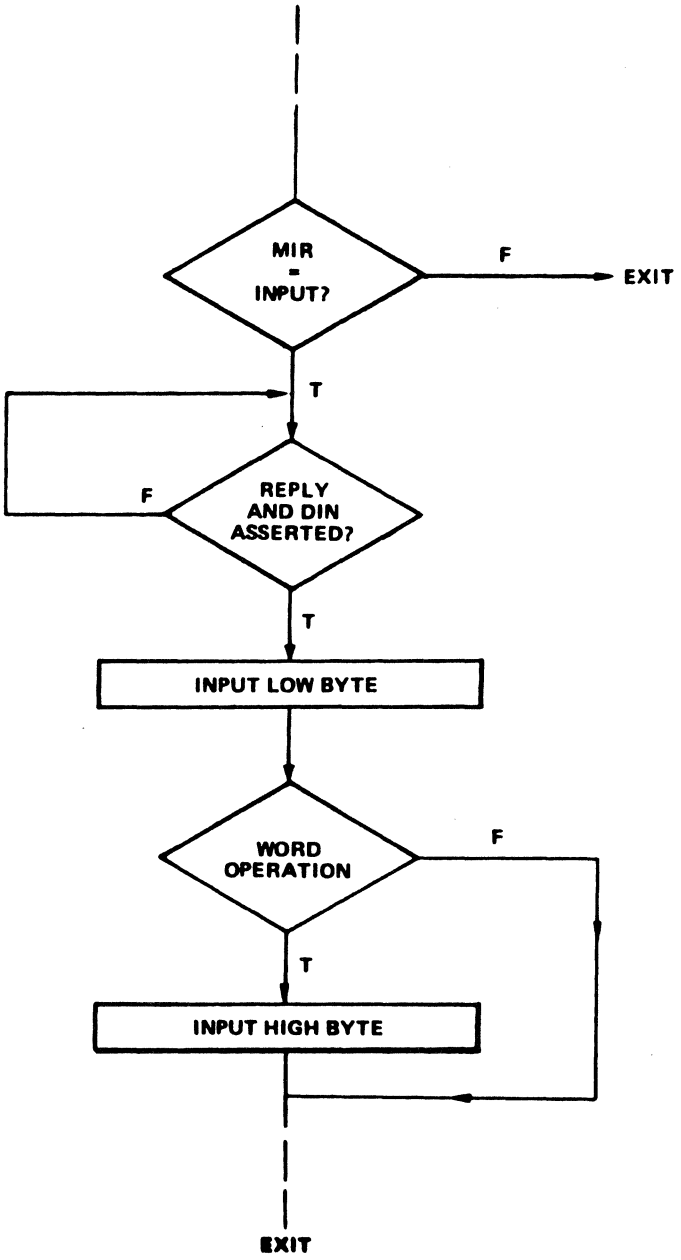
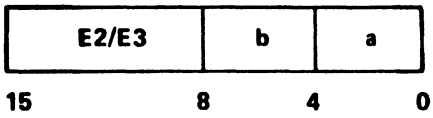


Figure 4-49. Execution of Input Instruction

IW b, a INPUT WORD

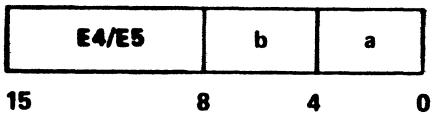


The 16-bit word from the Data Lines is placed in Registers Ra+1: Ra. Code E3 causes the condition flags, except C, to be updated. The Read Data Access operation is terminated unless Bit 6 is a one, which allows a Read-Modify-Write (RMW) operation requiring termination by an Output instruction. If Bit 4 or 5 is a one, the word on the Data Lines is loaded in the Translation Register; and at the same time, either Bits 6-4 or Bits 8-6 of the DAL are loaded into the G Register. The instruction will not execute until after a Reply signal has been received from the addressed unit. The Lower Byte is loaded before the Upper Byte. The b options are listed below:

b = 0
b = 1 Load TR; DAL 6-4 to GR; Sets ICS
b = 2 Load TR; DAL 8-6 to GR; Sets ICS
b = 3 Load TR; Sets ICS
b = 4 RMW
b = 5 Load TR; DAL 6-4 to GR; RMW; Sets ICS
b = 6 Load TR; DAL 8-6 to GR; RMW: Sets ICS
b = 7 Load TR; RMW; Sets ICS

Timing: 2 cycles (Minimum)

ISB b, a INPUT STATUS BYTE

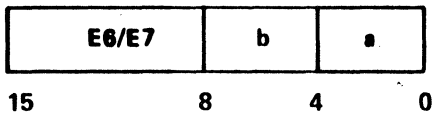


The 8-bit byte from the Data Lines, as specified by b, is placed in Register Ra. Code E5 causes the condition flags, except C, to be updated. The instruction will input regardless of the state of the Reply signal. These optional inputs are listed below:

b = 0 Upper Byte (Bits 15-8)
b = 1 Lower Byte (Bits 7-0)
b = 2 Upper Byte if M(0) = 1; Lower Byte if M(0) = 0
b = 3 Lower Byte if M(0) = 1; Upper Byte if M(0) = 0

Timing: 1 cycle

ISW b, a INPUT STATUS WORD



The 16-bit word from the Data Lines is placed in Registers Ra+1: Ra. Code E7 causes the condition flags, except C, to be updated. This instruction inputs regardless of the state of the Reply signal. The Lower Byte is loaded before the Upper Byte.

Timing: 2 cycles

OUTPUT Instructions. OUTPUT Instructions cause data to be transferred from the processor to the addressed peripheral devices. The OUTPUT instructions need the REPLY signal to be asserted before execution will proceed. DATA-OUT is set Hi by the processor during the first phase of the instruction to inform the peripheral device that data is presently available on the Data Access. Also during this time, the WRITE/BYTE signal may be set to indicate the length of data on the bus.

The OUTPUT STATUS command operates as the OUTPUT instructions except that it does not pay attention to the state of the REPLY line.

After the execution of the OUTPUT instruction, data becomes valid on the bus. Also, at this time, WRITE/BYTE is asserted if the data size is one byte. If the data size is a word, WRITE/BYTE becomes passive.

Refer to Figure 4-50 for the flow diagram of the OUTPUT instruction operation.

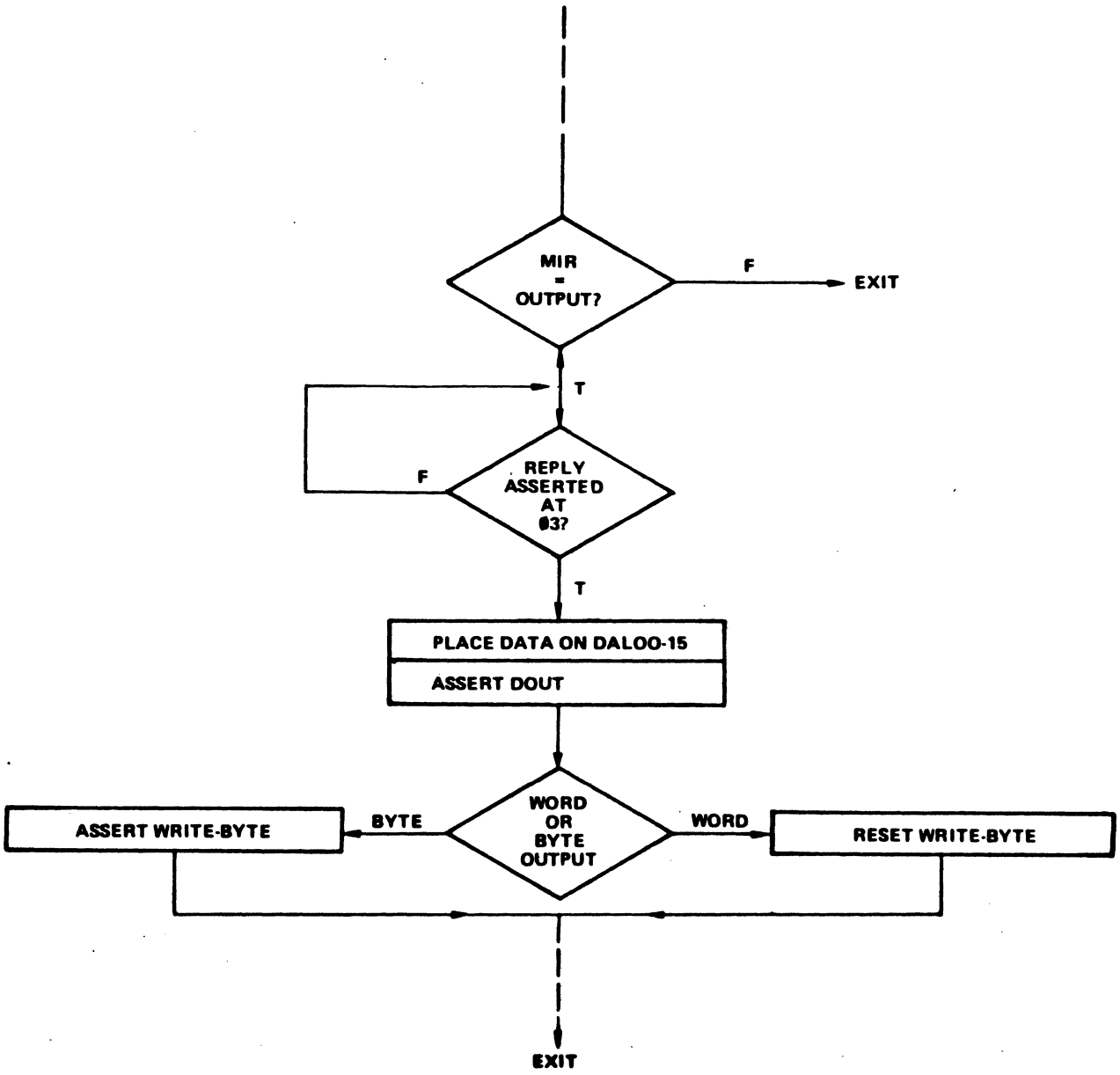
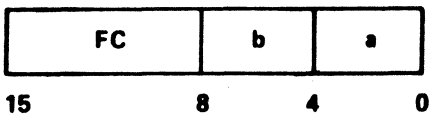


Figure 4-50. Execution of Output Instruction

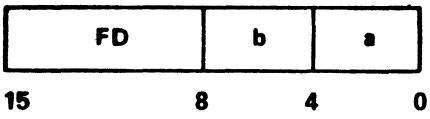
OB b, a OUTPUT BYTE



The 16-bit contents of Registers Rb:Ra are transferred to the M Register and the Data Lines. The Data Out signal is activated. Registers Rb and Ra are not changed. To provide proper operation with a 16-bit Data Path, b must equal a so that the same byte is placed in both byte positions of the M Register. Output does not take place until Reply has been received from the addressed unit.

Timing: 1 cycle (minimum)

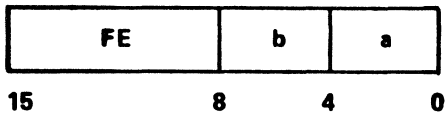
OW b, a OUTPUT WORD



The 16-bit contents of Registers Rb:Ra are transferred to the M Register and the Data Lines. The Data Out signal is activated. Registers Rb and Ra are not changed. Output does not take place until Reply has been received from the addressed unit.

Timing: 1 cycle (minimum)

OS b, a OUTPUT STATUS



The 16-bit contents of Registers Rb:Ra are transferred to the M Register and the Data Lines. Registers Rb and Ra are not changed. Output takes place regardless of the state of the Reply signal. This instruction is normally used without a Write instruction and cannot turn off Sync.

Timing: 1 cycle

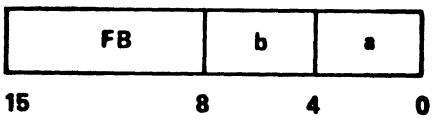
Interrupt Acknowledge Instructions. These instructions perform in identical fashion to the READ/WRITE instructions. Additionally, they raise the Interrupt Acknowledge (IACK) line. The Interrupt Acknowledge signal and a predetermined address placed on the Data Access bus by this instruction can be used as a signal to the I/O devices to inform the I/O set that the device requesting service should place its device number on the bus for transmission to the processor. Depending upon whether or not a READ acknowledge or a WRITE acknowledge was executed, the succeeding instruction may be an INPUT or OUTPUT Instruction respectively.

RA b, a READ ACKNOWLEDGE

The 16-bit address in Registers Rb:Ra is transferred to the M Register and a DATA READ operation is initiated with the Interrupt Acknowledge line high. Registers Rb and Ra are not changed. The Interrupt Acknowledge signal along with one or more address bits is used to form a signal which polls I/O units for the one interrupting the processor on the Interrupt line corresponding to the address. The processor inputs a byte or word containing the identification of the interrupting unit.

Timing: 1 cycle

WA b, a WRITE ACKNOWLEDGE



The 16-bit address in Registers Rb:Ra is transferred to the M Register and a DATA WRITE operation is initiated with the Interrupt Acknowledge line high. Registers Rb and Ra are not

changed. The Interrupt Acknowledge signal along with one or more address bits is used to form a signal which polls I/O units for the one interrupting the processor on the Interrupt line corresponding to the address. This unit receives the next data output.

Timing: 1 cycle

4.3.6.2 DATA/ADDRESS LINES.
Sixteen Data/Address Lines, denoted DAL00-DAL15, are used to transfer addresses and data from the processor and to receive data into the processor. This bus is TTL compatible and signals on it are logical true data.

An address is output by the Data Chip as a result of executing a READ or a WRITE instruction. It appears on the bus at Ø1 following the execution of the instruction. The address is valid on the bus during phases 2, 3 and 4.

Data is output by the processor as a result of executing an OUTPUT instruction. It appears on the bus at the first Ø1 after the microinstruction is executed and remains on the bus for a minimum of one cycle. Data is received from the Data Access into the processor by the INPUT instruction. It is clocked in at Ø4 of the instruction cycle.

4.3.6.3 CONTROL LINES.
This section describes the Control lines generated by the CP1621 Control Chip as a result of I/O instruction execution. It provides a functional description of the control lines. Their timing and interactions with the Data Access lines are further defined in Section 4.3.6.5.

Sync (TTL). The SYNC is a Control signal used to initiate and signify the length of a Data Access operation. SYNC is made high as soon as an address becomes valid. This occurs at Ø2

following a READ or WRITE. It remains high until the termination of the operation.

Reply (TTL). The REPLY is a Control signal used by the addressed unit to respond to the Processor's Data Access signals.

The REPLY signal must be high during Ø3 of the INPUT or OUTPUT microinstruction execution cycle in order to complete this operation. The REPLY signal is also interrogated by READ and WRITE microinstructions, and it must be low during Ø3 for these operations to take place.

Data-In (TTL). The DATAIN (DIN) is a Control signal from the Processor that causes the address unit to gate its Read data on the Data lines. It is made high at the time the address is removed from the lines, or one cycle after the SYNC is made high (the second Ø2 of the INPUT) and is a function of the READ instruction. The DATA-IN is made low at the end of the Input Byte or Input Word instruction or when SYNC is made low. This signal can be used to control the enabling of external TTL Tri-State Bus Driver/Receivers.

Data-Out (TTL). The DATA-OUT (DOUT) is a Control signal from the Processor which is made high at the same time as the Write data (Ø1 following the OUTPUT) is placed on the DAL bus by the Processor. It remains high for the duration of the OUTPUT instruction, dropping one phase before the data is taken off the DAL bus.

Write/Byte (TTL). The WRITE-BYTE (WB) is a Control signal from the Processor which is high during the time the address is on the bus to signify a WRITE rather than a READ operation; and is high during Data-Out to signify a Byte output rather than a Word output. To indicate an Output, it comes up at Ø1 following a WRITE.

Interrupt Acknowledge (TTL). The IACK is a Control signal from the Processor which signifies that the Processor is responding to an Interrupt. This signal is made high at the same time the SYNC is made high as a result of either 'RA' or 'WA' instructions, and stays high as long as SYNC is high.

Busy (TTL). The BUSY is a Control signal from an external unit to the Processor requesting access to the bus. The signal can be used, for example, by a DMA unit to access the memory. The BUSY signal is interrogated at Ø3 by the Processor every time READ or WRITE instructions are taking place. Whenever the BUSY signal is found to be one, the Processor enters a WAIT state inhibiting any access operation from taking place. The Processor will resume normal operation as soon as BUSY is turned off.

4.3.6.4 INTERRUPT, RESET, AND COMPUTE.
This section describes signals which would, in the normal scheme of things, be defined as processor control signals.

Reset. RESET is a TTL level line that may be controlled by an external device. Activation of the RESET line causes the Microprocessor to force 001 into the Location Counter. A NOP is also forced into the MIR and the MI registers. SYNC and DATA-IN are both reset. The RESET line can be wired to a POWER ON reset, or it may be used by the program for its own purposes.

Note that the activation of RESET is a hard action in that everything stops and the above mentioned conditions are forced.

Compute. COMPUTE is also a TTL level signal and it controls the processor's execution of microinstructions. The processor examines COMPUTE during every Ø1 to determine whether or not it should execute the present microinstruction. In the case

of a two cycle instruction, COMPUTE need be high only during Ø1 of the first cycle. Among other things, COMPUTE may be used to control single stepping of microinstructions. This line should not be confused with the WAIT signal on the MIB bus.

Interrupts. The external INTERRUPT lines of the MCP1600 system are I0-I3. These are microprogrammable. These lines provide inputs to Array 3 of the Programmable Translation Array and thus may be checked at certain user defined addresses in the Microprogram. As the examination of these lines is controlled by the contents of the Location Counter and the Macroinstruction being translated, it is easy for the microprogram to examine them for instance, before the FETCH cycle of every macroinstruction. There is no discipline associated with the use of these interrupt lines. Their state may be changed at any time. The system designer may implement his own interrupt control scheme by use of these lines and the ACKNOWLEDGE instructions.

4.3.6.5 INPUT/OUTPUT OPERATIONS.

The Data Access of the MCP1600, because of its flexibility, can be used to configure a variety of I/O schemes. Several of the schemes that can be implemented were mentioned in the introduction. This section will describe the interactions of the elements that comprise the Data Access and then define some canned operations that have been found useful. The combination of these canned operations and a discipline that structures interrupts and device addresses can result in a fast and powerful I/O structure.

Standard I/O Sequences. There are five standard I/O sequences that have been developed for the MCP1600 Processor System. These sequences provide for the orderly transfer of data to and from the processor. Two sequences provide for normal READs and WRITEs. One sequence provides a READ/MODIFY/WRITE capability which is useful for controlling random access memory devices. The last two sequences are READ/WRITE INTERRUPT ACKNOWLEDGE sequences.

Note that in the timing diagrams, the first clock cycle (Ø1, --- Ø4) is devoted to instruction interpretation. Bus operations do not begin to occur until the second clock cycle.

Write/Output Sequence. This sequence consists of two instructions:

WRITE	Device Number
OUTPUT	Data Source

The write operation transfers data from the processor to the addressed unit. The Write is initiated by a Write instruction which transfers a 16-bit address to the Data Access port. The address is present on the lines for one cycle. Data is transferred from the processor registers to the Data Access port by an Output Byte or Output Word instruction. The operation is terminated after the data has been on the lines for a minimum of one cycle. When outputting a byte with a 16-bit data path, the same byte must be placed in both the upper and the lower bytes of the port and the addressed unit takes care of storing the byte in the proper half of the word as selected by the low-order address bit.

