AIM-280AE

OPERATION MANUAL

Publication Number 79650
<table>
<thead>
<tr>
<th>SECTION NUMBER</th>
<th>PARAGRAPH NUMBER</th>
<th>TITLE</th>
<th>PAGE NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>GENERAL DESCRIPTION</td>
<td>1-1</td>
</tr>
<tr>
<td></td>
<td>1.1</td>
<td>INTRODUCTION</td>
<td>1-1</td>
</tr>
<tr>
<td></td>
<td>1.2</td>
<td>REFERENCES</td>
<td>1-4</td>
</tr>
<tr>
<td></td>
<td>1.3</td>
<td>CONVENTIONS</td>
<td>1-4</td>
</tr>
<tr>
<td></td>
<td>1.4</td>
<td>TERMINOLOGY</td>
<td>1-5</td>
</tr>
<tr>
<td></td>
<td>1.5</td>
<td>BUFFER MODULE</td>
<td>1-6</td>
</tr>
<tr>
<td></td>
<td>1.6</td>
<td>CONTROL MODULE</td>
<td>1-9</td>
</tr>
<tr>
<td></td>
<td>1.7</td>
<td>HISTORY MODULE</td>
<td>1-11</td>
</tr>
<tr>
<td>2</td>
<td>2.1</td>
<td>AIM-Z80AE INSTALLATION</td>
<td>2-1</td>
</tr>
<tr>
<td></td>
<td>2.2</td>
<td>INTRODUCTION</td>
<td>2-1</td>
</tr>
<tr>
<td></td>
<td>2.3</td>
<td>COMPATIBLE SYSTEMS</td>
<td>2-1</td>
</tr>
<tr>
<td></td>
<td>2.4</td>
<td>EQUIPMENT REQUIRED</td>
<td>2-1</td>
</tr>
<tr>
<td></td>
<td>2.5</td>
<td>JUMPER OPTIONS, CONTROL MODULE</td>
<td>2-2</td>
</tr>
<tr>
<td></td>
<td>2.6</td>
<td>JUMPER OPTIONS, HISTORY MODULE</td>
<td>2-4</td>
</tr>
<tr>
<td></td>
<td>2.7</td>
<td>TARGET SYSTEM CONFIGURATION</td>
<td>2-6</td>
</tr>
<tr>
<td></td>
<td>2.8</td>
<td>INSTALLATION</td>
<td>2-7</td>
</tr>
<tr>
<td></td>
<td>2.8</td>
<td>INITIAL CHECKOUT PROCEDURE</td>
<td>2-9</td>
</tr>
<tr>
<td>3</td>
<td>3.1</td>
<td>AIM-Z80AE OPERATION</td>
<td>3-1</td>
</tr>
<tr>
<td></td>
<td>3.2</td>
<td>INTRODUCTION</td>
<td>3-1</td>
</tr>
<tr>
<td></td>
<td>3.3</td>
<td>PREPARATION</td>
<td>3-3</td>
</tr>
<tr>
<td></td>
<td>3.3</td>
<td>INITIALIZATION</td>
<td>3-3</td>
</tr>
<tr>
<td></td>
<td>3.4</td>
<td>ERRORS</td>
<td>3-3</td>
</tr>
<tr>
<td></td>
<td>3.5</td>
<td>COMMAND FORMAT</td>
<td>3-7</td>
</tr>
<tr>
<td></td>
<td>3.6</td>
<td>HEXADECIMAL NUMBERS</td>
<td>3-7</td>
</tr>
<tr>
<td></td>
<td>3.7</td>
<td>ASCII LITERAL</td>
<td>3-7</td>
</tr>
<tr>
<td></td>
<td>3.8</td>
<td>MNEMONICS</td>
<td>3-7</td>
</tr>
<tr>
<td></td>
<td>3.9</td>
<td>RELATIVE ADDRESS</td>
<td>3-9</td>
</tr>
<tr>
<td>SECTION NUMBER</td>
<td>PARAGRAPH NUMBER</td>
<td>TITLE</td>
<td>PAGE NUMBER</td>
</tr>
<tr>
<td>----------------</td>
<td>------------------</td>
<td>----------------------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>3</td>
<td>3.10</td>
<td>IMPLIED MEMORY ADDRESS POINTER</td>
<td>3-9</td>
</tr>
<tr>
<td></td>
<td>3.11</td>
<td>ADDED OR SUBTRACTED NUMBERS</td>
<td>3-9</td>
</tr>
<tr>
<td></td>
<td>3.12</td>
<td>EQUAL SIGN</td>
<td>3-9</td>
</tr>
<tr>
<td></td>
<td>3.13</td>
<td>SPECIAL KEYS</td>
<td>3-10</td>
</tr>
<tr>
<td></td>
<td>3.14</td>
<td>B-BREAKPOINT COMMAND</td>
<td>3-11</td>
</tr>
<tr>
<td></td>
<td>3.15</td>
<td>DESCRIPTION, B COMMAND</td>
<td>3-11</td>
</tr>
<tr>
<td></td>
<td>3.16</td>
<td>EXAMPLES, B COMMAND</td>
<td>3-15</td>
</tr>
<tr>
<td></td>
<td>3.17</td>
<td>C-COPY COMMAND</td>
<td>3-18</td>
</tr>
<tr>
<td></td>
<td>3.18</td>
<td>DESCRIPTION, C COMMAND</td>
<td>3-18</td>
</tr>
<tr>
<td></td>
<td>3.19</td>
<td>EXAMPLE, C COMMAND</td>
<td>3-18</td>
</tr>
<tr>
<td></td>
<td>3.20</td>
<td>D-DUMP COMMAND</td>
<td>3-19</td>
</tr>
<tr>
<td></td>
<td>3.21</td>
<td>DESCRIPTION, D COMMAND</td>
<td>3-19</td>
</tr>
<tr>
<td></td>
<td>3.22</td>
<td>EXAMPLE, D COMMAND</td>
<td>3-19</td>
</tr>
<tr>
<td></td>
<td>3.23</td>
<td>E-EXECUTE COMMAND</td>
<td>3-20</td>
</tr>
<tr>
<td></td>
<td>3.24</td>
<td>DESCRIPTION, E COMMAND</td>
<td>3-20</td>
</tr>
<tr>
<td></td>
<td>3.25</td>
<td>EXAMPLES, E COMMAND</td>
<td>3-21</td>
</tr>
<tr>
<td></td>
<td>3.26</td>
<td>ESCAPE-EXECUTION ESCAPE COMMAND</td>
<td>3-22</td>
</tr>
<tr>
<td></td>
<td>3.27</td>
<td>DESCRIPTION, ESCAPE COMMAND</td>
<td>3-22</td>
</tr>
<tr>
<td></td>
<td>3.28</td>
<td>EXAMPLE, ESCAPE COMMAND</td>
<td>3-22</td>
</tr>
<tr>
<td></td>
<td>3.29</td>
<td>F-FILL COMMAND</td>
<td>3-23</td>
</tr>
<tr>
<td></td>
<td>3.30</td>
<td>DESCRIPTION, F COMMAND</td>
<td>3-23</td>
</tr>
<tr>
<td></td>
<td>3.31</td>
<td>EXAMPLES, F COMMAND</td>
<td>3-23</td>
</tr>
<tr>
<td></td>
<td>3.32</td>
<td>G-GET COMMAND</td>
<td>3-24</td>
</tr>
<tr>
<td></td>
<td>3.33</td>
<td>DESCRIPTION, G COMMAND</td>
<td>3-24</td>
</tr>
<tr>
<td></td>
<td>3.34</td>
<td>EXAMPLES, G COMMAND</td>
<td>3-24</td>
</tr>
<tr>
<td></td>
<td>3.35</td>
<td>H-HEXADECIMAL ARITHMETIC</td>
<td>3-25</td>
</tr>
<tr>
<td></td>
<td>3.36</td>
<td>DESCRIPTION, H COMMAND</td>
<td>3-25</td>
</tr>
<tr>
<td></td>
<td>3.37</td>
<td>EXAMPLES, H COMMAND</td>
<td>3-25</td>
</tr>
<tr>
<td></td>
<td>3.38</td>
<td>I-INITIALIZE MEMORY MAP COMMAND</td>
<td>3-26</td>
</tr>
<tr>
<td></td>
<td>3.39</td>
<td>DESCRIPTION, I COMMAND</td>
<td>3-26</td>
</tr>
<tr>
<td>SECTION NUMBER</td>
<td>PARAGRAPH NUMBER</td>
<td>TITLE</td>
<td>PAGE NUMBER</td>
</tr>
<tr>
<td>----------------</td>
<td>------------------</td>
<td>-------------------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>3</td>
<td>3.40</td>
<td>EXAMPLES, I COMMAND</td>
<td>3-27</td>
</tr>
<tr>
<td></td>
<td>3.41</td>
<td>L-LOCATE COMMAND</td>
<td>3-29</td>
</tr>
<tr>
<td></td>
<td>3.42</td>
<td>DESCRIPTION, L COMMAND</td>
<td>3-29</td>
</tr>
<tr>
<td></td>
<td>3.43</td>
<td>EXAMPLES, L COMMAND</td>
<td>3-29</td>
</tr>
<tr>
<td></td>
<td>3.44</td>
<td>M-MODIFY COMMAND</td>
<td>3-30</td>
</tr>
<tr>
<td></td>
<td>3.45</td>
<td>DESCRIPTION, M COMMAND</td>
<td>3-30</td>
</tr>
<tr>
<td></td>
<td>3.46</td>
<td>EXAMPLE, M COMMAND</td>
<td>3-33</td>
</tr>
<tr>
<td></td>
<td>3.47</td>
<td>O-SET OFFSET COMMAND</td>
<td>3-34</td>
</tr>
<tr>
<td></td>
<td>3.48</td>
<td>DESCRIPTION, O COMMAND</td>
<td>3-34</td>
</tr>
<tr>
<td></td>
<td>3.49</td>
<td>EXAMPLE, O COMMAND</td>
<td>3-34</td>
</tr>
<tr>
<td></td>
<td>3.50</td>
<td>P-PORT COMMAND</td>
<td>3-35</td>
</tr>
<tr>
<td></td>
<td>3.51</td>
<td>DESCRIPTION, P COMMAND</td>
<td>3-35</td>
</tr>
<tr>
<td></td>
<td>3.52</td>
<td>EXAMPLES, P COMMAND</td>
<td>3-36</td>
</tr>
<tr>
<td></td>
<td>3.53</td>
<td>Q-QUIT COMMAND</td>
<td>3-37</td>
</tr>
<tr>
<td></td>
<td>3.54</td>
<td>DESCRIPTION, Q COMMAND</td>
<td>3-37</td>
</tr>
<tr>
<td></td>
<td>3.55</td>
<td>EXAMPLE, Q COMMAND</td>
<td>3-37</td>
</tr>
<tr>
<td></td>
<td>3.56</td>
<td>R-DISPLAY REGISTERS COMMAND</td>
<td>3-37</td>
</tr>
<tr>
<td></td>
<td>3.57</td>
<td>DESCRIPTION, R COMMAND</td>
<td>3-37</td>
</tr>
<tr>
<td></td>
<td>3.58</td>
<td>EXAMPLES, R COMMAND</td>
<td>3-38</td>
</tr>
<tr>
<td></td>
<td>3.59</td>
<td>S-STEP COMMAND</td>
<td>3-39</td>
</tr>
<tr>
<td></td>
<td>3.60</td>
<td>DESCRIPTION, S COMMAND</td>
<td>3-39</td>
</tr>
<tr>
<td></td>
<td>3.61</td>
<td>EXAMPLES, S COMMAND</td>
<td>3-40</td>
</tr>
<tr>
<td></td>
<td>3.62</td>
<td>T-TRACE HISTORY COMMAND</td>
<td>3-41</td>
</tr>
<tr>
<td></td>
<td>3.63</td>
<td>DESCRIPTION, T COMMAND</td>
<td>3-41</td>
</tr>
<tr>
<td></td>
<td>3.64</td>
<td>EXAMPLES, T COMMAND</td>
<td>3-43</td>
</tr>
<tr>
<td></td>
<td>3.65</td>
<td>V-VERIFY COMMAND</td>
<td>3-48</td>
</tr>
<tr>
<td></td>
<td>3.66</td>
<td>DESCRIPTION, V COMMAND</td>
<td>3-48</td>
</tr>
<tr>
<td></td>
<td>3.67</td>
<td>EXAMPLE, V COMMAND</td>
<td>3-48</td>
</tr>
<tr>
<td></td>
<td>3.68</td>
<td>W-WRITE TO ALTERNATE LUN COMMAND</td>
<td>3-49</td>
</tr>
</tbody>
</table>
### TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>SECTION NUMBER</th>
<th>PARAGRAPH NUMBER</th>
<th>TITLE</th>
<th>PAGE NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>3.69</td>
<td>DESCRIPTION, W COMMAND</td>
<td>3-49</td>
</tr>
<tr>
<td></td>
<td>3.70</td>
<td>EXAMPLE, W COMMAND</td>
<td>3-49</td>
</tr>
<tr>
<td></td>
<td>3.71</td>
<td>Z-INITIALIZE COMMAND</td>
<td>3-49</td>
</tr>
<tr>
<td></td>
<td>3.72</td>
<td>DESCRIPTION, Z COMMAND</td>
<td>3-49</td>
</tr>
<tr>
<td></td>
<td>3.73</td>
<td>%-TEST MEMORY COMMAND</td>
<td>3-50</td>
</tr>
<tr>
<td></td>
<td>3.74</td>
<td>DESCRIPTION, % COMMAND</td>
<td>3-50</td>
</tr>
<tr>
<td></td>
<td>3.75</td>
<td>EXAMPLES, % COMMAND</td>
<td>3-50</td>
</tr>
<tr>
<td></td>
<td>3.76</td>
<td>EXECUTION TIMER OPERATION</td>
<td>3-51</td>
</tr>
<tr>
<td></td>
<td>3.77</td>
<td>OPERATION WITH MD-SBC1</td>
<td>3-51</td>
</tr>
<tr>
<td></td>
<td>3.78</td>
<td>OPERATION WITH MDX</td>
<td>3-51</td>
</tr>
<tr>
<td></td>
<td>3.79</td>
<td>OPERATION WITH OEM-80</td>
<td>3-52</td>
</tr>
</tbody>
</table>

