SBC 80/10 AND SBC 80/10A
SINGLE BOARD COMPUTER
HARDWARE REFERENCE MANUAL

Manual Order Number: 9800230F
The information in this document is subject to change without notice.

Intel Corporation makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Intel Corporation assumes no responsibility for any errors that may appear in this document. Intel Corporation makes no commitment to update nor to keep current the information contained in this document.

No part of this document may be copied or reproduced in any form or by any means without the prior written consent of Intel Corporation.

The following are trademarks of Intel Corporation and may be used only to describe Intel products:

ICE-30  MCS
ICE-80  MEGACHASSIS
INSITE  MICROMAP
INTEL  MULTIBUS
INTELLEC  PROMPT
LIBRARY  UPI
          uSCOPE
PREFACE

This manual provides general information, installation, programming information, principles of operation, and service information for the Intel SBC 80/10 and SBC 80/10A Single Board Computers. Unless specified otherwise, references to the SBC 80/10 are valid for both systems. The areas where differences occur are identified as "SBC 80/10 only" or "SBC 80/10A only." Additional systems information and component part details are available in the following documents:

- Intel Microcomputer Systems Data Book, Part No. 98-414
- Intel Multibus Interfacing Application Note, AP-28
- Intel 8255 Programmable Peripheral Interface Application Note, AP-15
- Intel 8251 Universal Synchronous/Asynchronous Receiver/Transmitter Application Note, AP-16
## TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER 1</th>
<th>PAGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTRODUCTION</td>
<td>1-1</td>
</tr>
<tr>
<td>FUNCTIONAL DESCRIPTION</td>
<td>2-1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CHAPTER 3</th>
<th>PAGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>THEORY OF OPERATION</td>
<td>3-1</td>
</tr>
<tr>
<td>THE CPU SET</td>
<td>3-1</td>
</tr>
<tr>
<td>3.1.1 INSTRUCTION TIMING</td>
<td>3-3</td>
</tr>
<tr>
<td>3.1.2 INTERRUPT SEQUENCES</td>
<td>3-12</td>
</tr>
<tr>
<td>3.1.3 HOLD SEQUENCES</td>
<td>3-13</td>
</tr>
<tr>
<td>3.1.4 HALT SEQUENCES</td>
<td>3-13</td>
</tr>
<tr>
<td>3.1.5 START-UP SEQUENCE</td>
<td>3-14</td>
</tr>
<tr>
<td>SYSTEM BUS INTERFACE LOGIC</td>
<td>3-15</td>
</tr>
<tr>
<td>3.2.1 SYSTEM CONTROL SIGNAL LOGIC</td>
<td>3-15</td>
</tr>
<tr>
<td>3.2.2 SYSTEM BUS DRIVERS</td>
<td>3-18</td>
</tr>
<tr>
<td>3.2.3 FAILSAFE TIMER</td>
<td>3-20</td>
</tr>
<tr>
<td>RANDOM ACCESS MEMORY (RAM)</td>
<td>3-21</td>
</tr>
<tr>
<td>3.3.1 SBC 80/10 RAM</td>
<td>3-22</td>
</tr>
<tr>
<td>3.3.2 SBC 80/10A RAM</td>
<td>3-23</td>
</tr>
<tr>
<td>READ ONLY MEMORY (ROM/EPROM)</td>
<td>3-24</td>
</tr>
<tr>
<td>SERIAL I/O INTERFACE</td>
<td>3-27</td>
</tr>
<tr>
<td>3.5.1 INTEL 8251 OPERATIONAL SUMMARY</td>
<td>3-28</td>
</tr>
<tr>
<td>3.5.2 SERIAL I/O CONFIGURATIONS</td>
<td>3-40</td>
</tr>
<tr>
<td>3.5.3 BAUD RATE CLOCK GENERATION</td>
<td>3-43</td>
</tr>
<tr>
<td>3.5.4 SERIAL I/O INTERRUPTS</td>
<td>3-44</td>
</tr>
<tr>
<td>PARALLEL I/O INTERFACE</td>
<td>3-45</td>
</tr>
<tr>
<td>3.6.1 INTEL 8255 OPERATIONAL SUMMARY</td>
<td>3-45</td>
</tr>
<tr>
<td>3.6.2 PARALLEL I/O CONFIGURATIONS</td>
<td>3-59</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CHAPTER 4</th>
<th>PAGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>USER SELECTABLE OPTIONS</td>
<td>4-1</td>
</tr>
<tr>
<td>SERIAL I/O INTERFACE OPTIONS</td>
<td>4-2</td>
</tr>
<tr>
<td>4.1.1 INTERFACE TYPE</td>
<td>4-2</td>
</tr>
<tr>
<td>4.1.2 BAUD RATE AND PROGRAM-SELECTABLE SERIAL I/O OPTIONS</td>
<td>4-2</td>
</tr>
<tr>
<td>PARALLEL I/O OPTIONS</td>
<td>4-6</td>
</tr>
<tr>
<td>4.2.1 PORT 1 (GROUP 1 PORT A)</td>
<td>4-7</td>
</tr>
<tr>
<td>4.2.2 PORT 2 (GROUP 1 PORT B)</td>
<td>4-12</td>
</tr>
<tr>
<td>4.2.3 PORT 3 (GROUP 1 PORT C)</td>
<td>4-14</td>
</tr>
<tr>
<td>4.2.4 PORT 4 AND 5 (GROUP 2 PORTS A AND B)</td>
<td>4-15</td>
</tr>
<tr>
<td>4.2.5 PORT 6 (GROUP 2 PORT C)</td>
<td>4-18</td>
</tr>
<tr>
<td>GENERAL OPTIONS</td>
<td>4-22</td>
</tr>
<tr>
<td>4.3.1 SYSTEM RESET OUTPUT</td>
<td>4-22</td>
</tr>
<tr>
<td>4.3.2 DISABLE BUS CLOCK SIGNALS</td>
<td>4-22</td>
</tr>
<tr>
<td>4.3.3 ACKNOWLEDGE INPUTS</td>
<td>4-22</td>
</tr>
<tr>
<td>4.3.4 INTERRUPT SOURCES</td>
<td>4-26</td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS (Continued)

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4 DEFAULT OPTIONS</td>
<td>4-26</td>
</tr>
<tr>
<td>4.5 JUMPER CONFIGURATION FOR ROM/EPROM INSTALLATION</td>
<td>4-28</td>
</tr>
</tbody>
</table>

## CHAPTER 5
SYSTEM INTERFACING
- 5.1 ELECTRICAL CONNECTIONS                                             | 5-1  |
- 5.2 EXTERNAL SBC 80/10 SYSTEM BUS SUMMARY                                | 5-7  |
- 5.3 RS232C CABLING                                                     | 5-11 |
- 5.4 TELETYPewriter MODIFICATION                                         | 5-13 |

## CHAPTER 6
INTERFACING TO MULTIBUS MASTERS                                         | 6-1  |

## CHAPTER 7
SPECIFICATIONS
- 7.1 DC POWER REQUIREMENTS                                             | 7-1  |
- 7.2 AC CHARACTERISTICS                                                | 7-1  |
- 7.3 DC CHARACTERISTICS                                                | 7-1  |
- 7.4 ENVIRONMENT                                                       | 7-1  |
- 7.5 BOARD OUTLINE                                                     | 7-2  |
- 7.6 COMPATIBLE CONNECTORS                                             | 7-2  |

## APPENDIX A
SCHEMATICS                                                              | A-1  |

## APPENDIX B
TELETYPewriter MODIFICATIONS                                           | B-1  |

## APPENDIX C
8080 INSTRUCTION SET SUMMARY                                           | C-1  |

## APPENDIX D
SBC-901, SBC-902 SCHEMATICS                                            | D-1  |
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1</td>
<td>FUNCTIONAL BLOCK DIAGRAM</td>
<td>2-2</td>
</tr>
<tr>
<td>3-0</td>
<td>THE CPU SET</td>
<td>3-2</td>
</tr>
<tr>
<td>3-1</td>
<td>TYPICAL FETCH MACHINE CYCLE</td>
<td>3-6</td>
</tr>
<tr>
<td>3-2</td>
<td>INPUT INSTRUCTION CYCLE</td>
<td>3-9</td>
</tr>
<tr>
<td>3-3</td>
<td>OUTPUT INSTRUCTION CYCLE</td>
<td>3-10</td>
</tr>
<tr>
<td>3-4</td>
<td>READY TIMING</td>
<td>3-19</td>
</tr>
<tr>
<td>3-5</td>
<td>RAM ACCESS TIMING</td>
<td>3-26</td>
</tr>
<tr>
<td>3-6</td>
<td>8251 PIN ASSIGNMENTS</td>
<td>3-29</td>
</tr>
<tr>
<td>3-7</td>
<td>TYPICAL 8251 DATA BLOCK</td>
<td>3-33</td>
</tr>
<tr>
<td>3-8</td>
<td>ASYNCHRONOUS MODE</td>
<td>3-35</td>
</tr>
<tr>
<td>3-9</td>
<td>SYNCHRONOUS MODE</td>
<td>3-35</td>
</tr>
<tr>
<td>3-10</td>
<td>COMMAND INSTRUCTION FORMAT</td>
<td>3-36</td>
</tr>
<tr>
<td>3-11</td>
<td>STATUS READ FORMAT</td>
<td>3-37</td>
</tr>
<tr>
<td>3-12</td>
<td>8255 PIN ASSIGNMENTS</td>
<td>3-47</td>
</tr>
<tr>
<td>3-13</td>
<td>MODE DEFINITION CONTROL WORD FORMAT</td>
<td>3-47</td>
</tr>
<tr>
<td>3-14</td>
<td>BIT SET/RESET CONTROL WORD FORMAT</td>
<td>3-49</td>
</tr>
<tr>
<td>3-15</td>
<td>8255 MODE 0 TIMING</td>
<td>3-50</td>
</tr>
<tr>
<td>3-16</td>
<td>EXAMPLES OF MODE 0 CONFIGURATION</td>
<td>3-50</td>
</tr>
<tr>
<td>3-17</td>
<td>MODE 1 INPUT CONFIGURATION</td>
<td>3-53</td>
</tr>
<tr>
<td>3-18</td>
<td>8255 MODE 1 INPUT TIMING</td>
<td>3-53</td>
</tr>
<tr>
<td>3-19</td>
<td>MODE 1 OUTPUT CONFIGURATION</td>
<td>3-54</td>
</tr>
<tr>
<td>3-20</td>
<td>MODE 1 BASIC OUTPUT TIMING</td>
<td>3-54</td>
</tr>
<tr>
<td>3-21</td>
<td>MODE 2 PORT CONFIGURATION</td>
<td>3-56</td>
</tr>
<tr>
<td>3-22</td>
<td>MODE 2 TIMING</td>
<td>3-57</td>
</tr>
<tr>
<td>4-1</td>
<td>READ COMMAND WITH XACK/</td>
<td>4-23</td>
</tr>
<tr>
<td>4-2</td>
<td>READ COMMAND WITH AACK/</td>
<td>4-24</td>
</tr>
<tr>
<td>4-3</td>
<td>WRITE COMMAND WITH AACK/</td>
<td>4-25</td>
</tr>
<tr>
<td>5-1</td>
<td>EDGE CONNECTORS</td>
<td>5-4</td>
</tr>
<tr>
<td>6-1</td>
<td>SERIAL PRIORITY CONFIGURATION WITH ANOTHER</td>
<td>6-2</td>
</tr>
<tr>
<td></td>
<td>MULTIBUS MASTER</td>
<td></td>
</tr>
<tr>
<td>7-1</td>
<td>MEMORY AND I/O READ TIMING (CONTINUOUS BUS</td>
<td>7-9</td>
</tr>
<tr>
<td></td>
<td>CONTROL)</td>
<td></td>
</tr>
<tr>
<td>7-2</td>
<td>MEMORY AND I/O WRITE TIMING (CONTINUOUS BUS</td>
<td>7-10</td>
</tr>
<tr>
<td></td>
<td>CONTROL)</td>
<td></td>
</tr>
<tr>
<td>7-3</td>
<td>BUS EXCHANGE (WRITE)</td>
<td>7-11</td>
</tr>
<tr>
<td>7-4</td>
<td>SBC 80/10 DIMENSION DRAWING</td>
<td>7-12</td>
</tr>
<tr>
<td>A-1</td>
<td>SBC 80/10 SCHEMATIC</td>
<td>A-2</td>
</tr>
<tr>
<td>A-2</td>
<td>SBC 80/10A SCHEMATIC</td>
<td>A-7</td>
</tr>
<tr>
<td>B-1</td>
<td>TELETYPETYPE COMPONENT LAYOUT</td>
<td>B-2</td>
</tr>
<tr>
<td>B-2</td>
<td>CURRENT SOURCE RESISTOR</td>
<td>B-2</td>
</tr>
<tr>
<td>B-3</td>
<td>TERMINATION BLOCK</td>
<td>B-2</td>
</tr>
<tr>
<td>B-4</td>
<td>TELETYPERWRITER MODIFICATIONS</td>
<td>B-3</td>
</tr>
<tr>
<td>B-5</td>
<td>RELAY CIRCUITS</td>
<td>B-3</td>
</tr>
<tr>
<td>B-6</td>
<td>MODE SWITCH</td>
<td>B-3</td>
</tr>
<tr>
<td>B-7</td>
<td>DISTRIBUTOR TRIP MAGNET</td>
<td>B-4</td>
</tr>
<tr>
<td>B-8</td>
<td>TTY ADAPTER CABLELING</td>
<td>B-4</td>
</tr>
</tbody>
</table>
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Number</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-0</td>
<td>3-41</td>
</tr>
<tr>
<td>3-1</td>
<td>3-58</td>
</tr>
<tr>
<td>3-2</td>
<td>3-60</td>
</tr>
<tr>
<td>3-3</td>
<td>3-60</td>
</tr>
<tr>
<td>3-4</td>
<td>3-65</td>
</tr>
<tr>
<td>4-1</td>
<td>4-3</td>
</tr>
<tr>
<td>4-2a</td>
<td>4-4</td>
</tr>
<tr>
<td>2-4b</td>
<td>4-4</td>
</tr>
<tr>
<td>4-3</td>
<td>4-6</td>
</tr>
<tr>
<td>4-4</td>
<td>4-8</td>
</tr>
<tr>
<td>4-5</td>
<td>4-8</td>
</tr>
<tr>
<td>4-6</td>
<td>4-9</td>
</tr>
<tr>
<td>4-7</td>
<td>4-10</td>
</tr>
<tr>
<td>4-8</td>
<td>4-11</td>
</tr>
<tr>
<td>4-9</td>
<td>4-12</td>
</tr>
<tr>
<td>4-10</td>
<td>4-13</td>
</tr>
<tr>
<td>4-11</td>
<td>4-13</td>
</tr>
<tr>
<td>4-12</td>
<td>4-14</td>
</tr>
<tr>
<td>4-13</td>
<td>4-15</td>
</tr>
<tr>
<td>4-14</td>
<td>4-16</td>
</tr>
<tr>
<td>4-15</td>
<td>4-16</td>
</tr>
<tr>
<td>4-16</td>
<td>4-17</td>
</tr>
<tr>
<td>4-17</td>
<td>4-17</td>
</tr>
<tr>
<td>4-18</td>
<td>4-18</td>
</tr>
<tr>
<td>4-19</td>
<td>4-19</td>
</tr>
<tr>
<td>4-20</td>
<td>4-20</td>
</tr>
<tr>
<td>4-21</td>
<td>4-21</td>
</tr>
<tr>
<td>4-22</td>
<td>4-20</td>
</tr>
<tr>
<td>4-23</td>
<td>4-20</td>
</tr>
<tr>
<td>4-24</td>
<td>4-27</td>
</tr>
<tr>
<td>4-25</td>
<td>4-28</td>
</tr>
<tr>
<td>4-26</td>
<td>4-28</td>
</tr>
<tr>
<td>5-1</td>
<td>5-3</td>
</tr>
<tr>
<td>5-2</td>
<td>5-3</td>
</tr>
<tr>
<td>5-3</td>
<td>5-5</td>
</tr>
<tr>
<td>5-4</td>
<td>5-6</td>
</tr>
<tr>
<td>5-5</td>
<td>5-7</td>
</tr>
<tr>
<td>5-6</td>
<td>5-10</td>
</tr>
<tr>
<td>7-1</td>
<td>7-2</td>
</tr>
<tr>
<td></td>
<td>Table Description</td>
</tr>
<tr>
<td>---</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>7-2</td>
<td>AC CHARACTERISTICS WITH BUS EXCHANGE</td>
</tr>
<tr>
<td>7-3</td>
<td>AC CHARACTERISTICS WITH CONTINUOUS BUS CONTROL</td>
</tr>
<tr>
<td>7-4</td>
<td>DC CHARACTERISTICS</td>
</tr>
<tr>
<td>7-5</td>
<td>COMPATIBLE CONNECTOR HARDWARE</td>
</tr>
</tbody>
</table>
CHAPTER 1
INTRODUCTION

The SBC 80/10 and SBC 80/10A are members of Intel's complete line of OEM computer systems which take full advantage of Intel's LSI technology to provide economical, self-contained computer based solutions for OEM applications. The SBC 80/10 and SBC 80/10A are complete computer systems, each on a single 6.75-by-12 inch printed circuit card. The CPU, system clock, read/write memory, non-volatile read-only-memory, I/O ports and drivers, serial communications interface, bus control logic and drivers all reside on the board.

Throughout this manual, reference to the SBC 80/10 are valid for both the SBC 80/10 and SBC 80/10A. The areas where differences occur are identified as "SBC 80/10 only" and SBC 80/10A only."

Intel's powerful 8-bit n-channel MOS 8080A CPU, fabricated on a single LSI chip, is the central processor for the SBC 80/10 and SBC 80/10A. The 8080A contains six 8-bit general purpose registers and an accumulator. The six general purpose registers may be addressed individually or in pairs providing both single and double precision operators. The 8080A has a 16-bit program counter which allows direct addressing of up to 64K of memory. An external stack, located within any portion of memory, may be used as a last in/first out stack to store and retrieve the contents of the program counter, flags, accumulator and all of the six general purpose registers. A sixteen bit stack pointer controls the addressing of this external stack. This stack provides almost unlimited subroutine nesting. Sixteen-line
address and eight-line bi-directional data busses are used to facilitate easy interface to memory and I/O.

The powerful 8080A instruction set allows the user to write efficient programs in a minimum amount of time. The accumulator group instructions include arithmetic and logical operators with direct, register direct, and immediate addressing modes. Move, load, and store instruction groups provide the ability to move either 8 or 16 bits of data between memory, the six working registers and the accumulator using all addressing modes. The ability to branch to different portions of a program is provided with jump, jump conditional, and computed jumps. The ability to conditionally and unconditionally call to and return from subroutines is provided. The RESTART (or single byte call instruction) is used for interrupt operation. Double precision operators such as stack manipulation and double add instructions extend both the arithmetic and interrupt handling capability of the 8080A. The ability to increment and decrement memory, the six general registers, and the accumulator is provided as well as extended increment and decrement instructions to operate on the register pairs and stack pointer.

The difference between the SBC 80/10 and SBC 80/10A is in the type and quantity of memory available on each board.

The SBC 80/10 contains 1K 8-bit words of read/write memory using Intel's 8111 Low Power Static RAMs. Sockets for up to 4K 8-bit words of non-volatile read-only memory may be added in 1K byte increments using Intel's 8708 Erasable and Electrically Reprogrammable ROMs (EPROMs) or Intel's 8308 Metal Masked ROMs.
The SBC 80/10A contains 1K 8-bit words of read/write memory using Intel's 8102 Low Power Static RAMs. Sockets for up to 4K or 8K words of non-volatile read-only memory are provided on the SBC 80/10A. Up to 4K of read-only memory may be added in 1K byte increments using Intel's 8708 Erasable and Electrically Reprogrammable ROMs (EPROMs), Intel's 2758 Erasable and Electrically Reprogrammable ROMs (EPROMs), or Intel's 8308 Metal Masked ROMs. Optionally up to 8K words of read-only memory may be added in 2K byte increments using Intel's 2716 Erasable and Electrically Reprogrammable ROMs (EPROMs) or Intel's 2316E Metal Masked ROMs.

The SBC 80/10 contains 48 programmable parallel I/O lines implemented using two Intel 8255 Programmable Peripheral Interface devices. The software is used to configure the I/O lines in combinations of uni-directional input/output, and bidirectional ports. Therefore, the I/O interface may be customized to meet specified peripheral requirements. In order to take full advantage of the large number of possible I/O configurations, sockets are provided for interchangable I/O line drivers and terminators. Hence, the flexibility of the I/O interface is further enhanced by the capability of selecting the appropriate optional line drivers and terminators for each application.

A programmable serial communications interface using Intel's 8251 Universal Synchronous/Asynchronous Receiver/Transmitter (USART) is contained on the board. The USART can be programmed by the systems software to provide virtually any serial data transmission technique presently in use (including IBM Bi-Sync). The mode of operation (i.e., synchronous or asynchronous), data format, control character format, parity, and asynchronous serial transmission rate (within limitations
given later) are all under program control). The 8251 provides full du-
plex, double buffered transmission and receive capability. Parity, over-
run, and framing error detection are all incorporated in the USART. The
inclusion of jumper selectable teletype, or RS232C compatible interfaces
on the board in conjunction with the USART provide a direct interface
to a teletype, CRT, RS232C compatible devices, and asynchronous and
synchronous modems.

A single-level interrupt may originate from any one of six sources
including the USART, Programmable I/O interface, and two user designated
interrupt request lines. When an interrupt request is recognized, a
RESTART 7 instruction is generated. The processor responds by suspend-
ing program execution and executing a user defined interrupt service
routine originating at location 3816.

Memory and I/O expansion may be achieved using standard Intel
boards. Memory may be expanded to 65,536 bytes by adding user speci-
fied combinations of SBC 016 16K byte RAM board, SBC 406 6K byte and
SBC 416 16K byte PROM boards. Input/output capacity may be expanded
in increments of 4 input ports and 4 output ports using SBC 508 Input/
Output boards. Expandable backplanes and cardcages are available to
support multi-board systems.

The development cycle of SBC 80/10 based OEM products may be
significantly reduced using the Intellec Microcomputer Development
System. The resident assembler, text editor, and system monitor
greatly simplify the design, development, and debug of SBC 80/10
based system software. A unique In-Circuit Emulator (ICE-80)
option provides the capability of executing and debugging OEM system
software directly on the SBC 80/10.
Intel's high-level language, PL/M, can be used to significantly decrease the time required to develop OEM system software.
CHAPTER 2

FUNCTIONAL DESCRIPTION

For descriptive purposes, the circuitry on the SBC-80/10 can be divided into six functional blocks:

1) CPU Set
2) System Bus Interface
3) Random Access Memory (RAM)
4) Read Only Memory (ROM/PROM) Logic
5) Serial I/O Interface
6) Parallel I/O Interface

as shown in Figure 2-1.

The CPU Set consists of the 8080A Control Processor, the 8224 Clock Generator and the 8238 System Controller. The CPU Set is the heart of the SBC-80/10. It performs all system processing functions and provides a stable timing reference for all other circuitry in the system. The CPU Set generates all of the address and control signals necessary to access memory and I/O ports both on the SBC-80/10 and external to the SBC-80/10. The CPU Set is capable of fetching and executing any of the 8080's seventy-eight instructions. The CPU Set responds to interrupt requests originating both on and off the SBC-80/10, to HOLD requests from modules wishing to acquire control of the system bus, and to WAIT requests from memory or I/O devices having an access time which is slower than the 8080's cycle time.

The System Bus Interface includes an assortment of circuitry which gates interrupt requests, HOLD requests, READY (no wait inputs and the system reset input to the appropriate pins of the CPU Set. Other circuits drive the various external system control signals. The
FIGURE 2-1. FUNCTIONAL BLOCK DIAGRAM
System Bus Interface also includes two 8216 bi-directional bus drivers which drive the memory data bus on the SBC-80/10. Six 8226 devices drive the external system data and address busses.

The Random Access Memory (RAM) section provides the SBC 80/10 and SBC 80/10A user with 1024 X 8-bits of on board read/write storage. Eight Intel 8111 Low Power Static RAMs (256 x 4-bit each) are mounted on the SBC 80/10. The SBC 80/10A has eight Intel 8102 Low Power Static RAM chips (1024 x 1-bit each). Both boards contain the necessary acknowledge and memory address decoding logic.

The Read Only Memory (ROM/EPROM) section provides the user with the necessary provisions for installing up to 4096 x 8-bits of ROM or EPROM on the SBC 80/10 and up to 8192 x 8-bits of ROM or EPROM on the SBC 80/10A. The 80/10 and 80/10A have four 24-pin sockets that can accept either Intel 8708 Erasable and Electrically Reprogrammable ROM (EPROM) chips, or Intel 8308 Metal Masked ROM chips. Optionally, the SBC 80/10A accepts Intel 2716 Erasable and Electrically Reprogrammable ROM (EPROM) chips, Intel 2758 Erasable and Electrically Reprogrammable ROM (EPROM) chips, or Intel 2316E Metal Masked ROM chips. The total ROM/EPROM memory capacity using 8208, 8308 or 2758 chips is 4K x 8-bits and 8K x 8-bits using 2716 or 2316 E chips. Both the 80/10 and 80/10A boards include the necessary acknowledge and memory address decoding circuitry.

The Serial I/O Interface, using Intel's 8251 USART device, provides a bi-directional serial data communications channel that can be programmed to operate with most of the current serial data transmission protocols. Synchronous or asynchronous mode, baud rate, character length, number of stop bits and the choice of even, odd or no parity are all program selectable. The user also has the option of configuring the Serial I/O Interface as an EIA RS232 interface or as a Teletype-compatible current loop interface.
The Parallel I/O Interface, using two Intel 8255 Programmable Peripheral interface devices, provides 48 signal lines for the transfer and control of data to or from peripheral devices. Eight lines already have a bidirectional driver and termination network permanently installed. This bidirectional network allows these eight lines to be inputs, outputs, or bidirectional (selected via jumpers). The remaining 40 lines, however, are uncommitted. Sockets are provided for the installation of drivers or termination networks as required to meet the specific needs of the user system.
CHAPTER 3

THEORY OF OPERATION

In the preceding chapter we introduced each of the SEC-SO/10 functional blocks and defined what each block was capable of doing. In this chapter we shall go one step further and describe how each block performs its particular function(s). The text will constantly refer to the SEC-SO/10 schematics, provided in Appendix A.

Note: Both active-high (positive true) and active-low (negative true) signals appear on the SEC-SO/10 schematics. To eliminate any confusion when reading this chapter, the following convention will be adhered to: whenever a signal is active-low, its mnemonic is followed by a slash; for example, MRDC/ means that the level on that line will be low when the memory read command is true (active). If the signal is subsequently inverted, thus making it active-high, the slash is omitted; for example, MRDC means that the level on that line will be high when the memory read command is true.

3.1 THE CPU SET

The CPU Set consists of three Intel® integrated circuit devices:

* 8080A Central Processor Unit
* 8224 Clock Generator
* 8238 System Controller

and an 18.432 MHz crystal that establishes the frequency of oscillation for the 8224 device via a 10pF capacitor, as shown in Figure 3-0. Together, the elements in the CPU Set perform all central processing functions. The following paragraphs describe how the elements within the CPU Set interact with all other logic on the SEC-80/10. The interaction between the IC's within the CPU Set, however, is not described. Instead, the reader is referred to the Intel® "8080 Microcomputer Systems User's Manual" for a detailed description of the 8080, 8224 and 8238 devices.
FIGURE 3-O. THE CPU SET
The CPU Set is shown on sheet 1 of the SBC-80/10 schematic (Appendix A).

3.1.1 INSTRUCTION TIMING

The activities of the CPU Set are cyclical. The CPU fetches an instruction, performs the operations required, fetches the next instruction, and so on. This orderly sequence of events requires precise timing. The 8224 Clock Generator, provides the primary timing reference for the CPU Set. The crystal in conjunction with a 10pF capacitor tunes an oscillator within the 8224 to precisely 18.432 MHz. The 8224 "divides" the oscillations by nine to produce two-phase timing inputs ($\phi_1$ and $\phi_2$) for the 8080. The $\phi_1$ and $\phi_2$ signals define a cycle of approximately 488 ns. duration. A TTL level phase 2 ($\phi_2$TTL) signal is also derived and made available to external logic. In addition, the output of the oscillator is buffered and brought out on OSC so that other system timing signals can be derived from this stable, crystal controlled source (e.g., the serial I/O baud rate is derived from OSC). All processing activities of the CPU Set are referred to the period of the $\phi_1$ and $\phi_2$ clock signals.

Within the 8080 CPU Set, an instruction cycle is defined as the time required to fetch and execute an instruction. During the fetch, a selected instruction (one, two or three bytes) is extracted from memory and deposited in the CPU's operating registers. During the execution part, the instruction is decoded and translated into specific processing activities.

Every instruction cycle consists of one, two, three, four or five machine cycles. A machine cycle is required each time the CPU accesses
memory or an I/O port. The fetch portion of an instruction cycle requires one machine cycle for each byte to be fetched. The duration of the execution portion of the instruction cycle depends on the kind of instruction that has been fetched. Some instructions do not require any machine cycles other than those necessary to fetch the instruction; other instructions, however, require additional machine cycles to write or read data to/from memory or I/O devices.

Each machine cycle consists of three, four or five states. A state is the smallest unit or processing activity and is defined as the interval between two successive positive-going transitions of the Ø1 clock pulse.

There are three exceptions to the defined duration of a state. They are the WAIT state, the hold (HLDA) state and the halt (HLTA) state. Because the WAIT, the HLDA, and the HLTA states depend upon external events, they are by their nature of indeterminate length. Even these exceptional states, however, must be synchronized with the pulses of the driving clock. Thus the duration of all states, including these, are integral multiples of the clock pulse.

To summarize, then, each clock period marks a state; three to five states summarize a machine cycle; and one to five machine cycles comprise an instruction cycle. A full instruction cycle requires anywhere from four to seventeen states for its completion, depending on the kind of instruction involved.

There is just one consideration that determines how many machine cycles are required in any given instruction cycle: the number of times that the processor must reference a memory address or an I/O address, in order to fetch and execute the instruction. Like many processors, the 8080 is so constructed that
it transmits one address per machine cycle. Thus, if the fetch­
ing and execution of an instruction requires two memory refer­
ences, then the instruction cycle associated with that instruction
consists of two machine cycles. If five such references are called
for, then the instruction cycle contains five machine cycles.

Every instruction cycle has at least one reference to memory,
during which the instruction is fetched. An instruction cycle must
always have a fetch, even if the execution of instruction requires
no further references to memory. The first machine cycle in every
instruction cycle is therefore a FETCH. Beyond that, there are no
fast rules. It depends on the kind of instruction. The input (INP)
and the output (OUT), instructions each require three machine cycles: a
FETCH, to obtain the instruction; a MEMORY READ, to obtain the address
of the object peripheral; and an INPUT or an OUTPUT machine cycle, to
complete the transfer.

Every machine cycle within an instruction cycle consists of three
to five active states (referred to as T1, T2, T3, T4, and T5). The
actual number of states depends upon the instruction being executed,
and on the particular machine cycle within the greater instruction
cycle. Figure 3-1 shows the timing relationships in a typical FETCH
machine cycle. Events that occur in each state are referred to tran­
sitions of the φ1 and φ2 clock pulses.

At the beginning of each machine cycle (in state T1), the 8080
activates its SYNC output and issues status information on its data
bus. The 8224 accepts SYNC and generates an active-low status strobe
(STSTB/) as soon as the status data is stable on the data bus. The
status information indicates the type of machine cycle in progress.
* Low on RDYIN here requires WAIT state (TW)

FIGURE 3-1. TYPICAL FETCH MACHINE CYCLE
The 8238 System Controller accepts the status bits from the 8080 and STSTB/ from the 8224, and uses them to generate the appropriate control signals (MEMR/, MEMW/, IOR/ and IOWR/) for the current machine cycle.

The rising edge of $g_2$ during T1 loads the processor's address lines (A0 - A15). These lines become stable within a brief delay of the $g_2$ clocking pulse, and they remain stable until the first $g_2$ pulse after state T3. This gives the processor ample time to read the data returned from memory.

Once the processor has sent an address to memory, there is an opportunity for the memory to request a WAIT. This it does by pulling the 8224's RDYIN line low. As long as the RDYIN line remains low, the CPU Set will idle, giving the memory time to respond to the addressed data request. The 8224 synchronizes RDYIN with internal processor timing and applies the result to the 8080's READY input. The processor responds to a wait request by entering an alternative state (TW) at the end of T2, rather than proceeding directly to the T3 state. A wait period may be of indefinite duration. The 8080 remains in the waiting condition until its READY line again goes high. The cycle may then proceed, beginning with the rising edge of the next $g_1$ clock. A WAIT interval will therefore consist of an integral number of TW states and will always be a multiple of the clock period.

The events that take place during the T3 state are determined by the kind of machine cycle in progress. In a FETCH machine cycle, the CPU Set interprets the data on its data bus as an instruction. During a MEMORY READ, signals on the same bus are interpreted as a data word.
The CPU Set itself outputs data on this bus during a MEMORY WRITE machine cycle. And during I/O operations, the CPU Set may either transmit or receive data, depending on whether an INPUT or an OUTPUT operation is involved. Consider the following two examples.

Figure 3-2 illustrates the timing that is characteristic of an input instruction cycle. During the first machine cycle (M1), the first byte of the two-byte IN instruction is fetched from memory. The 8080 places the 16-bit memory address on the system bus near the end of state T1. The 8238 activates the memory read control signal (MEMR/) during states T2 and T3 (and any intervening wait states, if required). During the next machine cycle (M2), the second byte of the instruction is fetched. During the third machine cycle (M3), the IN instruction is executed. The 8080 duplicates the 8-bit I/O address on address lines ADRO-7 and ADR8-F. The 8238 activates the I/O read control signal (IOR/) during states T2 and T3 of this cycle. In all cases the system bus enable input (BUSEN/) to the 8238 allows for normal operation of the data bus buffers and the read/write control signals. If BUSEN/ goes high the data bus output buffers and control signal buffers are forced into a high-impedance state.

Figure 3-3 illustrates an instruction cycle during which the CPU Set outputs data. During the first two machine cycles (M1 and M2), the CPU Set fetches the two-byte OUT instruction. During the third machine cycle (M3), the OUT instruction is executed. The 8080 duplicates the 8-bit I/O address on lines ADRO-7 and ADR8-F. The 8238 activates an advanced I/O write control signal (IOWR/) at the beginning of state T2 of this cycle. The nature and implications of the 8238 timing
Figure 3-2. Input Instruction Cycle

- **M1** (Fetch)
- **M2** (Fetch 2nd Byte)
- **M3** (Input)

- **T1** | **T2** | **T3** | **T4** | **T1** | **T2** | **T3**

**Addresses**

- **Address Bus**
- **Memory Address**
- **Memory Address**
- **I/O Port Address**

- **MEMR/**
- **IOR/**

- **Data Bus**

- **BUSEN/**

1st byte of IN instruction

2nd byte of IN instruction

Data From I/O Port

---

**Footnotes:**

3-9
FIGURE 3-3. OUTPUT INSTRUCTION CYCLE
will be explained later (p. 3-17). The 8238 outputs the data onto the system bus at the end of state T2. Data on the bus remains stable throughout the remainder of the machine cycle. BUSEn/ must be low to prevent the output and control buffers from being forced into the high-impedance state.

Observe that a RDYIN signal is necessary for completion of an output machine cycle. Unless such an indication is present, the processor enters the TW state, following the T2 state. Data on the output lines remains stable in the interim, and the processing cycle will not proceed until the RDYIN line again goes high.

The 8080 generates a WR! output for qualification of the advanced I/O write (IOWR/) and memory write (MEMW/) control signals from the 8238, during those machine cycles in which the CPU Set outputs data. The negative-going leading edge of WR! is referred to the rising edge of the first clock pulse following T2. WR! remains low until re-triggered by the leading edge of G2, during the state following T3. Note that any TW states intervening between T2 and T3 of the output machine cycle will necessarily extend WR!.

All processor machine cycles consist of at least three states: T1, T2, and T3 as just described. If the CPU Set has to wait for a RDYIN response, then the machine cycle may also contain one or more TW states. During the three basic states, data is transferred to or from the CPU Set.

After the T3 state, however, it becomes difficult to generalize. T4 and T5 states are available, if the execution of a particular instruction requires them. But not all machine cycles make use of these states. It depends upon the kind of instruction being executed, and
on the particular machine cycle within the instruction cycle. The processor will terminate any machine cycle as soon as its processing activities are completed, rather than proceeding through the T4 and T5 states every time. Thus the 8080 may exit a machine cycle following the T3, the T4, or the T5 state and proceed directly to the T1 state of the next machine cycle.

3.1.2 INTERRUPT SEQUENCES

The 8080 has the built-in capacity to handle external interrupt requests. Peripheral logic can initiate an interrupt simply by driving the processor's interrupt (INT) line high. The interrupt (INT) input is asynchronous, and a request may therefore originate at any time during any instruction cycle. Internal logic re-clocks the external request, so that a proper correspondence with the driving clock is established. An interrupt request (INT) arriving during the time that the interrupt enable line (INTE) is high, acts in coincidence with the Q2 clock to set the internal interrupt latch. This event takes place during the last state of the instruction cycle in which the request occurs, thus ensuring that any instruction in progress is completed before the interrupt can be processed.

The INTERRUPT machine cycle which follows the arrival of an enabled interrupt request resembles an ordinary FETCH machine cycle in most respects. The contents of the program counter are latched onto the address lines during T1, but the counter itself is not incremented during the INTERRUPT machine cycle, as it otherwise would be. In this way, the pre-interrupt status of the program counter is preserved, so that data in the counter may be saved in the stack.
This in turn permits an orderly return to the interrupted program after the interrupt request has been processed.

Because the 8238's INTR output (pin 23) is tied to +12 volts, the 8238 blocks incoming data and automatically inserts a Restart (RST 7) instruction onto the 8080 data bus during state T3, when the interrupt is acknowledged by the 8080. RST is a special one-byte call instruction that facilitates the processing of interrupts (the ordinary program call instruction is three bytes long). The RST 7 instruction causes the 8080 to branch program control to the instruction being stored in memory location 38 16.