The time period between selection of the device by the WRITE instruction and the transfer of the data is not critical if the device controller latches the selection. The sequence is graphically shown in Figure 4-51.

Read/Input Sequence. This sequence consists of two instructions which select the device, then transfer data.

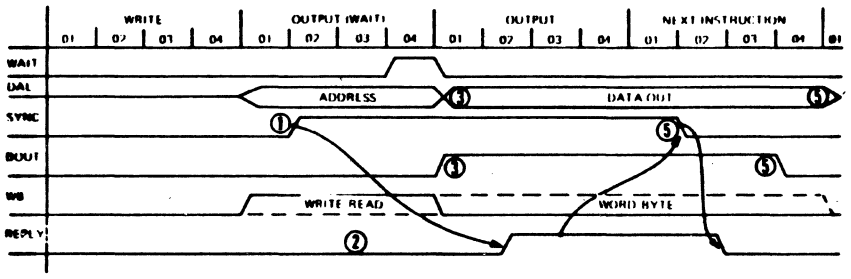
READ	Device Number
INPUT	Data Destination

The Read operation transfers data from the addressed unit to the processor. The Read is initiated by a Read instruction which transfers a 16-bit address to the Data Access port. The address is present on the lines for one cycle and then the

processor signals the addressed unit to put its data on the lines. Data is input to the processor registers by an Input Byte or Input Word instruction. The Read operation is terminated by either Input or Output instructions. An Input Byte instruction allows for selection of the Upper Byte, Lower Byte or the Byte selected by the lower-order bit of the address.

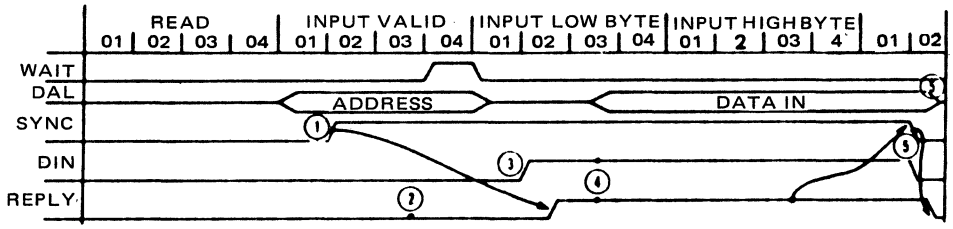
The time period between execution of the READ and INPUT INSTRUCTIONS is not critical if selection latching is used in the device controller.

Figure 4-52 illustrates the READ/INPUT sequence.



1 SYNC IS MADE HIGH AS SOON AS A VALID ADDRESS IS ON THE BUS (02)
2 REPLY BEING LOW ON 03 CAUSES THE OUTPUT INSTRUCTION TO BE EXECUTED REPEATEDLY
3 DOUT IS MADE HIGH ON 01 AS DATA OUT IS PLACED ON THE BUS
4 OUTPUT INSTRUCTION IS TERMINATED BY RAISING REPLY BEFORE 03
5 SYNC IS MADE LOW ON 02 DOUT IS MADE LOW ON 04 AND DAL IS TRISTATED ON 01
REPLY CAN BE TURNED OFF AS SOON AS SYNC IS MADE LOW

Figure 4-51. Write/Output Sequence



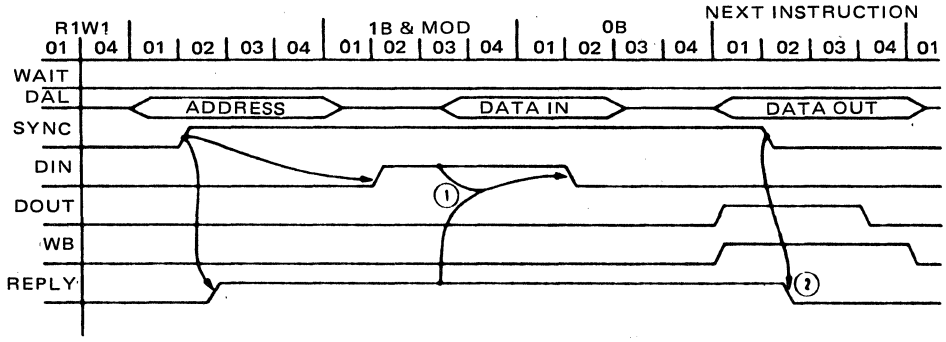
1 'SYNC' IS MADE HIGH AS SOON AS A VALUD ADDRESS IS ON THE BUS (02)
2 'REPLY' BEING LOW ON 03 CAUSES THE 'OUTPUT' INSTRUCTION TO BE EXECUTED REPEATEDLY
3 'DIN' IS MADE HIGH AS SOON AS THE ADDRESS IS TAKEN OFF THE BUS (02) AS FUNCTION OF READ MICROINSTRUCTION
4 'REPLY' AND 'DIN' MUST BE HIGH ON 03 IN ORDER FOR 'INPUT' INSTRUCTION TO TAKE PLACE
5 UPON TERMINATION OF 'INPUT' INSTRUCTION, 'SYNC' AND 'DIN' ARE MADE LOW ON 02

Figure 4-52. Read/Input Sequence

Read/Modify/Write Sequence (See Figure 4-53). The length of this sequence is variable. At least three instructions are required:

READ	Device Number
INPUT	Destination Register
	Microprogram can modify data
OUTPUT	Source Register

A Read/Modify/Write operation is both a Read and Write in a single Data Access operation. This provides for inputting data from an addressed unit, modifying it, and outputting the changed data to the same addressed unit. The Read/Modify/Write is initiated by a Read instruction which transfers a 16-bit address to the Data Access port. The address is present on the lines for one cycle and then data is placed on the lines by the addressed unit. Data is input to the processor registers by an Input Byte or Input Word instruction. The Data Access operation is not terminated as in the case of a normal Read, but continues for a subsequent output as specified by the Input instruction. This suspended period can be used to modify the data, if desired. An example might be an Increment Memory instruction. After modification, data is transferred from the processor registers to the Data Access port by an output byte or Output Word instruction. The operation is terminated after the data has been on the line for a minimum of one cycle.



1 'INPUT & MOD' WILL TURN 'DIN' OFF BUT WILL KEEP 'SYNC' ON
2 'REPLY' CAN BE MADE LOW AS SOON AS 'SYNC' AND 'DIN' ARE OFF

Figure 4-53. Read/Modify/Write Sequence

Interrupt Acknowledge Sequences. These two sequences provide the system designer with the ability to structure his own interrupt system. In addition to the normal READ or WRITE sequences, the processor will raise IACK at the specified time.

The combination of IACK and a special reserved device number on DAL00-15 could, for example, order the device controller requesting attention to return its device number on the subsequent INPUT instruction.

Figure 4-54 illustrates the operation of both the READ and WRITE Interrupt Acknowledge.

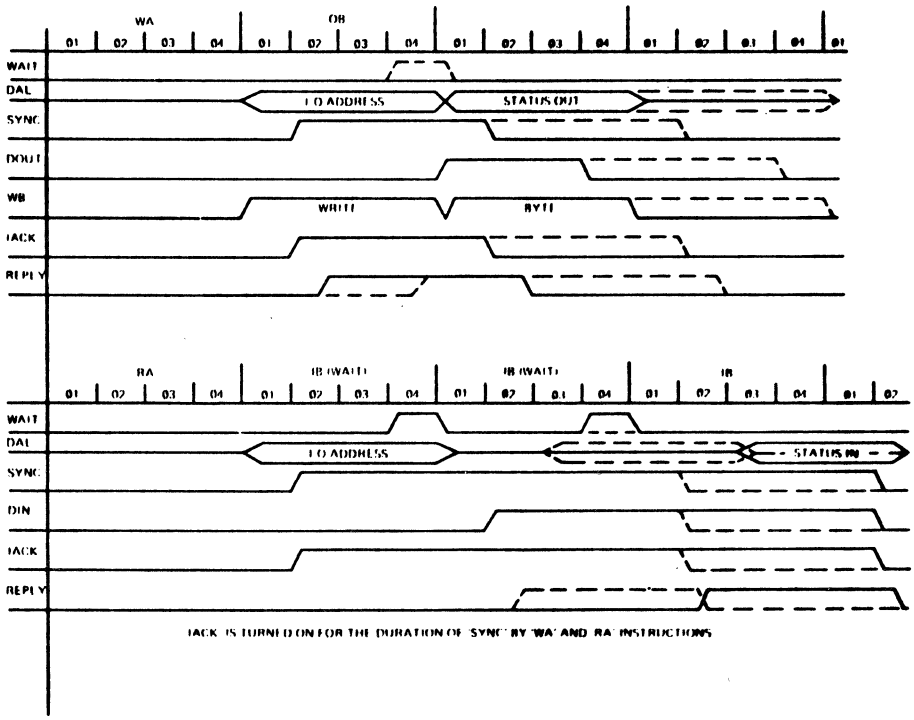


Figure 4-54. Interrupt Acknowledge Sequence

4.4 CIRCUIT MODULE DESCRIPTION.

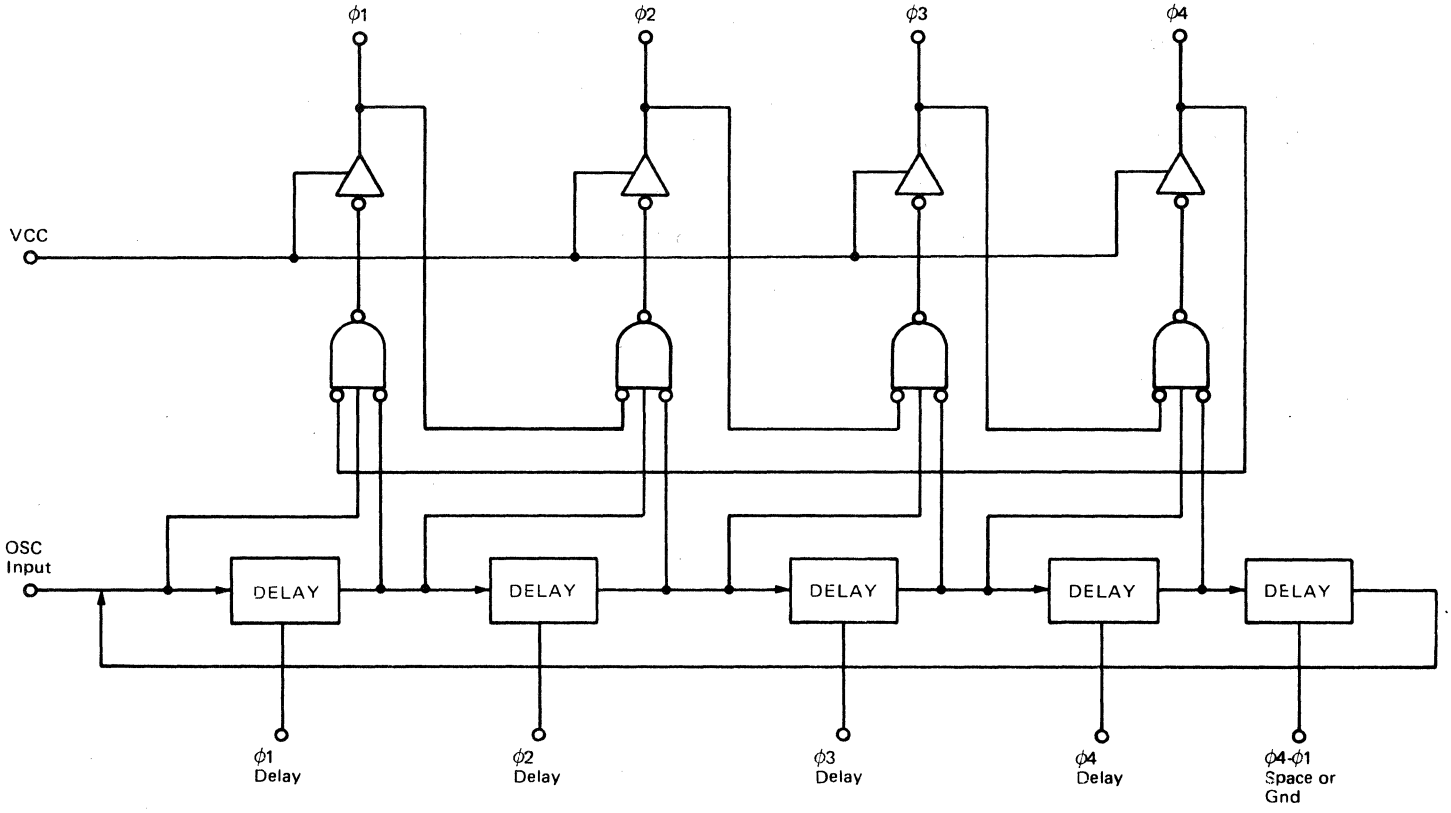
This Section describes the operation of the individual circuit packages (DIPS) contained on the AM-100 two-board set. The CPU processing is handled by the set of five DIPS as described in paragraph 4.3. The control and interface modules are also described in the following paragraphs with logic and connection diagrams for each one.

4.4.1 FOUR-PHASE CLOCK GENERATOR (Board 2.U12).
This device is a 4-phase clock generator using an external oscillator to initiate the phase-1 clock. By floating pins 9 and 15, the chip also operates in a free-running mode where the phase-1 clock starts due to internal oscillation. Each of the other three clock phases follow in sequence provided the chip does not see another positive going edge from the oscillator. Once phase-4 has returned to Ø volts, all four clock phases remain idle until the next positive going oscillator transition. If this transition occurs during one of the clock phases, the chip immediately tries to generate another phase-1. For stable operation, phase-4 should not overlap phase-1.

Each clock phase width is independently controlled by an externally applied capacitor to ground. A 1Ø to 2Ø pf capacitor provides for an output clock phase width of 1ØØ ns. Logic and timing are shown in Figure 4-55.

Pin Description

Pin No.	Symbol	Function
1	VBB	-5 volts
2	NC	No connection
3	Ø4 Out	Phase-4 Output
4	Ø3 Out	Phase-3 Output
5	Ø2 Out	Phase-2 Output
6	Ø1 Out	Phase-1 Output
7	VSS	Ground
8	VCC	+5 to +15 volts
9	Osc Input	Oscillator Input
10	VDD	+12 volts
11	Ø1 Cap	Phase-1 Capacitor
12	Ø2 Cap	Phase-2 Capacitor
13	Ø3 Cap	Phase-3 Capacitor
14	Ø4 Cap	Phase-4 Capacitor
15	Gnd	Ground
16	NC	No Connection



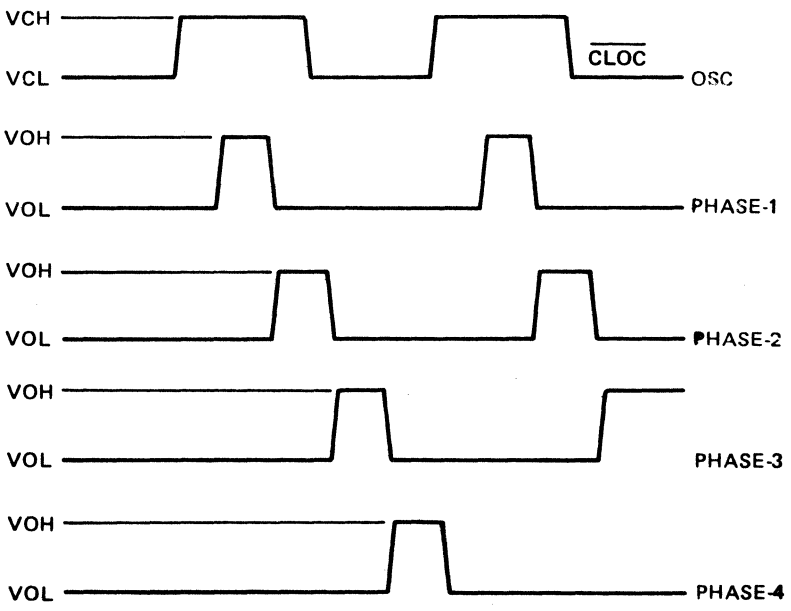
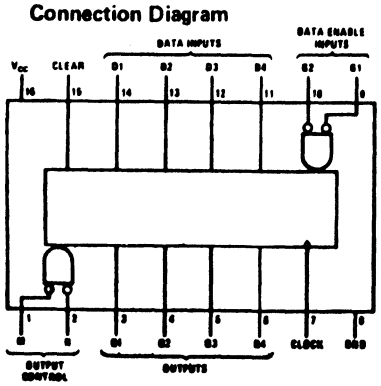


Figure 4-55. Four-phase Clock Logic and Timing

4.4.2 TRI-STATE 4-BIT D TYPE REGISTER (Board 1 U1, U9, U10). These four-bit registers contain D-type flip-flops with totem-pole TRI-STATE outputs, capable of driving highly capacitive or low-impedance loads.

Gated enable inputs are provided for controlling the entry of data into the flip-flops. When both data-enable inputs are low, data at the D inputs are loaded into their respective flip-flops on the next positive transition of the buffered clock input. Gate output control inputs are also provided. When both are low, the normal logic states of the four outputs are available for driving the loads or bus lines. The outputs are disabled independently from the level of the clock by a high logic level at either output-control input. The outputs then present a high impedance and neither load nor drive the bus line. Detailed operation and logic is shown in Figure 4-56.

Connection Diagram



Truth Table

INPUTS					OUTPUT
CLEAR	CLOCK	DATA ENABLE G1	DATA ENABLE G2	DATA D	Q
H	X	X	X	X	L
L	L	X	X	X	Q_0
L	↑	H	X	X	Q_0
L	↑	X	H	X	Q_0
L	↑	L	L	L	L
L	↑	L	L	H	H

When either M or N (or both) is (are) high the output is disabled to the high-impedance state; however, sequential operation of the flip-flops is not affected.

H = high level (steady state)
L = low level (steady state)
↑ = low-to-high level transition
X = don't care (any input including transitions)
Q_0 = the level of Q before the indicated steady state input conditions were established

Logic Diagram

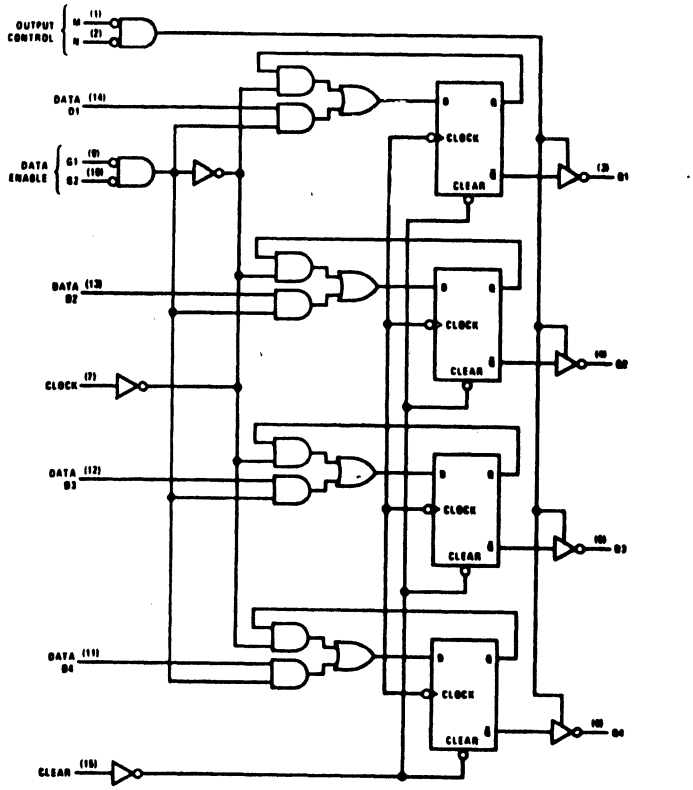


Figure 4-56. Tri-State 4-Bit D Type Register Connections

4.4.3 HIGH SPEED HEX INVERTER (Board 1, U7).
This device provides high speed low current interface logic. Logical inversion is provided with active pullups. See Figure 4-57 for logic and connections.