APPENDIX A
FACTORY NOTICES A-1

APPENDIX B
SPECIFICATIONS B-1

APPENDIX C
SCHEMATIC DIAGRAMS
- BUFFER BOX SCHEMATIC C-1
- BUFFER BOX ASSEMBLY DRAWING C-4
- BUFFER BOX PARTS LIST C-5
- HISTORY MODULE SCHEMATIC C-6
- HISTORY MODULE ASSEMBLY DRAWING C-10
- HISTORY PARTS LIST C-11
- CONTROL MODULE SCHEMATIC C-12
- CONTROL MODULE ASSEMBLY DRAWING C-16
- CONTROL MODULE PARTS LIST C-17
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>DESCRIPTION</th>
<th>NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>AIM-Z80AE SYSTEM</td>
<td>1-3</td>
</tr>
<tr>
<td>1.2</td>
<td>BUFFER MODULE</td>
<td>1-8</td>
</tr>
<tr>
<td>1.3</td>
<td>CONTROL MODULE</td>
<td>1-10</td>
</tr>
<tr>
<td>1.4</td>
<td>HISTORY MODULE</td>
<td>1-12</td>
</tr>
<tr>
<td>2.1</td>
<td>INSTALLATION IN MATRIX OR SYS-80F</td>
<td>2-9</td>
</tr>
<tr>
<td>2.2</td>
<td>MINIMUM TARGET SYSTEM</td>
<td>2-14</td>
</tr>
<tr>
<td>3.1</td>
<td>AIM-Z80 COMMAND SUMMARY</td>
<td>3-1</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE</th>
<th>DESCRIPTION</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>ERRORS</td>
<td>3-4</td>
</tr>
<tr>
<td>3.2</td>
<td>AIM-Z80 MNEMONICS</td>
<td>3-8</td>
</tr>
<tr>
<td>3.3</td>
<td>SPECIAL KEYS</td>
<td>3-10</td>
</tr>
</tbody>
</table>
1.1 INTRODUCTION

AIM-Z80AE is an advanced development tool which provides debug assistance for both software and hardware via in-circuit emulation of the Z80 microprocessor. Use of the AIM-Z80AE is completely transparent to the Target system configuration. No memory space or ports are used and all signals including RESET, INT, NMI, BUSRQ and 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-Z80AE 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 Z80 CPU socket. After AIM-Z80AE 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-Z80AE system software (AIMZ80) may be initialized by using the implied run command. AIMZ80 will sign on, take control of the Target system and allow the user to initialize the Target system and use any of the AIM-Z80AE 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-Z80AE, 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-Z80AE hardware. The System software includes the MOSTEK disk-based development systems software as well as the AIM-Z80AE 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 Buffer module with the following signals.

<table>
<thead>
<tr>
<th>TEST POINT</th>
<th>SIGNAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLUE</td>
<td>FETCH  (M1 for first opcode of instruction)</td>
</tr>
<tr>
<td>GREEN</td>
<td>PHI    (Z80 clock)</td>
</tr>
<tr>
<td>YELLOW</td>
<td>SRAM   (Active when Emulation RAM is accessed)</td>
</tr>
<tr>
<td>RED</td>
<td>+5 VOLTS</td>
</tr>
<tr>
<td>BLACK</td>
<td>GND</td>
</tr>
</tbody>
</table>

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 Interrupt) 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
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-Z80AE is initialized, the control signals $\overline{M1}$, $\overline{MREQ}$, $\overline{RD}$ and $\overline{WR}$ 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 Z80 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 Z80 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.
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 2.0
AIM-Z80AE INSTALLATION

2.1 INTRODUCTION

This section gives the background information and a step by step procedure for installing AIM-Z80AE in the MOSTEK disk-based development system and checking the basic operation.

2.2 COMPATIBLE SYSTEMS

AIM-Z80AE 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-Z80AE:

- MOSTEK disk-based development system with console terminal.
- Target system with Z80 CPU in socket, power and clock circuit.
- AIM-Z80AE Buffer module with cable assemblies
- AIM-Z80AE Control module
- AIM-Z80AE History module
- Connector assembly, 10 pin flat
- MOSTEK system diskette with AIMZ80.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:

<table>
<thead>
<tr>
<th>MK4116/MK4164 (16k/64k)</th>
<th>MK4332 (32k)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 → 2</td>
<td>1 → 2</td>
</tr>
<tr>
<td>3 → 4</td>
<td>3 → 4</td>
</tr>
</tbody>
</table>

3. J6 selects the block of system ports used by the Control module. Ports 80-87 are used so the following jumpers should be installed:

| BOARD 0                  |
| 80H-87H                 |
| 2 8                     |
| : [: :]                 |
| 1 7                     |
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-80 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 40H-47H are used so the following jumpers should be installed:

   BOARD 0
   40H-47H

   1 ... 16
   2 ... 15
   3 ... 14
   4 -- 13
   5 -- 12
   6 ... 11
   7 ... 10
   8 ... 9
3. Pins 2, 7, 8, 9, 10, and 15 allow the event counter to be extended to 16 bits by using channel one of the OEM-80 CTC for the upper eight bits. Only one AIM-Z80AE can have the event counter extended to sixteen bits.

<table>
<thead>
<tr>
<th>EIGHT BIT EVENT COUNTER</th>
<th>SIXTEEN BIT EVENT COUNTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 ... 16</td>
<td>1 ... 16</td>
</tr>
<tr>
<td>2 ... 15</td>
<td>2  →  15</td>
</tr>
<tr>
<td>3 ... 14</td>
<td>3 ... 14</td>
</tr>
<tr>
<td>4 ... 13</td>
<td>4 ... 13</td>
</tr>
<tr>
<td>5 ... 12</td>
<td>5 ... 12</td>
</tr>
<tr>
<td>6 ... 11</td>
<td>6 ... 11</td>
</tr>
<tr>
<td>7  →  10</td>
<td>7 ... 10</td>
</tr>
<tr>
<td>8 ... 9</td>
<td>8  →  9</td>
</tr>
</tbody>
</table>
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 RESET, BUSRQ, WAIT cannot be in the active state during initialization.

