MOSTEK®

Z80 MICROCOMPUTER SYSTEMS

Operations Manual

AIM-Z80AE APPLICATION INTERFACE MODULE

AIM-Z8ØAE

OPERATION MANUAL

Publication Number 7965Ø

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SECTION 1.Ø
GENERAL DESCRIPTION

1.1 INTRODUCTION

AIM-Z8ØAE is an advanced development tool which provides debug assistance for both software and hardware via in-circuit emulation of the Z8Ø microprocessor. Use of the AIM-Z8ØAE is completely transparent to the Target system configuration. No memory space or ports are used and all signals including $\overline{\text{RESET}}$, $\overline{\text{INT}}$, $\overline{\text{NMI}}$, $\overline{\text{BUSRQ}}$ and $\overline{\text{WAIT}}$ are functional during emulation.

Single step circuitry allows user to execute Target instructions one at a time to see the exact effect of each instruction. Single step is functional in ROM as well as RAM.

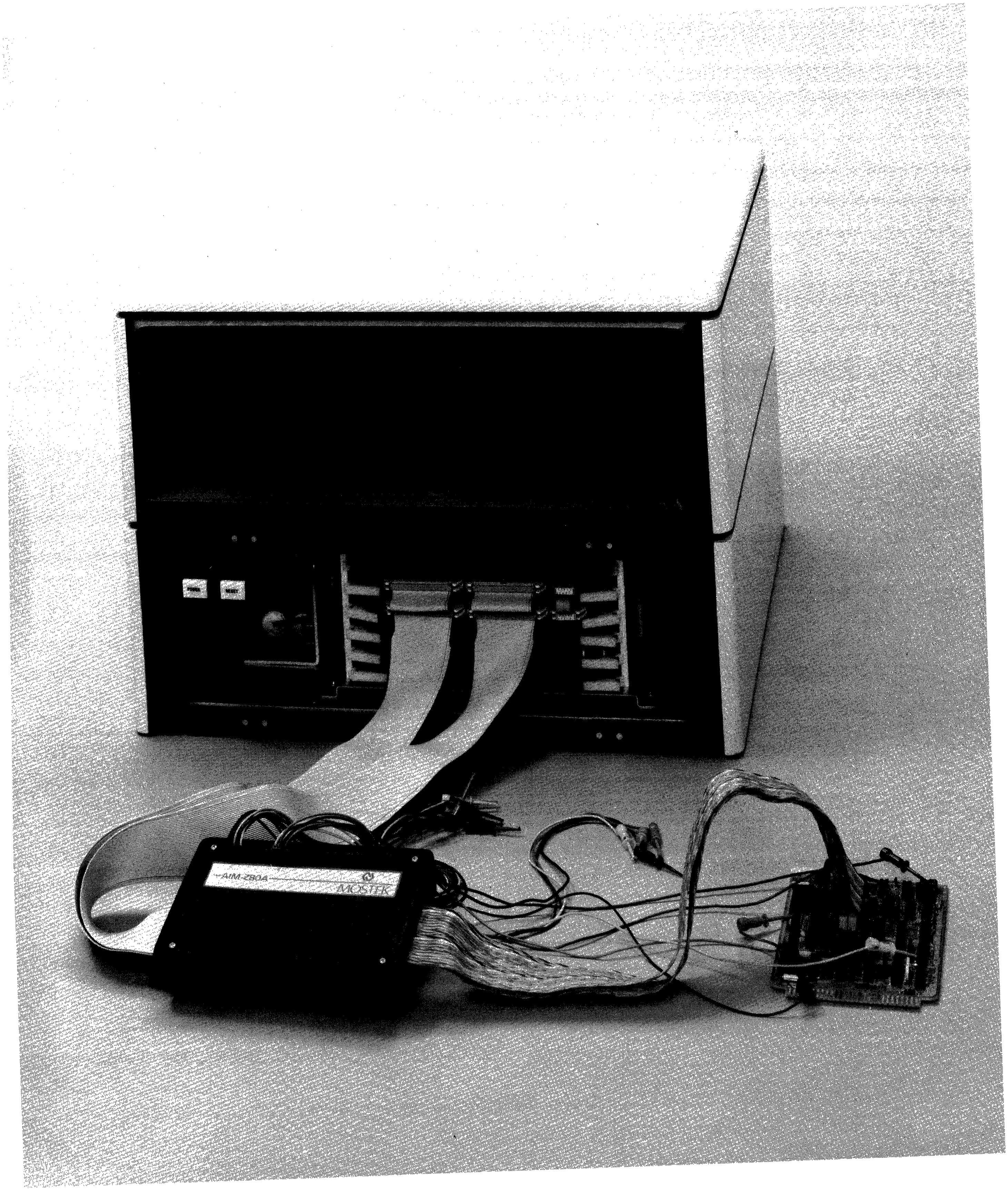
Sixteen K bytes of emulation RAM may be mapped into the Target memory space at any desired address so that software may be developed even before Target memory is available.

Breakpoint detect circuitry allows real time execution to proceed to any desired point in the users program and then terminate with all registers and status information saved so that execution may later be resumed. Real time execution may also be terminated at any time with the Escape key. EVENT and DELAY counters give added flexibility for viewing the exact point of interest in the users program.

The 48 channel history module will simultaneously record any bus transaction which the user may desire to see. Address bus, Data bus and Control signals plus eighteen external probes which can be used to monitor the Target circuitry at other points are sampled by the history RAM.

AIM-Z8ØAE is partitioned into three modules. The Control and History modules are .installed directly into the MOSTEK disk-based development system. Cables from these modules connect to the Buffer module which plugs directly into the Target system Z8Ø CPU socket. After AIM-Z8ØAE is installed, the development system is powered up and the system booted up as normal. All development system software and hardware is still functional. AIM-Z8ØAE system software (AIMZ8Ø) may be initialized by using the implied run command. AIMZ8Ø will sign on, take control of the Target system and allow the user to initialize the Target system and use any of the AIM-Z8ØAE commands to load, test and debug his Target program.

FIGURE 1.1 AIM-Z80AE SYSTEM



1.2 REFERENCES

MATRIX Operation manual, MK79730

FLP-80DOS Operation manual, MK78557

Z80 DATA BOOK, MK79602

Z80 Micro Reference manual, MK78516

Z80 Programming manual, MK78515

1.3 CONVENTIONS

The following conventions apply throughout this manual.

1. Hexadecimal input to the system does not require a leading numeric digit and cannot have a subscript H.

2. (CR) represents Carriage Return.

3. (LF) represents Line Feed.

4. (UP) represents Upcaret (ASCII code 5EH).

5. (SP) represents Space bar.

6. (.) represents period.

7. Bracketed items [] in a command line are optional.

8. Items in a command line which must be entered exactly as they appear are shown as upper case.

9. Items in a command line which are variables are shown as lower case.

10. Characters in a command line which are entered by the user are underlined.

11. Signal names are upper case. i.e. the signal EXECUTING.

12. (ESC) represents the ESCAPE key (ASCII code 1BH).

13. Characters referenced in Text are enclosed with double quotes. (i.e. when the character "Y" is entered).

1.4 TERMINOLOGY

1. Target. The Target is the users system under development. When using AIM-Z8ØAE, the Target CPU is contained in the Buffer module, but the remainder of the Target system is the users hardware. The Target program is the users program under development.

2. System. The System refers to the MOSTEK disk-based development system including the AIM-Z8ØAE hardware. The System software includes the MOSTEK disk-based development systems software as well as the AIM-Z8ØAE software.

3. Emulation RAM. The Emulation RAM is the 16k byte (expandable to 64k byte) dynamic RAM on the Control module. The Emulation Ram can be mapped to any address in the Target memory map and is normally used to emulate PROM or ROM which will eventually contain the Target program.

4. Interface RAM. The Interface RAM is a 1k byte static RAM on the Control module which is used for interface between the system and the Target. This RAM can be accessed only by the system and is loaded with the interface control program. The Interface RAM does not appear in the Target memory map when the Target program is executed or examined, but is shadowed in during debug.

5. Sampling. This refers to the operation of the History module. When the History module is storing bus cycles into the History RAM, it is sampling.

1.5 BUFFER MODULE

The Buffer module is located near the Target system and contains the Target Z80 microprocessor, Target adaptor connector and interface buffers. Cables connect the Buffer module to the Control and History modules which are installed in the development system. The Target adapter connector is attached to the Buffer module with eighteen inches of flat cable and plugs directly into the Target Z80 CPU socket. There are five test sockets available on the edge of the of the Buffer module with the following signals.

TEST POINT	SIGNAL
BLUE	$\overline{\text{FETCH}}$ ($\overline{\text{M1}}$ for first opcode of instruction)
GREEN	PHI (Z80 clock)
YELLOW	SRAM (Active when Emulation RAM is accessed)
RED	+5 VOLTS
BLACK	GND

The signal SRAM is an active high signal which becomes active when the Emulation RAM is accessed. This signal may be used to reverse the direction of buffers in the Target system so that RETI (Return from Interupt) instructions executed from the Emulation RAM can be recognized by the Target system.

There are also two sets of nine probes designated the A and B which plug into the upper left corner of the Buffer module. The probes are color coded with standard EIA color codes as follows:

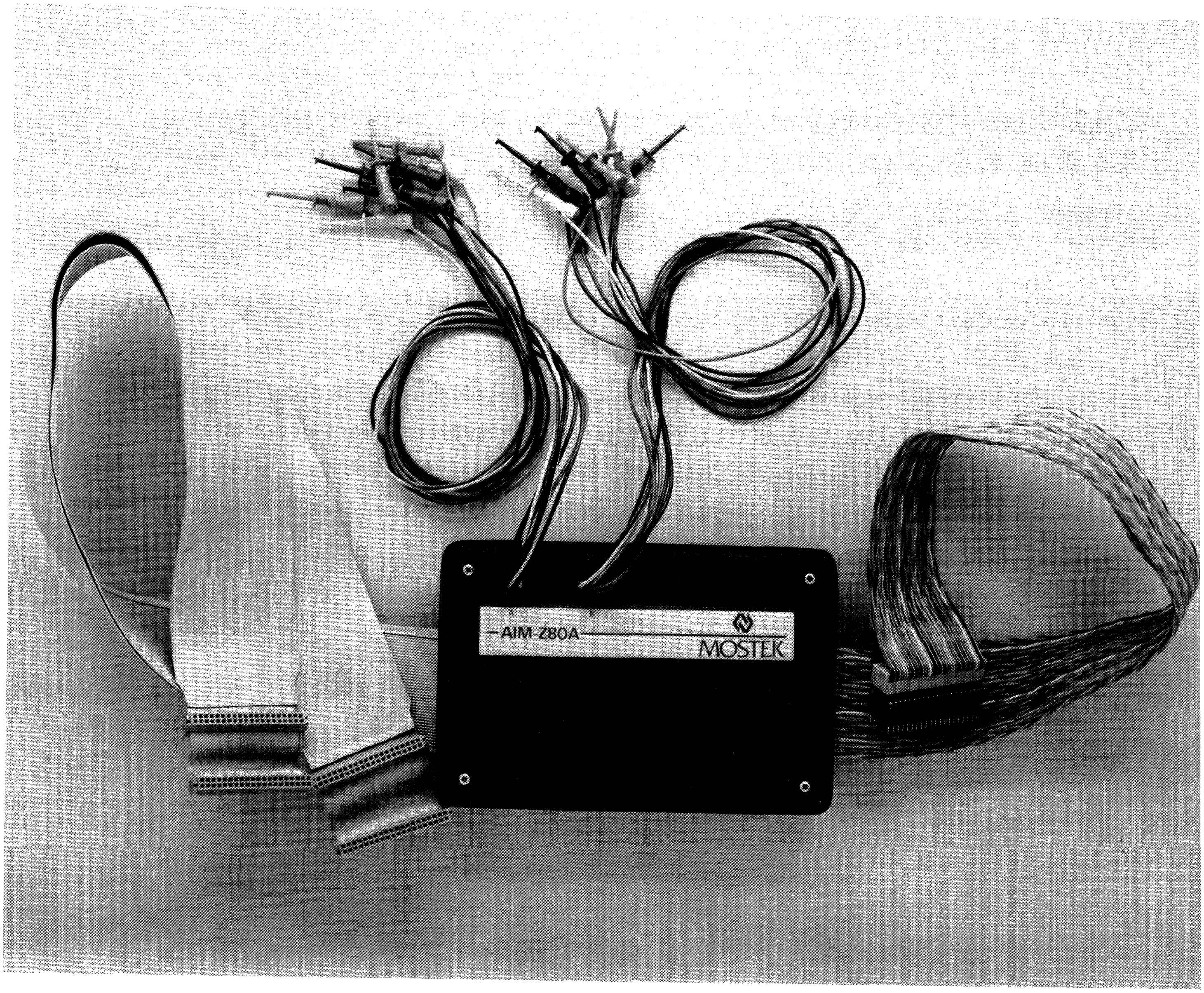
0	BLACK
1	BROWN

2	RED
3	ORANGE
4	YELLOW
5	GREEN
6	BLUE
7	VIOLET
8	GREY

When a probe is referenced in this manual or in a print out, P(for Probe) will be followed by either A or B and then the probe number. PB6 for example refers to the blue probe of the B set.

When the AIM-Z8ØAE is initialized, the control signals $\overline{\text{M1}}$, $\overline{\text{MREQ}}$, $\overline{\text{RD}}$ and $\overline{\text{WR}}$ are disabled to the Target system. The signals are disabled without introducing any glitches which could interfere with proper dynamic RAM operation. These control signals are not disabled during execution from the Emulation RAM so Z8Ø peripheral devices can recognize the RETI instruction when it is fetched from Emulation RAM. This requires that the user disable any memory in the Target system from responding on the CPU data bus when Emulation RAM is accessed.

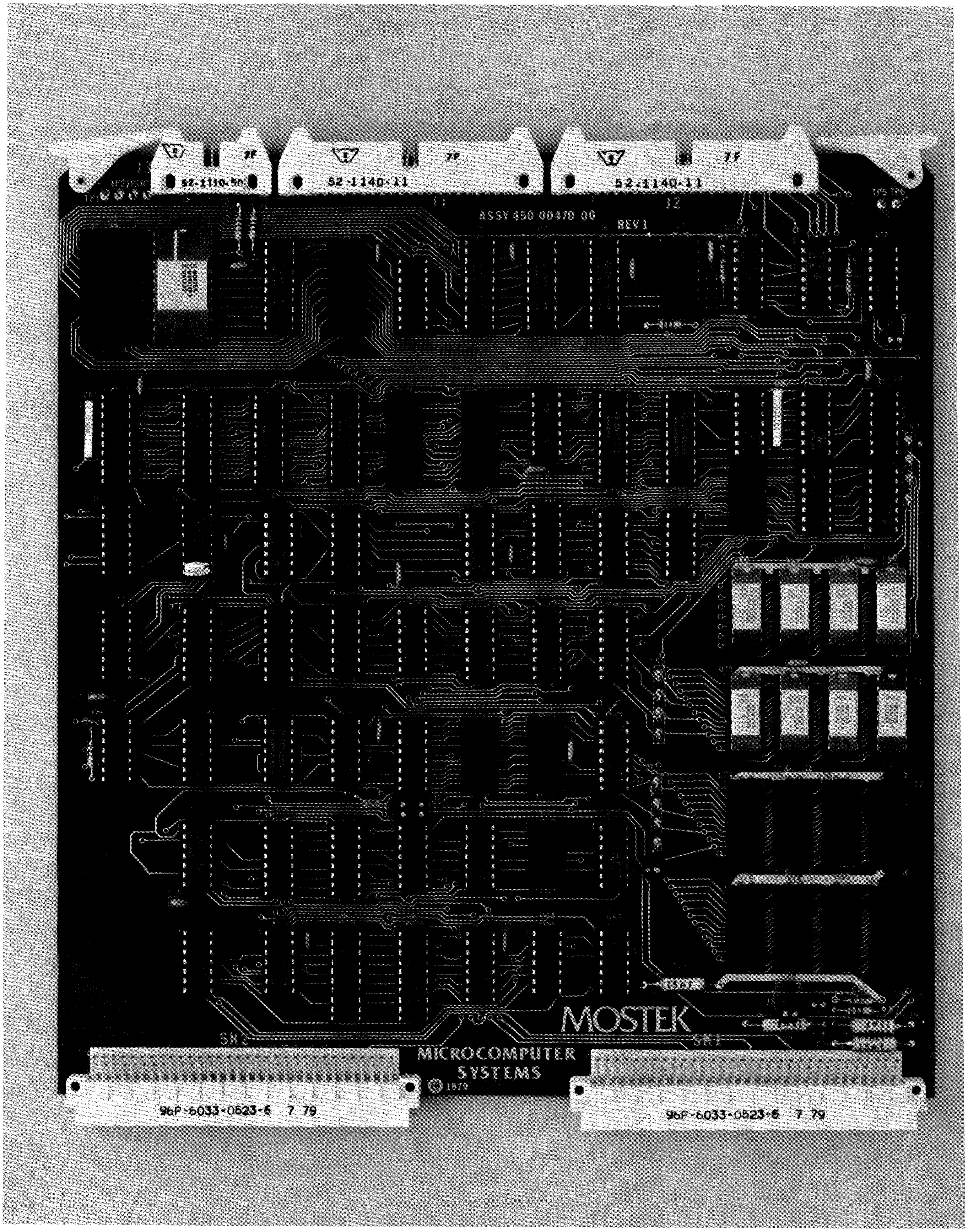
FIGURE 1.2 BUFFER MODULE



1.6 CONTROL MODULE

The Control module interfaces to the development system bus and has circuitry for detecting the breakpoint conditions and forcing execution to begin in the System Interface RAM. Connectors J1 and J2 bring the Address, Data and Control signals from the Target Z8Ø CPU in the Buffer module. Connector J3 brings the hardware breakpoint signal from the History module. The System Interface RAM which is loaded with an interface program is shadowed into the Target memory space. This control program makes the Target CPU a slave to the development system. When the user desires to resume execution, the control program activates the execution control circuit and execution resumes at the desired address. The memory control circuit is used to map the sixteen k byte emulation RAM to appear at any address in the Target memory space.

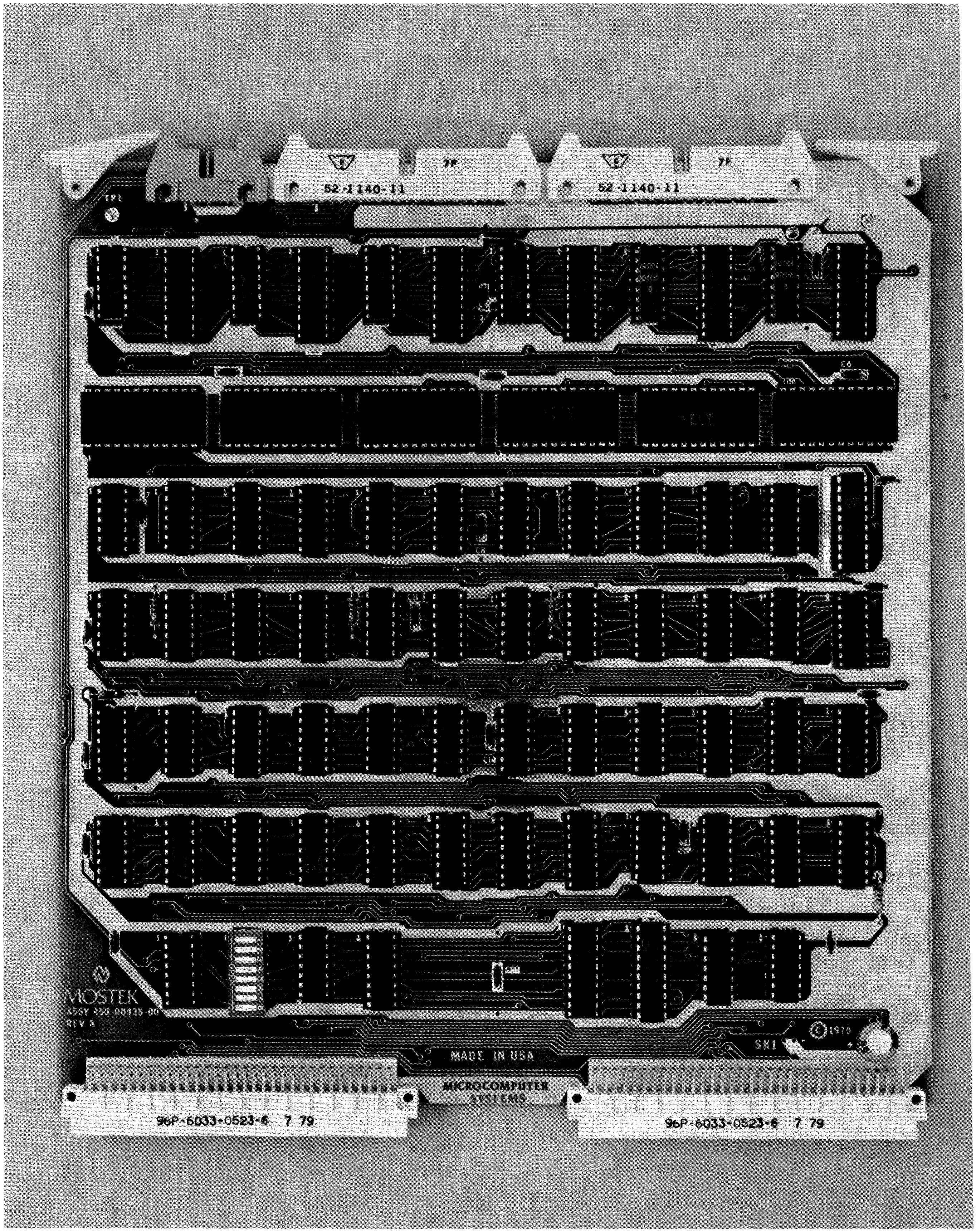
FIGURE 1.3 CONTROL MODULE



1.7 HISTORY MODULE

The History module also interfaces to the development system bus and Connectors J1 and J2 bring the Address, Data and control signals from the Buffer module. Connector J3 brings the signal EXECUTING from the control module which is used to enable the History RAM only when Target instructions are executed. The History module has a 24 bit comparator circuit to detect the hardware breakpoint condition, and EVENT counter and DELAY counter. Sampling into the 48 by 1k history RAM is enabled and disabled according to user selected conditions by the History control circuit. The execution timer is used to count Target processor clocks for logging elapsed execution time or generating timer breakpoints.

FIGURE 1.4 HISTORY MODULE



SECTION 2.Ø
AIM-Z8ØAE INSTALLATION

2.1 INTRODUCTON

This section gives the background information and a step by step procedure for installing AIM-Z8ØAE in the MOSTEK disk-based development system and checking the basic operation.