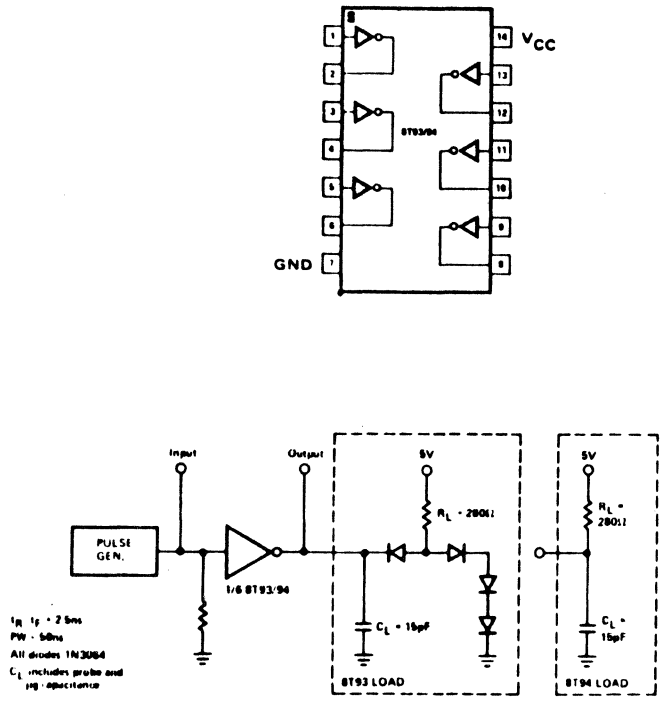
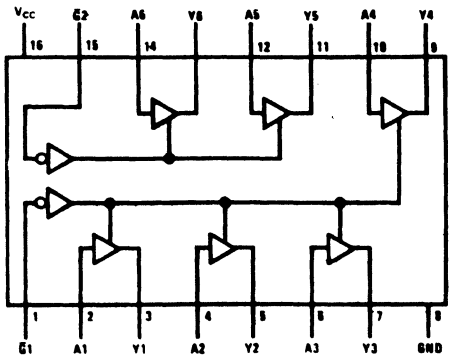


Figure 4-57. High Speed Hex Inverter Connections

4.4.4 TRI-STATE BUFFERS (Board 1, U5, U8, U22, U24, U26; Board 2, U45, U46, U48).

These devices provide six, two-input buffers in each package. One of the two inputs to each buffer is used as a control line to gate the output onto the high-impedance state, while the other input passes the data through the buffers. The outputs are placed in the tri-state condition by applying a high logic level to the control pins. See Figure 4-58 for logic diagram and truth table.

Logic and Connection Diagram



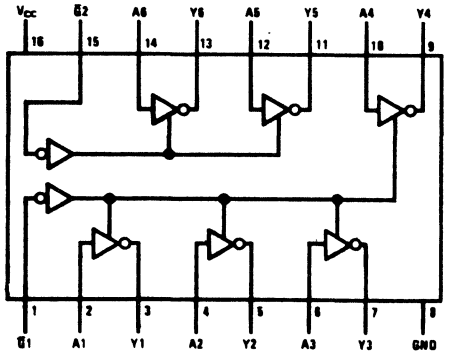
Truth Table

INPUTS		OUTPUT
$\overline{G}$	A	Y
H	X	Hi-Z
L	H	H
L	L	L

Figure 4-58. Tri-State Buffer Connections

4.4.5 HEX TRI-STATE BUFFERS (Board 2, U11, U47).
These devices provide six, two-input buffers in each package. One of the two inputs to each buffer is used as a control line to gate the output onto the high-impedance state, while the other input passes the data through the buffer. The outputs are placed in the tri-state condition by applying a high logic level to the control pins. See Figure 4-59 for logic diagram and truth table.

Logic and Connection Diagram

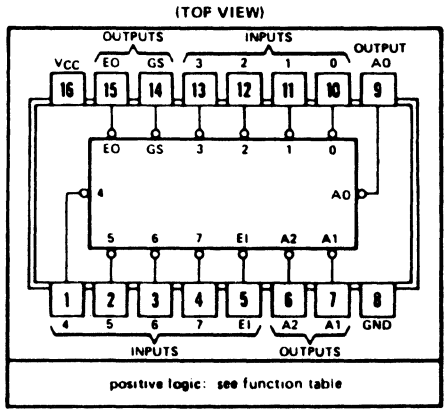


Truth Table

INPUTS		OUTPUT
$\overline{G}$	A	Y
H	X	Hi-Z
L	H	L
L	L	H

Figure 4-59. Hex Tri-State Buffer Connections

4.4.6 EIGHT-LINE TO THREE-LINE ENCODER (Board 2, U29, U30). This device encodes eight data lines to three line (4-2-1) binary (octal). Cascading circuitry (enable input E1 and output E0) are provided to allow octal expansion without the need for external circuitry. Data inputs and outputs are active at the low logic level. See Figure 4-60 for logic diagram and truth table.



FUNCTION TABLE

INPUTS								OUTPUTS				
EI	0	1	2	3	4	5	6	7	A2	A1	A0	GS	EO
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	X	X	X	X	X	X	X	L	L	L	L	L	H
L	X	X	X	X	X	X	L	H	L	L	H	L	H
L	X	X	X	X	X	L	H	H	L	H	L	L	H
L	X	X	X	X	L	H	H	H	L	H	H	L	H
L	X	X	X	L	H	H	H	H	H	L	L	L	H
L	X	X	L	H	H	H	H	H	H	L	H	L	H
L	X	L	H	H	H	H	H	H	H	H	L	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L	H

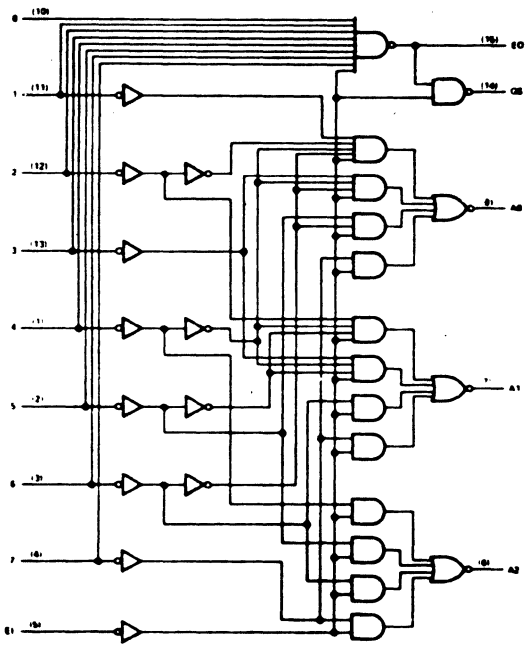
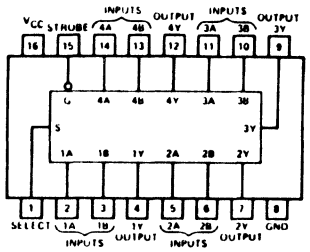


Figure 4-60. Eight-Line to Three-Line Encoder Connections

4.4.7 DATA SELECTOR/MULTIPLEXER (Board 1, U16, U17, U18, U21). These Schottky clamped devices select a four-bit word from one of the two sources and route it to the four outputs as true data (no inversion). See Figure 4-61 for logic diagram and connections.



POSITIVE LOGIC:
Low logic level at S selects A inputs.
High logic level at S selects B inputs.

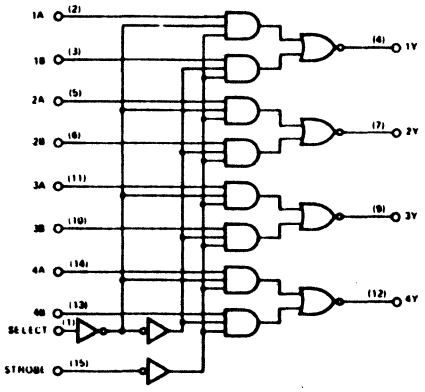


Figure 4-61. Data Selector/Multiplexer Connections

4.4.8 NAND TTL-TO-MOS DRIVER (Board 1, U2, U3).
This device is a monolithic, integrated, dual TTL-TO-MOS driver and interface circuit. It accepts standard TTL and DTL input signals and provides high-current and high voltage output levels suitable for driving MOS circuits. Specifically it can be used to drive address, control, and timing inputs for several types of MOS RAMS. See Figure 4-62 for logic and connections.

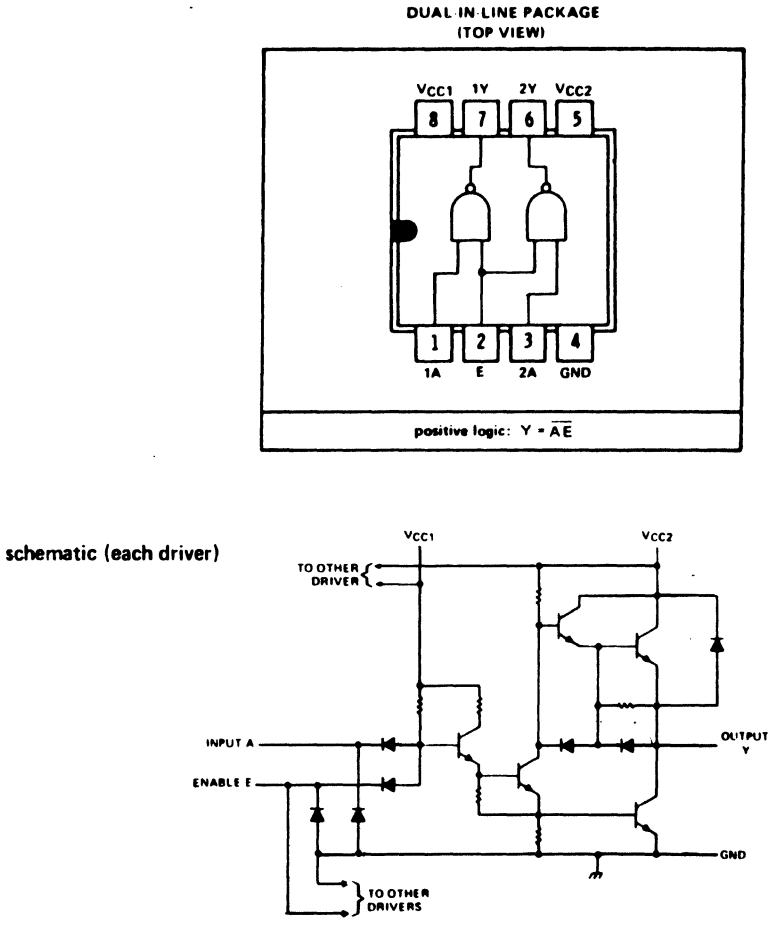


Figure 4-62. NAND TTL-TO-MOS Driver Connections

4.4.9 D POSITIVE-EDGE-TRIGGERED FLIP-FLOPS WITH PRESET AND CLEAR (Board 2, U6, U18, U19, U23, U31, U32, U34, U36). See Figure 4-63 for logic diagram and truth table.

TRUTH TABLE

INPUTS				OUTPUTS	
PR	CLR	CLK	D	Q	$\bar{Q}$
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H*	H*
H	H	↑	H	H	L
H	H	↑	L	L	H
H	H	L	X	Q0	$\bar{Q}$0

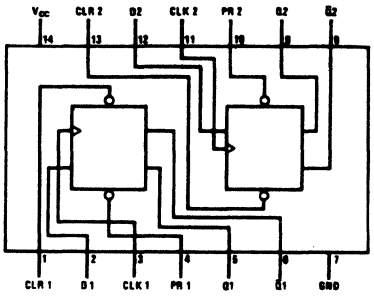
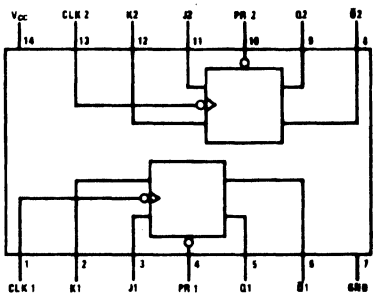


Figure 4-63. D Flip-Flop Connections

4.4.10 DUAL J-K NEGATIVE-EDGE TRIGGERED FLIP-FLOPS WITH PRESET (Board 2, U20, U28, U40, U50, U51). See Figure 4-69 for logic diagram and truth table.

Connection Diagram



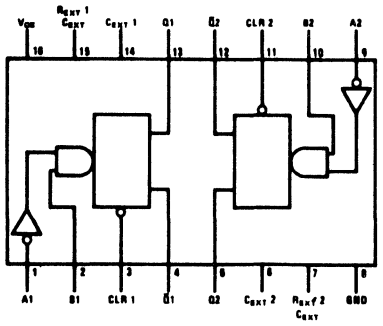
Truth Table

TRUTH TABLE

INPUTS				OUTPUTS	
PR	CLK	J	K	Q	$\bar{Q}$
L	X	X	X	H	L
H	↓	L	L	Q0	$\bar{Q}$0
H	↓	H	L	H	L
H	↓	L	H	L	H
H	↓	H	H	TOGGLE	
H	H	X	X	Q0	$\bar{Q}$0

Figure 4-64. J-K Flip-Flop Connections

4.4.11 DUAL RETRIGGERABLE ONE-SHOTS WITH CLEAR (Board 2, U22). See Figure 4-65 for logic diagram and truth table.



TRUTH TABLE

INPUTS			OUTPUTS	
A	B	CLR	Q	$\overline{Q}$
H	X	H	L	H
X	L	H	L	H
L	↑	H	⎍	⎎
↓	H	H	⎍	⎎
X	X	L	L	H

Notes: ⎍ = one high-level pulse, ⎎ = one low-level pulse.
To use the internal timing resistor of 54121/74121, connect R_{INT} to V_{CC}.
An external timing capacitor may be connected between C_{EXT} and R_{EXT}/C_{EXT} (positive).
For accurate repeatable pulse widths, connect an external resistor between R_{EXT}/C_{EXT} and V_{CC} with R_{INT} open-circuited.
To obtain variable pulse widths, connect external variable resistance between R_{INT} or R_{EXT}/C_{EXT} and V_{CC}.

Figure 4-65. One Shot Connections

4.4.12 DUAL VOLTAGE CONTROLLED OSCILLATORS (Board 2, U5). This device features two fully independent voltage-controlled oscillators (VCO's) in a single monolithic chip. The output frequency of each is established by a single external component, either a capacitor or a crystal, in combination with two voltage-sensitive inputs, one for frequency range and one for frequency control. An enable input is provided that can be used to start or stop the output pulses when it is low or high, respectively. The internal oscillator runs continuously, even while the output is disabled. A pulse synchronizer ensures that the first output pulse is neither clipped nor extended. Duty cycle of the output pulses is fixed at approximately 50 percent. See Figure 4-66 for logic diagram and connections.

The highly stable oscillator can be set to operate at any frequency between 0.12 Hz and 50 MHz typically. The output frequency can be approximated as follows:

$$f_o = \frac{500}{C_{ext}}$$

where: F_o = output frequency in MHz

C_{ext} = external capacitance in pF

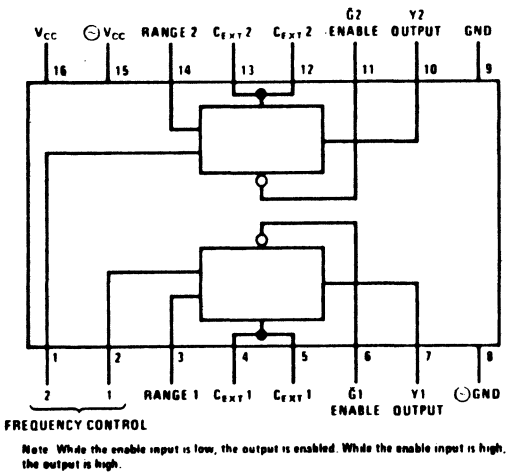
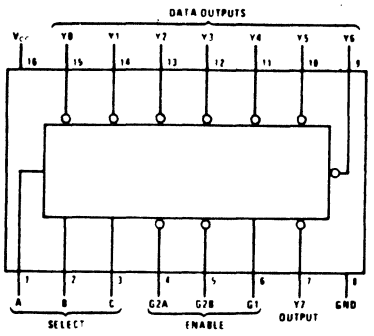


Figure 4-66. Dual Voltage Controlled Oscillator Connections

4.3.13 DECODER (Board 2, U7, U8, U42).

These are Schottky-clamped circuits designed for memory-decoding or data-routing applications requiring very short propagation delay times. This DIP decodes one of eight lines, based on the conditions at the three binary select inputs and the three enable inputs. See Figure 4-67 for logic diagram and truth table.

Connection and Logic Diagram



Truth Table

INPUTS					OUTPUTS							
ENABLE		SELECT										
G1	G2*	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

*G2 = G2A + G2B

H = High level, L = low level, X = don't care

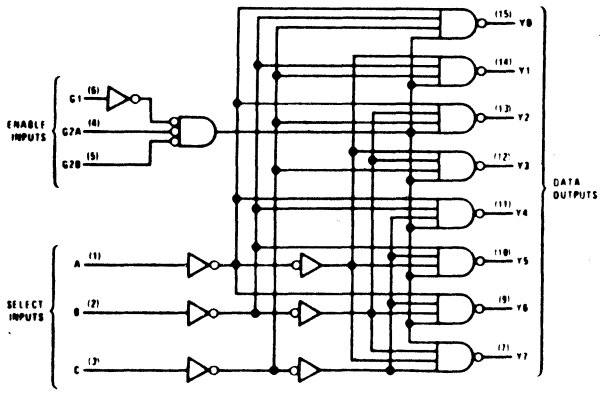
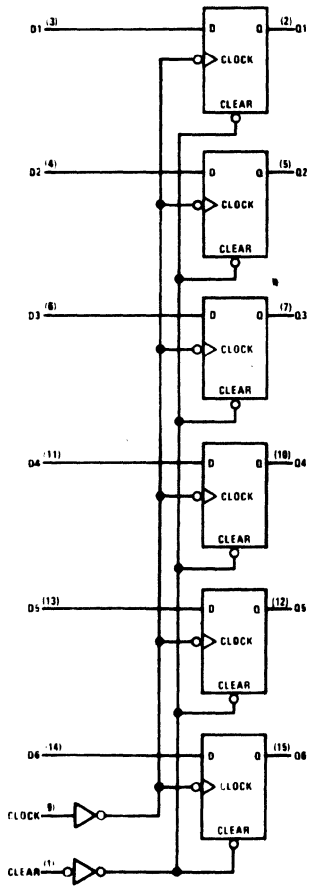


Figure 4-67. Decoder Connections

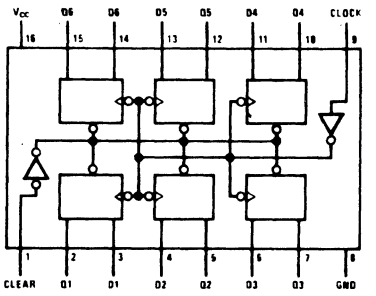
4.4.14 D FLIP-FLOPS WITH CLEAR (Board 1, U13, U14, U25; Board 2, U10).