3.1.3 HOLD SEQUENCES

By activating the 8080's HOLD input, an external device can cause the CPU Set to suspend its normal operations and relinquish control of the address and data busses. The CPU Set responds to a request of this kind by floating its address and data outputs, so that these exhibit a high impedance to other devices sharing the busses. At the same time, the processor acknowledges the HOLD by placing a high on on its HLDA output pin. During an acknowledged HOLD, the address and data busses are under control of the peripheral which originated the request, enabling it to conduct off board memory transfers without processor intervention.

3.1.4 HALT SEQUENCES

When a halt instruction (HLT) is executed, the 8080 enters the halt state after state T2 of the next machine cycle. There are only three ways in which the 8080 can exit the halt state:
A high on the 8224 reset input (RESIN/) will always reset the 8080 to state T1; reset also clears the program counter.

A HOLD input will cause the 8080 to enter the hold state, as previously described. When the HOLD line goes low, the 8080 re-enters the halt state on the rising edge of the next $\phi 1$ clock pulse.

An interrupt (i.e., INT goes high while INTE is enabled) will cause the 8080 to exit the halt state and enter state T1 on the rising edge of the next $\phi 1$ clock pulse. NOTE: The interrupt enable (INTE) flag must be set when the halt state is entered; otherwise, the 8080 will only be able to exit via a reset signal.

3.1.5 START-UP SEQUENCE

When power is applied initially to the 8080, the processor begins operating immediately. The contents of its program counter, stack pointer, and the other working registers are naturally subject to random factors and cannot be specified. For this reason, the CPU Set power-up sequence begins with a reset. An external RC network is connected to the 8224's RESIN/ input. The slow transition of the power supply rise is sensed by an internal Schmitt Trigger which converts the slow transition into a clean, fast edge on the RESIN/ line when the input level reaches a predetermined value.

An active RESIN/ input to the 8224 produces a synchronized RESET signal which restores the processor's internal program counter to zero. Program execution thus begins with memory location zero, following
a reset. Systems which require the processor to wait for an explicit start-up signal will store a halt instruction (HLT) after enabling interrupts in this location. A manual or an automatic INTERRUPT will be used for starting. In other systems, the processor may begin executing its stored program immediately. Note, however, that the reset has no effect on status flags, or on any of the processor's working registers (accumulator, indices, or stack pointer). The contents of these registers remain indeterminate, until initialized explicitly by the program.

In addition to generating a RESET signal, the RESIN/ input causes the 8224's status strobe (STSTB/) output to remain true (low). This allows both the 8080 and 8238 to be reset by a power-up sequence or an externally generated RESIN/ condition.

3.2 SYSTEM BUS INTERFACE LOGIC

The System Bus Interface logic consists of three general groups of circuitry:

1) assorted gates that accept the various bus control signals, the interrupt request lines, the ready indications and then applies these signals to the CPU Set,

2) the system bus drivers, and

3) the Failsafe circuitry which generates an acknowledgment during interrupt sequences and during those cycles in which an acknowledgment is not returned because a non-existent device was inadvertently addressed.

Each group is described in the following paragraphs.

3.2.1 SYSTEM CONTROL SIGNAL LOGIC

Interrupt Requests:

Four interrupt request lines are ORed together at A17-6 (ref. Appendix A) and applied to the 8080's INT input. Two of the interrupt request lines are from external sources: EXT INTR 1/ which enters the SBC-80/10
at connector J1 pin 49 and EXT INTR 2/ which enters the SBC-80/10 at P1-42. The other two interrupt requests originate on the SBC-80/10: INT 55/ is an interrupt request from ports 1 or 2 in the Parallel I/O Interface (see Section 3.6.2); and INT 51/ is an interrupt request from the 8251 USART in the Serial I/O Interface (see Section 3.5.4).

Hold Requests:

If the SBC-80/10 is operating in a system with other modules sharing a common external bus, another module can acquire control of the external bus by activating the 8080's HOLD/ input (connector pin P1-15). HOLD/ is inverted and applied to the 8080's HOLD pin. As described in Section 3.1.3, the 8080 will subsequently activate its hold acknowledge (HLDA) output. HLDA is, in turn, latched by a 74LS74 flip flop (at A29). The Q output from the D-type latch (DHLDA) disables the 8097 circuits (A47) that drive the external read/write control outputs: MRDC/, MWTC/, IORC/ and IOWC/. DHLDA also disables the external system address and data bus drivers by asserting a high at their active-low chip select (CS/) input pins. As a result of DHLDA, all of the above-mentioned drivers enter the high-impedance state. The Q output from the DHLDA output informs other modules of this condition via the BUSY/ output (connector pin P1-17). BUSY/ is driven by transistor Q5.

System Reset:

Connector pin P1-14 on the SBC-80/10 can be used to accept an externally generated SYSTEM RESET signal and to transfer a SBC-80/10
generated RESET signal to other modules in the system. If jumper pair 54-55 is connected, a RESET from the 8224 will be gated through the Q4 transistor to connector pin P1-14, thus resetting other modules in the system during power-up sequences. An externally generated SYSTEM RESET is accepted at P1-14, buffered, applied to the 8080's RESET input and made available to other logic on the SBC-80/10.

I/O Ready Generation

During each serial or parallel I/O cycle, a "ready" indication (IORDYIN/) is returned to the CPU Set. The three chip select lines for the 8251 and the two 8255 devices are ORed together (at A17-8 on sheet 3 of the schematic). The resultant output is then NANDed (at A44-11) with the I/O read (IOR) or the advanced I/O write (ADV IOW) signal to produce IORDYIN/. Recall from Section 3.1 that the 8238 System Controller (in the CPU Set) generates the I/O write control output at the beginning of all I/O write cycles. The IOW/ signal occurs earlier than the 8080's WR/ output. The 8238's IOW/ signal, alone, is labeled ADV IOW/. IOW/ is also synchronized with the 8080's WR/ output to produce the system write command IOWC/. ADV IOW/ allows the ready indication to be returned early enough to avoid an unnecessary wait state (see Figure 3-4). The IOWC/ signal causes an I/O device to actually write the data, later in the I/O cycle.

Ready Inputs:

Recall from Section 3.1.1 that the CPU Set must see a ready indication before proceeding to internal state T3 during all machine cycles. The 74S20 section at A57 on sheet 1 of the schematic OR's
the following ready indications:

1) INT ACK/ or TIME OUT ACK/ from the Failsafe logic (see Section 3.2.3),

2) IORDYIN/ from the Serial and Parallel I/O Interfaces,

3) PROM R DYIN/ from the ROM/PROM logic (see Section 3.4), and

4) RAM R DYIN/ from the RAM section (see Section 3.3).

The resultant output indicates an on-board memory or I/O access and is used to disable the external data bus drivers at A53 and A54. This output from A57-8 is also ORed (at A30-3) with the externally generated AACK/ (connector pin P1-25) and XACK/ (connector pin P1-23) inputs. The output from A30-3 is then applied to the CPU Set's R DYIN input (pin 3 on the 8224). When the SBC-80/10 CPU Set accesses an external module, the AACK/ or XACK/ input informs the CPU Set that the external device is ready. AACK/ is an advanced acknowledge that allows certain OEM modules to be accessed faster.

Figure 3-4 illustrates basic timing for the ready indications.

Bus Clock Generation:

The OSC output from the CPU Set (18.432 MHz frequency) is applied to the clock input of a 74LS74 D-type flip flop (at A29-11 on sheet 1 of the schematic). The \( \bar{Q} \) output from this latch is tied to its own D input. Consequently, the Q output exhibits half the frequency of the OSC input. This 9.216 MHz output is buffered and made available to external modules on the common clock (CCLK/) line (via connector pin P1-31) and the bus clock (BCLK/) line (via connector pin P1-13).

3.2.2 SYSTEM BUS DRIVERS

The SBC-80/10 internal memory data bus (DM0-DM7) is driven by
8080 MUST SENSE READY HERE TO AVOID WAIT STATE

8080 MUST SENSE READY HERE TO EXIT FIRST WAIT STATE

FIGURE 3-4. READY TIMING
two 8216 bidirectional bus drivers, shown at A55 and A56 on sheet 3 of the schematic. All data being transferred to/from the RAM memory (see Section 3.3) or ROM/PROM memory (see Section 3.4) is routed through these two devices. The chip select (CS/) input is provided by the MEM CMD/ signal which is the result of ORing RAM RDYIN/ and PROM RDYIN/. The direction enable (DIEN) input to the 8216's is provided by the memory read (MEMR) signal.

When the SBC-80/10 communicates with an external module, the data is driven by two 8226 bidirectional data bus drivers at A53 and A54 on sheet 1 of the schematic. The direction input to the 8226's is provided by the OR of memory read (MEMR) and I/O read (IOR). The 8226 devices will be disabled during 8080 HOLD sequences. The eight data bus lines to the 8226 bus drivers enter/leave the SBC-80/10 via the P1 edge connector.

The external 16-bit system address bus is driven by four 8226 bidirectional bus drivers. However, because the direction enable pin (EN/) on these 8226 devices is tied to ground, they can only be used to transmit addresses to external modules; they will not receive addresses from external modules. Consequently, the SBC-80/10 can access other modules, but other modules cannot access the memory or I/O controllers on the SBC-80/10. Like the data bus drivers, these 8226 devices are disabled during 8080 HOLD sequences.

3.2.3 FAILSAFE TIMER

When the 8080 acknowledges an interrupt request, the 8238 System Controller "forces" an RST 7 instruction onto the 8080's data bus
In order to read this RST 7 instruction, however, the 8080 must sense a ready indication. The 8080 acknowledges an interrupt by setting status bit 0 (DO) during the status output portion of each machine cycle (i.e., when STATUS STROBE is true). When this occurs, the 9602 one-shot (shown at A28 on sheet 5 of the schematic) is reset causing a low signal on its output (INTR ACK/). This output is then gated through to the RDYIN pin on the 8224 as described in Sections 3.2.1.

The Failsafe timer also performs another function. If the CPU Set tries to access a memory or I/O device but that device, for some reason, does not return a ready indication, then the 8080 remains in a wait state until ready is received. The Failsafe timer is designed to prevent hanging the system up in this way. The 9602 one-shot is triggered by STATUS STROBE at the beginning of each machine cycle. If the one-shot is not re-triggered (i.e., if another cycle does not begin) within 9 ms., then the 9602 times out and its output (also labeled TIME OUT ACK/) is gated through to the RDYIN pin on the 8224, thus allowing the 8080 to exit the wait state. This can be very helpful during system debugging.

3.3 RANDOM ACCESS MEMORY (RAM)

The Random Access Memory (RAM) provides the user with 1024 (1K) x 8-bits of read/write storage that requires no clocks or refresh to operate. The SBC 80/10 and SBC 80/10A utilize two different configurations, therefore each configuration is discussed separately in paragraphs 3.3.1 and 3.3.2.
3.3.1 SBC 80/10 RAM

The RAM logic consists of eight Intel 8111 256 x 4-bit Low Power Static RAM chips, an Intel 3205 three-to-eight decoder for chip selection and assorted gates as shown on sheet 2 of the SBC-80/10 schematic (Figure A-1).

The 8111 RAM devices used on the SBC 80/10 have a maximum access time of 500 nsec. Each chip has eight address inputs (A0-A7) that select one of the 256 four-bit segments, active-low write (W/) and chip enable (CE/) inputs and an output disable (OD) input. Each chip also has four common data input/output pins (I/01-I/04). A high on the OD input disables output and allows the I/O pins to be used for input. During memory read accesses, the data is read out nondestructively and has the same polarity as the input data.

The least significant system address lines (ADR0-ADR7) are applied to the eight address input pins on each 8111 RAM. The most significant eight system address lines (ADR8-ADRF) feed a 3205 decoder. Each of the four most significant decoder outputs are applied to the chip enable (CE/) inputs on two RAM chips. One RAM in each pair reads or writes data bits 0 to 3 (DM0-DM3) while the other RAM reads or writes data bits 4 to 7 (DM4-DM7) for each RAM access. One of the decoder outputs will be activated (low) whenever the value on the system address bus is within the range 3COO-3FFF (hexadecimal).

During memory write cycles, the advanced memory write signal (ADV MEMW/) is applied to the write input (W/) on each RAM. A high on the active-low memory read line (MEMR/) allows the selected RAM's I/O pins to be used to accept the data which is to be written into the addressed location. During memory read cycles, the level on ADV MEMW/ is high but is low on MEMR/ thus allowing the addressed data to be ready out and onto the data bus.
During all RAM access cycles, the active decoder output is NANDed with ADV MEMW or MEMR (at A44-3) to produce a ready indication for the CPU Set (RAM RDYIN/). The 8238 System Controller (see Section 3.1) generates ADV MEMW or MEMR early enough in the memory cycle to allow RAM RDYIN/ to appear at the CPU Set in time to prevent the occurrence of any wait states. Figure 3-5 illustrates RAM access timing.

Whenever SBC 80/10 RAM is accessed, the data is transferred to/from the RAM chips on the memory data bus (DMO-DM7). Lines DMO-DM7 are interfaced to the system data bus through two Intel 8216 bidirectional bus drivers (shown at A55 and A56 on sheet 3 of the schematic) as described in Section 3.2.

3.3.2 SBC 80/10A RAM

The SBC 80/10 RAM logic consists of eight Intel 8102 1024 x 1-bit Low-Power Static RAM chips, an Intel 3205 three-to-eight decoder, and assorted gates as shown on sheet 2 of the SBC 80/10A schematic (Figure A-2).

The 8102 RAM devices used on the SBC 80/10A have a maximum access time of 450 nsec. Each RAM chip has ten address inputs (ADRO-ADR9) that select one of the 1024 bits, an active low write (ADV MEMW/) and chip enable. A high on the ADV MEMW/ input allows a memory read access.

The ten least significant address lines (ADRO-ADR9) are applied to the ten address input pins on each 8102 RAM. The six most significant address lines (ADRA-ADRF) feed a 3205 decoder. The output of the 3205 decoder is applied to each Chip Enable (CE/) input to the eight 8102 RAM's. When the value on the system address bus is within the range 3C00-3FFF the decoder output will be activated (low).
During all RAM access cycles, the active decoder output produces a ready indication for the CPU set (RAM RDYIN/). The 8238 System Controller (see Section 3.1) generates ADV MEMW/ or MEMR/ early enough in the memory cycle to allow RAM RDYIN/ to appear at the CPU set in time to prevent the occurrence of any wait states. Figures 3-5 illustrates RAM access timing.

Whenever SBC 80/10A RAM is accessed, the data is transferred to/from the RAM chips on the memory data bus (DMO-DM7). Lines DMO-DM7 are interfaced to the system data bus through two Intel 8216 bidirectional bus drivers (shown at A55 and A56 on sheet 3 of the schematic) as described in section 3.2.

3.4 READ ONLY MEMORY (ROM/EPROM)

The SBC 80/10 and 80/10A have provisions for installing 4096 (4K) x 8-bit words of read only memory in sockets already on the PC board. Four Intel 8708 1K x 8-bit Erasable and Electrically Reprogrammable Read Only Memory (EPROM) chips or four 8308 1K x 8-bit Metal Masked Read Only Memory (ROM) chips can be installed in the four 24-pin sockets shown on sheet 3 of the schematics (APPENDIX A). Optionally the SBC 80/10A has provisions for installing 4096 (4K) x 8-bits of read only memory in the sockets using four Intel 2758 (1K x 8-bits) Erasable and Electrically Reprogrammable Read Only Memory (EPROM) chips or installing 8192 (8K) x 8-bit words of read only memory using either Intel 2716 2K x 8-bit Erasable and Electrically Reprogrammable Read Only Memory (EPROM) chips or Intel 2316E 2K x 8-bit Metal Masked Read Only Memory (ROM) chips.
In addition to the four 24-pin sockets, the ROM/PROM logic includes an Intel 3205 Decoder for address decoding and several assorted gates used in generating the ready indication.

When addressing up to 4K of ROM, address lines ADRO-ADR9 are applied to the address pins AO-A9 at each of the four sockets. The remaining address lines, ADRA-ADRF are decoded by the 3205 device at A42. Each of the four least significant decoder outputs are applied to the chip select (CS/) pin at one of four sockets. One chip select line will be activated whenever the value on the system address bus is between 0000 and OFFF (hexadecimal). In addition, when the four most significant address lines are low (i.e., the address is less than OFFF) during a memory read cycle, the output from the 74LS00 section at A39-3 is NANDed with MEMR to produce a ready indication (PROM RDYIN/) for the CPU Set. PROM RDYIN/ is thus generated in time to allow all ROM/PROM reads to occur without any wait states. PROM RDYIN/ has the same timing as RAM DRYIN/, as shown in Figure 3-5.

When using the optional 2716 or 2316E chips with the 80/10A, address lines ADRO-ADRA are applied to the address pins at each of the four sockets. The remaining address lines, ADRB-ADRF are decoded by the 3205 three-to-eight decoder. Each of the four least significant decoder outputs are applied to the Chip Select (CS/) pin at one of four sockets. One chip select line will be enabled when the value on the system address bus is between 0000 and 1FFF (hexadecimal).
FIGURE 3-5. RAM ACCESS TIMING
In addition when the three most significant address lines are low (i.e., the address is less than 1FFF) during a memory read cycle, the output from the 74LS00 at A39-3 is NANDed with MEMR/ to produce a ready indication PROM RDYIN/ for the CPU set. PROM RDYIN/ is generated in time to allow all ROM/PROM reads to occur without any wait states. PROM RDYIN/ has the same timing as RAM RDYIN/, as shown in figure 3-5.

Whenever one of the ROM/PROM devices are read, the data from the chips output pins (01-08) is placed on the memory data bus (DM0-DM7) which is interfaced to the system bus via two Intel 8216 bidirectional bus drivers (at A55 and A56), as described in Section 3.2.

3.5 SERIAL I/O INTERFACE

The Serial I/O Interface logic provides the SBC 80/10 with a serial data communications channel that can be programmed to operate with most of the current serial data transmission protocols, synchronous or asynchronous. Baud rate, character length, number of stop bits and even/odd parity are program selectable. In addition, the serial I/O Interface can be configured (through jumper connections) as an EIA RS232C interface or as a Teletype-compatible current loop interface.

The Serial I/O Interface logic consists primarily of an Intel 8251 USART device and a counting network for baud rate selection, as shown on sheet 4 of the SBC 80/10 schematic (Appendix A). Before describing the specific operation of the Serial I/O logic however, we will summarize the general operational characteristics of the 8251
USART, because it essentially defines the character of the Serial I/O Interface.

3.5.1 INTEL 8251 OPERATIONAL SUMMARY

The 8251 is a Universal Synchronous/Asynchronous Receiver/Transmitter designed specifically for the 8080 Micro-computer System. Like other I/O devices in the 8080 Micro-computer System its functional configuration is programmed by the systems software for maximum flexibility. The 8251 can support virtually any serial data technique currently in use (including IBM "Bi-Sync").

Modem Control

The 8251 has a set of control inputs and outputs that can be used to simplify the interface to almost any Modem. The modem control signals are general purpose in nature and can be used for functions other than Modem control, if necessary.

**DSR** (Data Set Ready)

The **DSR** input signal is general purpose in nature. Its condition can be tested by the CPU using a Status Read operation. The **DSR** input is normally used to test Modem conditions such as Data Set Ready.

**DTR** (Data Terminal Ready)

The **DTR** output signal is general purpose in nature. It can be set "low" by programming the appropriate bit in the Command Instruction word. The **DTR** output signal is normally used for Modem control such as Data Terminal Ready or Rate Select.
**RTS (Request to Send)**

The RTS output signal is general purpose in nature. It can be set "low" by programming the appropriate bit in the Command Instruction word. The RTS output signal is normally used for Modem control such as Request to Send.

**CTS (Clear to Send)**

A "low" on this input enables the 8251 to transmit data (serial) if the TxEN bit in the Command byte is set to a "one". This is very important to remember!

---

**USART PIN CONFIGURATION**

<table>
<thead>
<tr>
<th>Pin Name</th>
<th>Pin Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>D0-D3</td>
<td>Data Bus (R bits)</td>
</tr>
<tr>
<td>C/D</td>
<td>Control or Data is to be Written or Read</td>
</tr>
<tr>
<td>RD</td>
<td>Read Data Command</td>
</tr>
<tr>
<td>WR</td>
<td>Write Data or Control Command</td>
</tr>
<tr>
<td>CS</td>
<td>Chip Enable</td>
</tr>
<tr>
<td>CLK</td>
<td>Clock Pulse (11T)</td>
</tr>
<tr>
<td>RESET</td>
<td>Reset</td>
</tr>
<tr>
<td>TxC</td>
<td>Transmitter Clock</td>
</tr>
<tr>
<td>TxD</td>
<td>Transmitter Data</td>
</tr>
<tr>
<td>RxC</td>
<td>Receiver Clock</td>
</tr>
<tr>
<td>RxD</td>
<td>Receiver Data</td>
</tr>
<tr>
<td>RTS</td>
<td>Receiver Ready (this character for 8080)</td>
</tr>
<tr>
<td>+5V</td>
<td>+5 Volt Supply</td>
</tr>
<tr>
<td>GND</td>
<td>Ground</td>
</tr>
</tbody>
</table>

**FIGURE 3-6. 8251 PIN ASSIGNMENTS**

**TXRDY (Transmitter Ready)**

This output signals the CPU that the transmitter is ready to accept a data character. It can be used as an interrupt to the system or for polled operation when the CPU can check TXRDY using a status read operation. TXRDY is active only when CTS is enabled. TXRDY is automatically reset when a character is loaded from the CPU.
TXE (Transmitter Empty)

When the 8251 has no characters to transmit, the TxE output will go "high". It resets automatically upon receiving a character from the CPU. TxE can be used to indicate the end of a transmission mode, so that the CPU "knows" when to "turn the line around" in the half-duplexed operational mode.

In SYNChronous mode, a "high" on this output indicates that a character has not been loaded and the SYNC character or characters are about to be transmitted automatically as "fillers".

\[ TXC \] (Transmitter Clock)

The Transmitter Clock controls the rate at which the character is to be transmitted. In the Synchronous transmission mode, the frequency of TxC is equal to the actual Baud Rate (1X). In Asynchronous transmission mode, the frequency of \( \overline{TXC} \) is a multiple of the actual Baud Rate. A portion of the mode instruction selects the value of the multiplier; it can be 1X, 16X or 64X the Baud Rate.

For Example:

If Baud Rate equals 4800 Baud,

- \( \overline{TXC} \) equals 4800 Hz (1X)
- \( \overline{TXC} \) equals 76.8 kHz (16X)
- \( \overline{TXC} \) equals 307.2 kHz (64X).

The falling edge of \( \overline{TXC} \) shifts the serial data out of the 8251.
RXRDY (Receiver Ready)

This output indicates that the 8251 contains a character that is ready to be input to the CPU. RXRDY can be connected to the interrupt structure of the CPU or for polled operation the CPU can check the condition of RXRDY using a status read operation. RXRDY is automatically reset when the character is read by the CPU.

RXC (Receiver Clock)

The Receiver Clock controls the rate at which the character is to be received. In Synchronous Mode, the frequency of RXC is equal to the actual Baud Rate (1X). In Asynchronous Mode, the frequency of RXC is a multiple of the actual Baud Rate. A portion of the mode instruction selects the value of the multiplier; it can be 1X, 16X or 64X the Baud Rate.

For Example:

If Baud Rate equals 300 Baud,
RXC equals 300 Hz (1X)
RXC equals 4800 Hz (16X)
RXC equals 19.2 kHz (64X).

If Baud Rate equals 2400 Baud,
RXC equals 2400 Hz (1X)
RXC equals 38.4 kHz (16X)
RXC equals 153.6 kHz (64X).

Data is sampled into the 8251 on the rising edge of RXC.

Note: In most communications systems, the 8251 will be handling both the transmission and reception operations of a single link. Consequently, the Receive and Transmit Baud Rates will be the same. Both TXC and RXC will require identical frequencies for this operation and can be tied together and connected to a single frequency source (Baud Rate Generator) to simplify the interface.
SYNDET (SYNC Detect)

This pin is used in SYNCHronous Mode only. It is used as either input or output, programmable through the Control Word. It is reset to "low" upon RESET. When used as an output (internal Sync mode), the SYNDET pin will go "high" to indicate that the 8251 has located the SYNC character in the Receive mode. If the 8251 is programmed to use double Sync characters, then SYNDET will go "high" in the middle of the last bit of the second Sync character. SYNDET is automatically reset upon a Status Read operation.

When used as an input, (external SYNC detect mode), a positive going signal will cause the 8251 to start assembling data characters on the falling edge of the next RXC. Once in SYNC, the "high" input signal can be removed. The duration of the high signal should be at least equal to the period of RXC.

Programming the 8251

Prior to starting data transmission or reception, the 8251 must be loaded with a set of control words generated by the CPU. These control signals define the complete functional definition of the 8251 and must immediately follow a Reset operation (internal or external).

The control words are split into two formats:

1. Mode Instruction,
2. Command Instruction.

Both the Mode and Command instructions must conform to a specified sequence for proper device operation. The Mode Instruction must be inserted immediately following a Reset operation, prior to using the 8251 for data communication.

* This function is not used or made available to the user on the SBC 80/10.
All control words written into the 8251 after the Mode Instruction will load the Command Instruction. Command Instructions can be written into the 8251 at any time in the data block during the operation of the 8251. To return to the Mode Instruction format a bit in the Command Instruction word can be set to initiate an internal Reset operation which automatically places the 8251 back into the Mode Instruction format. Command Instructions must follow the Mode Instructions or Sync characters (see Figure 3-7).

**FIGURE 3-7. TYPICAL 8251 DATA BLOCK**

Mode Instruction:

This format defines the general operational characteristics of the 8251. It must follow a Reset operation (internal or external). Once the Mode instruction has been written into the 8251 by the CPU, Sync characters or Command instructions may be inserted.

The 8251 can be used for either synchronous or asynchronous
Data communications. The two least significant bits of the Mode Instruction control word specify synchronous or asynchronous operation. The format for the remaining bits in the control word depends on the mode chosen by bits 0 and 1. Figure 3-8 shows the control word format for the asynchronous mode, while Figure 3-9 illustrates the control word format for the synchronous mode.

Command Instruction:

Once the functional definition of the 8251 has been programmed by the Mode Instruction and the Sync Characters are loaded (if in Sync Mode) then the device is ready to be used for data communication. The Command Instruction controls the actual operation of the selected format. Functions such as: Enable Transmit/Receive, Error Reset and Modem Controls are provided by the Command Instruction.

Once the Mode instruction has been written into the 8251 and Sync characters inserted, if necessary, then all further "control writes" (C/D = 1) will load the Command Instruction. A Reset operation (internal or external) will return the 8251 to the Mode Instruction Format.

Figure 3-10 illustrate the format of a Command Instruction control word.

Status Read Definition

In data communication systems it is often necessary to examine the "status" of the active device to ascertain if errors have occurred or other conditions that require the processor's attention. The 8251 has facilities that allow the programmer to "read" the status of the device at any time during the functional operation.
Mode Instruction Format, Asynchronous Mode

TRANSMITTER OUTPUT

RxO MARKING

RECEIVER INPUT

Synchronous Mode

CPU BYTES (8 BITS/CHAR)

ASSEMBLED SERIAL DATA OUTPUT (TxD)

RECEIVE FORMAT

SERIAL DATA INPUT (RxD)

CPU BYTES (8 BITS/CHAR)*

DATA CHARACTERS

NOTE: IF CHARACTER LENGTH IS DEFINED AS 5, 6, OR 7 BITS THE UNUSED BITS ARE SET TO "ZERO"

FIGURE 3-8. ASYNCHRONOUS MODE.

FIGURE 3-9. SYNCHRONOUS MODE.
FIGURE 3-10. COMMAND INSTRUCTION FORMAT

A normal "read" command is issued by the CPU with the C/D input at one to accomplish this function.

Some of the bits in the Status Read Format have identical meanings to external output pins so that the 8251 can be used in a completely Polled environment or in an interrupt driven environment (refer to Figure 3-11).
FIGURE 3-11. STATUS READ FORMAT

8251 DATA TRANSFERS

Once programmed, the 8251 is ready to perform its communication functions. The TXRDY output is raised "high" to signal the CPU that the 8251 is ready to receive a character. This output (TXRDY) is reset automatically when the CPU writes a character into the 8251. On the other hand, the 8251 receives serial data from the MODEM or
I/O device; upon receiving an entire character the RXRDY output is raised "high" to signal the CPU that the 8251 has a complete character ready for the CPU to fetch. RXRDY is reset automatically upon the CPU read operation.

The 8251 cannot begin transmission until the TXEN (Transmitter Enable) bit is set in the Command Instruction and it has received a Clear To Send (CTS) input. The TXD output will be held in the marking state upon Reset.

Asynchronous Mode (Transmission):

Whenever a data character is sent by the CPU the 8251 automatically adds a Start bit (low level) and the programmed number of Stop bits to each character. Also, an even or odd Parity bit is inserted prior to the Stop bit(s), as defined by the Mode Instruction. The character is then transmitted as a serial data stream on the TXD output. The serial data is shifted out on the falling edge of TXC at a rate equal to 1, 1/16 or 1/64 that of the TXC, as defined by the Mode Instruction. BREAK characters can be continuously sent to the TXD if commanded to do so.

When no data characters have been loaded into the 8251 the TXD output remains "high" (marking) unless a BREAK (continuously low) has been programmed.

Asynchronous Mode (Receive):

The RXD line is normally high. A falling edge on this line triggers the beginning of a START bit. The validity of this START bit is checked by again strobing this bit at its nominal center. If a low is detected again, it is a valid START bit, and the bit
counter will start counting. The bit counter locates the center of the data bits, the parity bit (if it exists) and the stop bits. If parity error occurs, the parity error flag is set. Data and parity bits are sampled on the RXD pin with the rising edge of RXC. If a low level is detected as the STOP bit, the Framing Error flag will be set. The STOP bit signals the end of a character. This character is then loaded into the parallel I/O buffer of the 8251. The RXRDY pin is raised to signal the CPU that a character is ready to be fetched. If a previous character has not been fetched by the CPU, the present character replaces it in the I/O buffer, and the OVERRUN flag is raised (thus the previous character is lost). All of the error flags can be reset by a command instruction. The occurrence of any of these errors will not stop the operation of the 8251.

Synchronous Mode (Transmission):

The TXD output is continuously high until the CPU sends its first character to the 8251 which usually is a SYNC character. When the CTS line goes low, the first character is serially transmitted out. All characters are shifted out on the falling edge of TXC. Data is shifted out at the same rate as the TXC.

Once transmission has started, the data stream at TXD output must continue at the TXC rate. If the CPU does not provide the 8251 with a character before the 8251 becomes empty, the SYNC characters (or character if in single SYNC word mode) will be automatically inserted in the TXD data stream. In this case, the TXEMPTY pin will momentarily go high to signal that the 8251 is empty and SYNC characters are being sent out. The TXEMPTY pin is internally reset by the next character being written into the 8251.

3-39
Synchronous Mode (Receive):

In this mode, character synchronization can be internally or externally achieved. If the internal SYNC mode has been programmed, the receiver starts in a HUNT mode. Data on the RXD pin is then sampled on the rising edge of RXC. The content of the RX buffer is continuously compared with the first SYNC character until a match occurs. If the 8251 has been programmed for two SYNC characters, the subsequent received character is also compared. When both SYNC characters have been detected, the USART ends the HUNT mode and is in character synchronization. The SYNDAT pin is then set high, and is reset automatically by a STATUS READ.

Parity error and overrun error are both checked in the same way as in the Asynchronous receive mode.

The CPU can command the receiver to enter the HUNT mode if synchronization is lost.

3.5.2 Serial I/O Configurations

The 8251 USART presents a parallel, eight-bit interface to the CPU Set via the system data bus (DB0-DB7) and presents an EIA RS232C* or TTY current loop* interface to an external device (via edge connector J3). The 8251's interface with the CPU Set is enabled by a low

*Electrical interfaces provided on SBC 80/10.
level on its chip select (CS/) pin. CS/ is low when the I/O address on the system address bus is between EC and EF (hexadecimal). Address bits 2 through 7 are decoded (at A14) to produce the CS/ input. The

<table>
<thead>
<tr>
<th>I/O ADDRESS (BASE 16)</th>
<th>COMMAND</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ED OR EF</td>
<td>OUTPUT</td>
<td>CONTROL WORD</td>
</tr>
<tr>
<td>EC OR EE</td>
<td>OUTPUT</td>
<td>DATA</td>
</tr>
<tr>
<td>ED OR EF</td>
<td>INPUT</td>
<td>STATUS</td>
</tr>
<tr>
<td>EC OR EE</td>
<td>INPUT</td>
<td>DATA</td>
</tr>
</tbody>
</table>
least significant address bit, ADR0, is applied to the 8251's C/D input (pin 12) thus indicating a control (if set) or data (if reset) byte on the data bus.

An output instruction (IOW/ is true) to port ED or EF (CS/ is low and ADR0 is high) causes the 8251 to accept a control byte through its data bus pins. The control byte can be either a mode instruction or a command instruction, depending on the sequence in which it is sent. The various bits in the mode control word specify the baud rate multiplexer, character length, parity and the number of stop bits as described in Section 3.5.1. Note that the actual baud rate selected is dependent on the configuration of the baud rate jumper network (refer to Section 3.5.3). The various bits in the command control word instruct the USART to enable/disable the receiver and transmitter, to reset errors, to reset internal control and return to the mode control cycle, and to set/clear the Data Terminal Ready output.

An output instruction to port EC or EE (CS/ and ADR0 are low) causes the 8251 USART to accept a data byte through its data bus pins. Bit 0 is the least significant bit and bit 7 is the most significant bit. The 8251 will subsequently transmit the data byte (if the transmitter is enabled), in serial fashion, to the external device as described in Section 3.5.1.

An input instruction (IOR/ is true) to port ED or EF (CS/ is low and ADR0 is high) causes the 8251 USART to place a status byte onto the system bus. The status bits are the result of status and error checking functions performed within the USART (see Section 3.5.1).

An input instruction (IOR/ is true) to port EC or EE (CS/ and ADR0 are low) causes the USART to output a data byte (previously
received from the external device) from its data bus pins. Bit 0 is the least significant bit and bit 7 is the most significant bit.

Timing for the USART's internal function is provided by the \(2\text{TTL}\) signal (see Section 3.1.1). The USART is reset by the occurrence of a high level on the RESET line.

The 8251 USART transmits and receives serial data, synchronously or asynchronously, as described in Section 3.5.1. By jumper-connecting the 8251 pins to different external lines, the Serial I/O logic can present either a Teletype-compatible current loop interface or an EIA RS232C interface to an external device. If the TTY-compatible current loop interface is used, the connections listed in Table 4-1 are required (see Section 4.1).

If the EIA RS232C interface is used, the connections listed in Table 4-2 are required (see Section 4.1).

3.5.3 BAUD RATE CLOCK GENERATION

The baud rate clock network consists of a 93S16 'divide-by-15' counter, two 74161 'divide-by-16' counters and wire-wrap jumpers for baud rate clock selection. The 93S16 counter is driven by the oscillator output (OSC) from the CPU Set. The QD output from this counter, in turn, drives the two 74161 counters. The outputs from these counters, each providing a different clock frequency, are tied to jumper pins that can be connected to the BAUD RATE CLK line. The available frequencies are listed in Table 4-3 (located in Section 4.2). Recall that the effective baud rate of the 8251 USART is also dependent on the state of the 8251's internal frequency divider and the mode of operation (refer to Section 3.5.1). The 8251 is capable of dividing the baud rate clock by 1, 16 or 64.
3.5.4 SERIAL I/O INTERRUPTS

The Serial I/O logic can be configured with different forms of an interrupt request mechanism. By connecting jumper pair 16-17 and disconnecting 15-16, the user can allow the 8251's Receiver Ready (RXRDY) output (pin 14) to generate an interrupt request (INT51/) to the CPU Set. RXRDY goes high whenever the receiver enable bit of the command word has been set and the 8251 contains a character that is ready to be input to the CPU Set. The user can also choose to have the 8251's Transmitter Ready (TXRDY) or the Transmitter Empty (TXE) output activate the INT51/ interrupt request. If jumper pair 19-21 is connected, a high on TXRDY (pin 15) will activate INT51/. If jumper pair 18-19 is connected instead, an active TXE (pin 18) output will generate INT51/. TXE goes high when the 8251 has no characters to transmit. TXRDY is high when the 8251 is ready to accept a character from the CPU Set. Both TXE and TXRDY are enabled by setting the transmit enable bit of the command word. Notice on the schematic that, if jumper pairs 19-20 and 15-16 are connected, Serial I/O interrupts are inhibited.

Upon receiving an interrupt, the program can determine the actual condition which is responsible for the interrupt (RXRDY, TXRDY or TXE) by reading the status of the 8251 device as described in Section 3.5.1. The interrupt request will be removed when the data is transferred to/from the 8251, as required. Note that the TXE or TXRDY output will be high, and consequently maintain an interrupt request, during all idle periods, since the 8251's transmit buffer will remain empty. To disable the transmitter, and the resultant interrupt request, the program can issue a command instruction to the 8251 with the TXEN bit (bit 0) equal to zero (refer to Section 3.5.1). The transmitter should not be disabled until TXE is high.
3.6 PARALLEL I/O INTERFACE

The Parallel I/O Interface logic on the SBC-80/10 provides forty-eight (48) signal lines for the transfer and control of data to or from peripheral devices. Eight lines have a bidirectional driver and termination network permanently installed. The remaining forty lines are uncommitted. Sockets are provided for the installation of active driver networks or passive termination networks. The optional drivers and terminators are installed in groups of four by insertion into the 14-pin sockets.

All forty-eight signal lines emanate from the I/O ports on two Intel 8255 Programmable Peripheral Interface devices, as shown on sheet 5 of the SBC-80/10 schematic (Appendix A). The two 8255 devices allow for a wide variety of I/O configurations. Before describing the possible configurations, however, we will summarize the general operational characteristics of the 8255 device.

3.6.1 INTEL 8255 OPERATIONAL SUMMARY

The 8255 contains three 8-bit ports (A, B, and C). All can be configured in a wide variety of functional characteristics by the system software but each has its own special features or "personality" to further enhance the power and flexibility of the 8255.

Port A: One 8-bit data output latch/buffer and one 8-bit data input latch.