4. The Target Data bus must be tristate except when MREQ OR 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-280AE in an MOSTEK disk-based development system. Begin with all power OFF and the diskettes NOT inserted. When installing AIM-280A in MATRIX or SYS-80F systems it is necessary to make the following interconections between SK2 of the OEM-80, Control and history modules. The signals should be wire wrap or plug wire installed on the backplane connectors. On Matrix system, TPI may be used for connections to the OEM-80. If these connections are not performed the T state timer and extended EVENT counter will not function. Figure 2.1 illustrates the interconnections.

<table>
<thead>
<tr>
<th>SIGNAL</th>
<th>FROM</th>
<th>TO</th>
</tr>
</thead>
<tbody>
<tr>
<td>CK/TG1B (BIT 7 of EVENT)</td>
<td>HISTORY SK2-4a</td>
<td>OEM-80E SK2-4a</td>
</tr>
<tr>
<td>ZC/T01B (BIT 15 of EVENT)</td>
<td>OEM-80E SK2-4C</td>
<td>HISTORY SK2-4C</td>
</tr>
<tr>
<td>CK/TG2B (BIT 7 of TIMER)</td>
<td>HISTORY SK2-5a</td>
<td>OEM-80E SK2-5a</td>
</tr>
<tr>
<td>ZC/T02B (BIT 15 of TIMER)</td>
<td>OEM-80E SK2-5C</td>
<td>CONTROL SK2-4C</td>
</tr>
</tbody>
</table>

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-80 is in the "EO000" 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-80F
2.8 INITIAL CHECKOUT PROCEDURE

The following checkout procedure may be used to verify the basic operation of AIM-Z80AE. 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 0-0FFH during the initial checkout procedure.

1. Perform the installation procedure in section 2.7.

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

$AIMZ80 TEST(CR)  < load and run AIM-Z80AE
                 < also load target program "TEST"
AIM-Z80AE VERSION 1.0

,I (CR)  < display target memory map

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

|   0000 | S   |   1000 | .   |   2000 | .   |   3000 | .   |   4000 | .   |   5000 | .   |   6000 | .   |   7000 | .   |   8000 | .   |   9000 | .   |  A000  | .   |  B000  | .   |  C000  | .   |  D000  | .   |  E000  | SP  |  F000  | SP  |
|-------|-----|-------|-----|-------|-----|-------|-----|-------|-----|-------|-----|-------|-----|-------|-----|-------|-----|-------|-----|-------|-----|-------|-----|-------|-----|-------|-----|-------|-----|
| S     | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |
| 1000  | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |
| 2000  | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |
| 3000  | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |
| 4000  | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |
| 5000  | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |
| 6000  | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |
| 7000  | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |
| 8000  | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |
| 9000  | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |
| A000  | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |
| B000  | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |
| C000  | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |
| D000  | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |       | .   |
| E000  | SP  |       | SP  |       | SP  |       | SP  |       | SP  |       | SP  |       | SP  |       | SP  |       | SP  |       | SP  |       | SP  |       | SP  |       | SP  |       | SP  |       | SP  |       | T   |
| F000  | SP  |       | SP  |       | SP  |       | SP  |       | SP  |       | SP  |       | SP  |       | SP  |       | SP  |       | SP  |       | SP  |       | SP  |       | SP  |       | SP  |       | SP  |       | T   |
,M 0,F(CR)  < display "TEST" program
00 00 3E 00 3C 3C 2F 3C 3C 2F 18 F6 00 00 00 00
,S 0,(CR)  < single-step thru program

PC  AF  I  IF  BC  DE  HL  DISASSEMBLY  IX  IY  SP
0000  FFA9 0041 FFFF FFFF 0200  LD  A,0    BFFF FFFF 01F0  7 uSec
0002  00A9 0041 FFFF FFFF 0200  INC  A     BFFF FFFF 01F0  11 uSec (CR)
0003  0101 0041 FFFF FFFF 0200  INC  A     BFFF FFFF 01F0  15 uSec (CR)
0004  0201 0041 FFFF FFFF 0200  CPL     BFFF FFFF 01F0
,B A4(CR)  < set a software breakpoint

,E 0(CR)  < execute target program

,SWBP ENCOUNTERED 15 uSec
PC  AF  I  IF  BC  DE  HL  DISASSEMBLY  IX  IY  SP
0004 0201 0041 FFFF FFFF 0200  CPL     BFFF FFFF 01F0
,T 5,-4(CR)  < display history memory

OFFS  ADDR  DB  DISASSEMBLY  TYPE  PA8------PA0  PB8------PB0
-004 0000 3E  LD  A,0    FETC  1 1111 1111  1 1111 1111
-003 0001 00  MRR  1 1111 1111  1 1111 1111
-002 0002 3C  INC  A     FETC  1 1111 1111  1 1111 1111
-001 0003 3C  INC  A     FETC  1 1111 1111  1 1111 1111
+000 0004 5B  BREAKPOINT CODE    FETC  1 1111 1111  1 1111 1111
,B C,A(CR)  < clear all breakpoints

,B 0,H(CR)  < set a hardware breakpoint

,E 0(CR)  < execute target program

,HWBP ENCOUNTERED 7 uSec
PC AF I IF BC DE HL DISASSEMBLY IX IY SP
0002 0001 0041 FFFF FFFF 0200 INC A BFFF FFFF 01F0
0005 FD3B 0041 FFFF FFFF 0200 INC A BFFF FFFF 01F0

< update history options

PA7—PA0 A15—A0
TRIGGER WORD IS: XXXX XXXX 0000 0000 0000 0000 (CR)
UPDATE: XXXX XXXX 0000 0000 0000 0000

TRIGGER STROBE IS (MRD,LE) ;TO CHANGE SELECT ONE:
MRD(0) MWR(1) MREQ(2) IORD(3)
IOWR(4) IORQ(5) INTA(6) PA8(7) [ ,LE(8) ,TE(9) ] -> (CR)

EVENT COUNT IS: 0001 -> 2(CR)

DELAY COUNT IS: 000 -> 4(CR)

HISTORY CLOCK IS (MRD MWR IORD ICWR ) ;TO CHANGE SELECT ANY:
MRD(0) MWR(1) MRF(2)
IORD(3) ICWR(4) PA8+(5) PA8+(6) -> (CR)

HISTORY CLOCK ENABLE IS (ALL) CYCLES ;TO CHANGE SELECT ONE:
ALL(0) DMA(1) CPU(2) PA7L(3) PA7H(4)
TWORD(6) [ ,PB8L(8) ,PB8H(9) ] -> (CR)

< execute again

HWBP ENCOUNTERED 62 uSec

OFFS ADDR DB DISASSEMBLY TYPE PA8—PA0 PB8—PB0
-00A 0000 3E LD A,0 FETC 1 1111 1111 1 1111 1111
-009 0001 00 MRD 1 1111 1111 1 1111 1111
-008 0002 3C INC A FETC 1 1111 1111 1 1111 1111
-007 0003 3C INC A FETC 1 1111 1111 1 1111 1111
-006 0004 2F CPL FETC 1 1111 1111 1 1111 1111
-005 0005 3C INC A FETC 1 1111 1111 1 1111 1111
-004 0006 3C INC A FETC 1 1111 1111 1 1111 1111
-003 0007 2F CPL FETC 1 1111 1111 1 1111 1111
-002 0008 18 JR -0AH FETC 1 1111 1111 1 1111 1111
-001 0009 F6 MRD 1 1111 1111 1 1111 1111
+000 0000 3E LD A,0 FETC 1 1111 1111 1 1111 1111
+001 0001 00 MRD 1 1111 1111 1 1111 1111
+002 0002 3C INC A FETC 1 1111 1111 1 1111 1111
+003 0003 3C INC A FETC 1 1111 1111 1 1111 1111
+004 0004 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 0(CR) < execute target program

,TIMER BREAK ENCOUNTERED 989 uSec
PC AF I IF BC DE HL DISASSEMBLY IX IY SP
0000 0093 0041 FFFF FFFF 0200 LD A,0 BFFF FFFF 01F0
,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
0002 0093 0041 FFFF FFFF 0200 INC A BFFF FFFF 01F0
FIGURE 2.2 MINIMUM TARGET SYSTEM
3.1 INTRODUCTION

This section describes the operation of the AIM-Z80AE control program (AIMZ80). 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-Z80 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.
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-80DOS is operating AIMZ80.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 AIMZ80 follows.

$AIMZ80 [file name](CR)
AIMZ80 VERSION 1.0

The AIMZ80 program will be loaded and executed and the sign on message and prompt character will be printed. If the file AIMZ80.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 60 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 Z80 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 Z80 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 though errors may occur.

*** ERROR 68 INVALID OPCODE

This message is generated if an attempt is made to disassemble an undocumented Z80 instruction.