These positive-edge-triggered flip-flops utilize TTL circuitry to implement D-type flip-flop logic. Information at the D inputs meeting the setup time requirements is transferred to the Q-outputs on the positive-going edge of the clock pulse. Clock triggering occurs at a particular voltage level and is not directly related to the transition time of the positive-going pulse. When the clock input is at either the high or low level, the D input signal has no effect at the output. See Figure 4-68 for logic diagram and truth table.

Logic Diagram



Connection Diagram



Truth Table

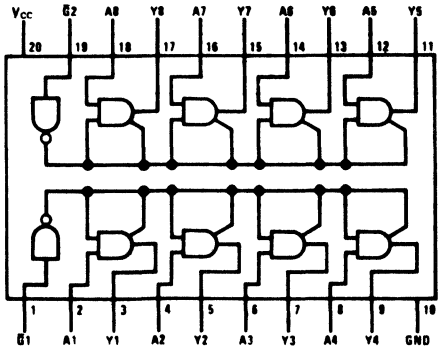
INPUTS			OUTPUTS	
CLEAR	CLOCK	D	Q	$\bar{Q}^{\dagger}$
L	X	X	L	H
H	↑	H	H	L
H	↑	L	L	H
H	L	X	Q_0	$\bar{Q}_0$

H = High Level (steady state)
L = Low Level (steady state)
X = Don't Care
↑ = Transition from low to high level
Q_0 = The level of Q before the indicated steady-state input conditions were established.
† = 175, LS175, and S175 only

Figure 4-68. D Flip-Flop Connections

44.15 TRI-STATE OCTAL BUFFERS (Board 1, U6, U11, U19, U27). This device provides six, two-input buffers in each package. One of the two inputs is used as a control line to gate the output into a high impedance state, while the other passes the data through the buffer. The outputs are placed in the tri-state condition by applying a high logic level to the control pins. See Figure 4-69 for logic diagram and truth table.

Logic Diagram



Truth Table

INPUTS		OUTPUT
$\overline{G}$	A	Y
H	X	Z
L	H	H
L	L	L

Figure 4-69. Tri-State Octal Buffer Connections

SECTION 5
MAINTENANCE AND TROUBLESHOOTING

5.0 INTRODUCTION.

The AM-100 circuit board performs to full capability with a minimum of maintenance. This Section describes maintenance procedures and procedures for handling warranty returns.

5.1 CIRCUIT BOARD CHECKOUT.

The AM-100 circuit board was fully tested before it left Alpha Microsystems and will operate satisfactorily in your system if the hardware and software requirements of Sections Two and Three of this manual are met. Should a problem arise after the circuit card has been in operation, perform the following preliminary checks to identify and locate the fault.

1. Check all cabling for proper seating of connectors.

2. Check the circuit board for proper seating in the slot.

3. Check all power connections for correct voltages.

4. Check all jumper options to ensure correctness for your application.

5. Verify that the fault is in the AM-100 and not either in the system or in the peripherals. This can best be accomplished with substitution of a known good circuit board if available.

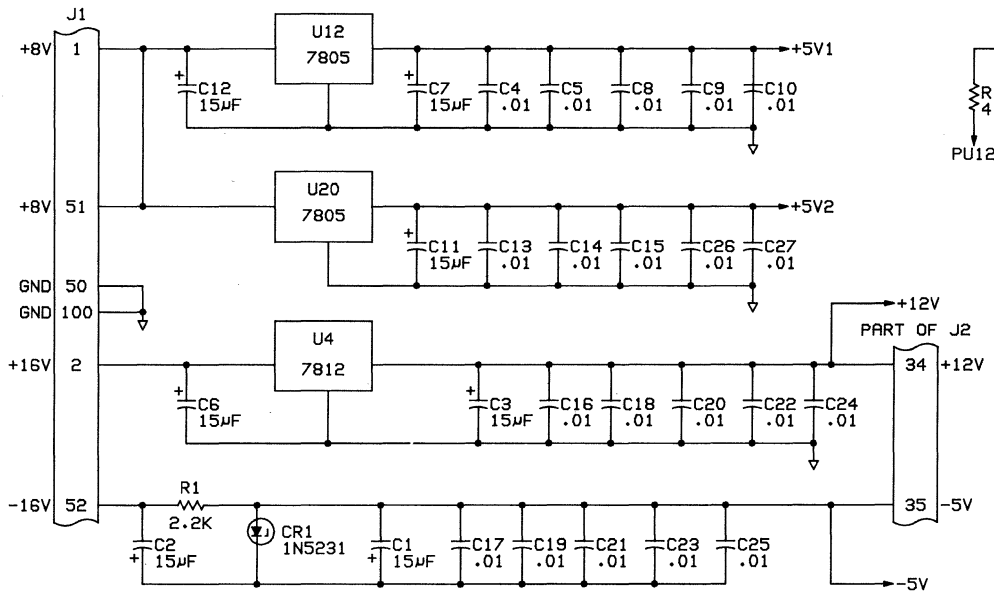
5.2 WARRANTY PROCEDURES.

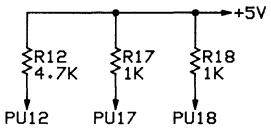
This circuit board is covered by warranty issued by Alpha Microsystems, Irvine, California. Complete details of the warranty are included with the circuit board. Should a problem arise with this circuit board, call your dealer or the Alpha Micro International Support Services Administrator for information.

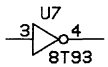
SECTION 6
SCHEMATIC AND PARTS LIST

APPR	DATE 10/27/78	EN. NO.	DESC.	LEVEL	
				IS	WAS
		ENA-00002-00	RTC SYNC	B03	B02

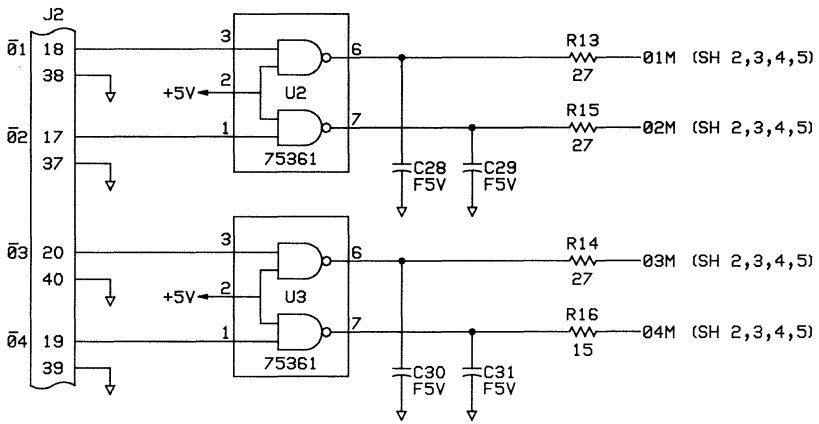
POWER







SPARE



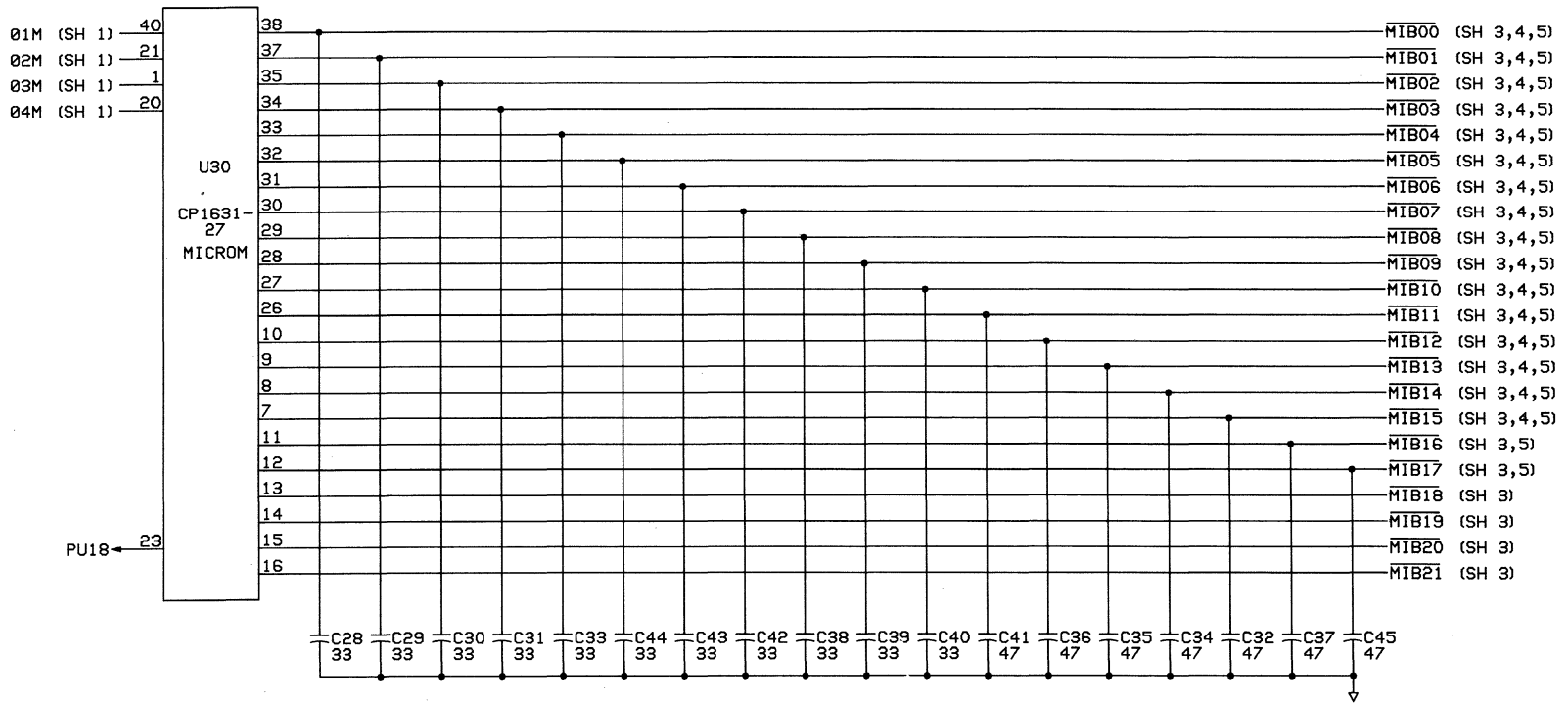
CLOCK

2. CAPACITORS ARE IN MICROFARADS.
1. RESISTORS ARE IN OHMS.

NOTES: UNLESS OTHERWISE SPECIFIED

ALPHA MICRO SYSTEMS
IRVINE, CA 92714

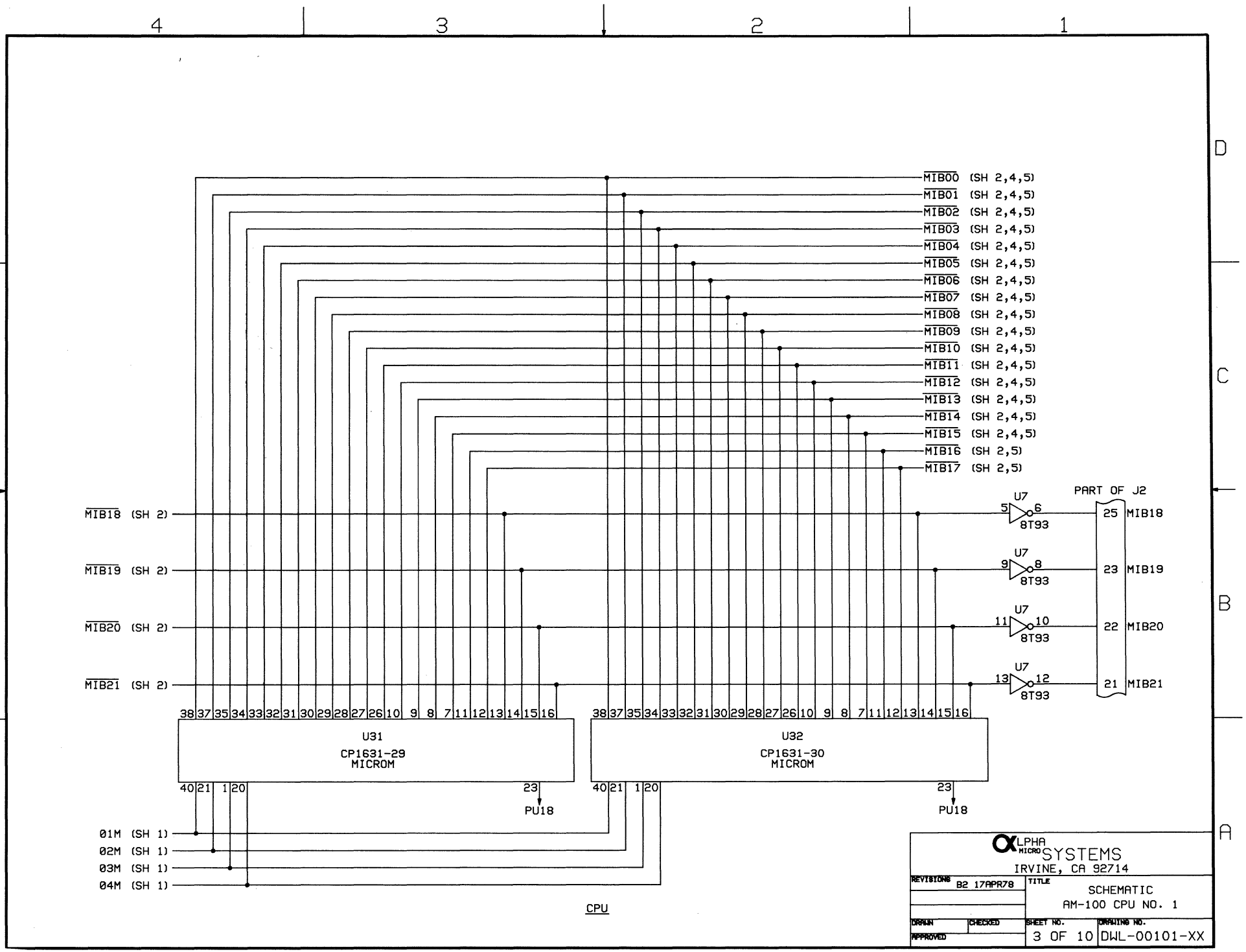
REVISIONS	TITLE	
B2 17APR78	SCHEMATIC AM-100 CPU NO. 1	
DRAWN / CHECKED / APPROVED	SHEET NO.	DRAWING NO.
	1 OF 10	DWL-00101-XX