2.2 COMPATIBLE SYSTEMS

AIM-Z8ØAE is compatible with the Mostek MOSTEK disk-based development system with thirty-two k bytes of RAM or more.

2.3 EQUIPMENT REQUIRED

The following equipment is required for operation of AIM-Z8ØAE:

MOSTEK disk-based development system with console terminal.
Target system with Z8Ø CPU in socket, power and clock circuit.
AIM-Z8ØAE Buffer module with cable assemblies
AIM-Z8ØAE Control module
AIM-Z8ØAE History module
Connector assembly, 1Ø pin flat
MOSTEK system diskette with AIMZ8Ø.BIN

2.4 JUMPER OPTIONS, CONTROL MODULE

This section describes the installation of jumpers for user selectable options on the Control module. Jumpers which must be installed are specified with a "-". Jumpers which must not be installed are specified with a " ". Jumpers which are described in a different paragraph are specified with a ".".

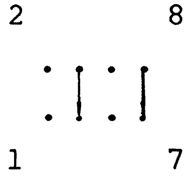
1. J4 is not used and no jumpers should be installed.

2. J5 is configured according to the type of emulation RAM used and should be jumpered as follows:

```
MK4116/MK4164 (16k/64k)        MK4332 (32k)

        1 . . 2                1 .—. 2
        3 .—. 4                3 . . 4
```

3. J6 selects the block of system ports used by the Control module. Ports 8Ø-87 are used so the following jumpers should be installed:

BOARD Ø
8ØH-87H



4. J7 is configured according to the type of emulation RAM used and should be jumpered as follows:

```
MK4116/MK4332 (16k/32k)          MK4164 (64k)
     1 . . 2                         1 •—• 2
     3 •—• 4                         3 . . 4
```

5. J8 is configured according to the type of emulation RAM used and should be jumpered as follows:

```
MK4116/MK4332 (16k/32k)          MK4164 (64k)

     2 . ┬ 4                         2 ┬ . 4
     1 . ┴ 3                         1 ┴ . 3
```

6. J9 is configured according to the type of emulation RAM used and should be installed only for MK4116/MK4332 devices.

```
MK4116/MK4332 (16k/32k)          MK4164 (64k)

  J9 1•—•2                         J9 1. .2
```

2.5 JUMPER OPTIONS, HISTORY MODULE

This section describes the installation of the jumpers which specify user selectable options. Jumpers which must be installed are specified with a "-". Jumpers which must not be installed are specified with a " ". Jumpers which are described in other paragraphs are specified with a ".". U68 selects the board which generates the execution timer signal, the block of system ports used by each board in multiboard systems, and if the OEM-8Ø CTC is used to extend the EVENT counter to sixteen bits.

1. Pins 1 and 16 of U68 select the board which generates the execution timer signal. If this jumper is not installed, the timer will not work, and XX USEC will be printed instead of the timer value.

2. Pins 3, 4, 5, 6, 11, 12, 13, and 14 of U68 select the block of system ports used by the History module. Ports 4ØH-47H are used so the following jumpers should be installed:

```
BOARD Ø
4ØH-47H

1 ... 16
2 ... 15
3 ... 14
4 ——— 13
5 ——— 12
6 ... 11
7 ... 1Ø
8 ...  9
```

3. Pins 2, 7, 8, 9, 1Ø, and 15 allow the event counter to be extended to 16 bits by using channel one of the OEM-8Ø CTC for the upper eight bits. Only one AIM-Z8ØAE can have the event counter extended to sixteen bits.

EIGHT BIT EVENT COUNTER	SIXTEEN BIT EVENT COUNTER
1 ... 16	1 ... 16
2 ... 15	2 •—• 15
3 ... 14	3 ... 14
4 ... 13	4 ... 13
5 ... 12	5 ... 12
6 ... 11	6 ... 11
7 •—• 1Ø	7 ... 1Ø
8 ... 9	8 •—• 9

2.6 TARGET SYSTEM CONFIGURATION

AIM-Z80AE is designed to be as independent of the Target configuration as practical; however, there are some restrictions on the Target system.

1. The Target PHI clock must be in the range of 500kHz to 4 MHz.

2. The Target system must be able to supply the extra specified 5 volt power to the buffer box, and the ground bus to the CPU socket must be substantial.

3. The signals $\overline{\text{RESET}}$, $\overline{\text{BUSRQ}}$, $\overline{\text{WAIT}}$ cannot be in the active state during initialization.

4. The Target Data bus must be tristate except when $\overline{\text{MREQ}}$ OR $\overline{\text{IORQ}}$ are active.

5. Most output signals from the Buffer module will only drive 1.4 mA and still meet the 0.4 volt output low voltage specification. These signals will still drive 1.8 mA at an output low voltage of 0.5 volt.

2.7 INSTALLATION

The following steps should be followed when installing AIM-Z8ØAE in an MOSTEK disk-based development system. Begin with all power OFF and the diskettes NOT inserted. When installing AIM-Z8ØA in MATRIX or SYS-8ØF systems it is necessary to make the following interconections between SK2 of the OEM-8Ø, Control and history modules. The signals should be wire wrap or plug wire installed on the backplane connectors. On Matrix system, TP1 may be used for connections to the OEM-8Ø. If these connections are not performed the T state timer and extended EVENT counter will not function. Figure 2.1 illustrates the interconnections.

SIGNAL	FROM	TO
CK/TG1B (BIT 7 of EVENT)	HISTORY SK2-4a	OEM-8ØE SK2-4a
ZC/TO1B (BIT 15 of EVENT)	OEM-8ØE SK2-4C	HISTORY SK2-4C
CK/TG2B (BIT 7 of TIMER)	HISTORY SK2-5a	OEM-8ØE SK2-5a
ZC/TO2B (BIT 15 of TIMER)	OEM-8ØE SK2-5C	CONTROL SK2-4C

1. Verify jumper options on the Control module according to section 2.4 of this manual.

2. Verify jumper options on the History module according to section 2.5.

3. Install Control module in the first available slot of the MOSTEK disk-based development system.

4. Install History module in the second available slot of the MOSTEK disk-based development system.

5. Verify that the reset select switch (S2) on the OEM-8Ø is in the "EØØØ" position (bat handle towards the LED).

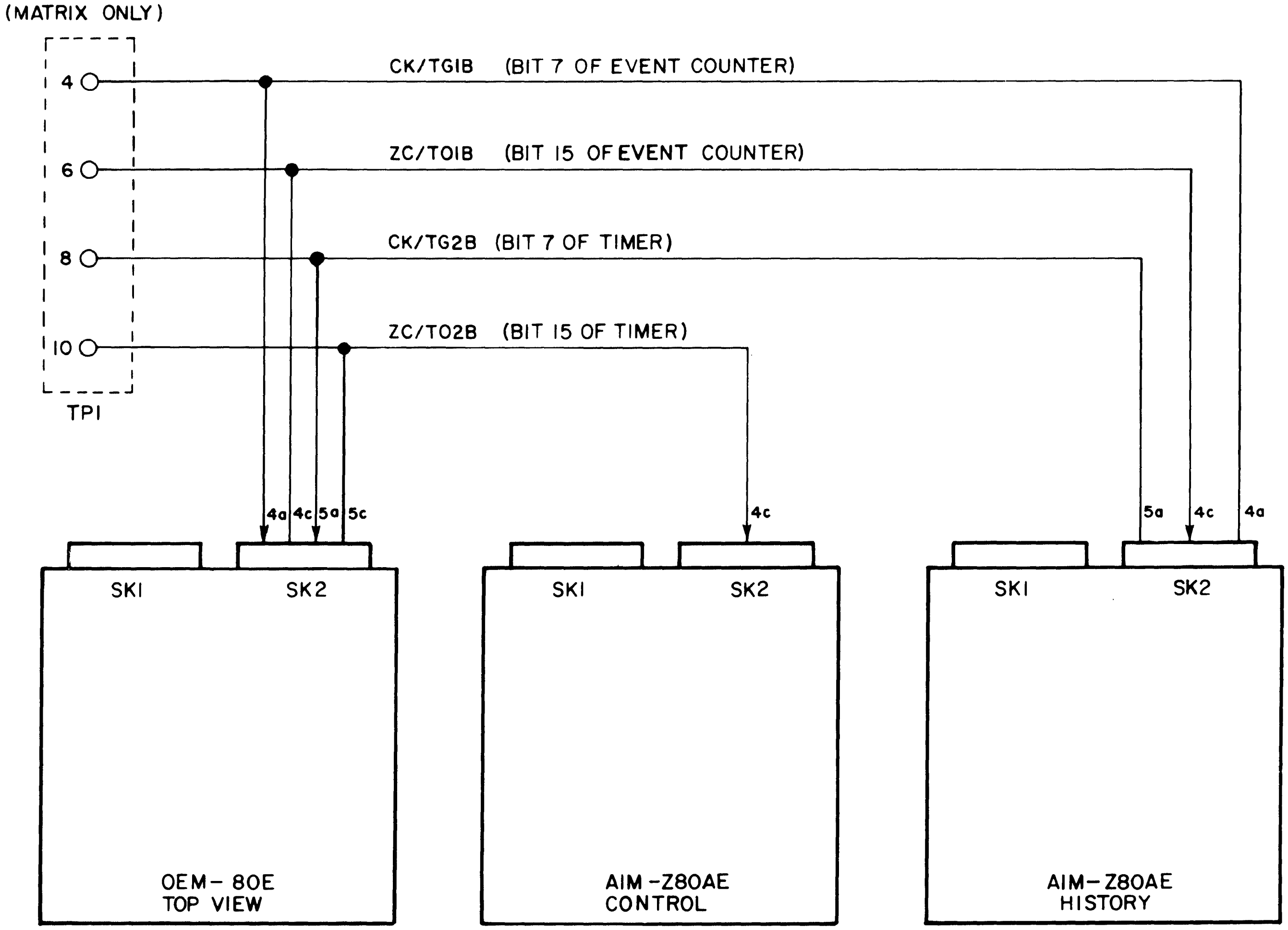
11. Recheck all cabling and card orientations and turn the MOSTEK disk-based development systems power on and then the Target system.

12. Insert the AIM-Z80AE diskette into DK0 and close the disk drive and boot up the system as normal.

13. Perform the Initial Checkout Procedure in the next section.

FIGURE 2.1

INSTALLATION IN MATRIX OR SYS-8ØF



2.8 INITIAL CHECKOUT PROCEDURE

The following checkout procedure may be used to verify the basic operationof AIM-Z8ØAE. To perform this checkout procedure, a suitable Target system is required. The minimal Target system of FIGURE 2.1 or Mostek MDX-CPU1 should be used. Whatever Target system is used, Target memory should not be enabled for address space Ø-ØFFH during the initial checkout procedure.

1. Perform the installation procedure in section 2.7.

2. Enter the following underlined text and verify the printout.

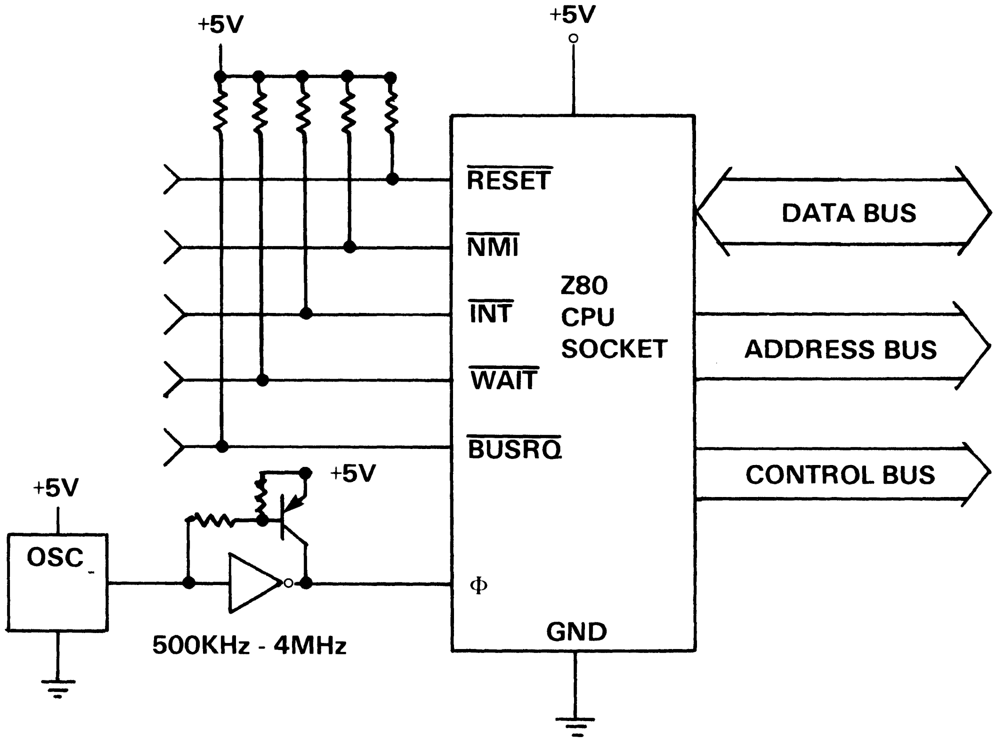
```
$AIMZ8Ø TEST(CR)          < load and run AIM-Z8ØAE
                          < also load target program "TEST"
AIM-Z8ØAE VERSION 1.Ø

,I (CR)          < display target memory map

S = SYSTEM MEMORY  T = TARGET MEMORY  P = WRITES PROTECTED

ØØØØ S  .  .  .   .  .  .  .   .  .  .  .   .  .  .  .
1ØØØ .  .  .  .   .  .  .  .   .  .  .  .   .  .  .  .
2ØØØ .  .  .  .   .  .  .  .   .  .  .  .   .  .  .  .
3ØØØ .  .  .  .   .  .  .  .   .  .  .  .   .  .  .  .
4ØØØ .  .  .  .   .  .  .  .   .  .  .  .   .  .  .  .
5ØØØ .  .  .  .   .  .  .  .   .  .  .  .   .  .  .  .
6ØØØ .  .  .  .   .  .  .  .   .  .  .  .   .  .  .  .
7ØØØ .  .  .  .   .  .  .  .   .  .  .  .   .  .  .  .
8ØØØ .  .  .  .   .  .  .  .   .  .  .  .   .  .  .  .
9ØØØ .  .  .  .   .  .  .  .   .  .  .  .   .  .  .  .
AØØØ .  .  .  .   .  .  .  .   .  .  .  .   .  .  .  .
BØØØ .  .  .  .   .  .  .  .   .  .  .  .   .  .  .  .
CØØØ .  .  .  .   .  .  .  .   .  .  .  .   .  .  .  .
DØØØ .  .  .  .   .  .  .  .   .  .  .  .   .  .  .  .
EØØØ SP SP SP SP  SP SP SP SP  SP SP SP SP  SP SP SP SP
FØØØ SP SP SP SP  SP SP SP SP  SP SP SP SP  SP SP SP T
```

```
,M Ø,F(CR)              < display "TEST" program
ØØ ØØ 3E ØØ 3C 3C  2F 3C 3C 2F  18 F6 ØØ ØØ  ØØ ØØ ØØ ØØ
,S Ø(CR)                < single-step thru program

 PC   AF   I IF  BC   DE   HL   DISASSEMBLY      IX   IY   SP
ØØØØ FFA9 ØØ41 FFFF FFFF Ø2ØØ  LD   A,Ø        BFFF FFFF Ø1FØ   7 uSec
ØØØ2 ØØA9 ØØ41 FFFF FFFF Ø2ØØ  INC  A          BFFF FFFF Ø1FØ  11 uSec (CR)
ØØØ3 Ø1Ø1 ØØ41 FFFF FFFF Ø2ØØ  INC  A          BFFF FFFF Ø1FØ  15 uSec (CR)
ØØØ4 Ø2Ø1 ØØ41 FFFF FFFF Ø2ØØ  CPL             BFFF FFFF Ø1FØ  .
,B 4(CR)                < set a software breakpoint

,E Ø(CR)                <execute target program

,
SWBP ENCOUNTERED   15 uSec
 PC   AF   I IF  BC   DE   HL   DISASSEMBLY        IX   IY   SP
ØØØ4 Ø2Ø1 ØØ41 FFFF FFFF Ø2ØØ  CPL               BFFF FFFF Ø1FØ  .
,T 5,-4(CR)             < display history memory

OFFS ADDR DB  DISASSEMBLY      TYPE  PA8-----PAØ  PB8-----PBØ
-ØØ4 ØØØØ 3E  LD   A,Ø         FETC  1 1111 1111  1 1111 1111
-ØØ3 ØØØ1 ØØ                   MRD   1 1111 1111  1 1111 1111
-ØØ2 ØØØ2 3C  INC  A           FETC  1 1111 1111  1 1111 1111
-ØØ1 ØØØ3 3C  INC  A           FETC  1 1111 1111  1 1111 1111
+ØØØ ØØØ4 5B  BREAKPOINT CODE  FETC  1 1111 1111  1 1111 1111
,B C,A(CR)              < clear all breakpoints

,B Ø,H(CR)              < set a hardware breakpoint

,E Ø(CR)                < execute target program

,
HWBP ENCOUNTERED   7 uSec
```

```
 PC   AF   I IF  BC   DE   HL   DISASSEMBLY        IX   IY   SP
ØØØ2 ØØØ1 ØØ41 FFFF FFFF Ø2ØØ  INC  A            BFFF FFFF Ø1FØ  .
,B H,O(CR)                < update history options


                  PA7----PAØ A15------------------AØ
TRIGGER WORD IS: XXXX XXXX ØØØØ ØØØØ ØØØØ ØØØØ  (CR)
UPDATE:          XXXX XXXX ØØØØ ØØØØ ØØØØ ØØØØ


TRIGGER STROBE IS (MRD,LE) ;TO CHANGE SELECT ONE:
      MRD(Ø)   MWR(1)  MREQ(2)  IORD(3)
     IOWR(4)  IORQ(5)  INTA(6)  PA8(7)   [ ,LE(8) ,TE(9)]  --> (CR)

EVENT COUNT IS: ØØØ1 --> 2(CR)

DELAY COUNT IS: ØØØ --> 4(CR)

HISTORY CLOCK IS (MRD MWR IORD IOWR ) ;TO CHANGE SELECT ANY:
      MRD(Ø)   MWR(1)   MRF(2)
     IORD(3)  IOWR(4)  PA8+(5)  PA8-(6)   --> (CR)


HISTORY CLOCK ENABLE IS (ALL) CYCLES ;TO CHANGE SELECT ONE:
       ALL(Ø)  DMA(1)  CPU(2)  PA7L(3)  PA7H(4)
             TWORD(6)        [ ,PB8L(8) ,PB8H(9)]  --> (CR)

,E Ø(CR)                  < execute again


,
HWBP ENCOUNTERED   62 uSec
 PC   AF   I IF  BC   DE   HL   DISASSEMBLY        IX   IY   SP
ØØØ5 FD3B ØØ41 FFFF FFFF Ø2ØØ  INC  A            BFFF FFFF Ø1FØ  .
,T F,-A(CR)               < display history memory

OFFS ADDR DB  DISASSEMBLY       TYPE  PA8------PAØ  PB8------PBØ
-ØØA ØØØØ 3E  LD   A,Ø          FETC  1 1111 1111  1 1111 1111
-ØØ9 ØØØ1 ØØ                    MRD   1 1111 1111  1 1111 1111
```

```
-ØØ8 ØØØ2 3C  INC  A          FETC  1 1111 1111  1 1111 1111
-ØØ7 ØØØ3 3C  INC  A          FETC  1 1111 1111  1 1111 1111
-ØØ6 ØØØ4 2F  CPL             FETC  1 1111 1111  1 1111 1111
-ØØ5 ØØØ5 3C  INC  A          FETC  1 1111 1111  1 1111 1111
-ØØ4 ØØØ6 3C  INC  A          FETC  1 1111 1111  1 1111 1111
-ØØ3 ØØØ7 2F  CPL             FETC  1 1111 1111  1 1111 1111
-ØØ2 ØØØ8 18  JR   -ØAH       FETC  1 1111 1111  1 1111 1111
-ØØ1 ØØØ9 F6                  MRD   1 1111 1111  1 1111 1111
+ØØØ ØØØØ 3E  LD   A,Ø        FETC  1 1111 1111  1 1111 1111
+ØØ1 ØØØ1 ØØ                  MRD   1 1111 1111  1 1111 1111
+ØØ2 ØØØ2 3C  INC  A          FETC  1 1111 1111  1 1111 1111
+ØØ3 ØØØ3 3C  INC  A          FETC  1 1111 1111  1 1111 1111
+ØØ4 ØØØ4 2F  CPL             FETC  1 1111 1111  1 1111 1111
,B C,A(CR)                < clear all breakpoints

,B 1,M,T(CR)              < set 1 millisecond timer breakpoint

,E Ø(CR)                  < execute target program

,TIMER BREAK ENCOUNTERED  989 uSec
 PC   AF   I IF  BC   DE   HL   DISASSEMBLY      IX   IY   SP
ØØØØ ØØ93 ØØ41 FFFF FFFF Ø2ØØ  LD   A,Ø         BFFF FFFF Ø1FØ  .
,B C,A(CR)                < clear all breakpoints

,E (CR)                   < resume execution
,(ESC)                    < press ESCape key
, 3,126,713 uSec
 PC   AF   I IF  BC   DE   HL   DISASSEMBLY      IX   IY   SP
ØØØ2 ØØ93 ØØ41 FFFF FFFF Ø2ØØ  INC  A           BFFF FFFF Ø1FØ  .
```

FIGURE 2.2 MINIMUM TARGET SYSTEM



SECTION 3.Ø
AIM-Z8ØAE OPERATION

3.1 INTRODUCTION

This section describes the operation of the AIM-Z8ØAE control program (AIMZ8Ø). A generalized description of the command format is provided as well as a detailed description of each command, complete with examples.

FIGURE 3.1 AIM-Z8Ø COMMAND SUMMARY

,B (CR)	< Tabulate breakpoints
,B a(CR)	< Set software breakpoint.
,B a,H(CR)	< Set hardware breakpoint.
,B a,C(CR)	< Clear breakpoint.
,B C,A(CR)	< Clear all breakpoints.
,B H,O(CR)	< Specify History or HWBP Options.
,B t,u,T(CR)	< Set timer breakpoint.
,B x,C,T(CR)	< Clear timer breakpoint.
,C s,f,d(CR)	< Copy s through f to d.
,D (CR)	< Create map file
,D s,f(CR)	< Dump s through f.
,E (CR)	< Continue execution.
,E s(CR)	< Begin execution at s.
,E s,f(CR)	< Execute with breakpoint at f.
,ESC	< Terminate execution.
,F s,f,d(CR)	< Fill s through f with d.
,G [file] (CR)	< Get binary file.
,H a+b-c+...+y-z=x(CR)	< Hexadecimal arithmetic.
,I (CR)	< Display Target memory map.