Port B: One 8-bit data input/output latch/buffer and one 8-bit data input buffer.

Port C: One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input). This port can be divided into
two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal outputs and status signal inputs in conjunction with Ports A and B.

The 8080 CPU dictates the operating characteristics of the ports by outputting two different types of control words to the 8255:

1) mode definition control word (bit 7 = 1)
2) port C bit set/reset control word (bit 7 = 0)

Bit 7 of each control word specifies its format, as shown in Figures 3-13 and 3-14, respectively.

Mode Selection

There are three basic modes of operation that can be selected by the system software:

Mode 0 - Basic Input/Output
Mode 1 - Strobed Input/Output
Mode 2 - Bi-Directional Bus

When the RESET input goes "high" all ports will be set to the Input mode 0 (i.e., all 24 lines will be in the high impedance state). After the RESET is removed the 8255 can remain in the Input mode with no additional initialization required. During the execution of the system program, the other modes may be selected using a single OUT- put instruction. This allows a single 8255 to service a variety of peripheral devices with a simple software maintenance routine.

The modes for Port A and Port B can be separately defined, while Port C is divided into two portions as required by the Port A and Port B definitions. All of the output registers, including the status flip-flops, will be reset whenever the mode is changed except for OBF in modes 1 and 2.
PIN CONFIGURATION

PIN NAMES

DATA BUS (IN DIRECTIONAL)

RESET INPUT

CHIP SELECT

READ INPUT

WRITE INPUT

PORT ADDRESS

PORT A (BIT)

PORT B (BIT)

PORT C (BIT)

+5 VOLTS

GND 0 VOLTS

CONTROL WORD

GROUP B

PORT C (LOWER)

0 = INPUT

1 = OUTPUT

GROUP A

PORT C (UPPER)

0 = INPUT

1 = OUTPUT

MODE SELECTION

00 = MODE 0

01 = MODE 1

10 = MODE 2

11 = Mode 3

MODE SET FLAG

1 = ACTIVE

FIGURE 3-12. 8255 PIN ASSIGNMENTS.

FIGURE 3-13. MODE DEFINITION CONTROL WORD FORMAT.
Modes may be combined so that their functional definition can be "tailored" to almost any I/O structure. For instance; Group B can be programmed in Mode 0 to monitor simple switch closings or display computational results, Group A could be programmed in Mode 1 to monitor a keyboard or tape reader on an interrupt-driven basis.

Single Bit Set/Reset Feature

Any of the eight bits of Port C can be Set or Reset using a single OUTput instruction (see Figure 3-14). This feature reduces software requirements in Control-based applications.

When Port C is being used as status/control for Port A or B, these bits can be set or reset by using the Bit Set/Reset operation just as if they were data output ports.

Interrupt Control Functions

When the 8255 is programmed to operate in Mode 1 or Mode 2, control signals are provided that can be used as interrupt request inputs to the CPU. The interrupt request signals, generated from Port C, can be inhibited or enabled by setting or resetting the associated INTE flip-flop, using the Bit set/reset function of Port C.

This function allows the Programmer to disallow or allow specific I/O devices to interrupt the CPU without effecting any other device in the interrupt structure.

INTE flip-flop definition:

(BIT-SET) - INTE is SET - Interrupt enable
(BIT-RESET) - INTE is RESET - Interrupt disable

Note: All Mask flip-flops are automatically reset during mode selection and device Reset.
FIGURE 3-14. BIT SET/RESET CONTROL WORD FORMAT.

Operating Modes

Mode 0 (Basic Input/Output):

This functional configuration provides simple Input and Output operations for each of the three ports. No "hand-shaking" is required, data is simply written to or read from a specified port.

Mode 0 timing is illustrated in Figure 3-15.

Mode 0 Basic Functional Definitions:

- Two 8-bit ports and two 4-bit ports.
- Any port can be input or output.
- Outputs are latched.
- Inputs are not latched.
16 different Input/Output configurations are possible in this Mode. Figure 3-16 shows two possible configurations.

FIGURE 3-15. 8255 MODE 0 TIMING

FIGURE 3-16. EXAMPLES OF MODE 0 CONFIGURATION.
Mode 1 (Strobed Input/Output):

This functional configuration provides a means for transferring I/O data to or from a specified port in conjunction with strobes or "handshaking" signals. In Mode 1, Port A and Port B use the lines on Port C to generate or accept these "handshaking" signals.

Mode 1 Basic Functional Definitions:

- Two transfer ports (A and B).
- Each transfer port contains one 8-bit data port and 4 bits from one half of the control/data port (Port C).
- The 8-bit data port can be either input or output. Both inputs and outputs are latched.

Input Control Signal Definition for Mode 1

STB (Strobe Input)

A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F)

A "high" on this output indicates that the data has been loaded into the input latch; in essence, an acknowledgement. IBF is set by the falling edge of the STB input and is reset by the rising edge of the RD input.

INTR (Interrupt Request)

A "high" on this output can be used to interrupt the CPU when an input device is requesting service. INTR is set by the rising edge of STB if IBF is a "one" and INTE is a "one". It is reset by the falling edge of RD. This procedure allows an input device to
request service from the CPU by simply strobing its data into the port.

INTE A
Controlled by bit set/reset of PC4.

INTE B
Controlled by bit set/reset of PC2.

Figure 3-17 illustrates the Mode 1 input configuration, while Figure 3-18 shows the basic timing for Mode 1 input.

Output Control Signal Definition for Mode 1

OBF (Output Buffer Full F/F)

The OBF output will go "low" to indicate that the CPU has written data out to the specified port. The OBF F/F will be set by the rising edge of the WR input and reset by the falling edge of the ACK input signal.

ACK (Acknowledge Input)

A "low" on this input informs the 8255 that the data from Port A or Port B has been accepted. In essence, a response from the peripheral device indicating that it has received the data output by the CPU.

INTR (Interrupt Request)

A "high" on this output can be used to interrupt the CPU when an output device has accepted data transmitted by the CPU. INTR is set by the rising edge of ACK if OBF is a "one" and INTE is a "one". It is reset by the falling edge of WR.

INTE A
Controlled by bit set/reset of PC6.
FIGURE 3-17. MODE 1 INPUT CONFIGURATION

FIGURE 3-18. 8255 MODE 1 INPUT TIMING
INTE B

Controlled by bit set/reset of PC2.

Figure 3-19 illustrates the Mode 1 output configuration, while Figure 3-20 shows basic Mode 1 output timing.

FIGURE 3-19. MODE 1 OUTPUT CONFIGURATION.

FIGURE 3-20. MODE 1 BASIC OUTPUT TIMING
Mode 2 (Strobed Bi-Directional Bus I/O):

This functional configuration provides a means for communicating with a peripheral device or structure on a single 8-bit bus for both transmitting and receiving data (bi-directional bus I/O). "Handshaking" signals are provided to maintain proper bus flow discipline in a similar manner to Mode 1. Interrupt generation and enable/disable functions are also available.

Mode 2 Basic Functional Definitions:

- Used in Port A only.
- One 8-bit, bi-directional data Port (Port A) and a 5-bit control Port (Port C).
- Both inputs and outputs are latched.
- The 5-bit control port (Port C) is used for control and status for the 8-bit, bi-directional data port (Port A).

Bi-Directional Bus I/O Control Signal Definition

**INTR (Interrupt Request)**

A high on this output can be used to interrupt the CPU for both input or output operations.

**Output Operation Control Signals**

**OBF (Output Buffer Full)**

The OBF output will go "low" to indicate that the CPU has written data out to Port A.

**ACK (Acknowledge)**

A "low" on this input enables the tri-state output buffer of Port A to send out the data. Otherwise, the output buffer will be in the high-impedance state.
INTR A and B (The INTE flip-flop associated with OBF)

Controlled by bit set/reset of PC6 (INTE1)

Input Operation Control Signals

STB (Strobed Input)

A "low" on this input indicates that data has been loaded into the input latch.

IBF (Input Buffer Full F/F)

A "high" on this output indicates that data has been loaded into the input latch.

INTE 2 (The INTE flip-flop associated with IBF)

Controlled by bit set/reset PC4 (INTE 2)

\[
\text{INTR}_A = \text{PC6} \cdot \text{OBFA} + \text{PC4} \cdot \text{IBFA}
\]

Figure 3-21 illustrates the port configuration for Mode 2, Figure 3-22 shows Mode 2 timing, and Table 3-1 summarizes 8255 Mode definition.
FIGURE 3-22. MODE 2 TIMING
## Mode Definition Summary Table

<table>
<thead>
<tr>
<th></th>
<th>Mode 0</th>
<th>Mode 1</th>
<th>Mode 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IN</td>
<td>OUT</td>
<td>IN</td>
</tr>
<tr>
<td>PA0</td>
<td>IN</td>
<td>OUT</td>
<td>IN</td>
</tr>
<tr>
<td>PA1</td>
<td>IN</td>
<td>OUT</td>
<td>IN</td>
</tr>
<tr>
<td>PA2</td>
<td>IN</td>
<td>OUT</td>
<td>IN</td>
</tr>
<tr>
<td>PA3</td>
<td>IN</td>
<td>OUT</td>
<td>IN</td>
</tr>
<tr>
<td>PA4</td>
<td>IN</td>
<td>OUT</td>
<td>IN</td>
</tr>
<tr>
<td>PA5</td>
<td>IN</td>
<td>OUT</td>
<td>IN</td>
</tr>
<tr>
<td>PA6</td>
<td>IN</td>
<td>OUT</td>
<td>IN</td>
</tr>
<tr>
<td>PA7</td>
<td>IN</td>
<td>OUT</td>
<td>IN</td>
</tr>
<tr>
<td>PB0</td>
<td>IN</td>
<td>OUT</td>
<td>IN</td>
</tr>
<tr>
<td>PB1</td>
<td>IN</td>
<td>OUT</td>
<td>IN</td>
</tr>
<tr>
<td>PB2</td>
<td>IN</td>
<td>OUT</td>
<td>IN</td>
</tr>
<tr>
<td>PB3</td>
<td>IN</td>
<td>OUT</td>
<td>IN</td>
</tr>
<tr>
<td>PB4</td>
<td>IN</td>
<td>OUT</td>
<td>IN</td>
</tr>
<tr>
<td>PB5</td>
<td>IN</td>
<td>OUT</td>
<td>IN</td>
</tr>
<tr>
<td>PB6</td>
<td>IN</td>
<td>OUT</td>
<td>IN</td>
</tr>
<tr>
<td>PB7</td>
<td>IN</td>
<td>OUT</td>
<td>IN</td>
</tr>
<tr>
<td>PC0</td>
<td>IN</td>
<td>OUT</td>
<td>IN</td>
</tr>
<tr>
<td>PC1</td>
<td>IN</td>
<td>OUT</td>
<td>IBF_B</td>
</tr>
<tr>
<td>PC2</td>
<td>IN</td>
<td>OUT</td>
<td>STB_B</td>
</tr>
<tr>
<td>PC3</td>
<td>IN</td>
<td>OUT</td>
<td>INTR_A</td>
</tr>
<tr>
<td>PC4</td>
<td>IN</td>
<td>OUT</td>
<td>STB_A</td>
</tr>
<tr>
<td>PC5</td>
<td>IN</td>
<td>OUT</td>
<td>IBF_A</td>
</tr>
<tr>
<td>PC6</td>
<td>IN</td>
<td>OUT</td>
<td>I/O</td>
</tr>
<tr>
<td>PC7</td>
<td>IN</td>
<td>OUT</td>
<td>I/O</td>
</tr>
</tbody>
</table>

### Table 3-1. 8255 Mode Definition Summary
3.6.2 PARALLEL I/O CONFIGURATIONS

Referring to sheet 5 of the schematic, we see that there are two 8255 devices, one located at A19, the other at A20. For convenience the following device designations will be used: The device at A19 is called the "group 1" device, while the device at A20 is referred to as the "group 2" device. Each device has three eight-bit ports. The "group 1" ports are designated Ports 1, 2 and 3 while the "group 2" ports are designated Ports 4, 5 and 6.

The group 1 and group 2 devices both communicate with the CPU Set using the same signal lines: the 8-bit data bus, DBO-DB7, and seven control/address lines; ADR0, ADR1, RESET, IOR/, IOW/, CS1/, and CS2/. The data lines bring control bytes or data bytes to an 8255 or deliver data from an 8255 to the CPU Set. The chip select control signals (CS1/ and CS2/) select the group 1 and group 2 devices, respectively, when the proper I/O address appears on the system address bus. CS1/ and CS2/ are the result of decoding address bits 2 through 7 (ADR2-ADR7), as shown on sheet 4 of the schematic (at A14). The two least significant address bits select the control register (when programming an 8255) or one of the three I/O ports (when reading or writing data). IOR/ (8255 → CPU Set) and IOW/ (CPU Set → 8255) indicate the direction of data flow, as summarized in Table 3-2. Specific I/O addresses for the six ports and two 8255 control registers on the SBC-80/10 are listed in Table 3-3.

A high on the RESET line clears all internal 8255 registers including the control register; all ports (A, B and C) are set for input.

Though both 8255's maintain the same interface (at different
### TABLE 3-2. 8255 BASIC OPERATION

<table>
<thead>
<tr>
<th>A1</th>
<th>A0</th>
<th>IOR/</th>
<th>IOW/</th>
<th>CS/</th>
<th>Input Operation (Read)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Port A → Data Bus</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Port B → Data Bus</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Port C → Data Bus</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Output Operation (Write)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 1 0 0</td>
<td>Data Bus → Port A</td>
</tr>
<tr>
<td>0 1 1 0 0</td>
<td>Data Bus → Port B</td>
</tr>
<tr>
<td>1 0 1 0 0</td>
<td>Data Bus → Port C</td>
</tr>
<tr>
<td>1 1 1 0 0</td>
<td>Data Bus → Control</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Disable Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>x x x x 1</td>
<td>Data Bus → High-Impedance</td>
</tr>
<tr>
<td>1 1 0 1 0</td>
<td>Illegal</td>
</tr>
</tbody>
</table>

### TABLE 3-3. PARALLEL I/O PORT ADDRESSES

<table>
<thead>
<tr>
<th>Port</th>
<th>8255 Device Location</th>
<th>*Eight Bit Address (Hexadecimal)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8255 #1 Port (A)</td>
<td>E4</td>
</tr>
<tr>
<td>2</td>
<td>8255 #1 Port (B)</td>
<td>E5</td>
</tr>
<tr>
<td>3</td>
<td>8255 #1 Port (C)</td>
<td>E6</td>
</tr>
<tr>
<td></td>
<td>8255 #1 Control</td>
<td>E7 For I/O write only.</td>
</tr>
<tr>
<td>4</td>
<td>8255 #2 Port (A)</td>
<td>E8</td>
</tr>
<tr>
<td>5</td>
<td>8255 #2 Port (B)</td>
<td>E9</td>
</tr>
<tr>
<td>6</td>
<td>8255 #2 Port (C)</td>
<td>EA</td>
</tr>
<tr>
<td></td>
<td>8255 #2 Control</td>
<td>EB For I/O write only.</td>
</tr>
</tbody>
</table>

*Note: If address = 111001xx, CS1/ is activated.  
If address = 111010xx, CS2/ is activated.*
I/O addresses) with the CPU Set, the interface between the group 1 device and edge connector J1 is significantly different than the interface between the group 2 device and its associated edge connector (J2). This gives the user a great deal of flexibility when configuring the system's external parallel I/O devices. Because of those flexible "external" interfaces, however, not all ports are capable of operating in each 8255 mode, though all ports can be programmed as either input or output. The group 1 ports can fully utilize the 8255's multi-mode and external interrupt capabilities as described in Section 3.6.1. The group 2 ports, however, are limited to a single mode of operation. The allowable port configurations for both groups are summarized below:

<table>
<thead>
<tr>
<th>Port 1 (Group 1 Port A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode 0  Input</td>
</tr>
<tr>
<td>Mode 0  Output (Latched)</td>
</tr>
<tr>
<td>Mode 1  Input (Strobed)</td>
</tr>
<tr>
<td>Mode 1  Output (Latched)</td>
</tr>
<tr>
<td>Mode 2  Bidirectional</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Port 2 (Group 1 Port B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode 0  Input</td>
</tr>
<tr>
<td>Mode 0  Output (Latched)</td>
</tr>
<tr>
<td>Mode 1  Input (Strobed)</td>
</tr>
<tr>
<td>Mode 1  Output (Latched)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Port 3 (Group 1 Port C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode 0  8 Bit Input</td>
</tr>
<tr>
<td>Mode 0  8 Bit Output (Latched)</td>
</tr>
</tbody>
</table>

Note: Control mode dependent upon Port A and B mode.

<table>
<thead>
<tr>
<th>Ports 4 and 5 (Group 2 Port A, B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode 0  Input</td>
</tr>
<tr>
<td>Mode 0  Output (Latched)</td>
</tr>
</tbody>
</table>
Port 6 (Group 2 Port C)

Mode 0 8 Bit Input
Mode 0 8 Bit Output
Mode 0 4 Bit Input/4 Bit Output (Unlatched/latched)
Mode 0 4 Bit Output/4 Bit Input (Latched/unlatched)

Group 1

Port 1 is the most versatile of the six ports. It can be pro-
grammed to function in any one of the three 8255 operating modes. This
first port is the only port that already includes a permanent bidirectional
driver/termination network (two 8226 bus driver devices at A1 and A2).

Before Port 1 is programmed for input or output in any one of three
operating modes (as described in Section 3.6.1), certain jumper connec-
tions must be made to allow the port to function properly in the chosen
mode. The 40-41-42-43 jumper pad specifies the direction of data flow
for the two 8226 bidirectional bus drivers. If input in mode 0 or mode
1 is to be programmed for Port 1, jumper pair 41-42 should be connected.
If output in mode 0 or mode 1 is to be used, jumper pair 40-41 should be
connected. If Port 1 is to be programmed for bidirectional mode 2, then
jumper pair 41-43 should be connected. This connection allows the out-
put acknowledge, ACK/, that is input on bit 6 of Port 3 to dynamically
dictate direction for the two 8226 devices.

Another jumper pad (48-49-50-51) enables interrupts for Port 1
when it is in mode 1 or mode 2. Jumper pair 49-50 should be connected
to allow the INTR output (see Section 3.6.1) from bit 3 of Port 3 to
activate an interrupt request (INT55/) from the 74LS02 gate at A45.
In mode 0, during which there is no provision for interrupts, jumper
pairs 48-49 and 50-51 must be connected to allow use of bit 3 of port
3 and to inhibit Port 1 interrupts.

3-62
Because the 8226 bus drivers are inverting devices, all data input to or output from Port 1 is considered to be negative true with respect to the levels at the J1 edge connector.

Port 2 can be programmed for input or output in either mode 0 or mode 1 (see Section 3.6.1). If Port 2 is to be used for input (in either mode), terminator networks must be installed in the sockets at A5 and A6. Because these networks must be passive, data that is input to Port 2 will be positive true. If Port 2 is to be used for output (in either mode), driver networks must be installed in the sockets at A5 and A6. Assuming that the drivers are inverting devices, then the data being output will be negative true at the J1 edge connector.

When Port 2 is programmed for mode 1, interrupts can be enabled by connecting jumper pair 45-46. This connection allows the INTR output from bit 0 of Port 3 to activate the interrupt request (INT55/) to the CPU set. When Port 2 is in mode 0, jumper pairs 44-45 and 46-47 must be connected to allow use of bit 0 of Port 3 and to inhibit Port 2 interrupts.

As was described in Section 3.6.1, the use of Port 3 is dependent on the modes programmed for Ports 1 and 2. If Port 1 is in mode 1 or mode 2, bits 3, 4, 5, 6 and 7 of Port 3 can have dedicated control functions.

| Port 3 bit 3 | INTR (interrupt request) | input or output |
| Port 3 bit 4 | STB/ (input strobe) | mode 1 input |
| Port 3 bit 5 | IBF (input buffer full flag) | or mode 2 |
| Port 3 bit 6 | ACK/ (output acknowledge) | mode 1 output |
| Port 3 bit 7 | OBF/ (output buffer full flag) | or mode 2 |

3-63
If Port 2 is in mode 1, bits 0, 1 and 2 of Port 3 have dedicated control functions:

- Port 3 bit 0 ↔ INTR (interrupt request) - input or output
- Port 3 bit 1 ↔ IBF (input buffer full) input only
- Port 3 bit 2 ↔ STB/ (input strobe)
- Port 3 bit 1 ↔ OBF/ (output buffer full) output only
- Port 3 bit 2 ↔ ACK/ (output acknowledge)

While certain Port 3 bits are available if Port 1 is in mode 1 or if Port 2 is in mode 0, the use of Port 3 as an eight-bit data path is restricted to those configurations that have both Port 1 and Port 2 programmed for mode 0. In this case all 8 bits of Port 3 can be programmed for mode 0 input (termination networks must be installed in the sockets at A3 and A4) or output (driver networks must be installed at A3 and A4). Note: If Port 1 and 2 are not both in mode 0, then a driver network must be installed in the sockets at A3 and A4, and a termination network must be installed at A4, so that the Port 3 control lines can function properly.

**Group 2**

The three ports on the group 2 device can be programmed for input or output, but only in mode 0. If Port 4 is programmed for input, termination networks must be installed in the sockets at A7 and A8. The data being input will be in positive true form. If Port 4 is programmed for output, driver networks must be installed at A7 and A8. Assuming that inverting drivers are used, then the data will be considered negative true at the J2 edge connector.

If Port 5 is programmed for input, termination networks must be installed in the sockets at A21 and A11. If Port 5 is programmed for output, driver networks must be installed at A21 and A11.
All eight bits of Port 6 can be programmed for input or output, or four bits can be programmed for input while the other four bits are programmed for output (see Section 3.6.1). Driver termination networks must be installed in the sockets at A9 and A10 as listed in Table 3-4.

**TABLE 3-4. Port 6 I/O CONFIGURATIONS**

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Sockets at A9</th>
<th>Sockets at A10</th>
</tr>
</thead>
<tbody>
<tr>
<td>8-bit Input</td>
<td>Terminators*</td>
<td>Terminators*</td>
</tr>
<tr>
<td>8-bit Output</td>
<td>Drivers**</td>
<td>Drivers**</td>
</tr>
<tr>
<td>Upper 4-bits Input/Lower 4-bits Output</td>
<td>Terminators*</td>
<td>Drivers**</td>
</tr>
<tr>
<td>Lower 4-bits Input/Upper 4-bits Output</td>
<td>Drivers**</td>
<td>Terminators*</td>
</tr>
</tbody>
</table>

* Positive-true data.
** Negative-true data if inverting drivers.

In Section 4.2, all of the user options for configuring parallel I/O on the SBC-80/10 are summarized for convenient reference.
The SEC-80/10 provides the user with a powerful, but flexible I/O capability for both parallel and serial transfers. The serial I/O Interface, using Intel's 8251 USART, provides a serial data communications channel that can be programmed to operate with most of the current serial data transmission protocols. Synchronous or asynchronous mode, baud rate, character length, number of stop bits and even/odd parity are all program selectable. In addition, the user has the option, through jumper connections, of configuring the Serial I/O Interface as an EIA RS232C interface or as a Teletype-compatible current loop interface.

The Parallel I/O Interface, using two Intel 8255 Programmable Peripheral Interface devices, provides 48 signal lines for the transfer and control of data to or from peripheral devices. Eight lines already have a bidirectional driver and termination network permanently installed. The remaining 40 lines, however, are uncommitted. Sockets are provided for the installation of active driver networks or passive termination networks as required to meet the specific needs of the user system.

In this chapter, we will reiterate each of the options available to the user, and summarize, for easy reference, the specific information required to implement the user's tailored I/O configuration. Section 4.1 deals with the Serial I/O Interface, while Section 4.2 covers Parallel I/O options. Section 4.3 will describe general options not covered in the other two sections.
4.1 SERIAL I/O INTERFACE OPTIONS

There are three general areas of Serial I/O options:

1) choice of interface type, RS232C or current loop,
2) baud rate and program-selectable mode options,
3) choice of an interrupt request mechanism.

The first two are covered in the following paragraphs; the third, choice of interrupt mechanism, is quite simple and is fully explained in Section 3.5.4.

4.1.1 INTERFACE TYPE

The user has the choice of configuring the Serial I/O logic to present either an EIA RS232C or a 20 mA current loop interface to an external device. If a Teletype-compatible current loop interface is used, the 8251 I/O pins should be connected to the external Teletype lines as listed in Table 4-1. The reader control logic is controlled by the output DSR/ from the 8251. If an EIA RS23?C interface is used, the 8251 can assume the role of a "data set" (see Table 4-2a) or a partial "data processing terminal" (see Table 4-2b). Pin definitions for the 8251 USART are listed in Section 3.5.1.

4.1.2 BAUD RATE AND PROGRAM-SELECTABLE SERIAL I/O OPTIONS

Before beginning Serial I/O operations, the 8251 must be program-initialized to support the desired mode of operation. The CPU initializes the 8251 by outputting a set of control bytes to the USART device. These control words specify:

* synchronous or asynchronous operation,
* baud rate factor,
* character length,
* number of stop bits,
* even/odd parity.
* parity/no parity

1 In this role, cable modifications must be made to conform with RS232 standards.
TABLE 4-1. 20 mA CURRENT LOOP SERIAL I/O INTERFACE

<table>
<thead>
<tr>
<th>8251 PIN MNEMONIC</th>
<th>PIN NO.</th>
<th>PIN</th>
<th>CONNECTOR PIN NO.</th>
<th>JUMPER(3) CONNECTIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>TXD 19</td>
<td>TTY Tx</td>
<td>J3-25</td>
<td>-</td>
<td>1-2, 23-24</td>
</tr>
<tr>
<td>DTR/ 24</td>
<td>TTY RD CONTROL</td>
<td>J3-6</td>
<td>-</td>
<td>27-29, 30-31</td>
</tr>
<tr>
<td>(1) RTS/ 23</td>
<td>(CTS/)</td>
<td>-</td>
<td>-</td>
<td>27-29</td>
</tr>
<tr>
<td>(1) CTS/ 17</td>
<td>(RTS/)</td>
<td>-</td>
<td>-</td>
<td>33-34 (8-4, 56-57)</td>
</tr>
<tr>
<td>(2) TXC 9</td>
<td>(Baud Rate Clk)</td>
<td>-</td>
<td>35-36 (8-4, 56-57)</td>
<td></td>
</tr>
<tr>
<td>(2) RXC 25</td>
<td>(Baud Rate Clk)</td>
<td>-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TXD 3</td>
<td>TTY Rx</td>
<td>J3-22</td>
<td>-</td>
<td>38-39</td>
</tr>
<tr>
<td>-</td>
<td>TTY Rx RET</td>
<td>J3-23</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>TTY Tx RET</td>
<td>J3-24</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>TTY RD CTL RET</td>
<td>J3-16</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Notes: (1) The 8251's RTS/ output is connected to the CTS/ input through jumper pair 27-28. The command instruction word for the 8251 must enable RTS/.
(2) TXC and RXC are connected to the Baud Rate Clk line via jumpers 33-34 and 35-36. The Baud Rate Clk should be configured for 110 baud by connecting jumpers 8-4 and 56-57 (see Table 4-3), and the 8251 should be programmed for a baud rate factor of 64 (see Section 4.2).
(3) The SBC 80/10 comes with these jumper connections made.
### TABLE 4-2a. RS232C INTERFACE, "DATA SET" ROLE

<table>
<thead>
<tr>
<th>8251 PIN MNEMONIC</th>
<th>PIN NO.</th>
<th>LINE FUNCTION</th>
<th>CONNECTOR PIN NO.</th>
<th>JUMPER CONNECTIONS</th>
<th>JUMPER REMOVAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>RXD</td>
<td>3</td>
<td>TRANSMITTED DATA</td>
<td>J3-3</td>
<td>37-38</td>
<td>39-38</td>
</tr>
<tr>
<td>TXD</td>
<td>19</td>
<td>RECEIVED DATA</td>
<td>J3-5</td>
<td>2-3</td>
<td>1-2</td>
</tr>
<tr>
<td>(1) CTS/</td>
<td>17</td>
<td>REQ TO SEND</td>
<td>J3-7</td>
<td>27-28</td>
<td>27-29</td>
</tr>
<tr>
<td>RTS/</td>
<td>23</td>
<td>CLEAR TO SEND</td>
<td>J3-9</td>
<td>29-30</td>
<td>-</td>
</tr>
<tr>
<td>DTR/</td>
<td>24</td>
<td>DATA SET READY</td>
<td>J3-11</td>
<td>22-23</td>
<td>23-22</td>
</tr>
<tr>
<td>(2) DSR/</td>
<td>22</td>
<td>DATA TERMINAL RDY</td>
<td>J3-14</td>
<td>25-26</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>PROTECTIVE GROUND</td>
<td>J3-1</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>SIGNAL GROUND</td>
<td>J3-13</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

Notes: (1) The CTS/ input pin on the 8251 must be "low" to enable the 8251 to transmit.

(2) When connector pin J3-14 is jumpered (25-26) to the DSR/ input, J3-14 cannot be used to supply an external transmit clock.

(3) In the asynchronous mode, TXC and RXC can be connected to externally supplied clocks via jumpers 32-33 and 36-39, or they can be connected to the internal Baud Rate Clk via jumpers 33-34 and 35-36, regardless of the mode.

### TABLE 4-2b. RS232C INTERFACE, "DATA PROCESSING TERMINAL" ROLE

<table>
<thead>
<tr>
<th>8251 PIN MNEMONIC</th>
<th>PIN NO.</th>
<th>LINE FUNCTION</th>
<th>CONNECTOR PIN NO.</th>
<th>JUMPER CONNECTIONS</th>
<th>JUMPER REMOVAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>TXD</td>
<td>19</td>
<td>TRANSMITTED DATA</td>
<td>J3-5</td>
<td>2-3</td>
<td>1-2</td>
</tr>
<tr>
<td>RXD</td>
<td>3</td>
<td>RECEIVED DATA</td>
<td>J3-3</td>
<td>37-38</td>
<td>36-39</td>
</tr>
<tr>
<td>RTS/</td>
<td>23</td>
<td>REQ TO SEND</td>
<td>J3-9</td>
<td>29-30</td>
<td>27-29</td>
</tr>
<tr>
<td>(1) CTS/</td>
<td>17</td>
<td>CLEAR TO SEND</td>
<td>J3-7</td>
<td>27-28</td>
<td>-</td>
</tr>
<tr>
<td>DTR/</td>
<td>24</td>
<td>DATA TERMINAL RDY</td>
<td>J3-11</td>
<td>22-23</td>
<td>23-24</td>
</tr>
<tr>
<td>(3) TXC</td>
<td>9</td>
<td>TRANSMIT CLOCK</td>
<td>J3-14</td>
<td>32-33</td>
<td>26-25</td>
</tr>
<tr>
<td>(2) DSR/</td>
<td>22</td>
<td>DATA SET RDY</td>
<td>J3-14</td>
<td>25-26</td>
<td>-</td>
</tr>
<tr>
<td>(3) RXC</td>
<td>25</td>
<td>RECEIVE CLOCK</td>
<td>J3-22</td>
<td>36-39</td>
<td>35-36</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>PROTECTIVE GROUND</td>
<td>J3-1</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>SIGNAL GROUND</td>
<td>J3-13</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

Notes: (1) The CTS/ input pin on the 8251 must be "low" to enable the 8251 to transmit.

(2) When connector pin J3-14 is jumpered (25-26) to the DSR/ input, J3-14 cannot be used to supply an external transmit clock.

(3) In the asynchronous mode, TXC and RXC can be connected to externally supplied clocks via jumpers 32-33 and 36-39, or they can be connected to the internal Baud Rate Clk via jumpers 33-34 and 35-36, regardless of the mode.

1 In this role, cable modifications must be made to conform with RS232 standards.
As explained in Section 3.5.1, there are two types of control words: (1) Mode instruction and (2) Command instruction. The Mode instruction initializes the 8251 USART. Because the USART supports either synchronous or asynchronous operation, the Mode instruction has one format for synchronous operation and another for asynchronous. The two least significant bits of the Mode instruction byte specify the format. If DO and D1 both equal 0, synchronous operation is indicated; otherwise, it is asynchronous. The Mode instruction format for asynchronous operation is illustrated in Figure 3-8. The Mode instruction for synchronous operation is shown in Figure 3-9.

Notice in Figure 3-8 that the baud rate factor is specified by the two least significant bits of the instruction byte (labeled B1 and B2). During asynchronous communications, the Baud Rate Clock frequency supplied to the 8251's TXC and RXC input pins is divided by the baud rate factor to produce the effective baud rate (i.e., the frequency at which data bits are serially transmitted by the 8251 USART). Consequently, the Baud Rate Clock, as well as the program-selected baud rate factor, must be considered in implementing the desired effective baud rate. The Baud Rate Clock frequency is selected through various jumper connections as shown on sheet 4 of the SBC-80/10 schematic (Appendix A). The selection of an effective baud rate is summarized in Table 4-3.

Notice from the schematic that TXC and RXC inputs can be supplied by externally supplied clocks (via connector pins J3-14 and J3-22, respectively), instead of using the Baud Rate Clock, if jumpers 32-33 and 36-39 are connected and jumpers 33-34 and 35-36 are disconnected.
TABLE 4-3. BAUD RATE SELECTION

<table>
<thead>
<tr>
<th>JUMPER CONNECTION</th>
<th>SYNCHRONOUS MODE</th>
<th>EFFECTIVE BAUD RATE (Hz)</th>
<th>ASYNCHRONOUS MODE</th>
<th>BAUD RATE FACTOR=16(2)</th>
<th>BAUD RATE FACTOR=64(2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10-4</td>
<td>-</td>
<td>9600(3)</td>
<td></td>
<td>4800</td>
<td></td>
</tr>
<tr>
<td>11-4</td>
<td>-</td>
<td>4800</td>
<td></td>
<td>2400</td>
<td></td>
</tr>
<tr>
<td>12-4</td>
<td>-</td>
<td>2400</td>
<td></td>
<td>1200</td>
<td></td>
</tr>
<tr>
<td>5-4</td>
<td>38,400</td>
<td>1200</td>
<td>600</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6-4</td>
<td>19,200</td>
<td>600</td>
<td>300</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7-4</td>
<td>9600</td>
<td>300</td>
<td>75</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1) 8-4</td>
<td>4800</td>
<td>-</td>
<td></td>
<td>110 (TTY)</td>
<td></td>
</tr>
<tr>
<td>(1) 8-4, j</td>
<td>6980</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: (1) If jumper pair 56-57 is not connected, the frequency at jumper pole 8 is 4.8 KHZ. If jumper 56-57 is connected, however, the frequency at jumper pole 8 is 6.98 KHZ which, with a programmed baud rate factor of 64, provides an effective baud rate of approximately 110 baud for Teletype use.

(2) Baud rate factor is software selectable.

(3) Caution: Baud Rate Factor = 16

4.2 PARALLEL I/O OPTIONS

The Parallel I/O Interface consists of six 8-bit I/O ports implemented with two Intel 8255 Programmable Peripheral Interface devices. The primary user considerations in determining how to use each of the six I/O ports are:

1) Choice of operating mode (as defined in Section 3.6.1),
2) direction of data flow (input, output or bidirectional),
3) choice of driver/termination networks for port’s data path.

In the following paragraphs, we will define the capabilities of each port and summarize, in tables, that information which is necessary to use the port in each of its potential configurations. Each table will list the port I/O address, the control register address and the format for the control word which is output to the 8255 by the CPU Set and which specifies the particular configuration to be used. Each
table will also summarize all of the relevant information concerning the choice and use of driver/termination networks, the data polarity, the connecting of jumpers and what they enable, and any restrictions on the use of the other two ports in each group. Examples of suitable driver/termination networks are listed in Section 5.1.

4.2.1 PORT 1 (GROUP 1 PORT A)

Port 1 is the only port that already includes a permanent bi-directional driver/termination network (two 8226 Bidirectional Bus Drivers). Port 1 is also the only port which can be programmed to function in any one of the three 8255 operating modes, which were defined in Section 3.6.1. Before Port 1 is programmed for input or output in any one of the three modes, certain jumper connections must be made to allow the port to function properly in the chosen mode. Other jumper connections must be made to enable interrupts when Port 1 is in mode 1 or mode 2. In all, there are five potential configurations for Port 1. All of the necessary information for implementing each configuration has been summarized in the following tables:

<table>
<thead>
<tr>
<th>PORT 1 CONFIGURATIONS</th>
<th>TABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode</td>
<td>Direction</td>
</tr>
<tr>
<td>1. Mode 0</td>
<td>Input</td>
</tr>
<tr>
<td>2. Mode 0</td>
<td>Output (Latched)</td>
</tr>
<tr>
<td>3. Mode 1</td>
<td>Input (Strobed)</td>
</tr>
<tr>
<td>4. Mode 1</td>
<td>Output (Latched)</td>
</tr>
<tr>
<td>5. Mode 2</td>
<td>Bidirectional</td>
</tr>
</tbody>
</table>
### TABLE 4-4. PORT 1, MODE 0 INPUT CONFIGURATION

<table>
<thead>
<tr>
<th>PORT 1 ADDRESS: E4, CONTROL REGISTER ADDRESS: E7</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONTROL WORD FORMAT: 7 6 5 4 3 2 1 0</td>
</tr>
<tr>
<td>1 0 0 1 x x x x</td>
</tr>
<tr>
<td>DRIVER/TERMINATION NETWORKS: Two Intel® 8226 Bidirectional Bus Drivers permanently installed at A1 and A2.</td>
</tr>
<tr>
<td>DATA POLARITY: Negative-true.</td>
</tr>
<tr>
<td>JUMPER CONNECTIONS: 41-42 to enable input at 8226's. Remove 40-41.</td>
</tr>
<tr>
<td>PORT 2 RESTRICTIONS: None; port 2 can be programmed for mode 0 or mode 1, input or output (see Section 4.2.2)</td>
</tr>
<tr>
<td>PORT 3 RESTRICTIONS: None; port 3 can be programmed for mode 0, 8-bit input or output, unless port 2 is in mode 1 (see Section 4.2.3).</td>
</tr>
</tbody>
</table>

### TABLE 4-5. PORT 1, MODE 0 LATCHED OUTPUT CONFIGURATION

<table>
<thead>
<tr>
<th>PORT 1 ADDRESS: E4, CONTROL REGISTER ADDRESS: E7</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONTROL WORD FORMAT: 7 6 5 4 3 2 1 0</td>
</tr>
<tr>
<td>1 0 0 0 x x x x</td>
</tr>
<tr>
<td>DRIVER/TERMINATION NETWORKS: Two Intel® 8226 Bidirectional Bus Drivers permanently installed at A1 and A2.</td>
</tr>
<tr>
<td>DATA POLARITY: Negative-true.</td>
</tr>
<tr>
<td>JUMPER CONNECTIONS: 40-41 to enable output at 8226's.</td>
</tr>
<tr>
<td>PORT 2 RESTRICTIONS: None; port 2 can be programmed for mode 0 or mode 1, input or output (see Section 4.2.2)</td>
</tr>
<tr>
<td>PORT 3 RESTRICTIONS: None; port 3 can be programmed for mode 0, input or output, unless port 2 is in mode 1 (see Section 4.2.3).</td>
</tr>
</tbody>
</table>

4-8
TABLE 4-6. PORT 1, MODE 1 STROBED INPUT CONFIGURATION

<table>
<thead>
<tr>
<th>PORT 1 ADDRESS: E4, CONTROL REGISTER ADDRESS: E7</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONTROL WORD FORMAT: 7 6 5 4 3 2 1 0</td>
</tr>
<tr>
<td>[[0/1</td>
</tr>
</tbody>
</table>

**DRIVER/TERMINATION NETWORKS:** Two Intel®8226 Bidirectional Bus
Drivers permanently installed at A1 and A2. A driver network must be installed at A3 and a termination network must be installed at A4.