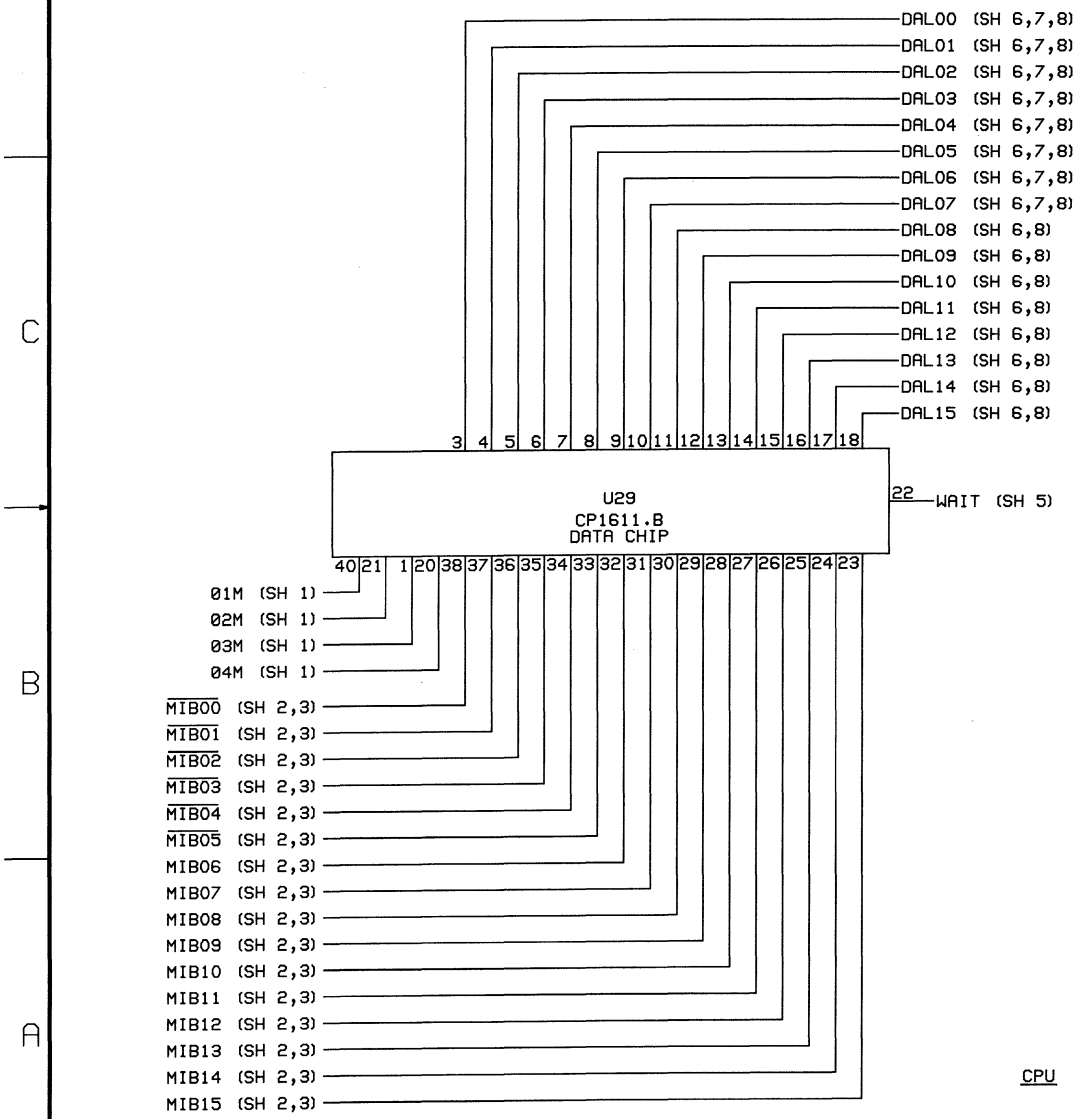


CPU

ALPHA MICRO SYSTEMS
IRVINE, CA 92714

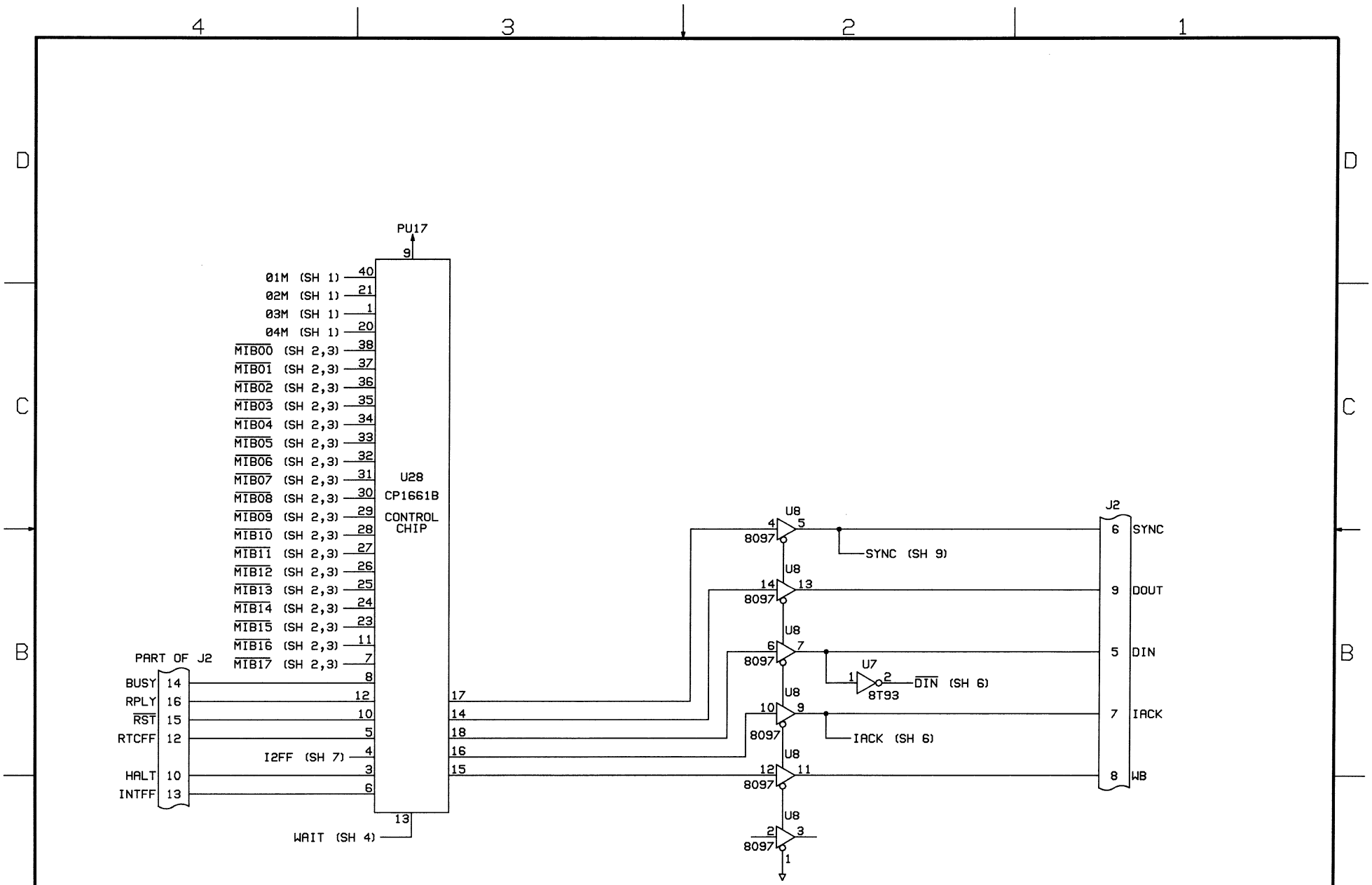
REVISIONS	TITLE
B2 17APR78	SCHEMATIC AM-100 CPU NO. 1
DRAWN / CHECKED / APPROVED	SHEET NO. 2 OF 10
	DRAWING NO. DWL-00101-XX


4
3
2
1
D
C
B
A
MIB00 (SH 2,4,5)
MIB01 (SH 2,4,5)
MIB02 (SH 2,4,5)
MIB03 (SH 2,4,5)
MIB04 (SH 2,4,5)
MIB05 (SH 2,4,5)
MIB06 (SH 2,4,5)
MIB07 (SH 2,4,5)
MIB08 (SH 2,4,5)
MIB09 (SH 2,4,5)
MIB10 (SH 2,4,5)
MIB11 (SH 2,4,5)
MIB12 (SH 2,4,5)
MIB13 (SH 2,4,5)
MIB14 (SH 2,4,5)
MIB15 (SH 2,4,5)
MIB16 (SH 2,5)
MIB17 (SH 2,5)
MIB18 (SH 2)
MIB19 (SH 2)
MIB20 (SH 2)
MIB21 (SH 2)
PART OF J2
U7
8T93
25 MIB18
23 MIB19
22 MIB20
21 MIB21
38 37 35 34 33 32 31 30 29 28 27 26 10 9 8 7 11 12 13 14 15 16
U31
CP1631-29
MICROM
U32
CP1631-30
MICROM
40 21 1 20
23
PU18
01M (SH 1)
02M (SH 1)
03M (SH 1)
04M (SH 1)
CPU
ALPHA MICRO SYSTEMS
IRVINE, CA 92714
REVISIONS
B2 17APR78
TITLE
SCHEMATIC
AM-100 CPU NO. 1
DRAWN
CHECKED
APPROVED
SHEET NO.
3 OF 10
DRAWING NO.
DWL-00101-XX



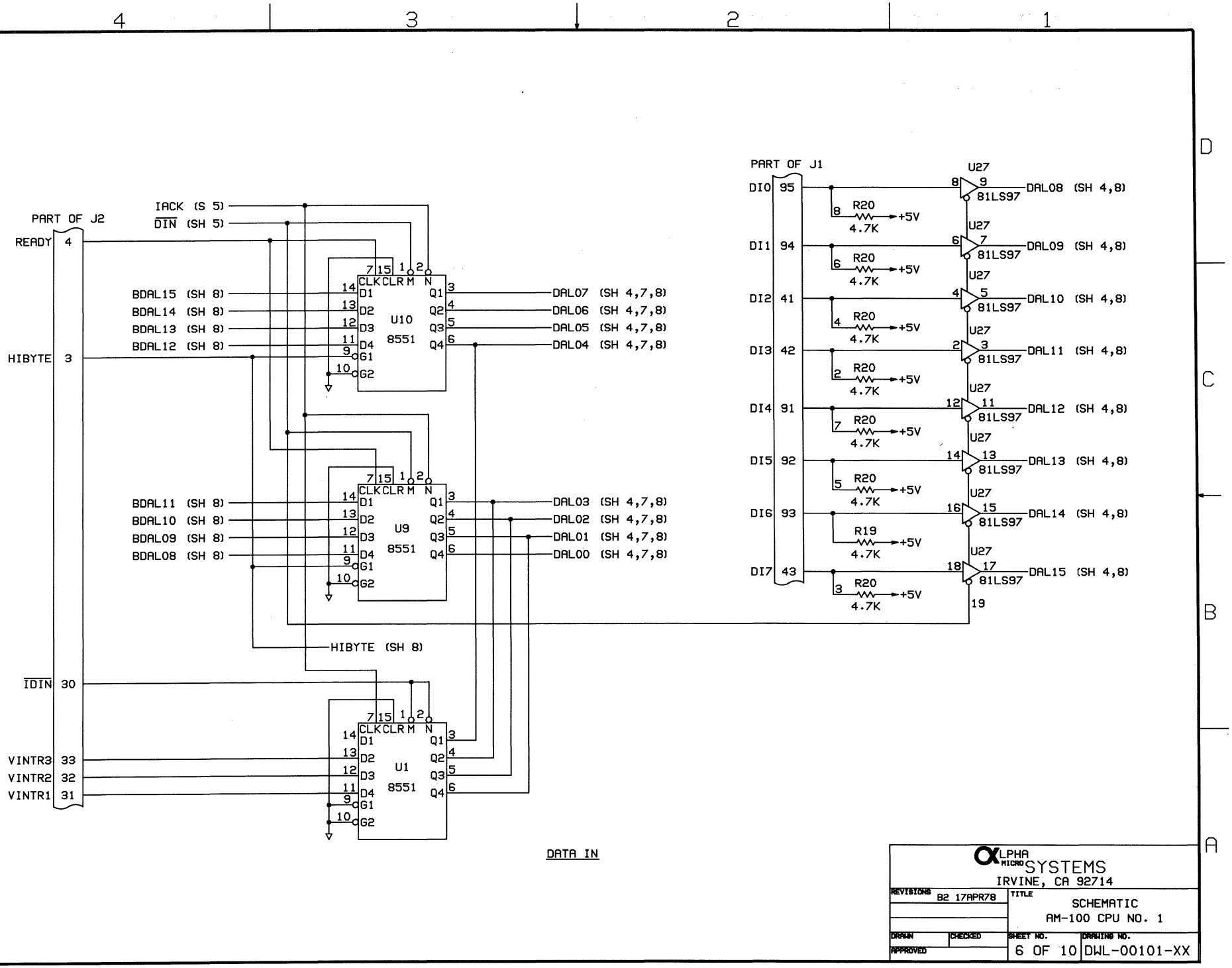
ALPHA MICRO SYSTEMS
IRVINE, CA 92714

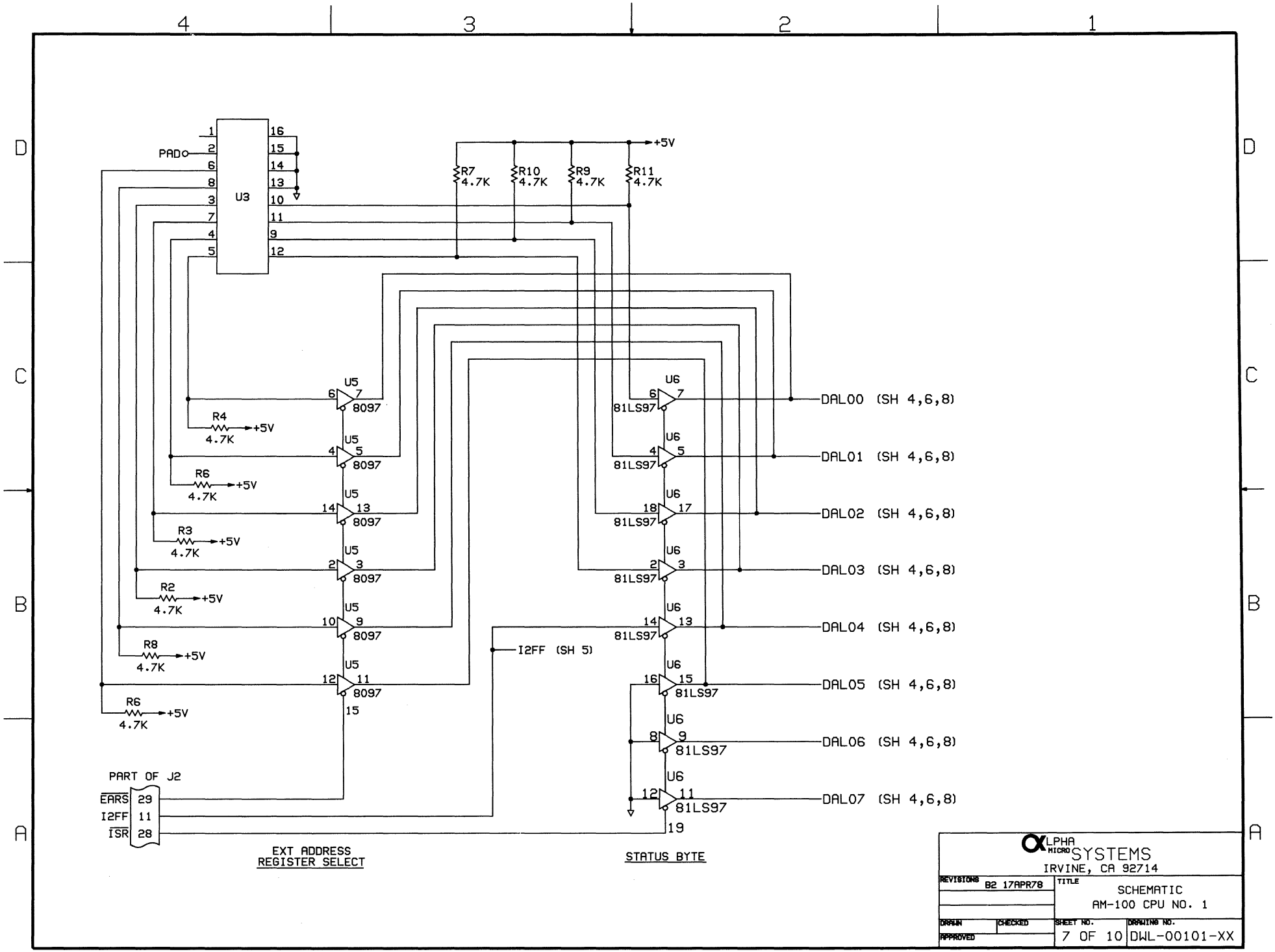
REVISIONS	TITLE
B2 17APR78	SCHEMATIC AM-100 CPU NO. 1
DRAWN / CHECKED / APPROVED	SHEET NO. 4 OF 10
	DRAWING NO. DWL-00101-XX

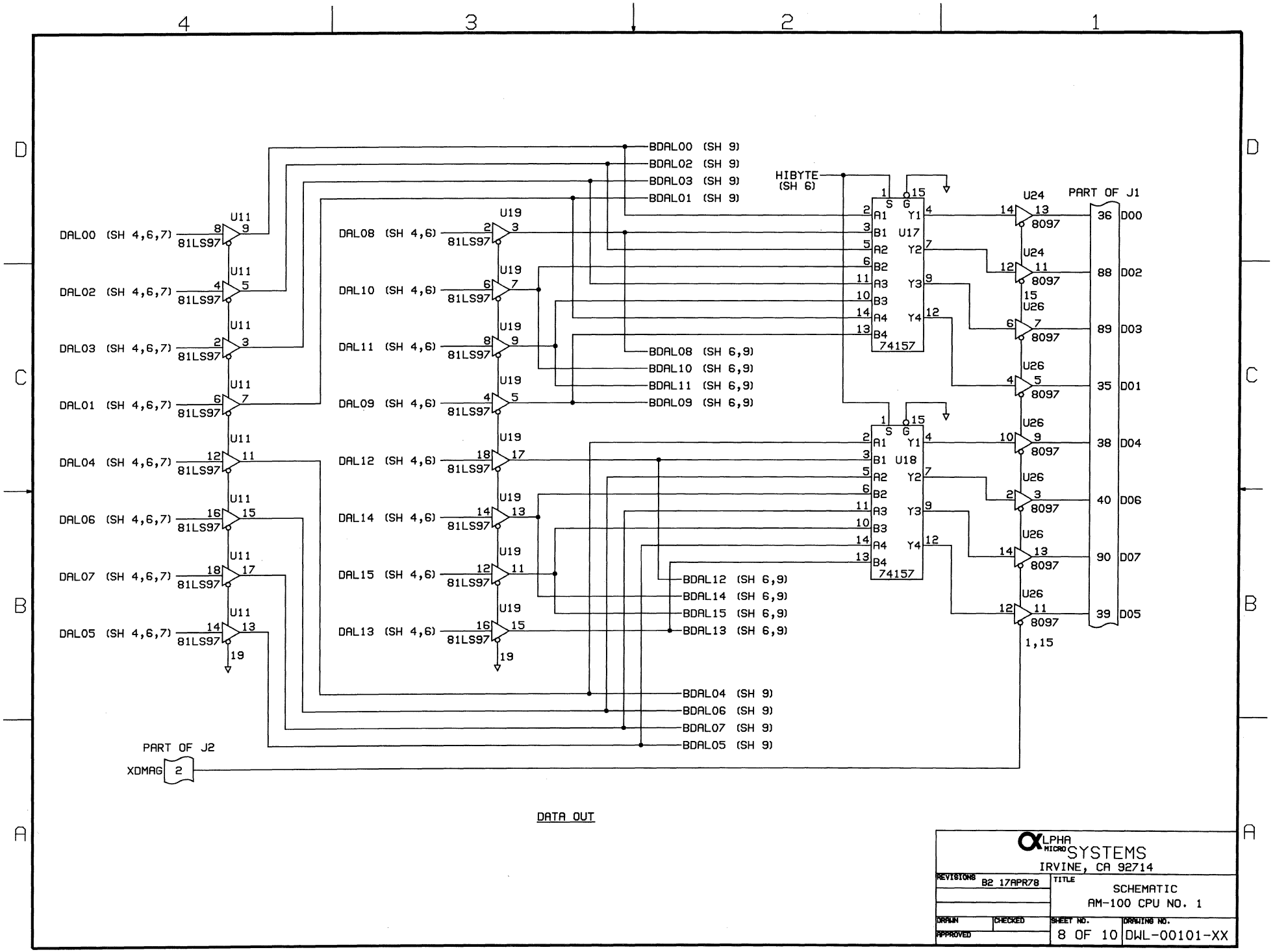


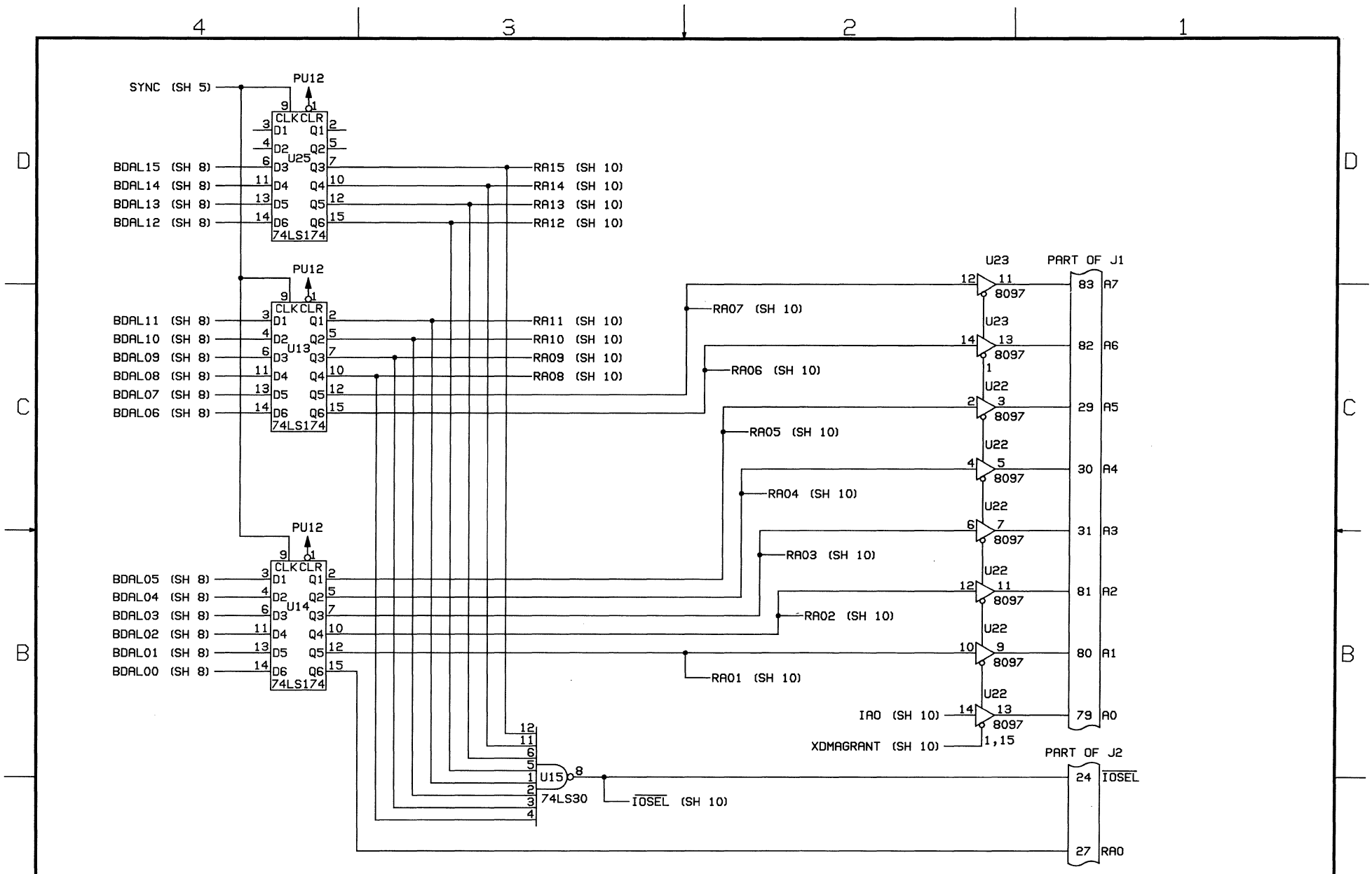
CPU

ALPHA MICRO SYSTEMS IRVINE, CA 92714	
REVISIONS B2 17APR78	TITLE SCHEMATIC AM-100 CPU NO. 1
DRAWN / CHECKED / APPROVED	SHEET NO. 5 OF 10
	DRAWING NO. DWL-00101-XX