,I s,f(CR)	< Initialize Target memory map.
,L s,f,d(CR)	< Locate all d in s through f.
,M a(CR)	< Examine/Modify Target memory.
,M s,f(CR)	< Tabulate or list Target memory.
,M s,x,a(CR)	< Select Disassembly mode.
,O (CR)	< Clear relative Offset.
,O a(CR)	< Set relative Offset.
,P a(CR)	< Examine/Modify Port a.
,P s,f(CR)	< Tabulate Ports s through f.
,Q (CR)	< Quit.
,R (CR)	< Display registers.
,R n(CR)	< Specify number of registers.
,R n,H(CR)	< Specify heading option.
,S (CR)	< Continue single-Step.
,S a(CR)	< Begin single-Step.
,S a,n(CR)	< Begin multi-Step.
,S a,n,m(CR)	< Change display mode.
,T (CR)	< Trace History.
,T n(CR)	< Specify number of lines printed.
,T n,o(CR)	< Specify Offset from breakpoint.
,T n,o,f(CR)	< Specify print Format.
,V [file](CR)	< Verify file.
,W (CR)	< Disable Write to alternate LUN.
,W n(CR)	< Enable Write to alternate LUN.
,Z (CR)	< Initialize Target, clear breakpoints, (recovery for ERRORS 61, 62, 63)
,% s,f(CR)	< Target memory test.
,% s,f,o(CR)	< Specify options.

3.2 PREPARATION

Target programs may be created, edited, assembled and linked using the MOSTEK disk-based development system and then loaded into Target memory and immediately tested. Loading a Target program at initialization is optional since it may be desired to debug a short program which is entered using the M command. When powering up the System, the diskettes should NOT be installed. The development system should be powered up first, then the Target system. After all power is on the system is booted up as normal. When powering down the reverse sequence should be taken.

3.3 INITIALIZATION

After the system is powered up and FLP-8ØDOS is operating AIMZ8Ø.BIN may be loaded and executed using the implied run command. An optional File name may also be entered to automatically load the Target program. The format for initializing AIMZ8Ø follows.

```
$AIMZ8Ø [file name](CR)
AIMZ8Ø VERSION 1.Ø
'
```

The AIMZ8Ø program will be loaded and executed and the sign on message and prompt character will be printed. If the file AIMZ8Ø.MAP exists then it also will be loaded to specify the system configuration and if a Target program is specified it will be loaded into Target memory space.

3.4 ERRORS

If an error is encountered during initialization, an error message will be printed noting the error, and the command mode will be entered as normal. Error messages may also be printed during operation if the system fails to function as expected, or if the operator attempts an operation which is not allowed or does not make sense. Table 1 summarizes the possible error messages and explains the probable cause.

TABLE 3.1 ERRORS

*** ERROR 6Ø INVALID SYNTAX

This message is generated if the wrong number or type of operands for a command are entered.

*** ERROR 61 TARGET NOT FUNCTIONAL

This message is generated when the system is first initialized if a write to the interface RAM does not function properly. This may be caused by the lack of Target clock or power.

*** ERROR 62 TARGET DOES NOT RESPOND

This message is generated when the system is first initialized if the Target Z8Ø CPU does not begin execution in the interface RAM properly. This may be caused if the signals RESET, BUSRQ, or WAIT are active during initialization.

*** ERROR 63 TARGET HANDSHAKE TIMEOUT

This message is generated if a Target access is attempted when the Target Z8Ø CPU is not executing in the interface RAM properly. This may be caused if the signals BUSRQ or WAIT are active during the access.

*** ERROR 64 INVALID MNEMONIC

This message is generated if the user enters a mnemonic which is not recognized by the system. TABLE 3.2 lists the valid mnemonics.

*** ERROR 65 OPERATION NOT ALLOWED WHEN EXECUTING

This message is generated if the user attempts a Target access when the Target program is being executed. To terminate execution enter ESCAPE, then attempt the access.

*** ERROR 66 MAXIMUM BREAKPOINTS EXCEEDED

This message is generated if the user attempts to set more than eight software breakpoints.

*** ERROR 67 TARGET MEMORY FAILURE

This message is generated by the F command if the data fails to be filled properly in RAM. The fill will be completed even tnough errors may occur.

*** ERROR 68 INVALID OPCODE

This message is generated if an attempt is made to disassemble an undocumented Z8Ø instruction.

*** ERROR 69 INVALID OPERAND

This message is generated if an attempt is made to disassemble a Z8Ø instruction with an invalid operand.

*** ERROR 6A INVALID COMMAND TERMINATOR

This message is generated if an attempt is made to terminate a command with an invalid character.

*** ERROR 6B DATA ENTERED EXCEEDS FOUR BYTES

This message is generated if an attempt is made to enter more than four bytes at time when entering code using the disassembly mode of the M command.

*** ERROR 6C DISK I/O ERROR

This message is generated if a disk unit or I/O device is not ready or fails to function when an output is attempted to an alternate output device using the W command.

*** ERROR 6D TARGET MEMORY COMPARE

This message is generated when using the V command if the file does not compare with the contents of memory. The address, memory data and file data which failed are printed after this message.

*** ERROR 6E TOO MANY BACK STEPS

This message is generated when using the M command in the disassembly mode if an attempt is made to backup more than sixteen instructions.

3.5 COMMAND FORMAT

The general command format for all AIMZ8Ø commands is:

```
,x a,b,c(CR)
```

Where x is the single character specifing the command. The System echos the command character followed by a space. Up to three operands may then be entered followed by a Carriage Return. The operands may be either address or data values depending on the command and may be entered in one of several formats described in the following paragraphs.

3.6 HEXADECIMAL NUMBERS

Address or data operands which are hexadecimal numbers may be entered as hexadecimal digits. Leading zeros are ignored, and if more than four digits are entered for addresses or two digits for data, then only the right most four or two digits are accepted. Hexadecimal numbers should NOT be terminated with an upper case H.

3.7 ASCII LITERAL

Operands may be entered as ASCII literals by preceeding the character with the character L. The eight bit binary code for the ASCII character will then be used as the value of the operand.

3.8 MNEMONICS

Operands may be entered as one or two character mnemonics if preceeded by the character ":". The value of the mnemonic is looked up in the resident mnemonic table. If the mnemonic is not found, an error message is generated. Mnemonics specify register values instead of memory locations. Table 3.2 lists the mnemonics recognized by AIMZ8Ø.

TABLE 3.2 AIM-Z8Ø MNEMONICS

:PC	Program Counter
:A	Accumulator
:F	Flags
:I	Interrupt Vector Register (upper 8 bits)
:IF	Interrupt flip-flop
:B	B register
:C	C register
:D	D register
:E	E register
:H	H register
:L	L register
:A'	A' register
:F'	F' register
:B'	B' register
:C'	C' register
:D'	D' register
:E'	E' register
:H'	H' register
:L'	L' register
:IX	IX register
:IY	IY register
:SP	Stack pointer
:TS	T - State length in nanoseconds

3.9 RELATIVE ADDRESS

Operands may be entered as a relative address if the character R preceeds the hexadecimal operand. The relative offset (specified by the O command) is added to the value entered to generate the absolute address.

3.1Ø IMPLIED MEMORY ADDRESS POINTER

The current value of the memory address pointer may be specified for the operand by using the character "$". This is useful when making calculations of parameters relative to the address in memory.

3.11 ADDED OR SUBTRACTED NUMBERS

Hexadecimal arithmetic operations may be performed in line as operands are entered. This is very useful for adding offsets and if used with the implied memory address pointer, for computing relative branch addresses.

3.12 EQUAL SIGN

When performing hexadecimal arithmetic, the current value of the operand may be displayed by entering "=".

3.13 SPECIAL KEYS

For some commands special keys allow extended flexibility. The function of these special keys are given in TABLE 3.3.

TABLE 3.3 SPECIAL KEYS

KEY	FUNCTION
PERIOD (.)	Abort command and return to the command mode with the operand not updated.
SPACE BAR (SP)	Suspend printout if printing or continue printout if printout is suspended.
SLASH(/)	Abort command and return to the command mode with the operand updated.
UPCARET (UP)	Backup to previous address or option or display updated address or option.
CARRIAGE RETURN (CR)	Advance to next address or option or return to command mode.

3.14 B-BREAKPOINT COMMAND

FORMATS:

,B (CR)	< Tabulate breakpoints	(1)
,B a(CR)	< Set software breakpoint	(2)
,B a,H(CR)	< Set hardware breakpoint	(3)
,B a,C(CR)	< Clear breakpoint	(4)
,B C,A(CR)	< Clear All breakpoints	(5)
,B H,O(CR)	< Specify History Options	(6)
,B t,u,T(CR)	< Set timeout breakpoint	(7)
,B x,C,T(CR)	< Clear timeout breakpoint	(8)

3.15 DESCRIPTION, B COMMAND

Format (1) of the B command is used to tabulate the hardware, software, and timer breakpoints. There are no operands associated with this format and it may be used at any time.

Format (2) is used to Set a software breakpoint. Up to eight software breakpoints may be set. The address of the breakpoint is stored in a table but the breakpoint code (5BH) is not inserted into Target memory until the E-EXECUTE command is entered. Hardware on the Control module detects the breakpoint code and forces the breakpoint to occur. Breakpoint codes are removed from Target memory when any type breakpoint occurs, but remain in the breakpoint table until the user clears them. If an attempt is made to set more than eight software breakpoints, then an error message is printed. Software breakpoints may only be set on instruction boundaries. If a software breakpoint is set at an address which is not the first byte of an instruction, it will not be recognized and the instruction will be executed incorrectly.

Format (3) is used to set a hardware breakpoint. Only one hardware breakpoint is allowed and when a new hardware breakpoint is set the previous hardware breakpoint is cleared. The hardware breakpoint does not replace any of the users code and is functional in ROM as well as RAM. The hardware breakpoint may be set anywhere in memory (program or data) and will be recognized. If the hardware breakpoint is set on subsequent bytes of an instruction, one additional instruction may be executed after the breakpoint.

Format (4) is used to clear a breakpoint at a specific address. Both hardware and software breakpoints (if any) are cleared. If no breakpoint is set at the specified address, the command has no effect.

Format (5) is used to clear all breakpoints (hardware, software, or timer, if any). If no breakpoints are set then the command has no effect.

Format (6) is used to specify the History or hardware breakpoint options. When this format is entered, a prompt sequence is started which allows the user to examine and/or update the following six options:

1. The first option allows the user to examine/update the twenty-four bit trigger word. The user may specify either 1 (high), Ø (low) or X (don't care) for each bit of the trigger word. The character (SP) will advance to the next bit of the trigger word while (BS) will backup to the previous bit. The left most eight bits of the trigger word (PA7-PAØ) corresponds to the A PROBES on the Buffer module. The probes are color coded to standard resistor color codes (Ø-black, 1-brown, 2-red, 3-orange, 4-yellow, 5-green, 6-blue, 7-violet, 8-grey). The right most sixteen bits of the trigger word (A15-AØ) corresponds to the Target Z8Ø address bus.

2. The second option is the trigger strobe. The trigger strobe may be updated by .entering one of the following numbers correspongingto the desired trigger strobe. The leading or trailing edge of the trigger strobe signal may be selected by entering a comma followed by a second digit corresponding to the desired edge. If no edge is specified the leading edge is selected.

Ø - MEMORY READ
1 - MEMORY WRITE
2 - MEMORY READ OR WRITE
3 - PORT READ
4 - PORT WRITE
5 - PORT READ OR WRITE
6 - INTERUPT ACKNOWLEDGE
7 - PROBE A8
x,8 - LEADING EDGE OF ABOVE SIGNAL
x,9 - TRAILING EDGE OF ABOVE SIGNAL

3. The third option is the EVENT COUNT. The EVENT COUNT may be specified by entering a Hexadecimal number (up to four digits if strapped for sixteen bits or two digits if strapped for eight bits). The EVENT COUNT is the number of occurances of the trigger word which must occur before the breakpoint sequence is initiated. The EVENT COUNT is initialized to "1" when the AIMZ8Ø is loaded.

4. The forth option is the DELAY COUNT. The DELAY COUNT may be specified by entering a Hexadecimal number up to three digits for the ten bit counter. The DELAY COUNT is the number of history clock cycles the system is allowed to execute after the EVENT COUNT is satisfied and before the hardware breakpoint actually occurs.

5. The fifth option allows the user to specify the History clock source. To update the history clock source one or more of the following digits corresponding to the desired history clock may be entered. If more than one digit is entered, the signals are "ORed" together. Spaces or commas between the digits are optional.

Ø	MEMORY READS
1	MEMORY WRITES
2	MEMORY REFRESHES
3	PORT READS
4	PORT WRITES
5	POSITIVE EDGE OF PROBE A8
6	NEGATIVE EDGE OF PROBE A8

6. The sixth option allows the user to specify the History clock enable signal. To update the History clock enable, one of the following numbers corresponding to the desired history clock enable signal is entered. An optional second number may be entered to further qualify the History clock depending on the state of probe B8 during the previous cycle.

Ø	ALL CYCLES, DMA AND CPU
1	DMA CYCLES ONLY
2	CPU CYCLES ONLY
3	ONLY IF PROBE A7 IS LOW
4	ONLY IF PROBE A7 IS HIGH
6	ONLY IS THE TRIGGER WORD MATCHES

x,8	THE ABOVE ONLY IF PROBE B8 WAS LOW
x,9	THE ABOVE ONLY IS PROBE B8 WAS HIGH

Format (7) is used to set a timeout breakpoint to occur after the specified execution time (measured by Target clock periods) has elapsed. The first operand "t" is the number of periods or time (in BCD). The second operand "u" is the scale multiplier and must be one of the following:

MULTIPLIER	SECOND OPERAND "u"
t states	T
microseconds	U
1Ø microseconds	1ØU
1ØØ microseconds	1ØØU
miliseconds	M
1Ø miliseconds	1ØM
1ØØ miliseconds	1ØØM
seconds	S
1Ø seconds	1ØS
1ØØ seconds	1ØØS

Before using the timer, set the t-state value to the proper value for your system (see example below).

Format (8) is used to clear the timer breakpoint if any. The first operand (x) is ignored and the remaining operands must be entered exactly as shown.

3.16 EXAMPLES, B COMMAND

```
$AIMZ8Ø TEST(CR)                < load AIM-Z8ØAE and "TEST" programs
AIM-Z8ØAE VERSION 1.Ø
,B 4(CR)              < set software breakpoint
,E Ø(CR)              < execute target program
,
SWBP ENCOUNTERED   15 USEC
 PC   AF  I IF  BC   DE   HL   DISASSEMBLY        IX   IY   SP
ØØØ4 Ø2Ø1 ØØ41 FFFF FFFF Ø2ØØ  CPL               BFFF FFFF Ø1FØ .
,B (CR)         < list breakpoints

SW BREAKPOINTS: ØØØ4
NO HW BREAKPOINT
NO TIMER BREAKPOINT
,B Ø,H(CR)              < set hardware breakpoint at Ø
,B 4,C(CR)              < clear breakpoint at 4
,B (CR)         < list breakpoints
NO SW BREAKPOINTS
HW BREAKPOINT: ØØØØ
NO TIMER BREAKPOINT
,B H,O(CR)              < update history options

                 PA7---PAØ A15--------------AØ
TRIGGER WORD IS: XXXX XXXX ØØØØ ØØØØ ØØØØ ØØØØ (CR)      < no change
UPDATE:          XXXX XXXX ØØØØ ØØØØ ØØØØ ØØØØ

TRIGGER STROBE IS (MRD,LE) ;TO CHANGE SELECT ONE:
      MRD(Ø)   MWR(1)  MREQ(2)  IORD(3)
     IOWR(4)  IORQ(5)  INTA(6)  PA8(7)   [ ,LE(8) ,TE(9)]  --> (CR)

EVENT COUNT IS: ØØØ2 --> 2(CR)

DELAY COUNT IS: ØØ4 --> (CR)

HISTORY CLOCK IS (MRD MWR IORD IOWR );TO CHANGE SELECT ANY:
      MRD(Ø)   MWR(1)   MRF(2)
     IORD(3)  IOWR(4)  PA8+(5)  PA8-(6)  -->(CR)
```

```
HISTORY CLOCK ENABLE IS (ALL) CYCLES ;TO CHANGE SELECT ONE:
        ALL(Ø)  DMA(1)  CPU(2)  PA7L(3)  PA7H(4)
              TWORD(6)       [ ,PB8L(8) ,PB8H(9)]  --> (CR)

 ,E Ø(CR)                 < execute target program
 ,
 HWBP ENCOUNTERED   62 USEC
  PC   AF   I IF  BC   DE   HL   DISASSEMBLY       IX   IY   SP
 ØØØ5 FD3B ØØ41 FFFF FFFF Ø2ØØ  INC A            BFFF FFFF Ø1FØ .
 ,B C,A(CR)               < clear all breakpoints
 ,B 1Ø,M,T(CR)            < set 1Ø millisecond timer breakpoint
 ,B (CR)          < list breakpoints

 NO SW BREAKPOINT
 NO HW BREAKPOINT
 TIMER BREAKPOINT:  9,963 USEC   < note slight round-off and conversion
                                   error
 ,E Ø(CR)                 < execute target program
 ,
 TIMER BREAK ENCOUNTERED   9,964 USEC
  PC   AF   I IF  BC   DE   HL   DISASSEMBLY       IX   IY   SP
 ØØØ8 ØØ93 ØØ41 FFFF FFFF Ø2ØØ  JR   -ØAH         BFFF FFFF Ø1FØ .

 ,M :TS(CR)               < change t-state value
 :TS 1ØØØ 25Ø(UP)
 :TS 25Ø .
 ,E Ø(CR)                 < execute again
 ,
 TIMER BREAK ENCOUNTERED  2,491 USEC     < note different time value
  PC   AF   I IF  BC   DE   HL   DISASSEMBLY       IX   IY   SP
 ØØØ8 ØØ93 ØØ41 FFFF FFFF Ø2ØØ  JR   -ØAH         BFFF FFFF Ø1FØ .
 ,M :TS(CR)               < restore old value
 :TS 25Ø 1ØØØ(UP)
 :TS 1ØØØ .
 ,B C,A(CR)
 ,
```

3.17 C-COPY COMMAND

FORMAT:

```
,C s,f,d(CR)
```

3.18 DESCRIPTION, C COMMAND

The COPY command is used to move a block of data (or code) from start address "s" through finish address "f" to destination address "d". The destination address can be in the range "s" through "f" (i.e. blocks may be moved a few bytes forward or backward if desired). When copying code, care should be taken to copy complete instructions on both ends, and the displacement should be corrected for relative branch instructions within the block which branch outside the block.

3.19 EXAMPLE, C COMMAND

```
,M Ø,1F(CR)               < note two different lines of code
ØØØØ 3E ØØ 3C 3C  2F 3C 3C 2F  18 F6 ØØ ØØ  ØØ ØØ ØØ ØØ
ØØ1Ø ØØ ØØ ØØ ØØ  ØØ ØØ ØØ ØØ  ØØ ØØ ØØ ØØ  ØØ ØØ ØØ ØØ
,C Ø,F,1Ø(CR)             < copy first line to second line
,M Ø,1F(CR)               < note result of copy
ØØØØ 3E ØØ 3C 3C  2F 3C 3C 2F  18 F6 ØØ ØØ  ØØ ØØ ØØ ØØ
ØØ1Ø 3E ØØ 3C 3C  2F 3C 3C 2F  18 F6 ØØ ØØ  ØØ ØØ ØØ ØØ
,
```

3.2Ø D-DUMP COMMAND.

FORMAT:

```
,D s,f(CR)
```

3.21 DESCRIPTION, D COMMAND

The DUMP command is used to save the block of Target memory starting at "s" through "f" on floppy diskette in a binary file. When the (CR) is received, the System will prompt the user to enter the desired file name. The System will then check if the file exists and and if the user specifies will erase the file and create a new file.

If a file name has been previously specified with the D, G, or V commands or when initially loading AIMZ8Ø, that file name has been retained. When a file name is requested, type only (CR) and the old file name will be used.

3.22 EXAMPLE, D COMMAND

```
,D Ø,F(CR)                  < dump address Ø thru ØFH

ENTER TARGET PROGRAM FILE NAME OR (CR) --> DK1:TEST(CR)
DKØ:TEST  .BIN[1],ALREADY EXISTS
ERASE? Y(CR)                < "TEST" should have already been on the disk
DO YOU WISH TO DUMP THE MAP FILE? (Y OR N) --> Y(CR)
ENTER MAP FILE NAME OR (CR) -->(CR)               < use default name, which
                                                    this time is "TEST"
DKØ:TEST  .MAP[1],ALREADY EXISTS
ERASE? Y(CR)
,                           < the machine does the rest
```

3.23 E-EXECUTE COMMAND

FORMAT:

,E s(CR)	< Begin execution at s	(1)
,E (CR)	< Continue execution	(2)
,E s,f(CR)	< Execute, breakpoint set at f	(3)

3.24 DESCRIPTION, E COMMAND

The E command is used to initiate real time execution of the Target Z8Ø CPU at the desired address. Before real time execution begins, all of the Z8Ø CPU registers are loaded from the register save area in system RAM and breakpoint codes are inserted in Target memory at the addresses where breakpoints are set. If an attempt is made to begin or continue execution at an address which has a breakpoint set, that instruction will be executed with the breakpoint codes not inserted, then the breakpoints will be inserted and real time execution started.

Format (1) is used to initiate execution at address "s".

Format (2) is used to continue execution from the last breakpoint or single step (i.e. the address in :PC is used).

Format (3) is used to automatically set a software breakpoint at the address specified by "f". If there are already eight breakpoints set, an error message will be printed.

3.25 EXAMPLES, E COMMAND

```
,B 2,H(CR)                     < Set hardware breakpoint at 2H.
,E Ø(CR)                       < Begin execution at ØH.
,
HWBP ENCOUNTERED  7Ø USEC
 PC   AF  I IF  BC   DE   HL   DISASSEMBLY        IX   IY   SP
ØØØ7 FFA9 ØØ41 FFFF FFFF Ø2ØØ  CPL               BFFF FFFF Ø1FØ .
,E (CR)          < continue execution
,
HWBP ENCOUNTERED  156 USEC
 PC   AF  I IF  BC   DE   HL   DISASSEMBLY        IX   IY   SP
ØØØ7 FFA9 ØØ41 FFFF FFFF Ø2ØØ  CPL               BFFF FFFF Ø1FØ .
,
,B C,A(CR)
,
```

3.26 ESCAPE-EXECUTION ESCAPE COMMAND

FORMAT:

```
,(ESC)
```

3.27 DESCRIPTION, ESCAPE COMMAND

The ESCAPE command is used to terminate real time execution and may be used any time a Target program is being executed. The ESCAPE command forces a breakpoint to occur wherever the Target Z8Ø CPU may be executing. If the Target Z8Ø CPU is halted, the breakpoint will not occur until an interrupt occurs. All registers and status is saved and printed so that real time execution may be resumed.