**DATA POLARITY:** Negative-true. The polarity of Port 3 control outputs is dependent on the type of driver installed at A3.

**JUMPER CONNECTIONS:** 41-42 to enable input at 8226's; connect 49-50 to enable interrupt request via INT55/. Remove 40-41, 48-49, 50-51.

**PORT 2 RESTRICTIONS:** None; port 2 can be programmed for mode 0 or mode 1, input or output (see Section 4.2.2).

**PORT 3 RESTRICTIONS:** Port 3 bits perform the following dedicated functions:

* Bits 0, 1 and 2 - dedicated to control of port 2 if port 2 is in mode 1 (see Tables 4-9 to 4-12).
* Bit 3 - INTR (interrupt request) output for port 1.
* Bit 4 - STB/ (strobe) input for port 1.
* Bit 5 - IBF (input buffer full) output for port 1.
* Bit 6 - Only one bit can be used. If input use bit 6; do not use bit 7. Bit 3 of Control Word=1. If output use bit 6.
* Bit 7 - 7 and remove jumper between 13-14; do not use bit 6. Bit 3 of Control Word=0.
TABLE 4-7. PORT 1, MODE 1 LATCHED OUTPUT CONFIGURATION

PORT 1 ADDRESS: E4, CONTROL REGISTER ADDRESS: E7

CONTROL WORD FORMAT: 

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

DRIVER/TERMINATION NETWORKS: Two Intel®8226 Bidirectional Bus Drivers permanently installed at A1 and A2. A driver network must be installed at A3 and a termination network must be installed at A4.

DATA POLARITY: Negative-true. The polarity of Port C control outputs is dependent on the type of driver installed at A3.

JUMPER CONNECTIONS: 40-41 to enable output at 8226's; connect 49-50 to enable interrupt request via INT55/. Remove 48-49, 50-51.

PORT 2 RESTRICTIONS: None; port 2 can be programmed for mode 0 or mode 1, input or output (see Section 4.2.2).

PORT 3 RESTRICTIONS: Port 3 bits perform the following dedicated functions:

*Bits 0, 1 and 2 - dedicated to the control of port 2 if port 2 is in mode 1 (see Tables 4-11 and 4-12).
*Bit 3 - INTR (interrupt request) output for port 1.
*Bit 4 - can be used for input if bit 3 of control word = 1
*Bit 5 - cannot be used if PC4 is used; can be used for output if control word bit 3 = 0 (PC4 cannot be used then).
*Bit 6 - ACK/ (acknowledge) input for port 1.
*Bit 7 - OBF/ (output buffer full) output for port 1.
TABLE 4-8. PORT 1, MODE 2 BIDIRECTIONAL CONFIGURATION

<table>
<thead>
<tr>
<th>Port 1 Address: E4,</th>
<th>Control Register Address: E7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control Word Format:</td>
<td>7 6 5 4 3 2 1 0</td>
</tr>
<tr>
<td></td>
<td>1 1 X X X X X X</td>
</tr>
</tbody>
</table>

Driver/Termination Networks: Two Intel® 8226 Bidirectional Bus Drivers permanently installed at A1 and A2. A driver network must be installed at A3 and a termination network must be installed at A4.

Data Polarity: Negative-true. The polarity of Port C control outputs is dependent on the type of driver installed at A3.

Jumper Connections: 41-43 to allow ACK/ input on PC6 to dynamically change data direction at 8226's (input when ACK/ = 1 and output when ACK/ = 0); connect 49-50 to enable interrupt request via INT55/.

Remove 40-41, 48-49, 50-51.

Port 2 Restrictions: None.

Port 3 Restrictions: Port 3 bits perform the following dedicated functions:

* Bits 0 and 1 - can be used for output if bit 3 of control word = 0
* Bit 2 - cannot be used if PC0 and PC1 are used; can be used for input if control word bit 3 = 1 (PC0 and PC1 cannot be used then).
* Bit 3 - INTR (interrupt request) output for port 1.
* Bit 4 - STB/ (strobe input for port 1.
* Bit 5 - IBF (input buffer full) output for port 1.
* Bit 6 - ACK/ (acknowledge) input for port 1.
* Bit 7 - OBF/ (output buffer full) output for port 1.
4.2.2 PORT 2 (GROUP 1 Port B)

Port 2 can be programmed for input or output in either mode 0 or mode 1. If Port 2 is to be used for input, in either mode, terminator networks must be installed in the sockets at A5 and A6. If Port 2 is to be used for output, in either mode, driver networks must be installed in the sockets at A5 and A6. When Port 2 is programmed for mode 1, interrupts can be enabled by connecting jumper pair 45-46. The four potential configurations for Port 2 are summarized in the following tables:

<table>
<thead>
<tr>
<th>Mode</th>
<th>Direction</th>
<th>Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Mode 0</td>
<td>Input</td>
</tr>
<tr>
<td>2.</td>
<td>Mode 0</td>
<td>Output (Latched)</td>
</tr>
<tr>
<td>3.</td>
<td>Mode 1</td>
<td>Input (Strobed)</td>
</tr>
<tr>
<td>4.</td>
<td>Mode 1</td>
<td>Output (Latched)</td>
</tr>
</tbody>
</table>

TABLE 4-9. PORT 2, MODE 0 INPUT CONFIGURATION

PORT 2 ADDRESS: E5, CONTROL REGISTER ADDRESS: E7

CONTROL WORD FORMAT: 7 6 5 4 3 2 1 0

| 1 | x | x | x | x | 0 | 1 | x |

DRIVER/TERMINATION NETWORKS: Termination networks must be installed at A5 and A6.

DATA POLARITY: Positive-true.

JUMPER CONNECTION: None.

PORT 1 RESTRICTIONS: None (see Section 4.2.1).

PORT 3 RESTRICTIONS: None, port 3 can be programmed for mode 0, input or output, unless port 1 is in mode 1 or mode 2 (see Section 4.2.3).
TABLE 4-10. PORT 2, MODE 0 LATCHED OUTPUT CONFIGURATION

<table>
<thead>
<tr>
<th>CONTROL WORD FORMAT:</th>
<th>7 6 5 4 3 2 1 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 x x x 0 0 x x</td>
<td></td>
</tr>
</tbody>
</table>

**DRIVER/TERMINATION NETWORKS:** Driver networks must be installed at A5 and A6.

**DATA POLARITY:** Negative-true, assuming that inverting drivers are at A5 and A6.

**JUMPER CONNECTIONS:** None.

**PORT 1 RESTRICTIONS:** None (see Section 4.2.1).

**PORT 3 RESTRICTIONS:** None, port 3 can be programmed for mode 0 or mode 1, 8-bit input or output, unless port 1 is in mode 1 or mode 2 (see Section 4.2.3).

TABLE 4-11. PORT 2, MODE 1 STROBED INPUT CONFIGURATION

<table>
<thead>
<tr>
<th>CONTROL WORD FORMAT:</th>
<th>7 6 5 4 3 2 1 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0 x x x 1 1 x x</td>
<td></td>
</tr>
</tbody>
</table>

**DRIVER/TERMINATION NETWORKS:** Termination networks must be installed at A5 and A6. A driver network must be installed at A3 and a termination network must be installed at A4.

**DATA POLARITY:** Positive-true. The polarity of Port C control outputs is dependent on the type of driver installed at A3.

**JUMPER CONNECTIONS:** 45-46 to enable interrupt request via INT55/. Remove 44-45, 46-47.

**PORT 1 RESTRICTIONS:** None.

**PORT 3 RESTRICTIONS:** Port 3 bits perform the following dedicated functions:

*Bit 0 - INTR (interrupt request) output for port 2.
*Bit 1 - IBF (input buffer full) output for port 2.
*Bit 2 - STB/ (strobe) input for port 2.
*Bit 3 to Bit 7 - dedicated to control of port 1 if port 1 is in mode 1 (see Tables 4-4 to 4-7).
### TABLE 4-12. PORT 2, MODE 1 LATCHED OUTPUT CONFIGURATION

<table>
<thead>
<tr>
<th>PORT 2 ADDRESS: E5, CONTROL REGISTER ADDRESS: E7</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CONTROL WORD FORMAT:</strong> 7 6 5 4 3 2 1 0</td>
</tr>
</tbody>
</table>
| ![Format Table](image)
| DRIVER/TERMINATION NETWORKS: Driver networks must be installed at A5 and A6. A driver network must be installed at A3 and a termination network must be installed at A4. |
| DATA POLARITY: Negative-true, assuming that inverting drivers are at A5 and A6. The polarity of Port C control outputs is dependent on the type of driver installed at A3. |
| JUMPER CONNECTIONS: 45-46 to enable interrupt request via INT55/Remove 44-45, 46-47. |
| PORT 1 RESTRICTIONS: None. |
| PORT 3 RESTRICTIONS: Port 3 bits perform the following dedicated functions: |
*Bit 0 - INTR (interrupt request) output for port 2. |
*Bit 1 - OBF/ (output buffer full) output for port 2. |
*Bit 2 - ACK/ (acknowledge) input for port 2. |
*Bit 3 - P3-7 - dedicated to control of port 1 if port 1 is in mode 1 (see Tables 4-4 to 4-7). |

### 4.2.3 PORT 3 (GROUP 1 PORT C)

The use of Port 3 is dependent on the modes programmed for Ports 1 and 2 (refer to Tables 4-4 to 4-12). While certain Port 3 bits are available if Port 1 is in mode 1 or if Port 2 is in mode 0, the use of Port 3 as an 8-bit data path is restricted to those configurations that have both Port 1 and Port 2 programmed for mode 0. In this case, all eight bits of Port 3 can be programmed for mode 0 input (see Table 4-13) or output (see Table 4-14). A 4-bit input/4-bit output configuration is never possible for group 1 Port C.
Note: If Ports 1 and 2 are not both in mode 0, then a driver network must be installed in the sockets at A3 and a termination network must be installed at A4, so that the Port 3 control lines can function properly.

4.2.4 PORTS 4 AND 5 (GROUP 2 PORTS A AND B)

Ports 4 and 5 can be programmed for input or output but only in mode 0. The two potential configurations for each port are summarized in the following tables:

<table>
<thead>
<tr>
<th>CONFIGURATIONS</th>
<th>TABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PORT</td>
<td>MODE</td>
</tr>
<tr>
<td>1. Port 4</td>
<td>Mode 0</td>
</tr>
<tr>
<td>2. Port 4</td>
<td>Mode 0</td>
</tr>
<tr>
<td>1. Port 5</td>
<td>Mode 0</td>
</tr>
<tr>
<td>2. Port 5</td>
<td>Mode 0</td>
</tr>
</tbody>
</table>

**TABLE 4-13. PORT 3, MODE 0, 8-BIT INPUT CONFIGURATION**

PORT 3 ADDRESS: E6, CONTROL REGISTER ADDRESS: E7

CONTROL WORD FORMAT: \[
\begin{array}{ccccccc}
& & & & & & 0 \\
7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\
\end{array}
\]

<table>
<thead>
<tr>
<th>DRIVER/TERMINATION NETWORKS:</th>
<th>Termination networks must be installed at A3 and A4.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA POLARITY:</td>
<td>Positive-true.</td>
</tr>
<tr>
<td>JUMPER CONNECTIONS:</td>
<td>46-47 and 44-45 to disable port 2 interrupts and enable P3-0; connect 48-49 and 50-51 to disable port 1 interrupts and enable P3-3.</td>
</tr>
<tr>
<td>PORT 1 AND 2 RESTRICTIONS:</td>
<td>Both ports 1 and 2 must be in mode 0.</td>
</tr>
</tbody>
</table>
### TABLE 4-14. PORT 3, MODE 0, 8-BIT LATCHED OUTPUT CONFIGURATION

<table>
<thead>
<tr>
<th>PORT 3 ADDRESS: E6, CONTROL REGISTER ADDRESS: E7</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONTROL WORD FORMAT: 7 6 5 4 3 2 1 0</td>
</tr>
<tr>
<td>1 0 0 x 0 0 x 0</td>
</tr>
<tr>
<td>DRIVER/TERMINATION NETWORKS: Driver networks must be installed at A3 and A4.</td>
</tr>
<tr>
<td>DATA POLARITY: Negative-true, assuming that inverting drivers are installed at A3 and A4.</td>
</tr>
<tr>
<td>JUMPER CONNECTIONS: 46-47, and 44-45 to disable port 2 interrupts and enable P3-0; connect 48-49 and 50-51 to disable port 1 interrupts and enable P3-3.</td>
</tr>
<tr>
<td>PORT 1 AND 2 RESTRICTIONS: Both ports 1 and 2 must be in mode 0.</td>
</tr>
</tbody>
</table>

### TABLE 4-15. PORT 4, MODE 0, INPUT CONFIGURATION

<table>
<thead>
<tr>
<th>PORT 4 ADDRESS: E8, CONTROL REGISTER ADDRESS: EB</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONTROL WORD FORMAT: 7 6 5 4 3 2 1 0</td>
</tr>
<tr>
<td>1 0 0 1 x 0 x x</td>
</tr>
<tr>
<td>DRIVER/TERMINATION NETWORKS: Termination networks must be installed at A7 and A8.</td>
</tr>
<tr>
<td>DATA POLARITY: Positive-true.</td>
</tr>
<tr>
<td>JUMPER CONNECTIONS: None.</td>
</tr>
<tr>
<td>PORT 5 AND 6 RESTRICTIONS: None; ports 5 and 6 can be programmed for mode 0, input or output (also see Section 4.2.5).</td>
</tr>
</tbody>
</table>
### TABLE 4-16. PORT 4, MODE 0 LATCHED OUTPUT CONFIGURATION

<table>
<thead>
<tr>
<th>Port 4 Address: E8</th>
<th>Control Register Address: EB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control Word Format:</td>
<td>7 6 5 4 3 2 1 0</td>
</tr>
<tr>
<td></td>
<td>1 0 0 0 x 0 x x</td>
</tr>
<tr>
<td>Driver/Termination Networks:</td>
<td>Driver networks must be installed at A7 and A8.</td>
</tr>
<tr>
<td>Data Polarity:</td>
<td>Negative-true, assuming that inverting drivers are installed at A7 and A8.</td>
</tr>
<tr>
<td>Jumper Connections:</td>
<td>None.</td>
</tr>
<tr>
<td>Port 5 and 6 Restrictions:</td>
<td>None; ports 5 and 6 can be programmed for mode 0, input or output (also see Section 4.2.5).</td>
</tr>
</tbody>
</table>

### TABLE 4-17. PORT 5, MODE 0 INPUT CONFIGURATION

<table>
<thead>
<tr>
<th>Port 5 Address: E9</th>
<th>Control Register Address: EB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control Word Format:</td>
<td>7 6 5 4 3 2 1 0</td>
</tr>
<tr>
<td></td>
<td>1 0 0 x x 0 1 x</td>
</tr>
<tr>
<td>Driver/Termination Networks:</td>
<td>Termination networks must be installed at A11 and A21.</td>
</tr>
<tr>
<td>Data Polarity:</td>
<td>Positive-true.</td>
</tr>
<tr>
<td>Jumper Connections:</td>
<td>None.</td>
</tr>
<tr>
<td>Port 4 and 6 Restrictions:</td>
<td>None; ports 4 and 6 can be programmed for mode 0, input or output (also see Section 4.2.5).</td>
</tr>
</tbody>
</table>
### TABLE 4-18. PORT 5, MODE 0 LATCHED OUTPUT CONFIGURATION

<table>
<thead>
<tr>
<th>PORT 5 ADDRESS:</th>
<th>E9, CONTROL REGISTER ADDRESS: EB</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONTROL WORD FORMAT:</td>
<td>7 6 5 4 3 2 1 0</td>
</tr>
<tr>
<td>1 0 0 x x 0 0 x</td>
<td></td>
</tr>
<tr>
<td>DRIVER/TERMINATION NETWORKS:</td>
<td>Driver networks must be installed at A11 and A21.</td>
</tr>
<tr>
<td>DATA POLARITY:</td>
<td>Negative-true, assuming that inverting drivers are installed at A11 and A21.</td>
</tr>
<tr>
<td>JUMPER CONNECTIONS:</td>
<td>None.</td>
</tr>
<tr>
<td>PORT 4 AND 6 RESTRICTIONS:</td>
<td>None; ports 4 and 6 can be programmed for mode 0, input or output (also Section 4.2.5).</td>
</tr>
</tbody>
</table>

4.2.5 PORT 6 (GROUP 2 PORT C)

All eight bits of Port 6 can be programmed for mode 0 input or output, or four bits can be programmed for mode 0 input while the other four bits are programmed for mode 0 output. The four potential configurations for Port 6 are summarized in the following tables:

<table>
<thead>
<tr>
<th>PORT 6 CONFIGURATIONS</th>
<th>TABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. MODE 0 8-BIT INPUT</td>
<td>Table 4-19</td>
</tr>
<tr>
<td>2. MODE 0 8-BIT OUTPUT (LATCHED)</td>
<td>Table 4-20</td>
</tr>
<tr>
<td>3. MODE 0 UPPER 4-BIT INPUT/LOWER 4-BIT OUTPUT</td>
<td>Table 4-21</td>
</tr>
<tr>
<td>4. MODE 0 UPPER 4-BIT OUTPUT/LOWER 4-BIT INPUT</td>
<td>Table 4-22</td>
</tr>
</tbody>
</table>
TABLE 4-19. PORT 6, MODE 0, 8-BIT INPUT CONFIGURATION

<table>
<thead>
<tr>
<th>PORT 6 ADDRESS: EA,</th>
<th>CONTROL REGISTER ADDRESS: EB</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONTROL WORD FORMAT:</td>
<td>[7 \ 6 \ 5 \ 4 \ 3 \ 2 \ 1 \ 0]</td>
</tr>
<tr>
<td>[1 \ 0 \ 0 \ x \ 1 \ 0 \ x \ 1]</td>
<td></td>
</tr>
<tr>
<td>DRIVER/TERMINATION NETWORKS:</td>
<td>Termination networks must be installed at A9 and A10.</td>
</tr>
<tr>
<td>DATA POLARITY:</td>
<td>Positive-true.</td>
</tr>
<tr>
<td>JUMPER CONNECTIONS:</td>
<td>None.</td>
</tr>
<tr>
<td>PORT 4 AND 5 RESTRICTIONS:</td>
<td>None (see Section 4.2.4).</td>
</tr>
</tbody>
</table>

TABLE 4-20. PORT 6, MODE 0, 8-BIT LATCHED OUTPUT CONFIGURATION

<table>
<thead>
<tr>
<th>PORT 6 ADDRESS: EA,</th>
<th>CONTROL REGISTER ADDRESS: EB</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONTROL WORD FORMAT:</td>
<td>[7 \ 6 \ 5 \ 4 \ 3 \ 2 \ 1 \ 0]</td>
</tr>
<tr>
<td>[1 \ 0 \ 0 \ x \ 0 \ 0 \ x \ 0]</td>
<td></td>
</tr>
<tr>
<td>DRIVER/TERMINATION NETWORKS:</td>
<td>Driver networks must be installed at A9 and A10.</td>
</tr>
<tr>
<td>DATA POLARITY:</td>
<td>Negative-true, assuming that inverting drivers are installed at A9 and A10.</td>
</tr>
<tr>
<td>JUMPER CONNECTIONS:</td>
<td>None.</td>
</tr>
<tr>
<td>PORT 4 AND 5 RESTRICTIONS:</td>
<td>None (see Section 4.2.4).</td>
</tr>
</tbody>
</table>
TABLE 4-21. PORT 6, MODE 0 UPPER 4-BIT INPUT/LOWER 4-BIT LATCHED OUTPUT CONFIGURATION

PORT 6 ADDRESS: EA, CONTROL REGISTER ADDRESS: EB

CONTROL WORD FORMAT: 7 6 5 4 3 2 1 0
                      \   \   \   \   \   \ 0
                      1 0 0 x 1 0 x 0

DRIVER/TERMINATION NETWORKS: A termination network must be installed at A9 and a driver network must be installed at A10.

DATA POLARITY: The upper 4-bits will be in positive-true form; however, the lower four bits will be in negative-true form if an inverting driver is installed at A10.

JUMPER CONNECTIONS: None.

PORT 4 AND 5 RESTRICTIONS: None (see Section 4.2.4).

TABLE 4-22. PORT 6, MODE 0 UPPER 4-BIT LATCHED OUTPUT/LOWER 4-BIT INPUT CONFIGURATION

PORT 6 ADDRESS: EA, CONTROL REGISTER ADDRESS: EB

CONTROL WORD FORMAT: 7 6 5 4 3 2 1 0
                      \   \   \   \   \   \ a
                      1 0 0 x 0 0 x 1

DRIVER/TERMINATION NETWORKS: A driver network must be installed at A9 and a termination network must be installed at A10.

DATA POLARITY: The lower 4-bits will be in positive-true form; however, the upper 4-bits will be in negative-true form if an inverting driver is installed at A9.

JUMPER CONNECTIONS: None.

PORT 4 AND 5 RESTRICTIONS: None (see Section 4.2.4).
### TABLE 4-23. PARALLEL I/O ADDRESS AND SOCKET ASSIGNMENTS

<table>
<thead>
<tr>
<th>PORT</th>
<th>I/O ADDRESS</th>
<th>SOCKET NUMBERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>E4</td>
<td>BI-DIRECTIONAL DRIVER/ TERMINATOR AT A1, A2</td>
</tr>
<tr>
<td>2</td>
<td>E5</td>
<td>A5, A6</td>
</tr>
<tr>
<td>3</td>
<td>E6</td>
<td>A3, A4*</td>
</tr>
<tr>
<td>4</td>
<td>E8</td>
<td>A7, A8</td>
</tr>
<tr>
<td>5</td>
<td>E9</td>
<td>A11, A21</td>
</tr>
<tr>
<td>6</td>
<td>EA</td>
<td>A9, A10**</td>
</tr>
</tbody>
</table>

*Note requirements specified in Tables 4-4 through 4-14.

**Note requirements specified in Tables 4-15 through 4-22.
4.3 GENERAL OPTIONS

There are several other options that may be useful. Details are provided in the following paragraphs.

4.3.1 SYSTEM RESET OUTPUT

The user can enable a SYSTEM RESET output from the SBC 80/10 by connecting jumper pair 54-55. This allows the reset signal which is generated on the SBC 80/10 during power-up sequences (see Section 3.1.5) to be made available to other modules in the system via connector Pl-14. Notice on the schematic that a SYSTEM RESET input is accepted by the SBC 80/10 and Pl-14 and applied to the 8080 regardless of jumper connections.

4.3.2 DISABLE BUS CLOCK SIGNALS

The bus clock BCLK/ (connector pin Pl-13) or the constant clock CCLK/ (P-31) outputs can be disabled (if more drive, or a different frequency is needed) by disconnecting jumper pair 61-63 or 62-64, respectively. When connected, both BCLK/ and CCLK/ provide a 9.216 MHz timing reference to other modules.

4.3.3 ACKNOWLEDGE INPUTS

The SBC bus has defined two types of acknowledges; transfer acknowledge (XACK/) and advance acknowledge (AACK/). XACK/ is the required response of a memory or I/O port which indicates that the specified read/write operation has been completed. That is, data has been placed on (READ command) or accepted from (WRITE command) the system data bus lines. XACK/ is asynchronous with BCLK/. AACK/ is an advance acknow-
ledge in response to a memory or I/O port command. This optional acknowledge is used only with 8080 CPU-based systems where maximum system performance is needed. Figure 4-2 shows timing of the SBC 80/10 "READING" memory using the AACK/ signal.

AACK/ is a response to a READ command indicating that data will be valid on the bus by the time the 8080 needs it. Thus, if the access time of the slave device is less than \( t_{\text{ACC}} \) (command to data needed by the 8080), the slave module has \( t_{\text{SKD}} \) (command to 8080 sample point of bus acknowledge) to indicate that the data on the bus will be valid when the 8080 needs it. If the access time of a module is less than \( t_{\text{SKD}} \) then only XACK/ need be used and the 8080 will run at maximum speed.
If the access time of a module is greater than $t_{\text{8KD}}$, but less than $t_{\text{ACC}}$, AACK/ must be used, if the 8080 is to run at maximum speed. If AACK/ is not used and instead XACK/ is used, the 8080 will execute one more wait state than is necessary. AACK/ is asynchronous with BCLK/.

\begin{center}
\begin{tikzpicture}
\node at (0,0) {READ COMMAND/};
\node at (0,-1) {DATA};
\node at (0,-2) {AACK/};
\node at (0,-3) {XACK/};
\node at (1.5,0) {\textsc{8080 Ready Sampling Point}};
\node at (1.5,-1) {\textsc{8080 'reads'}};
\node at (1.5,-2) {DATA VALID};
\node at (1.5,-3) {\textsc{8080 Writing}};
\draw [->] (-1,0) -- (-1,-1) node [midway, right] {$t_{\text{ACC}}$};
\draw [->] (-1,-1) -- (-1,-2) node [midway, right] {$t_{\text{8KD}}$};
\draw [->] (-1,-2) -- (-1,-3);
\draw [->] (0,0) -- (0,-1) node [midway, right] {$\star$};
\draw [->] (0,-1) -- (0,-2) node [midway, right] {$\star$};
\draw [->] (0,-2) -- (0,-3);
\end{tikzpicture}
\end{center}

FIGURE 4-2. READ COMMAND WITH AACK/

AACK/ is also an advance response to a WRITE command indicating that the slave module will have accepted the data from the system bus by the time the 8080 has completed the WRITE. Figure 4-3 shows timing of the SBC 80/10 "WRITING" memory using the AACK/ signal.

4-24
When modules that generate proper AACK/ are used with the SBC 80/10, jumper pair 52-53 should be connected to allow AACK/ to be accepted (at P1-25) and gated to the RDYIN pin on the 8224 clock generator. If this option is used, caution should be taken to insure that all the modules on the bus meet the SBC 80/10 timing requirements.
4.3.4 INTERRUPT SOURCES

There are six sources of interrupts on the SBC 80/10 board, two from the serial I/O section (see 3.5.4), two from the parallel I/O section (see 3.6), and two from external sources. One of the external sources is INTR1/. INTR1/ is connected to the SBC bus (P1-42) and is the only interrupt line that other SBC modules can use to interrupt the SBC 80/10. The other external interrupt is EXT INTR0/. This interrupt is connected to the parallel I/O connector J1 (pin 49) and can be used by an external device to interrupt the SBC 80/10. Both external interrupts are negative true logic, a TTL low ($V_{IN} < 0.4$ volts) will cause the 8080 to interrupt and execute a RST 7 instruction. The processor can then read in the status registers of the possible interrupting devices to determine which device generated the interrupt. Then the processor can jump to the correct interrupt service routine, service that device, enable interrupts, and return.

4.4 DEFAULT OPTIONS

Table 4-24 lists the default options jumpered on the SBC 80/10. These options permit the SBC 80/10 to communicate to a TTY; they also provide power-up reset, bus clock, and the communication clock to the system bus. If the SBC 80/10 is driving the bus clock (BCLK/) and/or the communications clock (CCLK/), the system bus must be limited to 7 inches. This limitation is due to the SBC 80/10's limited drive capability on these clock lines. The system bus can be extended beyond 7 inches if the user provides BCLK/ and CCLK/.
### TABLE 4-24. DEFAULT OPTION

<table>
<thead>
<tr>
<th>DEFAULT JUMPERS</th>
<th>REFERENCE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 2</td>
<td>4.1.1</td>
<td>Connect 8251 T_D to 20 mA Current Loop Driver</td>
</tr>
<tr>
<td>23 - 24</td>
<td></td>
<td>Connect 8251 DTR/ to TTY Reader Control Circuit</td>
</tr>
<tr>
<td>39 - 38</td>
<td></td>
<td>Connect 8251 R_D to 20 mA Current Loop Receiver</td>
</tr>
<tr>
<td>4 - 8</td>
<td>4.1.2</td>
<td>Generates 6.98K Baud Rate Clock</td>
</tr>
<tr>
<td>57 - 56</td>
<td></td>
<td>Generates 6.98K Baud Rate Clock</td>
</tr>
<tr>
<td>34 - 33</td>
<td></td>
<td>Connect 8251 T_x Clock to Baud Rate Clock</td>
</tr>
<tr>
<td>35 - 36</td>
<td></td>
<td>Connect 8251 R_x Clock to Baud Rate Clock</td>
</tr>
<tr>
<td>27 - 29</td>
<td></td>
<td>Connect 8251 RTS/ to 8251 CTS/</td>
</tr>
<tr>
<td>19 - 20</td>
<td>3.5.4</td>
<td>Disable T_RDY Interrupt from 8251</td>
</tr>
<tr>
<td>16 - 15</td>
<td>3.5.4</td>
<td>Disable R_RDY Interrupt from 8251</td>
</tr>
<tr>
<td>26 - 25</td>
<td>4.1.2</td>
<td>Connect DTR/ Receiver to 8251 DSR/ Input</td>
</tr>
<tr>
<td>30 - 31</td>
<td>4.1.2</td>
<td>Connect Set Clear to Send Driver to +12V</td>
</tr>
<tr>
<td>40 - 41</td>
<td>4.2.1</td>
<td>Enable Port 1 Bi-directional Drivers to Output</td>
</tr>
<tr>
<td>54 - 55</td>
<td>4.3.1</td>
<td>Connect Power-Up Reset to System Bus</td>
</tr>
<tr>
<td>62 - 64</td>
<td>4.3.2</td>
<td>Connect 9.216 MHz Clock to Communication Clock Line</td>
</tr>
<tr>
<td>61 - 63</td>
<td>4.3.2</td>
<td>Connect 9.216 MHz Clock to Bus Clock Line</td>
</tr>
<tr>
<td>*65 - 66</td>
<td>4.5</td>
<td>Configures SBC 80/10A for 4K ROM/PROM</td>
</tr>
<tr>
<td>*68 - 69</td>
<td>4.5</td>
<td></td>
</tr>
<tr>
<td>*73 - 74</td>
<td>4.5</td>
<td></td>
</tr>
<tr>
<td>*76 - 78</td>
<td>4.5</td>
<td></td>
</tr>
</tbody>
</table>

*Used on SBC 80/10A only.
4.5 JUMPER CONFIGURATION FOR ROM/EPROM INSTALLATION

The SBC 80/10A has jumpers which allow installation of up to 4K or up to 8K bytes of read only memory. Up to 4K bytes can be installed using Intel's 8708 Erasable and Electrically Reprogrammable ROMs (EPROM), Intel's 8308 Metal Masked ROMs, or Intel's 2758 Erasable and Electrically Reprogrammable ROMs (EPROM). Up to 8K bytes can be installed using Intel's 2716 Erasable and Electrically Reprogrammable ROMs (EPROM) or Intel's 2316E Metal Masked ROMs. Table 4-25 lists the jumper configurations for 4K and 8K bytes of read only memory. Table 4-26 lists the addresses for each PROM socket in 4K and 8K configurations.

### TABLE 4-25. PROM JUMPER CONFIGURATION

<table>
<thead>
<tr>
<th>JUMPER</th>
<th>4K</th>
<th></th>
<th>8K</th>
<th></th>
<th>4K</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>65 - 66</td>
<td>68 - 69</td>
<td>73 - 74</td>
<td>76 - 78</td>
<td>66 - 67</td>
<td>69 - 70</td>
<td>74 - 75</td>
</tr>
</tbody>
</table>

*Using Intel's 2758 Erasable and Electrically Reprogrammable ROMs (EPROM)*

### TABLE 4-26. PROM ADDRESSES

<table>
<thead>
<tr>
<th>CHIP ADDRESS</th>
<th>A23</th>
<th>A24</th>
<th>A25</th>
<th>A26</th>
</tr>
</thead>
<tbody>
<tr>
<td>4K</td>
<td>0 - 3FF</td>
<td>400 - 7FF</td>
<td>800 - BFF</td>
<td>C00 - FFF</td>
</tr>
<tr>
<td>8K</td>
<td>0 - 7FF</td>
<td>1000 - 17FF</td>
<td>800 - FFF</td>
<td>1800 - 1FFF</td>
</tr>
</tbody>
</table>
CHAPTER 5

SYSTEM INTERFACING

The SBC-80/10, with its memory and I/O ports, is a complete computer on a single printed circuit board. However, the SBC-80/10 can also serve as a primary master module within an expanded system, communicating with numerous memory and I/O modules. In this chapter we identify each of the SBC-80/10's external connections and define all signals on the external system bus.

5.1 ELECTRICAL CONNECTIONS

The SBC-80/10 comes on a 12.00 X 6.75 inch printed circuit board, 0.50 inch thick and weighing 12 oz. The DC power requirements are listed in Table 7-1.

The SBC-80/10 has five edge connectors, as shown in Figure 5-1. Edge connectors at the top of the module are designed for compatibility with both flat cable and round cable hardware. All parallel I/O functions are paired with an independent signal ground pin. This allows flat cable implementation to utilize an alternate signal/ground scheme for reduction of cross talk. Round cables may easily be implemented as twisted pair with an individual ground pin for every return wire. The serial connection hardware has similar flexibility but ground return lines are not as extensive. The connector is wired for RS232C compatibility, thus, only one signal ground is provided.
CAUTION

All pin numbers listed in the following tables refer to numbers printed on the board, not to mating connector pin positions. When specifying pin numbers for cable harnesses, use caution since pin numbering is not necessarily the same as the connector pin numbering scheme.

The Parallel I/O Interface communicates with external I/O devices via two 50-pin double-sided PC edge connectors (J1 and J2), 0.1 inch centers. External devices can be attached to J1 or J2 using any of the following mating connectors:

<table>
<thead>
<tr>
<th>J1 and J2 Mating Connectors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connector Type</td>
</tr>
<tr>
<td>Flat Cable</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Soldered</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Wire-wrap</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Crimp</td>
</tr>
</tbody>
</table>

Tables 5-1 and 5-2 provide pin lists for the J1 and J2 connectors, respectively. The following TTL line drivers and Intel terminators are all compatible with the I/O driver sockets in the Parallel I/O Interface:
### Table 5-1. Pin Assignments for Connector J1

<table>
<thead>
<tr>
<th>PIN</th>
<th>SIGNAL</th>
<th>PIN</th>
<th>SIGNAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PORT 2 - BIT 3</td>
<td>2</td>
<td>GND</td>
</tr>
<tr>
<td>3</td>
<td>PORT 2 - BIT 2</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>PORT 2 - BIT 1</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>PORT 2 - BIT 0</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>PORT 2 - BIT 4</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>PORT 2 - BIT 5</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>PORT 2 - BIT 6</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>PORT 2 - BIT 7</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>PORT 3 - BIT 3</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>PORT 3 - BIT 2</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>PORT 3 - BIT 4</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>PORT 3 - BIT 6</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>PORT 3 - BIT 0</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>PORT 3 - BIT 5</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>PORT 3 - BIT 1</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>PORT 3 - BIT 7</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>PORT 1 - BIT 7</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>PORT 1 - BIT 6</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>PORT 1 - BIT 5</td>
<td>38</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>PORT 1 - BIT 4</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>PORT 1 - BIT 1</td>
<td>42</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>PORT 1 - BIT 0</td>
<td>44</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>PORT 1 - BIT 2</td>
<td>46</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>PORT 1 - BIT 3</td>
<td>48</td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>EXT INTR 1/</td>
<td>50</td>
<td>GND</td>
</tr>
</tbody>
</table>
The Serial I/O Interface communicates with an external I/O device via a 26-pin double-sided PC edge connector (J3), 0.1 inch centers. An external device can be connected to J3 using a 3M 3462-0001 flat cable connector or one of the following soldered connectors: TI H312113 or AMP 1-583715-1. Table 5-3 provides a pin list for connector J3.

The SBC-80/10 communicates with other system modules via an 86-pin double-sided edge connector (P1), 0.156 inch centers. This

---

**TABLE 5-2. PIN ASSIGNMENTS FOR CONNECTOR J2**
(Parallel I/O Interface - Group 2)

<table>
<thead>
<tr>
<th>PIN</th>
<th>SIGNAL</th>
<th>PIN</th>
<th>SIGNAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GND</td>
<td>2</td>
<td>GND</td>
</tr>
<tr>
<td>3</td>
<td>PORT 5 - BIT 3</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>PORT 5 - BIT 0</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>PORT 5 - BIT 1</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>PORT 5 - BIT 2</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>PORT 5 - BIT 4</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>PORT 5 - BIT 5</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>PORT 5 - BIT 6</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>PORT 5 - BIT 7</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>PORT 6 - BIT 3</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>PORT 6 - BIT 2</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>PORT 6 - BIT 1</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>PORT 6 - BIT 0</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>PORT 6 - BIT 4</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>PORT 6 - BIT 5</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>PORT 6 - BIT 6</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>PORT 6 - BIT 7</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>PORT 4 - BIT 7</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>PORT 4 - BIT 6</td>
<td>38</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>PORT 4 - BIT 5</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>PORT 4 - BIT 4</td>
<td>42</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>PORT 4 - BIT 0</td>
<td>44</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>PORT 4 - BIT 1</td>
<td>46</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>PORT 4 - BIT 2</td>
<td>48</td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>PORT 4 - BIT 3</td>
<td>50</td>
<td></td>
</tr>
</tbody>
</table>

---
edge connector will accept any of the following mating connectors:

CDC VPB01E43A000A1, Micro Plastics MP-0156-43-BW-4 or ARCO AE 443WP1.