*** ERROR 69 INVALID OPERAND

This message is generated if an attempt is made to disassemble a Z80 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 AIMZ80 commands is:

\[ x_{\text{a,b,c}} \text{(CR)} \]

Where \( x \) is the single character specifying 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 AIMZ80.
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>:PC</td>
<td>Program Counter</td>
</tr>
<tr>
<td>:A</td>
<td>Accumulator</td>
</tr>
<tr>
<td>:F</td>
<td>Flags</td>
</tr>
<tr>
<td>:I</td>
<td>Interrupt Vector Register (upper 8 bits)</td>
</tr>
<tr>
<td>:IF</td>
<td>Interrupt flip-flop</td>
</tr>
<tr>
<td>:B</td>
<td>B register</td>
</tr>
<tr>
<td>:C</td>
<td>C register</td>
</tr>
<tr>
<td>:D</td>
<td>D register</td>
</tr>
<tr>
<td>:E</td>
<td>E register</td>
</tr>
<tr>
<td>:H</td>
<td>H register</td>
</tr>
<tr>
<td>:L</td>
<td>L register</td>
</tr>
<tr>
<td>:A'</td>
<td>A' register</td>
</tr>
<tr>
<td>:F'</td>
<td>F' register</td>
</tr>
<tr>
<td>:B'</td>
<td>B' register</td>
</tr>
<tr>
<td>:C'</td>
<td>C' register</td>
</tr>
<tr>
<td>:D'</td>
<td>D' register</td>
</tr>
<tr>
<td>:E'</td>
<td>E' register</td>
</tr>
<tr>
<td>:H'</td>
<td>H' register</td>
</tr>
<tr>
<td>:L'</td>
<td>L' register</td>
</tr>
<tr>
<td>:IX</td>
<td>IX register</td>
</tr>
<tr>
<td>:IY</td>
<td>IY register</td>
</tr>
<tr>
<td>:SP</td>
<td>Stack pointer</td>
</tr>
<tr>
<td>:TS</td>
<td>T = State length in nanoseconds</td>
</tr>
</tbody>
</table>
3.9 RELATIVE ADDRESS

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

3.10 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**

<table>
<thead>
<tr>
<th>KEY</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>PERIOD (.)</td>
<td>Abort command and return to the command mode with the operand not updated.</td>
</tr>
<tr>
<td>SPACE BAR (SP)</td>
<td>Suspend printout if printing or continue printout if printout is suspended.</td>
</tr>
<tr>
<td>SLASH(/)</td>
<td>Abort command and return to the command mode with the operand updated.</td>
</tr>
<tr>
<td>UPCARET (UP)</td>
<td>Backup to previous address or option or display updated address or option.</td>
</tr>
<tr>
<td>CARRIAGE RETURN (CR)</td>
<td>Advance to next address or option or return to command mode.</td>
</tr>
</tbody>
</table>
3.14 B-BREAKPOINT COMMAND

FORMATS:

\[ B \] (CR) \quad \text{< Tabulate breakpoints} \quad (1)
\[ B \text{ a(CR)} \quad \text{< Set software breakpoint} \quad (2)
\[ B \text{ a,H(CR)} \quad \text{< Set hardware breakpoint} \quad (3)
\[ B \text{ a,C(CR)} \quad \text{< Clear breakpoint} \quad (4)
\[ B \text{ C,A(CR)} \quad \text{< Clear All breakpoints} \quad (5)
\[ B \text{ H,O(CR)} \quad \text{< Specify History Options} \quad (6)
\[ B \text{ t,u,T(CR)} \quad \text{< Set timeout breakpoint} \quad (7)
\[ B \text{ x,C,T(CR)} \quad \text{< Clear timeout breakpoint} \quad (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 user's 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), 0 (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-PA0) corresponds to the A PROBES on the Buffer module. The probes are color coded to standard resistor color codes (0-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-A0) corresponds to the Target Z80 address bus.
2. The second option is the trigger strobe. The trigger strobe may be updated by entering one of the following numbers corresponding to 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.

   0 - MEMORY READ
   1 - MEMORY WRITE
   2 - MEMORY READ OR WRITE
   3 - PORT READ
   4 - PORT WRITE
   5 - PORT READ OR WRITE
   6 - INTERRUPT 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 occurrences of the trigger word which must occur before the breakpoint sequence is initiated. The EVENT COUNT is initialized to "1" when the AIMZ80 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.

- 0 - 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.

- 0 - 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:

<table>
<thead>
<tr>
<th>MULTIPLIER</th>
<th>SECOND OPERAND &quot;u&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>t states</td>
<td>T</td>
</tr>
<tr>
<td>microseconds</td>
<td>U</td>
</tr>
<tr>
<td>10 microseconds</td>
<td>10U</td>
</tr>
<tr>
<td>100 microseconds</td>
<td>100U</td>
</tr>
<tr>
<td>milliseconds</td>
<td>M</td>
</tr>
<tr>
<td>10 milliseconds</td>
<td>10M</td>
</tr>
<tr>
<td>100 milliseconds</td>
<td>100M</td>
</tr>
<tr>
<td>seconds</td>
<td>S</td>
</tr>
<tr>
<td>10 seconds</td>
<td>10S</td>
</tr>
<tr>
<td>100 seconds</td>
<td>100S</td>
</tr>
</tbody>
</table>

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

$AIMZ80 TEST(CR) < load AIM-Z80AE and "TEST" programs
AIM-Z80AE VERSION 1.0
,B 4(CR) < set software breakpoint
,E 0(CR) < execute target program
SWBP ENCOUNTERED 15 USEC

PC AF I IF BC DE HL DISASSEMBLY IX IY SP
0004 0201 0041 FFFF FFFF 0200 CPL
BFFF FFFF 01F0 _
,B (CR) < list breakpoints

SW BREAKPOINTS: 0004
NO HW BREAKPOINT
NO TIMER BREAKPOINT

,B 0,H(CR) < set hardware breakpoint at 0
,B 4,C(CR) < clear breakpoint at 4
,B (CR) < list breakpoints

NO SW BREAKPOINTS
HW BREAKPOINT: 0000
NO TIMER BREAKPOINT

,B H,2O(CR) < update history options

PA7---PA0 A15---------------A0
TRIGGER WORD IS: XXXX XXXX 0000 0000 0000 0000 (CR) < no change
UPDATE: XXXX XXXX 0000 0000 0000 0000

TRIGGER STROBE IS (MRD,LE) ;TO CHANGE SELECT ONE:

MRD(0) MWR(1) MREQ(2) IORD(3)
IOWR(4) IORQ(5) INTA(6) PA8(7) [ ,LE(8) ,TE(9)] --> (CR)

EVENT COUNT IS: 0002 --> 2(CR)

DELAY COUNT IS: 004 --> (CR)

HISTORY CLOCK IS (MRD MWR IORD IOWR );TO CHANGE SELECT ANY:

MRD(0) MWR(1) MRF(2)
IORD(3) IOWR(4) PA8+(5) PA8-(6) -->(CR)
HISTORY CLOCK ENABLE IS (ALL) CYCLES; TO CHANGE SELECT ONE:

ALL(0) DMA(1) CPU(2) PA7L(3) PA7H(4)
TWORD(6) [ ,PB8L(8) ,PB8H(9)] --> (CR)

,E 0(CR)  < execute target program

HWBP ENCOUNTERED 62 USEC
PC AF I IF BC DE HL DISASSEMBLY IX IY SP
0005 FD3B 0041 FFFF FFFF 0200 INC A BFFF FFFF 01F0 .
,B C,A(CR)  < clear all breakpoints
,B 10,M,T(CR)  < set 10 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 0(CR)  < execute target program

TIMER BREAK ENCOUNTERED 9,964 USEC
PC AF I IF BC DE HL DISASSEMBLY IX IY SP
0008 0093 0041 FFFF FFFF 0200 JR -0AH BFFF FFFF 01F0 .

,M :TS(CR)  < change t-state value
:TS 1000 250(UP)
:TS 250 -
,E 0(CR)  < execute again

TIMER BREAK ENCOUNTERED 2,491 USEC < note different time value
PC AF I IF BC DE HL DISASSEMBLY IX IY SP
0008 0093 0041 FFFF FFFF 0200 JR -0AH BFFF FFFF 01F0 .
,M :TS(CR)  < restore old value
:TS 250 1000(UP)
:TS 1000 -
,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.

3.19 EXAMPLE, C COMMAND

,M 0,lF(CR)       < note two different lines of code
0000 3E 00 3C 3C 2F 3C 3C 2F 18 F6 00 00 00 00 00 00
0010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
,C 0,F,10(CR)     < copy first line to second line
,M 0,lF(CR)       < note result of copy
0000 3E 00 3C 3C 2F 3C 3C 2F 18 F6 00 00 00 00 00 00 00 00 00 00
0010 3E 00 3C 3C 2F 3C 3C 2F 18 F6 00 00 00 00 00 00 00 00 00 00
3.20 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 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 AIMZ80, 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 0,F(CR)  < dump address 0 thru 0FH

ENTER TARGET PROGRAM FILE NAME OR (CR) --> DK1:TEST(CR)
DK0: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"

DK0:TEST .MAP[1],ALREADY EXISTS
ERASE? Y(CR)
   < the machine does the rest
3.23 E-EXECUTE COMMAND

FORMAT:

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

3.24 DESCRIPTION, E COMMAND

The E command is used to initiate real time execution of the Target Z80 CPU at the desired address. Before real time execution begins, all of the Z80 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 2H(CR)` < Set hardware breakpoint at 2H.
`E 0H(CR)` < Begin execution at 0H.

HWBP ENCOUNTERED 70 USEC
PC AF I IF BC DE HL DISASSEMBLY IX IY SP
0007 FFA9 0041 FFFF FFFF 0200 CPL BFFF FFFF 01F0 ·
`E (CR)` < continue execution

HWBP ENCOUNTERED 156 USEC
PC AF I IF BC DE HL DISASSEMBLY IX IY SP
0007 FFA9 0041 FFFF FFFF 0200 CPL BFFF FFFF 01F0 ·

`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 Z80 CPU may be executing. If the Target Z80 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 0(CR)          < Begin execution at 0H.
,(ESC)            < (ESC) entered.
, 2,507,298 USEC

PC AF I IF BC DE HL DISASSEMBLY IX IY SP
0003 0101 0041 FFFF FFFF 0200 CPL       BFFF FFFF 01F0 _
3.29 F-FILL COMMAND

FORMAT:

,F s,f,d(CR)

3.30 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 10,1F(CR)    < look at it first
0010 3E 00 3C 3C 2F 3C 3C 2F 18 F6 00 00 00 00 00 00
,F 10,1F,AA(CR) < fill it with 0AAH
,M 10,1F(CR)    < look at it again
0010 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 AIM280, 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 0,F,AA(CR) < destroy old program by overwriting
,M 0,F(CR) < look at it
0000 AA AA AA AA AA AA AA AA AA AA AA AA AA
,G DK1:TEST(CR) < reload the program
,M 0,F(CR) < look at it again
0000 3E 00 3C 3C 2F 3C 3C 2F 18 F6 00 00 00 00 00 00
,
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**

\[ H \ 0100+0900=0A00 (CR) \]
\[ H \ 100+900-800=0200 (CR) \]
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.

