ZEBRA® 1700/1760

Hardware Reference Manual

88A00820-02

Nov 7, 1985

RECORD OF REVISIONS

Title: ZEBRA[R] 1700/1750 Hardware Reference Manual

Document No. 88A00820A02

Date	Revision Record
May 85	Original Issue
Nov 85	Revision A02, ECO 15760, 15709

FCC COMPLIANCE

ZEBRA System equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with instructions, may cause interference to radio communications. It has been tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference, in which case the user at his own expense will be required to take whatever measures may be required to correct the interference.

FOREWORD

This document describes the hardware structure and basic operation of the General Automation ZEBRA® 1700/1750 Series.

Section 1 describes the ZEBRA System organization and provides an introduction to the operating systems available with ZEBRA: XENIX™ (ZEBRA 1700) and PICK® (ZEBRA 1750).

Section 2 provides a description of the ZEBRA 1700/1750 motherboard and its major subsystems. Following this, Sections 3 through 6 describe the SASI Input/Output (I/O) Interface, the ZEBRA 1700 Memory Management Unit, I/O Subsystems, and Memory Expansion. Section 7 concludes with a description of ZEBRA 1700/1750 power supply characteristics.

Appendix A describes the ZEBRA 1700/1750 Firmware Executive, a facility providing the user with support tools for ZEBRA operation, programming, test, and fault diagnostics.

Related ZEBRA documents available to the user from General Automation or from ZEBRA subsystems manufacturers are:

Document No.	Title
88A00757A	PICK Operator Guide
88A00760A	Quick Guide for the PICK Operating System
88A00774A	PICK Utilities Guide
88A00817A	ZEBRA 1700/1750 Installation Guide
88A00800A	ZEBRA XENIX Programmer Manual
88A00803A	ZEBRA XENIX Installation, Operation and Administration Guide
88A00818A	ZEBRA 1700/1750 Expansion Enclosure Installation Guide

```
**********************************************
*                                            *
*                  CAUTION                   *
*                                            *
*  BEFORE disconnecting peripherals,         *
*  unplugging power cables, or removing      *
*  boards from ZEBRA, system power must      *
*  be OFF.                                   *
*                                            *
**********************************************
```

ZEBRA 1700/1750 Subsystem Manufacturers:

IOMEGA Corporation
4646 South 1500 West
Ogden, Utah 84403

RODIME PLC
25801 Obrero, Suite 6
Mission Viejo, CA 92691

OMTI
A Subsidiary of Scientific MicroSystems, Inc.
557 Salmar Ave.
Campbell, CA 95008

ARCHIVE Corporation
3540 Cadillac
Costa Mesa, CA 92626

Motorola, Inc.
3501 Ed Bluestein Blvd.
Austin, TX 78721

Standard Microsystems Corporation
35 Marcus Blvd.
Hauppauge, NY 11788

TABLE OF CONTENTS

LIST OF APPENDIXES

LIST OF ILLUSTRATIONS

LIST OF TABLES

1 introduction

The ZEBRA® system is a compact, high-speed, table-top computer system supported with the PICK® and XENIX™ operating systems. The PICK version is identified as ZEBRA 1750, the XENIX version as ZEBRA 1700. Figures 1-1 and 1-2 illustrate the functional structure of ZEBRA.

1.1 SPECIFICATIONS

Size:	5" high by 17-1/4" wide by 16" deep.
Weight:	Thirty (30) pounds.
Color:	Bone white with black front panel.
Power Requirements:	100/120 VAC, 3 Amps. 220/240 VAC, 1.5 Amps. 48/63 Hz, single phase. 115 VAC ±5% isolated line to breaker box with separate breakers is preferred.
Environmental:	55°F to 85°F ambient temperature; 70°F is optimum. Maximum temperature change per hour: 25°F. 20 to 80% relative humidity, non-condensing; 65% is optimum.
CPU Speed:	10 MHz clock rate.
Fixed Winchester:	20MB or 40MB formatted.
Removable Winchester or Cartridge Tape:	5MB 40MB standard 1/4-inch cartridge with QIC-24 data format.