PART OF J2
READY 4
HIBYTE 3
IDIN 30
VINTR3 33
VINTR2 32
VINTR1 31
IACK (S 5)
DIN (SH 5)
BDAL15 (SH 8)
BDAL14 (SH 8)
BDAL13 (SH 8)
BDAL12 (SH 8)
BDAL11 (SH 8)
BDAL10 (SH 8)
BDAL09 (SH 8)
BDAL08 (SH 8)
HIBYTE (SH 8)
U10
U9
U1
8551
CLK CLR M N
D1 D2 D3 D4 G1 G2
Q1 Q2 Q3 Q4
DAL07 (SH 4,7,8)
DAL06 (SH 4,7,8)
DAL05 (SH 4,7,8)
DAL04 (SH 4,7,8)
DAL03 (SH 4,7,8)
DAL02 (SH 4,7,8)
DAL01 (SH 4,7,8)
DAL00 (SH 4,7,8)
PART OF J1
DI0 95
DI1 94
DI2 41
DI3 42
DI4 91
DI5 92
DI6 93
DI7 43
R20
R19
4.7K
+5V
U27
81LS97
DAL08 (SH 4,8)
DAL09 (SH 4,8)
DAL10 (SH 4,8)
DAL11 (SH 4,8)
DAL12 (SH 4,8)
DAL13 (SH 4,8)
DAL14 (SH 4,8)
DAL15 (SH 4,8)
19
DATA IN
ALPHA MICRO SYSTEMS
IRVINE, CA 92714
REVISIONS B2 17APR78
TITLE
SCHEMATIC
AM-100 CPU NO. 1
DRAWN
CHECKED
APPROVED
SHEET NO. 6 OF 10
DRAWING NO. DWL-00101-XX

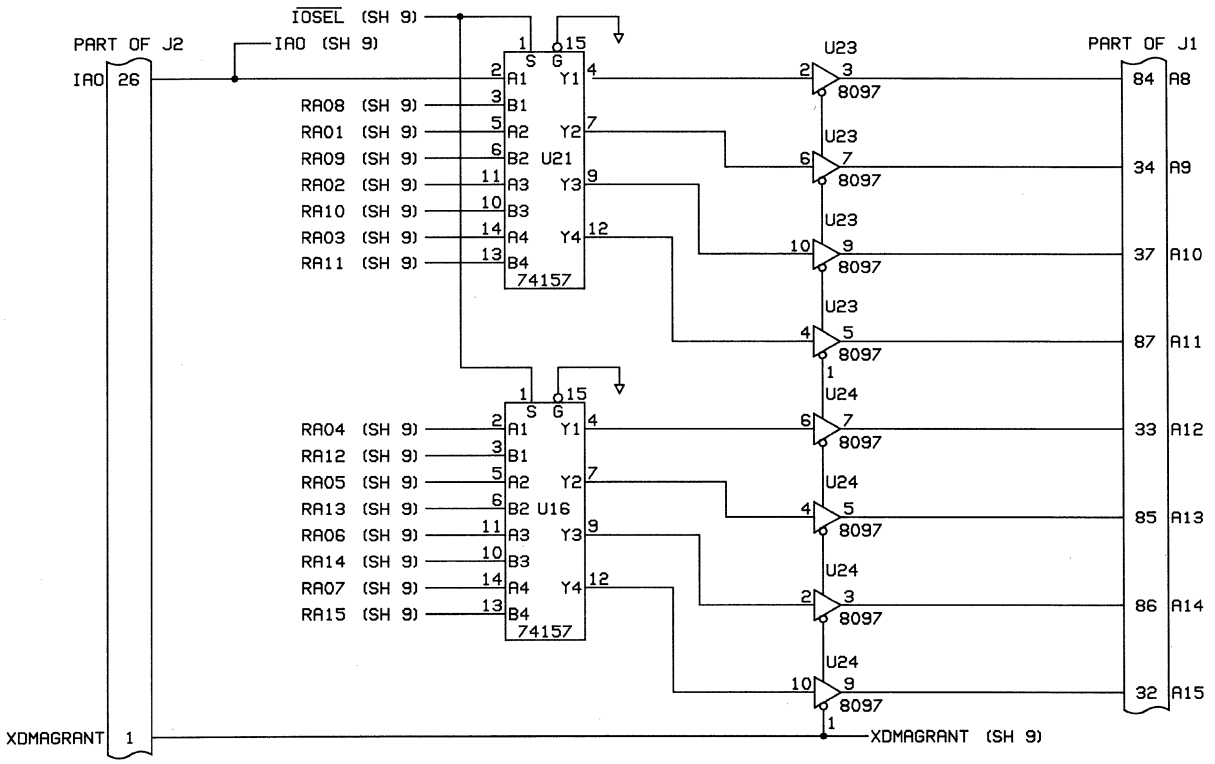

4
3
2
1
D
C
B
A
U3
PAD0
R7 4.7K
R10 4.7K
R9 4.7K
R11 4.7K
+5V
U5 8097
U6 81LS97
R4 4.7K
R6 4.7K
R3 4.7K
R2 4.7K
R8 4.7K
R6 4.7K
I2FF (SH 5)
DAL00 (SH 4,6,8)
DAL01 (SH 4,6,8)
DAL02 (SH 4,6,8)
DAL03 (SH 4,6,8)
DAL04 (SH 4,6,8)
DAL05 (SH 4,6,8)
DAL06 (SH 4,6,8)
DAL07 (SH 4,6,8)
PART OF J2
EARS 29
I2FF 11
ISR 28
EXT ADDRESS REGISTER SELECT
STATUS BYTE
ALPHA MICRO SYSTEMS
IRVINE, CA 92714
REVISIONS B2 17APR78
TITLE
SCHEMATIC
AM-100 CPU NO. 1
DRAWN
CHECKED
APPROVED
SHEET NO. 7 OF 10
DRAWING NO. DWL-00101-XX


4
3
2
1
D
C
B
A
DAL00 (SH 4,6,7)
DAL02 (SH 4,6,7)
DAL03 (SH 4,6,7)
DAL01 (SH 4,6,7)
DAL04 (SH 4,6,7)
DAL06 (SH 4,6,7)
DAL07 (SH 4,6,7)
DAL05 (SH 4,6,7)
DAL08 (SH 4,6)
DAL10 (SH 4,6)
DAL11 (SH 4,6)
DAL09 (SH 4,6)
DAL12 (SH 4,6)
DAL14 (SH 4,6)
DAL15 (SH 4,6)
DAL13 (SH 4,6)
U11
U19
81LS97
BDAL00 (SH 9)
BDAL02 (SH 9)
BDAL03 (SH 9)
BDAL01 (SH 9)
BDAL08 (SH 6,9)
BDAL10 (SH 6,9)
BDAL11 (SH 6,9)
BDAL09 (SH 6,9)
BDAL12 (SH 6,9)
BDAL14 (SH 6,9)
BDAL15 (SH 6,9)
BDAL13 (SH 6,9)
BDAL04 (SH 9)
BDAL06 (SH 9)
BDAL07 (SH 9)
BDAL05 (SH 9)
HIBYTE
(SH 6)
U17
U18
74157
U24
U26
8097
PART OF J1
D00
D02
D03
D01
D04
D06
D07
D05
PART OF J2
XDMAG 2
DATA OUT
ALPHA MICRO SYSTEMS
IRVINE, CA 92714
REVISIONS
B2 17APR78
TITLE
SCHEMATIC
AM-100 CPU NO. 1
DRAWN
CHECKED
APPROVED
SHEET NO.
8 OF 10
DRAWING NO.
DWL-00101-XX



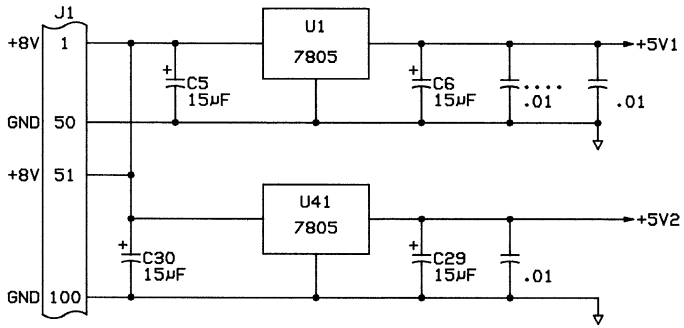
ALPHA MICRO SYSTEMS
IRVINE, CA 92714

REVISIONS	TITLE
B2 17APR78	SCHEMATIC AM-100 CPU NO. 1
DRAWN / CHECKED / APPROVED	SHEET NO. 9 OF 10
	DRAWING NO. DWL-00101-XX

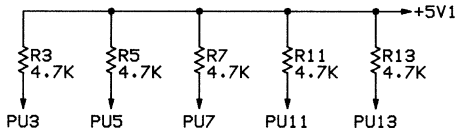


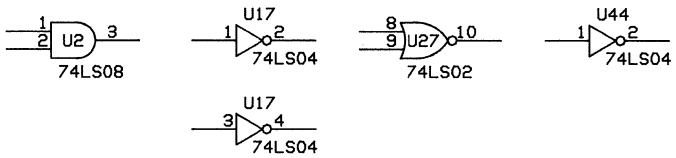
ADDRESS OUT

ALPHA MICRO SYSTEMS IRVINE, CA 92714	
REVISIONS B2 17APR78	TITLE SCHEMATIC AM-100 CPU NO. 1
DRAWN / CHECKED / APPROVED	SHEET NO. 10 OF 10
	DRAWING NO. DWL-00101-XX



POWER





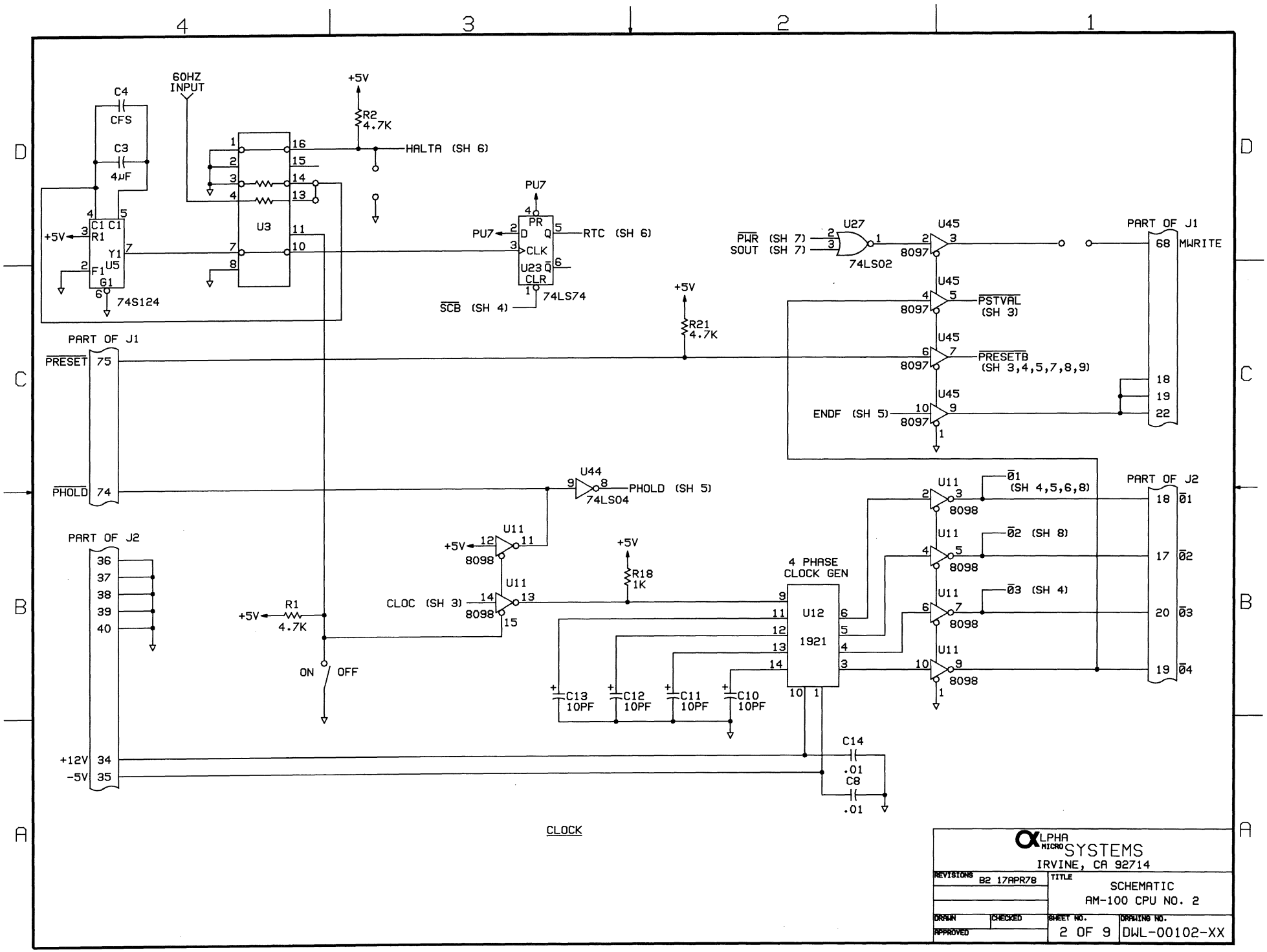
SPARE

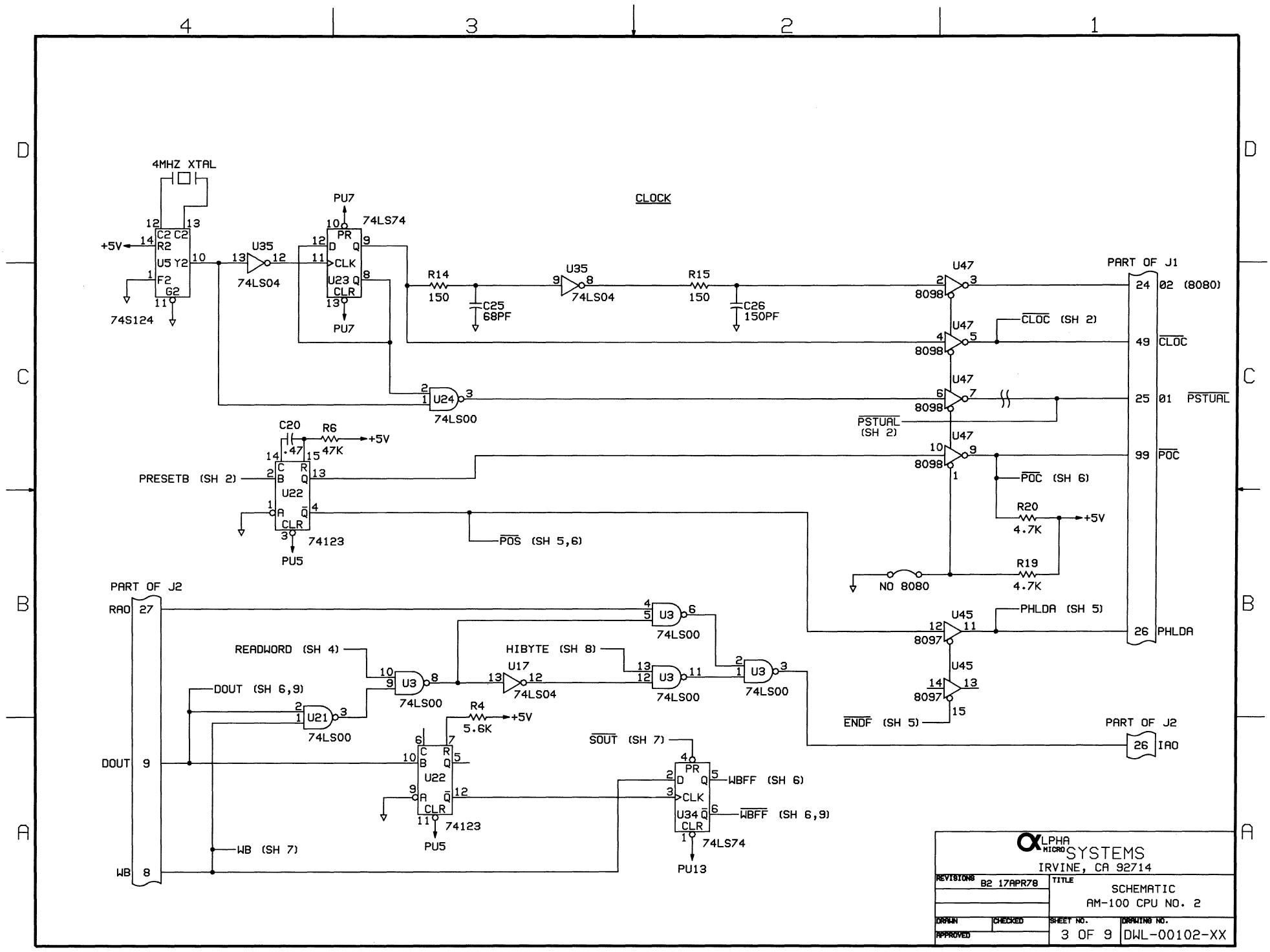
2. CAPACITORS ARE IN MICROFARADS.
1. RESISTORS ARE IN OHMS.

