ASSOCIATE


Revision 3.0

for

System versions 3.0

Direct Comments Concerning this manual to:

DATA TECHNOLOGY INDUSTRIES, SAN LEANDRO, CALIFORNIA

(415) 638-1206

This Revision applies to:

ASSOCIATE Version 3.0
ASSOCIATE PLUS
1000 CPU Board Revision G and H
1005 CRT Board through Revision G
PROM Monitor Version 5.07
CRT PROM Version 1.7 through 1.A
Character PROM Version 5F and 60
Keyboard PROM Version 3, 4

The major change for Revision 3.0 is the name change to the ASSOCIATE. The ASSOCIATE is built by GNAT Computers Inc. and sold as the ASSOCIATE through Data Technology Industries. Systems having F revision boards should refer to manual revision 1.4 or 1.5.

Version 3.0 of the ASSOCIATE uses the H revision of the 1000 CPU board. The main changes on this board are in the area of the data separator which now uses a redesigned analog device to improve the reading of the double tracking disk drives on the ASSOCIATE PLUS.

Copyright (c) 1980, 1981, 1982 by GNAT Computers Inc and Data Technology Industries. All rights reserved. No part of this publication or the software which it describes may be reproduced, transmitted, transcribed, or translated into any language or computer language, in any form or by any means, mechanical, electronic, magnetic, optical, or otherwise without the written consent of GNAT Computers Inc.

Considerable time and expense has been expended in development of the software described herein. It has been made available at a reasonable price. Accordingly, we are seriously concerned about any unauthorized copying or use thereof and will take any and all available legal action against copying or distribution of this product.

Printed in the United States of America
APPENDIX 1. REVISION STATUS

This Manual Revision applies to:

- ASSOCIATE Revision 3.0
- CRT PROM Version 1.7, 1.8, 1.9, 1.A
- Keyboard PROM Version 4
- PROM Monitor 5.06, 5.07
- Character PROM 5F,60
- CP/M and Bios Rev: 2.2.4

<table>
<thead>
<tr>
<th>File name (.COM)</th>
<th>Version Number</th>
<th>Vendor</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASM</td>
<td>2.0</td>
<td>C</td>
</tr>
<tr>
<td>BANR</td>
<td>2.00</td>
<td>U</td>
</tr>
<tr>
<td>BAUD</td>
<td>2.0</td>
<td>U</td>
</tr>
<tr>
<td>COMPARE</td>
<td>2.02</td>
<td>U</td>
</tr>
<tr>
<td>COPY</td>
<td>2.03</td>
<td>U</td>
</tr>
<tr>
<td>D</td>
<td>3.01</td>
<td>U</td>
</tr>
<tr>
<td>DDT</td>
<td>2.2</td>
<td>C</td>
</tr>
<tr>
<td>DUMP</td>
<td>2.10</td>
<td>U</td>
</tr>
<tr>
<td>ED</td>
<td>n/a</td>
<td>C</td>
</tr>
<tr>
<td>EJEC</td>
<td>1.2</td>
<td>U</td>
</tr>
<tr>
<td>FORMAT</td>
<td>3.47</td>
<td>U</td>
</tr>
<tr>
<td>FREE</td>
<td>1.0</td>
<td>C</td>
</tr>
<tr>
<td>INITSYS</td>
<td>1.53</td>
<td>U</td>
</tr>
<tr>
<td>LOAD</td>
<td>n/a</td>
<td>C</td>
</tr>
<tr>
<td>PIP</td>
<td>1.5</td>
<td>C</td>
</tr>
<tr>
<td>PRINT</td>
<td>2.03</td>
<td>U</td>
</tr>
<tr>
<td>SIOPORTS.LIB</td>
<td></td>
<td>U</td>
</tr>
<tr>
<td>STDDEFS.LIB</td>
<td></td>
<td>U</td>
</tr>
<tr>
<td>STAT</td>
<td>n/a</td>
<td>C</td>
</tr>
<tr>
<td>SUBMIT</td>
<td>n/a</td>
<td>C</td>
</tr>
<tr>
<td>SYSGEN</td>
<td>2.05</td>
<td>U</td>
</tr>
<tr>
<td>T9511</td>
<td>2.1</td>
<td>U</td>
</tr>
<tr>
<td>TAPI</td>
<td>1.9</td>
<td>T</td>
</tr>
<tr>
<td>TBAUD</td>
<td>1.0</td>
<td>T</td>
</tr>
<tr>
<td>TCRT</td>
<td>1.6</td>
<td>T</td>
</tr>
<tr>
<td>TCTC</td>
<td>1.2</td>
<td>T</td>
</tr>
<tr>
<td>TDISK</td>
<td>2.12</td>
<td>T</td>
</tr>
<tr>
<td>TDMA</td>
<td>1.04</td>
<td>T</td>
</tr>
<tr>
<td>TIME</td>
<td>1.05</td>
<td>T</td>
</tr>
<tr>
<td>TINT</td>
<td>1.02</td>
<td>T</td>
</tr>
<tr>
<td>TPAT</td>
<td>1.00</td>
<td>T</td>
</tr>
<tr>
<td>TPIO</td>
<td>1.02</td>
<td>T</td>
</tr>
<tr>
<td>TRS232</td>
<td>2.03</td>
<td>T</td>
</tr>
<tr>
<td>TRTC</td>
<td>1.02</td>
<td>T</td>
</tr>
<tr>
<td>TSERIAL</td>
<td>1.02</td>
<td>T</td>
</tr>
<tr>
<td>TSPD</td>
<td>1.0</td>
<td>U</td>
</tr>
<tr>
<td>VPRINT</td>
<td>1.0</td>
<td>U</td>
</tr>
<tr>
<td>WM (optional)</td>
<td>1.07A</td>
<td>U</td>
</tr>
<tr>
<td>WM.HLP</td>
<td>1.07A</td>
<td>U</td>
</tr>
<tr>
<td>XSUB</td>
<td>2.0</td>
<td>C</td>
</tr>
</tbody>
</table>

C = CP/M Utility,  U = ASSOCIATE Utility,  T = ASSOCIATE Test Routine

ASSOCIATE FEATURES
ASSOCIATE FEATURES

GENERAL

Attractive Desktop Cabinet
Portable
Selectric Style Keyboard
Software Definable Function Keys
Accounting Style Numeric Pad
Low Glare Screen
Full Screen Editing

HARDWARE

Z80A CPU
65K RAM
700K Mass Storage on Dual Minifloppys
  Optional 1.6 Megabytes
  Optional Hard Disk
DMA Data Transfer
Hard Disk Interface for Additional Mass Storage
2 RS232 Serial Ports (Printer and Modem)
1 RS449 Serial Communication Port to 500K Baud
Programmable Baud Rates
Separate CRT Microprocessor
IEEE 488 GPIB Parallel I/O*
High Speed Arithmetic Processor*

SOFTWARE

PROM Resident Disk Boot and Diagnostic Monitor
CP/M** Version 2 Disk Operating System
Screen-Oriented Editor
Word Processing Program*
Business Software*
Communications Software*
Extensive Software Support*
  BASIC FORTRAN PASCAL RATFOR
  COBOL ASSEMBLER PL/1 "C"

*Optional
**Trademark of Digital Research
### TABLE OF CONTENTS

1. **INTRODUCTION** .................................................. 1-1

2. **SYSTEM ARCHITECTURE** ........................................ 2-1
   
   2.1 Electrical - A/C Wiring & Power Supply ............... 2-1
      
      2.1.1 Power Supply
      2.1.2 Battery Backup Option
      2.1.3 Power Up/Power Down
      2.1.4 110/220 Option
      2.1.5 Line Filtering

   2.2 CPU Board .................................................... 2-2
      
      2.2.1 Memory and Memory Assignment
      2.2.2 RESET
      2.2.3 System Timing
      2.2.4 CPU and Instruction Set
      2.2.5 CPU Jumper Table

3. **INPUT/OUTPUT** .................................................. 3-1
   
   3.1 DMA Controller ............................................. 3-1
   3.2 Arithmetic Processor ...................................... 3-2
   3.3 Interrupt Controller and Timer ......................... 3-4
   3.4 IEEE 488 GPIB I/O ......................................... 3-4
   3.5 High Speed Parallel I/O .................................. 3-6
   3.6 Network I/O RS449 ........................................ 3-7
   3.7 Console ...................................................... 3-9
   3.8 Modem I/O ................................................... 3-9
   3.9 List Device I/O ............................................ 3-11
   3.10 Floppy Disk ................................................. 3-12
   3.11 Status and Control ....................................... 3-13
   3.12 Real Time Clock .......................................... 3-14
   3.13 Baud Rate Generator ...................................... 3-16
   3.14 RAM On Control ........................................... 3-17
   3.15 Serial I/O Assignment ................................... 3-17
   3.16 S10 Summary ............................................... 3-18
   3.17 System Interconnection (Cables) ....................... 3-18

4. **VIDEO PROCESSOR BOARD** ...................................... 4-1
   
   4.1 Internal Architecture ...................................... 4-1
   4.2 Keyboard I/O ................................................ 4-3
   4.3 Utility I/O .................................................. 4-3
   4.4 CPU Interface ............................................... 4-4
   4.5 Character Generator and Video Interface .............. 4-5
   4.6 Light Pen Interface ...................................... 4-6
5. VIDEO PROCESSOR OPERATION. 5-1

5.1 CRT Display Format. 5-1
5.2 Control Sequences. 5-2
5.3 Soft Key Operation. 5-10
5.4 CRT Memory Allocation. 5-12
5.5 CRT PROM Entry Points. 5-12
5.6 ASCII Chart. 5-13

6. KEYBOARD 6-1

7. PROM MONITOR 7-1

7.1 PROM Monitor Operator Commands. 7-1
7.2 PROM Monitor Entry Points. 7-8

8. SERVICE PROCEDURES 8-1

8.1 Trouble Shooting Chart 8-1
8.2 Diagnostics 8-4

8.2.1 TRS232/ - Serial Port Test 8-4
8.2.2 TPIO - Parallel Port Test 8-5
8.2.3 TAPU - Arithmetic Processor Unit 8-5
8.2.4 TBAUD - Baud Rate Generator Test 8-6
8.2.5 TPAT - Pattern Test 8-6
8.2.6 TCCT - Counter Timer Test 8-7
8.2.7 TCRT - CRT Test 8-7
8.2.8 TDISK - Disk Drive Test 8-8
8.2.9 TDMA - Direct Memory Access Test 8-8
8.2.10 TINT - Interrupt Test 8-8
8.2.11 TRTC - Real Time Clock Test 8-9
8.2.12 TSPD - Rotational Speed Disk Test 8-9

8.3 Disassembly and Reassembly 8-10
8.4 Return of Material 8-12

APPENDIX

A. ASSOCIATE Wiring Diagram
   1000 CPU Board Assembly Diagram
   1000 CPU Board Schematic Diagram - 5 pages
   1005 CRT Board Assembly Diagram
   1005 CRT Board Schematic Diagram

B. Engineering Change Orders

OPTIONAL INFORMATION - Available To Qualified OEM's

BIOS Assembly Listing
GNAT Monitor Assembly Listing
Video Processor Assembly Listing
Keyboard PROM Listing
Character Generator PROM Listing
Bibliography


Kernigan and Plauger, "Software Tools"

Micropro, "WordMaster Release 1.06", Micropro International Corp., San Rafael, California, 1978


MOSTEK, "MOSTEK Z80 Handbook", MOSTEK Corp., Carrollton, Texas, 1977

Bowles, Kenneth L. "Problem Solving Using PASCAL", Springer-Verlag


REFERENCED DATA SHEETS

MOSTEK Z80 Handbook
INTEL MCS 80 Users Handbook
CP/M Disk Operating System Manual
WordMaster Screen Editor Manual
Advanced Micro Devices
9517A DMA Controller, 9511A APU, 26LS30 RS449 Driver
26LS32 RS449 Receiver
Mostek 4116 RAM
Western Digital 1793 Disk Controller
Texas Instrument 9914 IEEE Controller
Tandon TM 100-2 Disk Drives
C.ITOH CRT Monitor
Intel 8251 USART; 8155 RAM, Timer, I/O
Boschert OL65 Power Supply or OSC OS65 XL Power Supply
Keytronics Keyboard
Corcom 6J4 Line Filter or 6J1, 3V1 Corcom
Chapter 1

INTRODUCTION

Purpose

This reference manual provides a detailed operating description of the ASSOCIATE. It is intended for technical users and contains complete information on modification, operation of optional features, interfacing to other devices, and services available to the System 10 user. Another publication, the ASSOCIATE Operator's Handbook, should be referred to for standard operation and an overall view of the ASSOCIATE.

The ASSOCIATE

The ASSOCIATE is a full function microcomputer built for the small businesses, communication, and front end terminal applications.

Extensive I/O capabilities include three RS232 Ports, one of which alternatively can be configured as RS449, an optional IEEE 488 Bus interface, and a high speed parallel interface for a hard disk. The I/O enables the ASSOCIATE to be a useful tool for laboratory, communication, and mass storage functions. With the Communications Software Package, the ASSOCIATE becomes an excellent "Smart Terminal" for mainframe preprocessing and data entry.

A wide variety of software is available for use on the ASSOCIATE. Extensive business software is available through the Business Software Library; specialized business software can be custom written for users with particular requirements.

With CP/M, BASIC, Fortran, and assembly language, software can be easily developed on the system by OEM's for their specialized applications.

The word processing software features full screen-oriented editing, cursor movement control, and automatic file management. The function keys are defined for operations such as line delete, file up 1 screen, and move right 1 word.

Optional word processing software provides right-hand margin control, proportional spacing, and paging. These features plus others make the ASSOCIATE an outstanding computer for business and documentation purposes.

1-1
This page intentionally blank.
Chapter 2

SYSTEM ARCHITECTURE

This chapter contains information about the architecture of the ASSOCIATE, both electrical and logical.

Different components of the system and their interconnections are illustrated in Figure 1, "Functional Block Diagram." A full representation of the electrical and logical architecture is contained in the schematics included as Appendix A. As will be noted, boards are designed for maximum utilization of the board and system area.

Figure 1. Functional Block Diagram
2.1 Electrical--A/C Wiring and Power Supply

The ASSOCIATE uses a switching power supply. Specifications for the power supply are as shown in Table 2-1.

<table>
<thead>
<tr>
<th>Voltage</th>
<th>Current Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>+12</td>
<td>.5 AMP</td>
</tr>
<tr>
<td>-12</td>
<td>.15 AMP</td>
</tr>
<tr>
<td>+5</td>
<td>6 AMP</td>
</tr>
<tr>
<td>+12</td>
<td>1.8A (3 Amp Surge)</td>
</tr>
<tr>
<td></td>
<td>(Disk Drives)</td>
</tr>
</tbody>
</table>

The supply is sufficient for internal operation, however, it is not recommended that peripheral devices requiring more than 0.5 amps at +5 or 0.1 amp at +/– 12 volts be powered by this supply. For details on the power supply connection see "System Wiring Diagram" in Appendix A.

2.1.2 110/220 Option

The ASSOCIATE may be wired for 220 volt operation as outlined below. These changes are to be installed by a qualified technician only. Failure to strictly observe all items may cause considerable damage to the computer and void the warranty on the unit.

**220 Volt Set Up**

1. Remove Jumper JPl from the system power supply if supply is a Boschert or set switch to 220 volt position for the OSC supply. The power supply is located below the disk drives. See Section 8.3 for access instructions.

2. Remove and reinstall voltage selector card to the 240 volt position. This card is located on the AC connector at the rear of the unit. Note: This must show the 240 volt label toward the top.

3. Plainly label the system above the AC connector as being modified for 220 volt operation.

2.1.3 Line Filtering

The incoming A/C line is filtered with a capacitive inductive line filter. This reduces the system’s susceptibility to extraneous RFI and also suppresses emissions by the system itself.
2.2 CPU Board

Central to the ASSOCIATE CPU Board is the Z80 microprocessor which is completely supported by a number of features including 65K RAM memory, AM9517 DMA Controller, Floppy Disk Controller, 4 Serial I/O Ports, and optional items – Parallel I/O, AM9511 or AM9512 Arithmetic Processor, Real Time Clock, and IEEE 488 I/O. A complete view of the CPU board is available in the "Assembly 1000 CPU" in Appendix A. The CPU board is built to fully utilize the capability of the Z80 microprocessor through extensive use of intelligent support controllers. Memory, CPU, and general purpose system features are described in this chapter; I/O features are described in Chapter 3.

2.2.1 Memory

The CPU board contains 65K of Read/Write RAM and a 2K PROM overlay controlled by restart hardware and software.

On RESET the system comes up in restart mode, which disables RAM and enables PROM until code in the PROM monitor at F800H enables RAM. The first instruction in the PROM is a JMP to "BEGIN". The second instruction, an OUT DOH, which enables RAM, thereby giving an initial operating configuration of 63K RAM and 2K PROM. The OUT DOH also sets the modem baud rate. The memory is normally reconfigured under the Disk Operating System to 65K RAM and 0K PROM. See Section 3.11 Status and Control for operational details.

The PROM Monitor normally loads the Disk Operating System upon power up or reset. The Disk Operating System can then immediately execute a user program. This capability gives users, without technical expertise, the ability to start the system and immediately run a dedicated program; it also provides the opportunity to go directly back into an application program after a power failure.

The physical layout of memory, in terms of address range and bit number, is illustrated in the "Assembly 1000 CPU" in Appendix A. Thus, if memory fails the memory test (operator command T under Section 7.1), the location of the particular memory chip to be replaced can be quickly determined from the assembly diagram.

Several areas of memory are used for various functions by the Monitor and Operating System. These are:

<table>
<thead>
<tr>
<th>Table 2.1 Memory Assignments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000H-0002H</td>
</tr>
<tr>
<td>0003H</td>
</tr>
<tr>
<td>0004H</td>
</tr>
<tr>
<td>0005H-0007H</td>
</tr>
<tr>
<td>0008H-000AH</td>
</tr>
</tbody>
</table>
2. Architecture

In Monitor Operation:

- F7D9H Monitor Stack
- F7D9H-F7FOH Monitor Workspace
- F7F1H-F7FFH Trap Exit Code
- F800H-FFFFH PROM Monitor

In DOS Operation: (61K)

- 0100H-D7FFH Normal User Area (TPA)
- 0800H-DFFFH Console Command Processor (CCP)
- EEOOH-EDFFH Disk Operating System (BDOS)
- EEOOH-FFFFH System I/O Drivers (BIOS)

2.2.2 Reset

After a RESET the PROM monitor will attempt to load the first 128 bytes of Track 0 Sector 1 from the disk in the lower drive (A) into memory beginning at location 80H. This will normally be a boot for the Disk Operating System. If a the Disk Operating System is not found, a second reset will take the system to the monitor command level. Two resets in rapid succession will also take the operator to the monitor command level.

There are four ways of generating a RESET in the ASSOCIATE:

1. Manual - This RESET is generated by momentarily depressing SI on the main CPU Board (see "Assembly 1000" schematic); SI is accessible on the back right of the unit. This resets both the CRT and the Main Processor.

2. Keyboard - This RESET is generated through the video processor by keying in the Control-Shift-RST. This resets only the main CPU Board and not the video processor.
3. Power Failure - This RESET is exactly the same as the Manual RESET. It is activated by power detection circuitry when a power fail is detected. This feature is on 1000G and later CPU boards.

4. Software - The two software reset functions are defined in Section 3.11, Internal Status and Control; these resets affect only the floppy controller and the 9511 Arithmetic Processor.

2.2.3 System Timing

The main system timing is derived from a 16 MHz oscillator. This is divided for distribution to the 8 MHz, 4 MHz, 1 MHz, and 1/2 MHz signals used throughout the board. The most used signal for system timing is 4 MHz; this signal has the necessary circuitry to insure that the Z80A has the required levels and transition speed for proper clocking.

Baud rate generation for serial ports is accomplished through a separate crystal; the frequency of that crystal (4.9152 MHz) is an even multiple of standard baud rates. A dual software programmable divider described in Section 3.13 gives the actual baud rates to the serial I/O devices.

Circuitry is provided on the board to take care of the disk drive timing requirements. The following digital delays are used:

Motor Turn On. If the drive motors are off, a one second delay is introduced in the head load command to allow time for motor startup.

Motor Turn Off. If the floppy disk controller is not accessed for 128 seconds the disks are deselected and the motor on signal is deactivated. The disk motor turn off timing may be optionally set at other values as follows:

<table>
<thead>
<tr>
<th>JP35 Motor Turn Off Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pin 5 jumpered to pin 6</td>
</tr>
<tr>
<td>Pin 3 jumpered to pin 4</td>
</tr>
<tr>
<td>Pin 1 jumpered to pin 2</td>
</tr>
<tr>
<td>All open</td>
</tr>
</tbody>
</table>

Screen Blanking. The CRT display will be turned off 20 minutes after the last CRT activity. The purpose of this turn off is to extend CRT phosphor life. The timeout is user programmable - See Section 5.2 for the procedure. The screen will turn back on when any keyboard key is pressed or the CPU sends a character to the video processor.

2.2.4 CPU & Instruction Set

For detailed information on the Z80 microprocessor refer to "MOSTEK Microcomputer: Z80 Data Book." Based on N-Channel MOS technology, the
Z80A-CPU is packaged in an industry standard 40-pin Dual In-Line Package. Significant features include a single power supply (+5V), a single system clock signal, multiple interrupt capability, and dynamic memory refresh.

Software generation on the ASSOCIATE is simplified through the Z80's 158 instructions, unlimited subroutine nesting, vectored interrupts, and DMA capabilities. Product development and system design are enhanced by the extensive set of instructions and the varied addressing modes of the Z80, as well as its capability of directly addressing 65K 8-bit words of memory. For a detailed description of the Operating System software aids for assembly program production, refer to the "CP/M Disk Operating System Manual."

The Z80 maintains full software compatibility with the 8080 microprocessor, since the Z80 instruction set is a superset of the 8080 instruction set. The Z80 has additional addressing modes, a larger instruction set, and extended registers.