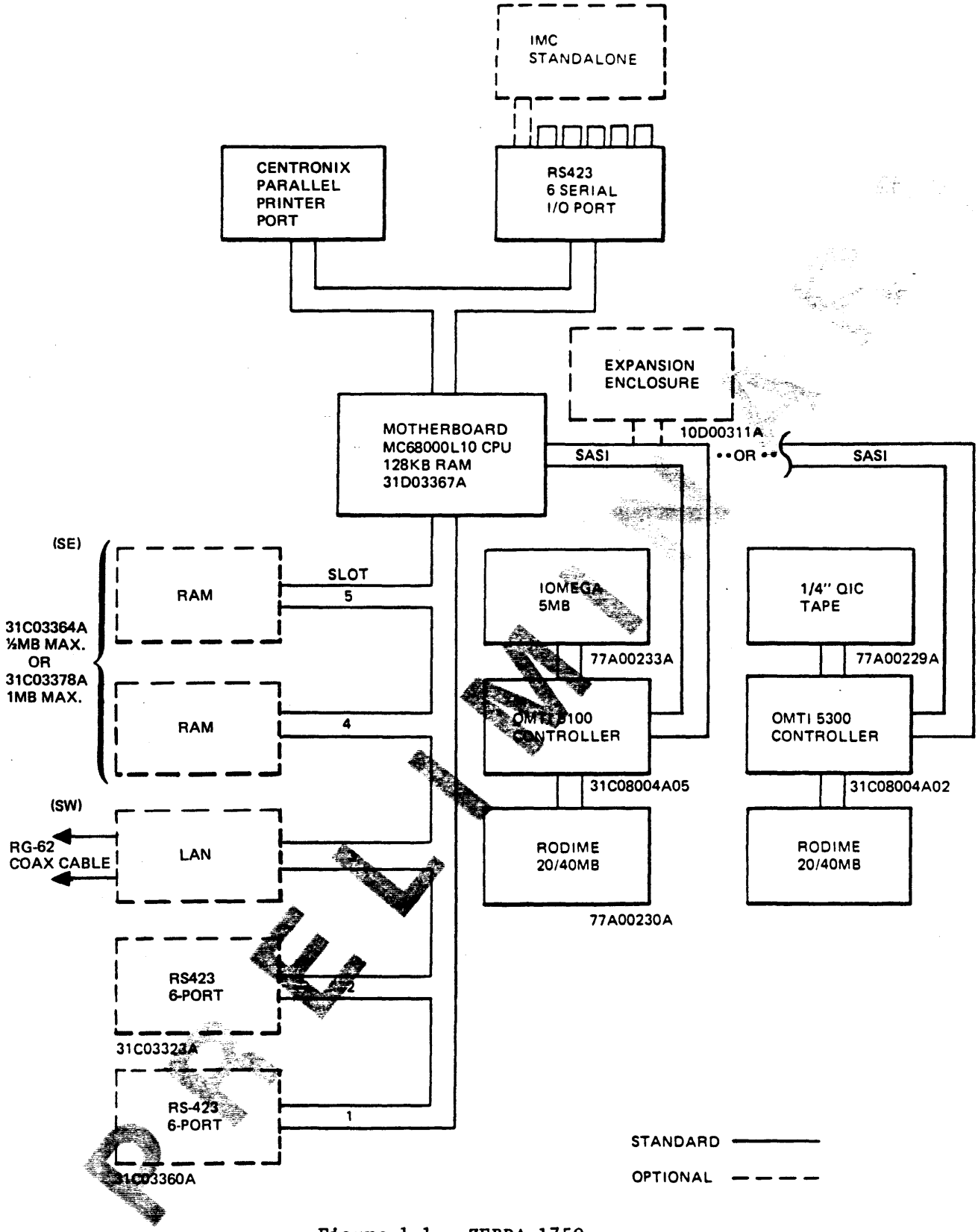


Figure 1-1. ZEBRA 1750

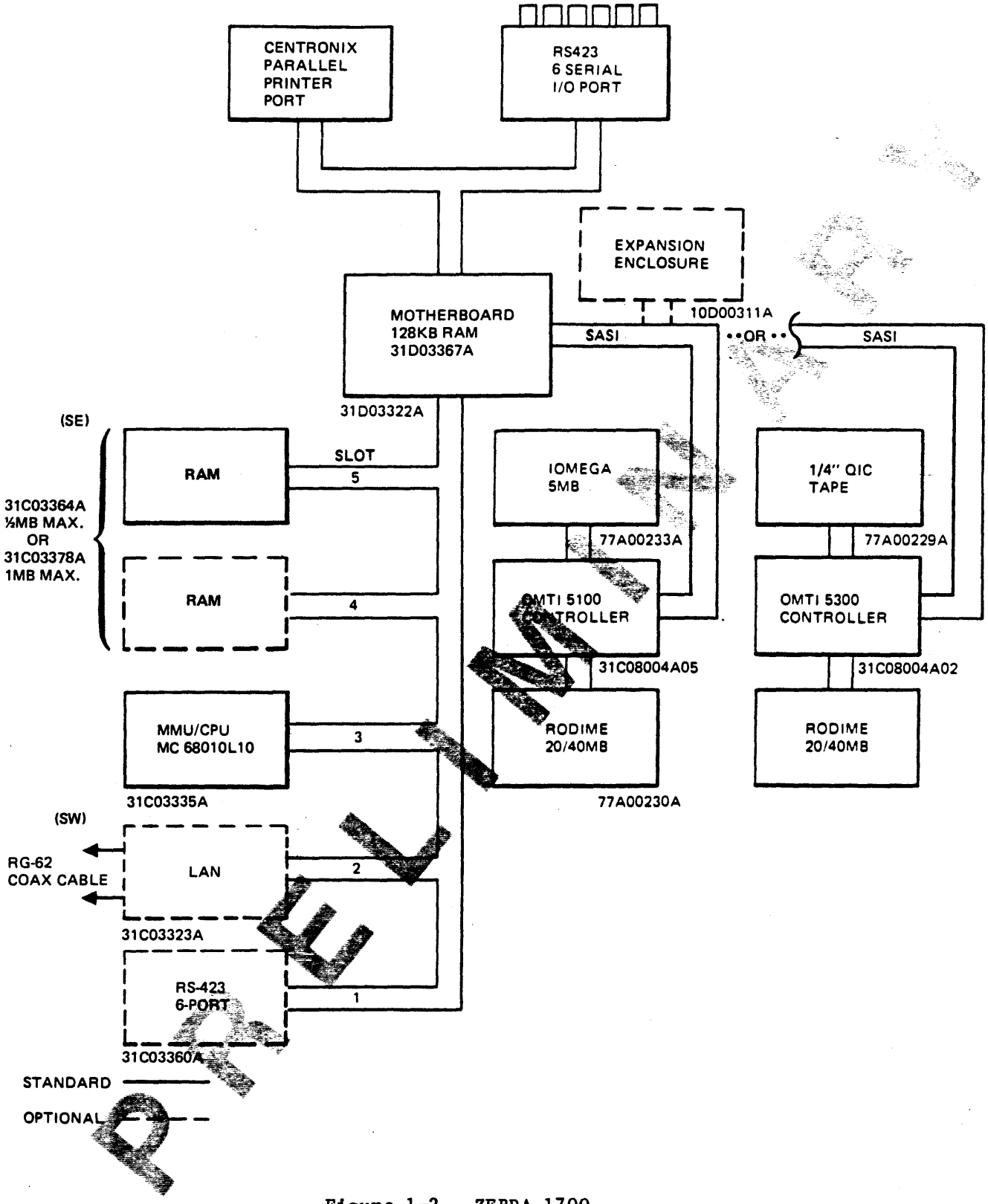


Figure 1-2. ZEBRA 1700

Communications Ports:

Port 0 - Port 4:	RS-423, 9-pin connector, 9600 baud, asynchronous, serial 8 bits with 1-1/2 stop bits. Port 0 is normally used with a local CRT terminal; Ports 1-4 are available for support of RTS, CTS, DTR and DSR.
Port 5:	RS-423, 25-pin connector, 9600 baud, asynchronous, serial 8 bits with 1-1/2 stop bits.
Parallel Printer:	Cable connector (36-pin) for Centronix-compatible parallel printer.
Network 1,2	Two Coax connectors provided if the LAN option is selected..

1.2 ZEBRA 1750

The ZEBRA 1750 series is centered on a single "motherboard" containing the 10 MHz MC68000^{R} microprocessor, 128KB of dynamic Random-Access Memory (RAM), Erasable Programmable Read-Only Memory (EPROM), six serial Input/Output (I/O) ports, one parallel printer port, five 50-pin slots for removable option boards and a Shugart Associates System Interface (SASI) bus. Figure 1-1 is a general layout of these modules. The SASI bus can drive one of the following OMTI Controllers:

1. OMTI 5100 for handling one Winchester + one 5MB cartridge disk (CD).

2. OMTI 5300 for handling one Winchester + one 1/4" cartridge tape (CT).

Available options for the ZEBRA 1750 include:

- 128KB or 384KB of memory add-on (J5) as the first memory expansion and a 512KB second memory expansion (in J4, yielding 768KB or 1MB total memory) of RAM
- 2-port Local Area Network (LAN)
- Inter-Machine Communications (IMC) system
- Twelve additional I/O RS-423 ports
- Expansion Enclosure to accommodate additional disk and/or 1/4" cartridge tape devices.

1.3 ZEBRA 1700

The ZEBRA 1700 series (Figure 1-2) is centered on the same motherboard as ZEBRA 1750 with one major difference: the CPU chip, MC68010^{R}, is a part of the Memory Management Unit (MMU) required for running under XENIX.

The basic RAM for the ZEBRA 1700 is 512KB, 128KB on the motherboard, and 384KB on a separate board mounted in slot 1. The maximum ZEBRA 1700 RAM is 1024KB, achieved by the addition of a 512KB board in slot 4. Other ZEBRA 1750 options described above (with the exception of IMC) are also available for the ZEBRA 1700.

1.4 XENIX 1700 OPERATING SYSTEM

ZEBRA 1700 is provided with the XENIX operating system, The XENIX operating system is a multi-user, multi-tasking, system. It requires the addition of the MMU board.