Section 5.2 defines each of the external system bus signals and includes a pin list for P1 (Table 5-5).

**TABLE 5-3. PIN ASSIGNMENTS FOR CONNECTOR J3**
(Serial I/O Interface)

<table>
<thead>
<tr>
<th>PIN</th>
<th>SIGNAL NAME</th>
<th>PIN</th>
<th>SIGNAL NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CHASSIS GND</td>
<td>2</td>
<td>TTY RD CONTROL</td>
</tr>
<tr>
<td>3</td>
<td>TRANSMITTED DATA</td>
<td>4</td>
<td>TTY RD CONTROL</td>
</tr>
<tr>
<td>5</td>
<td>RECEIVED DATA</td>
<td>6</td>
<td>TX CLK/DATA TERMINAL RDY</td>
</tr>
<tr>
<td>7</td>
<td>REQ TO SEND</td>
<td>8</td>
<td>TTY RD CONTROL RETURN</td>
</tr>
<tr>
<td>9</td>
<td>CLEAR TO SEND</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>DATA SET READY</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>GND</td>
<td>14</td>
<td>RECEIVE CLK/TYY RX DATA</td>
</tr>
<tr>
<td>15</td>
<td>DATA CARRIER RETURN</td>
<td>16</td>
<td>TTY Tx DATA RETURN</td>
</tr>
<tr>
<td>17</td>
<td></td>
<td>18</td>
<td>GND</td>
</tr>
<tr>
<td>19</td>
<td></td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td></td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>TTY Rx DATA</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>TTY Tx DATA</td>
<td>26</td>
<td></td>
</tr>
</tbody>
</table>

The 60-pin double-sided edge connector labeled P2 in Figure 5-1 allows access to various test points on the SBC-80/10 (see Table 5-4).

The following wire-wrap connectors will attach to P2:

CDC VPB01E30A00A2,
TI H311130 and
AMP PE5-14559
### TABLE 5-4. PIN ASSIGNMENTS FOR CONNECTOR P2 (Auxiliary Connector)

<table>
<thead>
<tr>
<th>SIGNAL NAME</th>
<th>PIN ASSIGNMENT</th>
<th>COMMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSC</td>
<td>P2 - 28</td>
<td>TEST POINT</td>
</tr>
<tr>
<td>RAM 3C00 ENABLE/</td>
<td>P2 - 30</td>
<td></td>
</tr>
<tr>
<td>RAM 3D00 ENABLE/</td>
<td>P2 - 32</td>
<td></td>
</tr>
<tr>
<td>RAM 3E00 ENABLE/</td>
<td>P2 - 34</td>
<td></td>
</tr>
<tr>
<td>RAM 3F00 ENABLE/</td>
<td>P2 - 36</td>
<td></td>
</tr>
<tr>
<td>OSC INH/</td>
<td>P2 - 40</td>
<td></td>
</tr>
<tr>
<td>DATA BUS INH/</td>
<td>P3 - 42</td>
<td></td>
</tr>
<tr>
<td>BAUD RATE CLK TTY</td>
<td>P2 - 44</td>
<td></td>
</tr>
<tr>
<td>COUNT 1 ENABLE 1</td>
<td>P2 - 46</td>
<td></td>
</tr>
<tr>
<td>BAUD RATE CLK</td>
<td>P2 - 50</td>
<td></td>
</tr>
<tr>
<td>COUNT 2 ENABLE/</td>
<td>P2 - 52</td>
<td></td>
</tr>
<tr>
<td>TIME OUT ENABLE/</td>
<td>P2 - 54</td>
<td></td>
</tr>
<tr>
<td>B &amp; C CLK SET/</td>
<td>P2 - 55</td>
<td></td>
</tr>
<tr>
<td>STATUS STROBE</td>
<td>P2 - 56</td>
<td></td>
</tr>
<tr>
<td>RDY IN INH/</td>
<td>P2 - 57</td>
<td></td>
</tr>
<tr>
<td>BAUD RATE CLEAR/</td>
<td>P2 - 58</td>
<td></td>
</tr>
<tr>
<td>OSC/2</td>
<td>P2 - 60</td>
<td>TEST POINT</td>
</tr>
</tbody>
</table>

### 5.2 EXTERNAL SBC 80/10 SYSTEM BUS SUMMARY

A significant measure of the SBC-80/10's power and flexibility can be attributed to its external system bus. In expanded systems, the external bus structure allows for master-slave relationships between the various system modules. The bus includes its own clock (BCLK/) which is derived independently from the processor clock. Actual transfers via the bus, however, proceed asynchronously with respect to the bus clock. Thus, the transfer speed is dependent on the transmitting and receiving devices only. Once a module has gained
control of the bus by activating the BPRN input to the SBC-80/10, single or multiple read/write transfers can proceed at a maximum rate of 5 million data bytes per second. The 16 system address lines allow the SBC-80/10 to support up to 65,536 bytes of storage. The signal lines on the external system bus are defined as follows:

**BCLK/**  
**Bus clock;** used to synchronize bus control circuits on all master modules. BCLK/ has a period of ~110 nanoseconds (9.216 MHz frequency), 30% - 70% duty cycle. BCLK/ may be slowed, stopped or single stepped, if desired (see Section 4.4).

**INIT/**  
**Initialization signal;** resets the entire system to a known internal state.

**BPRN**  
**Bus priority input signal;** indicates to the SBC-80/10 that a higher priority master module is requesting use of the system bus. BPRN suspends the processing activity and drivers of the SBC-80/10.

**BUSY/**  
**Bus busy signal;** indicates that the bus is currently in use. BUSY/ prevents all other master modules from gaining control of the bus. BUSY/ is driven by the HLDA/ output from the SBC-80/10 in response to a BPRN input. It indicates that the bus is available.

**MRDC/**  
**Memory read command;** indicates that the address of a memory location has been placed on the system address lines and specifies that the contents of the addressed location are to be read and placed on the system data bus.

5-8
MWTC/ Memory write command; indicates that the address of a memory location has been placed on the system address lines and that a data word has been placed on the system data bus. MWTC/ specifies that the data word is to be written into the addressed memory location.

IORC/ I/O read command; indicates that the address of an input port has been placed on the system address bus and that the data at that input port is to be read and placed on the system data bus.

IOWC/ I/O write command; indicates that the address of an output port has been placed on the system address bus and that the contents of the system data bus are to be output to the addressed port.

XACK/ Transfer acknowledge signal; the required response of an external memory location or I/O port which indicates that the specified read/write operation has been completed. That is, data has been placed on, or accepted from, the system data bus lines.

AACK/ Advance acknowledge signal; used with 8080 CPU-based systems. AACK/ is an advance acknowledge, in response to a memory read command, that allows the memory to complete the access without requiring the CPU to wait.
## TABLE 5-5. PIN ASSIGNMENTS FOR CONNECTOR P1
(External System Bus)

<table>
<thead>
<tr>
<th>(COMPONENT SIDE)</th>
<th>(CIRCUIT SIDE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIN</td>
<td>MNEMONIC</td>
</tr>
<tr>
<td>1</td>
<td>GND</td>
</tr>
<tr>
<td>3</td>
<td>VCC</td>
</tr>
<tr>
<td>5</td>
<td>VCC</td>
</tr>
<tr>
<td>7</td>
<td>VDD</td>
</tr>
<tr>
<td>9</td>
<td>VBB</td>
</tr>
<tr>
<td>11</td>
<td>GND</td>
</tr>
<tr>
<td>13</td>
<td>BCLK/</td>
</tr>
<tr>
<td>15</td>
<td>BPRN</td>
</tr>
<tr>
<td>17</td>
<td>BUSY/</td>
</tr>
<tr>
<td>19</td>
<td>MRDC/</td>
</tr>
<tr>
<td>21</td>
<td>IORC/</td>
</tr>
<tr>
<td>23</td>
<td>XACK/</td>
</tr>
<tr>
<td>25</td>
<td>AACK/</td>
</tr>
<tr>
<td>27</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>CCLK/</td>
</tr>
<tr>
<td>33</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>ADRE/</td>
</tr>
<tr>
<td>45</td>
<td>ADRC/</td>
</tr>
<tr>
<td>47</td>
<td>ADR/A</td>
</tr>
<tr>
<td>49</td>
<td>ADR8/</td>
</tr>
<tr>
<td>51</td>
<td>ADR6/</td>
</tr>
<tr>
<td>53</td>
<td>ADR4/</td>
</tr>
<tr>
<td>55</td>
<td>ADR2/</td>
</tr>
<tr>
<td>57</td>
<td>ADR0/</td>
</tr>
<tr>
<td>59</td>
<td></td>
</tr>
<tr>
<td>61</td>
<td></td>
</tr>
<tr>
<td>63</td>
<td></td>
</tr>
<tr>
<td>65</td>
<td></td>
</tr>
<tr>
<td>67</td>
<td>DAT6/</td>
</tr>
<tr>
<td>69</td>
<td>DAT4/</td>
</tr>
<tr>
<td>71</td>
<td>DAT2/</td>
</tr>
<tr>
<td>73</td>
<td>DAT0/</td>
</tr>
<tr>
<td>75</td>
<td>GND</td>
</tr>
<tr>
<td>77</td>
<td>VBB/</td>
</tr>
<tr>
<td>79</td>
<td>VAA</td>
</tr>
<tr>
<td>81</td>
<td>VCC</td>
</tr>
<tr>
<td>83</td>
<td>VCC</td>
</tr>
<tr>
<td>85</td>
<td>GND</td>
</tr>
</tbody>
</table>

>D> Used by Intellec® MDS Bus.
CCLK/ constant clock; provides a clock signal of constant frequency (9.216 MHz) for use by option memory and I/O expansion boards. CCLK/ coincides with BCLK/ and has a period of ~110 nanoseconds.

INTR1/ Externally generated interrupt request.

ADRO/-ADRF/ 16 Address lines: used to transmit the address of the memory location or I/O port to be accessed. ADRF/ is the most significant bit.

DATO/-DAT7/ Bi-directional data lines; used to transmit/receive information to/from a memory location or I/O port. DAT7/ is the most significant bit.

5.3 RS232C CABLING

When the Serial I/O Interface is configured as an RS232C interface, the J3 edge connector can be cabled such that a RS232C pin-compatible connector is presented to the user's terminal or modem. A 26-pin mating connector, 3M 3462-0001, should be attached to the J3 edge connector on the SBC-80/10 and to a 25-wire flat cable, 3M 3349/25. The flat cable is, in turn, attached to the RS232C pin-compatible connector, 3M 3483-1000. Table 5-6 equates the J3 edge connector pins with the associated RS232-compatible pins on the 3M 3483-1000 connector.

Note: Using this 3M cable assemble, the RS232C connector pin-outs are MDS compatible. That is, if the SBC 80/10 is set up to drive a TTY, an MDS modified TTY can be used directly with the SBC 80/10.
### Table 5-6: J3/RS232C Connector Pin Correspondence

<table>
<thead>
<tr>
<th>SIGNAL NAME</th>
<th>J3 CONNECTOR PIN NO.</th>
<th>RS232C CONNECTOR PIN NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHASSIS GND</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>TRANSMITTED DATA</td>
<td>2</td>
<td>14</td>
</tr>
<tr>
<td>RECEIVED DATA</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>TTY RD CONTROL</td>
<td>4</td>
<td>15</td>
</tr>
<tr>
<td>REQ TO SEND</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>CLEAR TO SEND</td>
<td>6</td>
<td>16</td>
</tr>
<tr>
<td>DATA SET READY</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>GND</td>
<td>8</td>
<td>17</td>
</tr>
<tr>
<td>Tx CLK/DATA TERMINAL RDY</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td>DATA CARRIER RETURN</td>
<td>10</td>
<td>18</td>
</tr>
<tr>
<td>TTY RD CONTROL RETURN</td>
<td>11</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>19</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>11</td>
</tr>
<tr>
<td>RECEIVE CLK/TTY Rx DATA RETURN</td>
<td>22</td>
<td>24</td>
</tr>
<tr>
<td>TTY Rx DATA</td>
<td>23</td>
<td>12</td>
</tr>
<tr>
<td>TTY Tx DATA RETURN</td>
<td>24</td>
<td>25</td>
</tr>
<tr>
<td>TTY Tx DATA</td>
<td>25</td>
<td>13</td>
</tr>
</tbody>
</table>
5.4 TELETYPe MODIFICATIONS

The ASR-33 Teletypewriter must be modified for use with the SBC 80/10 Boards. Appendix B is a procedure for modifying the ASR-33 Teletypewriter.
Chapter 6

Interfacing to Multibus Masters

The SBC 80/10's system bus structure permits interfacing to one other Multibus-Compatible Master module. This interface is accomplished using the serial priority scheme as shown in figure 6-1 using the Intel SBC 604 Cardcage/Backplane. The SBC 80/10 does not provide the Bus Priority Request Out (BPRO/) signal and therefore, the SBC 80/10 can only be used with one other Multibus master. For these configurations, the SBC 80/10 must always have lower priority than the other Multibus master and a wire must be added from the master's BREQ/ pin (pin 18) to the SBC 80/10 BPRN pin (pin 15). In the configuration shown in figure 6-1 the SBC 80/10 acquires control of the Multibus whenever BREQ/ generated by the Diskette Controller is in the high state. This occurs whenever the Diskette Controller is not using the Multibus. Similarly BREQ/ is driven to the low state when the Diskette Controller acquires control of the Multibus disabling the SBC 80/10 from accessing the Multibus.

For a detailed description of Multibus interfacing refer to the Intel Multibus Interfacing Application Note (AP-28).
Figure 6-1. Serial Priority Configuration with another Multibus Master
CHAPTER 7

SPECIFICATIONS

7.1 DC POWER REQUIREMENTS

DC Power Requirements are given in Table 7-1.

7.2 AC CHARACTERISTICS

Detailed timing diagrams for memory, I/O and Bus exchange operations are provided in Figures 7-1 through 7-3. Tables 7-3 and 7-4 provide design limits for SBC-80/10 outputs and requirements for its inputs. These values are theoretical limits based on a "worst-on-worst" case analysis using vendor information and approximations where necessary. Approximations include establishing non-zero propagation delay minimums and extended delays if capacitive loading exceeds vendor ratings. In all such cases, approximations are conservative (e.g., 2 ns minimum for standard TTL, 4 ns minimum for tri-state turn-offs or turn-ons). Rise and fall times are assumed to be zero unless a three-state high impedance state or open collector circuit is involved.

7.3 DC CHARACTERISTICS

DC Characteristics are given in Table 7-4.

7.4 ENVIRONMENT

Temperature extremes can cause instability, or result in permanent damage to the circuits on the module. Ambient temperature must, therefore, be maintained within the limits of 0°C to 55°C. Exercise caution in locating the module, giving particular attention to radiant and conducive sources of heat. Remember that the module itself, when installed,
will contribute some heat to the environment. Maintain an adequate clearance to permit the convective dissipation of heat from the elements on the card.

Relative humidity should not exceed 90%, non-condensing.

7.5 BOARD OUTLINE

See Figure 7-4.

7.6 COMPATIBLE CONNECTORS

Table 7-5 lists compatible connectors which mate to the SBC-80/10 and SBC 80/10A PC edge connectors.

<table>
<thead>
<tr>
<th>TABLE 7-1. DC POWER REQUIREMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without</td>
</tr>
<tr>
<td>( V_{CC} )</td>
</tr>
<tr>
<td>+5V ± 5%</td>
</tr>
<tr>
<td>( V_{DD} )</td>
</tr>
<tr>
<td>+12V ± 5%</td>
</tr>
<tr>
<td>( V_{BB} )</td>
</tr>
<tr>
<td>-5V ± 5%</td>
</tr>
<tr>
<td>-12V ± 5%</td>
</tr>
</tbody>
</table>

1. Does not include power required for optional ROM/EPROM, I/O drivers or I/O terminators.
2. With four Intel 8708 EPROMs and 220Ω/330Ω terminators installed for 48 input lines; all terminator inputs low.
3. With four Intel 2758 or 2716 EPROMs and 220Ω/330Ω terminators installed for 48 input ports; all terminator inputs low.
4. Required for RS232C drivers.
TABLE 7-2. AC CHARACTERISTICS WITH BUS EXCHANGE

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>OVERALL</th>
<th>WITH BUS EXCHANGE</th>
<th>DESCRIPTION</th>
<th>REMARKS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MIN (ns)</td>
<td>MAX (ns)</td>
<td>MIN (ns)</td>
<td>MAX (ns)</td>
</tr>
<tr>
<td>t_AS</td>
<td>82</td>
<td>82</td>
<td>658</td>
<td></td>
</tr>
<tr>
<td>t_AH</td>
<td>61</td>
<td>0</td>
<td>61</td>
<td></td>
</tr>
<tr>
<td>t_DS</td>
<td>140</td>
<td>-</td>
<td>140</td>
<td></td>
</tr>
<tr>
<td>t_DH</td>
<td>61</td>
<td>0</td>
<td>61</td>
<td></td>
</tr>
<tr>
<td>t_ACK0</td>
<td>68</td>
<td>191</td>
<td></td>
<td></td>
</tr>
<tr>
<td>t_ACK1</td>
<td>551</td>
<td>684</td>
<td>-60</td>
<td>132</td>
</tr>
<tr>
<td>t_ACK2</td>
<td>1034</td>
<td>1174</td>
<td>423</td>
<td>625</td>
</tr>
<tr>
<td>t_CY</td>
<td>483</td>
<td>493</td>
<td></td>
<td></td>
</tr>
<tr>
<td>t_WC</td>
<td>596</td>
<td>796</td>
<td>1412</td>
<td>1516</td>
</tr>
<tr>
<td>t_ACC</td>
<td>344</td>
<td>344</td>
<td>344</td>
<td>344</td>
</tr>
<tr>
<td>t_SKD</td>
<td>68</td>
<td>68</td>
<td>-60</td>
<td>-60</td>
</tr>
<tr>
<td>t_BKO</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>t_XKD</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>t_XKO</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>t_DBS</td>
<td>3500</td>
<td>3500</td>
<td>3500</td>
<td>3500</td>
</tr>
<tr>
<td>t_BSC</td>
<td>0</td>
<td>493</td>
<td></td>
<td></td>
</tr>
<tr>
<td>t_DBY</td>
<td>358</td>
<td>700</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Memory and I/O access occurs with no wait states.
<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>DESCRIPTION</th>
<th>REMARKS</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_{AS} )</td>
<td>Address Setup Time to Command</td>
<td>Generates 0 Wait States</td>
</tr>
<tr>
<td>( t_{AH} )</td>
<td>Address Hold Time</td>
<td>Generates 1 Wait State</td>
</tr>
<tr>
<td>( t_{DS} )</td>
<td>Data Setup Time to Command</td>
<td>Generates 2 Wait States</td>
</tr>
<tr>
<td>( t_{DH} )</td>
<td>Data Hold Time</td>
<td></td>
</tr>
<tr>
<td>( t_{ACK0} )</td>
<td>First ACK Sampling Point of Current Cycle</td>
<td></td>
</tr>
<tr>
<td>( t_{ACK1} )</td>
<td>Second ACK Sampling Point of Current Cycle</td>
<td></td>
</tr>
<tr>
<td>( t_{ACK2} )</td>
<td>Third ACK Sampling Point of Current Cycle</td>
<td></td>
</tr>
<tr>
<td>( t_{CY} )</td>
<td>ACK &amp; BPRN Sample Cycle Time</td>
<td></td>
</tr>
<tr>
<td>( t_{SEP} )</td>
<td>Command Separation</td>
<td>Read, 0 Wait States</td>
</tr>
<tr>
<td>( t_{WC} )</td>
<td>Command Width</td>
<td>Write, 2 Wait States</td>
</tr>
<tr>
<td>( t_{ACC} )</td>
<td>Read Access Time</td>
<td>▲</td>
</tr>
<tr>
<td>( t_{8KD} )</td>
<td>Advanced ACK Response Time for Minimum Delay</td>
<td>▲</td>
</tr>
<tr>
<td>( t_{8KO} )</td>
<td>Advanced ACK Turn Off Delay</td>
<td></td>
</tr>
<tr>
<td>( t_{XKD} )</td>
<td>XACK Delay From Valid Data or Write</td>
<td></td>
</tr>
<tr>
<td>( t_{XKO} )</td>
<td>XACK Turn Off Delay</td>
<td></td>
</tr>
<tr>
<td>( t_{BCY} )</td>
<td>Bus Clock Cycle Time</td>
<td>80/10 Generator</td>
</tr>
<tr>
<td>( t_{BW} )</td>
<td>Bus Clock Low or High Periods</td>
<td>80/10 Generator</td>
</tr>
<tr>
<td>( t_{INT} )</td>
<td>Initialization Width</td>
<td>After all voltages have stabilized</td>
</tr>
</tbody>
</table>

▲ MAX assumes no acknowledge delays.

▲ Write Command to next Read Command separation.
TABLE 7-4. DC CHARACTERISTICS

<table>
<thead>
<tr>
<th>SIGNALS</th>
<th>SYMBOL</th>
<th>PARAMETER DESCRIPTION</th>
<th>TEST CONDITIONS</th>
<th>MIN</th>
<th>MAX</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADR0/-ADR1/</td>
<td>V&lt;sub&gt;OL&lt;/sub&gt;</td>
<td>Output Low Voltage</td>
<td>I&lt;sub&gt;OL&lt;/sub&gt; = 50 mA</td>
<td>0.6</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td>V&lt;sub&gt;OH&lt;/sub&gt;</td>
<td>Output High Voltage</td>
<td>I&lt;sub&gt;OH&lt;/sub&gt; = -10 mA</td>
<td>0.95</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td>V&lt;sub&gt;IL&lt;/sub&gt;</td>
<td>Input Low Voltage</td>
<td>I&lt;sub&gt;IL&lt;/sub&gt;</td>
<td>2.0</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td>V&lt;sub&gt;IH&lt;/sub&gt;</td>
<td>Input High Voltage</td>
<td>I&lt;sub&gt;IH&lt;/sub&gt;</td>
<td>2.0</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td>I&lt;sub&gt;IL&lt;/sub&gt;</td>
<td>Input Current at Low V</td>
<td>V&lt;sub&gt;IN&lt;/sub&gt;</td>
<td>0.45</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>I&lt;sub&gt;IH&lt;/sub&gt;</td>
<td>Input Current at High V</td>
<td>V&lt;sub&gt;IN&lt;/sub&gt;</td>
<td>5.25V</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>*&lt;sub&gt;CL&lt;/sub&gt;</td>
<td>Capacitive Load</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MROC/-MTC/</td>
<td>V&lt;sub&gt;OL&lt;/sub&gt;</td>
<td>Output Low Voltage</td>
<td>I&lt;sub&gt;OL&lt;/sub&gt; = 32 mA</td>
<td>0.4</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td>V&lt;sub&gt;OH&lt;/sub&gt;</td>
<td>Output High Voltage</td>
<td>I&lt;sub&gt;OH&lt;/sub&gt; = -5.2 mA</td>
<td>2.4</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td>I&lt;sub&gt;IL&lt;/sub&gt;</td>
<td>Output Leakage High</td>
<td>V&lt;sub&gt;O&lt;/sub&gt;</td>
<td>2.4</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>I&lt;sub&gt;IH&lt;/sub&gt;</td>
<td>Output Leakage Low</td>
<td>V&lt;sub&gt;O&lt;/sub&gt;</td>
<td>0.4</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>*&lt;sub&gt;CL&lt;/sub&gt;</td>
<td>Capacitive Load</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DAT0/-DAT7/</td>
<td>V&lt;sub&gt;OL&lt;/sub&gt;</td>
<td>Output Low Voltage</td>
<td>I&lt;sub&gt;OL&lt;/sub&gt; = 50 mA</td>
<td>0.6</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td>V&lt;sub&gt;OH&lt;/sub&gt;</td>
<td>Output High Voltage</td>
<td>I&lt;sub&gt;OH&lt;/sub&gt; = -10 mA</td>
<td>0.95</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td>V&lt;sub&gt;IL&lt;/sub&gt;</td>
<td>Input Low Voltage</td>
<td>I&lt;sub&gt;IL&lt;/sub&gt;</td>
<td>2.0</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td>V&lt;sub&gt;IH&lt;/sub&gt;</td>
<td>Input High Voltage</td>
<td>I&lt;sub&gt;IH&lt;/sub&gt;</td>
<td>2.0</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td>I&lt;sub&gt;IL&lt;/sub&gt;</td>
<td>Input Current at Low V</td>
<td>V&lt;sub&gt;IN&lt;/sub&gt;</td>
<td>0.45</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>I&lt;sub&gt;IH&lt;/sub&gt;</td>
<td>Input Current at High V</td>
<td>V&lt;sub&gt;IN&lt;/sub&gt;</td>
<td>5.25V</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>*&lt;sub&gt;CL&lt;/sub&gt;</td>
<td>Capacitive Load</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INT1/</td>
<td>V&lt;sub&gt;IL&lt;/sub&gt;</td>
<td>Input Low Voltage</td>
<td>I&lt;sub&gt;IL&lt;/sub&gt;</td>
<td>2.0</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td>V&lt;sub&gt; IH&lt;/sub&gt;</td>
<td>Input High Voltage</td>
<td>I&lt;sub&gt;IH&lt;/sub&gt;</td>
<td>2.0</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td>I&lt;sub&gt;IL&lt;/sub&gt;</td>
<td>Input Current at Low V</td>
<td>V&lt;sub&gt;IN&lt;/sub&gt;</td>
<td>0.4V</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>I&lt;sub&gt;IH&lt;/sub&gt;</td>
<td>Input Current at High V</td>
<td>V&lt;sub&gt;IN&lt;/sub&gt;</td>
<td>5.5V</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>*&lt;sub&gt;CL&lt;/sub&gt;</td>
<td>Capacitive Load</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BP0N/XACK ACK</td>
<td>V&lt;sub&gt;IL&lt;/sub&gt;</td>
<td>Input Low Voltage</td>
<td>I&lt;sub&gt;IL&lt;/sub&gt;</td>
<td>2.0</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td>V&lt;sub&gt; IH&lt;/sub&gt;</td>
<td>Input High Voltage</td>
<td>I&lt;sub&gt;IH&lt;/sub&gt;</td>
<td>2.0</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td>I&lt;sub&gt;IL&lt;/sub&gt;</td>
<td>Input Current at Low V</td>
<td>V&lt;sub&gt;IN&lt;/sub&gt;</td>
<td>0.5</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>I&lt;sub&gt;IH&lt;/sub&gt;</td>
<td>Input Current at High V</td>
<td>V&lt;sub&gt;IN&lt;/sub&gt;</td>
<td>2.7V</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>*&lt;sub&gt;CL&lt;/sub&gt;</td>
<td>Capacitive Load</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BUSY/ OPEN COLLECTOR</td>
<td>V&lt;sub&gt;OL&lt;/sub&gt;</td>
<td>Output Low Voltage</td>
<td>I&lt;sub&gt;OL&lt;/sub&gt; = 25 mA</td>
<td>0.4</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td>*&lt;sub&gt;CL&lt;/sub&gt;</td>
<td>Capacitive Load</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INT (SYSTEM RESET)</td>
<td>V&lt;sub&gt;OL&lt;/sub&gt;</td>
<td>Output Low Voltage</td>
<td>I&lt;sub&gt;OL&lt;/sub&gt; = 32 mA</td>
<td>0.6</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td>V&lt;sub&gt;OH&lt;/sub&gt;</td>
<td>Output High Voltage</td>
<td>I&lt;sub&gt;OH&lt;/sub&gt; = -1 mA</td>
<td>0.7</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td>V&lt;sub&gt;IL&lt;/sub&gt;</td>
<td>Input Low Voltage</td>
<td>I&lt;sub&gt;IL&lt;/sub&gt;</td>
<td>2.0</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td>V&lt;sub&gt;IH&lt;/sub&gt;</td>
<td>Input High Voltage</td>
<td>I&lt;sub&gt;IH&lt;/sub&gt;</td>
<td>2.0</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td>I&lt;sub&gt;IL&lt;/sub&gt;</td>
<td>Input Current at High V</td>
<td>V&lt;sub&gt;IN&lt;/sub&gt;</td>
<td>5.5</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>I&lt;sub&gt;IH&lt;/sub&gt;</td>
<td>Input Current at Low V</td>
<td>V&lt;sub&gt;IN&lt;/sub&gt;</td>
<td>0.3</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>*&lt;sub&gt;CL&lt;/sub&gt;</td>
<td>Capacitive Load</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BCLK/ + CCLK/</td>
<td>V&lt;sub&gt;OL&lt;/sub&gt;</td>
<td>Output Low Voltage</td>
<td>I&lt;sub&gt;OL&lt;/sub&gt; = 20 mA</td>
<td>0.5</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td>V&lt;sub&gt;OH&lt;/sub&gt;</td>
<td>Output High Voltage</td>
<td>I&lt;sub&gt;OH&lt;/sub&gt; = -1 mA</td>
<td>2.7</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td>*&lt;sub&gt;CL&lt;/sub&gt;</td>
<td>Capacitive Load</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Capacitive values are approximations only.
**TABLE 7-4. DC CHARACTERISTICS (Continued)**

<table>
<thead>
<tr>
<th>SIGNALS</th>
<th>SYMBOL</th>
<th>PARAMETER DESCRIPTION</th>
<th>TEST CONDITIONS</th>
<th>MIN</th>
<th>MAX</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EXT INTRØ/</strong></td>
<td></td>
<td>Input Low Voltage</td>
<td></td>
<td>2.0</td>
<td>0.8</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Input High Voltage</td>
<td></td>
<td>6.8</td>
<td>V</td>
<td>mA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Input Current at Low V</td>
<td>(V_{IN} = 0.4)</td>
<td>2</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Input Current at High V</td>
<td>(V_{IN} = 5.5)</td>
<td>18</td>
<td>pF</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Capacitive Load</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>PORT E4 BIDIRECTIONAL DRIVERS</strong></td>
<td></td>
<td>Output Low Voltage</td>
<td>(I_{OL} = 20)</td>
<td>.45</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Output High Voltage</td>
<td>(I_{OH} = -12)</td>
<td>.95</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Input Low Voltage</td>
<td></td>
<td>2.0</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Input High Voltage</td>
<td></td>
<td>5.25</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Input Current at Low V</td>
<td>(V_{IN} = 0.45)</td>
<td>.30</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Input Current at High V</td>
<td>(V_{O} = 5.25)</td>
<td>18</td>
<td>pF</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Capacitive Load</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>8255 DRIVER/RECEIVER</strong></td>
<td></td>
<td>Output Low Voltage</td>
<td>(I_{OL} = 1.7)</td>
<td>.45</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Output High Voltage</td>
<td>(I_{OH} = -50)</td>
<td>.8</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Input Low Voltage</td>
<td></td>
<td>2.0</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Input High Voltage</td>
<td></td>
<td>10</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Input Current at Low V</td>
<td>(V_{IN} = 0.45)</td>
<td>10</td>
<td>mA</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Input Current at High V</td>
<td>(V_{IN} = 5.0)</td>
<td>18</td>
<td>pF</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Capacitive Load</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Capacitive values are approximations only.*
### TABLE 7-5. COMPATIBLE CONNECTOR HARDWARE

<table>
<thead>
<tr>
<th>FUNCTION</th>
<th># OF PINS</th>
<th>CENTERS (inches)</th>
<th>CONNECTOR TYPE</th>
<th>VENDOR</th>
<th>VENDOR PART #</th>
<th>INTEL PART #</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARALLEL I/O</td>
<td>25/50</td>
<td>0.1</td>
<td>FLAT CRIMP</td>
<td>3M</td>
<td>3415-0000 WITH EARS</td>
<td>SBC-955 (CABLE ASSY.)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3M</td>
<td>3415-0001 W/O EARS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AMP</td>
<td>88083-1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ANSLEY</td>
<td>609-5015</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SAE</td>
<td>SD6750 SERIES</td>
<td></td>
</tr>
<tr>
<td>SERIAL I/O</td>
<td>13/26</td>
<td>0.1</td>
<td>FLAT CRIMP</td>
<td>3M</td>
<td>3462-0001 CRIMP</td>
<td>SBC-956 (CABLE ASSY.)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AMP</td>
<td>88106-1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ANSLEY</td>
<td>609-2615</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SAE</td>
<td>SD6726 SERIES</td>
<td></td>
</tr>
<tr>
<td>PARALLEL I/O</td>
<td>25/50</td>
<td>0.1</td>
<td>SOLDEROED</td>
<td>AMP</td>
<td>2-583485-6</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>VIKING</td>
<td>3VH25/1JVS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TI</td>
<td>H312125</td>
<td></td>
</tr>
<tr>
<td>SERIAL I/O</td>
<td>13/26</td>
<td>0.1</td>
<td>SOLDEROED</td>
<td>AMP</td>
<td>1-583485-5</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>VIKING</td>
<td>H312113</td>
<td></td>
</tr>
<tr>
<td>AUXILIARY</td>
<td>30/60</td>
<td>0.1</td>
<td>SOLDEROED</td>
<td>CDC</td>
<td>3VH30/1JN5</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MICRO PLASTICS</td>
<td>VFB01E43DOOA1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ARCO</td>
<td>MP-0156-43-BW-4</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>VIKING</td>
<td>AE443361 LESS EARS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TI</td>
<td>2VH43/1JV5</td>
<td></td>
</tr>
<tr>
<td>BUS</td>
<td>43/86</td>
<td>0.156</td>
<td>WIREWRAP</td>
<td>TI</td>
<td>3VH11125</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>VIKING</td>
<td>3VH25/1JNDS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CDC</td>
<td>VFB01B25DOOA1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>VIKING</td>
<td>E04050OA1</td>
<td></td>
</tr>
<tr>
<td>PARALLEL I/O</td>
<td>25/50</td>
<td>0.1</td>
<td>WIREWRAP</td>
<td>TI</td>
<td>H311125</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>VIKING</td>
<td>3VH25/1JNDS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CDC</td>
<td>VFB01B25D0OA1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>VIKING</td>
<td>E04050OA1</td>
<td></td>
</tr>
<tr>
<td>SERIAL I/O</td>
<td>13/26</td>
<td>0.1</td>
<td>WIREWRAP</td>
<td>TI</td>
<td>H311113</td>
<td>N/A</td>
</tr>
<tr>
<td>AUXILIARY</td>
<td>30/60</td>
<td>0.1</td>
<td>WIREWRAP</td>
<td>CDC</td>
<td>VFB01B30A00A2</td>
<td>MDS-980</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TI</td>
<td>H311110</td>
<td></td>
</tr>
<tr>
<td>BUS</td>
<td>43/86</td>
<td>0.156</td>
<td>WIREWRAP</td>
<td>CDC</td>
<td>VFB01E43D00A1</td>
<td>MDS-985</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>VIKING</td>
<td>2VH43/1JNDS</td>
<td></td>
</tr>
<tr>
<td>SBC 201</td>
<td>50/100</td>
<td>0.1</td>
<td>SOLDER TAIL</td>
<td>VIKING</td>
<td>3VH50/1JN5</td>
<td>MDS-990</td>
</tr>
<tr>
<td>SBC 501</td>
<td></td>
<td></td>
<td></td>
<td>VIKING</td>
<td>MDS-980</td>
<td></td>
</tr>
<tr>
<td>SBC 508</td>
<td></td>
<td></td>
<td></td>
<td>VIKING</td>
<td>MDS-990</td>
<td></td>
</tr>
<tr>
<td>SBC 905, etc.</td>
<td></td>
<td></td>
<td></td>
<td>VIKING</td>
<td>MDS-990</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>VATIC</td>
<td>VFB04B50B00A1E</td>
<td>N/A</td>
</tr>
</tbody>
</table>

- Connector heights are not guaranteed to conform to OEM packaging equipment. Intel OEM and Intellect Development System motherboards offer complete mechanical compatibility.
- Wirewrap pin lengths are not guaranteed to conform to OEM packaging equipment. Intel connectors and OEM and Intellect Development System motherboards offer complete mechanical compatibility.
- CDC VFB01 ..., VFB02 ..., VFB04 ..., etc. are identical connectors with different electroplating thicknesses or metal surfaces.

**NOTE:** See next page for vendor addresses, telephone numbers and TWX numbers.
VENDORS ADDRESSES

The following information is for our customers' convenience only. Intel does not represent these vendors, guarantee availability nor continued quality of their products.

**CDC CONNECTOR DIVISION**
31829 W. LaTienda Drive
Westlake Village, CA 91361
213-889-3535
TWX 910-494-1224

**T & B/ANSLEY**
Subsidiary of Thomas & Betts Corporation
3208 Humbolt Street
Los Angeles, CA 90031
213-223-2331
TWX 910-321-3938

**VIKING INDUSTRIES, INC.**
21001 Nordhoff Street
Chatsworth, CA 91311
213-341-4330
TWX 910-494-2094

**STANFORD APPLIED ENGINEERING, INC. (SAE)**
340 Martin Avenue
Santa Clara, CA 95050
408-243-9200
TWX 910-338-0132

**Connector Systems**
**TEXAS INSTRUMENTS, INC.**
34 Forest Street
Attleboro, MA 02703
617-222-2800

**3M CONNECTORS**
Electronic Products Division, Bldg. 223-4E
3M COMPANY
3M Center
St. Paul, MN 55101
612-733-1110

**AMP Incorporated**
P.O. Box 3608
Harrisburg, PA 17105
717-564-0100
TWX 510-657-4110
*FIGURE 7-1. MEMORY AND I/O READ TIMING (CONTINUOUS BUS CONTROL)

*NOT DRAWN TO SCALE.
*FIGURE 7-2. MEMORY AND I/O WRITE TIMING (CONTINUOUS BUS CONTROL)

*NOT DRAWN TO SCALE.
*FIGURE 7-3. BUS EXCHANGE (WRITE)

*NOT DRAWN TO SCALE.
Figure 7-4. SBC 80/10 and SBC 80/10A Dimension
Drawing - Page 1 of 2
Figure 7-4. SBC 80/10 and SBC 80/10A Dimension Drawing - Page 2 of 2

NOTES:

1. MATERIAL: .062 INK, 1 OZ COPPER CLAD, NATURAL EPOXY GLASS; TYPE 3-G (RE: AFTER PLATING THRU).
2. BOARD EDGES ARE LOCATED FROM INDEX HOLES. INDEX HOLES ARE ON .050 GRID INTERSECTION AND ARE USED FOR ARTWORK REGISTRATION AND MAY BE USED AS TOOLING HOLES, PLATING OPTIONAL.
3. HOLES ARE PLATED THRU WITH COPPER WALL THICKNESS OF .0007 MINIMUM.
4. HOLE SIZES SPECIFIED ARE AFTER PLATING; .003 TOLERANCE
5. CONTACT FINGERS ARE OVERPLATED WITH A MINIMUM OF .0003 MILLIONTHS GOLD OVER NICKEL TO DIMENSION SHOWN.
6. APPLY SOLDER MASK OVER SOLDER PLATE USING MATERIAL: MEGAMASK GREEN
7. DRILL FROM CIRCUIT SIDE.
8. TRACE WIDTHS MUST BE WITHIN .005 OF ARTWORK NEGATIVES.
9. APPLY SILKSCREEN ON COMPONENT SIDE AFTER SOLDER MASK IS APPLIED, USING WHITE EPOXY INK.