The internal architecture of the Z80 consists of eighteen 8-bit registers, four 16-bit registers, arithmetic and logic unit (ALU), and instruction register.

2.2.5 CPU Jumper Table

The following table summarizes the jumper options for the G and H revision CPU board.
<table>
<thead>
<tr>
<th>JP</th>
<th>PG</th>
<th>DEFAULT</th>
<th>NAME-FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>used</td>
<td>pins 1 and 2 used by inverter in reset</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>open</td>
<td>pin 3 O.C. reset output</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>open</td>
<td>pin 4 O.C. /ten from 74S287 - PIO Control signal</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>open</td>
<td>pin 5 O.C. /outg from 74S287 - PIO Control signal</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>open</td>
<td>pin 6 O.C. /wait from 74S287 - PIO Control signal</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>open</td>
<td>pin 7 O.C. /tack from 74S287 - PIO Control signal</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>open output</td>
<td>pin 2 open 8304 PIO port A output mode</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>open</td>
<td>pin 2 to 1 8304 PIO port A input mode</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>open</td>
<td>pin 2 to 3 8304 controlled by latch bit D2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>open</td>
<td>74S287 address inputs-1 to 2 A5 to B7</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>open</td>
<td>PIO port B data DO-D7</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>open</td>
<td>PIO port A data DO-D7</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>open</td>
<td>parallel DMA port data DO-D7</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>open</td>
<td>pin 2 P2 pin 49, pin 1 +5volts, pin 3 ground</td>
</tr>
<tr>
<td>7A</td>
<td>3</td>
<td>enabled</td>
<td>network I/O J1 Remote Gnd to chassis ground</td>
</tr>
<tr>
<td>7B</td>
<td>3</td>
<td>open</td>
<td>network I/O pin 8, Carrier Detect Signal</td>
</tr>
<tr>
<td>7D</td>
<td>3</td>
<td>open</td>
<td>network I/O pin 17, RSET</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>2-3 grounded</td>
<td>RS422 input option, Network Receiver Option</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>2-3 grounded</td>
<td>RS422 input option, Network Receiver Option</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>2-3 grounded</td>
<td>RS422 input option, Network Receiver Option</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>1-2 RS423</td>
<td>1-2 26LS30 in RS423 mode, 2-3 in RS422 mode</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>2-3 grounded</td>
<td>RS422 input option, Network Receiver Option</td>
</tr>
<tr>
<td>12A</td>
<td>3</td>
<td>enabled</td>
<td>RS422 input option, Network Receiver Option</td>
</tr>
<tr>
<td>13</td>
<td>3</td>
<td>open</td>
<td>1-2 list/network port -12volts on pin 10</td>
</tr>
<tr>
<td>13A</td>
<td>3</td>
<td>enabled</td>
<td>list I/O J2 Remote Gnd to chassis ground</td>
</tr>
<tr>
<td>13B</td>
<td>3</td>
<td>open</td>
<td>list I/O pin 6, Data Set Ready signal</td>
</tr>
<tr>
<td>14</td>
<td>3</td>
<td>2-3 RS423</td>
<td>26LS30 Wee 1-2 ground of RS422, 2-3 -5volts for RS423</td>
</tr>
<tr>
<td>15</td>
<td>3</td>
<td>open</td>
<td>1-2 modem port -12volts on pin 10</td>
</tr>
<tr>
<td>15A</td>
<td>3</td>
<td>enabled</td>
<td>1-2 modem I/O tset signal</td>
</tr>
<tr>
<td>15B</td>
<td>3</td>
<td>open</td>
<td>modem I/O pin 8, Carrier Detect signal</td>
</tr>
<tr>
<td>15C</td>
<td>3</td>
<td>enabled</td>
<td>modem I/O Remote Gnd to chassis ground</td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>1,2,3, open</td>
<td>1 ground, 2 active low parallel output enable, 3 PIO port B bit 0 for output control</td>
</tr>
<tr>
<td>16A</td>
<td>4</td>
<td>4-5</td>
<td>output enable active low,4-2 for active hi</td>
</tr>
<tr>
<td>16A</td>
<td>4</td>
<td>6-7</td>
<td>6-7 for /dreq0 from PIO port B bit 3</td>
</tr>
<tr>
<td>17</td>
<td>4</td>
<td>not used</td>
<td>PIO strobes-1 astb, 2 bstb, 3 ardy, 4 nc, 5 brdy</td>
</tr>
<tr>
<td>18</td>
<td>4</td>
<td>open</td>
<td>not used</td>
</tr>
<tr>
<td>20</td>
<td>3</td>
<td>test points</td>
<td>pin 1 console serial receive data, pin 2 xmit data</td>
</tr>
<tr>
<td>20A</td>
<td>3</td>
<td>open</td>
<td>S1O 1 sync A</td>
</tr>
<tr>
<td>21</td>
<td>3</td>
<td>1-2 sclk0</td>
<td>S1O 0 A Modem receive clock</td>
</tr>
<tr>
<td>22</td>
<td>3</td>
<td>1-2 sclkl</td>
<td>S1O 0 B List Device clock</td>
</tr>
<tr>
<td>22A</td>
<td>3</td>
<td>open</td>
<td>S1O 0 rtcb 1-2 to tset driver to list pin 15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>open</td>
<td>S1O 0 rtcb 3-4 to rsst driver to list pin 17</td>
</tr>
<tr>
<td>23</td>
<td>3</td>
<td>1-2 sclk0</td>
<td>S1O 0 A Modem xmit clock</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>5</td>
<td>open</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>4</td>
<td>1-2 enabled</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>test points</td>
<td>data bus D0-D7</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>3</td>
<td>1-2 sclk0</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>3</td>
<td>1-2 RS232</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>3</td>
<td>1-2 RS232</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>3</td>
<td>1-2 RS232</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>5</td>
<td>2-3 01</td>
<td></td>
</tr>
<tr>
<td>31A</td>
<td>3</td>
<td>open</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>4</td>
<td>open</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>5</td>
<td>1-2 enabled</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>1</td>
<td>open</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>5-6</td>
<td>128 sec</td>
<td></td>
</tr>
<tr>
<td>35A</td>
<td>5</td>
<td>open</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>5</td>
<td>open</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>Drive Side Source Select</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>not used</td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>Double Density Jumper, Single Density Open</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>1</td>
<td>open</td>
<td></td>
</tr>
<tr>
<td>40A</td>
<td>1</td>
<td>test points</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>1-2</td>
<td>02</td>
<td></td>
</tr>
<tr>
<td>41A</td>
<td>1</td>
<td>test point</td>
<td></td>
</tr>
<tr>
<td>41B</td>
<td>1</td>
<td>open</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>1</td>
<td>open</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>5</td>
<td>open</td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>5</td>
<td>test point</td>
<td></td>
</tr>
</tbody>
</table>

In the above table: O.C. = Open Collector Output
/in front signal name indicates inverted signal
Chapter 3

INPUT/OUTPUT

The I/O addressing field of the ASSOCIATE is entirely committed on the main CPU Board: there is no provision for direct external user access to the Z80 data bus. Communication with the external world is handled through the five connectors located at the back of the unit. The file S10PORTS.LIB on the system distribution diskette contains further information on the standard port names and assignments.

Table 3-1 below describes the address range and assignments of devices addressed through the I/O instructions.

Table 3-1. I/O Addresses and Assignments

<table>
<thead>
<tr>
<th>Address</th>
<th>Signal Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>00H-OFH</td>
<td>S9517</td>
<td>DMA Controller</td>
</tr>
<tr>
<td>10H-1FH</td>
<td>S9511</td>
<td>Arithmetic Processor</td>
</tr>
<tr>
<td>20H-2FH</td>
<td>CTC</td>
<td>Interrupt Controller, Timer</td>
</tr>
<tr>
<td>30H-3FH</td>
<td>S9914</td>
<td>IEEE 488 Controller</td>
</tr>
<tr>
<td>40H-4FH</td>
<td>PDMA1</td>
<td>Parallel DMA Data Port 1</td>
</tr>
<tr>
<td>50H-5FH</td>
<td>PIO0</td>
<td>Parallel I/O</td>
</tr>
<tr>
<td>60H-61H</td>
<td>SI01</td>
<td>Network I/O RS449</td>
</tr>
<tr>
<td>62H-63H</td>
<td>SI01</td>
<td>Console (CRT)</td>
</tr>
<tr>
<td>70H-71H</td>
<td>SI00</td>
<td>Modem RS232</td>
</tr>
<tr>
<td>72H-73H</td>
<td>SI00</td>
<td>List RS232</td>
</tr>
<tr>
<td>80H-8FH</td>
<td>S1793</td>
<td>Floppy Disk Controller</td>
</tr>
<tr>
<td>90H-9FH</td>
<td>PDMA0</td>
<td>Parallel DMA Data Port 0</td>
</tr>
<tr>
<td>A0H-AFH</td>
<td>LATCH</td>
<td>Status and Control</td>
</tr>
<tr>
<td>B0H-BFH</td>
<td>RTC0</td>
<td>Real Time Clock 0</td>
</tr>
<tr>
<td>C0H-CFH</td>
<td>RTC1</td>
<td>Real Time Clock 1</td>
</tr>
<tr>
<td>D0H-DFH</td>
<td>BAUDO</td>
<td>Modem Clock, RAM on Control</td>
</tr>
<tr>
<td>E0H-EFH</td>
<td>BAUD1</td>
<td>RS449 Clock</td>
</tr>
<tr>
<td>F0H-FFH</td>
<td>Not Used</td>
<td>List Clock</td>
</tr>
</tbody>
</table>

Refer to the specific Data Sheet for I/O sub-maps and operational details of a particular device.

3.1 DMA Controller 00H-0FH

The DMA Controller is an Advanced MicroDevices AM9517A chip. This chip provides direct memory access to system memory (through four channels) for the following high-speed devices:

1) High speed Parallel I/O (Channel 0)
2) Floppy Disk I/O (Channel 1)
3) IEEE Bus Interface (Channel 2)
4) Network Serial I/O (Channel 3)
The fourth channel can also be used for memory to memory transfer. For further details concerning the DMA Controller, refer to Advanced MicroDevices AM9517A data sheet.

The AM9517A uses all 16 addresses of its block of 16; it is further divided into four blocks of two addresses each, where each block of addresses corresponds to one channel of DMA control. This is followed by a sub-block of control registers occupying eight addresses. For each channel, the function of the addresses are:

1) Base current address
2) Base current word count

3.2 **Arithmetic Processor 10H,11H**

The Arithmetic Processor Option 9511/9512 provides the ASSOCIATE with a complete high performance arithmetic processor. It considerably enhances the arithmetic computational speed of the system and includes not only floating-point, but fixed-point processing as well. In addition to add, subtract, multiply, and divide operations the 9511 includes transcendental functions and control and conversion commands.

The 9512 is a four function (add, subtract, multiply, and divide) processor which provides single precision (32 bit) and double precision (64 bit) operations in the IEEE floating point standard. This option is specified as APU-2 when ordering.

The Arithmetic Processor occupies two address locations (10H and 11H) in the Z80 addressing field. Operation requires that the operands be written on the processor's stack at location 10H followed by the issuance of a command to the command word at location 11H. Two, four, or eight byte values are written with the least significant byte first. All required bytes must be read or written in sequence for proper operation. The desired operation code is then output to the Processor command port (11H) and the processor will calculate the result. The result can be read after completion at port 10H most significant byte first.

Completion of the processing is indicated in two ways. Port 11H serves as the status register which can be read for completion status; or the end of processing may be signaled by an interrupt.

The pause jumper JP42 must have pins 13 and 14 jumpered for proper operation with the 9511 or 9512. This jumper will cause the Z80 to enter a wait state in order to synchronize the slower read and write timing of the APU.

The user is cautioned that no attempt to read the result of the calculation should be made before the calculation is complete. The wait timeout logic will override the APU and an invalid result will be returned.
There is a jumper option for interrupt and wait operation of the Arithmetic Processor. The AM9511 uses DINT2 (interrupt 2 into the interrupt controller); that interrupt is normally driven by the END function and by a service request. Three jumper options are available for use with the APU as described below:

<table>
<thead>
<tr>
<th>Jumper Pin JP42</th>
<th>APU Jumper Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pin 13 Jumpered to 14</td>
<td>APU Pause generates CPU Wait</td>
</tr>
<tr>
<td>Pin 7 Jumpered to 8</td>
<td>APU SVREQ activates Interrupt 2</td>
</tr>
<tr>
<td>Pin 3 Jumpered to 4</td>
<td>APU END activates Interrupt 2</td>
</tr>
</tbody>
</table>

The Arithmetic Processor may be reset under software control at any time. It is sent a reset by the power up sequence. A zero in Bit 3 (APURES) of the Control Word at Address AOH resets the APU. The APURES bit should be held to zero for 50 microseconds. When using this reset be sure to get the status of the control latch from location 000BH in memory for the setting of the other bits in the control word.

Two sets of standard software are supplied on diskette with the APU option. These are the 9511 test routines and the 9511 Fortran Library subroutines. The file containing the test routine source is named TAPU.MAC. It is written in 280 assembly code.

The Fortran Library subroutines are in the file APUI1.REL. These can be linked into a Fortran program by linking this file in front of the FORLIB.REL. The APUI1.REL file contains the following entry points, normally taken from the FORTRAN library:

<table>
<thead>
<tr>
<th>ENTRY POINT</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSTAPU</td>
<td>Perform APU Reset</td>
</tr>
<tr>
<td>Saa</td>
<td>add real and integer</td>
</tr>
<tr>
<td>Sab</td>
<td>add real and real</td>
</tr>
<tr>
<td>Sda</td>
<td>divide real by integer</td>
</tr>
<tr>
<td>Sdb</td>
<td>divide real by real</td>
</tr>
<tr>
<td>Sea</td>
<td>exponentiation of real by integer</td>
</tr>
<tr>
<td>Seb</td>
<td>exponentiation of real by real</td>
</tr>
<tr>
<td>Smm</td>
<td>multiply real by integer</td>
</tr>
<tr>
<td>Smb</td>
<td>multiply real by real</td>
</tr>
<tr>
<td>Ssa</td>
<td>subtract integer from real</td>
</tr>
<tr>
<td>Ssb</td>
<td>subtract real from real</td>
</tr>
<tr>
<td>Scf, float</td>
<td>convert integer to real</td>
</tr>
<tr>
<td>Sh, int,ifix</td>
<td>convert real to integer</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
\text{sqrt} & \quad \text{real square root} \\
\text{sin} & \quad \text{real sine} \\
\text{cos} & \quad \text{real cosine} \\
\text{tan} & \quad \text{real tangent} \\
\text{asin} & \quad \text{real arcsine} \\
\text{acos} & \quad \text{real arccosine} \\
\text{atan} & \quad \text{real arctangent} \\
\text{alog10} & \quad \text{real log base 10} \\
\text{alog} & \quad \text{real log base e} \\
\text{exp} & \quad \text{real e**x}
\end{align*}
\]
3.3 Interrupt Controller and Timer 20H-23H

The Interrupt Controller provides system timing and manages inputs for the various interrupts that can be generated by other I/O devices. For detailed information refer to the Data Sheet on the MK3882/Z80A CTC. The Interrupt Controller can do three different subsets of operations:

1. Direct Interrupt Controller. The four devices which utilize the CTC for interrupt control are connected to the trigger inputs as follows:
   a. DMA Controller Interrupt Request - Trigger 0
   b. Floppy Disk Controller - Trigger 1
   c. 9511 Arithmetic Processor/Power fail - Trigger 2
   d. IEEE Bus Controller/Real Time Clock - Trigger 3

2. Interrupt Priority Enabling. The CTC is the highest priority device on the interrupt enabled daisy chain. The PIO is the second device, SIO #1 is third, and SIO #0 is lowest priority.

3. Timer Operation. Any of the four CTC channels may be used as a timer if its corresponding trigger input is not enabled. The slowest frequency timeout available is 62.5 Hz.

See Section 2.2.1 Memory for the memory map recommended for the interrupt vectors.

3.4 IEEE - 488 GPIB 30H-37H

The IEEE 488 GPIB Interface Option gives the ability to communicate over the general purpose interface bus. The option is designed around the Texas Instrument TMS 9914 GPIB Interface chip. The 9914 allows operation as a bus controller or as a listener/talker. A complete description of this chip and an introduction to the bus itself are given in the 9914 data sheets.

Sample software for use on the GPIB is supplied on diskette with the option and summarized in this manual.

The TMS 9914 occupies a block of 8 addresses in the addressing field. These addresses are 30H to 37H. A complete description of the function of each of these addresses is given in detail in the TMS 9914 data sheets.

A jumper option is available on the CPU board to either enable or disable a resistor termination network for the GPIB. As the ASSOCIATE is normally expected to be at the end of the GPIB cable, the termination resistor network is normally enabled. The termination enable option is JP25 located next to IC2B.
Jumper JP25  Terminator Enable

1 jumpered to 2  Terminator enabled. (Normal)
2 jumpered to 3  Terminator disabled. Clad on
back of board between pins
1 and 2 must be cut!

The IEEE controller may be operated under interrupt control on CTC channel
3. This option is control at Jumper Pin Block JP42 as described below. The
9914 must be programmed to generate the desired interrupt output on
pin 9 and the CTC must be programmed to use channel 3 as an external
interrupt input.

Jumper JP42  IEEE Interrupt Request Option

17 jumpered to 18  Interrupt connected. (Normal)
17 open to 18     Interrupt not connected
(clad on board must be cut)

The 9517A hand shake lines DREQ2 and DACK2 will perform data transfers
between the 9914 and memory as programmed by the third DMA controller
channel. Use of this feature requires the appropriate programming to be
done for the 9517A DMA Controller chip.

With the GPIB option, sample software is provided for demonstration of its
use. Currently, three sets of software are provided on disk. The first
set is a collection of subroutines which may be called from Basic or
Fortran. These subroutines illustrate how to initialize the TMS 9914 and
how to put data on the bus and take data off the bus. The file names are:

488SUBS.DOC  Documentation file
488SUBS.MAC  Subroutine source code

The second sample software is a Rational Fortran (RATFOR) program which
is used to transfer data from a IEEE 488 tape deck to the microcomputer's
floppy disks. This program named TAPE488.RAT uses many of the utility
subroutines described above.

The third set of sample software is a program to test the IEEE port. The
T488.MAC tests the IEEE-488 option and demonstrates the use of DMA and
interrupts with the 9914 device. In order for the T488.MAC to run, a
second ASSOCIATE connected to the system under test must also be running
the T488.MAC program.

3.5  High Speed Parallel I/O 50H-53H, 90H, 40H

High speed parallel I/O is optionally available through the rear panel
connector P2 for interfacing to such devices as a hard disk. The
interface consists of two 8-bit bidirectional data channels and eight-bits
of control signals. These functions are provided by an MK3881/280A-PIO,
bidirectional drivers and a 74S287 control PROM. This PROM can be defined
for the application the parallel port is used on. A resistor pullup
network is normally installed if the PROM functions are not required.

DMA control signals from Port 90H and DMA channel 0 are used as inputs for the control PROM.

WAIT LOGIC (IC1B) allows external devices to hold the system bus for up to 15.5 microseconds during CPU or DMA data transfers. It may also be controlled by the 74LS287 PROM and jumper options.

Further details on configurations for different external devices will be given in Application Notes for those devices.

Definition of Pinouts of the external connector is listed in the following table; all signals are active low.

<table>
<thead>
<tr>
<th>Pin #</th>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PC0</td>
<td>Parallel Control Signal</td>
</tr>
<tr>
<td>3</td>
<td>PC1</td>
<td>Parallel Control Signal</td>
</tr>
<tr>
<td>5</td>
<td>PC2</td>
<td>Parallel Control Signal</td>
</tr>
<tr>
<td>7</td>
<td>PC3</td>
<td>Parallel Control Signal</td>
</tr>
<tr>
<td>9</td>
<td>PC4</td>
<td>Parallel Control Signal</td>
</tr>
<tr>
<td>11</td>
<td>PC5</td>
<td>Parallel Control Signal</td>
</tr>
<tr>
<td>13</td>
<td>PC6</td>
<td>Parallel Control Signal</td>
</tr>
<tr>
<td>15</td>
<td>PC7</td>
<td>Parallel Control Signal</td>
</tr>
<tr>
<td>17</td>
<td>PD0</td>
<td>Parallel data 0</td>
</tr>
<tr>
<td>19</td>
<td>PD1</td>
<td>Parallel data 1</td>
</tr>
<tr>
<td>21</td>
<td>PD2</td>
<td>Parallel data 2</td>
</tr>
<tr>
<td>23</td>
<td>PD3</td>
<td>Parallel data 3</td>
</tr>
<tr>
<td>25</td>
<td>PD4</td>
<td>Parallel data 4</td>
</tr>
<tr>
<td>27</td>
<td>PD5</td>
<td>Parallel data 5</td>
</tr>
<tr>
<td>29</td>
<td>PD6</td>
<td>Parallel data 6</td>
</tr>
<tr>
<td>31</td>
<td>PD7</td>
<td>Parallel data 7</td>
</tr>
<tr>
<td>33</td>
<td>HD0</td>
<td>High Speed Parallel Data 0</td>
</tr>
<tr>
<td>35</td>
<td>HD1</td>
<td>High Speed Parallel Data 1</td>
</tr>
<tr>
<td>37</td>
<td>HD2</td>
<td>High Speed Parallel Data 2</td>
</tr>
<tr>
<td>39</td>
<td>HD3</td>
<td>High Speed Parallel Data 3</td>
</tr>
<tr>
<td>41</td>
<td>HD4</td>
<td>High Speed Parallel Data 4</td>
</tr>
<tr>
<td>43</td>
<td>HD5</td>
<td>High Speed Parallel Data 5</td>
</tr>
<tr>
<td>45</td>
<td>HD6</td>
<td>High Speed Parallel Data 6</td>
</tr>
<tr>
<td>47</td>
<td>HD7</td>
<td>High Speed Parallel Data 7</td>
</tr>
<tr>
<td>49</td>
<td>SPARE</td>
<td>Spare (+5 volts optional)</td>
</tr>
<tr>
<td>2-50 EVEN</td>
<td>GND</td>
<td>Ground</td>
</tr>
</tbody>
</table>

When installing the PIO option, cut JP18 6 from 7. With the PIO not in the system, this jumper is necessary to pass interrupt requests through to the interrupt controller.
3.6 Network I/O 60H, 61H

This serial output port provides I/O for network communication applications with RS449 protocol. This protocol provides several features not available in RS232: higher speed and differential drive. The RS449 is a combined specification for RS422 and RS423. The following description is in terms of the RS422 and RS423.