1.5 PICK OPERATING SYSTEM

ZEBRA 1750 is provided with the PICK operating system. The PICK operating system is a multi-user data base management system. This is based on a virtual disk memory management system providing over 8 billion bytes of logical address space. Multiple users can access any part of this address space up to the available disk storage.

motherboard 2

The ZEBRA Motherboard is an integrated system component on a multiple-layer printed circuit board. This board is a complete single-card computer consisting of the MC68000 microprocessor for the ZEBRA 1750, 128K bytes of Random-Access Memory (RAM), Read-Only Memory (ROM), six serial Input/Output (I/O) channels, and the parallel printer interface. The ZEBRA 1700 microprocessor (MC68010®) is mounted on the Memory Management Unit (MMU) board for use under XENIX (see Section 4.0).

The major functional areas of the CPU board and their physical locations are shown in Figure 2-1 and described in subsequent sections.

NOTE

The terminology used in this document observes the following conventions:

1. An active signal is referred to as asserted; an inactive signal is referred to as deasserted.

2. Names of signals which are active low are suffixed by an asterisk (*), e.g., HALT*.

3. Hexadecimal values are prefixed by a dollar sign ($), e.g., $8000.

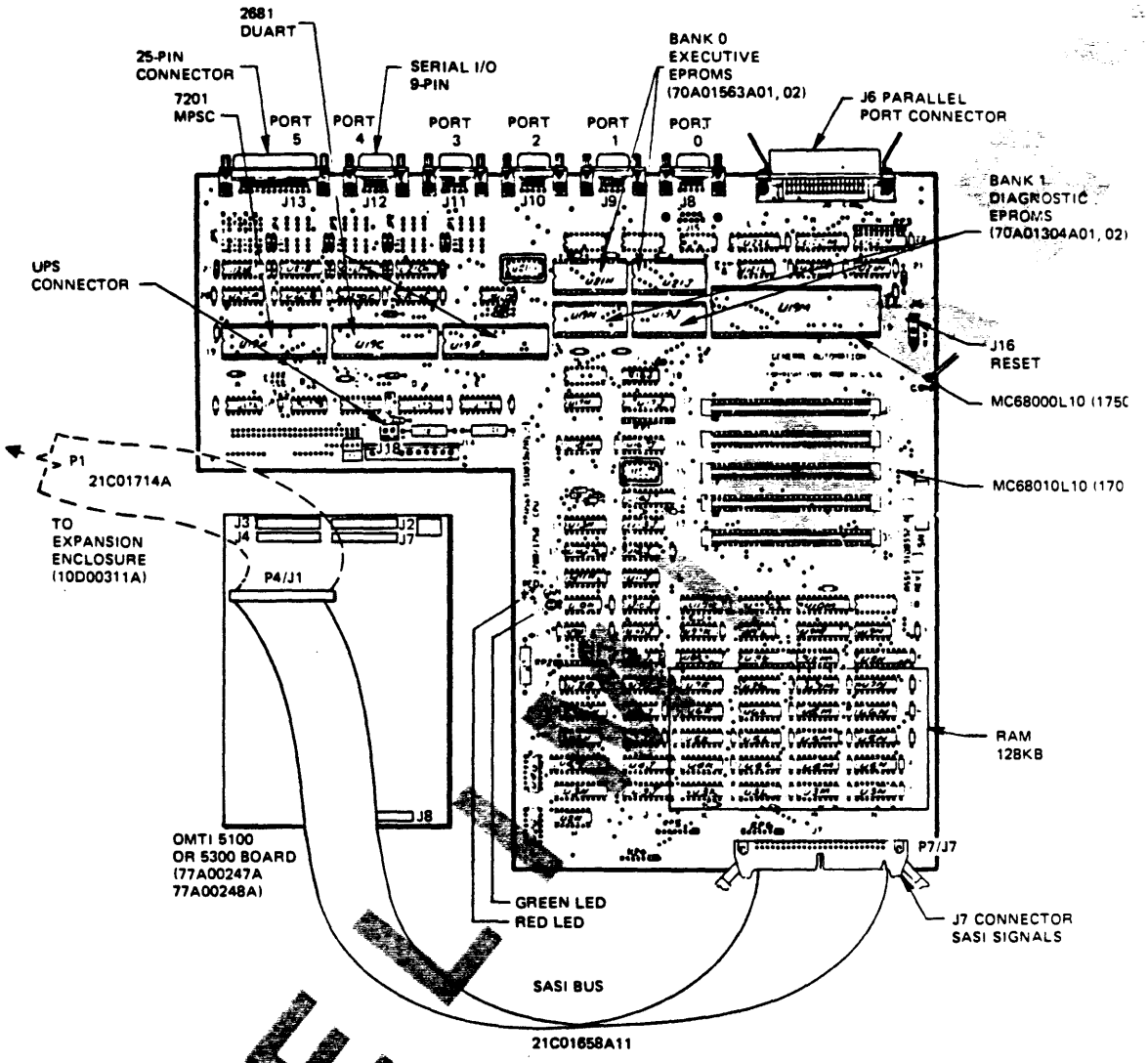


Figure 2-1. ZEBRA 1700/1750 Motherboard - 31D03367A

2.1 MICROPROCESSOR

The microprocessor (Motorola MC68000 for ZEBRA 1750, MC68010 for ZEBRA 1750) is a high-performance computer chip with an internal 32-bit architecture and a 16-bit data bus. This microprocessor features seventeen 32-bit registers: nine address registers and eight data registers. Two of the address registers are alternates depending on whether the system is in the supervisor or user state, so that eight address registers are used at any one time.

The microprocessor instruction set and addressing modes are both extremely regular in their implementation with a minimum of special cases, thus making high-level language code generation fairly simple.

The microprocessor manipulates three major data formats:

1. 8-bit bytes,

2. 16-bit words,

3. 32-bit long words.

The microprocessor can operate in supervisor or user states, assuring a secure operating system. The CPU has been designed to fully utilize the high performance of the microprocessor by providing on-card RAM that will operate without wait states at the 10 MHz speed of the system.

2.1.1 MEMORY ALLOCATION

The MC68000 microprocessor has a 16MB address range which is divided into six specific areas. These are defined as follows:

$000000 > $1FFFFF - Main memory (2.097MB RAM)
$200000 > $3FFFFF - ROM bank #0
$400000 > $5FFFFF - ROM bank #1
$600000 > $7FFFFF - On-board system resources
$800000 > $9FFFFF - Reserved area for expansion options
$A00000 > $FFFFFF - Reserved for memory map option

User mode access from the microprocessor is limited to the main memory address range ($000000 through $1FFFFF). Access outside of this range in user mode causes a bus-error condition. Supervisor access is permitted over the entire address space.

The main microprocessor bus is available at five individual 60-pin expansion connectors (J1-J5) to allow the addition of memory and Input/Output (I/O) expansion options.