### 3.40 EXAMPLES, I COMMAND.

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

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

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 0000 | S | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 1000 | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 2000 | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 3000 | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 4000 | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 5000 | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 6000 | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 7000 | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 8000 | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 9000 | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| A000 | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| B000 | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| C000 | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| D000 | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| E000 | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP |
| F000 | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP | SP |

,I 0,FFF(CR)

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

,I 1000,1FFF(CR)

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

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

<table>
<thead>
<tr>
<th>Address</th>
<th>System Memory</th>
<th>Target Memory</th>
<th>Write Protected</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>S</td>
<td>S</td>
<td>P</td>
</tr>
<tr>
<td>1000</td>
<td>S</td>
<td>S</td>
<td>P</td>
</tr>
<tr>
<td>2000</td>
<td>S</td>
<td>S</td>
<td>P</td>
</tr>
<tr>
<td>3000</td>
<td>S</td>
<td>S</td>
<td>P</td>
</tr>
<tr>
<td>4000</td>
<td>S</td>
<td>S</td>
<td>P</td>
</tr>
<tr>
<td>5000</td>
<td>S</td>
<td>S</td>
<td>P</td>
</tr>
<tr>
<td>6000</td>
<td>S</td>
<td>S</td>
<td>P</td>
</tr>
<tr>
<td>7000</td>
<td>S</td>
<td>S</td>
<td>P</td>
</tr>
<tr>
<td>8000</td>
<td>S</td>
<td>S</td>
<td>P</td>
</tr>
<tr>
<td>9000</td>
<td>S</td>
<td>S</td>
<td>P</td>
</tr>
<tr>
<td>A000</td>
<td>S</td>
<td>S</td>
<td>P</td>
</tr>
<tr>
<td>B000</td>
<td>S</td>
<td>S</td>
<td>P</td>
</tr>
<tr>
<td>C000</td>
<td>S</td>
<td>S</td>
<td>P</td>
</tr>
<tr>
<td>D000</td>
<td>S</td>
<td>S</td>
<td>P</td>
</tr>
<tr>
<td>E000</td>
<td>S</td>
<td>S</td>
<td>P</td>
</tr>
<tr>
<td>F000</td>
<td>S</td>
<td>S</td>
<td>P</td>
</tr>
</tbody>
</table>

\[ , \]
3.41 L-LOCATE COMMAND

FORMAT:

,L s,f,d(CR)

3.42 DESCRIPTION, L COMMAND

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

3.43 EXAMPLES, L COMMAND

,G (CR) < load program
,M 0,F(CR) < look at it
0000 3E 00 3C 3C 2F 3C 3C 2F 18 F6 00 00 00 00 00
,L 0,F,3C(CR) < locate all 03CH
0002 3C
0003 3C
0005 3C
0006 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.

<table>
<thead>
<tr>
<th>SPECIAL KEY</th>
<th>UPDATE ENTERED</th>
<th>NO UPDATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>(.)</td>
<td>Return to AIM monitor</td>
<td>Return to AIM monitor.</td>
</tr>
<tr>
<td></td>
<td>Return to AIM monitor.</td>
<td>not updated.</td>
</tr>
<tr>
<td>(/)</td>
<td>Return to AIM monitor</td>
<td>Return to AIM monitor.</td>
</tr>
<tr>
<td></td>
<td>Return to AIM monitor.</td>
<td>updated.</td>
</tr>
<tr>
<td>(UP)</td>
<td>Redisplay address a.</td>
<td>Display address a-1.</td>
</tr>
<tr>
<td>(CR)</td>
<td>Display address a+1.</td>
<td>Display address a+1.</td>
</tr>
</tbody>
</table>

If "a" is a register mnemonic, this format is used to examine and/or update the register. Operation is exactly the same in other respects when examining or modifying registers as when examining or modifying memory. Table 3-3 is a complete list of the mnemonics 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 Z80 registers by using mnemonics 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:
<table>
<thead>
<tr>
<th>SPECIAL KEY</th>
<th>UPDATE ENTERED</th>
<th>NO UPDATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>(.)</td>
<td>Return to monitor without updating.</td>
<td>Return to monitor.</td>
</tr>
<tr>
<td>(SP)</td>
<td>Update, begin printing.</td>
<td>Begin printing.</td>
</tr>
<tr>
<td>(/)</td>
<td>Update, Return to monitor.</td>
<td>Return to monitor.</td>
</tr>
<tr>
<td>(UP)</td>
<td>Update, disassemble current instruction.</td>
<td>Disassemble previous instruction.</td>
</tr>
<tr>
<td>(CR)</td>
<td>Update, disassemble current instruction.</td>
<td>Disassemble next instruction.</td>
</tr>
</tbody>
</table>

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 0(CR) < examine memory starting at 0
0000 3E(CR)
0001 00(CR)
0002 3C(CR)
0003 3C(CR)
0004 2F(CR)
0005 3C(CR)
0006 3C(CR)
0007 2F(CR)
0008 18(CR)
0009 F6(CR)
000A 00 AA(UP) < change this one and confirm with (UP)
000A AA(UP)
0009 F6(UP) < back up
0008 18 .
,M 0,F(CR) < tabulate a whole line
0000 3E 00 3C 3C 2F 3C 3C 2F 18 F6 00 00 00 00 00 00
,M 0,F,D(CR) < disassemble
0000 3E00 LD A,0 (CR)
0002 3C INC A (CR)
0003 3C INC A (CR)
0004 2F CPL .
,M 5(CR) < disassembly mode still set
0005 3C INC A (CR)
0006 3C INC A (CR)
0007 2F CPL (CR)
0008 18F6 JR -0AH C3,00,00(UP) < update this one and confirm
0008 C30000 JP 0000H .
,M 0,F,0(CR) < turn off disassembly
0000 3E 00 3C 3C 2F 3C 3C 2F C3 00 00 00 00 00 00 00
,M 8(CR)
0008 C3 18(CR) < restore relative jump
0009 00 0-$=FFF6(UP) < compute displacement = destination-$
0009 F6 .
3.47 O-SET OFFSET COMMAND

FORMAT:

,0 a(CR) < Set relative Offset (1)
,0 (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

,0 100(CR) < Set relative offset to 100H.
,R (CR)   < Display registers
PC PC AF I IF BC DE HL DISASSEMBLY IX IY SP
'FF03 0003 0101 0041 FFFF FFFF 0200 INC A BFFF FFFF 01F0

,0 (CR)   < Disable relative offset.
,R (CR)   < Display registers.
PC AF I IF BC DE HL DISASSEMBLY IX IY SP
0003 0101 0041 FFFF FFFF 0200 INC A BFFF FFFF 01F0
3.50 P-PORT COMMAND

FORMAT:

\[ P \ a \ (CR) \]  \< \text{Examine or modify Port.} \quad (1)
\[ P\ s, f \ (CR) \]  \< \text{Tabulate ports.} \quad (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.

<table>
<thead>
<tr>
<th>SPECIAL KEY</th>
<th>OUTPUT DATA ENTERED</th>
<th>NO OUTPUT DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>(.)</td>
<td>Return to AIM monitor.</td>
<td>Return AIM monitor.</td>
</tr>
<tr>
<td>(/)</td>
<td>Output data, return to AIM monitor.</td>
<td>Return to AIM monitor.</td>
</tr>
<tr>
<td>(UP)</td>
<td>Output data, Display current port.</td>
<td>Display previous port.</td>
</tr>
</tbody>
</table>

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 D0(CR) < Examine port 0D0H.
D0 AA 55(UP) < Output 55H and verify.
D0 55 (CR) < Advance.
D1 FF(CR) < Advance.
D2 AA 55(UP) < Output 55 to port 0D2H.
D2 55 (UP) < Backup.
D1 FF (UP)
D0 55 -
,P 0, FF(CR) < Tabulate all ports.

00 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78
10 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78
20 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78
30 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78
40 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78
50 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78
60 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78
70 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78
80 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78
90 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78
A0 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78
B0 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78
C0 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78
D0 55 FF 55 FF FF FF FF FF 00 00 00 00 00 00 00 00
E0 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78
F0 78 78 78 78 78 78 78 78 78 78 78 78 78 78 78
3.53 **Q-QUIT COMMAND**

**FORMAT:**

\[,Q \text{(CR)}\]  \(\text{< quit}\)

3.54 **DESCRIPTION, Q COMMAND**

The Quit command is used to exit AIMZ80 and return to the system monitor, closing any open files.

3.55 **EXAMPLE, Q COMMAND**

\[,Q \text{(CR)}\]  \(\text{< Quit}\)

$
$

3.56 **R-DISPLAY REGISTERS COMMAND**

**FORMAT:**

\[,R \text{(CR)}\]  \(\text{< Display registers}\)  (1)
\[,R \ n\text{(CR)}\]  \(\text{< Specify number}\)  (2)
\[,R \ n,h\text{(CR)}\]  \(\text{< Specify optional heading}\)  (3)

3.57 **DESCRIPTION, R COMMAND**

The R command (any format) is used to display the Z80 CPU register values which are updated and displayed any time a breakpoint is encountered or when single/multi-stepping.

Format (1) will cause the Z80 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 0 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
0003 0101 0041 FFFF FFFF 0200 INC A
0000 0101 0041 FFFF FFFF 0200 INC A
,S 0,1,ND(CR)  < turn off disassembly

PC AF I IF BC DE HL A'F' B'C' D'E' H'L' IX IY SP
0000 0101 0041 FFFF FFFF 0200 0040 0101 FFFF 30FF BFSF FFFF 01F0 7 USEC
0002 0001 0041 FFFF FFFF 0200 0040 0101 FFFF 30FF BFSF FFFF 01F0 .
,R (CR)  < display all registers

PC AF I IF BC DE HL A'F' B'C' D'E' H'L' IX IY SP
0002 0001 0041 FFFF FFFF 0200 0040 0101 FFFF 30FF BFSF FFFF 01F0
,R 6,0(CR)  < turn off heading

0002 0001 0041 FFFF FFFF 0200
,R 6,H(CR)  < turn on heading

PC AF I IF BC DE HL
0002 0001 0041 FFFF FFFF 0200
3.59 S-STEP COMMAND

FORMAT:

\texttt{\textbackslash S a(CR)} \quad \textless \text{Begin single-step} \quad (1) \\
\texttt{\textbackslash S (CR)} \quad \textless \text{Continue single-step} \quad (2) \\
\texttt{\textbackslash S a,n(CR)} \quad \textless \text{Begin multi-step} \quad (3) \\
\texttt{\textbackslash S a,n,m(CR)} \quad \textless \text{Change display mode} \quad (4)

3.60 DESCRIPTION, S COMMAND

The \texttt{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 Z80 CPU registers are printed and the system waits in the single-step mode for one of the following special keys to be entered.