3.28 EXAMPLE, ESCAPE COMMAND

```
,E Ø(CR)                          < Begin execution at ØH.
,(ESC)                            < (ESC) entered.
, 2,5Ø7,298 USEC
 PC   AF  I IF  BC   DE   HL  DISASSEMBLY          IX   IY   SP
ØØØ3 Ø1Ø1 ØØ41 FFFF FFFF Ø2ØØ   CPL               BFFF FFFF Ø1FØ .
,
```

3.29 F-FILL COMMAND

FORMAT:

```
,F s,f,d(CR)
```

3.3Ø DESCRIPTION, F COMMAND

The F command is used to fill the block of Target memory "s" through "f" with the data "d". Each location is checked to verify the operation and if any location fails to fill, an error message is printed. The complete fill is attempted even if some locations may fail.

3.31 EXAMPLES, F COMMAND

```
,M 1Ø,1F(CR)              < look at it first
ØØ1Ø 3E ØØ 3C 3C  2F 3C 3C 2F  18 F6 ØØ ØØ  ØØ ØØ ØØ ØØ
,F 1Ø,1F,AA(CR)           < fill it with ØAAH
,M 1Ø,1F(CR)              < look at it again
ØØ1Ø AA AA AA AA  AA AA AA AA  AA AA AA AA  AA AA AA AA
,
,
```

3.32 G-GET COMMAND

FORMAT:

```
,G [file](CR)
```

3.33 DESCRIPTION, G COMMAND

The G command is used to load a binary file into Target RAM. For long files (greater than 16k) there may be some delay in loading the file. The G command may be used any time the Target system is not executing the Target program.

If a file name has been previously specified with the D, G, or V commands, or when initially loading AIMZ8Ø, then that file name has been retained. When a file name is requested, type only (CR) and the old file name will be used.

3.34 EXAMPLES, G COMMAND

```
,F Ø,F,AA(CR)           < destroy old program by overwriting
,M Ø,F(CR)              < look at it
ØØØØ AA AA AA AA  AA AA AA AA  AA AA AA AA  AA AA AA AA
,G DK1:TEST(CR)         < reload the program
,M Ø,F(CR)              < look at it again
ØØØØ 3E ØØ 3C 3C  2F 3C 3C 2F  18 F6 ØØ ØØ  ØØ ØØ ØØ ØØ
,
```

3.35 H-HEXADECIMAL ARITHMETIC

FORMAT:

```
,H a+b-c+...+y-z=x(CR)
```

3.36 DESCRIPTION, H COMMAND

The H command is used to perform hexadecimal arithmetic operations Signed addition and subtraction may be performed on any number of hexadecimal numbers. To display the answer x, the character '=' is entered. When (CR) is entered, the command mode is again entered.

3.37 EXAMPLES, H COMMAND

```
,H Ø1ØØ+Ø9ØØ=ØAØØ(CR)
,H 1ØØ+9ØØ-8ØØ=Ø2ØØ(CR)
,
```

3.38 I-INITIALIZE MEMORY MAP COMMAND

FORMAT:

```
,I (CR)                        < Display memory map     (1)
,I s,f(CR)                     < Update memory map      (2)
```

3.39 DESCRIPTION, I COMMAND

The I command is used to initialize or display the configuration of the Target memory map. Each 256 byte block of Target memory space may be designated as Target memory, Emulation RAM, or Non existant. Each block may also be optionally designated as write protected. If an access is attempted to non existant memory or a write is attempted to write protected memory, a breakpoint will automatically be generated.

Format (1) is used to display the current memory map. This map is displayed as a table of 16 lines with 16 entries in each line with the starting address of the first entry at the start of each line. Each entry corresponds to one 256 byte block of Target memory space. There are five possible types of entry designated as follows.

.	Non existant, access generates a breakpoint.
T	Target memory is used (if it exists).
TP	Target memory is used, write generates a breakpoint.
S	Emulation RAM is used.
SP	Emulation RAM is used, write generates a breakpoint.

Format (2) is used to update the memory map for the address space starting at "s" through "f". When (CR) is entered with two operands, the user will be prompted to enter either "Y" (for Yes) or "N" (for No) for up to three of the following questions.

```
IS THIS BLOCK SYSTEM MEMORY? (Y/N) -->_

IS THIS BLOCK TARGET MEMORY? (Y/N) -->_

ARE WRITES ALLOWED? (Y/N) -->_
```

The system will update the memory map accordingly for this memory block. The configuration map is also updated by the G command.

3.4Ø EXAMPLES, I COMMAND.

```
,I (CR)          < display target memory map

S = SYSTEM MEMORY  T = TARGET MEMORY  P = WRITES PROTECTED

ØØØØ  S  .  .  .    .  .  .  .    .  .  .  .    .  .  .  .
1ØØØ  .  .  .  .    .  .  .  .    .  .  .  .    .  .  .  .
2ØØØ  .  .  .  .    .  .  .  .    .  .  .  .    .  .  .  .
3ØØØ  .  .  .  .    .  .  .  .    .  .  .  .    .  .  .  .
4ØØØ  .  .  .  .    .  .  .  .    .  .  .  .    .  .  .  .
5ØØØ  .  .  .  .    .  .  .  .    .  .  .  .    .  .  .  .
6ØØØ  .  .  .  .    .  .  .  .    .  .  .  .    .  .  .  .
7ØØØ  .  .  .  .    .  .  .  .    .  .  .  .    .  .  .  .
8ØØØ  .  .  .  .    .  .  .  .    .  .  .  .    .  .  .  .
9ØØØ  .  .  .  .    .  .  .  .    .  .  .  .    .  .  .  .
AØØØ  .  .  .  .    .  .  .  .    .  .  .  .    .  .  .  .
BØØØ  .  .  .  .    .  .  .  .    .  .  .  .    .  .  .  .
CØØØ  .  .  .  .    .  .  .  .    .  .  .  .    .  .  .  .
DØØØ  .  .  .  .    .  .  .  .    .  .  .  .    .  .  .  .
EØØØ  SP SP SP SP   SP SP SP SP   SP SP SP SP   SP SP SP SP
FØØØ  SP SP SP SP   SP SP SP SP   SP SP SP SP   SP SP SP T
,I Ø,FFF(CR)

IS THIS BLOCK SYSTEM MEMORY? (Y/N) -->Y
ARE WRITES ALLOWED? (Y/N) -->Y
,I 1ØØØ,1FFF(CR)

IS THIS BLOCK SYSTEM MEMORY? Y/N) -->N
IS THIS BLOCK TARGET MEMORY? (Y/N) -->Y
ARE WRITES ALLOWED? (Y/N) -->N
```

```
,I 2ØØØ,2FFF(CR)

IS THIS BLOCK SYSTEM MEMORY? (Y/N) -->N
IS THIS BLOCK TARGET MEMORY? (Y/N) -->N
,I (CR)

S = SYSTEM MEMORY  T = TARGET MEMORY  P = WRITES PROTECTED

ØØØØ  S  S  S  S   S  S  S  S   S  S  S  S   S  S  S  S
1ØØØ  TP TP TP TP  TP TP TP TP  TP TP TP TP  TP TP TP TP
2ØØØ  .  .  .  .   .  .  .  .   .  .  .  .   .  .  .  .
3ØØØ  .  .  .  .   .  .  .  .   .  .  .  .   .  .  .  .
4ØØØ  .  .  .  .   .  .  .  .   .  .  .  .   .  .  .  .
5ØØØ  .  .  .  .   .  .  .  .   .  .  .  .   .  .  .  .
6ØØØ  .  .  .  .   .  .  .  .   .  .  .  .   .  .  .  .
7ØØØ  .  .  .  .   .  .  .  .   .  .  .  .   .  .  .  .
8ØØØ  .  .  .  .   .  .  .  .   .  .  .  .   .  .  .  .
9ØØØ  .  .  .  .   .  .  .  .   .  .  .  .   .  .  .  .
AØØØ  .  .  .  .   .  .  .  .   .  .  .  .   .  .  .  .
BØØØ  .  .  .  .   .  .  .  .   .  .  .  .   .  .  .  .
CØØØ  .  .  .  .   .  .  .  .   .  .  .  .   .  .  .  .
DØØØ  .  .  .  .   .  .  .  .   .  .  .  .   .  .  .  .
EØØØ  SP SP SP SP  SP SP SP SP  SP SP SP SP  SP SP SP SP
FØØØ  SP SP SP SP  SP SP SP SP  SP SP SP SP  SP SP SP T
,
```

3.41 L-LOCATE COMMAND

FORMAT:

```
,L s,f,d(CR)
```

3.42 DESCRIPTION, L COMMAND

The L command is used to locate all occurances of data byte "d" in Target memory starting at address "s" through "f".

3.43 EXAMPLES, L COMMAND

```
,G (CR)           < load program
,M Ø,F(CR)                < look at it
ØØØØ 3E ØØ 3C 3C  2F 3C 3C 2F  18 F6 ØØ ØØ  ØØ ØØ ØØ ØØ
,L Ø,F,3C(CR)             < locate all Ø3CH
ØØØ2 3C
ØØØ3 3C
ØØØ5 3C
ØØØ6 3C
,
```

3.44 M-MODIFY COMMAND

FORMAT:

,M a(CR)	< Examine/Modify Target memory	(1)
,M s,f(CR)	< Tabulate Target memory	(2)
,M s,x,a(CR)	< Disassemble Target memory	(3)

3.45 DESCRIPTION, M COMMAND.

Format (1) of the M command is used to examine or modify Target memory starting at address "a". When this format is entered, the address "a" is printed followed by the contents of address "a" and then (SP). The user may optionally update memory by entering the desired hexadecimal value. He may also optionally specify another address to be displayed by entering a comma and the address in addition to the update data. To perform the next M command operation (if any) one of the special keys (.), (/), (UP), or (CR) must then be entered. The function of the special key depends if memory is updated as follows.

SPECIAL KEY	UPDATE ENTERED	NO UPDATE
(.)	Return to AIM monitor not updated.	Return to AIM monitor.
(/)	Return to AIM monitor updated.	Return to AIM monitor.
(UP)	Redisplay address a.	Display address a-1.
(CR)	Display address a+1.	Display address a+1.

If "a" is a register nmenonic, this format is used to examine and/or update the register. Operation is exactly the same in other respects when examining or modifing registers as when examining or modifing memory. Table 3-3 is a complete list of the nmenonics which are recognized.

Format (2) of the M command is used to Tabulate Target memory addresses "s" through "f". Target memory is tabulated with sixteen bytes per line with the address of the first byte of each line at the start of each line. The printout may be suspended by entering (SP) and may be continued by entering (SP) again. When the printout is complete, the monitor mode is reentered. This format may also be used to tabulate the Z8Ø registers by using nmenonics for "s" and "f".

Format (3) is used to display Target memory and to switch between the tabulate and disassembly modes. If operand 3 is 'D' the disassembly mode is selected, otherwise the tabulate mode is selected. The disassembly mode remains selected until format 3 is used with operand 3 not "D". The mode remains selected when other formats are used.

When the disassembly mode is used, the address is printed followed by the opcode and operands in Hexadecimal, followed by the disassembled opcode. The user may then optionally enter up to four Hexadecimal bytes (delimited by commas) to update the code desired at this address. Next one of the following special keys must be entered. The function of the special key depends upon whether or not update data has been entered:

SPECIAL KEY	UPDATE ENTERED	NO UPDATE
(.)	Return to monitor without updating.	Return to monitor.
(SP)	Update, begin printing.	Begin printing.
(/)	Update, Return to monitor.	Return to monitor.
(UP)	Update, disassemble current instruction.	Disassemble previous instruction.
(CR)	Update, disassemble current instruction.	Disassemble next instruction.

Note that if (UP) is entered with no update, backing up is allowed only for sixteen instructions or to the start address whichever is less. Entering any other key will result in an error message.

3.46 EXAMPLE, M COMMAND

```
,M Ø(CR)                   < examine memory starting at Ø
ØØØØ 3E(CR)
ØØØ1 ØØ(CR)
ØØØ2 3C(CR)
ØØØ3 3C(CR)
ØØØ4 2F(CR)
ØØØ5 3C(CR)
ØØØ6 3C(CR)
ØØØ7 2F(CR)
ØØØ8 18(CR)
ØØØ9 F6(CR)
ØØØA ØØ AA(UP)             < change this one and confirm with (UP)
ØØØA AA(UP)
ØØØ9 F6(UP)                < back up
ØØØ8 18 .
,M Ø,F(CR)                 < tabulate a whole line
ØØØØ 3E ØØ 3C 3C  2F 3C 3C 2F  18 F6 ØØ ØØ  ØØ ØØ ØØ ØØ
,M Ø,F,D(CR)               < disassemble
ØØØØ 3EØØ       LD   A,Ø          (CR)
ØØØ2 3C         INC  A            (CR)
ØØØ3 3C         INC  A            (CR)
ØØØ4 2F         CPL               .
,M 5(CR)                   < disassembly mode still set
ØØØ5 3C         INC  A            (CR)
ØØØ6 3C         INC  A            (CR)
ØØØ7 2F         CPL               (CR)
ØØØ8 18F6       JR   -ØAH         C3,ØØ,ØØ(UP) < update this one and
                                                   confirm
ØØØ8 C3ØØØØ     JP   ØØØØH        .
,M Ø,F,Ø(CR)              < turn off disassembly
ØØØØ 3E ØØ 3C 3C  2F 3C 3C 2F  C3 ØØ ØØ ØØ  ØØ ØØ ØØ ØØ
,M 8(CR)
ØØØ8 C3 18(CR)             < restore relative jump
ØØØ9 ØØ Ø-$=FFF6(UP)       < compute displacement = destination-$
ØØØ9 F6 .
,
```

3.47 O-SET OFFSET COMMAND

FORMAT:

```
,O a(CR)                      < Set relative Offset          (1)
,O (CR)                       < Clear relative Offset        (2)
```

3.48 DESCRIPTION, O COMMAND

Format (1) is used to set the relative offset a. Once the relative offset has been set, the relative address (absolute address + offset) will be printed in addition to the absolute address. Also when entering operands which are addresses, the relative address may be specified with the prefix R. If no offset is set, then the Printing of Relative addresses is disabled.

Format (2) is used to clear the relative offset.

3.49 EXAMPLE, O COMMAND

```
,O 1ØØ(CR)                    < Set relative offset to 1ØØH.
,R (CR)                       < Display registers
'PC   PC   AF  I IF  BC   DE   HL   DISASSEMBLY      IX   IY   SP
'FFØ3 ØØØ3 Ø1Ø1 ØØ41 FFFF FFFF Ø2ØØ  INC  A          BFFF FFFF Ø1FØ

,O (CR)                       < Disable relative offset.
,R (CR)                       < Display registers.
 PC   AF  I IF  BC   DE   HL   DISASSEMBLY      IX   IY   SP
ØØØ3 Ø1Ø1 ØØ41 FFFF FFFF Ø2ØØ  INC  A          BFFF FFFF Ø1FØ
,
```

3.5Ø P-PORT COMMAND

FORMAT:

,P a(CR)	< Examine or modify Port.	(1)
,P s,f(CR)	< Tabulate ports.	(2)

3.51 DESCRIPTION, P COMMAND

Format (1) is used to examine or modify port "a". When this format is entered, the port address is displayed followed by space and the value read from the port. The user may optionally enter a hexadecimal number to be output to the port. Another port may also be examined by entering a comma and the address in addition to the update data. To perform the next P command operation, one of the special keys (.), (/), (UP) or (CR) must be entered. The function of the special key depends if output data had been entered as follows.

SPECIAL KEY	OUTPUT DATA ENTERED	NO OUTPUT DATA
(.)	Return to AIM monitor.	Return AIM monitor.
(/)	Output data, return to AIM monitor.	Return to AIM monitor.
(UP)	Output data, Display current port.	Display previous port.
(CR)	Output data, Display next port.	Display next port.

Format (2) is used to tabulate ports "s" through "f". Sixteen ports are tabulated per line with the address of the first port at the start of each line.

3.52 EXAMPLES, P COMMAND.

```
,P DØ(CR)                        < Examine port ØDØH.
DØ AA 55(UP)                     < Output 55H and verify.
DØ 55 (CR)                       < Advance.
D1 FF(CR)                        < Advance.
D2 AA 55(UP)                     < Output 55 to port ØD2H.
D2 55 (UP)                       < Backup.
D1 FF (UP)
DØ 55 .
,P Ø,FF(CR)                      < Tabulate all ports.

ØØ 78 78 78 78  78 78 78 78  78 78 78 78  78 78 78 78
1Ø 78 78 78 78  78 78 78 78  78 78 78 78  78 78 78 78
2Ø 78 78 78 78  78 78 78 78  78 78 78 78  78 78 78 78
3Ø 78 78 78 78  78 78 78 78  78 78 78 78  78 78 78 78
4Ø 78 78 78 78  78 78 78 78  78 78 78 78  78 78 78 78
5Ø 78 78 78 78  78 78 78 78  78 78 78 78  78 78 78 78
6Ø 78 78 78 78  78 78 78 78  78 78 78 78  78 78 78 78
7Ø 78 78 78 78  78 78 78 78  78 78 78 78  78 78 78 78
8Ø 78 78 78 78  78 78 78 78  78 78 78 78  78 78 78 78
9Ø 78 78 78 78  78 78 78 78  78 78 78 78  78 78 78 78
AØ 78 78 78 78  78 78 78 78  78 78 78 78  78 78 78 78
BØ 78 78 78 78  78 78 78 78  78 78 78 78  78 78 78 78
CØ 78 78 78 78  78 78 78 78  78 78 78 78  78 78 78 78
DØ 55 FF 55 FF  FF FF FF FF  ØØ ØØ ØØ ØØ  ØØ 8Ø FØ 78
EØ 78 78 78 78  78 78 78 78  78 78 78 78  78 78 78 78
FØ 78 78 78 78  78 78 78 78  78 78 78 78  78 78 78 78
,
```

3.53 Q-QUIT COMMAND

FORMAT:

```
,Q (CR)                       < quit
```

3.54 DESCRIPTION, Q COMMAND

The Quit command is used to exit AIMZ8Ø and return to the system monitor, closing any open files.

3.55 EXAMPLE, Q COMMAND

```
,Q (CR)                       < Quit
$
```

3.56 R-DISPLAY REGISTERS COMMAND

FORMAT:

```
,R (CR)                       < Display registers          (1)
,R n(CR)                      < Specify number             (2)
,R n,h(CR)                    < Specify optional heading   (3)
```

3.57 DESCRIPTION,R COMMAND

The R command (any format) is used to display the Z8Ø CPU register values which are updated and displayed any time a breakpoint is encountered or when single/multi-stepping.

Format (1) will cause the Z8Ø registers to be printed with the optional heading. The number of registers printed and the heading is specified by formats (2) and (3).

Format (2) is used to specify the number of register pairs to be printed and is a hexadecimal number between Ø and ØDH. Once the number of register pairs is specified it remains until changed with the R command.

Format (3) is used to specify whether the heading will also be printed. If operand "h" is "H" then the heading will be printed. Otherwise, the heading will not be printed. Once the heading option is specified it remains set until changed by the R command.

The number of registers printed and the heading option apply to the register printout which occurs after a breakpoint or when single stepping as well as when the R command is used.

3.58 EXAMPLES, R COMMAND

```
,R (CR)          < display registers
 PC   AF  I IF  BC   DE   HL   DISASSEMBLY       IX   IY   SP
ØØØ3 Ø1Ø1 ØØ41 FFFF FFFF Ø2ØØ  INC  A          BFFF FFFF Ø1FØ
,S Ø,1,ND(CR)             < turn off disassembly

 PC   AF  I IF  BC   DE   HL  A'F' B'C' D'E' H'L'  IX   IY   SP
ØØØØ Ø1Ø1 ØØ41 FFFF FFFF Ø2ØØ ØØ4Ø Ø1Ø1 FFFF 3ØFF BFFF FFFF Ø1FØ  7 USEC
ØØØ2 ØØØ1 ØØ41 FFFF FFFF Ø2ØØ ØØ4Ø Ø1Ø1 FFFF 3ØFF BFFF FFFF Ø1FØ  .
,R(CR)           < display all registers

 PC   AF  I IF  BC   DE   HL  A'F' B'C' D'E' H'L'  IX   IY   SP
ØØØ2 ØØØ1 ØØ41 FFFF FFFF Ø2ØØ ØØ4Ø Ø1Ø1 FFFF 3ØFF BFFF FFFF Ø1FØ
,R 6,Ø(CR)                < turn off heading

ØØØ2 ØØØ1 ØØ41 FFFF FFFF Ø2ØØ
,R 6,H(CR)                < turn on heading

 PC   AF  I IF  BC   DE   HL
ØØØ2 ØØØ1 ØØ41 FFFF FFFF Ø2ØØ
,
```

3.59 S-STEP COMMAND

FORMAT:

,S a(CR)	< Begin single-step	(1)
,S (CR)	< Continue single-step	(2)
,S a,n(CR)	< Begin multi-step	(3)
,S a,n,m(CR)	< Change display mode	(4)

3.6Ø DESCRIPTION, S COMMAND

The S command is used to single-step execute instructions in Target memory.

Format (1) is used to begin single-step execution at address "s". When (CR) is entered, the instruction at address "s" is executed and the Z8Ø CPU registers are printed and the system waits in the single-step mode for one of the following special keys to be entered.

SPECIAL KEY	FUNCTION
(.)	Terminate S command, return to monitor
(SP)	Begin multi-step if suspended, suspend multi-step if stepping.
(CR)	Single Step next instruction.

Format (2) is used to continue single-step execution. This format is the same as format (1) except the begin address is taken from the Program Counter save area register (:PC).

Format (3) is used to begin multi-step execution. The first operand "a" is the begin address and the second operand "n" is the number of instructions to execute up to ØFFH.

Note that the next instruction to be executed will also be displayed.