2.2 RANDOM ACCESS MEMORY (RAM)

The ZEBRA 1700/1750 motherboard contains a standard base memory of 128K bytes. This RAM is located at the low end of the 2MB addressable memory space (i.e., $000000 to $1FFFFF). Memory capacity can be expanded with the following memory boards:

Memory Board		Base	Memory Option	Total
1750		128KB		
31C03364A01	J5		128KB	256KB
or				
31C03364A11	J5		384KB	512KB
31C03364A21	J4		512KB	768KB or 1MB
1700		128KB		
31C03364A11	J5	384KB		512KB
31C03364A21	J4		512KB	1MB

Memory read cycles by the microprocessor are performed without wait states. Memory write operations have one wait state added.

The dynamic RAM is refreshed every two milliseconds by a firmware routine that is activated by a level 7 interrupt of the microprocessor. This interrupt is generated by the counter/timer in Dual Universal Asynchronous Receiver/Transmitter (DUART) 0 (see Section 2.4.1). The refresh operation takes approximately 70 microseconds.

Memory protection is provided by byte parity. A parity error during a read cycle causes a bus error condition at the microprocessor during the next memory cycle. A specific procedure must be used to determine if the bus error was caused by parity or by some other event (see Section 2.8).

NOTE

Since the microprocessor is not informed of the parity error until the following memory cycle, the specific address of the parity error may be difficult or impossible to ascertain.

Immediately following RESET, the memory parity detection is disabled and the RAM can only be written into. Reading from address $000000 and above returns data from Erasable Programmable Read-Only Memory (EPROM) bank #0. This condition is called the "boot mode" and is exited as described in Section 2.3.

An expansion connector is integral to the main memory circuitry and provides the basis for addition of up to 896K bytes of dynamic memory, bringing the total system RAM to a maximum of 1024K bytes (1MB).

2.3 PROGRAMMABLE READ-ONLY MEMORY (PROM)

The ZEBRA 1700/1750 motherboard (Figure 2-1) provides two banks (32K bytes total) of Programmable Read-Only Memory (PROM) employed by Executive and Diagnostic firmware (Appendix A). The address range for each bank and jumper positions are as follows (see Figure 2-1 for jumper location):

Address Range - Bank #0	Address Range - Bank #1	JP6	JP7
$200000 > $203FFF	$400000 > $403FFF	1-2	1-2

The decoding of the EPROM address rolls over at the end of each device so that multiple images of the EPROM fill the entire 2MB space for each bank.

Immediately after a system reset, bank #0 appears at address $000000 for read operations as well as $200000. This condition is called the "boot mode." While in the boot mode, memory parity errors are ignored, read operations from main memory return data from EPROM bank #0, and the green SELFTEST PASS LED is turned off.

Writing a word value of $0000 to address $200000 will exit the boot mode, reset the Power Fail Flag, and turn off the SELFTEST PASS LED. Writing a value of $0001 to address $200000 will exit the boot mode, reset the Power Fail Flag, and turn on the green SELFTEST PASS LED.

A red HALT LED is connected to the microprocessor halt line. This LED is illuminated during reset and whenever the microprocessor is in halt mode.

2.4 INTERRUPTS

The CPU board has seven interrupt levels numbered 1 through 7; level 7 is the highest priority. At any time, the CPU board has an interrupt priority number set as a part of the microprocessor status register. Interrupts are acknowledged for all priority levels greater than the current microprocessor priority contained in the microprocessor status register. Interrupts are prohibited for all priority levels less than, or equal to, the current microprocessor priority contained in the microprocessor status register, except for level 7 interrupts which are non-maskable. When an interrupt is acknowledged, the microprocessor priority is set to the level of the interrupt request. The MC68000 microprocessor is configured to handle interrupts in the auto-vector mode. The interrupt levels are assigned as follows:

Interrupt Level	Source
INT7*	Refresh timer in DUART 0
INT6*	Power Fail Flag
INT5*	Real-Time Clock (RTC) timer in DUART 1
INT4*	Both 2681 DUARTs and the 7201 MPSC (Multi-Protocol Serial Controller)
INT3*	Parallel printer interface
INT2*	Available for option boards
INT1*	SASI bus interface

2.5 SERIAL I/O PORTS

This section describes a set of I/O ports that are standard with all ZEBRA 1700/1750 systems. These systems can be equipped with up to two added sets of six I/O ports. Section 2.6 describes the 6-port Expansion Board, its pin assignments, and register addresses.

As shown by the physical layout (Figure 2-1), the CPU board is equipped to support six fully independent I/O ports, designated as J8 through J13 on the back panel. Logically, these are identified as Port 0 through Port 5, as noted on Figure 2-1. Each port is programmable with respect to baud rate, character length, and number of stop bits, connecting to RS-423A drivers and receivers. The RS-423 interface permits more reliable operation over long cable distances and is compatible with RS-232C devices.

As shown by Figure 2-2, two unique serial communication controllers are employed and implemented as follows:

- Ports 0 through 3 are implemented using two Signetics SCN2681 DUART integrated circuits. Each DUART supports two channels, using asynchronous transmission only.

- Ports 4 and 5 are implemented using a single NEC 7201 MPSC integrated circuit. Port 4 supports asynchronous transmission only. Port 5 is programmable to support either asynchronous or synchronous transmission and configured to allow the use of an external clock (at present, only asynchronous transmission is supported by operating systems supplied by General Automation, Inc.).

  Ports 4 and 5 are handled by a different driver than that used for Ports 0 through 3 and allow three additional baud rates. The baud rates are as follows:

Ports 0 - 3	Ports 4 & 5	Ports 0 - 3	Ports 4 & 5
-	50.0	1,800.0	1,800.0
75.0	75.0	2,000.0	2,000.0
110.0	110.0	2,400.0	2,400.0
134.5	134.5	-	3,600.0
150.0	150.0	4,800.0	4,800.0
300.0	300.0	-	7,200.0
600.0	600.0	9,600.0	9,600.0
1,200.0	1,200.0	19,200.0	19,200.0

- Port 0 is used as the system console during bootload operations, but functions as a normal user port at other times.