<table>
<thead>
<tr>
<th>SPECIAL KEY</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>(.)</td>
<td>Terminate \texttt{S} command, return to monitor</td>
</tr>
<tr>
<td>(SP)</td>
<td>Begin multi-step if suspended, suspend multi-step if stepping.</td>
</tr>
<tr>
<td>(CR)</td>
<td>Single Step next instruction.</td>
</tr>
</tbody>
</table>

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 \texttt{0FFH}.

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 Z80 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 0 (CR) < step from address 0

PC  AF  I  IF  BC  DE  HL  DISASSEMBLY  IX  IY  SP
0000  FFA8  0040  0101  FFFF  0200  LD  A,0  BFFF  FFFF  01F0  7 USEC
0002  00A8  0040  0101  FFFF  0200  INC  A  BFFF  FFFF  01F0  11 USEC(CR)
0003  0100  0040  0101  FFFF  0200  INC  A  BFFF  FFFF  01F0  15 USEC(CR)
0004  0200  0040  0101  FFFF  0200  CPL  BFFF  FFFF  01F0  19 USEC
,S  (CR) < resume stepping
19 USEC
0005  FD3A  0040  0101  FFFF  0200  INC  A  BFFF  FFFF  01F0  23 USEC(CR)
0006  FEA8  0040  0101  FFFF  0200  INC  A  BFFF  FFFF  01F0  15 USEC(CR)
,S  0,4 (CR) < multi-step
< see R command for additional example

PC  AF  I  IF  BC  DE  HL  DISASSEMBLY  IX  IY  SP
0000  0100  0040  0101  FFFF  0200  LD  A,0  BFFF  FFFF  01F0  7 USEC
0002  0000  0040  0101  FFFF  0200  INC  A  BFFF  FFFF  01F0  11 USEC
0003  0100  0040  0101  FFFF  0200  INC  A  BFFF  FFFF  01F0  15 USEC
0004  0200  0040  0101  FFFF  0200  CPL  BFFF  FFFF  01F0  19 USEC
0005  FD3A  0040  0101  FFFF  0200  INC  A  BFFF  FFFF  01F0  15 USEC
,S  0,4 (CR)
```


3.62 T-TRACE HISTORY COMMAND

FORMAT:

,.TR(CR) < Trace from start of history. (1)
,.Tn(CR) < Trace n history samples. (2)
,.Tn,o(CR) < Trace relative to breakpoint. (3)
,.Tn,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-PA0) 34-44
- B Probes (PB8-PB0) 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.

<table>
<thead>
<tr>
<th>SPECIAL KEY</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>(. )</td>
<td>Terminate T command return to monitor</td>
</tr>
<tr>
<td>(SP)</td>
<td>Suspend printout if printing, continue printout if not printing.</td>
</tr>
<tr>
<td>(UP)</td>
<td>Display previous history sample</td>
</tr>
<tr>
<td>(CR)</td>
<td>Display next history sample</td>
</tr>
<tr>
<td>(LF)</td>
<td>Display heading and next history sample</td>
</tr>
</tbody>
</table>
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 0FFH 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.

<table>
<thead>
<tr>
<th>Bit</th>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Display B probes</td>
</tr>
<tr>
<td>1</td>
<td>Display A probes</td>
</tr>
<tr>
<td>2</td>
<td>Display Cycle Type</td>
</tr>
<tr>
<td>3</td>
<td>Display Disassembly</td>
</tr>
</tbody>
</table>
3.64 EXAMPLES, T COMMAND

\[ \texttt{,B \_O,H(CR)} \quad < \text{set hardware breakpoint} \]

\[ \texttt{,B H,O(CR)} \quad < \text{update history options} \]

PA7---PA0 A15----------A0
TRIGGER WORD IS: XXXX XXXX 0000 0000 0000 0000 (CR)
UPDATE: XXXX XXXX 0000 0000 0000 0000

TRIGGER STROBE IS (MRD,LE) ; TO CHANGE SELECT ONE:
MRD(0) MWR(1) MREQ(2) IORD(3)
IOWR(4) IORQ(5) INTA(6) PA8(7) [ ,LE(8) ,TE(9)] \( \rightarrow \) (CR)

EVENT COUNT IS: 0001 \( \rightarrow \) 2(CR)

DELAY COUNT IS: 000 \( \rightarrow \) 4(CR)

HISTORY CLOCK IS (MRD MWR IORD IOWR ) ; TO CHANGE SELECT ANY:
MRD(0) MWR(1) MRF(2)
IORD(3) IOWR(4) PA8+(5) PA8-(6) \( \rightarrow \) 01234(CR) < enable all cycles

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

ALL(0) DMA(1) CPU(2) PA7L(3) PA7H(4)
TWORD(6) \[ ,PB8L(8) ,PB8H(9)] \( \rightarrow \) (CR)

\[ \texttt{,E \_O(CR)} \quad < \text{execute target program} \]

HWBP ENCOUNTERED  58 USEC
PC AF I IF BC DE HL DISASSEMBLY IX IY SP
0004 0200 0040 FFFF FFFF 0200 CPL FBFF FF5D 01F0 .
\[ \texttt{,T \_CR} \quad < \text{display history memory} \]
<table>
<thead>
<tr>
<th>OFFSET ADDR DB</th>
<th>DISASSEMBLY</th>
<th>TYPE</th>
<th>PB-----PA0</th>
<th>PB-----PA0</th>
</tr>
</thead>
<tbody>
<tr>
<td>-012 0000 3E</td>
<td>LD A,0</td>
<td>FETC</td>
<td>11111111</td>
<td>11111111</td>
</tr>
<tr>
<td>-011 0010 FF</td>
<td></td>
<td>RFSH</td>
<td>11111111</td>
<td>11111111</td>
</tr>
<tr>
<td>-010 0001 00</td>
<td></td>
<td>MRD</td>
<td>11111111</td>
<td>11111111</td>
</tr>
<tr>
<td>-00F 0002 3C</td>
<td>INC A</td>
<td>FETC</td>
<td>11111111</td>
<td>11111111</td>
</tr>
<tr>
<td>-00E 0011 FF</td>
<td></td>
<td>RFSH</td>
<td>11111111</td>
<td>11111111</td>
</tr>
<tr>
<td>-00D 0003 3C</td>
<td>INC A</td>
<td>FETC</td>
<td>11111111</td>
<td>11111111</td>
</tr>
<tr>
<td>-00C 0012 FF</td>
<td></td>
<td>RFSH</td>
<td>11111111</td>
<td>11111111</td>
</tr>
<tr>
<td>-00B 0004 2F</td>
<td>CPL</td>
<td>FETC</td>
<td>11111111</td>
<td>11111111</td>
</tr>
<tr>
<td>-00A 0013 FF</td>
<td></td>
<td>RFSH</td>
<td>11111111</td>
<td>11111111</td>
</tr>
<tr>
<td>-009 0005 3C</td>
<td>INC A</td>
<td>FETC</td>
<td>11111111</td>
<td>11111111</td>
</tr>
<tr>
<td>-008 0014 FF</td>
<td></td>
<td>RFSH</td>
<td>11111111</td>
<td>11111111</td>
</tr>
<tr>
<td>-007 0006 3C</td>
<td>INC A</td>
<td>FETC</td>
<td>11111111</td>
<td>11111111</td>
</tr>
<tr>
<td>-006 0015 FF</td>
<td></td>
<td>RFSH</td>
<td>11111111</td>
<td>11111111</td>
</tr>
<tr>
<td>-005 0007 2F</td>
<td>CPL</td>
<td>FETC</td>
<td>11111111</td>
<td>11111111</td>
</tr>
<tr>
<td>-004 0016 FF</td>
<td></td>
<td>RFSH</td>
<td>11111111</td>
<td>11111111</td>
</tr>
<tr>
<td>-003 0008 1B</td>
<td>JR -0AH</td>
<td>FETC</td>
<td>11111111</td>
<td>11111111</td>
</tr>
<tr>
<td>-002 0017 FF</td>
<td></td>
<td>RFSH</td>
<td>11111111</td>
<td>11111111</td>
</tr>
<tr>
<td>-001 0009 F6</td>
<td></td>
<td>MRD</td>
<td>11111111</td>
<td>11111111</td>
</tr>
<tr>
<td>+000 0000 3E</td>
<td>LD A,0</td>
<td>FETC</td>
<td>11111111</td>
<td>11111111</td>
</tr>
<tr>
<td>+001 0018 FF</td>
<td></td>
<td>RFSH</td>
<td>11111111</td>
<td>11111111</td>
</tr>
<tr>
<td>+002 0001 00</td>
<td></td>
<td>MRD</td>
<td>11111111</td>
<td>11111111</td>
</tr>
<tr>
<td>+003 0002 3C</td>
<td>INC A</td>
<td>FETC</td>
<td>11111111</td>
<td>11111111</td>
</tr>
<tr>
<td>+004 0019 FF</td>
<td></td>
<td>RFSH</td>
<td>11111111</td>
<td>11111111</td>
</tr>
<tr>
<td>+005 0003 3C</td>
<td>INC A</td>
<td>FETC</td>
<td>11111111</td>
<td>11111111</td>
</tr>
<tr>
<td>+006 001A FF</td>
<td></td>
<td>RFSH</td>
<td>11111111</td>
<td>11111111</td>
</tr>
</tbody>
</table>

END OF HISTORY

START OF HISTORY

<table>
<thead>
<tr>
<th>OFFSET ADDR DB</th>
<th>DISASSEMBLY</th>
<th>TYPE</th>
<th>PB-----PA0</th>
<th>PB-----PA0</th>
</tr>
</thead>
<tbody>
<tr>
<td>-012 0000 3E</td>
<td>LD A,0</td>
<td>FETC</td>
<td>11111111</td>
<td>11111111</td>
</tr>
<tr>
<td>-011 0010 FF</td>
<td></td>
<td>RFSH</td>
<td>11111111</td>
<td>11111111</td>
</tr>
<tr>
<td>-010 0001 00</td>
<td></td>
<td>MRD</td>
<td>11111111</td>
<td>11111111</td>
</tr>
<tr>
<td>-00F 0002 3C</td>
<td>INC A</td>
<td>FETC</td>
<td>11111111</td>
<td>11111111</td>
</tr>
</tbody>
</table>
-00E 0011 FF
-00D 0003 3C INC A
stop display with '.'
\texttt{,T 2(CR)} < display 2 samples, beginning at start of history

\begin{verbatim}
OFFS ADDR DB DISASSEMBLY TYPE PA8-----PA0 PB8-----PB0
-012 0000 3E LD A,0
-011 001/ FF
\texttt{,T 2,-4(CR)} < display 2 samples, beginning 4 samples before breakpoint
\end{verbatim}

\begin{verbatim}
OFFS ADDR DB DISASSEMBLY TYPE PA8-----PA0 PB8-----PB0
-004 0016 FF
-003 0008 18 JR -0AH
\texttt{,T 2,-4,E(CR)} < disable b probe printout
\end{verbatim}

\begin{verbatim}
OFFS ADDR DB DISASSEMBLY TYPE PA8-----PA0
-004 0016 FF
-003 0008 18 JR -0AH
\texttt{,T 2,-4,C(CR)} < disable all probe printout
\end{verbatim}

\begin{verbatim}
OFFS ADDR DB DISASSEMBLY TYPE
-004 0016 FF
-003 0008 18 JR -0AH
\texttt{,T 2,-4,8(CR)} < disable TYPE
\end{verbatim}

\begin{verbatim}
OFFS ADDR DB DISASSEMBLY
-004 0016 FF
-003 0008 18 JR -0AH
\texttt{,T 2,-4,0(CR)} < disable disassembly
\end{verbatim}
OFFS ADDR DB
-004 0016 FF
-003 0008 18_
\(T\) (CR) \(<\text{format is remembered for later T commands}\)

OFFS ADDR DB
-012 0000 3E
-011 0010 FF
-010 0001 00
-00F 0002 3C
-00E 0011 FF
-00D 0003 3C
-00C 0012 FF
-00B 0004 2F
-00A 0013 FF
-009 0005 3C
-008 0014 FF
-007 0006 3C
-006 0015 FF
-005 0007 2F
-004 0016 FF
-003 0008 18
-002 0017 FF
-001 0009 F6
+000 0000 3E
+001 0018 FF
+002 0001 00
+003 0002 3C
+004 0019 FF
+005 0003 3C
+006 001A FF
END OF HISTORY

START OF HISTORY
-012 0000 3E
-011 0010 FF
3C
3C
3F2 0050 FF
3F1 0009 F6
3F0 0000 3E
3EF 0051 FF
3EE 0001
3ED 0002 3C
3EC 0052 FF
3EB 0003 3C
3EA 0053 FF
3E9 0004 2F
3E8 0054 FF
3E7 0005 3C
3E6 0055 FF
3E5 0006 3C
3E4 0056 FF
3E3 0007 2F
3E2 0057 FF
3E1 0008 18
3E0 0058 FF
3DF 0009 F6
3DE 0000 3E
3DD 0059 FF . < stop printout with ' .'
3.65 **V-VERIFY COMMAND**

**FORMAT:**