Port 60H is data and Port 61H is status and control. The Network I/O baud rate is software programmable by the baud rate generator addressed at Port D0H and described in Section 3.13. Note that its clock is shared by the modem port. When used with the serial manifold external clocking should be selected using JPI7. The Network Port can also be optionally configured to operate in RS232 protocol. When setup as RS232 the port functions as a Data Set. For each of the different protocols the following table defines the functions of the connector.

The normal configuration for C Revision and later boards is RS232.

<table>
<thead>
<tr>
<th>Pin#</th>
<th>RS422</th>
<th>RS423</th>
<th>RS232</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ground</td>
<td>Ground</td>
<td>Ground</td>
</tr>
<tr>
<td>2</td>
<td>Tx Data -</td>
<td>Tx Data</td>
<td>Tx Data (in)</td>
</tr>
<tr>
<td>3</td>
<td>Rx Data -</td>
<td>Rx Data</td>
<td>Rx Data (out)</td>
</tr>
<tr>
<td>4</td>
<td>Request to Send -</td>
<td>Request to Send</td>
<td>Request to Send (in)</td>
</tr>
<tr>
<td>5</td>
<td>Rx Clock +</td>
<td>Clear to Send</td>
<td>Clear to Send (out)</td>
</tr>
<tr>
<td>6</td>
<td>Rx Data +</td>
<td>Data Set Rdy</td>
<td>Data Set Rdy (out)</td>
</tr>
<tr>
<td>7</td>
<td>Signal Ground</td>
<td>Signal Ground</td>
<td>Signal Ground</td>
</tr>
<tr>
<td>8</td>
<td>+12</td>
<td>External Power</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>-12</td>
<td>External Power</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Tx Data +</td>
<td>Not Used</td>
<td>Not Used (in)</td>
</tr>
<tr>
<td>11</td>
<td>Rx Clock -</td>
<td>Rx Clock</td>
<td>Rx Clock (out)</td>
</tr>
<tr>
<td>12</td>
<td>Data Terminal Rdy +</td>
<td>Not Used</td>
<td>Not Used (in)</td>
</tr>
<tr>
<td>13</td>
<td>Request to Send +</td>
<td>Not Used</td>
<td>Not Used (in)</td>
</tr>
<tr>
<td>14</td>
<td>Data Terminal Rdy -</td>
<td>Data Terminal Rdy</td>
<td>Data Set Ready (in)</td>
</tr>
<tr>
<td>15</td>
<td>Clock In -</td>
<td>Clock In</td>
<td>Clock In (in)</td>
</tr>
<tr>
<td>16</td>
<td>Clock In +</td>
<td>Not Used</td>
<td>Not Used (in)</td>
</tr>
</tbody>
</table>

The mode option defines the operation of the 26LS30 driver as follows:

Jumper JP11. Driver Mode Select

RS423 or RS232  2 Jumpered to 1(normal)
RS422  2 Jumpered to 3

Inputs to the receivers may be changed by the jumper options JP8, JP9, JP10, and JP12. For single ended modes (RS232 and RS423) the + input of the receiver is grounded. The jumper options are:


RS423 or RS232  2 Jumpered to 3 (normal)
RS422  2 Jumpered to 1
Transmitted Data may be inverted by use of Option JP28. For RS232 it must be inverted for proper RS232 level signals. The clad must be cut on the back side of the circuit board if the inverted option is implemented.

Jumper JP28. Transmitted Data Inversion Option

<table>
<thead>
<tr>
<th>Option</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inverted (RS232)</td>
<td>2 Jumped to 1 (normal)</td>
</tr>
<tr>
<td>Noninverting (RS422, RS423)</td>
<td>2 Jumped to 3</td>
</tr>
</tbody>
</table>

The Network port also has a clock option as follows:

Jumper JP27. Auxiliary Clock Option

<table>
<thead>
<tr>
<th>Option</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal Modem Clock</td>
<td>2 Jumped to 1 (normal)</td>
</tr>
<tr>
<td>External Clock</td>
<td>2 Jumped to 3</td>
</tr>
</tbody>
</table>

The inputs of the network port have an option for installing termination resistors. These resistors RT1 through RT4, located next to device IN, can be used in either the RS422 or RS423 mode. See the schematic and assembly drawings for further information.

The slew rate control option allows output slew rate to be controlled by capacitors CS1 through CS4 located between devices IP and P4. For value determination refer to the 26LS30 Data Sheet.

For applications requiring a small amount of external power at + and - 12 volts, a jumper option JP13 provides power to pins 9 and 10 of the serial connector.

Jumper JP13 + and -12 volt power option

<table>
<thead>
<tr>
<th>JP13 Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2</td>
<td>for -12 on pin 10</td>
</tr>
<tr>
<td>3-4</td>
<td>for +12 on pin 9</td>
</tr>
</tbody>
</table>

3.7 Console I/O 62H, 63H

The Console I/O transfers data between the CPU board and the Video and keyboard processor. One-half of an SIO device (2G) handles the RS232 protocol for data transfer. The clocking for this half of the SIO is supplied by the Video Processor. Refer to Chapter 5 on the Video Processor Board for complete information from the Video Board side of the I/O.

Port 62H is data and Port 63H is Status and Control.

The following table reflects pinouts from the CPU Board end of the Process:
3. Input/Output

Table 3-5. Video I/O Pinouts

<table>
<thead>
<tr>
<th>Pin #</th>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,2,3,4,7,8</td>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>10,16,17,19,20</td>
<td>DTR</td>
<td>Data Terminal Ready (from CPU)</td>
</tr>
<tr>
<td>6</td>
<td>TxD</td>
<td>Transmit Data (from CPU)</td>
</tr>
<tr>
<td>11</td>
<td>RxD</td>
<td>Receive Data (into CPU)</td>
</tr>
<tr>
<td>12</td>
<td>TRxC</td>
<td>Transmitter Receive Clock</td>
</tr>
<tr>
<td>13</td>
<td>CTS</td>
<td>Clear to Send (into CPU)</td>
</tr>
<tr>
<td>14</td>
<td>RTS</td>
<td>Request to Send (from CPU)</td>
</tr>
<tr>
<td>15</td>
<td>EXRESET</td>
<td>External Reset (into CPU)</td>
</tr>
<tr>
<td>18</td>
<td>VR</td>
<td>Video Reset (from CPU)</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>No Connection</td>
</tr>
</tbody>
</table>

3.8 MODEM I/O 70H, 71H

The Modem I/O is handled by half of the SIO device at location 2H and is configured according to the EIA specifications for a Data Terminal. 70H is data and Port 71H is Status and Control. Refer to the Data book describing the MK3884/Z80-SIO and the appropriate schematics for additional detail.

The modem port baud rate is software programmable by the baud rate generator at Port DOH and described in Section 3.13. Note that this baud rate generator is shared with the Network I/O Port.

Connection to a modem is made through P3, a male DB25P connector. The following table describes pinouts for the modem I/O:

Table 3-6. MODEM I/O Pinouts

<table>
<thead>
<tr>
<th>Pin #</th>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GND</td>
<td>Chassis Ground</td>
</tr>
<tr>
<td>2</td>
<td>TxD</td>
<td>Transmit of Data (out)</td>
</tr>
<tr>
<td>3</td>
<td>RxD</td>
<td>Receive Data (in)</td>
</tr>
<tr>
<td>4</td>
<td>RTS</td>
<td>Request to Send (out)</td>
</tr>
<tr>
<td>5</td>
<td>CTS</td>
<td>Clear to Send (in)</td>
</tr>
<tr>
<td>6</td>
<td>DSR</td>
<td>Data Set Ready (in)</td>
</tr>
<tr>
<td>7</td>
<td>GND</td>
<td>Signal Ground</td>
</tr>
<tr>
<td>9</td>
<td>+12</td>
<td>External Power - Jumper JP15 3-4</td>
</tr>
<tr>
<td>10</td>
<td>-12</td>
<td>External Power - Jumper JP15 1-2</td>
</tr>
<tr>
<td>15</td>
<td>TSET</td>
<td>Transmitter Signal Element Timing (in)</td>
</tr>
<tr>
<td>17</td>
<td>RSET</td>
<td>Receiver Signal element Timing (in)</td>
</tr>
<tr>
<td>20</td>
<td>DTR</td>
<td>Data Terminal Ready (out)</td>
</tr>
<tr>
<td>23</td>
<td>DSRS</td>
<td>Data Set Ready Secondary (terminated with 22K to +12)</td>
</tr>
<tr>
<td>24</td>
<td>STSET</td>
<td>Source Transmitter Signal element Timing (out)</td>
</tr>
</tbody>
</table>
There are two jumper options available with modem I/O on the timing clocks. One set of jumper pads controls the receive clock and one set is for the transmit clock.

The Transmitter Clock option is identified as JP23 on the PROM and Serial I/O schematic in Appendix A. The transmitter clock can be sourced internally or externally, but it is normally jumpered for internal.

**Jumper JP23. Modem Transmitter Clock Option**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal</td>
<td>2 Jumpered to 1 (normal)</td>
</tr>
<tr>
<td>External</td>
<td>2 Jumpered to 3</td>
</tr>
<tr>
<td>Receiver</td>
<td>Open</td>
</tr>
</tbody>
</table>

To use the external option the clad on the back of the circuit board must be cut between pins 1 and 2.

The Receiver Clock option jumpers are identified as JP21. The clock is normally jumpered to the internal clock but may be jumpered to the external receiver element timing signal or to the transmitter clock.

**Jumper JP21. Modem Receiver Clock Option**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal</td>
<td>2 Jumpered to 3 (normal)</td>
</tr>
<tr>
<td>External</td>
<td>2 Jumpered to 1</td>
</tr>
<tr>
<td>Transmitter</td>
<td>2 Jumpered to 4</td>
</tr>
</tbody>
</table>

To use the external option, the clad on the back of the circuit board must be cut between pins 2 and 3.

For applications requiring a small amount of external power at + and −12 volts, a jumper option JP15 provides power to pins 9 and 10 of the serial connector.

**Jumper JP15 + and −12 volt power option**

JP15 1-2 for −12 on pin 10  JP15 3-4 for +12 on pin 9

### 3.9 List 72H, 73H

The list function is handled by half of the SIO device at location 2H and it is designed to match EIA Data Set specifications. The interfacing chip is an MK3884/Z80A-SIO. Port 72H is data and Port 73H is Status and Control.

The list baud rate is selectable using the baud rate generator at Port EOH described in Section 3.13 of this manual. Connection to a printer is made through J3, a female DB25S connector. The following table describes pinouts for the printer:
Table 3-7. List I/O Pinouts

<table>
<thead>
<tr>
<th>Pin #</th>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GND</td>
<td>Chassis Ground</td>
</tr>
<tr>
<td>2</td>
<td>TxD</td>
<td>Transmit Data (in)</td>
</tr>
<tr>
<td>3</td>
<td>RxD</td>
<td>Receive Data (out)</td>
</tr>
<tr>
<td>4</td>
<td>RTS</td>
<td>Request to Send (in)</td>
</tr>
<tr>
<td>5</td>
<td>CTS</td>
<td>Clear to Send (out)</td>
</tr>
<tr>
<td>6</td>
<td>DSR</td>
<td>Data Set Ready (out)</td>
</tr>
<tr>
<td>7</td>
<td>GND</td>
<td>Signal Ground</td>
</tr>
<tr>
<td>9</td>
<td>+12</td>
<td>External Power - Jumper JP13 3-4</td>
</tr>
<tr>
<td>10</td>
<td>-12</td>
<td>External Power - Jumper JP13 1-2</td>
</tr>
<tr>
<td>15</td>
<td>TSET</td>
<td>Transmitter Signal Element Timing (out)</td>
</tr>
<tr>
<td>17</td>
<td>RSET</td>
<td>Receiver Signal Element Timing (out)</td>
</tr>
<tr>
<td>20</td>
<td>DTR</td>
<td>Data Terminal Ready (in)</td>
</tr>
<tr>
<td>21</td>
<td>SQD</td>
<td>Signal Quality Detector (Driven by a 22K resistor to +12V)</td>
</tr>
<tr>
<td>22</td>
<td>RI</td>
<td>Ring Indicator (Driven by a 22K resistor to +12V)</td>
</tr>
<tr>
<td>23</td>
<td>DSRS</td>
<td>Data Signal Rate Selector (Driven by a 22K resistor to +12V)</td>
</tr>
<tr>
<td>24</td>
<td>TSET</td>
<td>External Clock (in)</td>
</tr>
</tbody>
</table>

Clock Source Option. Jumper Pin Block JP22 gives the option of driving the list SIO from the internal Baud Rate generator or externally from Pin 24 on the list device connector.

Jumper JP22. List Device Clock Option

Internal | 2 Jumped to 1 (normal) |
External  | 2 Jumped to 3 |

To use the external option the cladding on the back of the circuit board must be cut between pins 1 and 2.

For applications requiring a small amount of external power at + and -12 volts, a jumper option JP13 provides power to pins 9 and 10 of the serial connector.

JP13 + and -12 volt power option

JP13 1-2 for -12 on pin 10  JP13 3-4 for +12 on pin 9
3.10 Floppy Disk 80H-83H

The floppy disk controller provides I/O to run the double-sided, double-density floppy disk drives. The controller chip is a Western Digital 1793B-02; refer to the data sheet for details on its operation.

The chip is normally run under DMA control as DMA device 1. Refer to section 3.1 "DMA Controller" for information. It can also be run under interrupt control; an interrupt request is driven from interrupt request signal DINT1 - refer to section 3.3 "Interrupt Controller" for operation.

Interface to the floppy disk is through Connector P5. The following table defines the pinouts; all signals are active low:

<table>
<thead>
<tr>
<th>Pin #</th>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>SPARE</td>
<td>Loading of head</td>
</tr>
<tr>
<td>4</td>
<td>HEAD LOAD</td>
<td>Timing Signal from Disk</td>
</tr>
<tr>
<td>6</td>
<td>SPARE</td>
<td>Disk Select One</td>
</tr>
<tr>
<td>8</td>
<td>INDEX PULSE</td>
<td>Disk Select Two</td>
</tr>
<tr>
<td>10</td>
<td>DS1</td>
<td>Turns on the Selected Motor</td>
</tr>
<tr>
<td>12</td>
<td>DS2</td>
<td>Direction in Toward Center of Disk (Low-Head Pulled in toward Center)</td>
</tr>
<tr>
<td>14</td>
<td>SPARE</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>MOTOR ON</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>DIRC IN</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>STEP</td>
<td>Track Step</td>
</tr>
<tr>
<td>22</td>
<td>WRITE DATA</td>
<td>Data Goes on Disk</td>
</tr>
<tr>
<td>24</td>
<td>WRITE GATE</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>TRACK 0</td>
<td>Detection of Track Zero for the Disk</td>
</tr>
<tr>
<td>28</td>
<td>WR PROTECT</td>
<td>Write Protect</td>
</tr>
<tr>
<td>30</td>
<td>READ DATA</td>
<td>Read Data</td>
</tr>
<tr>
<td>32</td>
<td>SIDE SELECT</td>
<td>Side Select</td>
</tr>
<tr>
<td>34</td>
<td>READY</td>
<td>Indicates to 1793 that disk drives are ready</td>
</tr>
<tr>
<td>1-33 ODD</td>
<td>GROUND</td>
<td>Ground Shield</td>
</tr>
</tbody>
</table>

An option is provided on the board for selecting control of Side Source Select:

**Jumper JP37 Side Select Source**

- Pin 2 jumpered to Pin 1: Side Select sourced from Control Latch (normal)
- Pin 2 jumpered to Pin 3: Side Select sourced from 1797 (Clad between 1 and 2 must be cut)
3.11 Status and Control AOH

The main CPU has a single byte stored in an 8 bit latch used to control key functions in the system. This latch is set to all 0's by a RESET. The memory location 0BH is reserved to store the status of this byte. This byte, written using an OUT instruction to AOH, has the following functions:

<table>
<thead>
<tr>
<th>Bit #</th>
<th>Signal Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>DS1</td>
<td>Drive Select (Drive A)</td>
</tr>
<tr>
<td>1</td>
<td>DS2</td>
<td>Drive Select (Drive B)</td>
</tr>
<tr>
<td>2</td>
<td>T/R</td>
<td>Parallel Port A Direction Control</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 = receive,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 = transmit</td>
</tr>
<tr>
<td>3</td>
<td>APURES</td>
<td>Reset the 9511 Arithmetic Processor</td>
</tr>
<tr>
<td>4</td>
<td>TG19</td>
<td>&quot;Track greater than 19,&quot; used by Write Precompensation</td>
</tr>
<tr>
<td>5</td>
<td>SIDE SELECT</td>
<td>Disk Side Select</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0 is bottom, 1 is top)</td>
</tr>
<tr>
<td>6</td>
<td>PROMEN</td>
<td>PROM Enable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(0 is PROM enable, 1 is PROM disable)</td>
</tr>
<tr>
<td>7</td>
<td>DISK RESET</td>
<td>Floppy Disk Controller Reset</td>
</tr>
</tbody>
</table>

Status is read by the processor at address AOH with the function of the bits as follows:

<table>
<thead>
<tr>
<th>Bit #</th>
<th>Signal Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>INTRQ</td>
<td>Disk Controller Done</td>
</tr>
<tr>
<td>1</td>
<td>HLT</td>
<td>Head Loaded</td>
</tr>
<tr>
<td>5</td>
<td>Spare</td>
<td>IC 5L Pin 13 (JP 36)</td>
</tr>
<tr>
<td>7</td>
<td>DRQ</td>
<td>Disk Controller Data Ready</td>
</tr>
</tbody>
</table>

A "1" indicates that the stated action has occurred.

3.12 Real Time Clock BOH,COH

The Real Time Clock (RTC) option provides the ASSOCIATE with time information ranging from thousandths of seconds to months. It is available for systems using "F" revision or later CPU boards.

The option is based on the MM58167 clock chip with rechargeable nicad batteries to keep the time information valid even when the computer is turned off or unplugged.
The RTC occupies a block of 32 addresses. These addresses B0H to COH are assigned as follows:

Table 3-11. Real Time Clock Functions

<table>
<thead>
<tr>
<th>Name</th>
<th>Address</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>ctrsm</td>
<td>OCOH</td>
<td>Counter - thousandths of seconds</td>
</tr>
<tr>
<td>ctrsth</td>
<td>OC1H</td>
<td>Counter - tenths and hundredths of seconds</td>
</tr>
<tr>
<td>ctrsec</td>
<td>OC2H</td>
<td>Counter - seconds</td>
</tr>
<tr>
<td>ctrmin</td>
<td>OC3H</td>
<td>Counter - minutes</td>
</tr>
<tr>
<td>ctrhr</td>
<td>OC4H</td>
<td>Counter - hours</td>
</tr>
<tr>
<td>ctrdow</td>
<td>OC5H</td>
<td>Counter - day of the week</td>
</tr>
<tr>
<td>ctrday</td>
<td>OC6H</td>
<td>Counter - day of the month</td>
</tr>
<tr>
<td>ctrmon</td>
<td>OC7H</td>
<td>Counter - months</td>
</tr>
<tr>
<td>latsm</td>
<td>OC8H</td>
<td>Latch - thousandths of seconds</td>
</tr>
<tr>
<td>latst</td>
<td>OC9H</td>
<td>Latch - tenths and hundredths of seconds</td>
</tr>
<tr>
<td>latsec</td>
<td>OCAH</td>
<td>Latch - seconds</td>
</tr>
<tr>
<td>latmin</td>
<td>OCBH</td>
<td>Latch - minutes</td>
</tr>
<tr>
<td>lathr</td>
<td>OCCB</td>
<td>Latch - hours</td>
</tr>
<tr>
<td>latdow</td>
<td>OCDH</td>
<td>Latch - day of the week</td>
</tr>
<tr>
<td>latday</td>
<td>OCEH</td>
<td>Latch - day of the month</td>
</tr>
<tr>
<td>latmon</td>
<td>OCFC</td>
<td>Latch - months</td>
</tr>
<tr>
<td>rtcisr</td>
<td>OBOH</td>
<td>Interrupt status register</td>
</tr>
<tr>
<td>rtcicr</td>
<td>OB1H</td>
<td>Interrupt control register</td>
</tr>
<tr>
<td>rtcctr</td>
<td>OB2H</td>
<td>Counter reset</td>
</tr>
<tr>
<td>rtccltr</td>
<td>OB3H</td>
<td>Latch reset</td>
</tr>
<tr>
<td>rtcst</td>
<td>OB4H</td>
<td>Status bit</td>
</tr>
<tr>
<td>rtccgo</td>
<td>OB5H</td>
<td>&quot;Go&quot; command</td>
</tr>
<tr>
<td>rtcstby</td>
<td>OB6H</td>
<td>Standby interrupt</td>
</tr>
<tr>
<td>rtcctst</td>
<td>OB7H</td>
<td>Test mode</td>
</tr>
</tbody>
</table>

Two software programs are provided with the Real Time Clock option. These are: TRTC.COM, a routine that allows the operator to manually set and test all the parameters of the clock operation, and TIME.COM, a sample routine to display the current time on the monitor.

The system time may be set or reset using the 'T' option in the TRTC program. In response to prompts from this routine the user sets the current hour, minute, second, day, day of week, and month. Once set, the current time is always available by doing an input from the proper counter as shown in the above table or by executing the program 'TIME'. Day 1 of the week is Sunday.