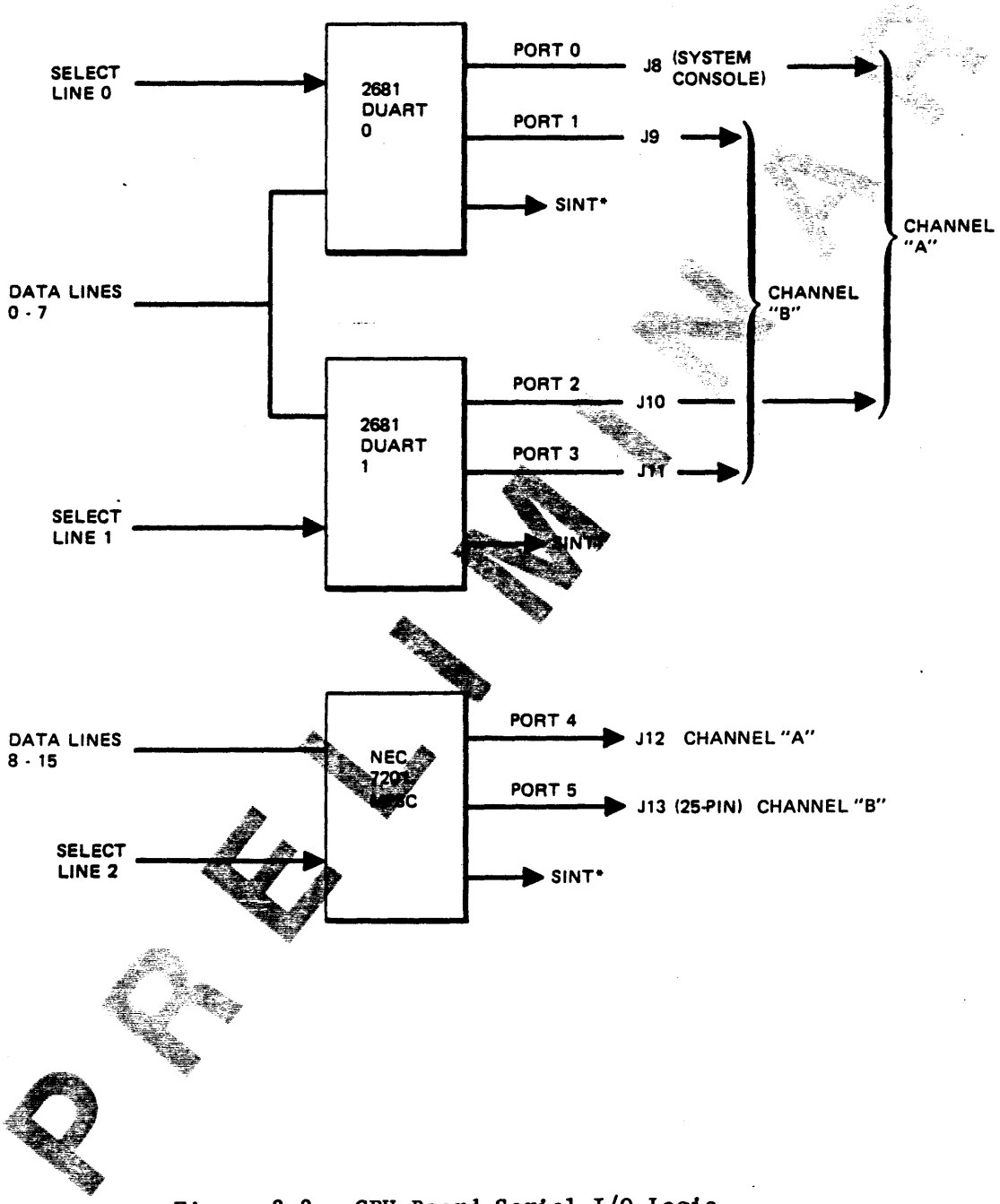


Figure 2-2. CPU Board Serial I/O Logic

A common serial I/O interrupt line (SINT*) is provided to allow interrupt driven I/O. A functional block diagram of the serial I/O logic is illustrated in Figure 2-2. The connector and pin assignments for each port are listed in Table 2-1.

Table 2-1. Serial I/O Port Connector/Pin Assignment

90C03367A

Port	Connector	Signal	Pin
0,1,2, 3,4	J8,J9,J10,J11,J12	CGND - Chassis Ground	1
		DSR - Data Set Ready	2
		TXD - Transmit Data	3
		RXD - Receive Data	4
		GND - Signal Ground	5
		DCD - Data Carrier Detect	6
		RTS - Ready-To-Send	7
		CTS - Clear-To-Send	8
		DTR - Data-Terminal Ready	9
5	J13	CGND - Chassis Ground	1
		TXD - Transmit Data	2
		RXD - Receive Data	3
		RTS - Ready-To-Send	4
		CTS - Clear-To-Send	5
		DSR - Data-Set-Ready	6
		GND - Signal Ground	7
		DCD - Data Carrier Detect	8
		TXCIN - Xmtr Signal Timing	15
		RXCIN - R´cvr Signal Timing	17
		DTR - Data-Terminal Ready	20
		TXCOUT - Xmtr Signal Timing	24

NOTE

Port 5 is wired as a Data Terminal Equipment (DTE) port rather than as a Data Computer Equipment (DCE) port. Port 5 was designed to be connected to a modem using a standard "straight-through" terminal cable. When connecting Port 5 to a terminal instead of a modem, pins 2 and 3 must be reversed.

2.5.1 SIGNETIC 2681 DUART

Serial Ports 0 and 1 are controlled by one 2681 DUART located at base addresses $600000 and $600010; Ports 2 and 3 are controlled by a second 2681 DUART located at base addresses $600020 and $600030. Ports 0 and 2 correspond to the 'A' channel in their respective DUARTs, while Ports 1 and 3 correspond to the 'B' channel. Two wait states are added for all access cycles to the DUARTs. The following list shows the relationship between the address offset and the internal registers (one byte) in the DUARTs.

Offset from Base Address	Register Description Read	Write
+ $01	Mode Register A	Mode Register A
+ $03	Status Register A	Clock Select Register A
+ $05	Reserved	Command Register A
+ $07	RX Holding Register A	TX Holding Register A
+ $09	Input Port Change Register	Auxiliary Control Register
+ $0B	Interrupt Status Register	Interrupt Mask Register
+ $0D	Counter/Timer Upper	Counter/Timer Upper Register
+ $0F	Counter/Timer Lower	Counter/Timer Lower Register
+ $11	Mode Register B	Mode Register B
+ $13	Status Register B	Clock Select Register B
+ $15	Reserved	Command Register B
+ $17	RX Holding Register B	TX Holding Register B
+ $19	Reserved	Reserved
+ $1B	Input Port	Output Port Configuration Register
+ $1D	Start Counter Command	Set Output Port Bits Command
+ $1F	Stop Counter Command	Reset Output Port Bits Command

The connector pin assignments for DUART ports are defined in Table 2-2.

Table 2-2. DUART Connector Pin Assignment

90C03367A

Port	Pin No.	Signal Name
Ports 0 to 3 (J8, J9, J10, J11)	1	CGND
	2	DSR
	3	TXD
	4	RXD
	5	GND
	6	DCD
	7	RTS
	8	CTS
	9	DTR

Interrupts from both DUARTs are connected to the auto-vector interrupt level 4 (INT4*, see Section 2.4) input to the MC68000.

The counter timer in the DUART located at base address $600000 is implemented as the refresh timer. A clock of 1 MHz is applied to input IP2 of the DUART. The output of the counter timer, OP3, is connected to the level 7 auto-vector interrupt input of the microprocessor. A transition from a high to a low level on this output causes a non-maskable interrupt of the microprocessor.

The counter timer in the DUART located at base address $600020 is utilized as the system Real-Time Clock (RTC). A clock of 1 MHz is applied to input IP2 of the DUART. The output of the counter timer, OP3, is connected to the level 5 auto-vector interrupt input of the microprocessor. A low level on this output causes a level 5 interrupt of the microprocessor.

When a DUART-driven I/O port is connected to RS-232C compatible devices, there are restrictions that must be observed:

1. Data rates and cable lengths must be restricted to those allowed under RS-232C. The CPU board is designed to operate with RS-232C at data rates up to 9600 baud and cable lengths to 50 feet maximum.