Format (4) is used to change the display mode for single-step operation. The system powers up in the disassembly. If the third operand "m" is "D" the disassembly mode is specified, otherwise the register mode is specified. Once a particular mode is specified, the mode is used until changed by the user. When the disassembly mode is used, instead of printing the Z8Ø registers each step, the Program Counter and Accumulator/Flags (before execution), the instruction opcode disassembled, and the Program Counter and Accumulator/Flags (after execution of the instruction) is printed for each step.

Interrupts (maskable or non-maskable) can occur during single-step execution. When an interrupt occurs, the program will begin single-step execution in the interrupt handling routine. WAIT states and DMA operations can also occur during single step execution. These occur at full speed and are transparent to the user.

3.61 EXAMPLES, S COMMAND

```
,S Ø(CR)                  < step from address Ø

 PC   AF   I IF  BC   DE   HL   DISASSEMBLY       IX   IY   SP
ØØØØ FFA8 ØØ4Ø Ø1Ø1 FFFF Ø2ØØ  LD   A,Ø          BFFF FFFF Ø1FØ  7 USEC
ØØØ2 ØØA8 ØØ4Ø Ø1Ø1 FFFF Ø2ØØ  INC  A            BFFF FFFF Ø1FØ  11 USEC(CR)
ØØØ3 Ø1ØØ ØØ4Ø Ø1Ø1 FFFF Ø2ØØ  INC  A            BFFF FFFF Ø1FØ  15 USEC(CR)
ØØØ4 Ø2ØØ ØØ4Ø Ø1Ø1 FFFF Ø2ØØ  CPL               BFFF FFFF Ø1FØ  .
,S (CR)           < resume stepping
 19 USEC
ØØØ5 FD3A ØØ4Ø Ø1Ø1 FFFF Ø2ØØ  INC  A            BFFF FFFF Ø1FØ  23 USEC(CR)
ØØØ6 FEA8 ØØ4Ø Ø1Ø1 FFFF Ø2ØØ  INC  A            BFFF FFFF Ø1FØ  .
,S Ø,4(CR)                 < multi-step
                          < see R command for additional example
 PC   AF   I IF  BC   DE   HL   DISASSEMBLY       IX   IY   SP
ØØØØ Ø1ØØ ØØ4Ø Ø1Ø1 FFFF Ø2ØØ  LD   A,Ø          BFFF FFFF Ø1FØ  7 USEC
ØØØ2 ØØØØ ØØ4Ø Ø1Ø1 FFFF Ø2ØØ  INC  A            BFFF FFFF Ø1FØ  11 USEC
ØØØ3 Ø1ØØ ØØ4Ø Ø1Ø1 FFFF Ø2ØØ  INC  A            BFFF FFFF Ø1FØ  15 USEC
ØØØ4 Ø2ØØ ØØ4Ø Ø1Ø1 FFFF Ø2ØØ  CPL               BFFF FFFF Ø1FØ  19 USEC
ØØØ5 FD3A ØØ4Ø Ø1Ø1 FFFF Ø2ØØ  INC  A            BFFF FFFF Ø1FØ  .
,
,
```

3.62 T-TRACE HISTORY COMMAND

FORMAT:

```
,T (CR)              < Trace from start of history.  (1)
,T n(CR)             < Trace n history samples.      (2)
,T n,o(CR)           < Trace relative to breakpoint. (3)
,T n,o,f(CR)         < Specify print format.         (4)
```

3.63 DESCRIPTION, T COMMAND

The T command is used to display the contents of the history RAM which was sampled during the last execute.

Format (1) is used to dump the history RAM starting at the earliest sample. When this format is entered, The heading is printed followed by the history data formatted in the following fields.

Offset from Breakpoint (OFFS)	1-4
Address bus (ADDR)	6-9
Data Bus (DB)	11-12
Data bus disassembled (DISASSEMBLY)	15-32
A Probes (PA8-PAØ)	34-44
B Probes (PB8-PBØ)	47-57

If a relative offset is set the relative address is printed in columns 1-5 and the other fields are moved to the right five spaces.

When using the T command, for any of the above formats the following special keys are used to control the printout.

SPECIAL KEY	FUNCTION
(.)	Terminate T command return to monitor
(SP)	Suspend printout if printing, continue printout if not printing.
(UP)	Display previous history sample
(CR)	Display next history sample
(LF)	Display heading and next history sample

The T command may be used at any time even if the Target program is being executed. If the Target program is not executing, the Trace printout will begin as soon as (CR) is entered for the T command. If the History module is not sampling (sampling stops if the Trigger word/strobe compares and no hardware breakpoint is set) the Trace printout will also begin immediately. If the History module is sampling, then the word "SAMPLING" will be printed. The user may then enter either (CR) or (.) If (.) is entered the AIM monitor will be reentered with no changes. If (CR) is entered, the history sampling will be terminated and the Trace printout begun.

Format (2) allows the number of lines of history printed to be specified. The operand "n" is a hexadecimal number between 1 and ØFFH which specifies the number of lines printed.

Format (3) is used to specify the offset from the breakpoint in the history RAM to begin display. The second operand "o" is a signed hexadecimal number between -3FFH and 3FFH which specifies this offset. If an offset is specified which is before the start of the history, the earliest sample is the default. If an offset is specified which is after the end of history, the most recent sample is the default.

Format (4) is used to specify the display format "f". The display format is a hexadecimal digit for which each bit enables a specific field of the history printout. The field enabled by each bit follows.

Bit Ø = 1	Display B probes
Bit 1 = 1	Display A probes
Bit 2 = 1	Display Cycle Type
Bit 3 = 1	Display Disassembly

3.64 EXAMPLES, T COMMAND

```
,B Ø,H(CR)                  < set hardware breakpoint

,B H,O(CR)                  < update history options

                  PA7---PAØ A15-------------AØ
TRIGGER WORD IS: XXXX XXXX ØØØØ ØØØØ ØØØØ ØØØØ  (CR)
UPDATE:          XXXX XXXX ØØØØ ØØØØ ØØØØ ØØØØ

TRIGGER STROBE IS (MRD,LE) ;TO CHANGE SELECT ONE:
      MRD(Ø)   MWR(1)  MREQ(2)  IORD(3)
     IOWR(4)  IORQ(5)  INTA(6)  PA8(7)   [ ,LE(8) ,TE(9)]  --> (CR)

EVENT COUNT IS: ØØØ1 --> 2(CR)

DELAY COUNT IS: ØØØ --> 4(CR)

HISTORY CLOCK IS (MRD MWR IORD IOWR ) ;TO CHANGE SELECT ANY:
      MRD(Ø)   MWR(1)   MRF(2)
     IORD(3)  IOWR(4)  PA8+(5)  PA8-(6) --> Ø1234(CR) < enable all cycles

HISTORY CLOCK ENABLE IS (ALL) CYCLES ;TO CHANGE SELECT ONE:

      ALL(Ø)  DMA(1)  CPU(2)  PA7L(3)  PA7H(4)
             TWORD(6)      [ ,PB8L(8) ,PB8H(9)]  --> (CR)

,E Ø(CR)                    < execute target program

,
HWBP ENCOUNTERED    58 USEC
 PC   AF  I IF  BC   DE   HL   DISASSEMBLY       IX   IY   SP
ØØØ4 Ø2ØØ ØØ4Ø FFFF FFFF Ø2ØØ  CPL               FBFF FF5D Ø1FØ  .
,T (CR)          < display history memory
```

```
OFFS ADDR DB  DISASSEMBLY       TYPE  PA8-----PAØ  PB8-----PBØ
-Ø12 ØØØØ 3E  LD    A,Ø         FETC  1 1111 1111  1 1111 1111
-Ø11 ØØ1Ø FF                    RFSH  1 1111 1111  1 1111 1111
-Ø1Ø ØØØ1 ØØ                    MRD   1 1111 1111  1 1111 1111
-ØØF ØØØ2 3C  INC   A           FETC  1 1111 1111  1 1111 1111
-ØØE ØØ11 FF                    RFSH  1 1111 1111  1 1111 1111
-ØØD ØØØ3 3C  INC   A           FETC  1 1111 1111  1 1111 1111
-ØØC ØØ12 FF                    RFSH  1 1111 1111  1 1111 1111
-ØØB ØØØ4 2F  CPL               FETC  1 1111 1111  1 1111 1111
-ØØA ØØ13 FF                    RFSH  1 1111 1111  1 1111 1111
-ØØ9 ØØØ5 3C  INC   A           FETC  1 1111 1111  1 1111 1111
-ØØ8 ØØ14 FF                    RFSH  1 1111 1111  1 1111 1111
-ØØ7 ØØØ6 3C  INC   A           FETC  1 1111 1111  1 1111 1111
-ØØ6 ØØ15 FF                    RFSH  1 1111 1111  1 1111 1111
-ØØ5 ØØØ7 2F  CPL               FETC  1 1111 1111  1 1111 1111
-ØØ4 ØØ16 FF                    RFSH  1 1111 1111  1 1111 1111
-ØØ3 ØØØ8 18  JR    -ØAH        FETC  1 1111 1111  1 1111 1111
-ØØ2 ØØ17 FF                    RFSH  1 1111 1111  1 1111 1111
-ØØ1 ØØØ9 F6                    MRD   1 1111 1111  1 1111 1111
+ØØØ ØØØØ 3E  LD    A,Ø         FETC  1 1111 1111  1 1111 1111
+ØØ1 ØØ18 FF                    RFSH  1 1111 1111  1 1111 1111
+ØØ2 ØØØ1 ØØ                    MRD   1 1111 1111  1 1111 1111
+ØØ3 ØØØ2 3C  INC   A           FETC  1 1111 1111  1 1111 1111
+ØØ4 ØØ19 FF                    RFSH  1 1111 1111  1 1111 1111
+ØØ5 ØØØ3 3C  INC   A           FETC  1 1111 1111  1 1111 1111
+ØØ6 ØØ1A FF                    RFSH  1 1111 1111  1 1111 1111
END OF HISTORY

START OF HISTORY
-Ø12 ØØØØ 3E  LD    A,Ø         FETC  1 1111 1111  1 1111 1111
-Ø11 ØØ1Ø FF                    RFSH  1 1111 1111  1 1111 1111
-Ø1Ø ØØØ1 ØØ                    MRD   1 1111 1111  1 1111 1111
-ØØF ØØØ2 3C  INC   A           FETC  1 1111 1111  1 1111 1111
```

```
-ØØE ØØ11 FF                  RFSH  1 1111 1111  1 1111 1111
-ØØD ØØØ3 3C  INC  A          FETC  1 1111 1111  1 1111 1111  .      <
stop display with '.'
,T 2(CR)                      < display 2 samples, beginning at start of
                                history

OFFS ADDR DB  DISASSEMBLY     TYPE  PA8-----PAØ  PB8-----PBØ
-Ø12 ØØØØ 3E  LD   A,Ø        FETC  1 1111 1111  1 1111 1111
-Ø11 ØØ1/ FF                  RFSH  1 1111 1111  1 1111 1111  .
,T 2,-4(CR)              < display 2 samples, beginning 4 samples before
                           breakpoint

OFFS ADDR DB  DISASSEMBLY     TYPE  PA8-----PAØ  PB8-----PBØ
-ØØ4 ØØ16 FF                  RFSH  1 1111 1111  1 1111 1111
-ØØ3 ØØØ8 18  JR   -ØAH       FETC  1 1111 1111  1 1111 1111  .
,T 2,-4,E(CR)    < disable b probe printout

OFFS ADDR DB  DISASSEMBLY     TYPE  PA8-----PAØ
-ØØ4 ØØ16 FF                  RFSH  1 1111 1111
-ØØ3 ØØØ8 18  JR   -ØAH       FETC  1 1111 1111
,T 2,-4,C(CR)            < disable all probe printout

OFFS ADDR DB  DISASSEMBLY     TYPE
-ØØ4 ØØ16 FF                  RFSH
-ØØ3 ØØØ8 18  JR   -ØAH       FETC  .
,T 2,-4,8(CR)            < disable TYPE

OFFS ADDR DB DISASSEMBLY
-ØØ4 ØØ16 FF
-ØØ3 ØØØ8 18  JR   -ØAH  .
,T 2,-4,Ø(CR)            < disable disassembly
```

```
OFFS ADDR DB
 -ØØ4 ØØ16 FF
 -ØØ3 ØØØ8 18.
 ,T (CR)          < format is remembered for later T commands

 OFFS ADDR DB
 -Ø12 ØØØØ 3E
 -Ø11 ØØ1Ø FF
 -Ø1Ø ØØØ1 ØØ
 -ØØF ØØØ2 3C
 -ØØE ØØ11 FF
 -ØØD ØØØ3 3C
 -ØØC ØØ12 FF
 -ØØB ØØØ4 2F
 -ØØA ØØ13 FF
 -ØØ9 ØØØ5 3C
 -ØØ8 ØØ14 FF
 -ØØ7 ØØØ6 3C
 -ØØ6 ØØ15 FF
 -ØØ5 ØØØ7 2F
 -ØØ4 ØØ16 FF
 -ØØ3 ØØØ8 18
 -ØØ2 ØØ17 FF
 -ØØ1 ØØØ9 F6
 +ØØØ ØØØØ 3E
 +ØØ1 ØØ18 FF
 +ØØ2 ØØØ1 ØØ
 +ØØ3 ØØØ2 3C
 +ØØ4 ØØ19 FF
 +ØØ5 ØØØ3 3C
 +ØØ6 ØØ1A FF
 END OF HISTORY

 START OF HISTORY
 -Ø12 ØØØØ 3E
 -Ø11 ØØ1Ø FF
```

```
-Ø1Ø ØØØ1 ØØ
-ØØF ØØØ2 3C
-ØØE ØØ11 FF
-ØØD ØØØ3 3C
-ØØC ØØ12 FF
-ØØB ØØØ4 2F  .         < stop printout with '.'
,B C,A(CR)              < clear the breakpoint
,E (CR)         < resume execution
,T (CR)         < ask for history printout
SAMPLING:  (CR)
OFFS ADDR DB
-3F2 ØØ5Ø FF
-3F1 ØØØ9 F6
-3FØ ØØØØ 3E
-3EF ØØ51 FF
-3EE ØØØ1
-3ED ØØØ2 3C
-3EC ØØ52 FF
-3EB ØØØ3 3C
-3EA ØØ53 FF
-3E9 ØØØ4 2F
-3E8 ØØ54 FF
-3E7 ØØØ5 3C
-3E6 ØØ55 FF
-3E5 ØØØ6 3C
-3E4 ØØ56 FF
-3E3 ØØØ7 2F
-3E2 ØØ57 FF
-3E1 ØØØ8 18
-3EØ ØØ58 FF
-3DF ØØØ9 F6
-3DE ØØØØ 3E
-3DD ØØ59 FF  .         < stop printout with '.'
,
```

3.65 V-VERIFY COMMAND

FORMAT:

```
,V [file](CR)
```

3.66 DESCRIPTION, V COMMAND

The V command is used to verify the contents of Target memory against a binary file on disk. If any locations do not compare, the hexadecimal address, data and expected data are printed.

If a file name has been previously specified with the D, G, or V commands, or when initially loading AIMZ8Ø, then that file name has been retained. When a file name is requested, type only (CR) and the old file name will be used.

3.67 EXAMPLE, V COMMAND

```
,G TEST(CR)                < get the test program
,M Ø(CR)                   < change some locations
ØØØØ 3E(CR)
ØØØ1 ØØ AA(CR)
ØØØ2 3C(CR)
ØØØ3 3C AA(CR)
ØØØ4 2F(CR)
ØØØ5 3C 55(CR)
ØØØ6 3C(CR)
ØØØ7 2F 55(CR)
ØØØ8 18 .
,V (CR)            < verify - expect the errors just introduced
*** ERROR 6D TARGET MEMORY COMPARE ***
ØØØ1 AA ØØ
```

```
ØØØ3 AA 3C
ØØØ5 55 3C
ØØØ7 55 2F.
,G (CR)          < restore the program
,V (CR)          < verify again - no errors
,
,
```

3.68 W-WRITE TO ALTERNATE LUN COMMAND

```
FORMAT:
,W n(CR)                    < Write to LUN n               (1)
,W (CR)                     < Disable Write                (2)
```

3.69 DESCRIPTION, W COMMAND

Format (1) is used to write in parallel any output to the console device to the logical unit number n. The logical unit number n may be assigned to a device or file using the FLP-8Ø DOS A command.

Format (2) is used to disable the writing and close the open file.

3.7Ø EXAMPLE, W COMMAND

```
$A 5,FILE1(CR)         < Assign FILE1 to LUN 5
$AIMZ8Ø(CR)            < Run AIMZ8Ø.BIN
AIMZ8Ø VERSION 1.Ø
,W 5(CR)               < Enable parallel output to LUN 5
,
```

3.71 Z-INITIALIZE COMMAND

Format

```
, Z <CR>
```

3.72 DESCRIPTION, Z COMMAND

The Z command establishes handshaking with the target system and clears all breakpoints. This command is primarily used as a recovery for handshaking errors (ERR 61, 62, 63), and for reinitializing the target after target powerdown (during hardware debug, for example).

3.73 %-TEST MEMORY COMMAND

FORMAT:

```
,% s,f(CR)                      < Test Target memory          (1)
,% s,f,o(CR)                    < Test with options.          (2)
```

3.74 DESCRIPTION, % COMMAND

The % command is used to test Target RAM.
Format (1) will run a sequence of tests of Target RAM starting at address s through f. If a failure occurs, the hexadecimal address, data and expected data will be printed. The following tests are performed.

TEST 1 Write Ø to each location and check.
TEST 2 Write ØFFH to each location and check.
TEST 3 Write Ø55H and ØAAH to each location and check.

Format (2) allows testing to be performed on system, target, or interface memory, if S, T, or I, respectively, is entered as the option. Attempting to test system memory from Ø to 4ØØØH will overwrite the AIMZ8ØA program itself, and, of course, won't work.

3.75 EXAMPLES, % COMMAND

```
,% Ø,FF

TEST Ø1
PASSED

TEST Ø2
PASSED

TEST Ø3
PASSED

,
```

3.76 EXECUTION TIMER OPERATION

The execution timer is implemented as a sixteen bit hardware counter which counts Target clock periods. The execution timer is extended to thirty-two bits with software and is automatically enabled during execution. The execution time is automatically printed when a breakpoint is encountered or when single stepping. The execution time is reset any time execution is started at a particular address. The time is not reset when execution is continued. If the user desires the time to be displayed in microseconds, the Target clock period (:TS) must be initialized (see examples in section 3.15). The AIMZ8ØA program defaults to 1ØØØnS (1.Ø MHz) unless the user changes it. For 1ØØØnS, the number of microseconds is coincidentally the same as the number of T-states.

The execution timer may also be used to cause execution to terminate after a specified number of Target clock periods or Time (Timer breakpoint). To set a timer breakpoint see section 3.15.

3.77 OPERATION WITH MD-SBC1

AIM-Z8ØAE can be used with the Mostek MD-SBC1 single board computer with no modifications or jumbers required. If the Emulation RAM is used, the PROM on the MD-SBC1 must be disabled or removed for the address space emulated.

3.78 OPERATION WITH MDX

AIM-Z8ØAE can also be used with the Mostek MDX-CPU1 in a system with no jumpers or modifications required. If the Emulation RAM is used, the PROM (U7) should be removed and the chip enable for U7 (U5 pin 5) should be enabled for all memory space emulated. AIM-Z8ØAE can also be used with the Mostek MDX-CPU2. If Emulation RAM is used to emulate PROM, the PROM must be removed and the PROM chip enable active for the emulated memory space to prevent Bus conflict.

3.79 OPERATION WITH OEM-8Ø

AIM-Z8ØAE can also be used with the Mostek OEM-8Ø single board computer. Only one modification is required for this system. Memory in the OEM-8Ø system must be disabled for address space emulated with the Emulation RAM.

To insure proper operation with the OEM-8Ø (SDB-8Ø), a jumper wire, 24 gauge or larger, should be installed on the circuit side of the OEM-8Ø between U63 pin 29, and the ground bus feedthrough located between U7Ø pin 9 and U71 pin 8. That feedthrough actually connects to U71 pin 8 on the component side of the board. This jumper insures adequate ground for the buffer box.

APPENDIX A

FACTORY NOTICES

FACTORY REPAIR SERVICE

In the event that difficulty is encountered with this unit, it may be returned directly to MOSTEK for repair. This service will be provided free of charge if the unit is returned within the warranty period. However, units which have been modified or abused in any way will not be accepted for service, or will be repaired at the owner's expense. When returning a circuit board, place it inside the conductive plastic bag in which it was delivered to protect the MOS devices from electrostatic discharge. THE CIRCUIT BOARD MUST NEVER BE PLACED IN CONTACT WITH STYROFOAM MATERIAL. Enclose a letter containing the following information with the returned circuit board.

Name, address, and phone number of purchaser
Date and place of purchase
Brief description of the difficulty

Mail a copy of this letter SEPARATELY to:

In USA:

MOSTEK Corporation
Microcomputer Service Manager
1215 West Crosby Road
Carrollton, TX 75006

OUTSIDE USA:

Please address the letter and board to the Mostek office or representtive in your country.

Securely package and mail the circuit board, prepaid and insured, to the same address.

LIMITED WARRANTY

MOSTEK warrants this product against defective materials and workmanship for a period of 90 days. This warranty does not apply to any product that has been subjected to misuse, accident, improper installation, improper application, or improper operation, nor does it apply to any product that has been repaired or altered by other than an authorized factory representative.

There are no warranties which extend beyond those herein specifically given.

NOTICE

The antistatic bag is provided for shipment of the Mostek PC boards to prevent damage to the components due to electrostatic discharge.

Failure to use the bag in shipment will **VOID** the warranty.

APPENDIX B

SPECIFICATIONS

SPECIFICATIONS

Operating Temperature Range Ø°C to +5Ø°C

Target Power Supply Requirements (typical)

+5V 5% @ 5ØØma

System Power Supply Requirements (typical)

+5V 5% @ 2.5A

+12V 5% @ 1ØØmA

-12V 1Ø% @ 1ØmA

Interface - MATRIX and SYS-8ØF compatible

Operating Frequency - 5ØØKHz to 4MHz (Z8Ø PHI clock)

Target Interface - All signals meet the specifications for the MK388Ø-4 (Z8ØA CPU) with the following exceptions:

1. The ouptut low voltage is Ø.5 V Max at 1.8 mA for the ADDRESS, DATA, $\overline{\text{IORQ}}$, $\overline{\text{RFSH}}$, $\overline{\text{HALT}}$, and $\overline{\text{BUSAK}}$ signals.

2. The input low current is 4ØØua for the PHI clock, $\overline{\text{RESET}}$, $\overline{\text{INT}}$, $\overline{\text{NMI}}$, and DATA signals.

3. The input high current is 2Øma for the PHI clock, $\overline{\text{RESET}}$, $\overline{\text{INT}}$, $\overline{\text{NMI}}$, and DATA signals.