```
,V [file]\(\text{CR}\)
```

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 AIM280, 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\(\text{CR}\)       \< get the test program
,M 0\(\text{CR}\)           \< change some locations
0000 3E\(\text{CR}\)
0001 00 AA\(\text{CR}\)
0002 3C\(\text{CR}\)
0003 3C AA\(\text{CR}\)
0004 2F\(\text{CR}\)
0005 3C 55\(\text{CR}\)
0006 3C\(\text{CR}\)
0007 2F 55\(\text{CR}\)
0008 18 -
,V \(\text{CR}\)             \< verify - expect the errors just introduced
*** ERROR 6D TARGET MEMORY COMPARE ***
0001 AA 00
```
3.68 W-WRITE TO ALTERNATE LUN COMMAND

FORMAT:

\[ \text{\underline{W} n} \text{(CR)} \quad \text{< Write to LUN n} \quad (1) \]
\[ \text{\underline{W}} \text{(CR)} \quad \text{< Disable Write} \quad (2) \]

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-80 DOS A command.

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

3.70 EXAMPLE, W COMMAND

\$A 5,FILE1(CR) \quad \text{< Assign FILE1 to LUN 5} \\
\$AIMZ80(CR) \quad \text{< Run AIMZ80.BIN} \\
AIMZ80 VERSION 1.0 \\
\text{\underline{W} 5} \text{(CR)} \quad \text{< Enable parallel output to LUN 5} \\
,

3.71 Z-INITIALIZE COMMAND

Format

\[ Z \text{<CR>} \]

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 0 to each location and check.
TEST 2 Write 0FFH to each location and check.
TEST 3 Write 055H and 0AAH 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 0 to 4000H will overwrite the AIMZ80A program itself, and, of course, won't work.