2. The signal level of RS-232C drivers must be ± 12 volts DC, or less.

3. The rise times of RS-423A drivers must be set to meet RS-232C specification.

The DUART serial ports are configured as standard RS-232 with RS-423 drivers and receivers. The transmit, receive, and modem control lines for the 'A' and 'B' channels are implemented as follows:

RS-232 Connector Signal	2681 DUART Signal
'A' - Ports 0 and 2:	
Transmit Data	RXDA
Receive Data	TXDA
Clear to Send	OP0 (Channel A RTS)*
Request to Send	IP3 (Channel A CTS)*
Data Set Ready	OP4 (Rx RDYA)
Data Terminal Ready	IP0 (General Purpose Input)
'B' - Ports 1 and 3:	
Transmit Data	RXDB
Receive Data	TXDB
Clear to Send	OP1 (Channel B RTS)*
Request to Send	IP2 (Channel B CTS)*
Data Set Ready	OP5 (Rx RDYB)
Data Terminal Ready	IP1 (General Purpose Input)

*The connection of RTS and CTS to the 2681 is optional. Normal connection is RTS tied to CTS and neither tied to the DUART. Extra jumpers must be added to connect RTS/CTS to the DUARTs.

2.5.2 NEC 7201 MPSC

Serial Ports 4 and 5 are controlled by one 7201 MPSC located at base addresses $600040 and $600044. Ports 4 and 5 correspond to the 'A' channel and 'B' channel of the 7201, respectively. Two wait states are added for all access cycles to the MPSC. The following list shows the relationship between the address offset and the internal registers (one byte) in the MPSC.

Offset from Base Address	Register Description Read	Write
+ $00	Channel A Data Read	Channel A Data Write
+ $04	Channel B Data Read	Channel B Data Write
+ $02	Channel A Status Read	Channel A Command/Parameter
+ $06	Channel B Status Read	Channel B Command/Parameter

The connector pin assignments for MPSC ports are defined in Table 2-3.

Table 2-3. MPSC Connector Pin Assignment

90C03367A

Port	Pin No.	Signal Name
Port 4, J12	1	CGND
	2	DSR
	3	TXD
	4	RXD
	5	GND
	6	DCD
	7	RTS
	8	CTS
	9	DTR
Port 5, J13 (25-Pin)	1	CGND
	2	TXD
	3	RXD
	4	RTS
	5	CTS
	6	DSR
	7	GND
	8	DCD
	15	TXCIN
	17	RXCIN
	20	DTR
	24	TXCOUT

Interrupts from the MPSC are connected to the auto-vector interrupt level 4 (INT4*, see Section 2.4) input to the microprocessor.

The internal baud rates for both the 'A' and 'B' channels are generated by a dual channel x16 baud-rate generator. The baud-rate generator is located at address $60006F. It is byte wide with the lower nibble controlling the baud rate for channel A transmit and receive, and the upper nibble controlling the baud rate for channel B transmit and receive. The nibble value and the corresponding baud rate are shown below:

Nibble	Baud Rate (x16)	Nibble	Baud Rate (x16)
0000b	50	1000b	1,800
0001b	75	1001b	2,000
0010b	110	1010b	2,400
0011b	134.5	1011b	3,600
0100b	150	1100b	4,800
0101b	300	1101b	7,200
0110b	600	1110b	9,600
0111b	1,200	1111b	19,200

The MPSC serial ports are configured as standard RS-232 with RS-423 drivers and receivers. The transmit, receive, and modem control lines for the 'A' and 'B' channels are implemented as follows:

RS-232 Connector Signal	7201 MPSC Signal
'A' - Port 4, J12:	
Transmit Data	RXDA
Receive Data	TXDA
Clear to Send	RTSA*
Request to Send	CTSA*
Data Set Ready	DTRA
Data Terminal Ready	DCDA/CTSA
'B' - Port 5, J13:	
Transmit Data	TXDB
Receive Data	RXDB
Clear to Send	CTSB
Request to Send	RTSB
Data Set Ready	(General Purpose Input)
Data Terminal Ready	DTRB
Data Carrier Detect	DCDB

*Note that RTS and CTS are not connected to the 7201 Port 4, but are connected to each other at the connector. Extra jumpers must be added to connect RTS/CTS of Port 4 to the MPSC.

Port 5 is configured to be selectively driven by external transmit and receive clocks. The channel B (Port 5) x16 baud rate generator output is always available as TXCOUT on the Port 5 connector. The following list defines jumper (JP5) positions for the Port 5 internal or external clock.

Transmitter internal clock (x16)	JP5	1-2
Receiver internal clock (x16)	JP5	4-5
Transmitter external clock (x1)	JP5	2-3
Receiver external clock (x1)	JP5	5-6

NOTE

Port 5 is wired as a DTE port rather than as a DCE port. Port 5 was designed to be connected to a modem using a standard "straight-through" terminal cable. When connecting Port 5 to a terminal instead of a modem, pins 2 and 3 must be reversed.

2.5.3 I/O PORT ADDRESSING

The serial I/O ports are addressed using byte mode addressing. The SCN2681 DUARTs are connected to the low-order data lines and are addressed at odd byte boundaries; the NEC 7201 MPSC is connected to the high-order data lines and is addressed at even byte boundaries. The addresss of the data, control, and status registers for each port are listed in Table 2-4.

2.5.4 I/O CONTROLLER RESET

Each serial controller is reset by hardware: 1) at power up, 2) via the reset switch, 3) via the watchdog timer, and 4) via the microprocessor reset instruction (which activates the reset signal to the CPU card) or with software by writing the hex value 18 to command register 0. A hardware reset affects both channels of the serial controller. A software reset affects only the channel referenced by the control register write (i.e., $600002 for channel A; $600006 for channel B). A reset initiates each serial I/O controller as follows:

1. Receivers and transmitters are disabled,
2. The outputs of the transmitters are set high (mark),
3. Modem control signals DTR, RTS are unasserted (set high), and
4. Interrupts are disabled and all pending interrupts are cleared.

Table 2-4. Serial I/O Port Register Addresses

Port	Register	Address (Hexadecimal)	Port	Register	Address (Hexadecimal)
0	Data	$600007	3	Data	$600037
	Control	$600005		Control	$600035
	Status	$600003		Status	$600033
1	Data	$600017	4	Data	$600040
	Control	$600015		Control	$600042
	Status	$600013		Status	$600042
2	Data	$600027	5	Data	$600044
	Control	$600025		Control	$600046
	Status	$600023		Status	$600046

2.6 SIX-PORT EXPANSION

Up to twelve serial I/O ports can be added to the ZEBRA 1700/1750. This is accomplished with the addition of one or two Expansion Boards (31D03360A) and cable assembly (21C01677A) shown in Figure 2-3.

Each pair of I/O ports are controlled with a 2681 DUART controller, using base addresses:

Ports 6 - 11		Ports 12 - 17	
$888000 and $888010	Ports 6, 7	$898000 and $898010	Ports 12, 13
$888020 and $888030	Ports 8, 9	$898020 and $898030	Ports 14, 15
$888040 and $888050	Ports 10, 11	$898040 and $898050	Ports 16, 17

When installed, the Expansion Boards occupy Slot J1 and J2 on the ZEBRA motherboard. See Figure 2-1.