The NICAD batteries and charging circuit provide a 10 to one ratio for backup versus on time. This ratio is set by the 330 ohm charge resistor R7. When the system is first used with new batteries, the AC power should be left on continuously for the first week. The voltage at pin 24 of the 58167 device should be greater than 3.4 volts with the AC power off. If the voltage is lower than this, the batteries need to be charged or replaced. 500 hours of back-up is available from fully charged batteries.

The Real Time Clock chip can be programmed to generate an interrupt at a preset time. To enable the interrupt, jumper pins 11 and 12 together on
jumper pad JP42 located between 7L and 7K. This ties the regular interrupt output of the MM58167 to Trigger 3 interrupt input of the CTC Interrupt controller. The CTC channel 3 must be set for a count of 1 to act as an interrupt controller.

Jumper Pin Block JP42

<table>
<thead>
<tr>
<th>Pin 11 jumpered to 12</th>
<th>RTC Interrupt Enabled</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pin 11 open to 12</td>
<td>RTC Interrupt Disabled</td>
</tr>
</tbody>
</table>

(1) Normal

The standby interrupt output from the device, normally not used is available on Jumper Pin JP32.

A variable capacitor (C6) on the board allows minor adjustments of the clock rate of the Real Time Clock. The capacitor is located at the edge of the board next to the Clock chip at location 3S. The procedure for setting the time is as follows:

Check time against a standard to determine the number of seconds per day of error. Then select the 'S' option in the TRTC program. A series of hex numbers will be provided which should stabilize to a value between FFEOH and 001FH. This number is the Real Time Clock offset in seconds per day. If the Clock is running too slow, adjust C6 to increase this number by one for each second per day the Clock was slow (0001 is greater than FFFF). After adjustment, 90 seconds is required for the number displayed to stabilize.

3.13 **Baud Rate Generator DOH, EOH**

Baud Rate Generation is done by a dual software controlled timer whose source crystal frequency is 4.9152 MHz.

The normal assignment of serial ports to the Baud Rate Generator is shown in Table 3-12. This assignment may be optionally changed as described in the section for each of the individual devices.

Table 3-12. Normal Baud Rate Generator Setup

<table>
<thead>
<tr>
<th>Serial Function</th>
<th>Baud Rate Generator Address</th>
<th>Default Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>List Device</td>
<td>EOH</td>
<td>9600</td>
</tr>
<tr>
<td>Modem</td>
<td>DOH</td>
<td>300</td>
</tr>
<tr>
<td>Network</td>
<td>DOH</td>
<td>300</td>
</tr>
<tr>
<td>Video</td>
<td>External</td>
<td>48000</td>
</tr>
</tbody>
</table>

Each port of the baud rate generator uses only four bits of the byte written to it. The port at DOH uses only the four least significant bits. EOH Port uses the four most significant. In the following table the left and right nybbles are shown to be duplicates, which simplifies operations: an "OUT" instruction with the data from this table to port DOH or EOH will set the baud rate for that device:
### Table 3-13. Baud Rate with 16x clock

<table>
<thead>
<tr>
<th>Byte</th>
<th>Baud Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>50</td>
</tr>
<tr>
<td>11</td>
<td>75</td>
</tr>
<tr>
<td>22</td>
<td>110</td>
</tr>
<tr>
<td>33</td>
<td>134.5</td>
</tr>
<tr>
<td>44</td>
<td>150</td>
</tr>
<tr>
<td>55</td>
<td>300</td>
</tr>
<tr>
<td>66</td>
<td>600</td>
</tr>
<tr>
<td>77</td>
<td>1200</td>
</tr>
<tr>
<td>88</td>
<td>1800</td>
</tr>
<tr>
<td>99</td>
<td>2000</td>
</tr>
<tr>
<td>AA</td>
<td>2400</td>
</tr>
<tr>
<td>BB</td>
<td>3600</td>
</tr>
<tr>
<td>CC</td>
<td>4800</td>
</tr>
<tr>
<td>DD</td>
<td>7200</td>
</tr>
<tr>
<td>EE</td>
<td>9600</td>
</tr>
<tr>
<td>FF</td>
<td>19200</td>
</tr>
</tbody>
</table>

For example in the PROM Monitor the procedure to set the List device to 1200 baud is:

```
Q0:EO,77 (the monitor provides the colon)
```

or to set the modem port to 9600 baud:

```
Q0:DO,EE
```

The baud rate generator is reset upon power up, CTRL-SHIFT-RST, or a manual reset from the rear panel. It is not changed by a CTRL-C warm reboot.

### 3.14 Ram On Control

An Output instruction to location DOH will clear the restart Flip/Flop thereby enabling RAM and restricting PROM addressing to locations F800H to FFFFH. No software provision is made to disable RAM once it is enabled.

### 3.15 Serial I/O Assignment

The CP/M Device assignments are described in the table below. The left side is the logical name and the right side of the assignment being the CP/M function.

<table>
<thead>
<tr>
<th>CON:</th>
<th>Monitor display and keyboard</th>
</tr>
</thead>
<tbody>
<tr>
<td>LST:</td>
<td>Printer device output</td>
</tr>
<tr>
<td>PUN:</td>
<td>Punch device output</td>
</tr>
<tr>
<td>RDR:</td>
<td>Reader device input</td>
</tr>
</tbody>
</table>

This assignment is controlled by two bytes referred to as IOBYTE and EXTIOB stored in memory locations 0003 and 0004. The default values are
11101000 (E8 HEX) and 00000000. Each of the CP/M functions is controlled by two bits out of each byte. The following table shows all the possible device assignments for each of the functions:

The bit pattern in IOBYTE determines the physical port assigned to each logical device.

<table>
<thead>
<tr>
<th>LST:</th>
<th>PUN:</th>
<th>RDR:</th>
<th>CON:</th>
<th>Back Panel Connector</th>
</tr>
</thead>
<tbody>
<tr>
<td>00xxxxxx</td>
<td>xx00xxxx</td>
<td>xxxx00xx</td>
<td>xxxxxxx00</td>
<td>Internal CRT</td>
</tr>
<tr>
<td>62hex</td>
<td>62hex</td>
<td>62hex</td>
<td>62hex</td>
<td>-</td>
</tr>
<tr>
<td>01xxxxxx</td>
<td>xx01xxxx</td>
<td>xxxx01xx</td>
<td>xxxxxxx01</td>
<td>Network Port</td>
</tr>
<tr>
<td>60hex</td>
<td>60hex</td>
<td>60hex</td>
<td>60hex</td>
<td>-</td>
</tr>
<tr>
<td>10xxxxxx</td>
<td>xx10xxxx</td>
<td>xxxx10xx</td>
<td>xxxxxxx10</td>
<td>Modem Port</td>
</tr>
<tr>
<td>70hex</td>
<td>70hex</td>
<td>70hex</td>
<td>70hex</td>
<td>-</td>
</tr>
<tr>
<td>11xxxxxx</td>
<td>xx11xxxx</td>
<td>xxxx11xx</td>
<td>xxxxxxx11</td>
<td>List Port</td>
</tr>
<tr>
<td>72hex</td>
<td>72hex</td>
<td>72hex</td>
<td>72hex</td>
<td>-</td>
</tr>
</tbody>
</table>

The bit pattern in IOBEXT determines the driver assigned to each logical device.

<table>
<thead>
<tr>
<th>LST:</th>
<th>PUN:</th>
<th>RDR:</th>
<th>CON:</th>
<th>Driver Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>00xxxxxx</td>
<td>xx00xxxx</td>
<td>xxxx00xx</td>
<td>xxxxxxx00</td>
<td>Std is the standard assignments</td>
</tr>
<tr>
<td>iostd</td>
<td>iostd</td>
<td>iostd</td>
<td>iostd</td>
<td>-</td>
</tr>
<tr>
<td>01xxxxxx</td>
<td>xx01xxxx</td>
<td>xxxx01xx</td>
<td>xxxxxxx01</td>
<td>SQ indicates a Control-S and Control-Q hand-shake output</td>
</tr>
<tr>
<td>iosq</td>
<td>iosq</td>
<td>iosq</td>
<td>remote</td>
<td>-</td>
</tr>
<tr>
<td>10xxxxxx</td>
<td>xx10xxxx</td>
<td>xxxx10xx</td>
<td>xxxxxxx10</td>
<td>Diab is driver for Diablo printer</td>
</tr>
<tr>
<td>iodiab</td>
<td>iodiab</td>
<td>iodiab</td>
<td>iostd</td>
<td>-</td>
</tr>
<tr>
<td>11xxxxxx</td>
<td>xx11xxxx</td>
<td>xxxx11xx</td>
<td>xxxxxxx11</td>
<td>Remote means that the Modem Port is parallel with the CRT</td>
</tr>
<tr>
<td>null</td>
<td>null</td>
<td>null</td>
<td>null</td>
<td>-</td>
</tr>
</tbody>
</table>

3-17
The ZBO-SIO is a multi-function peripheral component designed to satisfy a wide variety of serial data communications requirements. The normal set up of the SIO is 1 start bit, 8 data bits, 1 stop bit, no parity, and x16 clock divider. The SIO has 8 write registers and 3 read registers whose functions are defined below:

**WRITE REGISTER 0**

- 0000 REGISTER 0
- 0001 REGISTER 1
- 0010 REGISTER 2
- 0011 REGISTER 3
- 0100 REGISTER 4
- 0101 REGISTER 5
- 0110 REGISTER 6
- 0111 REGISTER 7
- 0000 NULL CODE
- 0001 SEND ABORT (SDLC)
- 0010 RESET EXT. STATU5 INTERUPTS
- 0011 CHANNEL RESET
- 1010 ERROR REGISTER
- 1100 RETURN FROM INT (CH-A ONLY)
- 0000 NULL CODE
- 0001 RESET Rx CRC CHECKER
- 0010 RESET Tx CRC GENERATOR
- 0011 Tx UNDERRUN/EOM LATCH

**WRITE REGISTER 1**

- 0000 EXT. INT ENABLE
- 0001 Tx INT ENABLE
- 1100 STATUS AFFECTS VECTOR (CH-B ONLY)
- 1101 Rx INT ON FIRST CHARACTER ONLY
- 1110 INT ON ALL Rx CHARACTERS (PARITY AFFECTS VECTOR)
- 1111 INT ON ALL Rx CHARACTERS (PARITY DOES NOT AFFECT VECTOR)

**WRITE REGISTER 2**

- 0000 WAIT READY ON R/T
- 0001 WAIT FN READY FN
- 0010 WAIT READY ENABLE

**WRITE REGISTER 3**

- 0000 Rx ENABLE
- 0001 LOAD INHIBIT
- 0010 LOAD ADDRESS (SDLC)
- 1000 Rx CRC ENABLE
- 1001 Rx UNDERRUN/EOM LATCH
- 0100 Rx 5 BITS/CHARACTER
- 0101 Rx 7 BITS/CHARACTER
- 1000 Rx 6 BITS/CHARACTER
- 1001 Rx 8 BITS/CHARACTER

**WRITE REGISTER 4**

- 0000 PARITY ENABLE
- 0001 PARITY EVEN/ODD
- 0010 SYNC MESSAGES ENABLE
- 0011 1 STOP BIT/CHARACTER
- 1011 2 STOP BITS/CHARACTER
- 0000 8 BITS SYNCC CHARACTERS
- 0001 16 BIT SYNCC CHARACTERS
- 0010 SDLC MODE (0011110 Sync Flag)
- 1000 EXTERNAL SYNCC MODE
- 0000 X1 CLOCK MODE
- 0001 X16 CLOCK MODE
- 0010 X32 CLOCK MODE
- 1000 X64 CLOCK MODE

**WRITE REGISTER 5**

- 0000 Tx CRC ENABLE
- 0001 RTS
- 0010 SDLC CRC/16
- 0011 Tx ENABLE
- 1000 SEND BREAK
- 0000 Rx 5 BITS (OR LESS) PER CHARACTERS
- 0001 Rx 7 BITS/CHARACTER
- 0010 Rx 6 BITS/CHARACTER
- 1000 Rx 8 BITS/CHARACTER
**3.17 System Interfacing (Cables)**

Several cables are available to facilitate connecting the ASSOCIATE to external devices. Several of the most commonly used cables are listed below for reference. All signal line nomenclature used is defined according to EIA RS232C conventions. The (in) or (out) is with reference to the microcomputer hardware, for example any line labeled (out) will have data coming from the CPU processor board.
CA-60 ASSOCIATE to NEC Printer

This cable is used to connect the ASSOCIATE List Port to the NEC Printer. This cable is an adapter to the NEC supplied cable. It is used so that no changes are required in the NEC cable. The wiring in the cable is one to one except for pin 20 as follows:

CA-60 Cable - ASSOCIATE to NEC Spinwriter

<table>
<thead>
<tr>
<th>DB25P (List Port)</th>
<th>Function</th>
<th>DB25S (NEC Cable)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pin #</td>
<td></td>
<td>Pin #</td>
</tr>
<tr>
<td>1</td>
<td>Chas Ground</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Tx Data (in)</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>Rx Data (out)</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>Request to Send (in)</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>Clear to Send (out)</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>Data Set Ready (out)</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>Sig Ground</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>Rec’d line sig (out)</td>
<td>8</td>
</tr>
<tr>
<td>20</td>
<td>Data Terminal Ready (in)</td>
<td>19</td>
</tr>
</tbody>
</table>

CA-61 ASSOCIATE to TTY Model 40 Line Printer

This cable is used to connect the ASSOCIATE List Port to the Teletype Model 40 line Printer. The wiring is one to one except for pin 4 as follows:

CA-61 ASSOCIATE to TTY Model 40 Line Printer

<table>
<thead>
<tr>
<th>DB25P (List Port)</th>
<th>DB25S (TTY 40)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pin #</td>
<td>Pin #</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>14</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
</tr>
</tbody>
</table>

CA-62 ASSOCIATE to MODEM

This cable is used to connect the ASSOCIATE Modem Port to a modem. This is a standard EIA - RS232C extension cable. It can also be used to connect the List or Network port to any RS232C Data Terminal Equipment such as an ASSOCIATE Terminal or the modem port of another ASSOCIATE. The wiring in the cable is one to one as follows:
CA-62 ASSOCIATE to MODEM Cable

DB25S (MODEM Port) | DB25P (MODEM)
---|---
Pin # | Function | Pin # |
1 | Chas Ground | 1 |
2 | Tx Data (out) | 2 |
3 | Rx Data (in) | 3 |
4 | Request to Send (out) | 4 |
5 | Clear to Send (in) | 5 |
6 | Data Set Ready (in) | 6 |
7 | Sig Ground | 7 |
8 | Rec'd line sig (in) | 8 |
15 | Trans Clock (in) | 15 |
17 | Rec Clock (in) | 17 |
20 | Data Term Ready (out) | 20 |
24 | Trans Clock (out) | 24 |

CA-63 ASSOCIATE to Terminal

This cable is used to connect the ASSOCIATE Network or List Port to a Terminal device with a female connector such as an ADM3A. The wiring in the cable is one to one as follows:

CA-63 ASSOCIATE to Terminal Cable

DB25P (NETWORK or LIST PORT) | DB25P (MODEM)
---|---
Pin # | Function | Pin # |
1 | Chas Ground | 1 |
2 | Tx Data (in) | 2 |
3 | Rx Data (out) | 3 |
4 | Request to Send (in) | 4 |
5 | Clear to Send (out) | 5 |
6 | Data Set Ready (out) | 6 |
7 | Sig Ground | 7 |
8 | Rec'd line sig (out) | 8 |
15 | Trans Clock (out) | 15 |
17 | Rec Clock (out) | 17 |
20 | Data Term Ready (in) | 20 |
24 | Trans Clock (in) | 24 |

CA-65 ASSOCIATE to TI Model 820 Printer

This cable is used to connect the ASSOCIATE List Port to the Texas Instrument Model 820 Printer. The wiring is one to one except for pin 4 as follows:
CA-65 ASSOCIATE to TI Model 820 Printer

DB25P (List Port)                  DB25P (TI 820)
Pin #   Function           Pin #
1       Chas Ground        1
2       Tx Data (in)       2
3       Rx Data (out)      3
4       Request to Send (in) 11
6       Data Set Ready (out) 6
7       Sig Ground         7
8       Carrier Det (out)  8
20      Data Term Ready (in) 20

The Secondary Request To Send signal from the printer is tied to the List port Request To Send.

The TI 820 is a programmable printer. To program it for operation with the ASSOCIATE, lift the cover, slide the "configure" switch to the local position and proceed with the programming. Start by typing in a <return> and then a <tab> which will put the printer in full duplex reverse channel mode. The suggested baud rate is 9600 programmable by typing 28 <return> <tab>. No other parameters need changing. Set the "configure" switch back to its normal position for operation. See the TI manual for complete programming details.

QUME Printer Operation

The QUME Sprint 5 printer will connect directly or through a CA-62 cable to the List port. The Qume should be set up as follows:

Front Panel Switches          Inside Switches
Reset                        Both switched to the left
Baud Rate                    side of the unit.
- 1200
Duplex                       Full
Parity                       Mark
- Modem = off
- Rate = Hi
Reset
Auto LF                      Off
Twintall                     Standard
Char Spac                     your choice
Form Length                   11 is standard
Top of Form
Form Feed

A Qume Sprint 9 is connected to the ASSOCIATE by a 1:1 cable. The WordStar setup for the Sprint 9 is the same as the Sprint 5. For normal Sprint 9 operation with the ASSOCIATE, the internal options are set as follows:

A-8 A-7 A-6 A-5 A-4 A-3 A-2 A-1 B-8 B-7 B-6 B-5 B-4 B-3 B-2 B-1
on on on off off on on on on on on on on on off

3-22
The video processor board provides control of the ASSOCIATE CRT and keyboard. The video processor board is built to enable the CRT to run as a stand alone terminal. A separate manual is available describing the 10T Terminal version of the ASSOCIATE.

Key chips on the video processor board are the Intel 8085 Microprocessor and Motorola 6845 CRT Controller. Refer to the appropriate manufacturer's manual for detailed information on these chips. The processor has 1K of local RAM for temporary storage. The 2K video RAM is dual-ported, so either the 8085 or CRTC can address it. The CRT controller controls read-out of RAM into the video generator, and the 8085 processor can both read and write into video RAM.

Two 2K PROM chips provide program storage and character generation for the video processor. The program is stored in the first 2K of the memory of the 8085 and begins execution immediately upon power up. The second PROM is used as a character generator for the $8 \times 12$ dot matrix character set. Standard PROMs are TMS2516's. TMS2532's can be used with no hardware changes if more extensive CRT control programs are needed. Use of a TMS2532 for alternate character sets requires IC30 pin 18 to be jumpered to an I/O pin on the 8155. A program for generation of custom character sets is available from Data Technology.

All data going to the main processor is transferred via a serial line, with the video processor providing the clock signals to the main CPU Board. The video processor board has two parallel data channels, one for the keyboard and one for the utility port. The utility port is a general purpose parallel port which can be used for a second keyboard.

The video processor board has on it a bell normally activated by a "Control C" character.

The video processor board is mounted, vertically, in the back of the CRT. Access to the video processor board is accomplished by removing the front panel assembly as described in Section 8 of this manual.

Sections in this chapter include descriptions of the video processor board's Internal Architecture (including a memory map), keyboard I/O, Utility I/O, CPU Interface, video generator interface, and operational characteristics.
4.1 Internal Architecture

The peripheral devices on the video processor board are addressed by memory mapping as shown in the following table:

Table 4-1. Video Processor Board Memory Map

<table>
<thead>
<tr>
<th>Address</th>
<th>Device</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000-07FF</td>
<td>CPU PROM</td>
<td>Stores Program for CPU</td>
</tr>
<tr>
<td>1000-13FF</td>
<td>CPU RAM</td>
<td>CPU RAM - Scratch area for 8085 Microprocessor</td>
</tr>
<tr>
<td>2000-27FF</td>
<td>VRAM</td>
<td>Video RAM</td>
</tr>
<tr>
<td>3000</td>
<td>RESET OUT</td>
<td>Generates a RESET to the main CPU board</td>
</tr>
<tr>
<td>4000-40FF</td>
<td>8155 RAM</td>
<td>CPU RAM scratch area</td>
</tr>
<tr>
<td>4800-4803</td>
<td>8155 I/O</td>
<td>Utility and keyboard I/O</td>
</tr>
<tr>
<td>4804-4805</td>
<td>8155 TIMER</td>
<td>Baud Rate Generator</td>
</tr>
<tr>
<td>5000-5001</td>
<td>8251 USART</td>
<td>Interface to the Main CPU Board</td>
</tr>
<tr>
<td>6000</td>
<td>BELL</td>
<td>Bell</td>
</tr>
<tr>
<td>7000-7001</td>
<td>CRTC</td>
<td>CRT Controller</td>
</tr>
</tbody>
</table>

An option is available to allow substitution of an 8156 for the 8155 I/O device.

8155/8156 Option

- CE jumpered to 5 8155 device (normal)
- CE jumpered to 6 8156 device

For operation with the 8156, the clad between CE and 5 must be cut.

The CRT Controller and I/O devices drive the processor through the interrupts. The following table specifies the interrupt priority:

Table 4-2. Interrupts and Priority

<table>
<thead>
<tr>
<th>Priority</th>
<th>Interrupt</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VSYNC</td>
<td>TRAP</td>
<td>Vertical Sync</td>
</tr>
<tr>
<td>2</td>
<td>RxRDY</td>
<td>R7.5</td>
<td>Receive Data from CPU</td>
</tr>
<tr>
<td>3</td>
<td>TxRDY</td>
<td>R6.5</td>
<td>Receive Data from Utility</td>
</tr>
<tr>
<td>4</td>
<td>INTA</td>
<td>R5.5</td>
<td>Receive Data from Keyboard</td>
</tr>
<tr>
<td>5</td>
<td>INTB</td>
<td>INTR</td>
<td>Transmitter to CPU Ready</td>
</tr>
</tbody>
</table>
4.2 Keyboard I/O

Keyboard I/O is handled through the 8155 I/O and Timer. When a key is pushed, it generates a strobe signal which clocks the keyboard data into the 8155. The 8155 will in turn generate both an interrupt and a data ready flag to the processor. Under normal software the Video Processor is interrupt driven, but could alternatively operate in a polled fashion. Refer to the Intel data sheet for complete information on the 8155.