Copyright ©1976 Intel Corporation
Schematic drawings for the SBC-80/80 and SBC 80/10A are provided in this appendix. Information and diagrams in this section are subject to change without notice. References should be made to schematics shipped with this module.
FIGURE A-2. SBC 80/10A SCHEMATIC (SHEET 4 OF 5)
FIGURE A-2. SBC 80/10A SCHEMATIC (SHEET 5 OF 5)

A-11
FIGURE A-1. SBC 80/10 SCHEMATIC (SHEET 3 OF 5)
FIGURE A-2. SBC 80/10A SCHEMATIC (SHEET 3 OF 5)
APPENDIX B
TELETYPETRITER MODIFICATIONS

B-1. INTRODUCTION

This appendix provides information required to modify a Model ASR-33 Teletypewriter for use with certain Intel SBC 80 computer systems.

B-2. INTERNAL MODIFICATIONS

**WARNING**

Hazardous voltages are exposed when the top cover of the teletypewriter is removed. To prevent accidental shock, disconnect the teletypewriter power cord before proceeding beyond this point.

Remove the top cover and modify the teletypewriter as follows:

a. Remove blue lead from 750-ohm tap on current source register; reconnect this lead to 1450-ohm tap. (Refer to figures B-1 and B-2.)

b. On terminal block, change two wires as follows to create an internal full-duplex loop (refer to figures B-1 and B-3):

   1. Remove brown/yellow lead from terminal 3; reconnect this lead to terminal 5.
   2. Remove white/blue lead from terminal 4; reconnect this lead to terminal 5.

c. On terminal block, remove violet lead from terminal 8; reconnect this lead to terminal 9. This changes the receiver current level from 60 mA to 20 mA.

A relay circuit card must be fabricated and connected to the paper tape reader drive circuit. The relay circuit card to be fabricated requires a relay, a diode, a thyractor, a small ‘vector’ board for mounting the components, and suitable hardware for mounting the assembled relay card.

A circuit diagram of the relay circuit card is included in figure B-4; this diagram also includes the part numbers of the relay, diode, and thyractor. (Note that a 470-ohm resistor and a 0.1 µF capacitor may be substituted for the thyractor.) After the relay circuit card has been assembled, mount it in position as shown in figure B-5. Secure the card to the base plate using two self-tapping screws. Connect the relay circuit to the distributor trip magnet and mode switch as follows:

a. Refer to figure B-4 and connect a wire (Wire ‘A’) from relay circuit card to terminal L2 on mode switch. (See figure B-6.)

b. Disconnect brown wire shown in figure B-7 from plastic connector. Connect this brown wire to terminal L2 on mode switch. (Brown wire will have to be extended.)

c. Refer to figure B-4 and connect a wire (Wire ‘B’) from relay circuit board to terminal L1 on mode switch.

B-3. EXTERNAL CONNECTIONS

Connect a two-wire receive loop, a two-wire send loop, and a two-wire tape reader control loop to the external device as shown in figure B-4. The external connector pin numbers shown in figure B-4 are for interface with an RS232C device.

B-4. SBC 530 TTY ADAPTER

The SBC 530, which converts RS232C signal levels to an optically isolated 20 mA current loop interface, provides signal translation for transmitted data, received data, and a paper tape reader relay. The SBC 530 interfaces an Intel SBC 80 computer system to a teletypewriter as shown in figure B-8.

The SBC 530 requires +12V at 98 mA and −12V at 98 mA. An auxiliary supply must be used if the SBC 80 system does not supply this power. A schematic diagram of the SBC 530 is supplied with the unit. The following auxiliary power connector (or equivalent) must be procured by the user:

- Connector, Molex 09-50-7071
- Pins, Molex 08-50-0106
- Polarizing Key, Molex 15-04-0219
Teletypewriter Modifications

Figure B-1. Teletype Component Layout

Figure B-2. Current Source Resistor

Figure B-3. Terminal Block
Figure B-4. Teletypewriter Modifications

Figure B-5. Relay Circuit

Figure B-6. Mode Switch
Teletypewriter Modifications

Figure B-7. Distributor Trip Magnet

Figure B-8. TTY Adapter Cabling
APPENDIX C

8080 INSTRUCTION SET SUMMARY

A computer, no matter how sophisticated, can only do what it is “told” to do. One “tells” the computer what to do via a series of coded instructions referred to as a Program. The realm of the programmer is referred to as Software, in contrast to the Hardware that comprises the actual computer equipment. A computer’s software refers to all of the programs that have been written for that computer.

When a computer is designed, the engineers provide the Central Processing Unit (CPU) with the ability to perform a particular set of operations. The CPU is designed such that a specific operation is performed when the CPU control logic decodes a particular instruction. Consequently, the operations that can be performed by a CPU define the computer’s Instruction Set.

Each computer instruction allows the programmer to initiate the performance of a specific operation. All computers implement certain arithmetic operations in their instruction set, such as an instruction to add the contents of two registers. Often logical operations (e.g., OR the contents of two registers) and register operate instructions (e.g., increment a register) are included in the instruction set. A computer’s instruction set will also have instructions that move data between registers, between a register and memory, and between register and an I/O device. Most instruction sets also provide Conditional Instructions. A conditional instruction specifies an operation to be performed only if certain conditions have been met; for example, jump to a particular instruction if the result of the last operation was zero. Conditional instructions provide a program with a decision-making capability.

By logically organizing a sequence of instructions into a coherent program, the programmer can “tell” the computer to perform a very specific and useful function.

The computer, however, can only execute programs whose instructions are in a binary coded form (i.e., a series of 1’s and 0’s), that is called Machine Code. Because it would be extremely cumbersome to program in machine code, programming languages have been developed. There are programs available which convert the programming language instructions into machine code that can be interpreted by the processor.

One type of programming language is Assembly Language. A unique assembly language mnemonic is assigned to each of the computer’s instructions. The programmer can write a program (called the Source Program) using these mnemonics and certain operands; the source program is then converted into machine instructions (called the Object Code). Each assembly language instruction is converted into one machine code instruction (1 or more bytes) by an Assembler program. Assembly languages are usually machine dependent (i.e., they are usually able to run on only one type of computer).

THE 8080 INSTRUCTION SET

The 8080 instruction set includes five different types of instructions:

- **Data Transfer Group** – move data between registers or between memory and registers.
- **Arithmetic Group** – add, subtract, increment or decrement data in registers or in memory.
- **Logical Group** – AND, OR, EXCLUSIVE-OR, compare, rotate or complement data in registers or in memory.
- **Branch Group** – conditional and unconditional jump instructions, subroutine call instructions and return instructions.
- **Stack, I/O and Machine Control Group** – includes I/O instructions, as well as instructions for maintaining the stack and internal control flags.
Instruction and Data Formats

Memory for the 8080 is organized into 8-bit quantities, called Bytes. Each byte has a unique 16-bit binary address corresponding to its sequential position in memory.

The 8080 can directly address up to 65,536 bytes of memory, which may consist of both read-only memory (ROM) elements and random-access memory (RAM) elements (read/write memory).

Data in the 8080 is stored in the form of 8-bit binary integers:

```
| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
```

When a register or data word contains a binary number, it is necessary to establish the order in which the bits of the number are written. In the Intel 8080, BIT 0 is referred to as the Least Significant Bit (LSB), and BIT 7 (of an 8-bit number) is referred to as the Most Significant Bit (MSB).

The 8080 program instructions may be one, two or three bytes in length. Multiple byte instructions must be stored in successive memory locations; the address of the first byte is always used as the address of the instructions. The exact instruction format will depend on the particular operation to be executed.

Addressing Modes

Often the data that is to be operated on is stored in memory. When multi-byte numeric data is used, the data, like instructions, is stored in successive memory locations, with the least significant byte first, followed by increasingly significant bytes. The 8080 has four different modes for addressing data stored in memory or in registers:

- **Direct** – Bytes 2 and 3 of the instruction contain the exact memory address of the data item (the low-order bits of the address are in byte 2, the high-order bits in byte 3).

- **Register** – The instruction specifies the register-pair in which the data is located.

- **Register Indirect** – The instruction specifies a register-pair which contains the memory address where the data is located (the high-order bits of the address are in the first register of the pair, the low-order bits in the second).

- **Immediate** – The instruction contains the data itself. This is either an 8-bit quantity or a 16-bit quantity (least significant byte first, most significant byte second).

Unless directed by an interrupt or branch instruction, the execution of instructions proceeds through consecutively increasing memory locations. A branch instruction can specify the address of the next instruction to be executed in one of two ways:

- **Direct** – The branch instruction contains the address of the next instruction to be executed. (Except for the ‘RST’ instruction, byte 2 contains the low-order address and byte 3 the high-order address.)

- **Register Indirect** – The branch instruction indicates a register-pair which contains the address of the next instruction to be executed. (The high-order bits of the address are in the first register of the pair, the low-order bits in the second.)

The RST instruction is a special one-byte call instruction (usually used during interrupt sequences).
RST includes a three-bit field; program control is transferred to the instruction whose address is eight times the contents of this three-bit field.

Condition Flags

There are five condition flags associated with the execution of instructions on the 8080. They are Zero, Sign, Parity, Carry, and Auxiliary Carry, and are each represented by a 1-bit register in the CPU. A flag is “set” by forcing the bit to 1; “reset” by forcing the bit to 0.

Unless indicated otherwise, when an instruction affects a flag, it affects it in the following manner:

Zero: If the result of an instruction has the value 0, this flag is set; otherwise it is reset.

Sign: If the most significant bit of the result of the operation has the value 1, this flag is set; otherwise it is reset.

Parity: If the modulo 2 sum of the bits of the result of the operation is 0 (i.e., if the result has even parity), this flag is set; otherwise it is reset (i.e., if the result has odd parity).

Carry: If the instruction resulted in a carry (from addition), or a borrow (from subtraction of a comparison) out of the high-order bit, this flag is set; otherwise it is reset.

Auxiliary Carry: If the instruction caused a carry out of bit 3 and into bit 4 of the resulting value, the auxiliary carry is set; otherwise it is reset. This flag is affected by single precision additions, subtractions, increments, decrements, comparisons, and logical operations, but is principally used with additions and increments preceding a DAA (Decimal Adjust Accumulator) instruction.

Symbols and Abbreviations

The following symbols and abbreviations are used in the subsequent description of the 8080 instructions:

<table>
<thead>
<tr>
<th>SYMBOLS</th>
<th>MEANING</th>
</tr>
</thead>
<tbody>
<tr>
<td>accumulator</td>
<td>Register A</td>
</tr>
<tr>
<td>addr</td>
<td>16-bit address quantity</td>
</tr>
<tr>
<td>data</td>
<td>8-bit data quantity</td>
</tr>
<tr>
<td>data 16</td>
<td>16-bit data quantity</td>
</tr>
<tr>
<td>byte 2</td>
<td>The second byte of the instruction</td>
</tr>
<tr>
<td>byte 3</td>
<td>The third byte of the instruction</td>
</tr>
<tr>
<td>port</td>
<td>8-bit address of an I/O device</td>
</tr>
<tr>
<td>r,r 1,r2</td>
<td>One of the registers A,B,C,D,E,H,L</td>
</tr>
<tr>
<td>DDD,SSS</td>
<td>The bit pattern designating one of the registers A,B,C,D,E,H,L. (DD=destination, SSS=source):</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DDD or SSS</th>
<th>REGISTER NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>111</td>
<td>A</td>
</tr>
<tr>
<td>000</td>
<td>B</td>
</tr>
<tr>
<td>001</td>
<td>C</td>
</tr>
<tr>
<td>010</td>
<td>D</td>
</tr>
<tr>
<td>011</td>
<td>E</td>
</tr>
<tr>
<td>100</td>
<td>H</td>
</tr>
<tr>
<td>101</td>
<td>L</td>
</tr>
</tbody>
</table>

The bit pattern designating one of the register pairs B,D,H,SP:

<table>
<thead>
<tr>
<th>RP</th>
<th>REGISTER PAIR</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>B-C</td>
</tr>
<tr>
<td>01</td>
<td>D-E</td>
</tr>
<tr>
<td>10</td>
<td>H-L</td>
</tr>
<tr>
<td>11</td>
<td>SP</td>
</tr>
</tbody>
</table>

The first (high-order) register of a designated pair.

The second (low-order) register of a designated register pair.
16-bit program counter register (PCH and PCL are used to refer to the high-order and low-order 8 bits, respectively).

16-bit stack pointer register (SPH and SPL are used to refer to the high-order and low-order 8 bits, respectively).

Bit m of the register r (bits are number 7 through 0 from left to right).

The condition flags:
- Zero,
- Sign,
- Parity,
- Carry, and
- Auxiliary Carry, respectively.

The contents of the memory location or registers enclosed in the parentheses.

"Is transferred to" A

Logical AND

Exclusive OR

Inclusive OR

Addition

Two's complement subtraction

Multiplication

"Is exchanged with"

The one's complement (e.g., (A))

The restart number 0 through 7

The binary representation 000 through 111 for restart number 0 through 7, respectively.

Data Transfer Group

This group of instructions transfer data to and from registers and memory. Condition flags are not affected by any instruction in this group.

MOV r1, r2 (Move Register)

(Move Register)

(r1) ← (r2)

The content of register r2 is moved to register r1.

MOV r,M (Move from memory)

(Move from memory)

(r) ← ((H) (L))

The content of the memory location, whose address is in registers H and L, is moved to register r.

Description Format

The following pages provide a detailed description of the instruction set of the 8080. Each instruction is described in the following manner:

1. The MAC 80 assembler format, consisting of the instruction mnemonic and operand fields, is printed in BOLDFACE on the left side of the first line.

   The name of the instruction is enclosed in parenthesis on the right side of the first line.
MOV M, r  (Move to memory)
((H)(L)) ← (r)
The content of register r is moved to the memory location whose address is in registers H and L.

MVI r, data  (Move Immediate)
(r) ← (byte 2)
The content of byte 2 of the instruction is moved to register r.

MVI M, data  (Move to memory immediate)
((H)(L)) ← (byte 2)
The content of byte 2 of the instruction is moved to the memory location whose address is in registers H and L.

LXI rp, data 16  (Load register pair immediate)
(rh) ← (byte 3),
(rl) ← (byte 2)
Byte 3 of the instruction is moved into the high-order register (rh) of the register pair rp. Byte 2 of the instruction is moved into the low-order register (rl) of the register pair rp.

LDA addr  (Load Accumulator direct)
(A) ← ((byte 3)(byte 2))
The content of the memory location, whose address is specified in byte 2 and byte 3 of the instruction, is moved to register A.

STA addr  (Store Accumulator direct)
((byte 3)(byte 2)) ← (A)
The content of the accumulator is moved to the memory location whose address is specified in byte 2 and byte 3 of the instruction.
**MOV M, r**  
(Move to memory)

\[(H)(L) \rightarrow (r)\]

The content of register \( r \) is moved to the memory location whose address is in registers H and L.

<table>
<thead>
<tr>
<th>Addressing:</th>
<th>reg. indirect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cycles:</td>
<td>2</td>
</tr>
<tr>
<td>States:</td>
<td>7</td>
</tr>
</tbody>
</table>

**MVI r, data**  
(Move Immediate)

\[(r) \leftarrow \text{byte 2}\]

The content of byte 2 of the instruction is moved to register \( r \).

<table>
<thead>
<tr>
<th>Addressing:</th>
<th>immediate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cycles:</td>
<td>2</td>
</tr>
<tr>
<td>States:</td>
<td>7</td>
</tr>
</tbody>
</table>

**MVI M, data**  
(Move to memory immediate)

\[\text{byte 2} \leftarrow \text{byte 3}\]

The content of byte 2 of the instruction is moved to the memory location whose address is in registers H and L.

<table>
<thead>
<tr>
<th>Addressing:</th>
<th>immed./reg. indirect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cycles:</td>
<td>3</td>
</tr>
<tr>
<td>States:</td>
<td>10</td>
</tr>
</tbody>
</table>

**LXI rp, data 16**  
(Load register pair immediate)

\[(rh) \leftarrow \text{byte 3}, (rl) \leftarrow \text{byte 2}\]

Byte 3 of the instruction is moved into the high-order register (rh) of the register pair rp. Byte 2 of the instruction is moved into the low-order register (rl) of the register pair rp.

<table>
<thead>
<tr>
<th>Addressing:</th>
<th>immediate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cycles:</td>
<td>3</td>
</tr>
<tr>
<td>States:</td>
<td>10</td>
</tr>
</tbody>
</table>

**LDA addr**  
(Load Accumulator direct)

\[(A) \leftarrow \text{byte 2}((byte 3))\]

The content of the memory location, whose address is specified in byte 2 and byte 3 of the instruction, is moved to register A.

<table>
<thead>
<tr>
<th>Addressing:</th>
<th>direct</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cycles:</td>
<td>4</td>
</tr>
<tr>
<td>States:</td>
<td>13</td>
</tr>
</tbody>
</table>

**STA addr**  
(Store Accumulator direct)

\[\text{byte 2}((byte 3)) \leftarrow (A)\]

The content of the accumulator is moved to the memory location whose address is specified in byte 2 and byte 3 of the instruction.

<table>
<thead>
<tr>
<th>Addressing:</th>
<th>direct</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cycles:</td>
<td>4</td>
</tr>
<tr>
<td>States:</td>
<td>13</td>
</tr>
</tbody>
</table>
Arithmetic Group

This group of instructions performs arithmetic operations on data in registers and memory.

Unless otherwise indicated, all instructions in this group affect the Zero, Sign, Parity, Carry, and Auxiliary Carry flags according to the standard rules.

All subtraction operations are performed via two’s complement arithmetic, and set the carry flag to one to indicate a borrow and clear it to indicate no borrow.

**ADD r**  (Add Register)

\[(A) \leftarrow (A) + (r)\]

The content of register r is added to the content of the accumulator. The result is placed in the accumulator.

**ADD M**  (Add Memory)

\[(A) \leftarrow (A) + ((H)(L))\]

The content of the memory location whose address is contained in the H and L registers is added to the content of the accumulator. The result is placed in the accumulator.

**ADI data**  (Add Immediate)

\[(A) \leftarrow (A) + \text{(byte 2)}\]

The content of the second byte of the instruction is added to the constant of the accumulator. The result is placed in the accumulator.

**ADC r**  (Add Register with Carry)

\[(A) \leftarrow (A) + (r) + (CY)\]

The content of register r and the content of the carry bit are added to the content of the accumulator. The result is placed in the accumulator.

**ADC M**  (Add Memory with Carry)

\[(A) \leftarrow (A) + ((H)(L)) + (CY)\]

The content of the memory location whose address is contained in the H and L registers and the content of the CY flag are added to the accumulator. The result is placed in the accumulator.
ACI data (Immediate with Carry)
(A) ← (A) + (byte 2) + (CY)
The content of the second byte of the instruction and the content of the CY flag are added to the contents of the accumulator. The result is placed in the accumulator.

```
data
1 1 0 0 1 1 1 0
```

Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

SUI data (Subtract Immediate)
(A) ← (A) − (byte 2)
The content of the second byte of the instruction is subtracted from the content of the accumulator. The result is placed in the accumulator.

```
data
1 1 0 0 1 1 1 0
```

Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

SUB r (Subtract Register)
(A) ← (A) − (r)
The content of register r is subtracted from the content of the accumulator. The result is placed in the accumulator.

```
data
1 0 0 1 0 0 0 1
```

Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

SBB r (Subtract Register with Borrow)
(A) ← (A) − (r) − (CY)
The content of register r and the content of the CY flag are both subtracted from the accumulator. The result is placed in the accumulator.

```
data
1 1 0 0 1 1 1 0
```

Cycles: 2
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

SUB M (Subtract Memory)
(A) ← (A) − ((H)(L))
The content of the memory location whose address is contained in the H and L registers is subtracted from the content of the accumulator. The result is placed in the accumulator.

```
data
1 0 0 1 0 1 1 0
```

Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

SBB M (Subtract Memory with Borrow)
(A) ← (A) − ((H)(L)) − (CY)
The content of the memory location whose address is contained in the H and L registers and the content of the CY flag are both subtracted from the accumulator. The result is placed in the accumulator.

```
data
1 1 0 0 1 1 1 0
```

Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC
SBI data  (Subtract Immediate with Borrow)  
(A) ← (A) − (byte 2) − (CY)  
The contents of the second byte of the instruction and the contents of the CY flag are both subtracted from the accumulator. The result is placed in the accumulator.

```
 1 1 1 0 1 1 1 0
| data |
```

Cycles: 2  
States: 7  
Addressing: immediate  
Flags: Z,S,P,CY,AC

INR r  (Increment Register)  
(r) ← (r) + 1  
The content of register r is incremented by one.  
Note: All condition flags except CY are affected.

```
 0 0 D D D 1 0 0
|       |
```

Cycles: 1  
States: 5  
Addressing: register  
Flags: Z,S,P,AC

DCX rp  (Decrement register pair)  
(rh)(rl) ← (rh)(rl) − 1  
The content of the register pair rp is decremented by one. Note: No condition flags are affected.

```
 0 0 R P 1 0 1 1
|       |
```

Cycles: 1  
States: 5  
Addressing: register  
Flags: none

INR M  (Increment Memory)  
((H)(L)) ← ((H)(L)) + 1  
The content of the memory location whose address is contained in the H and L registers is incremented by one. Note: All condition flags except CY are affected.

```
 0 0 1 1 0 1 0 0
|       |
```

Cycles: 3  
States: 10  
Addressing: reg. indirect  
Flags: Z,S,P,AC

DCR r  (Decrement Register)  
(r) ← (r) − 1  
The content of register r is decremented by one.  
Note: All condition flags except CY are affected.

```
 0 0 D D D 1 0 1
|       |
```

Cycles: 1  
States: 5  
Addressing: register  
Flags: Z,S,P,AC

DAD rp  (Add register pair to H and L)  
(H)(L) ← (H)(L) + (rh)(rl)  
The content of the register pair rp is added to the content of the register pair H and L. The result is placed in the register pair H and L. Note: Only the CY flag is affected. It is set if there is a carry out of the double precision add; otherwise it is reset.

```
 0 0 R P 1 0 0 1
|       |
```

Cycles: 3  
States: 10  
Addressing: register  
Flags: CY

DAA  (Decimal Adjust Accumulator)  
The 8-bit number in the accumulator is adjusted to form two 4-bit Binary-Coded-Decimal digits by the following process:

1. If the value of the least significant 4 bits of the accumulator is greater than 9 or if the AC flag is set, 6 is added to the accumulator.
2. If the value of the most significant 4 bits of
the accumulator is now greater than 9, or if
the CY flag is set, 6 is added to the most sig­
nificant 4 bits of the accumulator.

NOTE: All flags are affected.

Logical Group

This group of instructions performs logical (Boo-
lean) operations on data in registers and memory
and on condition flags.

Unless indicated otherwise, all instructions in this
group affect the Zero, Sign, Parity, Auxiliary Car-
ry, and Carry flags according to the standard rules.

ANA r  (AND Register)
(A) ← (A) ∧ (r)
The content of register r is logically ANDed with
the content of the accumulator. The result is
placed in the accumulator. The CY flag is
cleared.

XRA r  (Exclusive OR Register)
(A) ← (A) ∨ (r)
The content of register r is exclusive-ORed with
the content of the accumulator. The result is
placed in the accumulator. The CY and AC flags
are cleared.

ANA M  (AND memory)
(A) ← (A) ∧ ((H)(L))
The contents of the memory location whose ad-
dress is contained in the H and L registers is
logically ANDed with the content of the accumu-
lator. The CY flag is cleared.

XRA M  (Exclusive OR Memory)
(A) ← (A) ∨ ((H)(L))
The content of the memory location whose ad-
dress is contained in the H and L registers is
exclusive-ORed with the content of the accumu-
lator. The result is placed in the accumulator.
The CY and AC flags are cleared.
XRI data  (Exclusive OR immediate)

(A) ← (A) V (byte 2)

The content of the second byte of the instruction is exclusive-ORed with the content of the accumulator. The result is placed in the accumulator. The CY and AC flags are cleared.

ORA r  (OR Register)

(A) ← (A) V (r)

The content of register r is inclusive-ORed with the content of the accumulator. The result is placed in the accumulator. The CY and AC flags are cleared.

ORA M  (OR Memory)

(A) ← (A) V ((H)(L))

The content of the memory location whose address is contained in the H and L registers is inclusive-ORed with the content of the accumulator. The result is placed in the accumulator. The CY and AC flags are cleared.

CMP r  (Compare Register)

(A) − (r)

The content of register r is subtracted from the accumulator. The accumulator remains unchanged. The condition flags are set as a result of the subtraction. The Z flag is set to 1 if (A) = (r). The CY flag is set to 1 if (A) < (r).

CMP M  (Compare memory)

(A) − ((H)(L))

The content of the memory location whose address is contained in the H and L registers is subtracted from the accumulator. The accumulator remains unchanged. The condition flags are set as a result of the subtraction. The Z flag is set to 1 if (A) = ((H)(L)). The CY flag is set to 1 if (A) < ((H)(L)).
CPI data  (Compare immediate)
(A) - (byte 2)
The content of the second byte of the instruction is subtracted from the accumulator. The condition flags are set by the result of the subtraction. The Z flag is set to 1 if (A) = (byte 2). The CY flag is set to 1 if (A) < (byte 2).

```
data
1 1 1 1 1 1 1 0
```

- Cycles: 2
- States: 7
- Addressing: immediate
- Flags: Z,S,P,CY,AC

RLC  (Rotate left)
\((A_{n+1}) \leftarrow (A_n) ; (A_0) \leftarrow (A_7) \)
\((CY) \leftarrow (A_7)\)
The content of the accumulator is rotated left one position. The low-order bits and the CY flag are both set to the value shifted out of the high-order bit position. Only the CY flag is affected.

```
0 0 0 0 1 1 1 1
```

- Cycles: 1
- States: 4
- Flags: CY

RAR  (Rotate right)
\((A_n) \leftarrow (A_{n-1}) ; (A_7) \leftarrow (A_0) \)
\((CY) \leftarrow (A_0)\)
The content of the accumulator is rotated right one position through the CY flag. The high-order bit is set to the CY flag and the CY flag is set to the value shifted out of the low-order bit position. Only the CY flag is affected.

```
0 0 0 0 1 1 1 1
```

- Cycles: 1
- States: 4
- Flags: CY

CMA  (Complement accumulator)
\((A) \leftarrow (A)\)
The contents of the accumulator are complemented (zero bits become 1, one bits become 0). No flags are affected.

```
0 0 1 0 1 1 1 1
```

- Cycles: 1
- States: 4
- Flags: none

CMC  (Complement carry)
\((CY) \leftarrow (CY)\)
The CY flag is complemented. No other flags are affected.

```
0 0 1 1 1 1 1 1
```

- Cycles: 1
- States: 4
- Flags: CY
The CY flag is set to 1. No other flags are affected.

Branch Group

This group of instructions alter normal sequential program flow.

Condition flags are not affected by an instruction in this group.

The two types of branch instructions are unconditional and conditional. Unconditional transfers simply perform the specified operation on register PC (the program counter). Conditional transfers examine the status of one of the four processor flags to determine if the specified branch is to be executed. The conditions that may be specified are as follows:

<table>
<thead>
<tr>
<th>CONDITION</th>
<th>CCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>NZ</td>
<td>000</td>
</tr>
<tr>
<td>Z</td>
<td>001</td>
</tr>
<tr>
<td>NC</td>
<td>010</td>
</tr>
<tr>
<td>C</td>
<td>011</td>
</tr>
<tr>
<td>PO</td>
<td>100</td>
</tr>
<tr>
<td>PE</td>
<td>101</td>
</tr>
<tr>
<td>P</td>
<td>110</td>
</tr>
<tr>
<td>M</td>
<td>111</td>
</tr>
</tbody>
</table>

JMP addr (Jump)

(PC) → (byte 3)(byte 2)

Control is transferred to the instruction whose address is specified in byte 3 and byte 2 of the current instruction.

Jcondition addr (Conditional jump)

If (CCC),

(PC) ← (byte 3)(byte 2)

If the specified condition is true, control is transferred to the instruction whose address is specified in byte 3 and byte 2 of the current instruction; otherwise, control continues sequentially.

CALL addr (Call)

((SP) − 1) ← (PCH)

((SP) − 2) ← (PCL)

(SP) ← (SP) − 2

(PC) ← (byte 3)(byte 2)

The high-order 8 bits of the next instruction address are moved to the memory location whose address is one less than the content of register SP. The low-order 8 bits of the next instruction address are moved to the memory location whose address is two less than the content of register SP. The content of register SP is decremented by 2. Control is transferred to the instruction whose address is specified in byte 3 and byte 2 of the current instruction.
Condition addr  (Condition call)
If (CCC),
(SP) ← (PCH)
(SP) ← (PCL)
(SP) ← (SP) − 2
(PC) ← (byte 3)(byte 2)
If the specified condition is true, the actions specified in the CALL instruction (see above) are performed; otherwise, control continues sequentially.

<table>
<thead>
<tr>
<th>low-order addr</th>
<th>high-order addr</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1</td>
<td>C</td>
</tr>
</tbody>
</table>

Cycles: 3/5
States: 11/17
Addressing: immed./reg. indirect
Flags: none

RET  (Return)
(PCL) ← ((SP));
(PCH) ← ((SP) + 1);
(SP) ← (SP) + 2;
The content of the memory location whose address is specified in register SP is moved to the low-order 8 bits of register PC. The content of the memory location whose address is one more than the content of register SP is moved to the high-order 8 bits of register PC. The content of register SP is incremented by 2.

| 1 1 0 0 1 0 0 1 |

Cycles: 3
States: 10
Addressing: reg. indirect
Flags: none

RST n  (Restart)
((SP) − 1) ← (PCH)
((SP) − 2) ← (PCL)
(SP) ← (SP) − 2
(PC) ← 8 * (NNN)
The high-order 8 bits of the next instruction address are moved to the memory location whose address is one less than the content of register SP. The low-order 8 bits of the next instruction address are moved to the memory location whose address is two less than the content of register SP. The content of register SP is decremented by two. Control is transferred to the instruction whose address is eight times the content of NNN.

| 1 1 N N N | 1 1 1 |

Cycles: 3
States: 11
Addressing: reg. indirect
Flags: none

PCHL  (Jump H and L indirect — move H and L to PC)
(PCH) ← (H)
(PCL) ← (L)
The content of register H is moved to the high-order 8 bits of register PC. The content of register L is moved to the low-order 8 bits of register PC.

| 1 1 1 0 1 0 0 1 |

Cycles: 1
States: 5
Addressing: register
Flags: none
Stack, I/O, and Machine Control Group

This group of instructions performs I/O, manipulates the Stack, and alters internal control flags.

Unless otherwise specified, condition flags are not affected by any instructions in this group.

**PUSH rp**  (Push)

\[ ((\text{SP}) - 1) \leftarrow (\text{rh}) \]
\[ ((\text{SP}) - 2) \leftarrow (\text{rl}) \]
\[ (\text{SP}) \leftarrow (\text{SP}) - 2 \]

The content of the high-order register of register pair rp is moved to the memory location whose address is one less than the content of register SP. The content of the low-order register of register pair rp is moved to the memory location whose address is two less than the content of register SP. The content of register SP is decremented by 2. Note: Register pair rp=SP may not be specified.

![Cycles: 3
States: 11
Addressing: reg. indirect
Flags: none](image)

**POP rp**  (Pop)

\[ (\text{rl}) \leftarrow ((\text{SP}) + 1) \]
\[ (\text{rh}) \leftarrow ((\text{SP}) + 2) \]

The content of the memory location whose address is specified by the content of register SP, is moved to the low-order register of register pair rp. The content of the memory location, whose address is one more than the content of register SP, is moved to the high-order register of register pair rp. The content of register SP is incremented by 2. Note: Register pair rp=SP may not be specified.

![Cycles: 3
States: 10
Addressing: reg. indirect
Flags: none](image)

**PUSH PSW**  (Push processor status word)

\[ ((\text{SP}) - 1) \leftarrow (A) \]
\[ ((\text{SP}) - 2) \leftarrow ((\text{CY}), ((\text{SP}) - 2) \leftarrow 1) \]
\[ ((\text{SP}) - 2) \leftarrow ((P), ((\text{SP}) - 2) \leftarrow 0) \]
\[ ((\text{SP}) - 2) \leftarrow ((AC), ((\text{SP}) - 2) \leftarrow 0) \]
\[ ((\text{SP}) - 2) \leftarrow ((Z), ((\text{SP}) - 2) \leftarrow (S)) \]
\[ (\text{SP}) \leftarrow (\text{SP}) - 2 \]

The content of register A is moved to the memory location whose address is one less than register SP. The contents of the condition flags are assembled into a processor status word and the word is moved to the memory location whose address is two less than the content of register SP. The content of register SP is decremented by two.

![Cycles: 3
States: 11
Addressing: reg. indirect
Flags: none](image)

**POP PSW**  (Pop processor status word)

\[ (\text{CY}) \leftarrow ((\text{SP}) + 0) \]
\[ (\text{P}) \leftarrow ((\text{SP}) + 2) \]
\[ (\text{AC}) \leftarrow ((\text{SP}) + 4) \]
\[ (\text{Z}) \leftarrow ((\text{SP}) + 6) \]
\[ (\text{S}) \leftarrow ((\text{SP}) + 7) \]
\[ (\text{A}) \leftarrow ((\text{SP}) + 1) \]
\[ (\text{SP}) \leftarrow (\text{SP}) + 2 \]

The content of the memory location whose address is specified by the content of register SP is used to restore the condition flags. The content of the memory location whose address is one more than the content of register SP is moved to register A. The content of register SP is incremented by 2.

![Cycles: 3
States: 10
Addressing: reg. indirect
Flags: Z,S,P,CY,AC](image)
XTHL  (Exchange stack top with H and L)
(L) ← ((SP))
(H) ← ((SP) + 1)
The content of the L register is exchanged with the content of the memory location whose address is specified by the content of register SP. The content of the H register is exchanged with the content of the memory location whose address is one more than the content of register SP.

SPHL  (Move HL to SP)
(SP) ← (H)(L)
The contents of registers H and L (16 bits) are moved to register SP.

EI    (Enable interrupt)
The interrupt system is enabled following the execution of the next instruction.

DI    (Disable interrupts)
The interrupt system is disabled immediately following the execution of the DI instruction.

HLT   (Halt)
The processor is stopped. The registers and flags are unaffected.

NOP   (No op)
No operation is performed. The registers and flags are unaffected.
RESISTOR NETWORK SPECIFICATIONS:

RESISTANCE VALUES:
± 2% (MAX)

OPERATING TEMPERATURE:
0°C TO +70°C

TEMPERATURE COEFFICIENT:
1200 PPM °C OVER TEMPERATURE RANGE OF 0°C TO +70°C

OPERATING VOLTAGE:
5.0 VDC (MAX)

POWER RATING:
AT 70°C, 0.7 WATT PER PACK

TRACKING RESISTANCE RATIO:
1.0 % (MAX)

STABILITY:
± 1% YEAR (MAX)

LOAD LIFE:
± 1% (DR) OVER 1000 HOURS

PACKAGES:
DUAL IN LINE - CERAMIC OR PLASTIC

NOTES:
UNLESS OTHERWISE SPECIFIED,
1. PART NO 4500645-01
2. INK STAMP PRODUCT CODE, RESISTOR VALUE, PART NO.
AND DASH NUMBER WITH CONTRASTING COLOR AND MIN .12 HIGH CHARACTERS.
NO OTHER MARKINGS ARE PERMITTED EXCEPT FOR MANUF BATCH NO.
E.G.) 5BC-302
     .1K
     4500645-01
3. FOR PROCUREMENT SEE LV4500645
4. IDENTIFY PIN ONE CLEARLY ON TOP OF PACKAGE.

Copyright ©1976 Intel Corporation
Schematic drawings for the SBC-901 and SBC-902 are provided in this appendix. Information and diagrams in this section are subject to change without notice. References should be made to schematics shipped with this module.
RESISTOR NETWORK SPECIFICATIONS:

RESISTANCE VALUES:
±2% (MAX)

OPERATING TEMPERATURE:
0°C TO +70°C

TEMPERATURE COEFFICIENT:
±150 PPM/°C OVER TEMPERATURE RANGE OF UP TO +70°C

RATING VOLTAGE:
2.0 VDC (MAX)

POWER RATING:
AT 70°C, 0.7 WATT PER PACK

TRACKING RESISTANCE RATIO:
1.0% (MAX)

STORAGE:
1% YEAR (MAX)

LOAD LIFE:
1.1% (DR) OVER 1000 HOURS

PACKAGE:
DUAL IN LINE - CERAMIC OR PLASTIC

NOTES:
UNLESS OTHERWISE SPECIFIED:

1. PART NO. IS 4500644-01.

2. INK STAMP PRODUCT CODE,
INJECT VALUE, PART NO., AND DASH NUMBER WITH CONTRASTING
COLORS INCLUDING MIN .05

SHADE SIMILARLY ON DOCK.

EXCEPT MANUFACTURER'S BATCH NO.