3.75 EXAMPLES, % COMMAND

```
,%, 0,FF

TEST 01
PASSED

TEST 02
PASSED

TEST 03
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 AIMZ80A program defaults to 1000nS (1.0 MHz) unless the user changes it. For 1000nS, 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-Z80AE can be used with the Mostek MD-SBC1 single board computer with no modifications or jumpers 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-Z80AE 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-Z80AE 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-80

AIM-Z80AE can also be used with the Mostek OEM-80 single board computer. Only one modification is required for this system. Memory in the OEM-80 system must be disabled for address space emulated with the Emulation RAM.

To insure proper operation with the OEM-80 (SDB-80), a jumper wire, 24 gauge or larger, should be installed on the circuit side of the OEM-80 between U63 pin 29, and the ground bus feedthrough located between U70 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 representative 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 0°C to +50°C

Target Power Supply Requirements (typical)
+5V 5% @ 500mA

System Power Supply Requirements (typical)
+5V 5% @ 2.5A
+12V 5% @ 100mA
-12V 10% @ 10mA

Interface - MATRIX and SYS-80F compatible

Operating Frequency - 500KHz to 4MHz (280 PHI clock)

Target Interface - All signals meet the specifications for the MK3880-4 (Z80A CPU) with the following exceptions:

1. The output low voltage is 0.5 V Max at 1.8 mA for the ADDRESS, DATA, IORQ, RFSH, HALT, and BUSAK signals.

2. The input low current is 400ua for the PHI clock, RESET, INT, NMI, and DATA signals.

3. The input high current is 20mA for the PHI clock, RESET, INT, NMI, and DATA signals.

4. The signals M1, MREQ, RD, and WR have a maximum of 25 ns added propagation delay.

5. The input signals RESET, INT and NMI have maximum of 45 ns added propagation delay.
APPENDIX C

SCHEMATIC DIAGRAMS

PARTS PLACEMENT DIAGRAMS

PARTS LISTS
PARTS PLACEMENT DIAGRAM

AIM-280 AE CONTROL MODULE
<table>
<thead>
<tr>
<th>PART NO.</th>
<th>QTY</th>
<th>DESCRIPTION</th>
<th>REFERENCE DESIGNATOR</th>
<th>USED ON</th>
</tr>
</thead>
<tbody>
<tr>
<td>4610173</td>
<td>1</td>
<td>AIM-XE CONTROL</td>
<td>AA:450-00470-00</td>
<td>XXXXX</td>
</tr>
<tr>
<td>0000001</td>
<td></td>
<td>450-00469-00</td>
<td>AZ:AIM-XE CONTROL</td>
<td>XXXXX</td>
</tr>
<tr>
<td>0000001</td>
<td></td>
<td>450-00471-00</td>
<td>AZ:AIM-XE CONTROL</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4150111</td>
<td>4e</td>
<td>CAPACITOR, 1UF</td>
<td>C 1-25</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4150140</td>
<td>4e</td>
<td>CAPACITOR, 1UF</td>
<td>C25-29</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4470073</td>
<td>1</td>
<td>DIODE, 4N751 5.1V</td>
<td>CR1</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4470059</td>
<td>1</td>
<td>RESISTOR, 1K</td>
<td>R1-6</td>
<td>XXXXX</td>
</tr>
<tr>
<td>9210057</td>
<td>2</td>
<td>CONNECTOR, EURO</td>
<td>SK1, SK2</td>
<td>XXXXX</td>
</tr>
<tr>
<td>2800007</td>
<td>5</td>
<td>STAKE PIN</td>
<td>TP1-6</td>
<td>XXXXX</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NOT USED</td>
<td>U 1, 4, 9, 17, 35, 22, 24</td>
<td></td>
</tr>
<tr>
<td>4313203</td>
<td>2</td>
<td>IC, 6N97</td>
<td>U 10, 11</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4313700</td>
<td>1</td>
<td>IC, MK4118-2</td>
<td>U 2</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4313504</td>
<td>1</td>
<td>IC, 74LS245</td>
<td>U 3, 4, 61</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4313585</td>
<td>1</td>
<td>IC, 74LS173</td>
<td>U 5</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4313507</td>
<td>4</td>
<td>IC, 74LS244</td>
<td>U 6, 7, 8, 40</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4313265</td>
<td>2</td>
<td>IC, 74S07</td>
<td>U17, 23</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4313509</td>
<td>8</td>
<td>IC, 74LS374</td>
<td>U13, 14, 15, 16, 20, 21</td>
<td></td>
</tr>
<tr>
<td>4313410</td>
<td>1</td>
<td>IC, 74S30</td>
<td>U19</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4313266</td>
<td>2</td>
<td>IC, 74S74</td>
<td>U22, 56</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4313267</td>
<td>1</td>
<td>IC, 74S22</td>
<td>U24</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4313205</td>
<td>2</td>
<td>IC, 74LS175</td>
<td>U25, 36</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4313271</td>
<td>1</td>
<td>IC, PROM 6301</td>
<td>U26</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4313234</td>
<td>1</td>
<td>IC, 74S00</td>
<td>U27</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4313287</td>
<td>3</td>
<td>IC, 74LS00</td>
<td>U28, 34, 59</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4313335</td>
<td>2</td>
<td>IC, 74LS393</td>
<td>U25, 40</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4313464</td>
<td>2</td>
<td>IC, 74LS367</td>
<td>U30, 63</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4313283</td>
<td>2</td>
<td>IC, 74LS08</td>
<td>U31, 39</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4313239</td>
<td>2</td>
<td>IC, 74S10</td>
<td>U32, 35</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4313513</td>
<td>2</td>
<td>IC, DELAY, 100</td>
<td>U33, 57</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4313302</td>
<td>2</td>
<td>IC, 74LS174</td>
<td>U37, 38</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4313264</td>
<td>4</td>
<td>IC, 74S157</td>
<td>U41, 42, 43, 51</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4313412</td>
<td>2</td>
<td>IC, 74LS38</td>
<td>U44, 45</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4313413</td>
<td>3</td>
<td>IC, 74LS74</td>
<td>U46, 52, 65</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4313300</td>
<td>1</td>
<td>IC, 74LS02</td>
<td>U47</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4313494</td>
<td>3</td>
<td>IC, 74S201</td>
<td>U49, 99, 50</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4313295</td>
<td>2</td>
<td>IC, 74LS138</td>
<td>U53, 54</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4313417</td>
<td>1</td>
<td>IC, 74LS86</td>
<td>U55, 57</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4313421</td>
<td>2</td>
<td>IC, 74LS32</td>
<td>U56, 60</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4313291</td>
<td>1</td>
<td>IC, 74LS14</td>
<td>U64</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4313929</td>
<td>15</td>
<td>IC, MK4116-2</td>
<td>U65, 81</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4470179</td>
<td>2</td>
<td>SIP+6PIN+5RES+1K</td>
<td>UR1+UR2</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4470273</td>
<td>3</td>
<td>SIP+6PIN+4RES+47</td>
<td>UR3, 45, 55</td>
<td>XXXXX</td>
</tr>
<tr>
<td>620017</td>
<td>17</td>
<td>SOCKET 15 PIN</td>
<td>XXX</td>
<td>XXXXX</td>
</tr>
<tr>
<td>620013</td>
<td>2</td>
<td>SOCKET 24 PIN</td>
<td>XXX</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4210244</td>
<td>7</td>
<td>MINI JUMPER</td>
<td>21-7</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4210282</td>
<td>2</td>
<td>CONN HDR RA 40PIN W/EJECT</td>
<td>J1-2</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4210291</td>
<td>1</td>
<td>CONN HDR RA 10PIN W/EJECT</td>
<td>J3</td>
<td>XXXXX</td>
</tr>
</tbody>
</table>
## PARTS LIST

### HISTORY MODULE

<table>
<thead>
<tr>
<th>PART NO.</th>
<th>QTY</th>
<th>DESCRIPTION</th>
<th>REFERENCE DESIGNATOR</th>
<th>USED ON</th>
</tr>
</thead>
<tbody>
<tr>
<td>4510453</td>
<td>1</td>
<td>FM 450-0043000 REV</td>
<td>A1: 450-03-35-00</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>3000002</td>
<td>1</td>
<td>SC 450-0043600 REV</td>
<td>A2: AIM-XE HISTORY</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>00000011</td>
<td>1</td>
<td>ASSY 450-0043500 REV</td>
<td>A1: AIM-XE HISTORY</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>4130111</td>
<td>21</td>
<td>CAPACITOR 0.1UF</td>
<td>C1-C21</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>4130114</td>
<td>1</td>
<td>CAPACITOR 0.63UF</td>
<td>C22</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>4210153</td>
<td>2</td>
<td>HEADER 10 PIN RT ANGLE</td>
<td>J1*2 SEE NOTE 1</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>4210217</td>
<td>1</td>
<td>HEADER 10 PIN RT ANGLE</td>
<td>J3</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>4470393</td>
<td>4</td>
<td>RESISTOR 58.0</td>
<td>R1-5 05</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>4470393</td>
<td>1</td>
<td>RESISTOR 4.7K</td>
<td>R4</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>4210037</td>
<td>2</td>
<td>CONNECTOR EURO</td>
<td>SK10SK2</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>4210037</td>
<td>7</td>
<td>STAKE PIN</td>
<td>TP1-4</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>4313333</td>
<td>6</td>
<td>IC 74LS165</td>
<td>U13 15 57 9 11</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>4313333</td>
<td>5</td>
<td>IC 74LS7473</td>
<td>U28 0 0 10 12</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>4313333</td>
<td>6</td>
<td>IC 74LS118-2</td>
<td>U13 14 15 16 17 18</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>4313333</td>
<td>5</td>
<td>IC 74LS193</td>
<td>U19 35 45 47 49</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>4313333</td>
<td>7</td>
<td>IC 74LS546</td>
<td>U20 22 23 25 26 24 49</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>4313333</td>
<td>4</td>
<td>IC 74LS164</td>
<td>U24 35 37 65</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>4313244</td>
<td>2</td>
<td>IC 74LS30</td>
<td>U29 55</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>4313244</td>
<td>2</td>
<td>IC 74LS244</td>
<td>U30 72</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>4313412</td>
<td>8</td>
<td>IC 74LS38</td>
<td>U32 34 33 37 39 40 51 56</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>4313293</td>
<td>3</td>
<td>IC 74LS98</td>
<td>U41 57 55 57 55 53</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>4313353</td>
<td>2</td>
<td>IC 74LS151</td>
<td>U42 33</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>4313353</td>
<td>2</td>
<td>IC 74LS87</td>
<td>U43 67</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>4313353</td>
<td>4</td>
<td>IC 74LS47</td>
<td>U44 55 57 34</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>4313353</td>
<td>12</td>
<td>IC 74LS125</td>
<td>U45 73</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>4313353</td>
<td>1</td>
<td>IC 74LS10</td>
<td>U50</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>4313245</td>
<td>1</td>
<td>IC 74LS04</td>
<td>U54</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>4313245</td>
<td>1</td>
<td>IC 74LS04</td>
<td>U55</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>4313296</td>
<td>2</td>
<td>IC 74LS133</td>
<td>U56 51</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>4313555</td>
<td>2</td>
<td>IC 74LS174</td>
<td>U57 90</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>4313555</td>
<td>1</td>
<td>IC 74LS32</td>
<td>U63</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>4313411</td>
<td>1</td>
<td>IC 74LS32</td>
<td>U69</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>4313232</td>
<td>1</td>
<td>IC 74LS20</td>
<td>U60</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>4313344</td>
<td>1</td>
<td>IC 74LS495</td>
<td>U71</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>4313321</td>
<td>1</td>
<td>IC 74LS14</td>
<td>U74</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>4470174</td>
<td>1</td>
<td>SIP 10 PIN RES 1K</td>
<td>U41</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>4520018</td>
<td>6</td>
<td>SOCKET 24 PIN</td>
<td>X13-18</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>4210244</td>
<td>5</td>
<td>MINI JUMPER</td>
<td>71-5</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>0000005</td>
<td>1</td>
<td>NOTE 1</td>
<td>2: CUT PIN 22 OF J1</td>
<td>AIMXHE</td>
</tr>
<tr>
<td>0000004</td>
<td>1</td>
<td>NOTE 1</td>
<td>2: CUT PIN 2 OF J2</td>
<td>AIMXHE</td>
</tr>
</tbody>
</table>
NOTE: NOT SHOWN, ALL BE TIED TO GND.

UNLIZED INPUTS
$=R=I=5V
<table>
<thead>
<tr>
<th>PART NO.</th>
<th>QTY</th>
<th>DESCRIPTION</th>
<th>REFERENCE DESIGNATOR</th>
<th>USED ON</th>
</tr>
</thead>
<tbody>
<tr>
<td>2065000</td>
<td>1</td>
<td>CABLE ASSY, TARGET</td>
<td>1</td>
<td>50-00650-00</td>
</tr>
<tr>
<td>2065100</td>
<td>1</td>
<td>CABLE ASSY, J1 BD INTERCONNECT</td>
<td>2</td>
<td>50-00651-00</td>
</tr>
<tr>
<td>2065200</td>
<td>1</td>
<td>CABLE ASSY, J2 BD INTERCONNECT</td>
<td>3</td>
<td>50-00652-00</td>
</tr>
<tr>
<td>2065300</td>
<td>1</td>
<td>CABLE ASSY, J3 BD INTERCONNECT</td>
<td>4</td>
<td>50-00653-00</td>
</tr>
<tr>
<td>2064900</td>
<td>2</td>
<td>CABLE ASSY, TEST PROBES</td>
<td>5</td>
<td>50-00649-00</td>
</tr>
<tr>
<td>2061700</td>
<td>1</td>
<td>AIM Z80A PCB</td>
<td>6</td>
<td>50-00417-00</td>
</tr>
<tr>
<td>2047000</td>
<td>1</td>
<td>AIM-XE CONTROL PCB ASSY</td>
<td>7</td>
<td>50-00470-00</td>
</tr>
<tr>
<td>2043500</td>
<td>1</td>
<td>AIM-XE HISTORY PCB ASSY</td>
<td>8</td>
<td>50-00435-00</td>
</tr>
<tr>
<td>4140342</td>
<td>1</td>
<td>BASE</td>
<td>9</td>
<td>50-00506-00</td>
</tr>
<tr>
<td>4140341</td>
<td>1</td>
<td>COVER</td>
<td>10</td>
<td>50-00505-00</td>
</tr>
<tr>
<td>5025974</td>
<td>1</td>
<td>NAMEPLATE</td>
<td>11</td>
<td>50-00518-00</td>
</tr>
<tr>
<td>4280051</td>
<td>4</td>
<td>FOOT, RUBBER</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>4210343</td>
<td>8</td>
<td>SCREW, #4-40 X 1/4 PPH 80</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>4210367</td>
<td>4</td>
<td>SPACER, HEX, #4-40 X 1/4, W/STUD</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>4210368</td>
<td>4</td>
<td>SPACER, HEX, #4-40 X 3/4, TAPPED</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>4210364</td>
<td>4</td>
<td>WASHER, #4 SLOTTED</td>
<td>16</td>
<td></td>
</tr>
</tbody>
</table>
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

Copyright 1980 by Mostek Corporation
All rights reserved