Refer to the keyboard table in Chapter 7 of this reference manual for a complete description of data transmitted by different keys.

The following table describes the various pinout functions of the keyboard I/O. A 20 pin 3M ribbon cable is used to attach the keyboard. This cable also supplies power for the keyboard electronics.

<table>
<thead>
<tr>
<th>Pin #</th>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ASTB</td>
<td>Data available from keyboard</td>
</tr>
<tr>
<td>3</td>
<td>ARDY</td>
<td>Keyboard transmit enable</td>
</tr>
<tr>
<td>5</td>
<td>PA7</td>
<td>Data bit 7</td>
</tr>
<tr>
<td>7</td>
<td>PA6</td>
<td>Data bit 6</td>
</tr>
<tr>
<td>9</td>
<td>PA5</td>
<td>Data bit 5</td>
</tr>
<tr>
<td>11</td>
<td>PA4</td>
<td>Data bit 4</td>
</tr>
<tr>
<td>13</td>
<td>PA3</td>
<td>Data bit 3</td>
</tr>
<tr>
<td>15</td>
<td>PA2</td>
<td>Data bit 2</td>
</tr>
<tr>
<td>17</td>
<td>PA1</td>
<td>Data bit 1</td>
</tr>
<tr>
<td>19</td>
<td>PA0</td>
<td>Data bit 0</td>
</tr>
<tr>
<td>2-8 Even</td>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>10,12</td>
<td>+5V</td>
<td>Power</td>
</tr>
<tr>
<td>14,16</td>
<td>+12V</td>
<td>Power</td>
</tr>
<tr>
<td>18,20</td>
<td>-12V</td>
<td>Power</td>
</tr>
</tbody>
</table>

4.3 Utility I/O

The utility I/O is designed to be a general purpose port which can be used to attach an additional keyboard.

The following table describes the various pinout functions of the utility port. A 20 pin 3M ribbon cable is used to attach to the port:
Table 4-4. Utility I/O Pinouts

<table>
<thead>
<tr>
<th>Pin #</th>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BSTB</td>
<td>Data available from keyboard</td>
</tr>
<tr>
<td>3</td>
<td>BRDY</td>
<td>Keyboard transmit enable</td>
</tr>
<tr>
<td>5</td>
<td>PA7</td>
<td>Data bit 7</td>
</tr>
<tr>
<td>7</td>
<td>PA6</td>
<td>Data bit 6</td>
</tr>
<tr>
<td>9</td>
<td>PA5</td>
<td>Data bit 5</td>
</tr>
<tr>
<td>11</td>
<td>PA4</td>
<td>Data bit 4</td>
</tr>
<tr>
<td>13</td>
<td>PA3</td>
<td>Data bit 3</td>
</tr>
<tr>
<td>15</td>
<td>PA2</td>
<td>Data bit 2</td>
</tr>
<tr>
<td>17</td>
<td>PA1</td>
<td>Data bit 1</td>
</tr>
<tr>
<td>19</td>
<td>PA0</td>
<td>Data bit 0</td>
</tr>
<tr>
<td>2-8</td>
<td>Even</td>
<td>Ground</td>
</tr>
<tr>
<td>10,12</td>
<td>+5V</td>
<td>Power</td>
</tr>
<tr>
<td>14,16</td>
<td>+12V</td>
<td>Power</td>
</tr>
<tr>
<td>18,20</td>
<td>-12V</td>
<td>Power</td>
</tr>
</tbody>
</table>

4.4 CPU Interface

An 8251 chip is used for the communication interface between the video processor board and the main CPU; it follows RS232 standard serial protocol. By having standard RS232 protocol the CRT/keyboard can function as a standalone terminal, similar in capability to an enhanced ADM 3A in terms of interaction with another computer.

The connection between the video processor board and the CPU Board is made through a 20-pin ribbon cable which connects to P6 on the CPU Board and J1 on the video board.

The following table describes Pin functions for the CPU Interface. All direction references are to the host CPU.

Table 4-5. Interface Pinout

<table>
<thead>
<tr>
<th>Pin #</th>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,2,3,4,7</td>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>5</td>
<td>DTR</td>
<td>Data Terminal Ready (to CPU)</td>
</tr>
<tr>
<td>6</td>
<td>DSR</td>
<td>Data Set Ready (from CPU)</td>
</tr>
<tr>
<td>9</td>
<td>RxD</td>
<td>Receive Data (from CPU)</td>
</tr>
<tr>
<td>11</td>
<td>TxD</td>
<td>Transmit Data (to CPU)</td>
</tr>
<tr>
<td>13</td>
<td>RTS</td>
<td>Request to Send (from CPU)</td>
</tr>
<tr>
<td>14</td>
<td>CTS</td>
<td>Clear to Send (to CPU)</td>
</tr>
<tr>
<td>15</td>
<td>RESOT</td>
<td>Reset Out (to CPU)</td>
</tr>
<tr>
<td>17</td>
<td>CLOCK</td>
<td>Clock (to CPU)</td>
</tr>
<tr>
<td>18</td>
<td>RESIN</td>
<td>Reset in (from CPU)</td>
</tr>
</tbody>
</table>

A reset from the keyboard (control shift RST) does not reset the video processor, it merely transmits the reset through to the main CPU. See Section 2.2.2 for further information on resets.
Two options are available on the Serial I/O clock. The first located below IC10 is for the clock source signal:

Clock Source Option

- BDR jumpered to IN: Internal clock
- BRR jumpered to EX: External clock

The clock source is normally internal; to use an external clock the jumper clad on the back of the circuit board must be cut.

The second option selects the frequency of the input into the Baud Rate divider. The source timing normally is 2.304 MHz but may be jumpered to 4.608 MHz. The faster source clock will allow operation at a higher baud rate but does exceed specification for the standard 8155 timer.

Clock Rate Option

- CK jumpered to 02: 0.2 MHz source (normal)
- CK jumpered to 04: 4 MHz source

To implement the 4 MHz option the clad to 02 must be cut on the back of the circuit board.

4.5 Character Generator and Video Interface

Video display information from the Video RAM is passed through a 2716 PROM used as a character generator. It is then combined with cursor information and transmitted to the CRT via connector P3 as EIA composite video. The board is configured to allow operation with a 2732 PROM thereby doubling the character set. To operate with a 2732 pin 18 must be connected to the proper level. It is normally connected to ground.

<table>
<thead>
<tr>
<th>Pin #</th>
<th>Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VID</td>
<td>Video</td>
</tr>
<tr>
<td>2</td>
<td>VER</td>
<td>Vertical Sync</td>
</tr>
<tr>
<td>3</td>
<td>HOR</td>
<td>Horizontal Sync</td>
</tr>
<tr>
<td>4</td>
<td>GND</td>
<td>Ground</td>
</tr>
</tbody>
</table>
Three options are available in the video generator section. They are:

**Composite Video**
- E jumpered to D: Horizontal and Vertical Sync add to video
- E jumpered to F: No sync information added to video (normal)

To add sync information, jumper between F and E.

The following options have been changed on differing levels of ECOs. See the ECO list for details — Section 2.2.6

**Attribute Option**
- CA7 jumpered to INT: The eight bit attribute is foreground/background (normal)
- CA7 jumpered to REV: The eight bit attribute is reverse/normal video

To operate the eight bit attribute in the reverse/normal mode the clad between CA7 jumpered to INT must be cut.

**Video Mode**
- B jumpered to A: Characters are displayed white on black background (normal)
- B jumpered to C: Characters are displayed black on white background

To change the display mode the clad between B and A must be cut. It is possible to jumper this option to unused pins on the 8155 and switch modes under software control.

### 4.6 Light Pen Interface

Interface to a standard light pen may be made by connecting to P4 on the Video Processor board. The light pen input may be either active high or active low depending on the invert option as follows:

**Light Pen Invert Option**
- Jumpered to High: Active high signal
- Jumpered to Low: Active lo signal

Chapter 5

VIDEO PROCESSOR OPERATION

This Chapter describes software and control characteristics of the video processor. The following subjects are covered: CRT Display Format, Control Sequences, Use of "Soft" keys, CRT Memory Data Areas, CRT PROM Entry Points, and Down-Loading a Program into the CRT.

5.1 CRT Display Format

The CRT has an 80 column, 24 line display area. Characters may be displayed in background (normal intensity) or foreground (high intensity or reverse video) with background as the default display mode.

Foreground/Background can be invoked in two ways:

The first method is to use control sequences to select either foreground or background mode. Once a mode has been selected, all displayable characters sent to the CRT will be displayed in that mode.

The second method is to use the high-order bit of characters sent to the CRT. If a character with the high order bit is received by the CRT, it will be displayed in the opposite of the currently selected mode.

The CRT can also display a twenty-fifth line which does not scroll with the other twenty-four. This line can display up to eighty characters or 72 characters plus time-of-day clock. Control of information displayed in the twenty-fifth line is described later in this Chapter.

The CRT is capable of displaying 128 different characters. Each character can be displayed in either foreground or background. 96 of these characters, the "printable" characters, hex values 20H to 7FH, are displayed by simply sending the character to the CRT. To display one of the other 32 special characters 00H to 1FH, an ESCAPE character is sent followed by a byte with the value of 60H added to the value of the character to display. The character will display in foreground or background, whichever is the currently selected mode. If following the ESCAPE character, the value EOH added to the value of the special character is sent, the character will display in the opposite of the currently selected mode.

The standard set of special characters is designed for use in communication applications. These are shown on the table on the next page. An alternative set of graphic characters (see chart next page) can be ordered by specifying the graphic character set with your order.

The 32 special characters may also be displayed with the alternate attribute set by setting the eighth bit. For example the character 80H will display as Ny in the opposite attribute.
CHARGEN is a program that allows design of custom character sets. With this program the user can create custom character set files that can then be programmed into the character generator PROM. Contact the Data Technology marketing department for further information.

The 32 special characters are listed in the following table:

Table 5.1A Special Character display

<table>
<thead>
<tr>
<th>Character</th>
<th>Bytes to Transmit</th>
<th>Name</th>
<th>Standard Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>00H</td>
<td>ESC 60H</td>
<td>ESC 96</td>
<td>NULL</td>
</tr>
<tr>
<td>01H</td>
<td>ESC 61H</td>
<td>ESC 97</td>
<td>SOH</td>
</tr>
<tr>
<td>02H</td>
<td>ESC 62H</td>
<td>ESC 98</td>
<td>STX</td>
</tr>
<tr>
<td>03H</td>
<td>ESC 63H</td>
<td>ESC 99</td>
<td>ETX</td>
</tr>
<tr>
<td>04H</td>
<td>ESC 64H</td>
<td>ESC 100</td>
<td>EOT</td>
</tr>
<tr>
<td>05H</td>
<td>ESC 65H</td>
<td>ESC 101</td>
<td>ENQ</td>
</tr>
<tr>
<td>06H</td>
<td>ESC 66H</td>
<td>ESC 102</td>
<td>ACK</td>
</tr>
<tr>
<td>07H</td>
<td>ESC 67H</td>
<td>ESC 103</td>
<td>BELL</td>
</tr>
<tr>
<td>08H</td>
<td>ESC 68H</td>
<td>ESC 104</td>
<td>BACKSPACE</td>
</tr>
<tr>
<td>09H</td>
<td>ESC 69H</td>
<td>ESC 105</td>
<td>HORIZONTAL TAB</td>
</tr>
<tr>
<td>0A</td>
<td>ESC 6AH</td>
<td>ESC 106</td>
<td>LINE FEED</td>
</tr>
<tr>
<td>0B</td>
<td>ESC 6BH</td>
<td>ESC 107</td>
<td>VERTICAL TAB</td>
</tr>
<tr>
<td>0C</td>
<td>ESC 6CH</td>
<td>ESC 108</td>
<td>FORM FEED</td>
</tr>
<tr>
<td>0D</td>
<td>ESC 6DH</td>
<td>ESC 109</td>
<td>CARRIAGE RETURN</td>
</tr>
<tr>
<td>0E</td>
<td>ESC 6EH</td>
<td>ESC 110</td>
<td>SO</td>
</tr>
<tr>
<td>0F</td>
<td>ESC 6FH</td>
<td>ESC 111</td>
<td>SI</td>
</tr>
<tr>
<td>10</td>
<td>ESC 70H</td>
<td>ESC 112</td>
<td>DLE</td>
</tr>
<tr>
<td>11</td>
<td>ESC 71H</td>
<td>ESC 113</td>
<td>DC1</td>
</tr>
<tr>
<td>12</td>
<td>ESC 72H</td>
<td>ESC 114</td>
<td>DC2</td>
</tr>
<tr>
<td>13</td>
<td>ESC 73H</td>
<td>ESC 115</td>
<td>DC3</td>
</tr>
<tr>
<td>14</td>
<td>ESC 74H</td>
<td>ESC 116</td>
<td>DC4</td>
</tr>
<tr>
<td>15</td>
<td>ESC 75H</td>
<td>ESC 117</td>
<td>NAK</td>
</tr>
<tr>
<td>16</td>
<td>ESC 76H</td>
<td>ESC 118</td>
<td>SYN</td>
</tr>
<tr>
<td>17</td>
<td>ESC 77H</td>
<td>ESC 119</td>
<td>ETB</td>
</tr>
<tr>
<td>18</td>
<td>ESC 78H</td>
<td>ESC 120</td>
<td>CAN</td>
</tr>
<tr>
<td>19</td>
<td>ESC 79H</td>
<td>ESC 121</td>
<td>EM</td>
</tr>
<tr>
<td>1A</td>
<td>ESC 7AH</td>
<td>ESC 122</td>
<td>SUB</td>
</tr>
<tr>
<td>1B</td>
<td>ESC 7BH</td>
<td>ESC 123</td>
<td>ESC</td>
</tr>
<tr>
<td>1C</td>
<td>ESC 7CH</td>
<td>ESC 124</td>
<td>FS</td>
</tr>
<tr>
<td>1D</td>
<td>ESC 7DH</td>
<td>ESC 125</td>
<td>GS</td>
</tr>
<tr>
<td>1E</td>
<td>ESC 7EH</td>
<td>ESC 126</td>
<td>RS</td>
</tr>
<tr>
<td>1F</td>
<td>ESC 7FH</td>
<td>ESC 127</td>
<td>US</td>
</tr>
</tbody>
</table>

The alternate graphics set is shown on the following page.
5.2 Control Sequences

The CPU sends commands to the CRT in the form of control sequences. Control sequence may be in the form of single or multiple characters; multiple-character control sequences begin with an ESCAPE character. If an invalid character follows the ESCAPE, both characters are ignored, and the CRT resumes processing with the next character.

### Single-Character Control Sequences

<table>
<thead>
<tr>
<th>HEX VALUE</th>
<th>DECIMAL VALUE</th>
<th>ASCII NAME</th>
<th>Function in CRT</th>
</tr>
</thead>
<tbody>
<tr>
<td>07</td>
<td>07</td>
<td>BELL</td>
<td>Sound audible tone.</td>
</tr>
<tr>
<td>08</td>
<td>08</td>
<td>BS</td>
<td>Move cursor left one column. If the cursor is on the first column of a row other than the top row, it will move to the last column of the next higher row.</td>
</tr>
<tr>
<td>0A</td>
<td>10</td>
<td>LF</td>
<td>Move cursor down one row. If the cursor is on the last row, the screen scrolls up one line.</td>
</tr>
<tr>
<td>0B</td>
<td>11</td>
<td>VT</td>
<td>Move cursor up one row. If the cursor is on the top row, it does not move.</td>
</tr>
<tr>
<td>0C</td>
<td>12</td>
<td>FF</td>
<td>Move the cursor right one column. If the cursor is in the last column of a row other than the bottom row, it moves to the first column on the next lower row.</td>
</tr>
<tr>
<td>0D</td>
<td>13</td>
<td>CR</td>
<td>Move cursor to the first column of the current row.</td>
</tr>
<tr>
<td>0E</td>
<td>14</td>
<td>SO</td>
<td>Unlock keyboard. This only functions if the keyboard lock/unlock is enabled -- see below.</td>
</tr>
<tr>
<td>0F</td>
<td>15</td>
<td>SI</td>
<td>Lock keyboard. This only functions if the keyboard lock/unlock is enabled -- see below.</td>
</tr>
<tr>
<td>1A</td>
<td>26</td>
<td>SUB</td>
<td>Clear screen and send cursor to home position.</td>
</tr>
<tr>
<td>1B</td>
<td>27</td>
<td>ESC</td>
<td>Lead-in character for multi-character control sequence.</td>
</tr>
<tr>
<td>1E</td>
<td>28</td>
<td>RS</td>
<td>Send cursor to home position.</td>
</tr>
</tbody>
</table>
Multi-Character Control Sequences

The ESCAPE lead-in character is not shown. A '(p)' indicates that the sequence takes further characters as parameters. An '(r)' indicates that the CRT will return one or more characters to the host CPU in response to the control sequence.

<table>
<thead>
<tr>
<th>HEX VALUE</th>
<th>DECIMAL VALUE</th>
<th>ASCII NAME</th>
<th>Function in CRT</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>33</td>
<td>!</td>
<td>Set Background. The characters that follow will display at normal intensity.</td>
</tr>
<tr>
<td>22</td>
<td>34</td>
<td>&quot;</td>
<td>Set Foreground. The characters that follow will display at high intensity.</td>
</tr>
<tr>
<td>23</td>
<td>35</td>
<td>#</td>
<td>Clear screen from cursor position to end of screen.</td>
</tr>
<tr>
<td>24</td>
<td>36</td>
<td>$</td>
<td>Clear screen from cursor position to end of line.</td>
</tr>
<tr>
<td>25</td>
<td>37</td>
<td>%</td>
<td>Insert line at cursor position. The line with the cursor and all lines below the cursor scroll down one line. A blank line is inserted at the cursor position and the cursor is positioned at the beginning of the new line.</td>
</tr>
<tr>
<td>26</td>
<td>38</td>
<td>&amp;</td>
<td>Delete line at cursor position. The line with the cursor is deleted and all lines below the cursor scroll up. A blank line is inserted at the bottom of the screen. The cursor remains on the same line of the screen, but moves to the beginning of the line.</td>
</tr>
<tr>
<td>27</td>
<td>39</td>
<td>(p)</td>
<td>Set baud rate. The parameter is a hexadecimal character (0-F) in ASCII which selects the baud rate to be used between the CRT and the CPU from the following table:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ASCII Digit</th>
<th>Hex Digit</th>
<th>Baud Rate Selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>30H</td>
<td>Top speed</td>
</tr>
<tr>
<td>1</td>
<td>31H</td>
<td>48000</td>
</tr>
<tr>
<td>2</td>
<td>32H</td>
<td>38400</td>
</tr>
<tr>
<td>3</td>
<td>33H</td>
<td>28800</td>
</tr>
<tr>
<td>4</td>
<td>34H</td>
<td>19200</td>
</tr>
<tr>
<td>5</td>
<td>35H</td>
<td>9600</td>
</tr>
<tr>
<td>6</td>
<td>36H</td>
<td>4800</td>
</tr>
<tr>
<td>7</td>
<td>37H</td>
<td>2400</td>
</tr>
<tr>
<td>8</td>
<td>38H</td>
<td>1200</td>
</tr>
<tr>
<td>9</td>
<td>39H</td>
<td>300</td>
</tr>
</tbody>
</table>
28 40 (p) Set time of day. The parameter is a string with the following form:

    HHMMSS <CR>
    HHMM <CR>
    HH <CR>

where HH sets the hour, MM the minute, and SS the seconds. Each field is sent as two ASCII digits. The <CR> is required as a delimiter and will not be processed with its usual meaning.

For example, the parameter string (in ASCII) to set 10:30:15 a.m. is:

    <esc> (103015 <CR>

Any omitted parameter will be set to zero.

To have the Time of Day appear on the screen, three separate functions must be done: Set Time of Day, Turn on the 25th Line, and Turn Clock Display on. For example to set and display the time to 10:30:15 a.m. from the Operating System:

    <esc> (103015 <cr> Set Time of Day
    <esc> 8 <cr> Turn on 25th Line
    <esc> 2 <cr> Turn on Clock Display

29 41 ) (p) Write 25th line and turn it on. The parameter string has the following form:

    <eighty characters>

    or

    <fewer than eighty characters><CR>

The 25th line buffer is cleared to all spaces, and then the characters are written into the buffer. If more than eighty characters are sent without a carriage return, the first eighty will be written into the 25th line. Any following characters will be treated with their normal significance, meaning that they will probably be written to the screen at the cursor position.

2A 42 * Turn off 25th line. The data is retained and may be redisplayed by using the sequence to turn the 25th line on

2B 43 (p,r) Read Memory. This sequence causes the CRT to return to the host CPU a string of hexadecimal characters in ASCII representing the contents of memory starting at the requested address. Thus for each
byte of memory data requested two characters are returned. The first character of each pair represents the most significant nibble of each byte.

The parameter string is:

```
<address> <length>
```

The "address" consists of four ASCII hexadecimal characters representing 2 bytes of address; the low order address byte, most significant nibble first.

The "length" is a single ASCII hexadecimal character requesting the number of bytes to be returned. If the character is 0 sixteen bytes are read.

The string returned to the host CPU has the following form:

```
<data> <CR>
```

where each byte of data is transmitted as two ASCII hexadecimal characters.

**2C 44 , (p)** Write Memory. This sequence causes the CRT to write data sent by the host CPU into the CRT memory. The parameter string has the following form:

```
<address> <data> <delimiter>
```

where "address" consists of four ASCII hexadecimal characters representing 2 bytes of address; the low order address, most significant nibble first.

The "data" consists of pairs of ASCII hexadecimal characters, with each pair representing 1 byte of data. The first character of each pair represents the most significant nibble of each byte.