Pin assignments for J1 of the Expansion Board, and for each of the added ports are defined in Figure 2-4. When installed, the added ports are accessed from the back panel, mounted directly above the basic six ports.

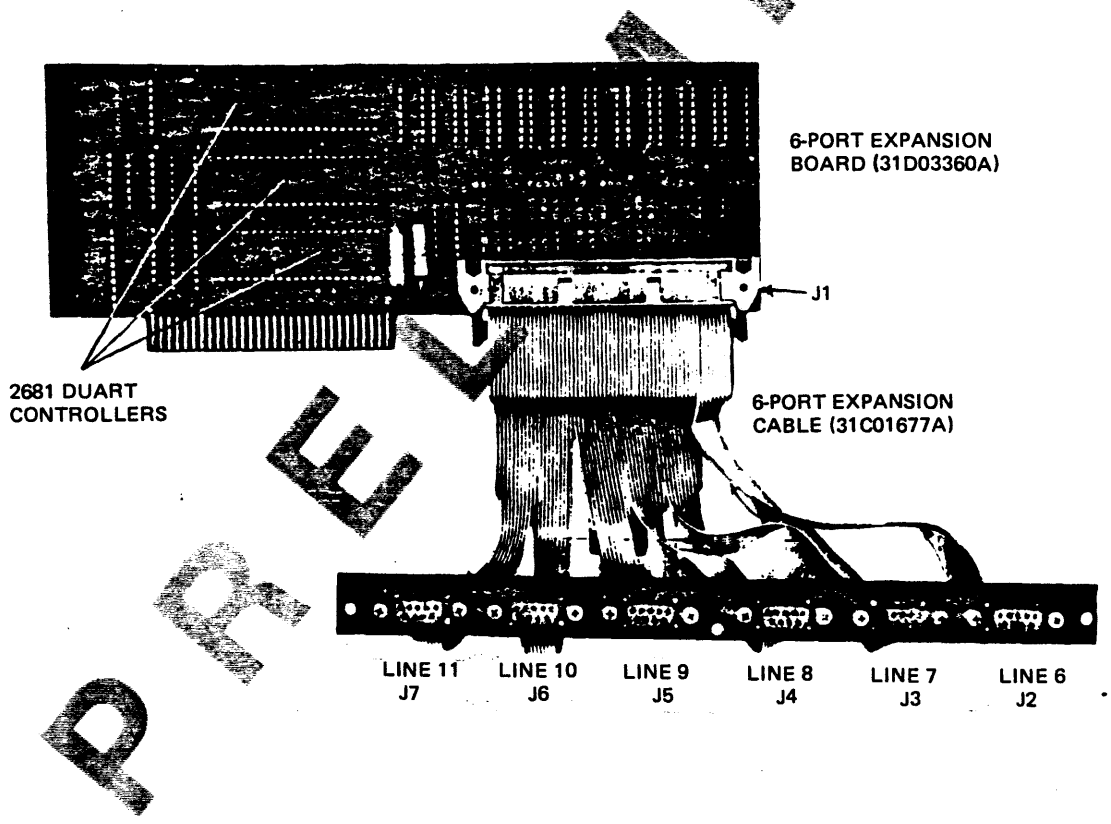


Figure 2-3. Six-Port Expansion

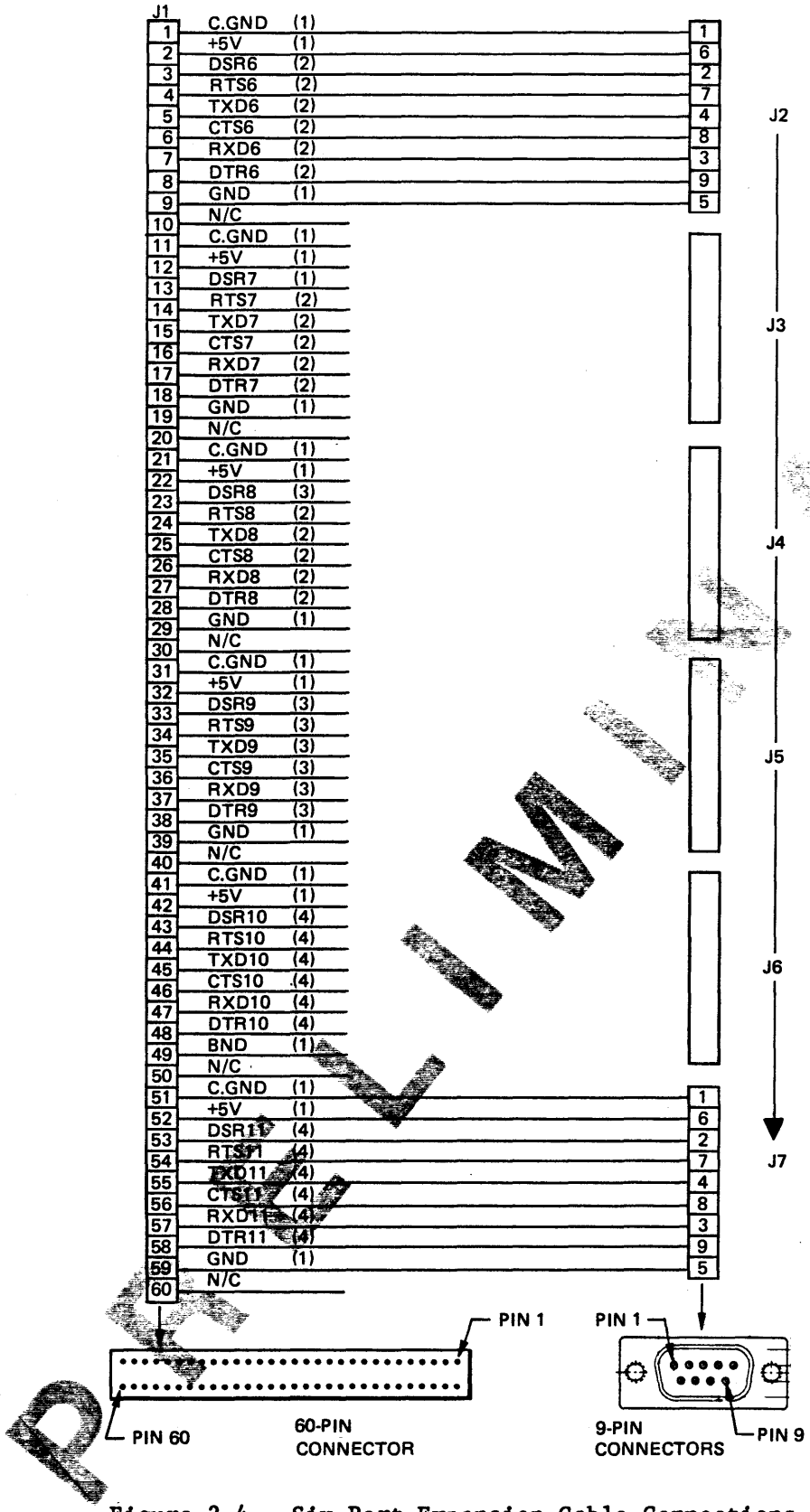


Figure 2-4. Six-Port Expansion Cable Connections

2.7 PARALLEL PRINTER INTERFACE

The Centronics-compatible parallel printer interface is located at base address $600068 and consists of three separately addressed byte-wide registers. They are as follows:

Offset from Base Address	Register Description Read	Write
+ $01	Reserved	Printer Output Data
+ $02	Reserved	Printer Control Line Command
+ $05	Printer Control Line Status	Reserved

2.7.1 PRINTER OUTPUT DATA

A write to this address puts the data on the printer interface. Two wait states are added for any access to this address.