E.G.) KC-201
R220Y/330
4500644-01

3. FOR PROCUREMENT SEE
LV 4500644-01.

IDENTIFY PIN ONE CLEARLY
ON TOP OF PACKAGE.

Copyright ©1976 Intel Corporation
ABSTRACT

THIS PROGRAM RUNS ON THE SBC 80/10 BOARD AND IS DESIGNED TO PROVIDE
THE USER WITH A MINIMAL MONITOR. BY USING THIS PROGRAM,
THE USER CAN EXAMINE AND CHANGE MEMORY OR CPU REGISTERS, LOAD
A PROGRAM (IN ABSOLUTE HEX) INTO RAM, AND EXECUTE INSTRUCTIONS
ALREADY IN MEMORY. THE MONITOR ALSO PROVIDES THE USER WITH
ROUTINES FOR PERFORMING CONSOLE I/O AND PAPER TAPE I/O.

PROGRAM ORGANIZATION

THE LISTING IS ORGANIZED IN THE FOLLOWING WAY. FIRST THE BASIC
MONITOR FUNCTIONS TOGETHER WITH THE CONSOLE I/O ARE LOCATED IN THE
FIRST 1K OF ROM FOLLOWED BY THE PAPER TAPE FUNCTIONS AND I/O IN THE
SECOND 1K OF ROM. WITHIN THE FIRST ROM IS CONTAINED THE COMMAND
RECOGNIZER, WHICH IS THE HIGHEST LEVEL ROUTINE IN THE PROGRAM.
NEXT THE ROUTINES TO IMPLEMENT THE VARIOUS COMMANDS. FINALLY,
THE UTILITY ROUTINES WHICH ACTUALLY DO THE DIRTY WORK. WITHIN
EACH SECTION, THE ROUTINES ARE ORGANIZED IN ALPHABETICAL
ORDER, BY ENTRY POINT OF THE ROUTINE. THE SECOND ROM IS ORGANIZED
IN THE SAME MANNER AS THE FIRST WITH THE ROUTINES WHICH IMPLEMENT
THE COMMANDS FOLLOWED BY THE UTILITY ROUTINES WHICH ACTUALLY DO THE
MORE DETAILED OPERATIONS.

THE PROGRAM HAS BEEN PARTITIONED IN SUCH A MANNER THAT THE SECOND
ROM NEED NOT BE PLUGGED INTO THE BOARD IF ONLY THE BASIC MONITOR
FUNCTIONS ARE REQUIRED. HOWEVER IF THE PAPER TAPE FUNCTIONS ARE DESIRED
BOTH ROMS ARE REQUIRED.

THIS PROGRAM EXPECTS TO RUN IN THE FIRST 2K OF ADDRESS SPACE.
IF, FOR SOME REASON, THE PROGRAM IS RE-ORG'ED, CARE SHOULD
BE TAKEN TO MAKE SURE THAT THE TRANSFER INSTRUCTIONS FOR RST 1
AND RST 7 ARE ADJUSTED APPROPRIATELY.

THE PROGRAM ALSO EXPECTS THAT RAM LOCATIONS 3C00H TO 3C3FH,
INCLUSIVE, ARE RESERVED FOR THE PROGRAM'S OWN USE. THESE
LOCATIONS MAY BE ALTERED, HOWEVER, BY CHANGING THE EQU'ED
SYMBOL "DATA" AS DESIRED.

LIST OF FUNCTIONS

1ST ROM

[ List of functions follows ]
INUST
NHOUT
PPVAL
REGDS
EOADR
BSTTP
SRET
STMP0
SCHLP
VALIG
VAL1D
-----

**********
2ND ROM
**********

RCMD
WCMD
----

BYTE
DELAY
LEAD
PAOR
PBYTE
PEOF
PEOL
PO
RI
RICH
-----

0010
ORG 2H

*******************************************************************************

MONITOR EQUATES

*******************************************************************************

001B  BRCHR EQU 1BH  ; CODE FOR BREAK CHARACTER (ESCAPE)
3C3D  BRLOC EQU 3C3DH  ; LOCATION OF USER BRANCH INSTRUCTION IN RAM
03FA  BRTAB EQU 3FAH  ; LOCATION OF START OF BRANCH TABLE IN ROM
0025  CMD EQU 025H  ; COMMAND INSTRUCTION FOR USART INITIALIZATION
00ED  CNCTL EQU 0EDH  ; CONSOLE (USART) CONTROL PORT
00EC  CNIN EQU 0UCH  ; CONSOLE INPUT PORT
00EE  CNOUT EQU 0ECH  ; CONSOLE OUTPUT PORT
MONITOR MACROS

TRUE MACRO WHERE ; BRANCH IF FUNCTION RETURNS TRUE (SUCCESS)
JNC WHERE
ENDM

FALSE MACRO WHERE ; BRANCH IF FUNCTION RETURNS FALSE (FAILURE)
JNC WHERE
ENDM

USART INITIALIZATION CODE
THE USART IS ASSUMED TO COME UP IN THE RESET POSITION (THIS
FUNCTION IS TAKEN CARE OF BY THE HARDWARE). THE USART WILL
BE INITIALIZED IN THE SAME WAY FOR EITHER A TTY OR CRT
INTERFACE. THE FOLLOWING PARAMETERS ARE USED:

- MODE INSTRUCTION
  ====== ===========
  2 STOP BITS
  PARITY DISABLED
  8 BIT CHARACTERS
  BAUD RATE FACTOR OF 64

- COMMAND INSTRUCTION
  ========= ===========
  NO HUNT MODE
  CRT(RTS) FORCED TO 0
  RECEIVE ENABLED
  TRANSMIT ENABLED

0000 JECF  ; OUTPUT MODE SET TO USART
0002 D3ED  ; OUTPUT MODE SET TO USART
0004 C3B202 ; OUTPUT MODE SET TO USART
0007 80  ; OUTPUT MODE SET TO USART

********************************************************************************

RESTART ENTRY POINT

********************************************************************************

; GO:
0008 22343C  ; SAVE HL REGISTERS
0009 E1      ; GET TOP OF STACK ENTRY
000C 22363C  ; ASSUME THIS IS LAST P COUNTER
000F F5      ; SAVE A,P/F'S
0010 210200  ; SET HL TO 2 SO THAT STACK POINTER SAVED CORRECTLY
0013 39      ; GET STACK POINTER VALUE
0014 22381C  ; SAVE USER'S STACK POINTER
0017 F1      ; RESTORE A,P/F'S
0018 31343C  ; NEW VALUE FOR STACK POINTER
001B C3B181  ; ADROUT

********************************************************************************
PRINT SIGNON MESSAGE

*****************************************************

SOMSG:

01E 219533 LXI H,SGNON ; GET ADDRESS OF SIGNON MESSAGE
021 5611 VVI B,SGNON ; COUNTER FOR CHARACTERS IN MESSAGE
023 4002 MOV C,M ; FETCH NEXT CHAR TO C REG
024 CDE01 CALL CO ; SEND IT TO THE CONSOLE
027 5C INX H ; POINT TO NEXT CHARACTER
028 65 DCR B ; DECREMENT BYTE COUNTER
029 C2300 JNZ MSGL ; RETURN FOR NEXT CHARACTER

*****************************************************

COMMAND RECOGNIZING ROUTINE

FUNCTION: GETCM

DESCRIPTION: GETCM RECEIVES AN INPUT CHARACTER FROM THE USER
AND ATTEMPTS TO LOCATE THIS CHARACTER IN ITS COMMAND
CHARACTER TABLE. IF SUCCESSFUL, THE ROUTINE
CORRESPONDING TO THIS CHARACTER IS SELECTED FROM
A TABLE OF COMMAND ROUTINE ADDRESSES, AND CONTROL
IS TRANSFERRED TO THIS ROUTINE. IF THE CHARACTER
DOES NOT MATCH ANY ENTRIES, CONTROL IS PASSED TO
THE ERROR HANDLER.

GETCM:

01C 312E3C LXI SP,MSTAX ; ALWAYS WANT TO RESET STACK PTR TO MONITOR
021 0E2E VVI C,',' ; STARTING VALUE SO ROUTINES NEEDN'T CLEAN UP
024 CDF901 CALL ECHO ; PROMPT CHARACTER TO USER TERMINAL
027 C3000 JMP GTC03 ; WANT TO LEAVE ROOM FOR RST BRANCH
028 C33D08 ORG RSTU ; ORG TO RST TRANSFER LOCATION
029 C33D3C JMP USRBR ; JUMP TO USER BRANCH LOCATION
031 00 NOP ; FILLER
032 C3 GTC03:
CALL GETCH ; GET COMMAND CHARACTER TO A
MOV A,C ; PUT COMMAND CHARACTER INTO ACCUMULATOR
LXI B,NCMD ; C CONTAINS LOOP AND INDEX COUNT
LXI H,CTAB ; HL POINTS INTO COMMAND TABLE
CALL GTC05: ; branch if equal - command recognized
CMP M ; compare table entry and character
JZ GTC10 ; else, increment table pointer
DCR C ; decrement loop count
JNZ GTC05 ; branch if not at table end
JMP ERROR ; else, command character is illegal
LXI H, CADR ; if good command, load address of table
GET COMMAND CHARACTER TO A
/OF COMMAND ROUTINE ADDRESSES
DAD B ; add what is left of loop count
DAD B ; add again - each entry in CADR is 2 bytes long
INX H ; point to next byte in table
MOV H,M ; get MSP of address of table entry to H
MOV L,A ; put LSP of address of table entry into L
; NEXT INSTRUCTION COMES FROM COMMAND ROUTINE
;**********************************************************************;

COMMAND IMPLEMENTING ROUTINES

FUNCTION: DCMD
INPUTS: NONE
OUTPUTS: NONE
CALLS: ECHO, NMOUT, HILO, GETCM, CROUT, GETNM
DESTROYS: A, B, C, D, E, H, L, F/F'S
DESCRIPTION: DCMD IMPLEMENTS THE DISPLAY MEMORY (D) COMMAND

WVI C, 2 ; get two numbers from input stream
CALL GETNM ; ENDING ADDRESS TO DE
POP D ; STARTING ADDRESS TO HL
DCM05:
CALL CROUT ; echo carriage return/line feed
CALL ADRD ; display address
WVI ECH0 ; use blank as separator
MOV A,M; GET CONTENTS OF NEXT MEMORY LOCATION
CALL NMOUT; DISPLAY CONTENTS
1 TRUE EXIT; IF SO, BRANCH TO EXIT
CALL BREAK; SEE IF USER WANTS OUT
CALL HILO; SEE IF ADDRESS OF DISPLAYED LOCATION IS
/GREATERTHAN OR EQUAL TO ENDING ADDRESS
1 TRUE EXIT; EXIT IF NO MORE TO DISPLAY
INX H; IF MORE TO GO, POINT TO NEXT LOC TO DISPLAY
MOV A,L; GET LOW ORDER BITS OF NEW ADDRESS
ANI NEWLN; SEE IF LAST HEX DIGIT OF ADDRESS DENOTES
/START OF NEW LINE
JNZ DCM10; NO - NOT AT END OF LINE
JMP DCM05; YES - START NEW LINE WITH ADDRESS

;*************************************************************
; ; FUNCTION: GCMD
; INPUTS: NONE
; OUTPUTS: NONE
; CALLS: ERROR,GETHX,RSTTF
; DESCRIPTION: GCMD IMPLEMENTS THE BEGIN EXECUTION (G) COMMAND.
;*************************************************************

CALL GETHX; GET ADDRESS (IF PRESENT) FROM INPUT STREAM
1 FALSE GCM05; BRANCH IF NO NUMBER PRESENT
JNC GCM05
MOV A,D; ELSE, GET TERMINATOR
CPI CR; SEE IF CARRIAGE RETURN
JNZ ERROR; ERROR IF NOT PROPERLY TERMINATED
LXI H,PSAVE; WANT NUMBER TO REPLACE SAVE PGM COUNTER
MOV M,C
INX H
MOV M,B
JMP GCM10
GCM05:
MOV A,D; IF NO STARTING ADDRESS, MAKE SURE THAT
CPI CR; /CARRIAGE RETURN TERMINATED COMMAND
JNZ ERROR; ERROR IF NOT
GCM10:
JMP RSTTF; RESTORE REGISTERS AND BEGIN EXECUTION

;*************************************************************
;
; FUNCTION: ICMD
; INPUTS: NONE
; OUTPUTS: NONE
; CALLS: ERROR, ECHO, GETCH, VALDL, VALDG, CVNVBN, STHLF, GETNM, CROUT
; DESTROYS: A, B, C, D, E, H, L, F/F'S
; DESCRIPTION: ICMD IMPLEMENTS THE INSERT CODE INTO MEMORY (I) COMMAND.

ICMD:

0CA9 0E81 MVI C,1
0CA9 2E92 CALL GETNM ; GET SINGLE NUMBER FROM INPUT STREAM
0CAE 3EFP XLI A, UPPER
0CB2 323A3C STA TEMP ; TEMP WILL HOLD THE UPPER/LOWER HALF BYTE FLAG
0CB3 D1 POP D ; ADDRESS OF START TO BE

ICMO5:

0CC4 CD2C52 CALL GETCH ; GET A CHARACTER FROM INPUT STREAM
0CC7 CDFFC1 CALL ECHO ; ECHO IT
0CCA 79 MOV A, C ; PUT CHARACTER BACK INTO A
0CB8 FE1B CPI TERM ; SEE IF CHARACTER IS A TERMINATING CHARACTER
0CBF CAL59 C2 ICM25 ; IF SO, ALL DONE ENTERING CHARACTERS
0CC0 CD6251 CALL VALDL ; ELSE, SEE IF VALID DELIMITER
0CC1 1 TRUE ICM25 ; IF SO SIMPLY IGNORE THIS CHARACTER
0CC3 1 DAB4C0 + JC ICM85
0CC6 CD6753 CALL VALDG ; ELSE, CHECK TO SEE IF VALID HEX DIGIT
0CC7 0 FALSE ICM20 ; IF NOT, BRANCH TO HANDLE ERROR CONDITION
0CC9 1 D2E9C0 + JNC ICM20
0CCB CDDB51 CALL CVNVBN ; CONVERT DIGIT TO BINARY
0CCF 4F MOV C, A ; MOVE RESULT TO C
0CD0 CD4B63 CALL STHLF ; STORE IN APPROPRIATE HALF WORD
0CD2 3A3A3C LDA TEMP ; GET HALF BYTE FLAG
0CD6 B7 GFA A ; SET F/F'S
0CD7 C2E9C0 JNZ ICM10 ; BRANCH IF FLAG SET FOR UPPER
0CDA 13 INX D ; IF LOWER, INC ADDRESS OF BYTE TO STORE IN

ICMO10:

0CEB EEFF XRI INVRT ; TOGGLE STATE OF FLAG
0CED 323A3C STA TEMP ; PUT NEW VALUE OF FLAG BACK
0CEC C359C0 JMP ICM05 ; PROCESS NEXT DIGIT

ICMO20:

0CE3 CD3D53 CALL STHF0 ; ILLEGAL CHARACTER
0CE6 C312C0 JMP ERROR ; MAKE SURE ENTIRE BYTE FILLED THEN ERROR

ICMO25:

0CE9 CD3D53 CALL STHF0 ; HERE FOR ESCAPE CHARACTER - INPUT IS DONE

0CEC C312C0 JMP EXIT

;*****************************************************************

FUNCTION: KCMD
; INPUTS: NONE
; OUTPUTS: NONE
; CALLS: GETCH, HILO, GETNM
DESTROYS: A,B,C,D,E,H,L,F/F'S

DESCRIPTION: MCMD IMPLEMENTS THE MOVE DATA IN MEMORY (M) COMMAND.

00EF 0E03  MVI C,3
00F1 CD5B05 CALL GETNM ; GET 3 NUMBERS FROM INPUT STREAM
00F4 C1 POP B ; DESTINATION ADDRESS TO BC
00F5 E1 POP H ; ENDING ADDRESS TO HL
00F6 D1 POP D ; STARTING ADDRESS TO DE
00F7 MCM05:
00F7 E5  PUSH H ; SAVE ENDING ADDRESS
00F8 62  MOV H,D
00F9 63  MOV L,E ; SOURCE ADDRESS TO HL
00FA 7E  MOV A,M ; GET SOURCE BYTE
00FB 60  MOV H,B
00FC 69  MOV L,C ; DESTINATION ADDRESS TO HL
00FD 77  MOV K,A ; MOVE BYTE TO DESTINATION
00FE 03  INX B ; INCREMENT DESTINATION ADDRESS
00FF 7E  MOV A,B
0100 81  GVA C ; TEST FOR DESTINATION ADDRESS OVERFLOW
0101 CA2C05 J2 GETCM ; IF SO, CAN TERMINATE COMMAND
0104 13  INX D ; INCREMENT SOURCE ADDRESS
0105 E1  POP H ; ELSE, GET BACK ENDING ADDRESS
0106 CD4A02 CALL HILO ; SEE IF ENDING ADDR>=SOURCE ADDR
1 + FALSE GETCM ; IF NOT, COMMAND IS DONE
0109 1 D22C00 + JNC GETCM
010C C3F707 JMP MCM05 ; MOVE ANOTHER BYTE

FUNCTION: SCMD
INPUTS: NONE
OUTPUTS: NONE
CALLS: GETNM,GETCM,NOUT,ECHO
DESTROYS: A,B,C,D,E,H,L,F/F'S
DESCRIPTION: SCMD IMPLEMENTS THE SUBSTITUTE INTO MEMORY (S) COMMAND.

010F CD2705 CALL GETHX ; GET A NUMBER, IF PRESENT, FROM INPUT
0112 C5  PUSH B ; GET NUMBER TO HL - DENOTES MEMORY LOCATION
0113 E1  POP H
0114 SCM05:
0114 7A  MOV A,D ; GET TERMINATOR
0115 FE20  CPI ',' ; SEE IF SPACE
0117 CA1805 J2 SCM10 ; YES - CONTINUE PROCESSING
011A FE2C  CPI ',' ; ELSE, SEE IF COMMA
011C C22C00 JNZ GETCM ; NO - TERMINATE COMMAND
011F SCM10:
011F 7E  MOV A,M ; GET CONTENTS OF SPECIFIED LOCATION TO A
CALL NMOUT ; DISPLAY CONTENTS ON CONSOLE
CALL ECHO ; USE DASH FOR SEPARATOR
CALL GETHX ; GET NEW VALUE FOR MEMORY LOCATION, IF ANY
JNC SCM15 ; IF NO VALUE PRESENT, BRANCH
MOV R,C ; ELSE, STORE LOWER 0 BITS OF NUMBER ENTERED
INX H ; INCREMENT ADDRESS OF MEMORY LOCATION TO VIEW

**************

FUNCTION: XCMD
INPUTS: NONE
OUTPUTS: NONE
CALLS: GETCH,ECHO,REGDS,GETCM,ERROR,RGADR,NMOUT,CROUT,GETHX
DESTROYS: A,E,C,D,E,H,L,F/F'S
DESCRIPTION: XCMD IMPLEMENTS THE REGISTER EXAMINE AND CHANGE (X) COMMAND.

CALL GETCH ; GET REGISTER IDENTIFIER
CALL ECHO ; ECHO IT
MOV A,C
CPI CS
JNZ XCM05 ; BRANCH IF NOT CARRIAGE RETURN
CALL REGDS ; ELSE, DISPLAY REGISTER CONTENTS
JMP GETCM ; THEN TERMINATE COMMAND

MOV C,A ; GET REGISTER IDENTIFIER TO C
CALL RGADR ; CONVERT IDENTIFIER INTO RTAB TABLE ADDR
PUSH R
POP H ; PUT POINTER TO REGISTER ENTRY INTO HL
MVI C,' '
CALL ECHO ; ECHO SPACE TO USER
MOV A,C
STA TEMP ; PUT SPACE INTO TEMP AS DELIMITER

LDA TEMP ; GET TERMINATOR
LDA ' ' ; SEE IF A BLANK
JMP XCM15 ; YES - GO CHECK POINTER INTO TABLE
CPI ' ' ; NO - SEE IF COMMA
JNZ GETCM ; NO - MUST BE CARRIAGE RETURN TO END COMMAND

MOV A,M ; SET F/F'S
ORA A
EXIT ; BRANCH IF AT END OF TABLE
PUSH H ; PUT POINTER ON STACK
8080 MACRO ASSEMBLER, VER 2.4  ERRORS = 0  PAGE 12
80/10 MONITOR, VERSION 1.1, 1 NOVEMBER 1976

**Utility Routines**

```
C167  5E  MOV E,M
C168  16B2  MVI D,DATA SHR 8  ; FETCH ADDRESS OF SAVE LOCATION FROM
C169  83  INX H  ; /TABLE
C16A  46  MOV B,M  ; FETCH LENGTH FLAG FROM TABLE
C16B  D5  PUSH D  ; SAVE ADDRESS OF SAVE LOCATION
C16C  D5  PUSH D
C16D  E1  POP H  ; MOVE ADDRESS TO HL
C16E  C5  PUSH E  ; SAVE LENGTH FLAG
C16F  7E  MOV A,M  ; GET 8 BITS OF REGISTER FROM SAVE LOCATION
C170  CEC202  CALL NSOUT  ; DISPLAY IT
C171  F1  POP PSW  ; GET BACK LENGTH FLAG
C172  F5  PUSH PSW  ; SAVE IT AGAIN
C173  B7  GPA A  ; SET F/F'S
C174  C7F01  JZ XCM20  ; IF 8 BIT REGISTER, NOTHING MORE TO DISPLAY
C175  25  DEX H  ; ELSE, FOR 16 BIT REGISTER, GET LOWER 8 BITS
C176  7E  MOV A,M
C177  CCE202  CALL NSOUT  ; DISPLAY THEM
C178  832D  MVI C,'-'  ; USE DASH AS SEPARATOR
C179  CCF901  CALL ECHO  ; USE DASH AS SEPARATOR
C17A  CCE202  CALL GETHX  ; SEE IF THERE IS A VALUE TO PUT INTO REGISTER
C17B  1 + FALSE XCM30  ; NO - GO CHECK FOR NEXT REGISTER
C17C  1 D29F61  + JNC XCM33
C17D  7A  MOV A,D
C17E  332A3C  STA TEMP  ; ELSE, SAVE THE TERMINATOR FOR NOW
C17F  E1  POP PSW  ; GET BACK LENGTH FLAG
C180  E1  POP H  ; PUT ADDRESS OF SAVE LOCATION INTO HL
C181  B7  GPA A  ; SET F/F'S
C182  C7F01  J2 XCM25  ; IF 8 BIT REGISTER, BRANCH
C183  70  MOV X,M  ; SAVE UPPER 6 BITS
C184  2B  DEX H  ; POINT TO SAVE LOCATION FOR LOWER 8 BITS
C185  XCM25:
C186  71  MOV N,C  ; STORE ALL OF 8 BIT OR LOWER 1/2 OF 16 BIT REG
C187  XCM27:
C188  110300  LXI D,RTABS  ; SIZE OF ENTRY IN RTAB TABLE
C189  E1  POP H  ; POINTER INTO REGISTER TABLE RTAB
C18A  19  BAD D  ; ADD ENTRY SIZE TO POINTER
C18B  C35421  JMP XCM10  ; DO NEXT REGISTER
C18C  XCM30:
C18D  7A  MOV A,D  ; GET TERMINATOR
C18E  332A3C  STA TEMP  ; SAVE IN MEMORY
C18F  D1  POP D  ; CLEAR STACK OF LENGTH FLAG AND ADDRESS
C190  D1  POP D  ; /OF SAVE LOCATION
C191  C397B1  JMP XCM27  ; GO INCREMENT REGISTER TABLE POINTER
```

-----------------------------

**Utility Routines**
FUNCTION ADRD
INPUTS: HL - ADDRESS TO BE DISPLAYED
OUTPUTS: NONE
CALLS: NMOUT
DESTROYS: A
DESCRIPTION: ADRD OUTPUTS TO THE CONSOLE THE ADDRESS CONTAINED IN THE HL REGISTERS.

ADRD:
01A8
01A8 7C MOV A,H ; DISPLAY FIRST HALF OF ADDRESS
01AC 7D MOV A,L ; DISPLAY SECOND HALF OF ADDRESS
01AD CDC282 CALL NMOUT
01B0 C9 RET ; RETURN TO CALLING ROUTINE

FUNCTION ADROUT
INPUTS: USER REGISTERS ON THE STACK
OUTPUTS: NOTHING
CALLS: ECHO, ADRD
DESTROYS: A, B, C, D, E, H, L, F/F'S
DESCRIPTION: ADROUT SAVES THE USER REGISTERS AND OUTPUTS TO THE CONSOLE THE USER P COUNTER AFTER A RST 1 INSTRUCTION.

ADROUT:
01B1
01B1 F5 PUSH PSA ; SAVE A AND FLAGS
01B2 C5 PUSH B ; SAVE B AND C
01B3 D5 PUSH D ; SAVE D AND E
01B4 0223 MOV C,'\n' 01B6 CDC90E CALL ECHO ; OUTPUT '\n'
01B9 2A336C LIHLD PSAVE ; LOAD USER P COUNTER
01DC CDC80E CALL ADRD ; DISPLAY ADDRESS
01EF C3172 JMP EXIT ; GET NEW COMMAND

FUNCTION: BREAK
INPUTS: NONE
OUTPUTS: CARRY - 1 IF ESCAPE CHARACTER INPUT
- 0 IF ANY OTHER CHARACTER OR NO CHARACTER PENDING
CALLS: NOTHING
DESTROYS: A,F/F'S
I
VI
8080 MACRO ASSEMBLER, VER 2.4
ERRORS = 0 PAGE 14
68/18 MONITOR, VERSION 1.1, 1 NOVEMBER 1976

; DESCRIPTION: BREAK IS USED TO SENSE AN ESCAPE CHARACTER FROM
; THE USER. IF NO CHARACTER IS PENDING, OR IF THE
; PENDING CHARACTER IS NOT THE ESCAPE, THEN A FAILURE
; RETURN (CARRY=0) IS TAKEN. IN THIS CASE, THE
; PENDING CHARACTER (IF ANY) IS LOST. IF THE PENDING
; CHARACTER IS AN ESCAPE CHARACTER, BREAK TAKES A SUCCESS
; RETURN (CARRY=1).
;
BREAK:
01C2 DBED IN CONST ; GET CONSOLE STATUS
01C4 E602 ANI BBR ; SEE IF CHARACTER PENDING
01C6 CA1D23 J2 FRET ; NO - TAKE FAILURE RETURN
01C9 D5EC IN CNIN ; YES - PICK UP CHARACTER
01C3 E67F ANI PEYO ; STRIP OFF PARITY BIT
01CD FE15 CPI' SRCHR ; SEE IF BREAK CHARACTER
01C7 CA3B23 J2 SRET ; YES - SUCCESS RETURN
01D2 C31D23 JMP FRET ; NO - FAILURE RETURN - CHARACTER LOST
;
FUNCTION: CI
; INPUTS: NONE
; OUTPUTS: A - CHARACTER FROM CONSOLE
; CALLS: NOTHING
; DESTROYS: A,F/F'S
; DESCRIPTION: CI WAITS UNTIL A CHARACTER HAS BEEN ENTERED AT THE
; CONSOLE AND THEN RETURNS THE CHARACTER, VIA THE A
; REGISTER, TO THE CALLING ROUTINE. THIS ROUTINE
; IS CALLED BY THE USER VIA A JUMP TABLE IN RAM.
;
CI:
01D5 DBED IN CONST ; GET STATUS OF CONSOLE
01D7 E632 ANI BBR ; CHECK FOR RECEIVER BUFFER READY
01D9 CAD523 J2 CI ; NOT YET - WAIT
01DC DBEC IN CNIN ; READY SO GET CHARACTER
01DE C9 RET
;
FUNCTION: CNVBN
; INPUTS: C - ASCII CHARACTER '0'-'9' OR 'A'-'F'
; OUTPUTS: A - 0 TO F HEX
; CALLS: NOTHING
; DESTROYS: A,F/F'S
; DESCRIPTION: CNVBN CONVERTS THE ASCII REPRESENTATION OF A HEX
; CHARACTER INTO ITS CORRESPONDING BINARY VALUE. CNVBN
; DOES NOT CHECK THE VALIDITY OF ITS INPUT.
FUNCTION: CO
INPUTS: C - CHARACTER TO OUTPUT TO CONSOLE
OUTPUTS: C - CHARACTER OUTPUT TO CONSOLE
CALLS: NOTHING
DESTROYS: A,F/F'S
DESCRIPTION: CO WAITS UNTIL THE CONSOLE IS READY TO ACCEPT A CHARACTER
AND THEN SENDS THE INPUT ARGUMENT TO THE CONSOLE.

CO:

81E8 99 IN    ; GET STATUS OF CONSOLE
81EA E621 ANI   ; TRY
81EC 12 JZ    ; NO - WAIT
81F0 79 MOV   ; ELSE, MOVE CHARACTER TO A REGISTER FOR OUTPUT
81F2 D3EC OUT ; SEND TO CONSOLE
81F4 C9 RET

FUNCTION: CROUT
INPUTS: NONE
OUTPUTS: NONE
CALLS: ECHO
DESTROYS: A,B,C,F/F'S
DESCRIPTION: CROUT SENDS A CARRIAGE RETURN (AND HENCE A LINE FEED) TO THE CONSOLE.

CROUT:

81F3 068D MVI C,CR ; OUTPUT CARRIAGE RETURN TO USER TERMINAL
81F5 CDF9 CALL ECHO
81F8 C9 RET

FUNCTION: ECHO
; INPUTS:  C - CHARACTER TO ECHO TO TERMINAL
; OUTPUTS:  C - CHARACTER ECHOED TO TERMINAL
; CALLS:  CO
; DESTROYS:  A,B,F/F'S
; DESCRIPTION:  ECHO TAKES A SINGLE CHARACTER AS INPUT AND, VIA
;  THE MONITOR, SENDS THAT CHARACTER TO THE USER
;  TERMINAL. A CARRIAGE RETURN IS ECHOED AS A CARRIAGE
;  RETURN LINE FEED, AND AN ESCAPE CHARACTER IS ECHOED AS $.
;
; ECHO:
01F9  41  MOV  B,C  ; SAVE ARGUMENT
01FA  3E1B  MVI  A,ESC
01FC  B8  CMP  B  ; SEE IF ECHOING AN ESCAPE CHARACTER
01FD  C21D02  JNZ  ECHO5  ; NO - BRANCH
0200  0E24  MVI  C,'$'  ; YES - ECHO AS $
;
0202  CDE801  CALL  CO  ; DO OUTPUT THROUGH MONITOR
0205  3E1D  MVI  A,CR
0207  B6  CMP  B  ; SEE IF CHARACTER ECHOED WAS A CARRIAGE RETURN
0208  C21D02  JNZ  ECH10  ; NO - NO NEED TO TAKE SPECIAL ACTION
020B  0E2A  MVI  C,LF  ; YES - WANT TO ECHO LINE FEED, TOO
020D  CDE801  CALL  CO
0210  48  MOV  C,B  ; RESTORE ARGUMENT
0211  C9  RET
;
;**********************************************************************
;  FUNCTION:  ERROR
;  INPUTS:  NONE
;  OUTPUTS:  NONE
;  CALLS:  ECHO,CROUT,GETCM
;  DESTROYS:  A,B,C,F/F'S
;  DESCRIPTION:  ERROR PRINTS THE ERROR CHARACTER (CURRENTLY AN ASTERISK)
;  ON THE CONSOLE, FOLLOWED BY A CARRIAGE RETURN-LINE FEED, AND THEN
;  RETURNS CONTROL TO THE COMMAND RECOGNIZER.
;
0212  0E23  MVI  C,'#'  ; SEND # TO CONSOLE
0214  CDF901  CALL  ECHO
0217  EXIT:
0217  CDF301  CALL  CROUT  ; SKIP TO BEGINNING OF NEXT LINE
021A  C32C88  JMP  GETCM  ; TRY AGAIN FOR ANOTHER COMMAND
;
;**********************************************************************
;  FUNCTION:  RET
INPUTS: NONE
OUTPUTS: CARRY - ALWAYS 0
CALLS: NOTHING
DESTROYS: CARRY
DESCRIPTION: FRET IS JUMPED TO BY ANY ROUTINE THAT WISHES TO
INDICATE FAILURE ON RETURN. FRET SETS THE CARRY
FALSE, DENOTING FAILURE, AND THEN RETURNS TO THE
CALLER OF THE ROUTINE INVOKING FRET.

FRET:

```assembly
021D 37 STC ; FIRST SET CARRY TRUE
021E 3F CXC ; THEN COMPLEMENT IT TO MAKE IT FALSE
021F C9 RET ; RETURN APPROPRIATELY
```

FUNCTION: GETCH
INPUTS: NONE
OUTPUTS: C - NEXT CHARACTER IN INPUT STREAM
CALLS: CI
DESTROYS: A,C,F/F'S
DESCRIPTION: GETCH RETURNS THE NEXT CHARACTER IN THE INPUT STREAM
TO THE CALLING PROGRAM.

GETCH:

```assembly
0220 CDD50 CALL CI ; GET CHARACTER FROM TERMINAL
0223 E67F ANI PRTY0 ; TURN OFF PARITY BIT IN CASE SET BY CONSOLE
0225 4F MOV C,A ; PUT VALUE IN C REGISTER FOR RETURN
0226 C9 RET
```

FUNCTION: GETHX
INPUTS: NONE
OUTPUTS: BC - 16 BIT INTEGER
D - CHARACTER WHICH TERMINATED THE INTEGER
CARRY - 1 IF FIRST CHARACTER NOT DELIMITER
0 IF FIRST CHARACTER IS DELIMITER
CALLS: GETCH,ECHO,VALDL,VALDG,CNVDN,ERROR
DESTROYS: A,B,C,D,E,F/F'S
DESCRIPTION: GETHX ACCEPTS A STRING OF HEX DIGITS FROM THE INPUT
STREAM AND RETURNS THEIR VALUE AS A 16 BIT BINARY
INTEGER. IF MORE THAN 4 HEX DIGITS ARE ENTERED,
ONLY THE LAST 4 ARE USED. THE NUMBER TERMINATES WHEN
A VALID DELIMITER IS ENCOUNTERED. THE DELIMITER IS
ALSO RETURNED AS AN OUTPUT OF THE FUNCTION. ILLEGAL
CHARACTERS (NOT HEX DIGITS OR DELIMITERS) CAUSE AN
ERROR INDICATION. IF THE FIRST (VALID) CHARACTER
ENCOUNTERED IN THE INPUT STREAM IS NOT A DELIMITER,
GETHX WILL RETURN WITH THE CARRY BIT SET TO 1;
OTHERWISE, THE CARRY BIT IS SET TO 0 AND THE CONTENTS
OF BC ARE UNDEFINED.

GETHX:

0227  E5       PUSH H ; SAVE HL
0228  21E002   LXI H,0 ; INITIALIZE RESULT
022B  1E00     MVI E,B ; INITIALIZE DIGIT FLAG TO FALSE
022D  GHX05:
022E  CD2A02   CALL GETCH ; GET A CHARACTER
0230  CDF602   CALL ECHO ; ECHO THE CHARACTER
0233  CD2303   CALL VALID ; SEE IF DELIMITER
1      +       FALSE GHX10 ; NO - BRANCH
0236  024502   JNC GHX10
0239  51       MOV D,C ; YES - ALL DONE, BUT WANT TO RETURN DELIMITER
023A  E5       PUSH H
023B  C1       POP B ; MOVE RESULT TO BC
023C  E1       POP H ; RESTORE HL
023D  7B       MOV A,E ; GET FLAG
023E  B7       ORA A ; SET F/F'S
023F  C23B03   JNZ SRET ; IF FLAG NON-0, A NUMBER HAS BEEN FOUND
0242  CASD02   JZ FRET ; ELSE, DELIMITER WAS FIRST CHARACTER
0245  GHX10:
0246  CD6703   CALL VALID ; IF NOT DELIMITER, SEE IF DIGIT
1      +       FALSE ERROR ; ERROR IF NOT A VALID DIGIT, EITHER
0249  021202   JNC ERROR
024B  CEDD01   CALL CONV ; CONVERT DIGIT TO ITS BINARY VALUE
024E  1EFF     MVI E,OFFH ; SET DIGIT FLAG NON-0
0250  29       LAD H ; *2
0251  29       DAD H ; *4
0252  29       DAD H ; *8
0253  29       DAD H ; *16
0254  B630     MVI B,0 ; CLEAR UPPER 8 BITS OF BC PAIR
0256  4F       MOV C,A ; BINARY VALUE OF CHARACTER INTO C
0257  09       DAD B ; ADD THIS VALUE TO PARTIAL RESULT
0258  C32D02   JMP GHX05 ; GET NEXT CHARACTER

FUNCTION: GETNM
INPUTS: C - COUNT OF NUMBERS TO FIND IN INPUT STREAM
OUTPUTS: TOP OF STACK - NUMBERS FOUND IN REVERSE ORDER (LAST ON TOP OF STACK)
CALLS: GETHX,HILO,ERROR
DESTROYS: A,B,C,D,E,H,L,F/F'S
DESCRIPTION: GETNM FINDS A SPECIFIED COUNT OF NUMBERS, BETWEEN 1 AND 3, INCLUSIVE, IN THE INPUT
; STREAM AND RETURNS THEIR VALUES ON THE STACK. IF 2
; OR MORE NUMBERS ARE REQUESTED, THEN THE FIRST MUST BE
; LESS THAN OR EQUAL TO THE SECOND, OR THE FIRST AND
; SECOND NUMBERS WILL BE SET EQUAL. THE LAST NUMBER
; REQUESTED MUST BE TERMINATED BY A CARRIAGE RETURN
; OR AN ERROR INDICATION WILL RESULT.