Memory write continues until the CRT receives a non-HEX character.

**2D 45 - (p)** Transfer to address. The parameter string has the form:

```
<address>
```

where "address" is the same as for the memory read and memory write sequences.

The CRT then calls to the address specified. The user may use a RET instruction to return.

**2E 46 . (r)** Read cursor address. The CRT sends three ASCII characters to the host CPU:
The relationship between the sent characters and the value of the position is as follows:

\[
\begin{align*}
\text{row} &= \langle y\text{-pos} \rangle - 20H \\
\text{column} &= \langle x\text{-pos} \rangle - 20H
\end{align*}
\]

where all values are bytes in hex.

**2F 47 /** Reset soft keys. This sequence resets all soft keys to their default sequences.

**30 48 0** Enable keyboard lock/unlock. This sequence enables the keyboard lock and unlock sequences (ASCII characters 50 and 51) and unlocks the keyboard.

The CRT is initialized with the keyboard unlocked and the keyboard lock disabled. The keyboard will not lock until an enable keyboard lock sequence is sent to the CRT, followed by a lock keyboard sequence.

**31 49 1** Disable keyboard lock/unlock. This sequence disables the keyboard lock and unlock sequences (ASCII characters 50 and 51) and unlocks the keyboard.

**32 50 2** Clock display on. The clock is displayed on the 25th line, at the right. The 25th line must be enabled using the sequence \(<\text{esc}>8\) or \(<\text{esc}>\).

**33 51 3** Clock display off. The clock display is turned off. The rest of the 25th line is unaffected.

**36 52 6 (r)** Read light pen. The last CRT display address sensed by the light pen is returned in a string of the form

\(<x\text{-pos}> <y\text{-pos}> <\text{CR}>\)

The first eight bytes are hexadecimal characters representing two two-byte values: the current display screen base address 'wwxx' and the last light pen screen address selected 'yyzz'.

For example, the string

\(10235624<\text{CR}>\)

indicates that the current screen address is 2301H, and that the last address sensed by the light pen is 2456H. The row and column are calculated as follows:
(screen displacement of light pen) =
[(light pen address selected)
- (screen base address)
- (fudge factor)] MOD 2000H

(row selected) = (displacement) / 80

(column selected) = (displacement) MOD 80

37 55 7 (p) Right to Left Ticker-tape display in the 25th line. The parameter string consists of a single character. The leftmost 72 characters of the 25th line are shifted one character to the left and the parameter character is inserted at the 72nd position. Continuous messages may be displayed by using this sequence repeatedly.

38 56 8 Turn on 25th line. The 25th line is displayed. No change is made to its data.

3C 60 < (p;r) Read characters from the display. This sequence causes the CRT to return to the host CPU the string of characters occupying the display memory beginning at the current cursor position. The parameter string consists of a single hexadecimal digit <length> which indicates how many characters are to be read back. If <length> is zero, sixteen bytes are returned.

3D 61 = (p) Position cursor. The parameter string has the following form:

<y-pos><x-pos>

where <y-pos> and <x-pos> are single characters representing the row and column of the new cursor position. The translation between <y-pos> and <x-pos> and the row and column numbers is:

<y-pos> = row + 20H
<x-pos> = column + 20H

where all values are bytes in hex.

Special Control Sequences

The following memory write sequences can be used to cause the described effects:

<ESC>,26402500<CR> Disables the "toggle" function of the RST key.

<ESC>,26402520<CR> Re-enables the "toggle" function of the RST key.
<ESC>,1940xx<CR> Changes the number of minutes until the screen goes blank to xx (value in hexadecimal).

<ESC>,3E40xx<CR> Sets the minimum time between characters sent to main CPU from video board to xx msec. This is implemented on CRT PROM 1.8 and later.

<ESC>,3F4001<CR> Disables Auto-~w1ne function. This is implemented on CRT PROM 1.8 and later.

<ESC>,0011210A09220070COC9<CR><ESC>=0011 Turns off cursor blink. This is an example of a call to a program in the CRT RAM.

<ESC>=0000 Completely resets video board by forcing CRT processor to branch to location zero.

5.3 Use of Soft Keys

The soft (or programmable) keys are the numeric pad keys and the top row of keys (F0-F9, cursor movement and HOME, but not RST). These are the keys whose transmitted codes may be modified.

"Soft" keys generate default codes, but they may be individually "loaded" with a variable-length string of characters. After a soft key has been loaded, whenever that key is depressed, the CRT sends to the host CPU the string that was loaded into that key, as if the operator had entered the string of characters.

These keys may be loaded with the KEYLOADER Programmable Function Key program (PFK) or as described below.

Each soft key can be loaded with up to four character strings. To illustrate, if four strings were loaded into the soft key F0, A string may be selected by entering either F0, shift-F0, ctrl-F0, or ctrl-shift-F0. These four combinations are called the four "levels" of the soft key. Or a soft key may be immune to shift and ctrl by loading the same character string into one or more levels of that key. As will be explained later, in this case, only one copy of the string needs to exist in the RAM storage of the CRT.

All the soft keys have a standard default code which is a single character with the high order bit set. For example, the default code for each numeric pad (no control, no shift) key is the character shown on the keytop. The default code for the cursor control keys is the character that causes the cursor to move in the direction shown on the keytop. See the Table 6.1 for the default codes for each soft key.

Each level of each soft key generates a specific code between 0 and 127, called the "soft key code." This code, the value listed in Table 6.1, minus 80H, is used to index into the soft key pointer area. This area is a table of entries (two bytes per entry, low-order byte stored first) located at 1000H. Given a soft key code of xxH, its entry can be found at 1000H + 2*(xxH). The value can also be found in Table 5.2. The entry taken from the table is processed as follows:
If the high-order byte of the entry is zero, the entry is in the "reset" state. The CRT will send to the host CPU the soft key code which is its default value.

If the high-order byte of the entry is negative (i.e., its high-order bit is 1), the CRT will "call" the absolute value of the entry, assumed to be the entry point of a user defined subroutine in the CRT.

If the high-order byte of the entry is positive (i.e., its high-order bit is zero), the entry is used as the pointer to a string of characters, called the "soft key string."

The CRT will send to the host CPU consecutive characters starting with the first byte of the soft key string. A hexadecimal "80" character marks the end of the soft key string. When this character is found, it is not sent to the host.

The following examples show how to use the programmable features of the function keys.

Example 1: Program Function Key FO to generate a "D (04H).

The soft key pointer for FO is at 10COH.
The soft key string is 0080. 
Assume the location of the soft key string will be 1108H.
Then the following sequence will set the key:
<ESC>,C0100811<CR>  (set soft key pointer)
<ESC>,08110480<CR>  (write soft key string)

Example 2: Program Function Key Cntl-FO and ctrl-shift-FO to both generate the string "RUN XYZ<CR>".

The soft key pointer for ctrl-FO is at 1040H.
The soft key pointer for ctrl-shift-FO is at 1000H.
The soft key string is 52564E2058595A0D80.
Assume the location of the soft key string will be 1100H.
Then the following sequence will set the keys:
<ESC>,40100011<CR>  (set soft key pointer for ctrl-FO)
<ESC>,00100011<CR>  (set soft key pointer for ctrl-shift-FO)
<ESC>,00152564E2058595A0D80<CR>  (write soft key string)

Example 3: Program Function Key FO is to generate a "D (04H).
Shift-FO is to generate "W"D (1704H).

The soft key pointer for FO is at 10COH.
The soft key pointer for shift-FO is at 1080H.
The soft key string for FO is 0480.
The soft key string for shift-FO is 170480.
Since the string for FO is a right-hand substring of the string for shift-FO, the two strings can share memory.
Assume the location of the soft key string for shift-FO will
be 1107H.
The location of the soft key string for FO will be 1108H.
Then the following sequence will set the keys:

\(<\text{ESC}},\text{C0100811<CR}>\) (set soft key pointer for FO)
\(<\text{ESC}},\text{80100811<CR}>\) (set soft key pointer for shift-FO)
\(<\text{ESC}},\text{071170480<CR}>\) (write soft key string)

<table>
<thead>
<tr>
<th>Key</th>
<th>Normal</th>
<th>Shift</th>
<th>Ctrl</th>
<th>Ctrl-Shift</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>keys</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fn key 0</td>
<td>10C0</td>
<td>1080</td>
<td>1040</td>
<td>1000</td>
</tr>
<tr>
<td>Fn key 1</td>
<td>10C2</td>
<td>1082</td>
<td>1042</td>
<td>1002</td>
</tr>
<tr>
<td>Fn key 2</td>
<td>10C4</td>
<td>1084</td>
<td>1044</td>
<td>1004</td>
</tr>
<tr>
<td>Fn key 3</td>
<td>10C6</td>
<td>1086</td>
<td>1046</td>
<td>1006</td>
</tr>
<tr>
<td>Fn key 4</td>
<td>10C8</td>
<td>1088</td>
<td>1048</td>
<td>1008</td>
</tr>
<tr>
<td>Fn key 5</td>
<td>10CA</td>
<td>108A</td>
<td>104A</td>
<td>100A</td>
</tr>
<tr>
<td>Fn key 6</td>
<td>10CC</td>
<td>108C</td>
<td>104C</td>
<td>100C</td>
</tr>
<tr>
<td>Fn key 7</td>
<td>10CE</td>
<td>108E</td>
<td>104E</td>
<td>100E</td>
</tr>
<tr>
<td>Fn key 8</td>
<td>10DE</td>
<td>109E</td>
<td>105E</td>
<td>101E</td>
</tr>
<tr>
<td>Fn key 9</td>
<td>10D2</td>
<td>1092</td>
<td>1052</td>
<td>1012</td>
</tr>
<tr>
<td>Cursor</td>
<td>movement</td>
<td>keys</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Back</td>
<td>1010</td>
<td>1090</td>
<td>1050</td>
<td>10D0</td>
</tr>
<tr>
<td>Down</td>
<td>1014</td>
<td>1094</td>
<td>1054</td>
<td>10D4</td>
</tr>
<tr>
<td>Up</td>
<td>1016</td>
<td>1096</td>
<td>1036</td>
<td>10D6</td>
</tr>
<tr>
<td>Left</td>
<td>1018</td>
<td>1098</td>
<td>1058</td>
<td>10D8</td>
</tr>
<tr>
<td>Home</td>
<td>103C</td>
<td>109C</td>
<td>101C</td>
<td>10DC</td>
</tr>
<tr>
<td>Numeric pad</td>
<td>keys</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1060</td>
<td>10A0</td>
<td>10E0</td>
<td>1020</td>
</tr>
<tr>
<td>1</td>
<td>1062</td>
<td>10A2</td>
<td>10E2</td>
<td>1022</td>
</tr>
<tr>
<td>2</td>
<td>1064</td>
<td>10A4</td>
<td>10E4</td>
<td>1024</td>
</tr>
<tr>
<td>3</td>
<td>1066</td>
<td>10A6</td>
<td>10E6</td>
<td>1026</td>
</tr>
<tr>
<td>4</td>
<td>1068</td>
<td>10A8</td>
<td>10E8</td>
<td>1028</td>
</tr>
<tr>
<td>5</td>
<td>106A</td>
<td>10AA</td>
<td>10EA</td>
<td>102A</td>
</tr>
<tr>
<td>6</td>
<td>106C</td>
<td>10AC</td>
<td>10EC</td>
<td>102C</td>
</tr>
<tr>
<td>7</td>
<td>106E</td>
<td>10AE</td>
<td>10EE</td>
<td>102E</td>
</tr>
<tr>
<td>8</td>
<td>1070</td>
<td>10B0</td>
<td>10F0</td>
<td>1030</td>
</tr>
<tr>
<td>9</td>
<td>1072</td>
<td>10B2</td>
<td>10F2</td>
<td>1032</td>
</tr>
<tr>
<td>-</td>
<td>105A</td>
<td>10BA</td>
<td>10FA</td>
<td>107A</td>
</tr>
<tr>
<td>+</td>
<td>1056</td>
<td>10B6</td>
<td>10F6</td>
<td>1076</td>
</tr>
<tr>
<td>*</td>
<td>105C</td>
<td>10BC</td>
<td>10FC</td>
<td>107C</td>
</tr>
<tr>
<td>ENTER</td>
<td>101A</td>
<td>109A</td>
<td>103A</td>
<td>10DA</td>
</tr>
<tr>
<td>&quot;under-0&quot;</td>
<td>1074</td>
<td>10B4</td>
<td>10F4</td>
<td>1034</td>
</tr>
<tr>
<td>&quot;under-ENTER&quot;</td>
<td>1078</td>
<td>10B8</td>
<td>10F8</td>
<td>1038</td>
</tr>
</tbody>
</table>
5.4 CRT Memory Allocation

The following data areas are defined:

<table>
<thead>
<tr>
<th>HEX ADDRESS</th>
<th>NAME</th>
<th>USAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0021H-0023H</td>
<td>CRT PROM Version #</td>
<td></td>
</tr>
<tr>
<td>1000H-10FFH</td>
<td>Soft Key pointer area.</td>
<td></td>
</tr>
<tr>
<td>1100H-13A8H</td>
<td>User area.</td>
<td></td>
</tr>
<tr>
<td>13BOH-13FFH</td>
<td>Buffer for 25th line.</td>
<td></td>
</tr>
<tr>
<td>4000H-4005H</td>
<td>CPU out queue control block.</td>
<td></td>
</tr>
<tr>
<td>4006H-400BH</td>
<td>Keyboard in queue control block.</td>
<td></td>
</tr>
<tr>
<td>400CH</td>
<td>XPOS</td>
<td>Column number of cursor (0..79).</td>
</tr>
<tr>
<td>400DH</td>
<td>YPOS</td>
<td>Row number of cursor (0..23).</td>
</tr>
<tr>
<td>400EH</td>
<td>INTEN</td>
<td>Intensity flag. 80H = Normal intensity, 00H = High intensity.</td>
</tr>
<tr>
<td>400FH-4010H</td>
<td>ABSIO</td>
<td>Absolute cursor address</td>
</tr>
<tr>
<td>4013H-4014H</td>
<td>LITEPEN</td>
<td>Last light pen address sensed</td>
</tr>
<tr>
<td>4011H-4012H</td>
<td>CRTBAS</td>
<td>Current start of screen display area</td>
</tr>
<tr>
<td>4013H-4014H</td>
<td>CRSPOS</td>
<td>Relative cursor address</td>
</tr>
<tr>
<td>4019H</td>
<td>TIMEOUTOFF</td>
<td>Number of minutes till screen blank</td>
</tr>
<tr>
<td>4021H</td>
<td>L25ENAB</td>
<td>Non-zero enables 25th line</td>
</tr>
<tr>
<td>4022H</td>
<td>CLOCKFLAG</td>
<td>Non-zero enables clock display</td>
</tr>
<tr>
<td>4023H</td>
<td>KBLFLAG</td>
<td>Non-zero locks keyboard</td>
</tr>
<tr>
<td>402FH</td>
<td>HOURS</td>
<td>Hours counter</td>
</tr>
<tr>
<td>4030H</td>
<td>MINS</td>
<td>Minutes counter</td>
</tr>
<tr>
<td>4031H</td>
<td>SECS</td>
<td>Seconds counter</td>
</tr>
<tr>
<td>4032H</td>
<td>C60THS</td>
<td>60ths of a second counter</td>
</tr>
<tr>
<td>4036H-403DH</td>
<td>UPCLOCK</td>
<td>ASCII unpacked clock field HH:MM:SS</td>
</tr>
</tbody>
</table>

5.5 CRT PROM Entry Points

The following entry points into the PROM code are defined for down-loaded programs to use:

<table>
<thead>
<tr>
<th>HEX ADDR</th>
<th>Function performed</th>
</tr>
</thead>
<tbody>
<tr>
<td>07E5H</td>
<td>Fills memory starting at HL with the value in 'B', for as many bytes as specified in 'C'. If the value in 'C' is zero, 256 bytes will be filled.</td>
</tr>
<tr>
<td>07E8H</td>
<td>Set cursor position from relative cursor position in HL.</td>
</tr>
<tr>
<td>07EBH</td>
<td>Set cursor position from x and y values in XPOS and YPOS. (x = column 0..79; y = row 0..23)</td>
</tr>
<tr>
<td>07EEH</td>
<td>Put character in 'C' into CRT memory at cursor location and advance cursor.</td>
</tr>
<tr>
<td>Opcode</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>07F1H</td>
<td>Send character in 'C' to host as two ASCII bytes representing a hex value.</td>
</tr>
<tr>
<td>07F4H</td>
<td>Returns, in 'C', the hex value of the hexascii digit in 'C'. C-latch set if not valid hex digit.</td>
</tr>
<tr>
<td>07F7H</td>
<td>Put character in 'C' into output queue.</td>
</tr>
<tr>
<td>07FAH</td>
<td>Return with character in 'C' from queue given by HL. Z-latch is set if no characters were in queue.</td>
</tr>
<tr>
<td>07FDH</td>
<td>Branch, using value in 'A' as an index, into control sequences branch table.</td>
</tr>
<tr>
<td>Bit</td>
<td>ASCII Code Chart</td>
</tr>
<tr>
<td>-----</td>
<td>-----------------</td>
</tr>
<tr>
<td></td>
<td>CONTROL</td>
</tr>
<tr>
<td>0</td>
<td>NUL</td>
</tr>
<tr>
<td>0</td>
<td>SOH</td>
</tr>
<tr>
<td>0</td>
<td>STX</td>
</tr>
<tr>
<td>0</td>
<td>ETX</td>
</tr>
<tr>
<td>0</td>
<td>EOT</td>
</tr>
<tr>
<td>1</td>
<td>ENQ</td>
</tr>
<tr>
<td>1</td>
<td>ACK</td>
</tr>
<tr>
<td>1</td>
<td>BEL</td>
</tr>
<tr>
<td></td>
<td>BS</td>
</tr>
<tr>
<td>1</td>
<td>HT</td>
</tr>
<tr>
<td>1</td>
<td>LF</td>
</tr>
<tr>
<td>1</td>
<td>VT</td>
</tr>
<tr>
<td>1</td>
<td>FF</td>
</tr>
<tr>
<td>1</td>
<td>CR</td>
</tr>
<tr>
<td>1</td>
<td>SO</td>
</tr>
<tr>
<td>1</td>
<td>SI</td>
</tr>
</tbody>
</table>

Note that either "LOY" column may be used for the XLOY byte, since bit 5 is not used.
Chapter 6

KEYBOARD

The ASSOCIATE keyboard is a selectric style keyboard with enhancements. It interfaces to the video processor board; for details (see Section 4.2, "Keyboard I/O").

Two keys have "local" functions which cause an immediate operation by the CRT processor. These keys are:

Local clear (control-delete), clears the CRT screen without affecting the host CPU. The 25th line remains unchanged.

Local clear (control-shift-delete), clears the CRT screen and the 25th line. The clock display remains unchanged.

Toggle key (RST, shift-RST, and control-RST) stops/stops the transfer of data to the CPU by toggling the Clear to Send (CTS) to the CPU.

Reset host key (Control-Shift-RST) unconditionally resets the host CPU.

The following features are also standard with the keyboard:

1) **Redefinition of Keyboard** - Keyboard PROM can be changed to change the codes generated by keyboard characters.

2) **Electro-Capacitive Keyboard** - The ASSOCIATE has an electro-capacitive keyboard. No mechanical contact is required to enter a character; a change of capacitance is detected on the underside of the board and that enters the character. Pressing a key changes the capacitance. All capacitive sensing is handled by a keyboard controller on the keyboard circuit board.

3) **Repeat Functions** - When a key is pressed the character is sent immediately to the video processor. If the key is held down for longer than 1/2 second the character is repeated at the rate of 10 characters per second.

4) **CRT 25th Line** - The CRT's 25th line can be downloaded in order to create prompts directly above the keyboard.

5) **Selectric Format** - The ASSOCIATE Keyboard is a standard selectric format.

6) **Accounting Style Numeric Pad** - The keyboard has an accounting style numeric pad with double wide zero and an ENTER key (the default value is the same as RETURN). The keypad keys are soft keys that can be downloaded from the host CPU as described in Chapter 5.
7) **Special Purpose Keys** - Several special purpose keys, such as escape, tilde, carat, and insert, are on the keyboard. All 128 ASCII codes (0 to 7FH) can be generated from the keyboard.

8) **Cap Lock Key** - The keyboard has a CAP LOCK key. The CAP LOCK causes all alpha characters to be transmitted as capital letters. It does not affect operation of the shift for non-alpha characters.

9) **Function Keys** - Four separate levels are possible on the function keys: NORMAL, SHIFT, CONTROL, and SHIFT CONTROL. Additional functions can be downloaded from the host CPU as described in Chapter 5.

10) **Cursor Control** - There are four cursor control keys (LEFT, RIGHT, UP, DOWN) plus a HOME key. Cursor control keys do not alter information, they reposition the cursor in the corresponding manner. HOME sends the cursor to the upper left corner of the screen. These five function keys can also be downloaded from the host CPU.