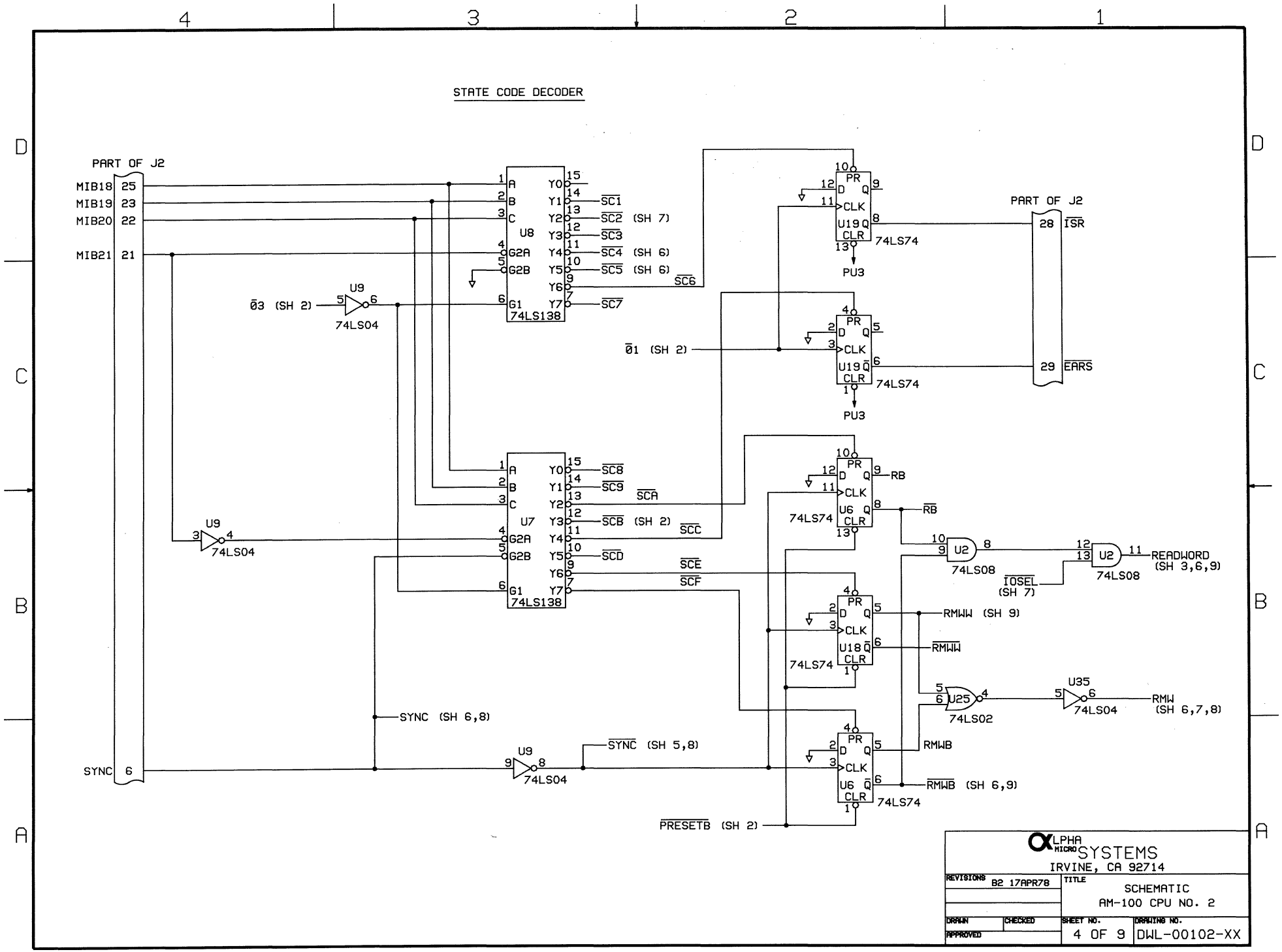
NOTES: UNLESS OTHERWISE SPECIFIED

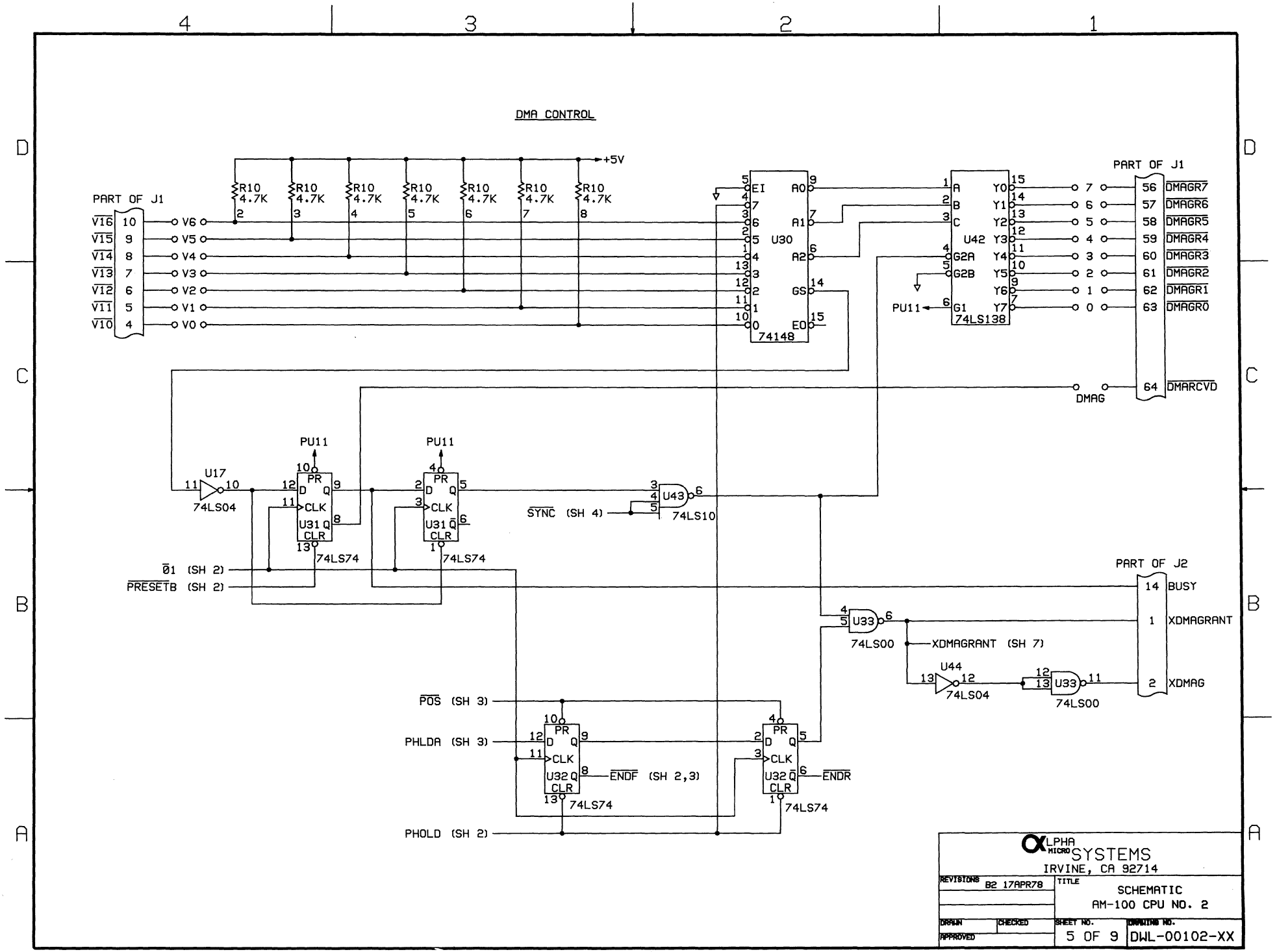
ALPHA MICRO SYSTEMS
IRVINE, CA 92714

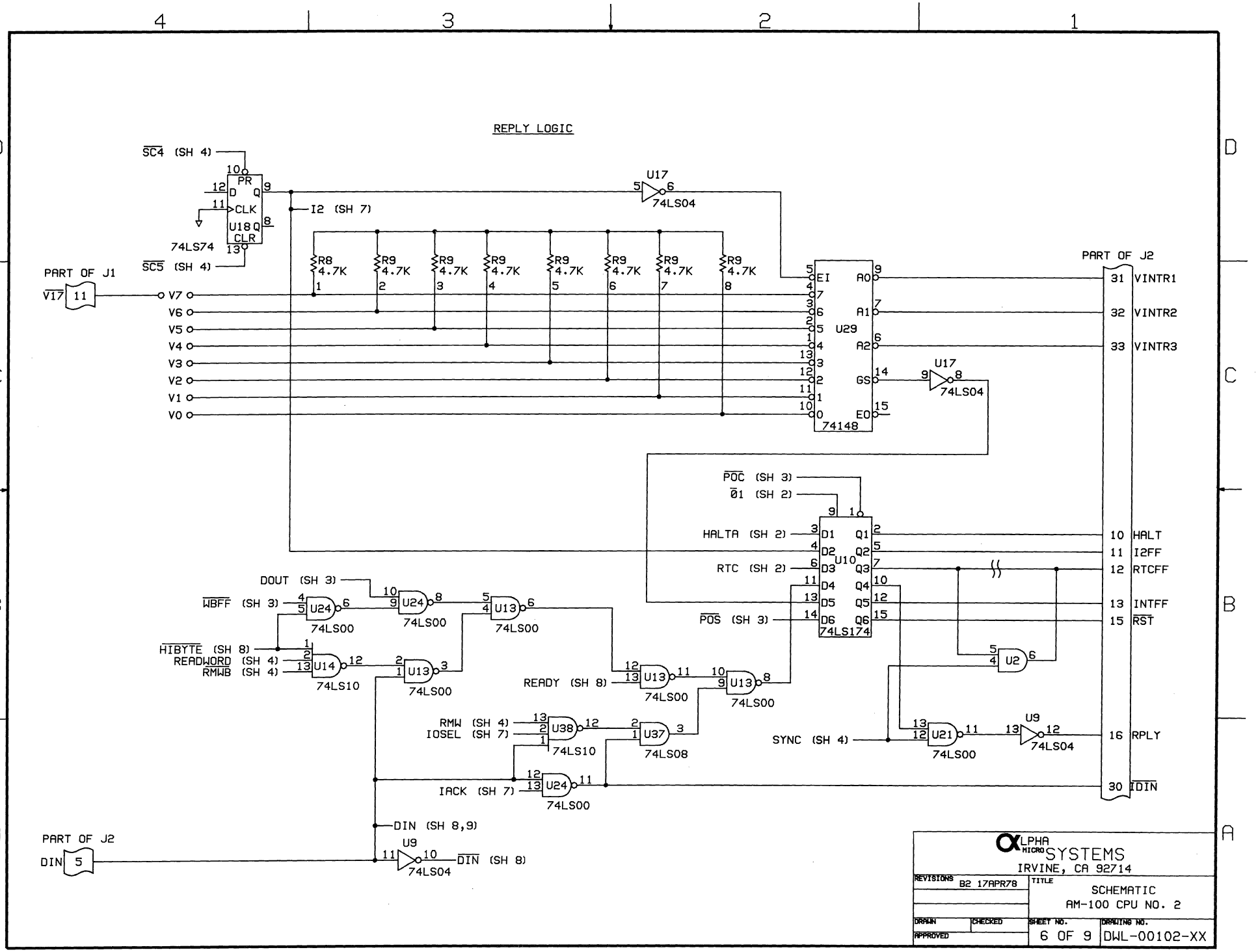
REVISIONS B2 17APR78	TITLE: SCHEMATIC AM-100 CPU NO. 2
DRAWN / CHECKED / APPROVED	SHEET NO. 1 OF 9
	DRAWING NO. DWL-00102-XX


4
3
2
1
D
C
B
A
60HZ INPUT
C4
CFS
C3
4µF
+5V
R2
4.7K
HALTA (SH 6)
U3
Y1
U5
74S124
PU7
PR
D
Q
CLK
CLR
U23
74LS74
SCB (SH 4)
RTC (SH 6)
PWR (SH 7)
SOUT (SH 7)
U27
74LS02
U45
8097
PSTVAL (SH 3)
PRESETB (SH 3,4,5,7,8,9)
ENDF (SH 5)
PART OF J1
68 MWRITE
18
19
22
R21
4.7K
PRESET 75
PHOLD 74
U44
74LS04
PHOLD (SH 5)
PART OF J2
36
37
38
39
40
U11
8098
CLOC (SH 3)
R1
4.7K
ON
OFF
R18
1K
4 PHASE CLOCK GEN
U12
1921
C13 10PF
C12 10PF
C11 10PF
C10 10PF
C14
.01
C8
.01
+12V 34
-5V 35
Ø1 (SH 4,5,6,8)
Ø2 (SH 8)
Ø3 (SH 4)
18 Ø1
17 Ø2
20 Ø3
19 Ø4
CLOCK
ALPHA MICRO SYSTEMS
IRVINE, CA 92714
REVISIONS
B2 17APR78
TITLE
SCHEMATIC
AM-100 CPU NO. 2
DRAWN
CHECKED
APPROVED
SHEET NO.
2 OF 9
DRAWING NO.
DWL-00102-XX


4
3
2
1
D
C
B
A
CLOCK
4MHZ XTAL
74S124
+5V
U35
74LS04
PU7
74LS74
PR
D
CLK
U23 Q
CLR
R14
150
C25
68PF
R15
150
C26
150PF
U24
74LS00
C20
.47
R6
47K
PRESETB (SH 2)
U22
74123
PU5
POS (SH 5,6)
U47
8098
CLOC (SH 2)
PSTUAL
(SH 2)
POC (SH 6)
R20
4.7K
R19
4.7K
NO 8080
PART OF J1
24 02 (8080)
49 CLOC
25 01 PSTUAL
99 POC
PHLDA (SH 5)
U45
8097
26 PHLDA
ENDF (SH 5)
PART OF J2
RA0 27
DOUT 9
WB 8
READWORD (SH 4)
HIBYTE (SH 8)
U17
U3
U21
DOUT (SH 6,9)
R4
5.6K
WB (SH 7)
SOUT (SH 7)
WBFF (SH 6)
WBFF (SH 6,9)
PU13
26 IA0
ALPHA MICRO SYSTEMS
IRVINE, CA 92714
REVISIONS B2 17APR78
TITLE
SCHEMATIC
AM-100 CPU NO. 2
DRAWN
CHECKED
APPROVED
SHEET NO.
3 OF 9
DRAWING NO.
DWL-00102-XX

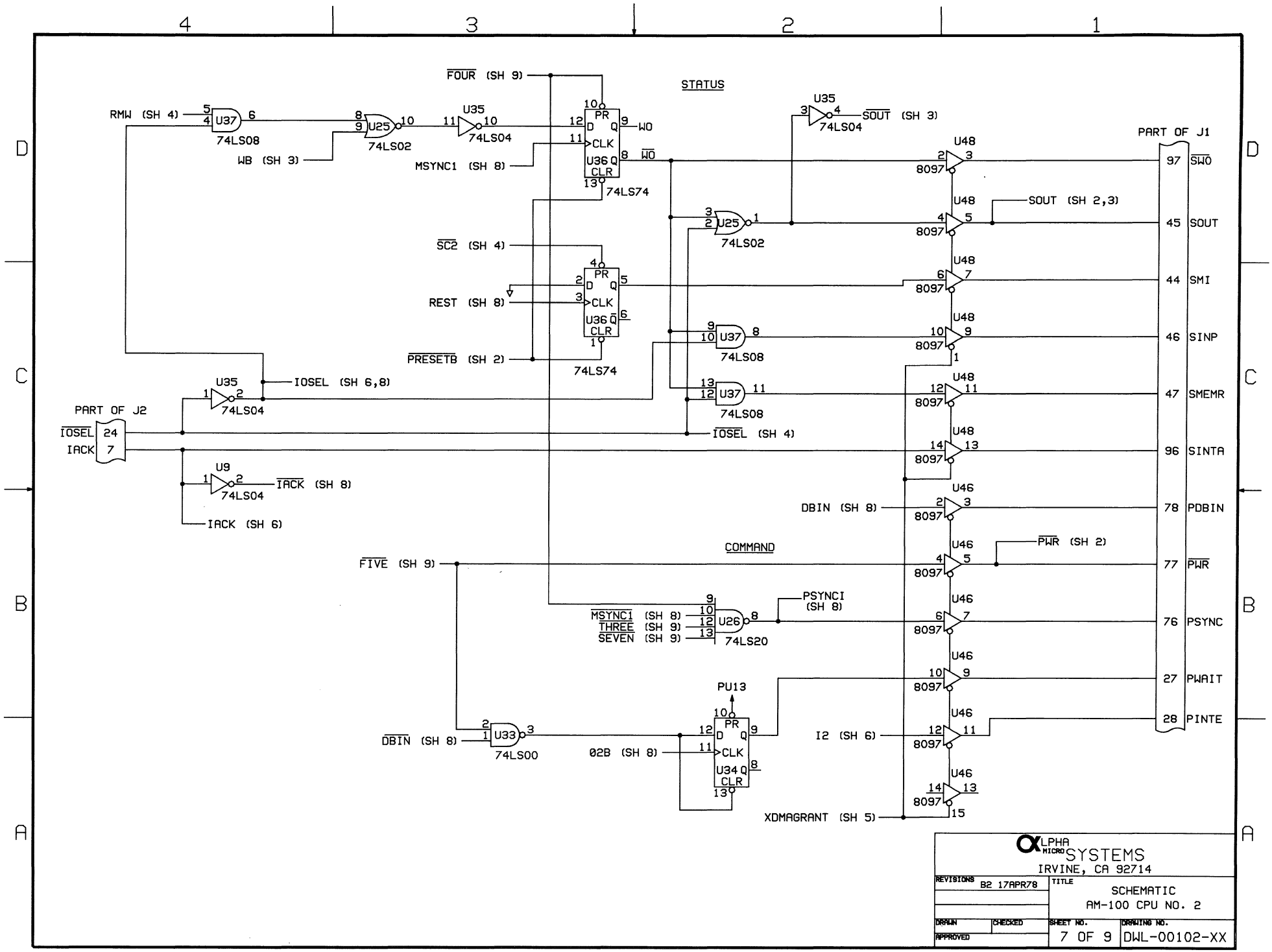

STATE CODE DECODER
PART OF J2
MIB18 25
MIB19 23
MIB20 22
MIB21 21
SYNC 6
U8
74LS138
U7
74LS138
U9
74LS04
Ø3 (SH 2)
Ø1 (SH 2)
SC1
SC2 (SH 7)
SC3
SC4 (SH 6)
SC5 (SH 6)
SC6
SC7
SC8
SC9
SCA
SCB (SH 2)
SCC
SCD
SCE
SCF
U19
74LS74
PU3
ISR
EARS
RB
U6
U2
74LS08
IOSEL (SH 7)
READWORD (SH 3,6,9)
RMWW (SH 9)
U18
U25
74LS02
U35
RMW (SH 6,7,8)
RMWB
RMWB (SH 6,9)
SYNC (SH 6,8)
SYNC (SH 5,8)
PRESETB (SH 2)
ALPHA MICRO SYSTEMS
IRVINE, CA 92714
REVISIONS B2 17APR78
TITLE
SCHEMATIC
AM-100 CPU NO. 2
DRAWN
CHECKED
APPROVED
SHEET NO. 4 OF 9
DRAWING NO. DWL-00102-XX