4. The signals $\overline{\text{M1}}$, $\overline{\text{MREQ}}$, $\overline{\text{RD}}$, and $\overline{\text{WR}}$ have a maximum of 25 ns added propagation delay.

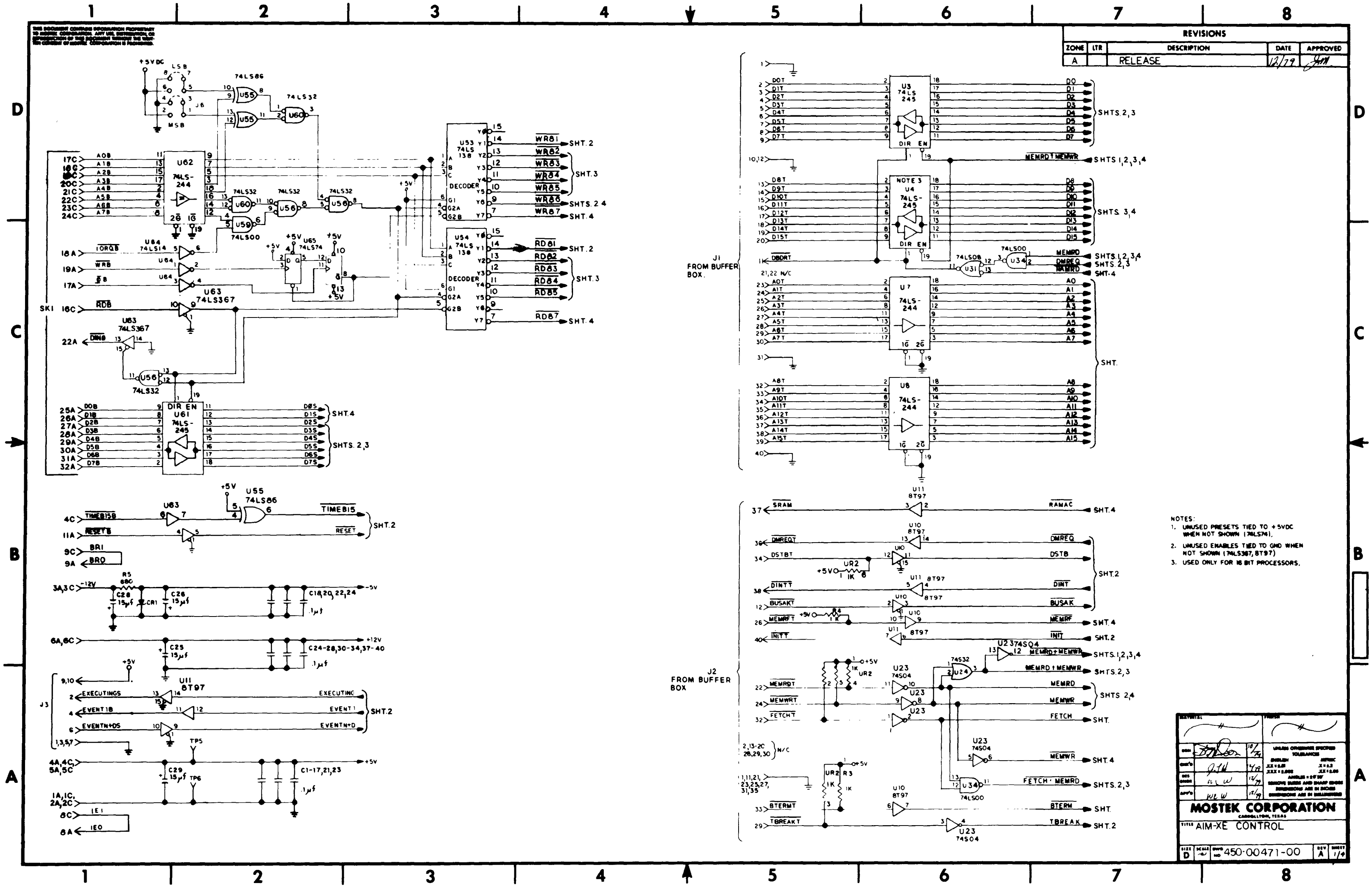
5. The input signals $\overline{\text{RESET}}$, $\overline{\text{INT}}$ and $\overline{\text{NMI}}$ have maximum of 45 ns added propagation delay.

APPENDIX C

SCHEMATIC DIAGRAMS

PARTS PLACEMENT DIAGRAMS

PARTS LISTS



NOTES:
1. UNUSED PRESETS TIED TO +5VDC WHEN NOT SHOWN (74LS74).
2. UNUSED ENABLES TIED TO GND WHEN NOT SHOWN (74LS367, 8T97)
3. USED ONLY FOR 16 BIT PROCESSORS.

REVISIONS

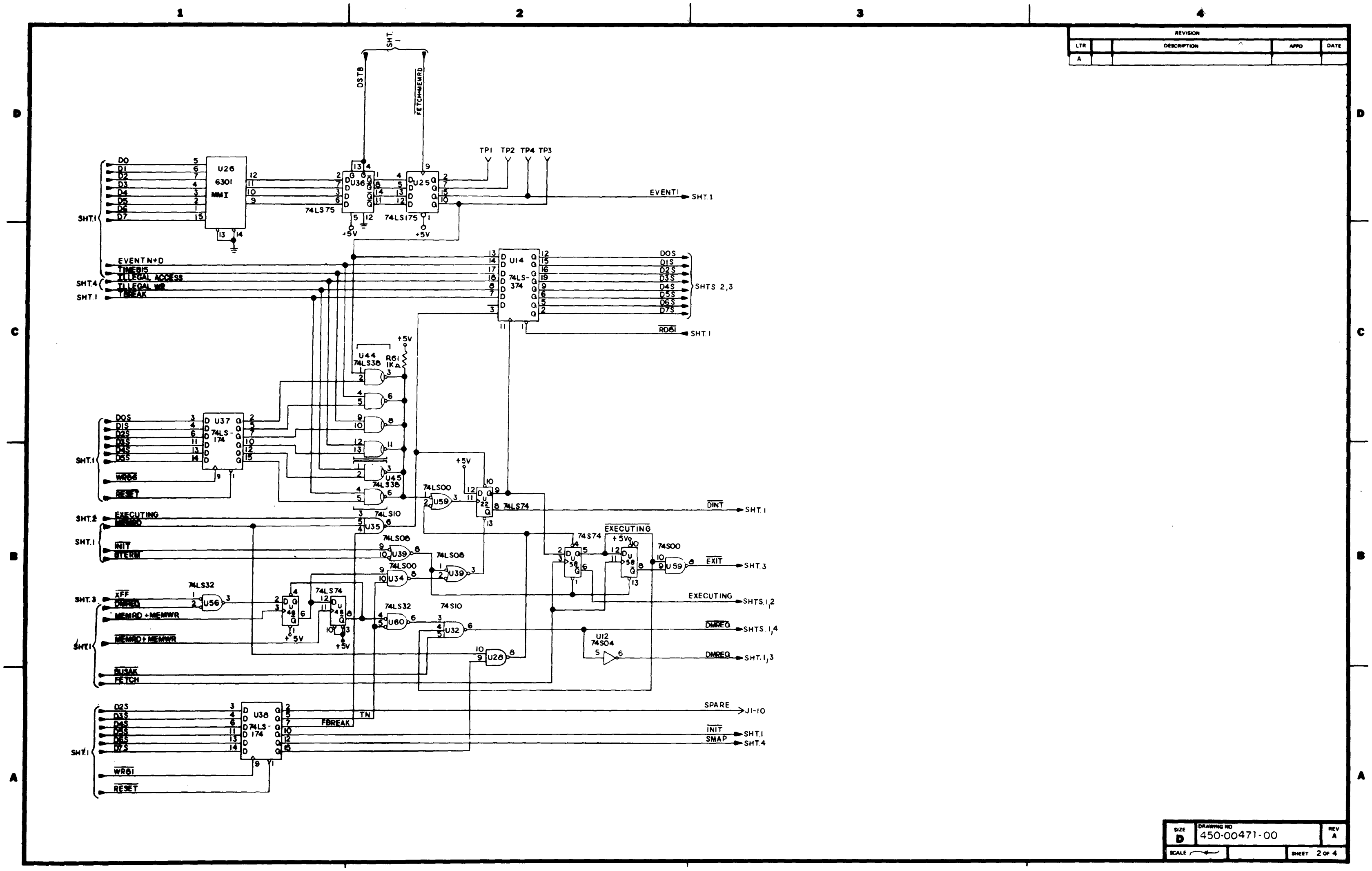
ZONE	LTR	DESCRIPTION	DATE	APPROVED
A		RELEASE	12/79	

MOSTEK CORPORATION

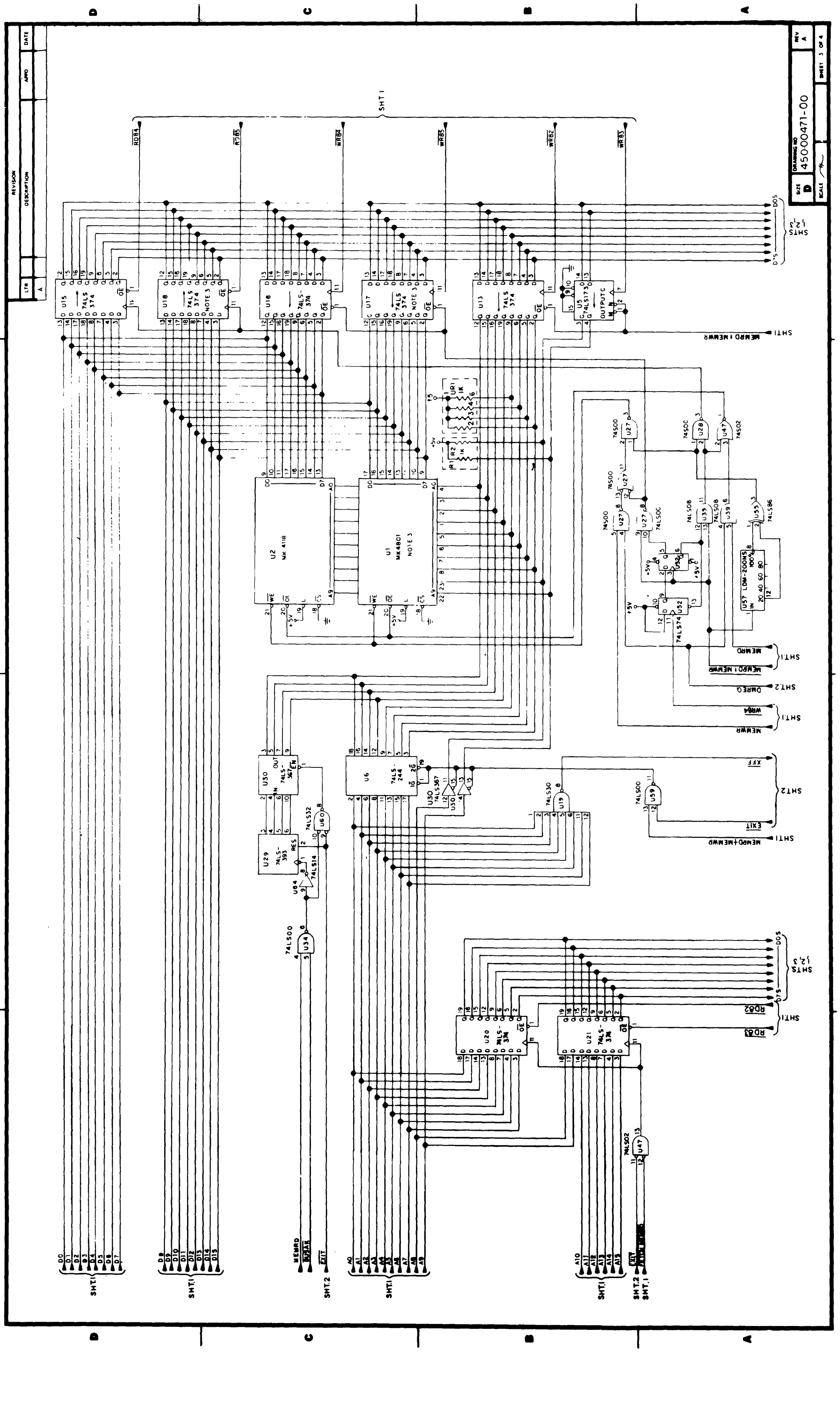
TITLE AIM-XE CONTROL

SIZE	SCALE	DWG NO	REV	SHEET
D	—	450-00471-00	A	1/4

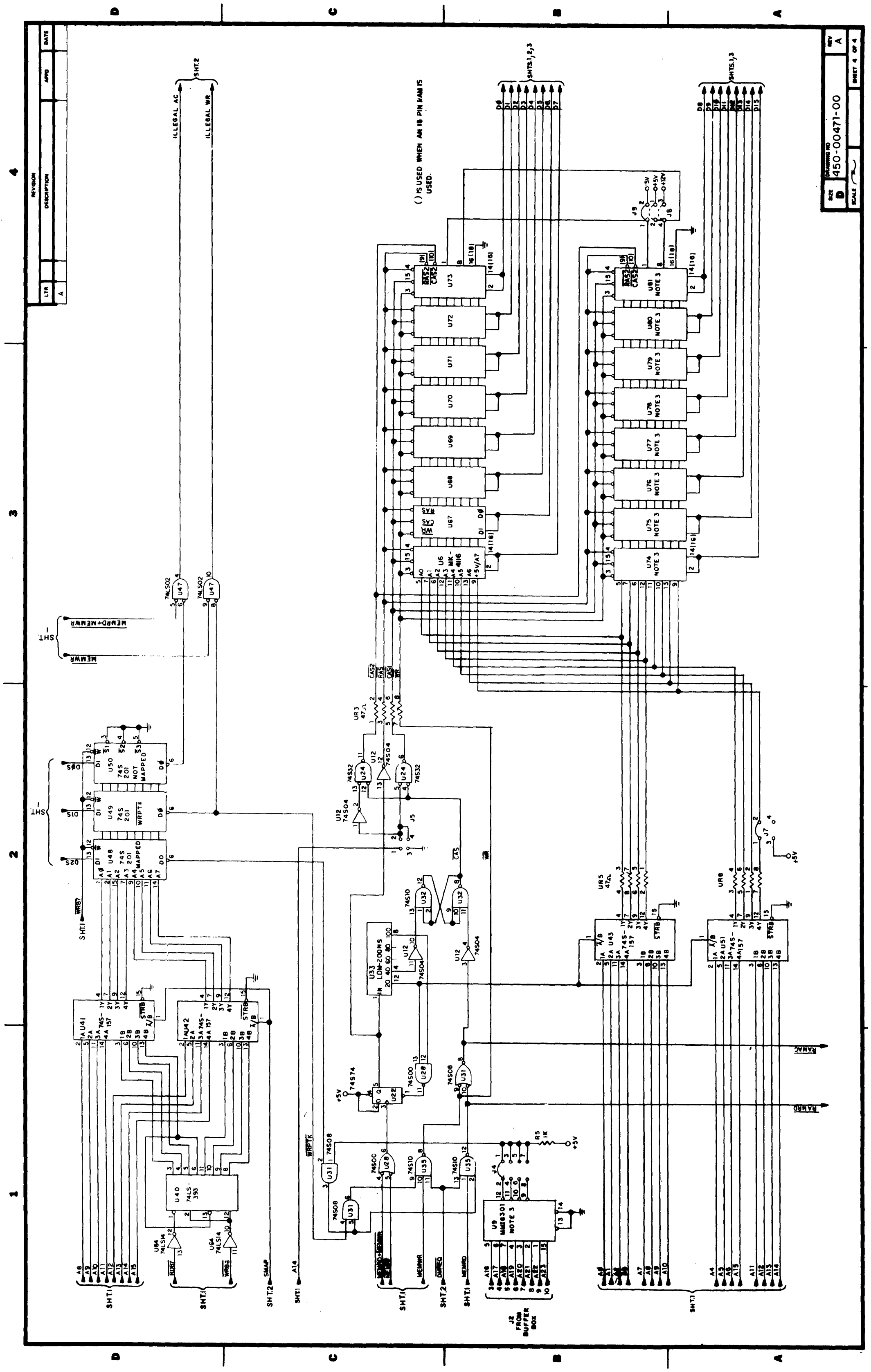
SCHEMATIC


1
2
3
4
D
C
B
A
REVISION
LTR
DESCRIPTION
APPD
DATE
A
SHT. 1
DSTB
FETCH+MEMRD
D0
D1
D2
D3
D4
D5
D6
D7
SHT.1
U26
6301
MMI
U36
74LS75
U25
74LS175
+5V
TP1
TP2
TP4
TP3
EVENT1
SHT. 1
EVENT N+D
TIMEB15
ILLEGAL ACCESS
ILLEGAL WR
TBREAK
SHT.4
SHT.1
U14
74LS-374
D0S
D1S
D2S
D3S
D4S
D5S
D6S
D7S
SHTS 2,3
RD81
SHT. 1
U44
74LS38
R61
1KΩ
+5V
U37
74LS-174
WR86
RESET
U45
74LS38
74LS00
U59
74LS74
U22
DINT
SHT. 1
SHT.2
EXECUTING
MEMRD
74LS10
U35
INIT
STERM
74LS08
U39
74LS00
U34
74S74
U58
EXECUTING
74S00
U59
EXIT
SHT.3
SHT.3
XFF
DMREQ
74LS32
U56
74LS74
U48
MEMRD+MEMWR
MEMRD+MEMWR
74LS32
U60
74S10
U32
EXECUTING
SHTS 1,2
DMREQ
SHTS 1,4
U12
74S04
U28
DMREQ
SHT.1,3
BUSAK
FETCH
D2S
D3S
D4S
D5S
D6S
D7S
U38
74LS-174
TN
FBREAK
SPARE
J1-10
INIT
SHT.1
SMAP
SHT.4
WR81
RESET
SIZE
D
DRAWING NO
450-00471-00
REV
A
SCALE
SHEET 2 OF 4

SCHEMATIC

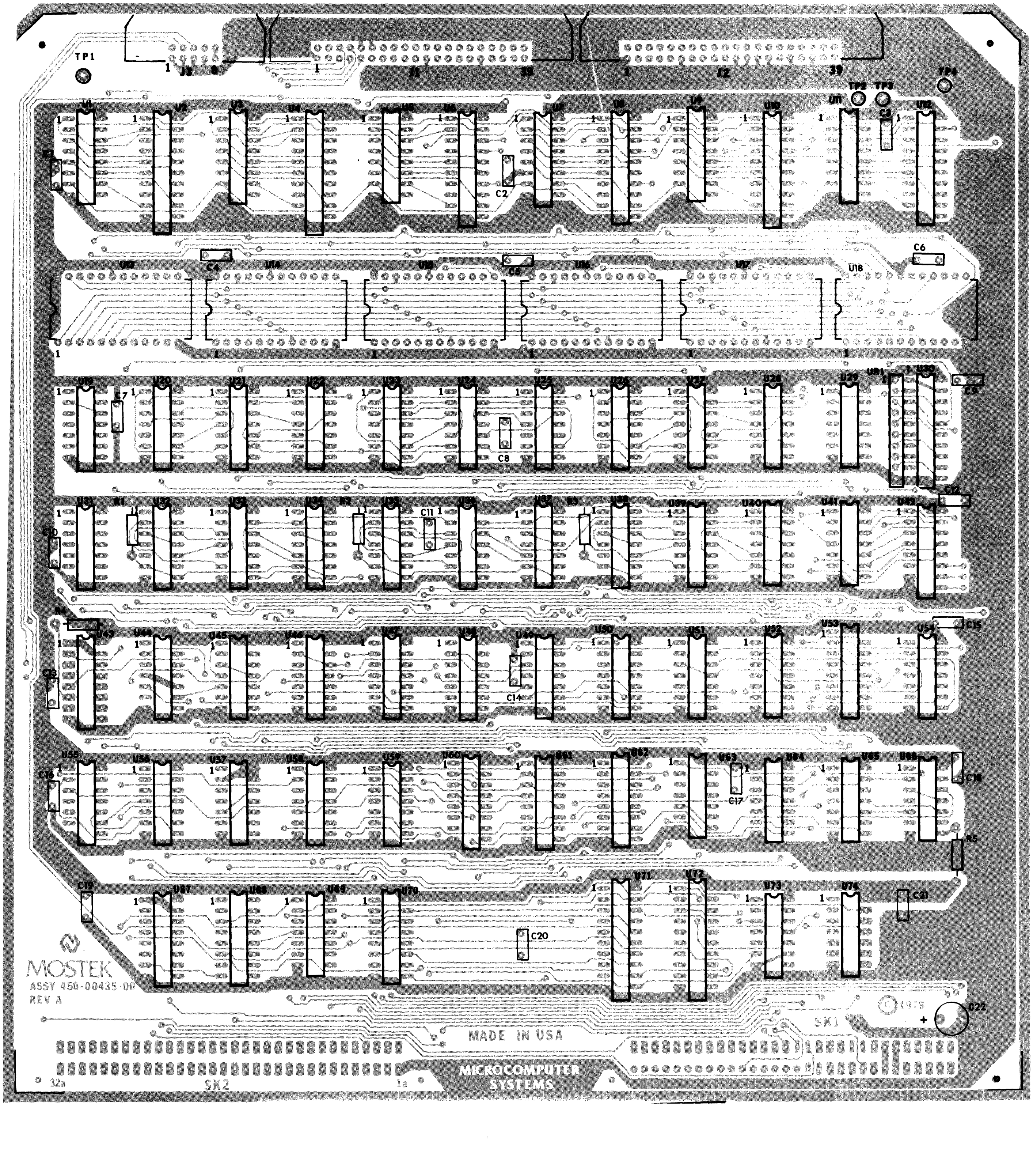

D
C
B
A
4
3
2
1
LTR
DESCRIPTION
REVISION
APPD
DATE
SIZE
D
DRAWING NO
450-00471-00
SCALE
SHEET 3 OF 4
REV
A
SHT 1
RD84
RD85
WR84
WR85
WR82
WR83
DOS
D7S
SHTS 1,2,3
U15 74LS 374
U18 74LS 374 NOTE 3
U16 74LS-374
U17 74LS 374 NOTE 3
U13 74LS 374
U5 74LS173
OUTPUTC
MEMRD+MEMWR
SHT1
UR1 1K
R2 1K
+5V
U2 MK 4118
U1 MK4801 NOTE 3
U27 74S00
U28 74S0C
U47 74S02
U39 74LS08
U55 74LS86
U52 74LS74
U57 LDM-200NS 100% 20 40 60 80
MEMRD
DMREQ
SHT2
WR84
MEMWR
XFF
EXIT
U6 74LS 244
U30 74LS-367
U29 74LS-393
U60 74LS32
U64 74LS14
U34 74LS00
U19 74LS30
U59 74LS00
U20 74LS-374
U21 74LS-374
U47 74LS02
RD82
RD83
D0
D1
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
D13
D14
D15
MEMRD
NOMAR
EXIT
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15
SHT.2
SHT.1