The table on the following page defines the eight standard combinations for each key in terms of the ASCII code generated:
## TABLE 6.1 KEYBOARD CODE CHART

<table>
<thead>
<tr>
<th>Key</th>
<th>Key Number</th>
<th>Normal</th>
<th>Shift</th>
<th>Ctrl</th>
<th>Ctrl-Shift</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>57</td>
<td>61</td>
<td>41</td>
<td>01</td>
<td>01</td>
</tr>
<tr>
<td>B</td>
<td>79</td>
<td>62</td>
<td>42</td>
<td>02</td>
<td>02</td>
</tr>
<tr>
<td>C</td>
<td>77</td>
<td>63</td>
<td>43</td>
<td>03</td>
<td>03</td>
</tr>
<tr>
<td>D</td>
<td>59</td>
<td>64</td>
<td>44</td>
<td>04</td>
<td>04</td>
</tr>
<tr>
<td>E</td>
<td>41</td>
<td>65</td>
<td>45</td>
<td>05</td>
<td>05</td>
</tr>
<tr>
<td>F</td>
<td>60</td>
<td>66</td>
<td>46</td>
<td>06</td>
<td>06</td>
</tr>
<tr>
<td>G</td>
<td>61</td>
<td>67</td>
<td>47</td>
<td>07</td>
<td>07</td>
</tr>
<tr>
<td>H</td>
<td>62</td>
<td>68</td>
<td>48</td>
<td>08</td>
<td>08</td>
</tr>
<tr>
<td>I</td>
<td>46</td>
<td>69</td>
<td>49</td>
<td>09</td>
<td>09</td>
</tr>
<tr>
<td>J</td>
<td>63</td>
<td>6A</td>
<td>4A</td>
<td>0A</td>
<td>0A</td>
</tr>
<tr>
<td>K</td>
<td>64</td>
<td>6B</td>
<td>4B</td>
<td>0B</td>
<td>0B</td>
</tr>
<tr>
<td>L</td>
<td>65</td>
<td>6C</td>
<td>4C</td>
<td>0C</td>
<td>0C</td>
</tr>
<tr>
<td>M</td>
<td>81</td>
<td>6D</td>
<td>4D</td>
<td>0D</td>
<td>0D</td>
</tr>
<tr>
<td>N</td>
<td>80</td>
<td>6E</td>
<td>4E</td>
<td>0E</td>
<td>0E</td>
</tr>
<tr>
<td>O</td>
<td>47</td>
<td>6F</td>
<td>4F</td>
<td>0F</td>
<td>0F</td>
</tr>
<tr>
<td>P</td>
<td>48</td>
<td>70</td>
<td>50</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Q</td>
<td>39</td>
<td>71</td>
<td>51</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>R</td>
<td>42</td>
<td>72</td>
<td>52</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>S</td>
<td>58</td>
<td>73</td>
<td>53</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>T</td>
<td>43</td>
<td>74</td>
<td>54</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>U</td>
<td>45</td>
<td>75</td>
<td>55</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>V</td>
<td>78</td>
<td>76</td>
<td>56</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>W</td>
<td>40</td>
<td>77</td>
<td>57</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>X</td>
<td>76</td>
<td>78</td>
<td>58</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>Y</td>
<td>44</td>
<td>79</td>
<td>59</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>Z</td>
<td>75</td>
<td>7A</td>
<td>5A</td>
<td>1A</td>
<td>1A</td>
</tr>
<tr>
<td>!</td>
<td>20</td>
<td>31</td>
<td>21</td>
<td>31</td>
<td>31</td>
</tr>
<tr>
<td>@</td>
<td>21</td>
<td>32</td>
<td>40</td>
<td>32</td>
<td>00</td>
</tr>
<tr>
<td>#</td>
<td>22</td>
<td>33</td>
<td>23</td>
<td>33</td>
<td>23</td>
</tr>
<tr>
<td>$</td>
<td>23</td>
<td>34</td>
<td>24</td>
<td>34</td>
<td>24</td>
</tr>
<tr>
<td>%</td>
<td>24</td>
<td>35</td>
<td>25</td>
<td>35</td>
<td>25</td>
</tr>
<tr>
<td>`</td>
<td>25</td>
<td>36</td>
<td>27</td>
<td>36</td>
<td>27</td>
</tr>
<tr>
<td>&amp;</td>
<td>26</td>
<td>37</td>
<td>26</td>
<td>37</td>
<td>26</td>
</tr>
<tr>
<td>*</td>
<td>27</td>
<td>38</td>
<td>2A</td>
<td>38</td>
<td>2A</td>
</tr>
<tr>
<td>(</td>
<td>28</td>
<td>39</td>
<td>28</td>
<td>39</td>
<td>28</td>
</tr>
<tr>
<td>)</td>
<td>29</td>
<td>30</td>
<td>29</td>
<td>30</td>
<td>29</td>
</tr>
<tr>
<td>Space</td>
<td>90</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>Return</td>
<td>68</td>
<td>0D</td>
<td>0D</td>
<td>0D</td>
<td>0D</td>
</tr>
<tr>
<td>Bksp</td>
<td>32</td>
<td>08</td>
<td>08</td>
<td>08</td>
<td>08</td>
</tr>
<tr>
<td>Tab</td>
<td>38</td>
<td>09</td>
<td>09</td>
<td>09</td>
<td>09</td>
</tr>
<tr>
<td>ESC 1</td>
<td>19</td>
<td>1B</td>
<td>7C</td>
<td>1B</td>
<td>7C</td>
</tr>
<tr>
<td>^</td>
<td>37</td>
<td>5E</td>
<td>7E</td>
<td>1E</td>
<td>7E</td>
</tr>
<tr>
<td>DEL CLR</td>
<td>55</td>
<td>7F</td>
<td>7F</td>
<td>DF</td>
<td>BF</td>
</tr>
</tbody>
</table>
### Keyboard

<table>
<thead>
<tr>
<th>Key</th>
<th>Number</th>
<th>Normal</th>
<th>Shift</th>
<th>Ctrl</th>
<th>Ctrl-Shift</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>30</td>
<td>2D</td>
<td>5F</td>
<td>2D</td>
<td>1F</td>
</tr>
<tr>
<td>+</td>
<td>31</td>
<td>3D</td>
<td>2B</td>
<td>3D</td>
<td>2B</td>
</tr>
<tr>
<td>(</td>
<td>49</td>
<td>5B</td>
<td>7B</td>
<td>5B</td>
<td>7B</td>
</tr>
<tr>
<td>)</td>
<td>50</td>
<td>5D</td>
<td>7D</td>
<td>1D</td>
<td>7D</td>
</tr>
<tr>
<td>;</td>
<td>66</td>
<td>3B</td>
<td>5C</td>
<td>3B</td>
<td>1C</td>
</tr>
<tr>
<td>:</td>
<td>67</td>
<td>3A</td>
<td>22</td>
<td>3A</td>
<td>22</td>
</tr>
<tr>
<td>,</td>
<td>82</td>
<td>2C</td>
<td>3C</td>
<td>2C</td>
<td>3C</td>
</tr>
<tr>
<td>&gt;</td>
<td>83</td>
<td>2E</td>
<td>3E</td>
<td>2E</td>
<td>3E</td>
</tr>
<tr>
<td>/</td>
<td>84</td>
<td>2F</td>
<td>3F</td>
<td>2F</td>
<td>3F</td>
</tr>
</tbody>
</table>

**Function keys:**

| Fn key 0 | 1 | E0 | C0 | A0 | 80 |
| Fn key 1 | 2 | E1 | C1 | A1 | 81 |
| Fn key 2 | 3 | E2 | C2 | A2 | 82 |
| Fn key 3 | 4 | E3 | C3 | A3 | 83 |
| Fn key 4 | 5 | E4 | C4 | A4 | 84 |
| Fn key 5 | 8 | E5 | C5 | A5 | 85 |
| Fn key 6 | 9 | E6 | C6 | A6 | 86 |
| Fn key 7 | 10 | E7 | C7 | A7 | 87 |
| Fn key 8 | 11 | EF | CF | AF | 8F |
| Fn key 9 | 12 | E9 | C9 | A9 | 89 |

**Cursor movement keys:**

| Back | 13 | 88 | C8 | A8 | E8 |
| Down | 14 | 8A | CA | AA | EA |
| Up   | 15 | 8B | CB | 9B | EB |
| Left | 16 | 8C | CC | AC | EC |
| Home | 17 | 9E | CE | 8E | EE |

**Numeric pad keys:**

| 0 | 86 | B0 | D0 | F0 | 90 |
| 1 | 69 | B1 | D1 | F1 | 91 |
| 2 | 70 | B2 | D2 | F2 | 92 |
| 3 | 71 | B3 | D3 | F3 | 93 |
| 4 | 51 | B4 | D4 | F4 | 94 |
| 5 | 52 | B5 | D5 | F5 | 95 |
| 6 | 53 | B6 | D6 | F6 | 96 |
| 7 | 33 | B7 | D7 | F7 | 97 |
| 8 | 34 | B8 | D8 | F8 | 98 |
| 9 | 35 | B9 | D9 | F9 | 99 |

-   | 36 | AD | DD | FD | BD |
+   | 54 | AB | DB | FB | BB |
.   | 88 | AE | DE | FE | BE |
ENTER | 72 | 8D | CD | 9D | ED |
"under-O" | 87 | BA | DA | FA | 9A |
"under-ENTER" | 89 | BC | DC | FC | 9C |
Chapter 7

PROM MONITOR

This section contains an overall description of the PROM Monitor and how it functions in the ASSOCIATE. At power up, or RESET, RAM is disabled and the system begins execution in the PROM at location 0. The first instruction is a jump into the F800 range and the second is an OUT DOH enabling RAM thereby entering its normal operating configuration. Entry into the PROM monitor disables the interrupts. The System Monitor will then attempt to boot the Disk Operating System. If the Disk Operating System cannot be loaded, a second RESET will take the Monitor to the command level.

To boot the Disk Operating System from Drive A, type the monitor command "A". After boot to disk, software can be used to disable PROM so that a full 65K of RAM will be available (see 3.2.1 Memory and 4.11 Status and Control for further detail). Return to the PROM Monitor from the CP/M Disk Operating system is accomplished by executing the program MNTR, which enables PROM and returns control to the PROM Monitor.

The following description of functions available in the PROM Monitor is in two parts: first, a brief introduction and complete listing of operator commands available through the PROM Monitor are presented; second, functions used by Monitor commands and also available to the user are described.

7.1 PROM Monitor Operator Commands

The PROM Monitor Operator Commands encompass many useful functions for working with memory and I/O. The commands perform functions such as testing, displaying, and changing memory, calculations involving addresses, performing basic I/O, and examining and changing registers. These commands are designed to give users a great deal of power in working with the processor at a machine level. The following table summarizes operator commands available through the PROM Monitor:
## Summary of Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Boot Disk Operating System</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Drive A(bottom)</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>Boot Disk Operating System</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Drive B(top)</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>Call External Subroutine</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>Display Memory in Hex and ASCII</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>Fill Memory</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>Go to Address</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(optionally set breakpoints)</td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>Compute Hex Sum and Difference</td>
<td></td>
</tr>
<tr>
<td>K</td>
<td>Calculate Check Value for Memory Range (to determine if memory accidentally modified)</td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>Move a Block of Memory</td>
<td></td>
</tr>
<tr>
<td>O</td>
<td>Calculate Relative Offset</td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>Make the ASSOCIATE a Terminal</td>
<td></td>
</tr>
<tr>
<td>Q</td>
<td>Query I/O</td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>Read a Hex Tape</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>Display/Alter Memory</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>Test Memory (destructive)</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>Verify One Memory Block Against Another</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>Examine/Modify CPU Registers</td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td>Search Memory for String</td>
<td></td>
</tr>
</tbody>
</table>

If the PROM Monitor is entered at startup or RESET, the System attempts to load the Disk Operating System. If the DOS is not successfully loaded a second RESET will give control to the Monitor command level. The message "GNAT System 10 Monitor Ver X" will be displayed followed by the Monitor prompt "?". After executing each operator command, with the exception of A (Boot to Disk) and, sometimes C (Go to), the prompt character is displayed. PROM Monitor Commands can be interrupted, to abort execution, by issuing a Control C. When Control C interrupts the execution of an Operator Command, Control again returns to the command level as signalled by a question mark. In separating command parameters, spaces or commas can be used interchangeably.

The following syntactic conventions will be used in specifying the format of each command:

a) Capitalized words (or letters) and symbols should be entered as shown;

b) Lower case letters or words enclosed by less than and greater than symbols indicate operator supplied parameters;

c) Items enclosed within square brackets can be optionally included, as needed.
Note that most of the operator commands are self-completing, but in some cases (such as when the number of parameters can vary) a RETURN will be necessary.

A Boot Disk System
Syntax: A
This command boots the disk operating system (CP/M in standard configurations) from Drive A(bottom) No parameters are needed.

B Boot Disk System
Syntax: B
This command boots the disk operating system (CP/M in standard configurations) from Drive B(top) No parameters are needed.

C Call External Subroutine
Syntax: C<transfer-address>[,<parml>[,<parm2>]]
This command calls a routine external to the PROM Monitor. The external routine can return to the PROM Monitor by executing a "RET" instruction. One or two additional parameters can be supplied on the command line to the external routine; they will be passed in the H/L and D/E register pairs. If no additional parameters are supplied, zeroes are placed in the register pairs. Note that the saved Z80 registers in the user workspace are not altered by this command.

D Display Memory in Hex and ASCII
Syntax: D<start-address>,<end-address>
This command displays the contents of memory beginning at start-address and ending with the end-address. Memory is displayed in hexadecimal and ASCII with 16 bytes per CRT line. The starting location is displayed in hexadecimal at the beginning of each line.

F Fill Memory
Syntax: F<start-address>,<end-address>,<fill-character>
This command will propagate the fill-character (specified in hexadecimal) through memory from location start-address to end-address. The command is valuable in initializing a block of memory, or all of memory, to a constant value before loading a program; zero is especially useful for this purpose. As an example, the command FO,FFFF,0 fills memory with zeros.

G Go To Address
Syntax: G<transfer-address>][,[breakpoint>][,[breakpoint>]]
This command allows transfer to another program while retaining some
Monitor control by setting breakpoints. A simple transfer to another program is executed if only transfer-address is specified on the command line. To set breakpoints, one or two addresses are added to the command. Note that this feature is software controlled, and breakpoints must occur on instruction OP-CODE in RAM. When a breakpoint is reached, the breakpoint address is printed and a Monitor prompt is given. Execution of the program can be continued by entering G or G,<breakpoint> if another breakpoint is to be implemented. The breakpoint is implemented by insertion of a RST7 instruction at the desired break location. After encountering the breakpoint the original instruction is restored. The processor status (contents of all registers) is saved in the monitor work space area.

**H  Compute Hex Sum and Difference**

**Syntax:**  \( \text{H(start-number)}, \text{<end-number>} \)

This command computes and displays the sum (start-number + end-number) and difference (start-number - end-number) of two hexadecimal numbers, where the numbers range between 0000H and FFFFH. If start-number is less than end-number, the result is equal to start-number +10000H - end-number.

**K  Calculate Check Value for Memory Range**

**Syntax:**  \( \text{K(start-address)}, \text{<end-address>} \)

This command computes and prints a 16-bit check value for a memory range: it is useful for determining if a memory range has been accidentally modified. The check value is calculated by the CRC equation: \( g(x)=X^{16}+X^{12}+X^{5}+X \).

**M  Move a Block of Memory**

**Syntax:**  \( \text{M(start-address)}, \text{<end-address>}, \text{<destination-address>} \)

This command moves the contents of a block of memory beginning at start-address and ending at end-address to another block of memory starting at destination-address. Caution should be used with this command in order to prevent the unintentional destruction of memory locations which should remain unchanged. Note that wrap around from FFFFH to 0000H takes place if the source or destination block of memory goes past FFFFH.

**O  Calculate Relative Offset**

**Syntax:**  \( \text{O(jump-address)}, \text{<destination-address>} \)

This command calculates offset for relative jump instructions. Jump-address is the address of the jump instruction, and destination-address is the address of the instruction jumped to. If destination is out of the 256 byte range, an "XX" will be printed.

7. Monitor

P  Port Echo
Syntax:  P<serial-port>,<duplex>

The ASSOCIATE becomes a terminal with connection through the MODEM, LIST, or NETWORK I/O PORT. The CTRL-SHIFT-RST KEY will cause an exit to the Monitor. For <duplex>, a 1 selects full duplex mode and a 0 selects half duplex. This does not accept null characters. Use 80H for a null character.

Q  Query I/O
Syntax:  QO<port>,<value>
        QI<port>

This command examines any input port or sends a value to any output port; it gives the operator a direct input/output capability from the keyboard. For example the baud rate could be changed on the list port baud rate generator by issuing:

QO:E0,77

Baud Rate Code
Baud Rate Generator
Colon provided by monitor
Output Command

Another example of the command's use might be to go to one of the I/O ports for checkout of external devices, for instance, to check input from the MODEM, issue: QI70. The computer would respond with the value found at port 70H.

R  Read a Hex Tape
Syntax:  R[<bias>]

This command reads check-summed hex files in the normal Intel format. A bias can be added, which will cause the object code to be placed in a location other than its intended execution location. The bias is added to what would have been the normal loading location and will wraparound to enable loading any program anywhere in memory. If non zero transfer address is received, the monitor transfers control directly to the program.

S  Display/Alter Memory
Syntax:  S<start-address>

This command displays, and allows modification of, memory on a byte-by-byte basis. The byte at the start-address is displayed in hexadecimal; if the value of start-address location is to be changed, the new value is entered followed by a space, otherwise entering only a space will cause display of the hexadecimal value at the next location. To terminate execution of this command, enter a RETURN. The system adds a carriage
return and line feed before displaying continuation locations with address ending with a zero or eight; the present address location is printed after each system issued carriage return and line feed.

**T**  
Test Memory (Destructive)  
Syntax: \( T<\text{start-address}>,<\text{end-address}> \)

This command uses a very fast pseudo-binary sequence generator with a period of 256 bytes to test memory. The period is relatively prime to 256 and, thus, provides a good test of pattern sensitivity. Note that each number in the range 0-255 is generated by the algorithm; therefore, each bit in the range is tested to see if both zero and one bits can be written into it. Error information is displayed in the form of an address and bit number. The memory chip with the location which failed the test can be quickly determined by reference to the appropriate schematic.

**V**  
Verify memory  
Syntax: \( V<\text{start-address}>,<\text{end-address}>,<\text{compare-address}> \)

This command compares the contents of the block of memory delineated by start-address and end-address to the block of memory beginning at compare-address; wraparound takes place if the compared addresses exceed FFFFH. Differences in the contents of the two memory blocks are reported in the form of address and the contents of the two locations.

**X**  
Examine/Modify CPU Registers  
Syntax: \( X['']<\text{register-name}> \)

This command displays the contents of the CPU registers; if the register-name is specified, contents of the register can be changed (or not) by specifying the new contents or a space. To display the normal system status enter only an X. To display the additional Z80 registers, enter X'. Single "prime" registers may be examined and modified in the same way as with "unprimed" registers described above.

**Y**  
Search Memory  
Syntax: \( Y<\text{search-string}> \)

This command searches all memory starting at location zero for the byte string specified by search-string. Hex characters are separated by commas in specifying the search-string and up to 255 may be indicated. The starting address of each byte string found to be identical to search-string is displayed by the command.

### 7.2 PROM Monitor Entry Points

The following functions are available to user programs and can simplify the handling of I/O from system to system; they are also used by PROM Monitor operator commands. The functions are accessed by calling to the location specified in the Assembly listing. Whatever the currently
assigned device, these functions will perform the specified I/O operation and return to the calling program. Register conventions are as follows for any input or output device: the character to be output is in the C register and the return character will be in the A register after input or output. The functions are:

<table>
<thead>
<tr>
<th>ADDRESS</th>
<th>NAME</th>
<th>FUNCTION</th>
<th>PARAMETER</th>
</tr>
</thead>
<tbody>
<tr>
<td>F803</td>
<td>CI</td>
<td>CONSOLE INPUT</td>
<td>&quot;A&quot; RETURNED WITH CHAR</td>
</tr>
<tr>
<td>F806</td>
<td>RI</td>
<td>READER INPUT</td>
<td>&quot;A&quot; RETURNED WITH CHAR</td>
</tr>
<tr>
<td>F809</td>
<td>CO</td>
<td>CONSOLE OUTPUT</td>
<td>&quot;C&quot; OUTPUT TO CONSOLE</td>
</tr>
<tr>
<td>F80C</td>
<td>PO</td>
<td>PUNCH OUTPUT</td>
<td>&quot;C&quot; OUTPUT TO PUNCH</td>
</tr>
<tr>
<td>F80F</td>
<td>LO</td>
<td>LIST OUTPUT</td>
<td>&quot;C&quot; OUTPUT TO LIST</td>
</tr>
<tr>
<td>F812</td>
<td>CSTS</td>
<td>CONSOLE STATUS</td>
<td>&quot;A&quot; RETURNED WITH STATUS</td>
</tr>
<tr>
<td>F815</td>
<td>IOCHK</td>
<td>I/O CHECK</td>
<td>&quot;A&quot; RETURNED WITH IOBYTE</td>
</tr>
<tr>
<td>F818</td>
<td>IOSET</td>
<td>I/O SET</td>
<td>&quot;C&quot; PUT INTO IOBYTE LOCATION</td>
</tr>
<tr>
<td>F81E</td>
<td>USET</td>
<td>INITIALIZE I/O</td>
<td>NONE</td>
</tr>
<tr>
<td>F824</td>
<td>DATE</td>
<td>DATE CODE ID</td>
<td>None</td>
</tr>
<tr>
<td>F826</td>
<td>TRAP</td>
<td>RESTART BREAKPOINT</td>
<td>None</td>
</tr>
<tr>
<td>F829</td>
<td>PRINT</td>
<td>PRINT ON CONSOLE</td>
<td>PRINT FROM DE TILL 00 FOUND</td>
</tr>
<tr>
<td>F82C</td>
<td>BOOT</td>
<td>LOAD FIRST SECTOR AND RUN</td>
<td>NONE</td>
</tr>
</tbody>
</table>

Chapter 8

SERVICE PROCEDURES

This chapter describes the service procedures for troubleshooting and testing the ASSOCIATE computer family. This information is intended only for the technically oriented and experienced service person.

You will find that this manual will give you most of the information you need to troubleshoot and repair the ASSOCIATE on a module basis. Repairing the various modules is a Depot function and should be referred to your nearest Depot Repair facility. Field repair should be limited to replacement of the modules only.

Operation of the machine will give you most of the initial clues you need to troubleshoot problems. Carefully note any symptoms. Try to duplicate the problem, using the operations which were in progress when trouble was first noted. Many problems can be traced to improper operation or faulty software/media. A list of potential troubles and the appropriate actions follows.