025B 2E03 MVI L,3 ; PUT MAXIMUM ARGUMENT COUNT INTO L
025D 79 MOV A,C ; GET THE ACTUAL ARGUMENT COUNT
025E E603 ANI 3 ; FORCE TO MAXIMUM OF 3
0260 C8 RZ ; IF 0, DON'T EITHER TO DO ANYTHING
0261 67 MOV H,A ; ELSE, PUT ACTUAL COUNT INTO H
0262 GNM05: ;
0262 CD2762 CALL GETNX ; GET A NUMBER FROM INPUT STREAM
1 1 D21202 0853 1 + FALSE ERROR ; ERROR IF NOT THERE - TOO FEW NUMBERS
0265 1 D21202 0853 1 JNC ERROR
0268 C5 PUSH B ; ELSE, SAVE NUMBER ON STACK
0269 2D DCR L ; DECREMENT MAXIMUM ARGUMENT COUNT
026A 25 DCR H ; DECREMENT ACTUAL ARGUMENT COUNT
026B CA762 J2 GNM10 ; BRANCH IF NO MORE NUMBERS WANTED
026D 7A MOV A,D ; ELSE, GET NUMBER TERMINATOR TO A
026F FE0D CPI CR ; SEE IF CARRIAGE RETURN
0271 CA1202 J2 ERROR ; ERROR IF SO - TOO FEW NUMBERS
0274 C36202 JMP GNM05 ; ELSE, PROCESS NEXT NUMBER
0277 GNM10: ;
0277 7A MOV A,D ; WHEN COUNT 0, CHECK LAST TERMINATOR
0278 FE0D CPI CR
027A C21202 JNZ ERROR ; ERROR IF NOT CARRIAGE RETURN
027D 01FFFF LXI B,FFFFFH ; HL GETS LARGEST NUMBER
0280 7D MOV A,L ; GET WHAT'S LEFT OF MAXIMUM ARG COUNT
0281 B7 GA A ; CHECK FOR 0
0282 CA762 J2 GNM20 ; IF YES, 3 NUMBERS WERE INPUT
0285 GNM15: ;
0285 C5 PUSH B ; IF NOT, FILL REMAINING ARGUMENTS WITH 0FFFFH
0286 2D DCR L
0287 C28502 JNZ GNM15
028A GNM20: ;
028A C1 POP B ; GET THE 3 ARGUMENTS OUT
028B E1 POP D
028C E1 POP H
028D CA762 CALL HILO ; SEE IF FIRST >= SECOND
1 1 D29502 0853 1 + FALSE GNM25 ; NO - BRANCH
0290 1 D29502 0853 1 JNC GNM25
0293 54 MOV D,H ; YES - MAKE SECOND EQUAL TO THE FIRST
0294 5D MOV E,L
0295 GNM25: ;
0295 E3 XTHL ; PUT FIRST ON STACK - GET RETURN ADDR
0296 E5 PUSH D ; PUT SECOND ON STACK
0297 E5 PUSH B ; PUT THIRD ON STACK
0298 E5 PUSH H ; PUT RETURN ADDRESS ON STACK
FUNCTION: GNM30

INPUTS:
- DE: 16-bit integer
- HL: 16-bit integer

OUTPUTS:
- CARRY: 0 if HL < DE, 1 if HL >= DE

CALLS: NONE

DESTROY: A, F/P'S

DESCRIPTION: GNM30 decrements the residual count by 1 if negative, or replaces the top result with the return address.

;*****************************************************************

FUNCTION: HILO

INPUTS:
- DE: 16-bit integer
- HL: 16-bit integer

OUTPUTS:
- CARRY: 0 if HL < DE, 1 if HL >= DE

CALLS: NOTHING

DESTROY: A, F/P'S

DESCRIPTION: HILO compares the two 16-bit integers in HL and DE. The integers are treated as unsigned numbers. The carry bit is set according to the result of the comparison.

;*****************************************************************

FUNCTION: INUST

INPUTS: NONE

OUTPUTS: NOTHING

CALLS: NOTHING

DESTROY: A, H/L, SP

DESCRIPTION: INUST outputs to the USART the command word.
; AND initializes the stack pointer.

; INUST:

0282 3E25 MVI A, CMD
0284 03 ED OUT CNCTL ; output command word to USART
0286 21 021C LXI H, MSTAK-44 ; load pointer to stack
0289 22 38 4C SHLD SSAVE ; initialize user stack pointer
028C 31 2E 4C LXI SP, MSTAK ; initialize monitor stack
028F C3 1E 0F JMP SMSG ; go to print signon message

;******************************************************************************

; FUNCTION: NMXOUT
; INPUTS: A - 8 BIT INTEGER
; OUTPUTS: NONE
; CALLS: ECHO, PRVAL
; DESTROYS: A, B, C, F/F's
; DESCRIPTION: NMXOUT CONVERTS THE 8 BIT, UNSIGNED INTEGER IN THE
; A REGISTER INTO 2 ASCII CHARACTERS. THE ASCII CHARACTERS
; ARE THE ONES REPRESENTING THE 8 BITS. THESE TWO
; CHARACTERS ARE SENT TO THE CONSOLE AT THE CURRENT PRINT
; POSITION OF THE CONSOLE.

; NMXOUT:

02C2 F3 PUSH PSW ; save argument
02C3 0F RRC
02C4 0F RRC
02C5 0F RRC
02C6 0F RRC
02C7 CDD5 02 CALL PRVAL ; convert lower 4 bits to ASCII
02CA CDF9 01 CALL ECHO ; send to terminal
02CD F1 POP PSW ; get back argument
02CE CDD5 02 CALL PRVAL
02CF CDF9 01 CALL ECHO
02D4 C9 RET

;******************************************************************************

; FUNCTION: PRVAL
; INPUTS: A - INTEGER, RANGE 0 TO 9
; OUTPUTS: A - ASCII CHARACTER
; CALLS: NOTHING
; DESTROYS: NOTHING
; DESCRIPTION: PRVAL CONVERTS A NUMBER IN THE RANGE 0 TO 9 HEX TO
; THE CORRESPONDING ASCII CHARACTER, 0-9, A-F. PRVAL
; DOES NOT CHECK THE validity of its input argument.
IN 8086 MACRO ASSEMBLER, VER 2.4  
ERRORS = 0  PAGE 22  
68/10 MONITOR, VERSION 1.1, 1 NOVEMBER 1976

02D5 PRVAL:
02D6 660F  ANI  HCHAR  ; MASK OUT UPPER 4 BITS - WANT 1 HEX CHAR
02D7 CE92  ADI  90H  ; SET UP A SO THAT A-F CAUSE A CARRY
02D8 27  DAA  ; ADJUST CONTENTS OF A REGISTER
02D9 CE40  ACI  40H  ; ADD CARRY AND ADJUST UPPER 4 BITS
02DA 27  DAA  ; ADJUST CONTENTS OF A REGISTER AGAIN
02DB 4F  MOVC A,C  ; MOVE ASCII CHARACTER TO C
02DE C9  RET  ; ALL DONE

;*****************************************************
; FUNCTION: REGDS
; INPUTS: NONE
; OUTPUTS: NONE
; CALLS: ECHO, NMOUT, ERROR, CROUT
; DESTROYS: A,B,C,D,E,H,L,F/F'S
; DESCRIPTION: REGDS DISPLAYS THE CONTENTS OF THE REGISTER SAVE
; LOCATIONS, IN FORMATTED FORM, ON THE CONSOLE. THE
; DISPLAY IS DRIVEN FROM A TABLE, RTAB, WHICH CONTAINS
; THE REGISTER'S PRINT SYMBOL, SAVE LOCATION ADDRESS,
; AND LENGTH (8 OR 16 BITS).
;  REGDS:
02DF 21C003  LXI  H,RTAB  ; LOAD HL WITH ADDRESS OF START OF TABLE
02E0 02E0  REG0:
02E1 02E0  MOV  C,M  ; GET PRINT SYMBOL OF REGISTER
02E2 79  MOV  A,C
02E3 DE 87  CFA  A  ; TEST FOR 0 - END OF TABLE
02E4 CE40  XHI  REG10  ; IF NOT END, BRANCH
02E5 CDF30  CALL  CROUT  ; ELSE, CARRIAGE RETURN/LINE FEED TO END
02E6 C9  RET  ; /DISPLAY
02E7 REG10:
02E8 CDF901  CALL  ECHO  ; ECHO CHARACTER
02E9 83D0  NVI  C,'.'
02EA CDF901  CALL  ECHO  ; OUTPUT EQUALS SIGN, I.E. A=
02EB 23  INX  H  ; POINT TO START OF SAVE LOCATION ADDRESS
02EC 5E  MOV  E,K  ; GET MSP OF SAVE LOCATION ADDRESS TO E
02ED 18C0  NVI  D,DATA 8  ; PUT MSP OF SAVE LOC ADDRESS INTO D
02EE 23  INX  H  ; POINT TO LENGTH FLAG
02EF 1A  LDAX  D  ; GET CONTENTS OF SAVE ADDRESS
02F0 CDC202  CALL  NMOUT  ; DISPLAY ON CONSOLE
02F1 7E  MOVC A,M  ; GET LENGTH FLAG
02F2 B7  CFA  A  ; SET SIGN F/P
02F3 CA0703  JZ  REG15  ; IF 0, REGISTER IS 8 BITS
02F4 1B  DCX  D  ; ELSE, 16 BIT REGISTER SO MORE TO DISPLAY
02F5 1A  LDAX  D  ; GET LOWER 8 BITS
02F6 CDC202  CALL  NMOUT  ; DISPLAY THEM
02F7 0E20  NVI  C,'.
FUNCTION: RGADR
INPUTS: C - CHARACTER DENOTING REGISTER
OUTPUTS: EC - ADDRESS OF ENTRY IN RTAB CORRESPONDING TO REGISTER
CALLS: ERROR
DESTROYS: A,B,C,D,E,H,L,F/F'S
DESCRIPTION: RGADR TAKES A SINGLE CHARACTER AS INPUT. THIS CHARACTER
DELCNES A REGISTER. RGADR SEARCHES THE TABLE RTAB
FOR A MATCH ON THE INPUT ARGUMENT. IF ONE OCCURS,
RGADR RETURNS THE ADDRESS OF THE ADDRESS OF THE
SAVE LOCATION CORRESPONDING TO THE REGISTER. THIS
ADDRESS POINTS INTO RTAB. IF NO MATCH OCCURS, THEN
THE REGISTER IDENTIFIER IS ILLEGAL AND CONTROL IS
PASSED TO THE ERROR ROUTINE.

FUNCTION: RSTTF
INPUTS: NONE
OUTPUTS: NONE
CALLS: NOTHING
DESTROYS: A,B,C,D,E,H,L,F/F'S
DESCRIPTION: RSTTF RESTORES ALL CPU REGISTER, FLIP/FLOPS, STACK
POINTER AND PROGRAM COUNTER FROM THEIR RESPECTIVE
SAVE LOCATIONS IN MEMORY. THE ROUTINE THEN TRANSFERS
CONTROL TO THE LOCATION SPECIFIED BY THE PROGRAM COUNTER (I.E. THE RESTORED VALUE). THE ROUTINE EXITS WITH THE INTERRUPTS ENABLED.

RSTTF:

DI ; DISABLE INTERRUPTS WHILE RESTORING THINGS
LXI SP,MSTAK ; SET MONITOR STACK POINTER TO START
; /OF STACK
POP D ; START ALSO END OF REGISTER SAVE AREA
POP B
POP PSW
LHLD SSAVE ; RESTORE USER STACK POINTER
SPHL
LHLD PSAVE
PUSH H ; PUT USER RETURN ADDRESS ON USER STACK
PUSH LSAVE ; RESTORE HL REGISTERS
EI ; ENABLE INTERRUPTS NOW
RET ; JUMP TO RESTORED PC LOCATION

FUNCTION: SRET
INPUTS: NONE
OUTPUTS: CARRY = 1
CALLS: NOTHING
DESTROYS: CARRY
DESCRIPTION: SRET IS JUMPED TO BY ROUTINES WISHING TO RETURN SUCCESS.
SRET SETS THE CARRY TRUE AND THEN RETURNS TO THE CALLER OF THE ROUTINE INVOKING SRET.

SRET:

STC ; SET CARRY TRUE
RET ; RETURN APPROPRIATELY

FUNCTION: STHF0
INPUTS: DE - 16 BIT ADDRESS OF BYTE TO BE STORED INTO
OUTPUTS: NONE
CALLS: NOTHING
DESTROYS: A,B,C,H,L,E/F/S
DESCRIPTION: STHF0 CHECKS THE HALF BYTE FLAG IN TEMP TO SEE IF IT IS SET TO LOWER. IF SO, STHF0 STORES A 0 TO PAD OUT THE LOWER HALF OF THE ADDRESSED BYTE;
OTHERWISE, THE ROUTINE TAKES NO ACTION.

STHF0:
**FUNCTION: STHLF**

**INPUTS:**
- C - 4 BIT VALUE TO BE STORED IN HALF BYTE
- DE - 16 BIT ADDRESS OF BYTE TO BE STORED INTO

**OUTPUTS:**
- NONE

**CALLS:**
- NOTHING

**DESTROYS:**
- A, B, C, H, L, F/F'S

**DESCRIPTION:**
- STHLF TAKES THE 4 BIT VALUE IN C AND STORES IT IN HALF OF THE BYTE ADDRESSED BY REGISTERS DE. THE HALF BYTE USED (EITHER UPPER OR LOWER) IS DENOTED BY THE VALUE OF THE FLAG IN TEMP. STHLF ASSUMES THAT THIS FLAG HAS BEEN PREVIOUSLY SET (NOMINALLY BY ICMD).

---

**STHLF:**

```
033D  3A3A3C  LDA TEMP   GET HALF BYTE FLAG
0340  B7    ORA  A      SET F/F'S
0341  C0    RNZ        IF SET TO UPPER, DON'T DO ANYTHING
0342  CE00  MVI C,C     ELSE, WANT TO STORE THE VALUE 0
0344  CD403  CALL STHLF; DO IT
0347  C9    RET

;*************************************************************************

;*************************************************************************

; FUNCTION: STHLF

; INPUTS: C - 4 BIT VALUE TO BE STORED IN HALF BYTE
; DE - 16 BIT ADDRESS OF BYTE TO BE STORED INTO
; OUTPUTS: NONE
; CALLS: NOTHING
; DESTROYS: A,B,C,H,L,F/F'S
; DESCRIPTION: STHLF TAKES THE 4 BIT VALUE IN C AND STORES IT IN HALF OF THE BYTE ADDRESSED BY REGISTERS DE. THE HALF BYTE USED (EITHER UPPER OR LOWER) IS DENOTED BY THE VALUE OF THE FLAG IN TEMP. STHLF ASSUMES THAT THIS FLAG HAS BEEN PREVIOUSLY SET (NOMINALLY BY ICMD).

; STHLF:

0348  D5    PUSH D      MOVE ADDRESS OF BYTE INTO HL
0349  E1    POP  H
034A  79    MOV A,C     GET VALUE
034B  E60F  ANI  $PH   FORCE TO 4 BIT LENGTH
034D  4F    MOV C,A     PUT VALUE BACK
034E  3A3A3C  LDA TEMP   GET HALF BYTE FLAG
0351  E7    ORA A       CHECK FOR LOWER HALF
0352  C25B33  JNZ STH05; BRANCH IF NOT
0355  7E    MOV A,M     ELSE, GET BYTE
0356  EE0C  ANI  $PH; CLEAR LOWER 4 BITS
0358  B1    ORA C       OR IN VALUE
0359  77    MOV M,A     PUT BYTE BACK
035A  C9    RET

; STH05:

035B  7E    MOV A,M     IF UPPER HALF, GET BYTE
035C  E60F  ANI  $PH   CLEAR UPPER 4 BITS
035E  47    MOV B,A     SAVE BYTE IN B
035F  79    MOV A,C     GET VALUE
0360  0F    RRC
0361  0F    RRC
0362  0F    RRC
0363  0F    RRC; ALIGN TO UPPER 4 BITS
0364  B0    ORA B      OR IN ORIGINAL LOWER 4 BITS
0365  77    MOV M,A     PUT NEW CONFIGURATION BACK
0366  C9    RET
```
FUNCTION: VALDG
INPUTS: C - ASCII CHARACTER
OUTPUTS: CARRY - 1 IF CHARACTER REPRESENTS VALID HEX DIGIT
- 0 OTHERWISE
CALLS: NOTHING
DESTROYS: A,F/F'S
DESCRIPTION: VALDG RETURNS SUCCESS IF ITS INPUT ARGUMENT IS
AN ASCII CHARACTER REPRESENTING A VALID HEX DIGIT
(0-9,A-F), AND FAILURE OTHERWISE.

VALDG:

0367 79 MOV A,C
0368 FE30 CPI '0'
0369 FA102 JM FRET ; TEST CHARACTER AGAINST '0'
036A FE39 CPI '9' ; IF ASCII CODE LESS, CANNOT BE VALID DIGIT
036B FA383 JM SRET ; ELSE, SEE IF IN RANGE '0'-9'
036C FA383 JZ SRET ; CODE BETWEEN '0' AND '9'
036D FE47 CPI 'A' ; NOT A DIGIT - TRY FOR A LETTER
036E FA102 JM SRET ; NO - CODE BETWEEN '9' AND 'A'
0370 FE47 CPI 'G' ; ONE DIGIT - TRY FOR A LETTER
0371 F2102 JP SRET ; NO - CODE GREATER THAN 'F'
0372 C33B3 JMP SRET ; OKAY - CODE IS 'A' TO 'F', INCLUSIVE

FUNCTION: VALDL
INPUTS: C - CHARACTERS
OUTPUTS: CARRY - 1 IF INPUT ARGUMENT VALID DELIMITER
- 0 OTHERWISE
CALLS: NOTHING
DESTROYS: A,F/F'S
DESCRIPTION: VALDL RETURNS SUCCESS IF ITS INPUT ARGUMENT IS A VALID
DELIMITER CHARACTER (SPACE, COMMA, CARRIAGE RETURN) AND
FAILURE OTHERWISE.

VALDL:

0382 79 MOV A,C
0383 FE2C CPI ',' ; CHECK FOR COMMA
0384 CA3B3 JZ SRET ; CHECK FOR COMMA
0385 FE2D CPI CR ; CHECK FOR CARRIAGE RETURN
0386 CA3B3 JZ SRET ; CHECK FOR CARRIAGE RETURN
0387 FE2A CPI ' ' ; CHECK FOR SPACE
0388 CA3B3 JZ SRET ; CHECK FOR SPACE
0389 C31D02 JMP SRET ; ERROR IF NONE OF THE ABOVE
;**********************************************************
;
;MONITOR TABLES
;
;
0395 SGNON: ; SIGNON MESSAGE
0395 0D1A3830 DB CR,LF,'80/10 MONITOR',CR,LF
0399 2F313020 ...(Continues)
039D 414F4E49
03A1 544F520D
03A5 0A
0311 LSGNON EQU $-SGNON ; LENGTH OF SIGNON MESSAGE
0346 CADR: ; TABLE OF ADDRESSES OF COMMAND ROUTINES
0346 006D DW 0 ; DUMMY
0350 3301 DW XCMD
0354 0000 DW SCMD
0358 0000 DW MCMD
0362 5400 DW ICMD
0366 6400 DW GCMD
036A 5400 DW DCMD
0374 0004 DW RCMD
0378 4404 DW WCMD
0382 CTAB: ; TABLE OF VALID COMMAND CHARACTERS
0382 57 DB 'X'
0386 52 DB 'R'
038A 44 DB 'D'
038E 49 DB 'I'
0392 40 DB 'Y'
0396 53 DB 'S'
039A 58 DB 'X'
039E NCMDS EQU $-CTAB ; NUMBER OF VALID COMMANDS
;
03C0 RTAB: ; TABLE OF REGISTER INFORMATION
03C0 41 DB 'A' ; REGISTER IDENTIFIER
03C1 33 DB ASAVERE AND OFFH ; ADDRESS OF REGISTER SAVE LOCATION
03C2 00 DB 0 ; LENGTH FLAG - 0=8 BITS, 1=16 BITS
03C3 RTAB2 EQU $-RTAB ; SIZE OF AN ENTRY IN THIS TABLE
03C3 42 DB 'B'
03C4 31 DB BSAAVE AND OFFH
03C5 08 DB 0
03C6 43 DB 'C'
03C7 38 DB CSAVERE AND OFFH
ORG BRTAB

CPUKT:

CPYRT:

DB (C) 1976 INTEL CORP

ORG BRTAB

JMP CO ; BRANCH TABLE FOR USER ACCESSIBLE ROUTINES
JMP CI
JMP RI
JMP PO
FUNCTION RCMD
; INPUTS: NONE
; OUTPUTS: NONE
; CALLS: GETCH, ECHO, CO, RICH, BYTE
; DESTRYS: A, D, E, H, L, F
; DESCRIPTION: RCMD IMPLEMENTS THE READ HEXADECIMAL TAPE (R) COMMAND.

FUNCTION WCMD
; INPUTS: NONE
; OUTPUTS: NONE
; CALLS: GETCH, LEAD, PO, PBYTE, PADR, PEOL, POEOL
DESTROYS: A, B, C, D, E, H, L, P/F'S

DESCRIPTION: WCMD IMPLEMENTS THE WRITE HEXADECIMAL TAPE (W) COMMAND.

WCMD:

WCM9:

WCM10:

WCM15:

WCM20:

WCM25:
FUNCTION BYTE
INPUTS: D - CURRENT VALUE OF CHECKSUM
OUTPUTS: A - HEXADECIMAL CHARACTER
D - UPDATED VALUE OF CHECKSUM
CALLS: RICH,CNVBN
DESTROYS: A,B,C,D,F/F'S
DESCRIPTION: BYTE READS 2 ASCII CHARACTERS FROM THE TELETYPewriter
AND CONVERTS THE CHARACTERS TO ONE HEXADECIMAL CHARACTER.
THE A REGISTER CONTAINS THE FINAL CHARACTER AND THE
D REGISTER CONTAINS THE UPDATED VALUE OF
THE CHECKSUM.

BYTE:

0496  C5       PUSH B       ; SAVE BC
0497  CD1305   CALL RICH    ; READ ASCII CHARACTER FROM TAPE
0498  4F       MOV C,A     ; GET ANOTHER CHARACTER FROM TAPE
0499  CDDP01   CALL CNVBN   ; CONVERT IT
049A  4F       MOV C,A     ; INCREMENT CHECKSUM
049B  CD1305   CALL RICH    ; READ ASCII CHARACTER FROM TAPE
049C  4F       MOV C,A     ; OR IN THE UPPER 4 BITS
049D  B0       GPA B       ; SAVE
049E  4F       MOV C,A     ; RESTORE HEX DATA TO A REGISTER
049F  CDDP01   CALL CNVBN   ; CONVERT IT
04A0  4F       MOV C,A     ; OR IN THE UPPER 4 BITS
04A1  4F       MOV C,A     ; SAVE
04A2  4F       MOV C,A     ; INCREMENT CHECKSUM
04A3  CD1305   CALL RICH    ; READ ASCII CHARACTER FROM TAPE
04A4  4F       MOV C,A     ; OR IN THE UPPER 4 BITS
04A5  4F       MOV C,A     ; RESTORE HEX DATA TO A REGISTER
04A6  C9       RET

FUNCTION DELAY
INPUTS: NONE
OUTPUTS: NONE
CALLS: NOTHING
DESTROYS: F/F'S

DESCRIPTION: DELAY PROVIDES A PROGRAMMED DELAY OF 1 MILLISECOND FOR TAPE READER OPERATION.

DELAY:

DELAY:

04B1 04B2 04B3 04B4
C5 8683 DEL:
04B4 05 04B5 04B8 04B9
DECR B PDP B RET
MVI B,ONEMS LOAD 1 MILLISECOND CONSTANT
DECAMENT INNER COUNTER
RESTORE BC REGISTERS
RETURN TO CALLING RUTINE

FUNCTION LEAD
INPUTS: NONE
OUTPUTS: ONE
CALLS: PO
DESTROYS: B,C,F/F'S
DESCRIPTION: LEAD OUTPUTS 60 NULL CHARACTERS TO PAPER TAPE TO FORM A LEADER.

LEAD:

LEAD:

04C6 04C7 04CA
7C C2BC04
MOO A,L ; PUNCH SECOND HALF OF ADDRESS
CALL PBYTE
MOV A,H ; PUNCH FIRST HALF OF ADDRESS

FUNCTION PADR
INPUTS: HL - ADDRESS TO BE PUNCHED
OUTPUTS: NONE
CALLS: PBYTE
DESTROYS: A
DESCRIPTION: PADR PUNCHES ON THE TELETYPewriter THE ADDRESS CONTAINED IN THE H,L REGISTERS.
FUNCTION PBYTE
; INPUTS: A - CHARACTER TO BE PUNCHED
; D - CURRENT VALUE OF CHECKSUM
; OUTPUTS: D - UPDATED VALUE OF CHECKSUM
; CALLS: PRVAL, PO
; DESTROYS: A,F/F'S
; DESCRIPTION: PBYTE CONVERTS THE HEXADECIMAL VALUE IN THE A REGISTER
; INTO TWO ASCII CHARACTERS AND PUNCHES THESE CHARACTERS
; ON PAPER TAPE. THE CHECKSUM CONTAINED IN D IS UPDATED.

PBYTE:

04CF  F5 PUSH  PSW ; SAVE A,F/F'S
04C6  0F RRC
04C5  0F RRC
04C4  0F RRC
04C3  0F RRC
04C2  CDD502 CALL PRVAL ; CONVERT UPPER 4 BITS JUST ROTATED TO ASCII
04C1  CDDF05 CALL PO ; PUNCH CHARACTER
04C0  F1 POP  PSW ; RESTORE A,F/F'S
04CF  F5 PUSH  PSW ; SAVE A AGAIN
04C4  CDD502 CALL PRVAL ; CONVERT LOWER 4 BITS TO ASCII CHARACTER
04C3  CDDF05 CALL PO ; PUNCH CHARACTER
04C2  F1 POP  PSW ; RESTORE A
04C1  $2 ADC  D ; ADD VALUE TO CHECKSUM
04C0  57 MOV  D,A ; UPDATE D REGISTER WITH NEW CHECKSUM
04E5  C9 RET ; RETURN TO CALLING ROUTINE

FUNCTION PEOF
; INPUTS: NONE
; OUTPUTS: NONE
; CALLS: PO,PBYTE,PAE,LEAD
; DESTROYS: A,C,D,H,L,F/F'S
; DESCRIPTION: PEOF PUNCHES THE END OF FILE RECORD CONSISTING OF A RECORD
; MARK, A LOAD ADDRESS OF 0, THE RECORD TYPE, AND THE
; RECORD CHECKSUM.

PEOF:

04C6  0E3A MVI  C,'p'
04E5  CDDF05 CALL  PO ; PUNCH RECORD MARK
04ED  4F XRA  A ; ZERO CHECKSUM
MOV D,A ; SAVE IN D REGISTER
CALL PBYTE ; PUNCH RECORD LENGTH
CALL PBR ; PUNCH IT
MVI A,1 ; LOAD A WITH RECORD TYPE
CALL PBYTE ; PUNCH IT
XRA A ; ZERO A
SUB D ; COMPUTE CHECKSUM
CALL PBYTE ; PUNCH IT
CALL LEAD ; PUNCH TRAILER

FUNCTION PEOL
INPUTS: NONE
OUTPUTS: NONE
CALLS: PO
DESTROYS: C
DESCRIPTION: PEOL PUNCHES A CARRIAGE RETURN AND LINE FEED ONTO PAPER TAPE.

PEOL:
MVI C,CR ; PUNCH CARRIAGE RETURN CHARACTER
CALL PO
MVI C,LF ; PUNCH LINE FEED CHARACTER
CALL PO
RET

FUNCTION PO
INPUTS: C - CHARACTER TO BE PUNCHED
OUTPUTS: NONE
CALLS: CO
DESTROYS: NOTHING
DESCRIPTION: PO PUNCHES THE CHARACTER SUPPLIED IN THE C REGISTER TO THE USER TELETYPETRITER.

PO:
CALL CO ; CALL CONSOLE OUT TO PERFORM CHARACTER OUTPUT
RET
; FUNCTION RICH
; INPUTS: NONE
; OUTPUTS: A - ZERO, CARRY - 1 IF END OF FILE
; A - CHARACTER, CARRY - 0 IF VALID CHARACTER
; CALLS: RI
; DESTROYS: A,F/F'S
; DESCRIPTION: RICH TESTS FOR AN END OF FILE CONDITION.

0513   CD1C95  CALL RI  ; READ A CHARACTER FROM TAPE
0516   DA1282  JC ERROR ; JUMP IF READER TIMEOUT ERROR
0519   E67F  ANI PRTY0 ; REMOVE PARITY BIT
051D   C9  RET  ; RETURN TO CALLING ROUTINE

;************************************************************************************

; FUNCTION RI
; INPUTS: NONE
; OUTPUTS: A - ZERO, CARRY - 1 IF END OF FILE
; A - CHARACTER, CARRY - 0 IF VALID CHARACTER
; CALLS: DELAY
; DESTROYS: A,F/F'S
; DESCRIPTION: RI READS A CHARACTER FROM THE TTY TAPE READER.

051C   C5  PUSH B  ; SAVE BC
051D   DBED  IN CXCTL  ; READ IN USART STATUS
051F   E624  ANI TXBE  ; CHECK FOR TRANSMITTER BUFFER EMPTY
0521   CA1D0F  JZ RI25 ; TRY AGAIN IF NOT EMPTY
0524   3E27  MVI A,TTYADV ; ADVANCE THE TAPE
0526   D3ED  OUT CXCTL  ; OUTPUT THE ADVANCE COMMAND
0528   0626  MVI B,40  ; INITIALIZE TIMER FOR 40 MS.
052A   RI25:  IN CXCTL  ; READ IN USART STATUS
052D   E5  DCR B  ; DECREMENT TIMER
052E   C22805  JNZ RI07 ; JUMP IF TIMER NOT EXPIRED
0531   3E25  MVI A,CND ; STOP THE READER ADVANCE
0533   D3ED  OUT CXCTL  ; OUTPUT STOP COMMAND
0535   06FA  MVI B,250 ; INITIALIZE TIMER FOR 250 MS.
0537   RI07:  CALL DELAY ; DELAY FOR 1 MILLISECONDS
053D   E5  DCR B  ; DECREMENT TIMER
053E   C22A05  JNZ RI10 ; JUMP IF TIMER NOT EXPIRED
0541   05  DCR B  ; DECREMENT TIMER
0542   C23705  JNZ RI15 ; JUMP IF TIMER NOT EXPIRED
0545   AF  XRA A  ; ZERO A
0546   37  STC  ; SET CARRY INDICATING EOF
8080 MACRO ASSEMBLER, VER 2.4  ERRORS = 0 PAGE 36
80/10 MONITOR, VERSION 1.1, 1 NOVEMBER 1976

0547  C1   POP   B   ; RESTORE BC
0548  C9   RET   ; RETURN TO CALLING ROUTINE
0549  RI15:
0549  DBEC   IN   CNIN   ; INPUT DATA CHARACTER
054B  B7   ORA   A   ; CLEAR CARRY
054C  C1   POP   B   ; RESTORE BC
054D  C9   RET   ; RETURN TO CALLING ROUTINE

; ***************************************************************
; COPYRT:
054E  DB   ' (C) 1976 INTEL CORP'
0552  31393136
0556  28494E54
055A  45424043
055E  4F5250

; ***************************************************************

E-37

3C20  ORG  DATA  ; ORG TO REGISTER SAVE - STACK GOES IN HERE
3C2E  ORG  REGS  ; ORG TO REGISTER SAVE - STACK GOES IN HERE

3C2E  MSTAK  EQU  $  ; START OF MONITOR STACK
3C2F  00  ESAVE:  DB  0  ; E REGISTER SAVE LOCATION
3C30  00  ESAVE:  DB  0  ; D REGISTER SAVE LOCATION
3C31  03  ESAVE:  DB  0  ; C REGISTER SAVE LOCATION
3C32  03  ESAVE:  DB  0  ; B REGISTER SAVE LOCATION
3C33  00  ESAVE:  DB  0  ; FLAGS SAVE LOCATION
3C34  00  ESAVE:  DB  0  ; A REGISTER SAVE LOCATION
3C35  00  ESAVE:  DB  0  ; L REGISTER SAVE LOCATION
3C36  00  ESAVE:  DB  0  ; H REGISTER SAVE LOCATION
3C38  0000  ESAVE:  DA  0  ; PGM COUNTER SAVE LOCATION
3C3A  00  TEMP:  DB  0  ; TEMPORARY MONITOR CELL
3C3D  ORG  BRLOC  ; ORG TO USER BRANCH LOCATION
3C3D  USRBR:  DS  3  ; BRANCH GOES IN HERE

NO PROGRAM ERRORS

8080 MACRO ASSEMBLER, VER 2.4  ERRORS = 0 PAGE 37
80/10 MONITOR, VERSION 1.1, 1 NOVEMBER 1976
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Address</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0007</td>
<td>ADRD 01A8</td>
</tr>
<tr>
<td>B</td>
<td>00020</td>
<td>BRCHR 001B</td>
</tr>
<tr>
<td>BPTAB</td>
<td>03FA</td>
<td>BRAVE 3C31</td>
</tr>
<tr>
<td>CADR</td>
<td>03A6</td>
<td>CJ 01A5</td>
</tr>
<tr>
<td>CNTR</td>
<td>03CC</td>
<td>CHOUT 00C2</td>
</tr>
<tr>
<td>CONST</td>
<td>03ED</td>
<td>CPYR 054E</td>
</tr>
<tr>
<td>CPUET</td>
<td>01E3</td>
<td>CSAVE 3C33</td>
</tr>
<tr>
<td>COUT</td>
<td>01F3</td>
<td>CTAD 0336</td>
</tr>
<tr>
<td>DATA</td>
<td>3C20</td>
<td>DCALL 026C</td>
</tr>
<tr>
<td>DEL1</td>
<td>04B4</td>
<td>DSAVE 032F</td>
</tr>
<tr>
<td>ECH05</td>
<td>0202</td>
<td>ECHO 01F9</td>
</tr>
<tr>
<td>EDAVE</td>
<td>03E2</td>
<td>EXIT 0327</td>
</tr>
<tr>
<td>PEC</td>
<td>021D</td>
<td>FSAVE 0332</td>
</tr>
<tr>
<td>GCH</td>
<td>0255</td>
<td>GETCH 0226</td>
</tr>
<tr>
<td>GLTX</td>
<td>025B</td>
<td>GIX05 022D</td>
</tr>
<tr>
<td>GNM10</td>
<td>0277</td>
<td>GNM20 028A</td>
</tr>
<tr>
<td>GCM5</td>
<td>0299</td>
<td>GH 0008</td>
</tr>
<tr>
<td>GCT10</td>
<td>0355</td>
<td>HCHAR 020F</td>
</tr>
<tr>
<td>HILO</td>
<td>02A0</td>
<td>ICXS 0234</td>
</tr>
<tr>
<td>ICM10</td>
<td>0003</td>
<td>ICMX 0019</td>
</tr>
<tr>
<td>INVAT</td>
<td>00FF</td>
<td>L 0035</td>
</tr>
<tr>
<td>LF</td>
<td>010A</td>
<td>LSAVE 3C34</td>
</tr>
<tr>
<td>KCM75</td>
<td>0077</td>
<td>MCMD 008F</td>
</tr>
<tr>
<td>MNTA5</td>
<td>03E2</td>
<td>MCTDS 0028</td>
</tr>
<tr>
<td>CMDS</td>
<td>0363</td>
<td>MCRE 0466</td>
</tr>
<tr>
<td>CMES</td>
<td>0352</td>
<td>MFETE 04CF</td>
</tr>
<tr>
<td>CTOL</td>
<td>0334</td>
<td>PO 03FF</td>
</tr>
<tr>
<td>PAVE</td>
<td>0336</td>
<td>PX 00F6</td>
</tr>
<tr>
<td>TCM010</td>
<td>042F</td>
<td>RCMD 0066</td>
</tr>
<tr>
<td>RGG15</td>
<td>03D7</td>
<td>REGS 022F</td>
</tr>
<tr>
<td>RGA10</td>
<td>03A3</td>
<td>RGAIR 0310</td>
</tr>
<tr>
<td>RAL11</td>
<td>03A0</td>
<td>RIL 0517</td>
</tr>
<tr>
<td>RCMP</td>
<td>0327</td>
<td>RCSTU 0016</td>
</tr>
<tr>
<td>SCVE</td>
<td>0114</td>
<td>SCX03 011F</td>
</tr>
<tr>
<td>SDNV</td>
<td>0355</td>
<td>SCM5G 001E</td>
</tr>
<tr>
<td>SSAVE</td>
<td>3C38</td>
<td>SHDS 035D</td>
</tr>
<tr>
<td>TEMP</td>
<td>030A</td>
<td>TERN 0015</td>
</tr>
<tr>
<td>TTYAD</td>
<td>0237</td>
<td>TXBS 0084</td>
</tr>
<tr>
<td>VALDG</td>
<td>0347</td>
<td>VALDL 0232</td>
</tr>
<tr>
<td>XCM15</td>
<td>0463</td>
<td>XCM05 044B</td>
</tr>
<tr>
<td>XCM05</td>
<td>0155</td>
<td>XCM15 0154</td>
</tr>
<tr>
<td>XCM25</td>
<td>0196</td>
<td>XCM27 0197</td>
</tr>
</tbody>
</table>

* 01

* 02

* 03

* 04

* 05
REQUEST FOR READER'S COMMENTS

The Microcomputer Division Technical Publications Department attempts to provide documents that meet the needs of all Intel product users. This form lets you participate directly in the documentation process.

Please restrict your comments to the usability, accuracy, readability, organization, and completeness of this document.

1. Please specify by page any errors you found in this manual.

2. Does the document cover the information you expected or required? Please make suggestions for improvement.

3. Is this the right type of document for your needs? Is it at the right level? What other types of documents are needed?

4. Did you have any difficulty understanding descriptions or wording? Where?

5. Please rate this document on a scale of 1 to 10 with 10 being the best rating.

NAME ________________________________________ DATE __________________________

TITLE ________________________________________

COMPANY NAME/DEPARTMENT __________________________

ADDRESS __________________________________________________________________________

CITY _______________________________________ STATE ___________________ ZIP CODE _______________

Please check here if you require a written reply. ☐
WE’D LIKE YOUR COMMENTS . . .

This document is one of a series describing Intel products. Your comments on the back of this form will help produce better manuals. Each reply will be carefully reviewed by the responsible person. All comments and suggestions become the property of Intel Corporation.

BUSINESS REPLY MAIL
No Postage Stamp Necessary if Mailed in U.S.A.

Postage will be paid by:

Intel Corporation
3065 Bowers Avenue
Santa Clara, CA 95051

Attention: MCD Technical Publications