SCHEMATIC


REVISION
LTR
DESCRIPTION
APPD
DATE
A
ILLEGAL AC
ILLEGAL WR
SHT2
( ) IS USED WHEN AN 18 PIN RAM IS USED.
SHTS.1,2,3
D0 D1 D2 D3 D4 D5 D6 D7
SHTS.1,3
D8 D9 D10 D11 D12 D13 D14 D15
U73 U72 U71 U70 U69 U68 U67
U6 MK-4116
U81 NOTE 3
U80 NOTE 3
U79 NOTE 3
U78 NOTE 3
U77 NOTE 3
U76 NOTE 3
U75 NOTE 3
U74 NOTE 3
RAS2 CAS2
J9 J8 +5V -5V +12V
74LS02 U47
MEMRD+MEMWR
MEMWR
SHT1
U48 74S 201 MAPPED
U49 74S 201 WRPTK
U50 74S 201 NOT MAPPED
DS0 DS1 DS2
WRT87
UR3 47Ω
CAS2 RAS CAS1 WR
U24 74S32
U12 74S04
J5
CAS
WR
U32 74S10
U33 LDM-200NS
U22 74S74
U28 74S00
U31 74S08
U35 74S10
U9 MM8301 NOTE 3
J4
R5 1K +5V
UR5 47Ω
UR8
J7
U43 74S157
U51 74S157
U41 74S157
U42 74S157
U40 74LS393
U64 74LS14
RAMAC
RAMRD
A8 A9 A10 A11 A12 A13 A14 A15
SHT.1
PORZ
WRT86
SHT.2 SMAP
SHT1 A14
WRPTK
SHT.1 MEMWR
SHT.2 DMREQ
SHT.1 MEMRD
A16 A17 A18 A19 A20 A21 A22 A23
J2 FROM BUFFER BOX
A0 A1 A2 A3 A7 A8 A9 A10
A4 A5 A6 A15 A11 A12 A13 A14
SIZE D
DRAWING NO 450-00471-00
REV A
SCALE
SHEET 4 OF 4

PARTS PLACEMENT DIAGRAM

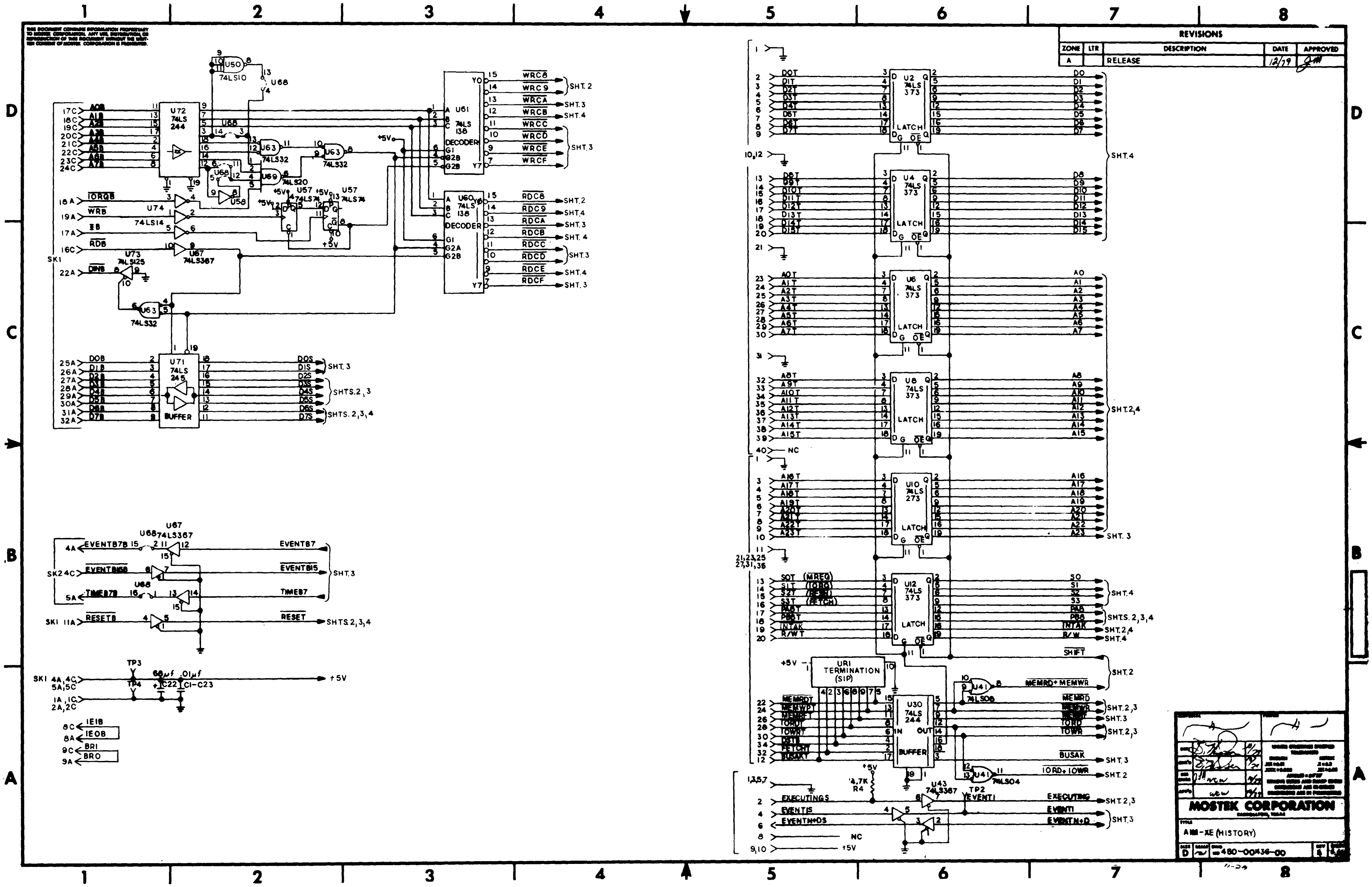
AIM-Z8Ø AE CONTROL MODULE



PARTS LIST

AIM#Z80AE CONTROL MODULE

PART NO.	QTY	DESCRIPTION	REFERENCE DESIGNATOR	USED ON
########	#####	AIM-XE CONTROL	AA:450-00470-00	XXXXX
4610173	1	FAB 450-00469-00	AZ:AIM-XE CONTROL	XXXXX
0000001		SCH 450-00471-00	AZ:AIM-XE CONTROL	XXXXX
0000001		ASSY 450-00470-00	AZ:AIM-XE CONTROL	XXXXX
4150111	46	CAPACITOR,.1UF	C 1-25	XXXXX
4150140	4	CAPACITOR,15UF	C26-29	XXXXX
4480047	1	DIODE,IN751 5.1V	CR1	XXXXX
4470073	6	RESISTOR,1K	R1-6	XXXXX
4470059	1	RESISTOR,680	R7	XXXXX
4210057	2	CONNECTOR,EURO	SK1,SK2	XXXXX
4280007	6	STAKE PIN	TP1-6	XXXXX
XXXXXXX		NOT USED	U 1,4,9,17,13,74-81	
4313203	2	IC, 8T97	U 10,11	XXXXX
4313700	1	IC, MK4118-2	U 2	XXXXX
4313508	2	IC, 74LS245	U 3,4,61	XXXXX
4313583	1	IC, 74LS173	U 5	XXXXX
4313507	4	IC, 74LS244	U 6,7,8,62	XXXXX
4313285	2	IC, 74S04	U12,23	XXXXX
4313509	8	IC, 74LS374	U13,14,15,16,,20,21	XXXXX
4313410	1	IC, 74LS30	U19	XXXXX
4313265	2	IC, 74S74	U22,58	XXXXX
4313267	1	IC, 74S32	U24	XXXXX
4313306	2	IC, 74LS175	U25,36	XXXXX
4313731	1	IC, PROM,6301,MK6283	U26	XXXXX
4313284	1	IC, 74S00	U27	XXXXX
4313287	3	IC, 74LS00	U28,34,59	XXXXX
4313335	2	IC, 74LS393	U29,40	XXXXX
4313464	2	IC, 74LS367	U30,63	XXXXX
4313289	2	IC, 74LS08	U31,39	XXXXX
4313329	2	IC, 74S10	U32,U35	XXXXX
4313513	2	IC, DELAY,LDM-200	U33,57	XXXXX
4313305	2	IC, 74LS174	U37,38	XXXXX
4313264	4	IC, 74S157	U41,42,43,51	XXXXX
4313412	2	IC, 74LS38	U44,45	XXXXX
4313413	3	IC, 74LS74	U46,52,65	XXXXX
4313300	1	IC, 74LS02	U47	XXXXX
4313494	3	IC, 74S201	U48,49,50	XXXXX
4313296	2	IC, 74LS138	U53,54	XXXXX
4313417	1	IC, 74LS86	U55	XXXXX
4313411	2	IC, 74LS32	U56,60	XXXXX
4313291	1	IC, 74LS14	U64	XXXXX
4313392	16	IC, MK4116-2	U66-81	XXXXX
4470178	2	SIP,6PIN,5RES,1K	UR1,UR2	XXXXX
4470273	3	SIP,8PIN,4RES,47	UR3,5,6	XXXXX
4620017	17	SOCKET 16 PIN	XXX	XXXXX
4620018	2	SOCKET 24 PIN	XXX	XXXXX
4210244	7	MINI JUMPER	Z1-7	XXXXX
4210282	2	CONN HDR RA 40PIN W/EJECT	J1-2	XXXXX
4210231	1	CONN HDR RA 10PIN W/EJECT	J3	XXXXX



REVISIONS				
ZONE	LTR	DESCRIPTION	DATE	APPROVED
A		RELEASE	12/79	JM

MOSTEK CORPORATION

CARROLLTON, TEXAS

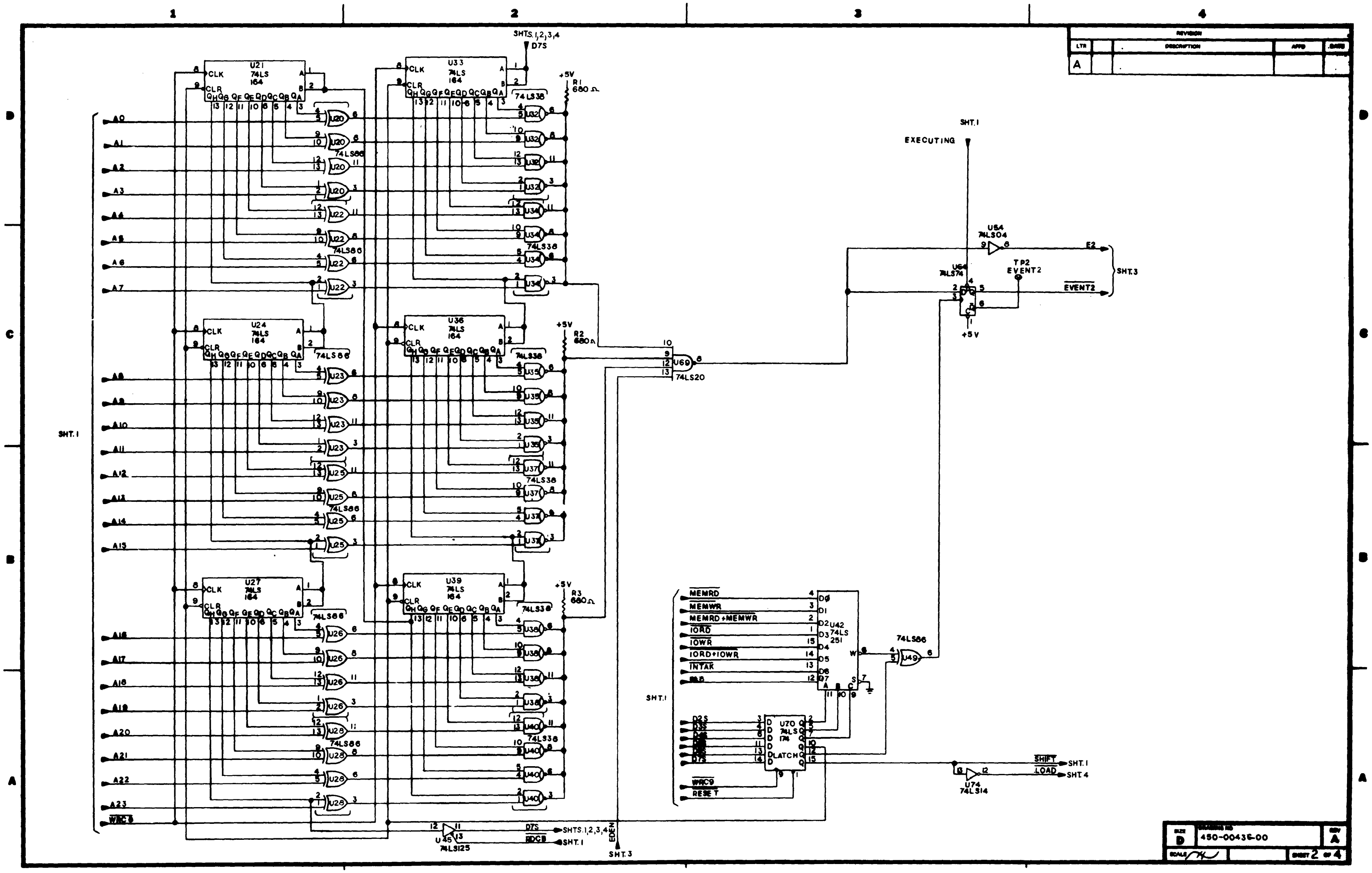
TITLE

AIM-XE (HISTORY)

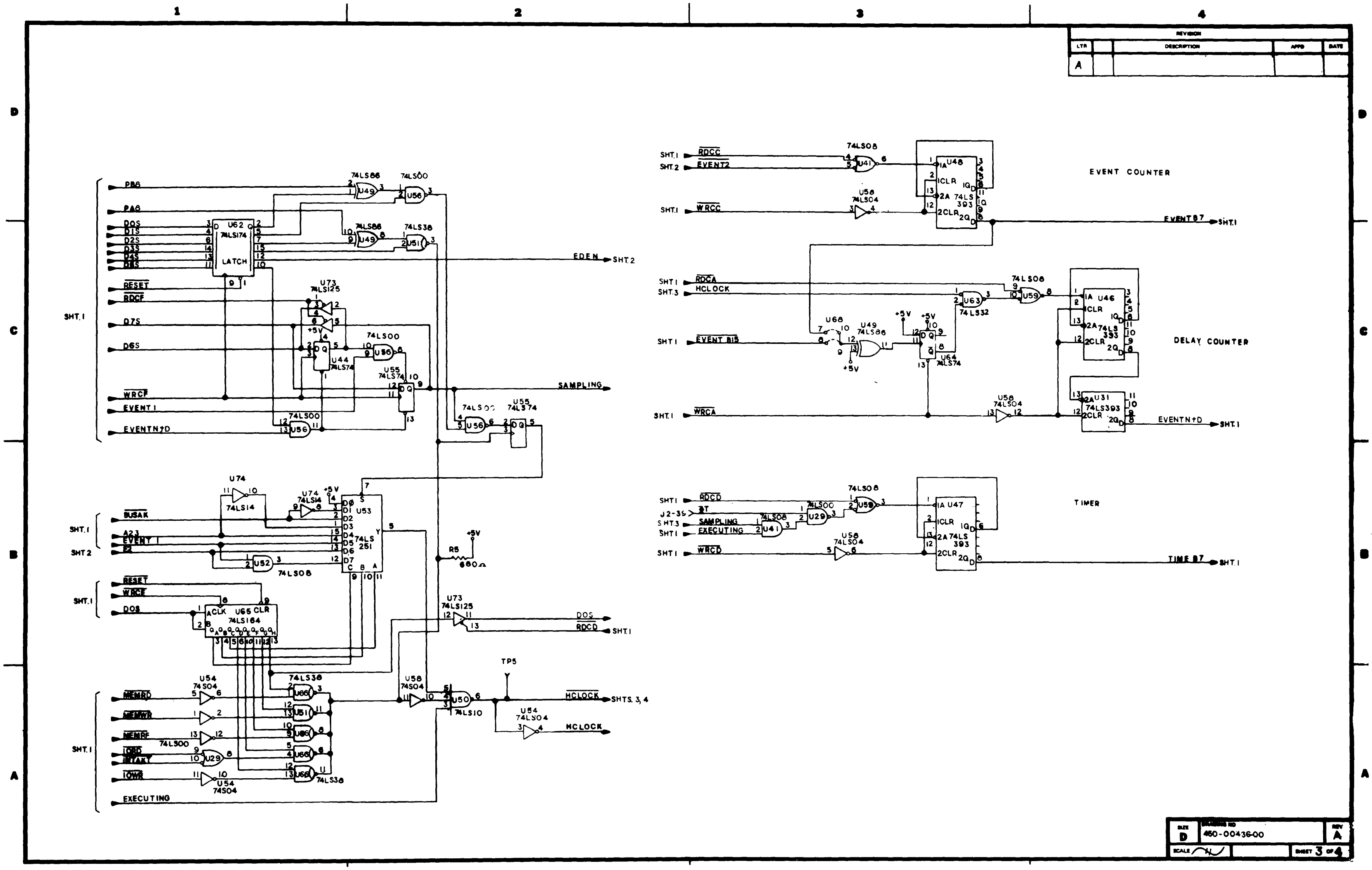
SIZE D SCALE ~ DWG NO 480-00436-00

SCHEMATIC

C#7


1
2
3
4
D
C
B
A
REVISION
LTR
DESCRIPTION
APPD
DATE
A
SHT.1
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15
A16
A17
A18
A19
A20
A21
A22
A23
WRC8
U21 74LS 164
U24 74LS 164
U27 74LS 164
U33 74LS 164
U36 74LS 164
U39 74LS 164
CLK
CLR
QH QG QF QE QD QC QB QA
74LS86
74LS38
U20
U22
U23
U25
U26
U28
U32
U34
U35
U37
U38
U40
SHTS.1,2,3,4
D7S
+5V
R1 680Ω
R2 680Ω
R3 680Ω
U69 74LS20
EDEN
SHT.3
U45 74LS125
D7S
SHTS.1,2,3,4
RDC8
SHT.1
EXECUTING
U64 74LS04
E2
U64 74LS74
TP2 EVENT2
EVENT2
SHT.3
+5V
MEMRD
MEMWR
MEMRD+MEMWR
IORD
IOWR
IORD+IOWR
INTAK
D0 D1 D2 D3 D4 D5 D6 D7
U42 74LS 251
W
S
A B C
74LS86
U49
D2S
D3S
D4S
D5S
D6S
D7S
WRC9
RESET
U70 74LS 174 LATCH
U74 74LS14
SHIFT
SHT.1
LOAD
SHT.4
SIZE
D
DRAWING NO
450-0043Ƨ-00
REV
A
SCALE
SHEET 2 OF 4

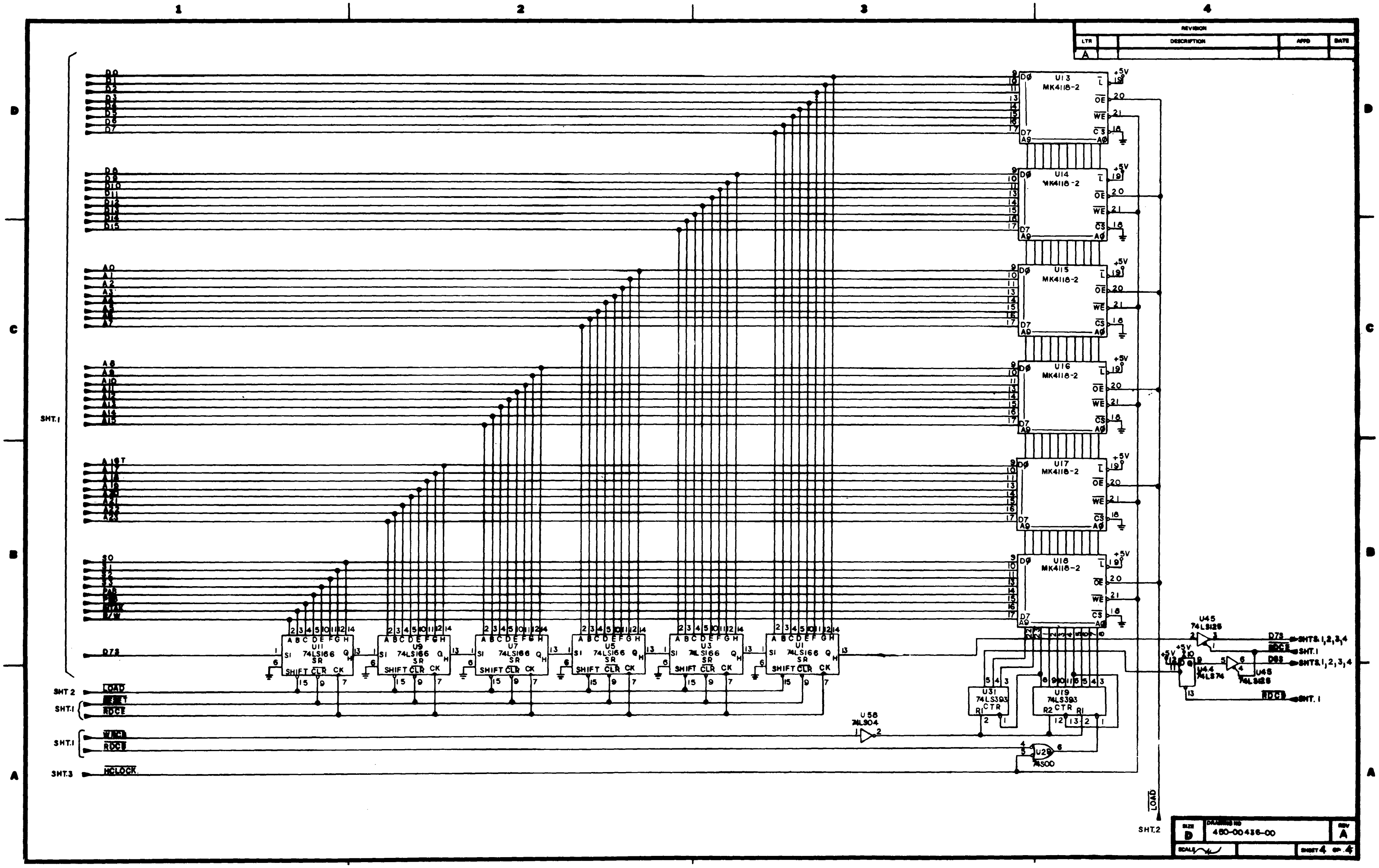
SCHEMATIC



1
2
3
4
D
C
B
A
REVISION
LTR
DESCRIPTION
APPD
DATE
A
SHT. I
PB8
PA8
D0S
D1S
D2S
D3S
D4S
D5S
RESET
RDCF
D7S
D6S
WRCF
EVENT I
EVENTN+D
U62
74LS174
LATCH
74LS86
U49
74LS00
U56
74LS38
U51
EDEN
SHT. 2
U73
74LS125
+5V
U44
74LS74
U55
74LS74
SAMPLING
U74
74LS14
+5V
U53
74LS 251
BUSAK
A23
EVENT I
E2
SHT 2
U52
74LS08
RESET
WRCE
D0S
U65
74LS164
CLK
CLR
R5
680Ω
DOS
RDCD
SHT. I
TP5
MEMRD
MEMWR
MEMRF
IORD
INTAKT
IOWR
EXECUTING
U54
74S04
74LS00
U29
U66
U51
74LS38
U58
74S04
U50
74LS10
HCLOCK
SHTS. 3, 4
74LS04
HCLOCK
SHT. I
RDCC
SHT. 2
EVENT2
74LS08
U41
U48
74LS 393
WRCC
U58
74LS04
EVENT COUNTER
EVENTB7
SHT. I
RDCA
SHT. 3
HCLOCK
U63
74LS32
74LS08
U59
U46
74LS 393
DELAY COUNTER
U68
U49
74LS86
EVENT BIS
U64
74LS74
WRCA
U58
74LS04
U31
74LS393
EVENTN+D
RDCD
J2-39
BT
SAMPLING
EXECUTING
WRCD
74LS00
U29
74LS08
U41
74LS08
U59
U47
74LS 393
TIMER
TIMEB7
SIZE
D
DRAWING NO
460-00436-00
REV
A
SCALE
SHEET 3 OF 4