SOME THINGS TO REMEMBER

*******************************************************************************
*** CAUTION ***
*** Dangerous voltages are present in this unit. ***
*** It is advised that personnel working Inside of the ***
*** ASSOCIATE or any electrical equipment should NOT wear ***
*** metal jewelry or watches. ***
*** Do not unscrew metal parts, disconnect cables, ***
*** or remove components with the power on. ***
*******************************************************************************

Observe cable connector polarity--
The pin 1 arrows on jack and cable connector must align.

HOW TO USE THE TROUBLESHOOTING CHART TO SOLVE PROBLEMS

Before consulting the TROUBLESHOOTING CHART, attempt to verbalize the difficulty you have encountered in a meaningful sentence. For example, "It doesn't work." is not nearly as descriptive as, "Fan does not run and video display is dark." The first statement isn't very helpful but the second statement might have a simple solution as: "Is the unit plugged into a working 120 VAC outlet?" or "Is the AC Line fuse blown?"

Now that you have a description of the problem, see column 1 of the TROUBLESHOOTING CHART for a match. Column 2 will explain the possible failure and column 3 will tell you what to check.

If you need further information after the TROUBLESHOOTING CHART has directed you to a possible solution, see the column 4 for a FIGure or SECTION number to refer you more tests or the probable cause of the trouble.
## ASSOCIATE TROUBLESHOOTING CHECK-LIST

<table>
<thead>
<tr>
<th>PROBLEM DESCRIPTION</th>
<th>PROBABLE CAUSE</th>
<th>REPAIR ACTION</th>
<th>REFER TO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous beep</td>
<td>Sticking keyboard &amp; Video Processor failure</td>
<td>Refit jammed key &amp; Replace 1005 PCB</td>
<td>SEC 8.2</td>
</tr>
<tr>
<td>Keys beep when pushed</td>
<td>Illegal operation &amp; BIOS failure</td>
<td>Check Operator's Manual &amp; Run diagnostics</td>
<td></td>
</tr>
<tr>
<td>Cooling fan inaudible</td>
<td>No 120 VAC power &amp; Blown fuse &amp; Fan unplugged/bad</td>
<td>Plug unit in &amp; Check power fuse or replace</td>
<td>SEC 8.3</td>
</tr>
<tr>
<td>No disk drive noise but fan is OK</td>
<td>Auto power-down &amp; Bad diskette &amp; Not rebooting &quot;A&quot; &amp; Disk drive failure &amp; Power Supply failure</td>
<td>30 sec inactivity &amp; Reboot another one &amp; CTRL-SHFT-RESET &amp; Replace drive assy &amp; Replace supply</td>
<td>CP/M</td>
</tr>
<tr>
<td>Unusual noise</td>
<td>Bad fan bearing &amp; Disk drive fail &amp; or diskette fail &amp; Power supply fail</td>
<td>Replace fan &amp; POWER DOWN NOW!! &amp; and replace &amp; adjust or replace</td>
<td>SEC 8.3</td>
</tr>
<tr>
<td>Character dot fade or twinkle</td>
<td>Power failure &amp; Unit is too hot &amp; Video process fail</td>
<td>Adjust pow. supply &amp; Fan working &amp; Run Disk tests &amp; Replace 1005 board</td>
<td>DEPOT*</td>
</tr>
<tr>
<td>Incorrect char appears</td>
<td>Improper SHFT/CTRL &amp; Sticking kybd keys &amp; CPU failure &amp; Video process fail</td>
<td>Check Operator's Manual &amp; Clean/replace kybd &amp; Run Disk Tests &amp; Replace 1005 board</td>
<td>SEC 8.2</td>
</tr>
<tr>
<td>Improper boldface or high intensity raster</td>
<td>Video process fail &amp; Brightness too high</td>
<td>Run TPAT test &amp; Adjust it lower</td>
<td>Sys disk</td>
</tr>
<tr>
<td>Typed char appears in more than one place at the same time</td>
<td>Memory failure on main CPU board</td>
<td>Run MTEST</td>
<td>Monitor</td>
</tr>
<tr>
<td>Typed char appears in more than one place at the same time</td>
<td>Memory failure on video board</td>
<td>Run TCRT</td>
<td>Sys Disk</td>
</tr>
</tbody>
</table>

**Note:**
- SEC 8.2
- SEC 8.3
- CP/M
- DEPOT*
## PROBLEM DESCRIPTION

<table>
<thead>
<tr>
<th>Missing characters or inoperative keys</th>
<th>PROBABLE CAUSE</th>
<th>REPAIR ACTION</th>
<th>REFER TO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct Oper. Sys.?</td>
<td>Check diskette</td>
<td>Replace kybd/1005</td>
<td>CP/M</td>
</tr>
<tr>
<td>Keyboard/video fail</td>
<td>Replace 1005 board</td>
<td>SEC 8.3.1</td>
<td></td>
</tr>
<tr>
<td>Garbage displayed</td>
<td>CPU non-operative</td>
<td>Replace 1000 board</td>
<td>SEC 8.1</td>
</tr>
<tr>
<td>Video process fail</td>
<td>Replace 1005 board</td>
<td>SEC 8.3.1</td>
<td></td>
</tr>
<tr>
<td>Power Supply fail</td>
<td>Adjust/replace</td>
<td>DEPOT*</td>
<td></td>
</tr>
</tbody>
</table>

| Bad raster, stretched compressed, deformed or rolling chars. | CRT out of adjust | Adjust/Replace | DEPOT* |
| Video process fail | Run TPAT test | Replace 1005 board | SEC 8.3.1 |
| Power supply fail | Adjust/replace | DEPOT* |

| Screen blank | Unit turned on? | Plug in/tum on | CP/M |
| Brightness too low | Adjust brighter | Below CRT |
| Video process fail | Run TPAT test | Replace 1005 board | SEC 8.3.1 |
| CPU inoperative | Replace 1000 board | SEC 8.1 |
| Power Supply fail | Adjust/replace | DEPOT* |

## Cursor not present

| (cursor might be turned off by software - try to reboot) | Does keyboard work? | YES? Bad Diskette | CP/M |
| | | | |
| | | NO? Bad 1000 or 1005 boards | Monitor |
| | | 1000 boards | SEC 8.3.1 |

## Cursor does not blink

| Does keyboard work? | YES? Run MTEST | Monitor |
| | NO? Failed 1005 | SEC 8.3.1 |

## Keys don’t respond

| During a prog. run? | Strike CNTL-S once | CP/M |
| Try running tests | Perform disk test | Sys disk |
| Does unit boot? | Try rebooting | CP/M |
| Run another program | Bad diskette | CP/M |
| Failed CPU/Video | Replace 1005/1000 | SEC 2 |

## Single key fails

| Run another prog. | Bad diskette | Monitor |
| Memory fail | Run MTEST | SEC 8.2 |
| Sticking key | Clean/replace | SEC 8.2 |
| CPU failure | Run Disk tests | Sys disk |
| Video process fail | Replace | SEC 8.3.1 |
| Keyboard failure | Replace | SEC 8.2 |

## Multiple keystrokes

| Key held down or sticking | Auto REPEAT | SEC 8.2 |
| Video process fail | Run Disk tests | Sys disk |
| | Replace 1005 board | SEC 8.3.1 |
8.2 Diagnostics

The following tests are used to check out various components. Errors are indicated by the audible tone and asterisks at the beginning of the displayed error line. Upon error the programs will wait for an operator input before proceeding. The tests are generally invoked by entering the name of the test at the System level followed by a Carriage Return. Prompts will ask for essential user-supplied information.

8.2.1 TSERIAL, TRS232 - Serial Port Test

These two programs perform the same tests on serial ports; they should be utilized whenever a user wants to check the correct action of a serial port--ports 60H, 70H or 72H. TSERIAL automatically tests the indicated ports. TRS232 tests individual ports as directed by the operator. In order to use these tests, a loop-back plug with the following pins connected together is required:

- 2 - 3, TxData to RxData
- 4 - 5, RTS to CTS
- 6 - 20 DSR to DTR (CD)

The test is invoked from CP/M by the following:

AC>TRS232 or AC>TSERIAL

TRS232 will inquire for the number:

TEST PORT # IN HEX -

If the port is error free:

BAUD RATE = xxx
NO DATA ERRORS DETECTED
PORT 70 TEST COMPLETE

To return to CP/M, reply to the port # query with any character which is not a hex number.

Error messages which may appear include:

*** CD/DSR not going on
*** CD/DSR not going off
*** CTS not going on
*** CTS not going off
*** Time out error (possibly non-existent port)
8.2.2 TPIO

TPIO tests the operation of the Parallel I/O port. For proper operation, TPIO must have the TFX003 test fixture installed in the 50-pin parallel connector on the back panel.

The test fixture is a parallel loopback plug with the following pins connected:

```
1  3  5  7  9 11 13 15
|   |   |   |   |   |   |
17 19 21 23 25 27 29 31
```

The PIO test is invoked by:

AO>TPIO

The program will respond with

```
TPIO - TEST PIO V1.00 as of 29-March-80
Test Complete
```

or give one of the following errors:

*** Output error - (port). Data = ww, Port # xx = yy. Error = zz
*** Input error - (port). Data = ww, Port # xx = yy. Error = zz
*** Timeout, data=vv, Port ww (Output)=xx, Port yy (Input)=zz

8.2.3 TAPU,T9512 - Arithmetic Processor Test

These programs test the optional arithmetic processor--AM9511 or AM9512. TAPU repeatedly tests the AM9511 operations of: multiply, divide, SIN, ARCTANGENT, exponentiation, and power. T9512 repeatedly tests of the operations of add, subtract, multiply, and divide. The results of each function is compared with the expected result stored in the program. Any error is indicated by a printout of actual result and anticipated result.

Setup requires that the AM9511 or AM9512 and required jumpers be installed. To invoke:

AO>TAPU

or

AO>T9512

Type any character to abort the test. The following error message may result from this test:

*** 9511 (function) error- xxxxxxxx expected result-yyyyyyyyy
8.2.4 TBAUD - Baud Rate Generator Test

This diagnostic tests each port through all 16 possible baud rates. A software timing loop measures the time between sending successive characters at a given baud rate and compares that value with an expected value, giving a resolution of about 35 usec. Any difference is an error. If the test fails, port number, baud rate, expected result, and actual result are displayed. The test should be run when errors are detected in trying to set baud rate.

Setup requires serial loopback plugs on ports 70 and 72. TBAUD is invoked by the following:

AO>TBAUD

Error messages which may appear are:

*** Timeout, Baud rate xxx, baud port yy, test values = (expected) (actual) possibly no loop back

*** Error, Baud rate xxxxx, baud port yy, test values - (expected) (actual)

8.2.5 TPAT - Pattern Test

TPAT causes the video board to fill up every character position on the CRT—all 25 lines and 80 characters. In addition it shows every character in the character PROM at both high and low intensity. This program allows inspection of the character set and the CRT raster. To invoke:

AO>TPAT

To get out of the Pattern Test enter a Control-Shift-RST or press the Reset switch on the back of the ASSOCIATE.

8.2.6 TCTC - Counter Timer Test

This test checks the counter timer chip to determine if it interrupts and times at the proper rate. There are four different timer circuits in the chip. For testing, each timer circuit is set up for a different rate. The program is then set to execute for a specified amount of time. The interrupts over that period of time are counted; if the count doesn't come out exactly right, TCTC indicates that the test failed. The Counter Timer Test is invoked from CP/M by the following:

AO>TCTC

The system will reply as follows when the Counter Timer checks out with no problem:

CTC - Test System 10 CTC Chip V1.1 as of 23-JAN-80
CTC Test Complete

The error message is:
8. Service Procedures

8.2.7 **TCRT - CRT Test**

This test exercises the CRT processor. It consists of several subtests:

- **PROM checksum.** A checksum is made of the PROM data and compared against a table in the TCRT program. In the event of an error, a message is printed on the screen and the test holds at that point.

- **RAM.** The CRT scratchpad RAM is tested. Any errors detected are printed on the screen.

- **VIDEO RAM is tested in two parts.** The top half of the screen is tested first. If any errors are detected, the erring locations are listed in the bottom half. Upon successful completion the bottom half is tested in the same manner.

To invoke TCRT, enter the following:

```
AO>TCRT
```

To end the test, enter two Control C's.

The error message is:

```
Ram error(s) at vvvv.. www.. xxxx.. yyy.. zzzz
```

8.2.8 **TDISK - Disk Drive Test**

TDISK provides options for testing the disk drives; the options are indicated after prompts during execution of the diagnostic program. The program indicates sectors and tracks under test. At the end of each pass, TDISK displays the number of accumulated errors. The number of read attempts is determined by the system BIOS. With the normal ten try BIOS the error rate should be less than one error per 1000 passes.

```
* *********************************************************
* *
* Note: Testing destroys all data on a disk. *
* Exercise care in disk selection. *
* *
* *********************************************************
```

To see if a disk can be read, DUMP validate can be used.

To invoke TDISK:

```
AO>TDISK
```

Control C aborts the test.
8.2.9 TDMA

This program tests the memory-to-memory DMA function. The process is accomplished by repetitively testing DMA memory-to-memory transfers. To use:

A0>TDMA

Test results may include the following error messages:

*** DMA modified location xxxx from yy to zz
*** DMA transfer error at xxxx should be yy but is zz

8.2.10 TINT

This program tests that all interrupt-generating devices can generate interrupts simultaneously. The process includes configuring devices for interrupts and testing if each device generates interrupts.

The devices are configured as follows:

SIO    Ports 60, 70, and 72 setup for transmit and receive interrupts.
Pio    Setup as in TPIO
CTC    Setup as in TCTC, but with lower frequency interrupts.

Loop back test fixtures are required in the serial and parallel ports.

After delaying for a specified number of software loops, the program tests if all devices have generated interrupts.

To use:

A0>TINT

Test result error messages which may appear:

*** No interrupts from CTC port 0
*** No interrupts from CTC port 1
*** No interrupts from CTC port 2
*** No interrupts from CTC port 3
*** No interrupts from PIO port A
*** No interrupts from PIO port B
*** No interrupts from SIO port 70 - receive
*** No interrupts from SIO port 70 - xmit
*** No interrupts from SIO port 72 - receive
*** No interrupts from SIO port 72 - xmit
*** No interrupts from SIO port 60 - receive
*** No interrupts from SIO port 60 - xmit
8.2.11 TRTC - Test Real Time Clock Option

TRTC exercises and tests the real time clock option. The program performs the following functions:

1. Displays the current time.
2. Monitors any interrupts generated by the Real Time Clock.
3. Allows the user to set the time and date.
4. Allows the user to set the RTC interrupt mask.
5. Allows the user to reset each individual counter and latch.
6. Performs the system clock test. This test compares the RTC timing with the system clock timing.
7. Performs the "pulse test" for fine tuning the Real Time Clock crystal.

** USER ENTRY **

** COMPUTER RESPONSE **

TRTC

Computer prompts user with options

8.2.12 TSPD - Test Disk Rotational Speed

TSPD tests the rotational speed of either the A or B disk drive. It is executed by:

A0>TSPD

A prompt menu gives the user the choice of running the rotational test, selecting the drive to test, or returning to the Operating System. Using the System clock the test operates by measuring the time it takes the diskette to make five revolutions (one second's worth). This time is converted to Revolution per Minute for display on the screen. An error message will be displayed if no Index Pulse is found.

Tandon and MPI specify 1.5% speed tolerance on the disk drives. This corresponds to an error of + or - 4.5 RPM.

8.3 Disassembly and Reassembly

This section provides information for disassembly and reassembly of the major components of the ASSOCIATE. Again, only qualified technical personnel should attempt to disassemble the Microcomputer. Improper reassembly or operation may damage the unit and void the warranty.

******************************************************************************
*** CAUTION ***
*** Dangerous voltages are present in this unit. ***
*** It is advised that personnel working inside of the ***
*** ASSOCIATE or any electrical equipment should NOT wear ***
*** metal jewelry or watches. ***
*** Do not unscrew metal parts, disconnect cables, ***
*** or remove components with the power on. ***
******************************************************************************

8-9
The ASSOCIATE Case is made out of injection molded structural foam. Individual components are mounted to an internal metal frame that provides mechanical support and electrical shielding.

8.3.1 Main Board

The main CPU board is accessible by removing the two screws at the lower rear panel. When these are removed, the main board slides out on tracks from the back of the case. Three cables pull out with the board allowing operation of the board outside the case.

To reassemble, slide the cables in ahead of the main board. The cables will roll back to lay across the top of main board.

8.3.2 Keyboard

To detach the keyboard assembly, remove the two screws which hold the keyboard cover to the bottom piece. These screws can be accessed from the bottom of the unit. The cover can be then be tilted up giving access to the keyboard.

The keyboard is attached with four screws which can be removed from the bottom of the case.

Key tops can be replaced by simply lifting them off, because they are attached by a press fit. If the key switches beneath the key tops need to be taken off, the bottom circuit board must be removed from the keyboard frame.

Note that the PROM for the keyboard is located at Z12 at the upper right side of the keyboard.

8.3.3 Anti-Glare Screen

The front screen is attached to the front assembly by Velcro attachments. The screen may be lifted with a blunt knife in the corner of the frame.

8.3.4 Top Cover and Front Bezel

Two bolts hold the top cover to the bottom piece. These bolts are located on the lower back of the unit. An ordinary philips screwdriver can be used.

After loosening these two bolts, the cover will swing up and forward. It can be removed and set aside giving complete access to the system components.

After removing the cover, the front bezel can now be removed. It is held in by indents in the cover and an overhanging hinge on the bottom.
8.3.5 Disk Drives

They may be removed by the following procedure:

1. Remove the Cover and Front Bezel

2. Remove the screw in between the drive assembly and CRT which holds the disk assembly side plate to the power supply bracket.

3. Take out the four screws on the outside of the drives.

4. Disconnect the ribbon cables and the power cables from the rear of the disk drives.

5. Slide the disk drives forward and out.

Note that the drive on the bottom is set up as drive A and the top as drive B. The jumper options and termination resistors must be correctly set on the drives.

Drive Jumper Set Up

<table>
<thead>
<tr>
<th>Pin #'s</th>
<th>Drive Label</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-16</td>
<td>H</td>
<td>SHORT</td>
<td>SHORT</td>
</tr>
<tr>
<td>2-15</td>
<td>DS0</td>
<td>SHORT</td>
<td>OPEN</td>
</tr>
<tr>
<td>3-14</td>
<td>DS1</td>
<td>OPEN</td>
<td>SHORT</td>
</tr>
<tr>
<td>4-13</td>
<td>DS2</td>
<td>OPEN</td>
<td>OPEN</td>
</tr>
<tr>
<td>5-12</td>
<td>DS3</td>
<td>OPEN</td>
<td>OPEN</td>
</tr>
<tr>
<td>6-11</td>
<td>MUX</td>
<td>OPEN</td>
<td>OPEN</td>
</tr>
<tr>
<td>7-10</td>
<td>SPARE</td>
<td>OPEN</td>
<td>OPEN</td>
</tr>
<tr>
<td>8-9</td>
<td>SPARE</td>
<td>OPEN</td>
<td>OPEN</td>
</tr>
<tr>
<td></td>
<td>RESISTOR NETWORK</td>
<td>REMOVED</td>
<td>INSTALLED</td>
</tr>
</tbody>
</table>

Drive interface pin 34 must be jumpered to ground on disk drives which do not have the signal "ready" on this pin.

8.3.6 Video Board

The video board is mounted vertically with four press fit attachments in back of the CRT unit. To remove:

1. Remove the Cover and Front Bezel

2. Disconnect all cable to the 1005 board.

3. Pull off the 1005 board up with gentle pressure.

On reassembly, be sure to connect all cables to their proper socket. Incorrect connection may damage unit. Specifically, the connector to the CRT must be correct! And the keyboard cable must be correct! On these cables Pin 1 of the cable must match pin 1 of the proper connector.
8.3.7 Power Supply

To remove the power supply unit at the bottom of the disk drives:

1. Remove the Cover and Front Bezel
2. Remove the Disk Drive assembly.
3. Remove the DC distribution power cable and the AC wiring from the supply.
4. Detach the power supply by removing the four top mounting screws.

8.3.8 CRT

To take out the CRT, the entire mounting frame must be removed from the case. To do this, follow the procedure below:

1. Remove cover and front bezel.
2. Remove the 1005 Video Processor Board.
3. Remove the AC Cable to the Power Supply.
4. Take out the six bolts holding the metal frame to the bottom section and remove the entire assembly.
5. Unbolt the CRT from the metal frame.

8.4 Return of Material

In the event that equipment must be returned for repair, it should be suitably packed for shipment. Failure to package properly may result in voiding the warranty or violating the terms of the service contract.

If the ASSOCIATE was purchased from a system house or dealer, contact them rather than Data Technology. For return to Data Technology, call customer service to obtain instructions and return authorization.
ASSOCIATE FEATURES

GENERAL

Attractive Desktop Cabinet
Portable
Selectric Style Keyboard
Software Definable Function Keys
Accounting Style Numeric Pad
Low Glare Screen
Full Screen Editing

HARDWARE

Z80A CPU
65K RAM
700K Mass Storage on Dual Minifloppys
  Optional 1.6 Megabytes
  Optional Hard Disk
DMA Data Transfer
Hard Disk Interface for Additional Mass Storage
2 RS232 Serial Ports (Printer and Modem)
1 RS449 Serial Communication Port to 500K Baud
Programmable Baud Rates
Separate CRT Microprocessor
IEEE 488 GPIB Parallel I/O*
High Speed Arithmetic Processor*

SOFTWARE

PROM Resident Disk Boot and Diagnostic Monitor
CP/M** Version 2 Disk Operating System
Screen-Oriented Editor
Word Processing Program*
Business Software*
Communications Software*
Extensive Software Support*

<table>
<thead>
<tr>
<th>BASIC</th>
<th>FORTRAN</th>
<th>PASCAL</th>
<th>RATFOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>COBOL</td>
<td>ASSEMBLER</td>
<td>PL/1</td>
<td>&quot;C&quot;</td>
</tr>
</tbody>
</table>

*Optional

**Trademark of Digital Research