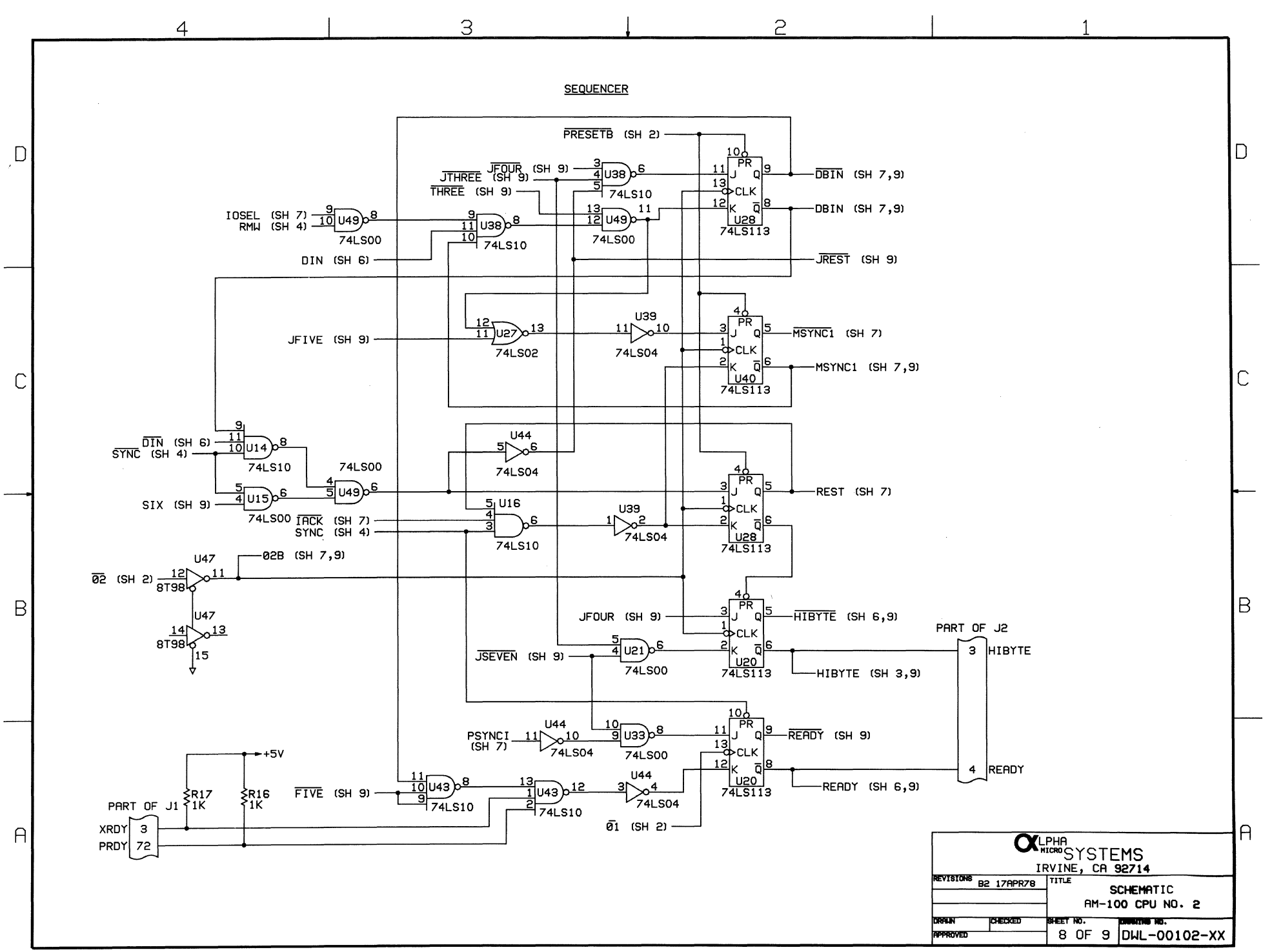

DMA CONTROL
PART OF J1
R10 4.7K
+5V
74148
74LS138
PART OF J1
DMAG
DMARCVD
PU11
U17
74LS04
74LS74
74LS10
SYNC (SH 4)
Ø1 (SH 2)
PRESETB (SH 2)
PART OF J2
BUSY
XDMAGRANT
XDMAG
74LS00
POS (SH 3)
PHLDA (SH 3)
PHOLD (SH 2)
ENDF (SH 2,3)
ENDR
ALPHA MICRO SYSTEMS
IRVINE, CA 92714
SCHEMATIC
AM-100 CPU NO. 2
B2 17APR78
5 OF 9
DWL-00102-XX

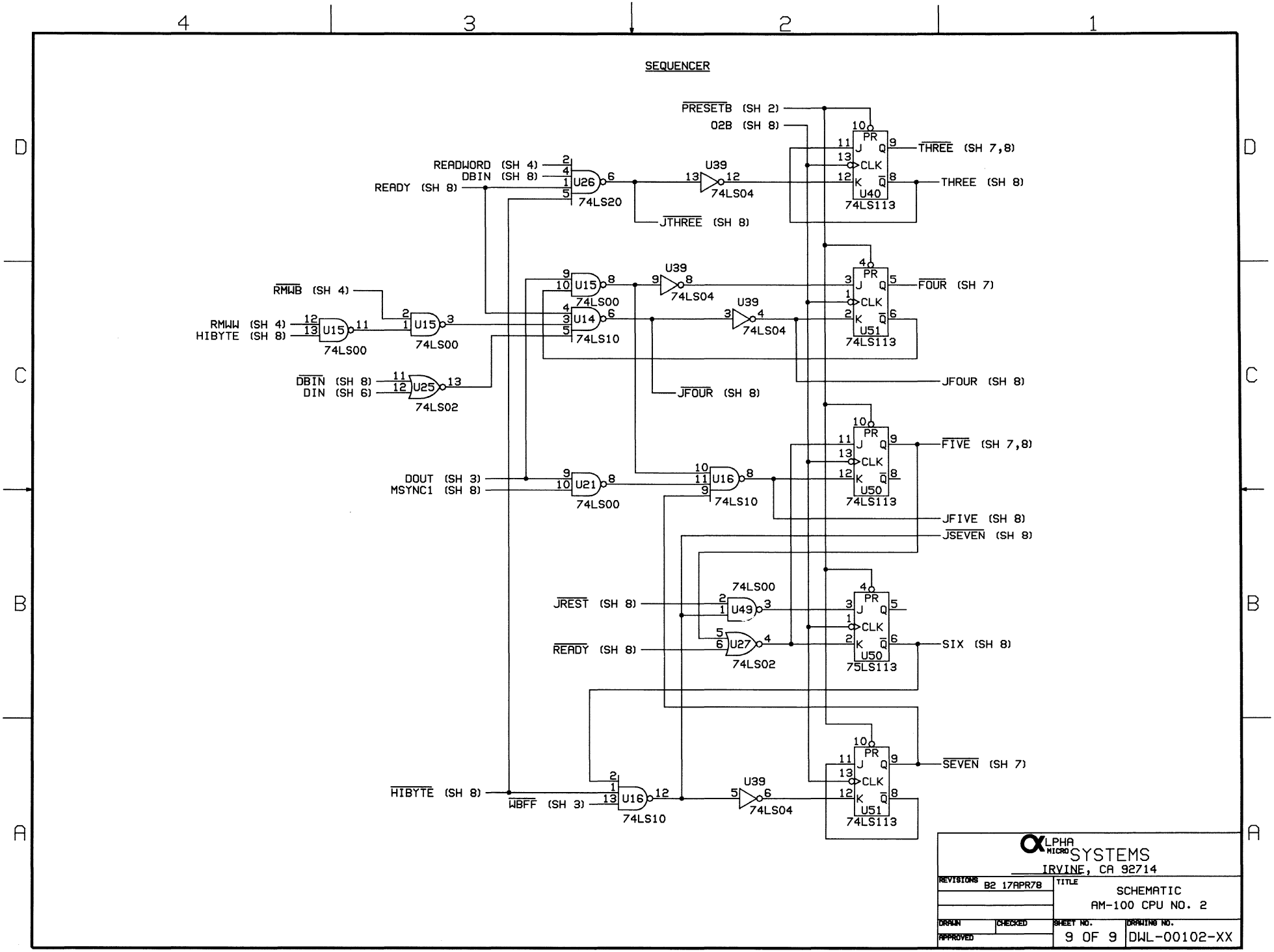


ALPHA MICRO SYSTEMS
IRVINE, CA 92714

REVISIONS	TITLE
B2 17APR78	SCHEMATIC AM-100 CPU NO. 2
DRAWN / CHECKED / APPROVED	SHEET NO. 6 OF 9
	DRAWING NO. DWL-00102-XX


4
3
2
1
D
C
B
A
STATUS
COMMAND
FOUR (SH 9)
RMW (SH 4)
WB (SH 3)
MSYNC1 (SH 8)
U37
74LS08
U25
74LS02
U35
74LS04
U36
74LS74
WO
WO
SC2 (SH 4)
REST (SH 8)
PRESETB (SH 2)
SOUT (SH 3)
SOUT (SH 2,3)
IOSEL (SH 6,8)
IOSEL (SH 4)
PART OF J2
IOSEL 24
IACK 7
U9
IACK (SH 8)
IACK (SH 6)
FIVE (SH 9)
MSYNC1 (SH 8)
THREE (SH 9)
SEVEN (SH 9)
U26
74LS20
PSYNCI (SH 8)
DBIN (SH 8)
U33
74LS00
02B (SH 8)
PU13
U34
I2 (SH 6)
XDMAGRANT (SH 5)
U48
U46
8097
PART OF J1
97 SWO
45 SOUT
44 SMI
46 SINP
47 SMEMR
96 SINTA
78 PDBIN
77 PWR
PWR (SH 2)
76 PSYNC
27 PWAIT
28 PINTE
ALPHA MICRO SYSTEMS
IRVINE, CA 92714
REVISIONS
B2 17APR78
TITLE
SCHEMATIC
AM-100 CPU NO. 2
DRAWN
CHECKED
APPROVED
SHEET NO.
7 OF 9
DRAWING NO.
DWL-00102-XX


SEQUENCER
PRESETB (SH 2)
JFOUR (SH 9)
JTHREE (SH 9)
THREE (SH 9)
IOSEL (SH 7)
RMW (SH 4)
DIN (SH 6)
JFIVE (SH 9)
DIN (SH 6)
SYNC (SH 4)
SIX (SH 9)
IACK (SH 7)
SYNC (SH 4)
02 (SH 2)
02B (SH 7,9)
JFOUR (SH 9)
JSEVEN (SH 9)
PSYNCI (SH 7)
FIVE (SH 9)
01 (SH 2)
DBIN (SH 7,9)
DBIN (SH 7,9)
JREST (SH 9)
MSYNC1 (SH 7)
MSYNC1 (SH 7,9)
REST (SH 7)
HIBYTE (SH 6,9)
HIBYTE (SH 3,9)
READY (SH 9)
READY (SH 6,9)
PART OF J2
HIBYTE
READY
PART OF J1
XRDY
PRDY
+5V
R17 1K
R16 1K
U49 74LS00
U38 74LS10
U28 74LS113
U27 74LS02
U39 74LS04
U40 74LS113
U14 74LS10
U15 74LS00
U44 74LS04
U16 74LS10
U47 8T98
U21 74LS00
U20 74LS113
U33 74LS00
U43 74LS10
U43 74LS10
U44 74LS04
ALPHA MICRO SYSTEMS
IRVINE, CA 92714
REVISIONS B2 17APR78
TITLE SCHEMATIC AM-100 CPU NO. 2
DRAWN CHECKED APPROVED
SHEET NO. 8 OF 9
DRAWING NO. DWL-00102-XX


SEQUENCER
PRESETB (SH 2)
02B (SH 8)
READWORD (SH 4)
DBIN (SH 8)
READY (SH 8)
U26
74LS20
U39
74LS04
JTHREE (SH 8)
U40
74LS113
THREE (SH 8)
THREE (SH 7,8)
RMWB (SH 4)
RMWW (SH 4)
HIBYTE (SH 8)
U15
74LS00
U14
74LS10
U51
FOUR (SH 7)
JFOUR (SH 8)
DBIN (SH 8)
DIN (SH 6)
U25
74LS02
DOUT (SH 3)
MSYNC1 (SH 8)
U21
U16
U50
FIVE (SH 7,8)
JFIVE (SH 8)
JSEVEN (SH 8)
JREST (SH 8)
READY (SH 8)
U49
U27
SIX (SH 8)
75LS113
HIBYTE (SH 8)
WBFF (SH 3)
SEVEN (SH 7)
ALPHA MICRO SYSTEMS
IRVINE, CA 92714
REVISIONS
B2 17APR78
TITLE
SCHEMATIC
AM-100 CPU NO. 2
DRAWN
CHECKED
APPROVED
SHEET NO.
9 OF 9
DRAWING NO.
DWL-00102-XX

FBILL # DWB-00101-00

REV.000

18 Dec,1979

Description ASSY CPU #1 AM-100

	PART NUMBER	DESCRIPTION		QTY
1	DWF-00101-00	PCB CPU #1	AM-100	1
2	CNF-00001-01	HEADER 40 PIN		1
3	CNS-00040-00	SOCKET 40 PIN DIP		5
4	CNS-00020-00	SOCKET 20 PIN DIP		4
5	CNS-00016-00	SOCKET 16 PIN DIP		15
6	CNS-00014-00	SOCKET 14 PIN DIP		2
7	CNS-00008-00	SOCKET 8 PIN DIP		2
8	HDM-00000-00	HEAT SINK 1.000WI .500HT .710LG		3
9	ICL-07805-00	IC REGULATOR + 5V		2
10	ICL-07812-00	IC REGULATOR +12V		1
11	CFP-00156-01	CAPACITOR 15 UF 20V		7
12	CPN-00103-01	CAPACITOR .01 UF		29
13	CPN-00330-01	CAPACITOR 33 PF		13
14		*** DELETED PART ***		0
15	RS2-00270-00	RESISTOR 27 OHM 1/4W 5% CAR		2
16	RS2-00150-00	RESISTOR 15 OHM 1/4W 5% CAR		1
17	RS2-00222-00	RESISTOR 2.2 K 1/4W 5% CAR		1
18	RS2-00102-00	RESISTOR 1 K 1/4W 5% CAR		2
19	RS2-00472-00	RESISTOR 4.7 K 1/4W 5% CAR		12
20	DIO-05231-00	DIODE ZENER 5V		1
21	IC1-08551-00	IC QUAD D FLIPFLOP		3
22	IC1-75361-00	IC TTL/MOS LEVEL CONVERTER		2
23	IC1-74367-00	IC HEX BUFFER		6
24	IC1-08197-01	IC BUFFER OCTAL		4
25	IC1-07404-01	IC HEX INVERTER		1
26	IC1-74174-01	IC HEX D FLIPFLOP		3
27	IC1-07430-01	IC 8 INPUT NAND GATE		1
28	IC1-74157-00	IC QUAD 2 TO 1 DATA SELECTOR		3
29	ICS-02151-00	IC DATA CHIP 3 MHZ		1
30	ICS-01661-03	IC CONTROL CHIP WD1600 .		1
31	CNH-00016-00	HEADER IC 16 PIN		3
32	RSN-00001-00	RES PACK SIP 4.7K		1
33	ICS-01631-27	IC MICROM #1 WD1600		1
34	ICS-01631-29	IC MICROM #2 WD1600		1
35	ICS-01631-30	IC MICROM #3 WD1600		1
36	HDS-00632-01	SCREW 6-32 X .250		3
37	HDN-00632-01	NUT HEX 6-32 STL SM PATTERN		3

	PART NUMBER	DESCRIPTION	QTY
33	MDW-00632-01	WASHER LOCK 6-32	3
39	DIC-04001-00	DIODE	1

BILL # DWB-00102-00

REV.C00

18 Dec,1979

Description ASSY CPU #2 AM-100

	PART NUMBER	DESCRIPTION	QTY
1	DWF-00102-00	PCB CPU #2 AM-100	1
2	CNF-00001-01	HEADER 40 PIN	1
3	CNS-00016-00	SOCKET 16 PIN DIP	15
4	CNS-00014-00	SOCKET 14 PIN DIP	34
5	HDM-00000-00	HEAT SINK 1.000WI .500HT .710LG	2
6	ICL-07805-00	IC REGULATOR + 5V	2
7	CRY-00001-00	CRYSTAL 4 MHZ	1
8	DWB-00103-03	ASSY HEADER OPTION AM-100	1
9	CPP-00156-01	CAPACITOR 15 UF 20V	4
10	CPN-00103-01	CAPACITOR .01 UF	18
11	CPN-00120-01	CAPACITOR CER 12 PF 100V 5%	4
12	CPN-00474-01	CAPACITOR .47 UF	1
13	CPN-00680-01	CAPACITOR 68 PF	1
14	CPN-00151-01	CAPACITOR 150 PF	1
15	RS2-00473-00	RESISTOR 47 K 1/4W 5% CAR	1
16	RS2-00562-00	RESISTOR 5.6 K 1/4W 5% CAR	1
17	RSN-00001-00	RES PACK SIP 4.7K	2
18	RS2-00151-00	RESISTOR 150 OHM 1/4W 5% CAR	2
19	RS2-00102-00	RESISTOR 1 K 1/4W 5% CAR	1
20	RS2-00472-00	RESISTOR 4.7 K 1/4W 5% CAR	12
21	IC1-07408-01	IC QUAD 2 INPUT AND GATE	2
22	IC1-07400-01	IC QUAD 2 INPUT NAND GATE	7
23	IC1-74124-01	IC DUAL OSCILLATOR	1
24	IC1-07474-02	IC DUAL D FLIPFLOP	2
25	IC1-74138-02	IC DECODER 3 TO 8 LINE	2
26	IC1-07404-01	IC HEX INVERTER	5
27	IC1-74174-01	IC HEX D FLIPFLOP	1
28	IC1-74368-00	IC HEX INVERTING BUFFER	2
29	ICS-01921-00	IC GENERATOR 4 PHASE CLOCK	1
30	IC1-07410-01	IC TRIPLE 3 INPUT NAND GATE	4
31	IC1-74113-01	IC DUAL J-K FLIPFLOP	5
32	IC1-26123-00	IC DUAL ONE SHOT	1
33	IC1-07402-01	IC QUAD 2 INPUT NOR GATE	2
34	IC1-07420-01	IC DUAL 4 INPUT NAND GATE	1
35	IC1-74148-00	IC ENCODER PRIORITY	2
36	IC1-74367-00	IC HEX BUFFER	3
37	HDS-00632-01	SCREW 6-32 X .250	2

BILL # DWB-00102-00 REV.C00 DESCRIPTION ASSY CPU #2 AM-1

	PART NUMBER	DESCRIPTION	QTY
38	HDN-00632-01	NUT HEX 6-32 STL SM PATTERN	2
39	HDW-00632-01	WASHER LOCK 6-32	2

Alpha Microsystems
17881 Sky Park North
Irvine, California 92714