SCHEMATIC

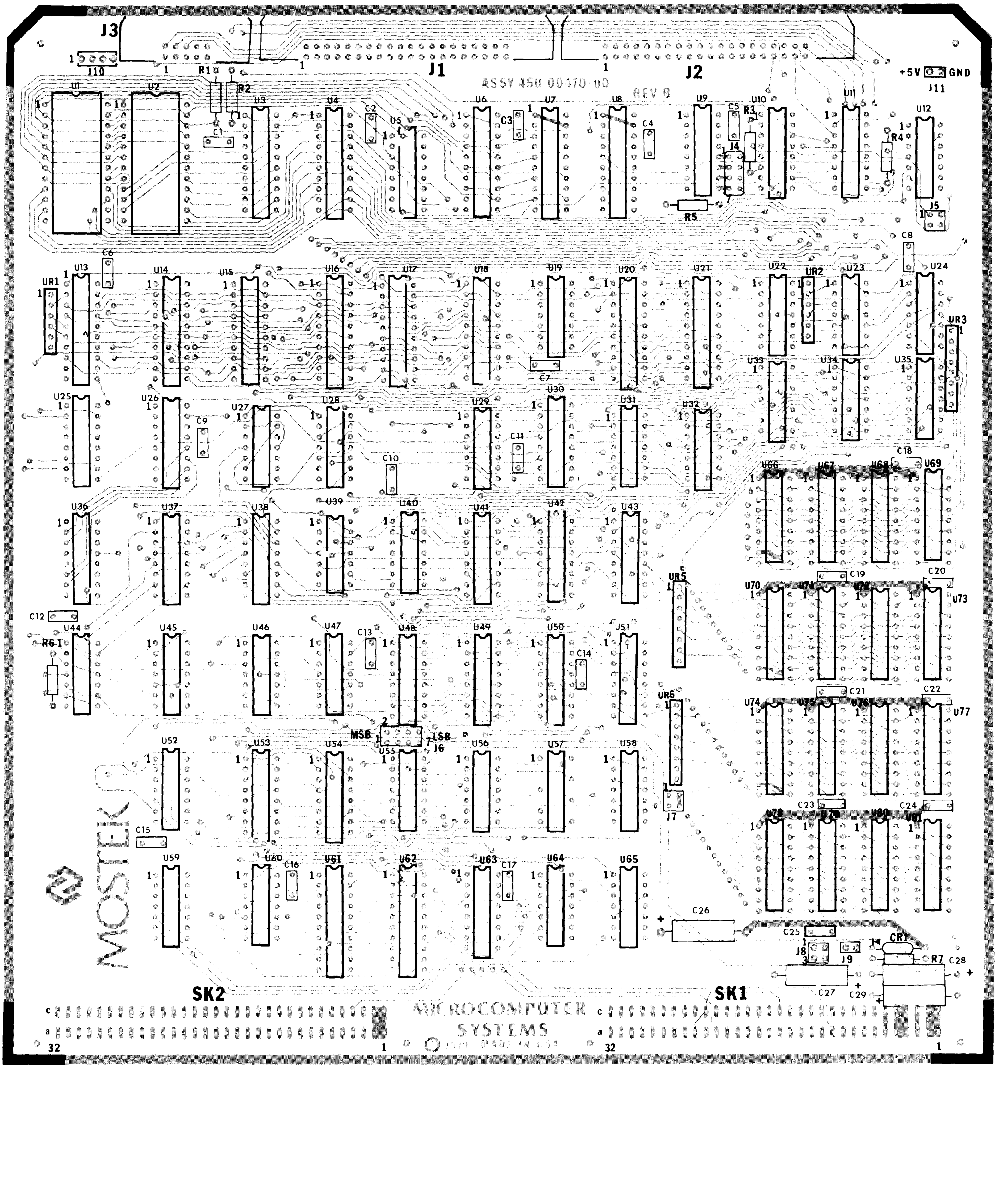
C#10



SCHEMATIC

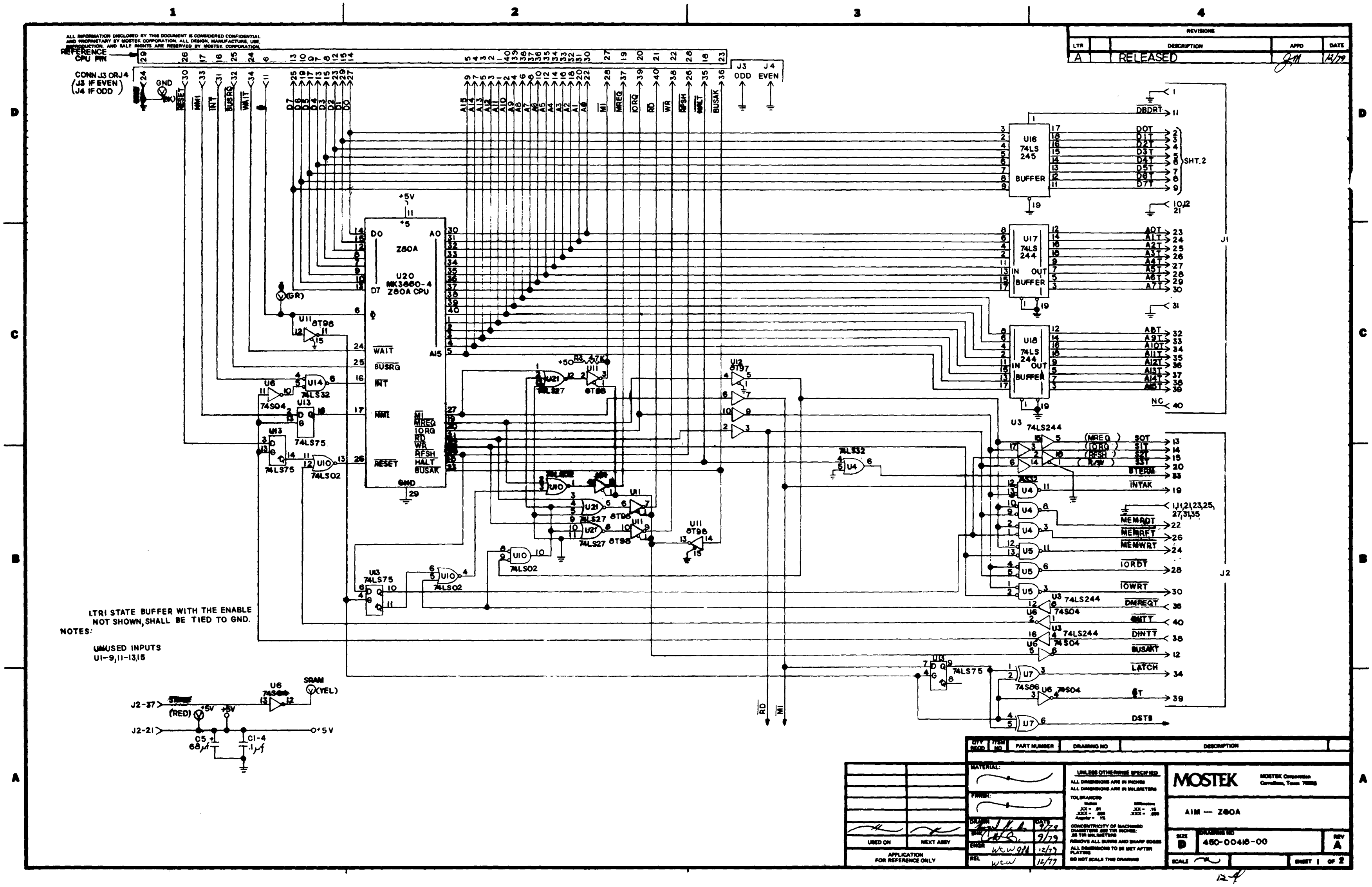
PARTS PLACEMENT

HISTORY MODULE



PARTS LIST
HISTORY MODULE

PART NO.	QTY	DESCRIPTION	REFERENCE DESIGNATOR	USED ON
########	####	AIM-XE HISTORY	AA: 450-00435-00	AIMXHE
4610165	1	FAB 450-00434000 REV	AZ: AIM-XE HISTORY	AIMXHE
0000002		SCH 450-00436-00 REV	AZ: AIM-XE HISTORY	AIMXHE
0000001		ASSY 450-00435-00 REV	AZ: AIM-XE HISTORY	AIMXHE
4150111	21	CAPACITOR,.1MF	C1-C21	AIMXHE
4150114	1	CAPACITOR,68UF	C22	AIMXHE
4210133	2	HEADER,40PIN,RT ANGLE	J1,2 *SEE NOTE 1	AIMXHE
4210217	1	HEADER,10PIN,RT ANGLE	J3	AIMXHE
4470069	4	RESISTOR,680	R1-3,R5	AIMXHE
4470089	1	RESISTOR,4.7K	R4	AIMXHE
4210057	2	CONNECTOR,EURO	SK1,SK2	AIMXHE
4280007	4	STAKE PIN	TP1-4	AIMXHE
4313599	6.	IC, 74LS166	U 1,3,5,7,9,11	AIMXHE
4313344	6	IC, 74LS373	U 2,4,6,8,10,12	AIMXHE
4313700	6	IC, MK4118-2	U13,14,15,16,17,18	AIMXHE
4313335	5	IC, 74LS393	U19,31,46,47,48	AIMXHE
4313417	7	IC, 74LS86	U20,22,23,25,26,28,49	AIMXHE
4313598	7	IC, 74LS164	U21,24,27,33,36,39,65	AIMXHE
4313284	2	IC, 74LS00	U29,56,	AIMXHE
4313507	2	IC, 74LS244	U30,72	AIMXHE
4313412	8	IC, 74LS38	U32,34,35,37,38,40,51,66	AIMXHE
4313289	3	IC, 74LS08	U41,52,59,	AIMXHE
4313703	2	IC, 74LS151	U42,53	AIMXHE
4313464	2	IC, 74LS367	U43,67	AIMXHE
4313413	4	IC, 74LS74	U44,55,57,64	AIMXHE
4313691	2	IC, 74LS125	U45,73	AIMXHE
4313501	1	IC, 74LS10	U50	AIMXHE
4313285	1	IC, 74S04	U54	AIMXHE
4313288	1	IC, 74LS04	U58	AIMXHE
4313296	2	IC, 74LS138	U60,61	AIMXHE
4313305	2	IC, 74LS174	U62,70	AIMXHE
4313411	1	IC, 74LS32	U63	AIMXHE
4640009	1	SW, DIP,8 POS	U68	AIMXHE
4313292	1	IC, 74LS20	U69	AIMXHE
4313508	1	IC, 74LS245	U71	AIMXHE
4313291	1	IC, 74LS14	U74	AIMXHE
4470174	1	SIP,10PIN,9RES,1K	UR1	AIMXHE
4620018	6	SOCKET 24 PIN	X13-18	AIMXHE
4210244	6	MINI JUMPER	Z1-6	AIMXHE
0000003		NOTE 1	Z: CUT PIN 22 OF J1	AIMXHE
0000004		NOTE 1	Z: CUT PIN 2 OF J2	AIMXHE



ALL INFORMATION DISCLOSED BY THIS DOCUMENT IS CONSIDERED CONFIDENTIAL AND PROPRIETARY BY MOSTEK CORPORATION. ALL DESIGN, MANUFACTURE, USE, REPRODUCTION, AND SALE RIGHTS ARE RESERVED BY MOSTEK CORPORATION.

REVISIONS			
LTR	DESCRIPTION	APPD	DATE
A	RELEASED	JM	12/79

NOTES:

1.TRI STATE BUFFER WITH THE ENABLE NOT SHOWN, SHALL BE TIED TO GND.

UNUSED INPUTS
U1-9,11-13,15

QTY REQD	ITEM NO	PART NUMBER	DRAWING NO	DESCRIPTION

MATERIAL:

FINISH:

DRAWN	DATE 9/79
CHK	9/79
ENGR wcw 9/11	12/79
REL wcw	12/79

UNLESS OTHERWISE SPECIFIED
ALL DIMENSIONS ARE IN INCHES
ALL DIMENSIONS ARE IN MILLIMETERS
TOLERANCES:
Inches .XX = .01 .XXX = .005 Angular = 1°
Millimeters .XX = .15 .XXX = .050
CONCENTRICITY OF MACHINED DIAMETERS .002 TIR INCHES: .05 TIR MILLIMETERS
REMOVE ALL BURRS AND SHARP EDGES
ALL DIMENSIONS TO BE MET AFTER PLATING
DO NOT SCALE THIS DRAWING

USED ON | NEXT ASSY

APPLICATION FOR REFERENCE ONLY

MOSTEK

MOSTEK Corporation
Carrollton, Texas 75006

AIM — Z80A

SIZE D | DRAWING NO 450-00416-00 | REV A

SCALE | SHEET 1 OF 2

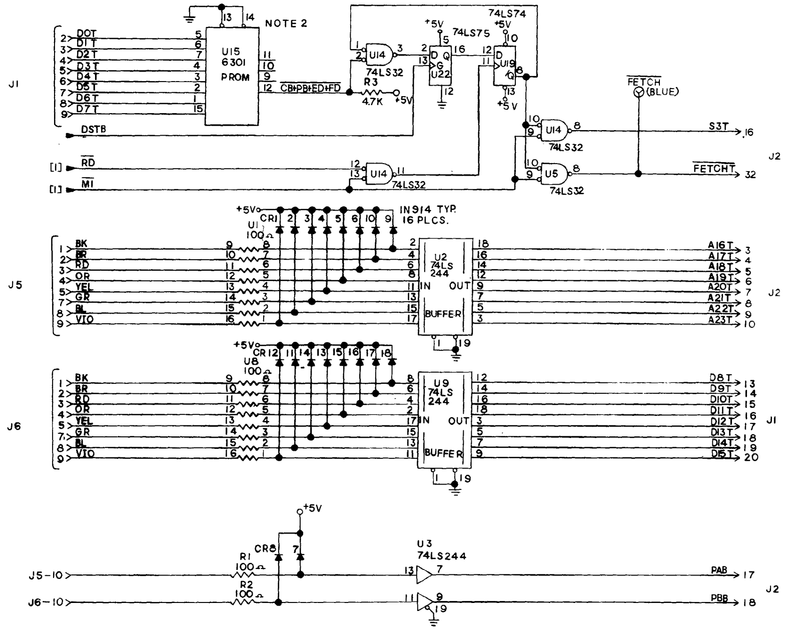
12-4

SCHEMATIC

C-13

C#14

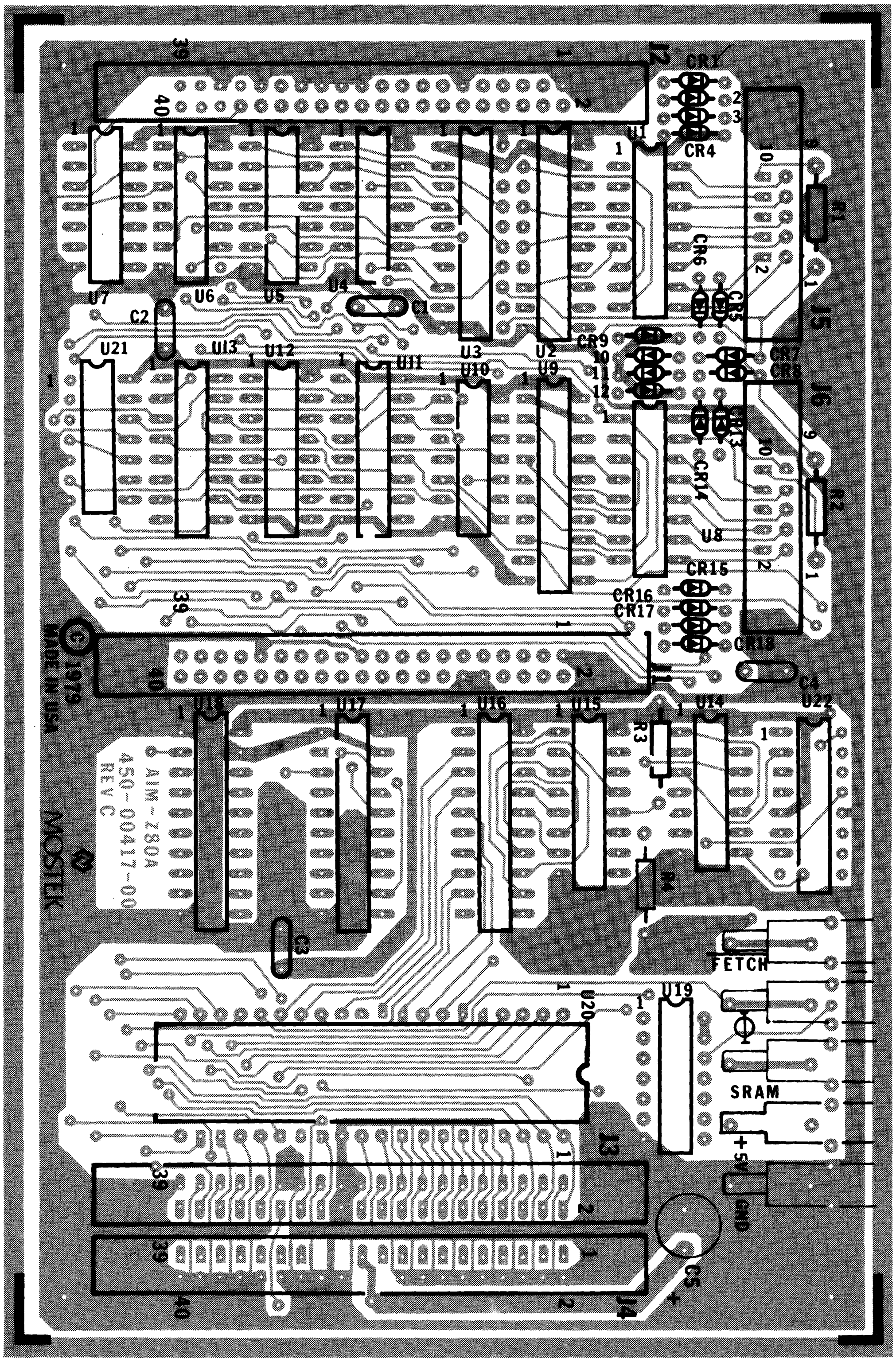
REVISION				
LTR		DESCRIPTION	APPD	DATE
C				
C1				



SIZE	DRAWING NO	REV
D	450-00418-00	C1
SCALE		SHEET 2 OF 2

PARTS PLACEMENT DIAGRAM

BUFFER BOX



PARTS LIST

BUFFER BOX

PART NO.	QTY	DESCRIPTION	REFERENCE DESIGNATOR	USED ON
########	#####	AIM Z80AE ASSY		450-00616-12
2065000	1	CABLE ASSY, TARGET	1	450-00650-00
2065100	1	CABLE ASSY, J1 BD INTERCONNECT	2	450-00651-00
2065200	1	CABLE ASSY, J2 BD INTERCONNECT	3	450-00652-00
2065300	1	CABLE ASSY, J3 BD INTERCONNECT	4	450-00653-00
2064900	2	CABLE ASSY, TEST PROBES	5	450-00649-00
2041700	1	AIM Z80A PCB	6	450-00417-00
2047000	1	AIM-XE CONTROL PCB ASSY	7	450-00470-00
2043500	1	AIM-XE HISTORY PCB ASSY	8	450-00435-00
4140342	1	BASE	9	450-00506-00
4140341	1	COVER	10	450-00505-00
5025974	1	NAMEPLATE	11	450-00518-00
4280051	4	FOOT,RUBBER	12	
4210343	8	SCREW,#4-40 X 1/4 PPH 80	13	
4210367	4	SPACER,HEX,4-40 X 1/4,W/STUD	14	
4210368	4	SPACER,HEX,4-40 X 3/4,TAPPED	15	
4210364	4	WASHER,#4 SLOTTED	16	

MOSTEK®

Z80•F8 Covering the full spectrum of
3870 microcomputer applications.

1215 W. Crosby Rd. • Carrollton, Texas 75006 • 214/323-6000
In Europe, Contact: MOSTEK Brussels
150 Chaussee de la Hulpe, B1170, Belgium;
Telephone: 660.69.24

Mostek reserves the right to make changes in specifications at any time and without notice. The information furnished by Mostek in this publication is believed to be accurate and reliable. However, no responsibility is assumed by Mostek for its use; nor for any infringements of patents or other rights of third parties resulting from its use. No license is granted under any patents or patent rights of Mostek.

PRINTED IN USA July 1980
Publication No. MK79650