2.7.2 PRINTER CONTROL LINE COMMAND

A write to this register is used to control the state of the printer interface control lines and the interrupt mask. Two wait states are added to accesses of this address. The bits are defined as shown below:

Bit 00 - 1 = Enable printer interrupts
0 = Mask/reset printer interrupts

Bit 01 - 1 = Printer interface line AUTOFEED asserted
0 = Printer interface line AUTOFEED deasserted

Bit 02 - 1 = Printer interface line SLCTIN* asserted
0 = Printer interface line SLCTIN* deasserted

Bit 03 - 1 = Printer interface line INIT asserted
0 = Printer interface line INIT deasserted

2.7.3 PRINTER CONTROL LINE STATUS

A read of this register is used to examine the state of the printer interface control lines and the printer interrupt flag. Two bits in this field are used for Power Fail detection (see Section 7.4). Two wait states are added to accesses of this address. The bits are defined as shown below:

Bit 00 - 1 = Printer interrupt request pending
0 = No printer interrupt pending

Bit 01 - 1 = Printer interface line BUSY asserted
0 = Printer interface line BUSY deasserted

Bit 02 - 1 = Printer interface line PE asserted
0 = Printer interface line PE deasserted

Bit 03 - 1 = Printer interface line SLCT asserted
0 = Printer interface line SLCT deasserted

Bit 04 - 1 = Printer interface line ERROR asserted
0 = Printer interface line ERROR deasserted

These bits are used by the Power Fail circuitry:

Bit 05 - 1 = Power fail input asserted
0 = Power fail input deasserted

Bit 06 - 1 = Power fail flag set
0 = Power fail flag not set

The printer interface interrupts the microprocessor at auto-vector level 3 (INT3*) whenever the ACK line on the interface transitions from a high to a low.

2.7.4 CONNECTOR PIN ASSIGNMENTS

The connector pin assignments for the parallel printer Port J6 are listed in Table 2-5.

Table 2-5. Parallel Printer J6 Pin Assignments
(All unlisted pins are GND)

90C03367A

Pin No.	Signal Mnemonic	Pin No.	Signal Mnemonic
1	STROBE*	11	BUSY
2	DATA1	12	PE
3	DATA2	13	SLCT
4	DATA3	14	AUTOFEED*
5	DATA4	17	CGND
6	DATA5	31	INIT*
7	DATA6	32	ERROR
8	DATA7	35	PU8
9	DATA8	36, 34	SLCTIN*
10	ACK*		

NOTE

An asterisk following signal name indicates LOW/TRUE.

2.8 BUS ERRORS

The MC68000 microprocessor can be forced into bus error processing by a number of conditions:

1. Access to nonexistent device above address $800000,

2. SASI bus interface timeout,

3. Segment map protection violation,

4. Invalid page access, or

5. Memory parity error.

Items 3 and 4 above can only occur in a ZEBRA 1700 System with MMU board installed. The following logic can be used to determine the source of a bus error:

1. If the bus error address is greater than $1FFFFF and the microprocessor was in user mode, illegal access was made from user mode.

2. If the bus error address is greater than $7FFFFF, an access to a nonexistent device was made.

3. If the bus error address is in the range from $600060 to 600063 and the SASI Bus Error Flag is set, there was a SASI bus timeout.

4. If the address and the CPU status indicated a violation of the segment protection in the memory map, then that was the cause of the bus error.

5. If the page entry for the specified address indicates an invalid page, then that was the cause of the bus error.

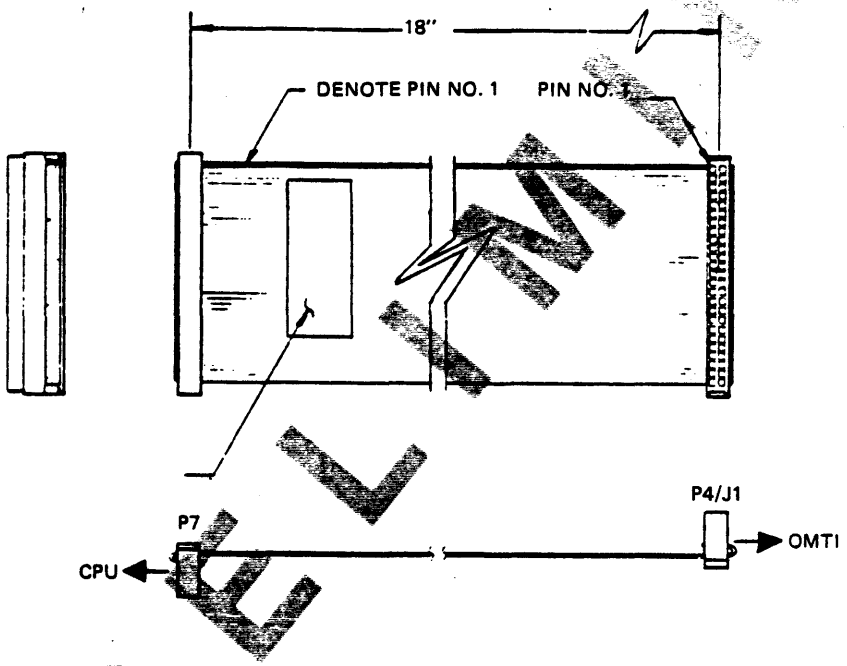
6. At this point, it must be assumed that the bus error was caused by a memory parity error. Parity will not be detected until the next memory cycle; therefore, the address will not be the exact address of the parity error. In most cases, it should be relatively close.

Any condition not specified above cannot be the cause of a bus error.

Shugart Associates System Interface (SASI) 3

3.1 SASI SIGNALS

SASI provides nine (9) control signals and nine (9) data signals (including parity). These signals are carried on a 50-pin SASI Bus with a 2-connector cable, 21C01658A; or with a 3-connector shielded cable, 21C01714A, which connects CPU, OMTI, and the System Expansion Enclosure.



The signals, signal descriptions, and pin assignments for each cable are summarized in Table 3-1.

Table 3-1. SASI Signal Pin Assignments (J7) (Sheet 1 of 2)

Signal	Driven By	Pin	Description
			Data bus between controller and ZEBRA for bi-directional signals.
Data Bit 0		2	Each of the eight data signals (0
Data Bit 1		4	through 7) is uniquely assigned
Data Bit 2		6	as a receiver or Initiator's own
Data Bit 3	Z/C	8	Bus address (i.e., Bus Device ID).
Data Bit 4		10	This Bus Device ID would normally
Data Bit 5		12	be assigned and strapped in a
Data Bit 6		14	Bus Device during system
Data Bit 7		16	configuration.
ATN* (Attention)		32	A signal driven by an Initiator to indicate the Attention condition.
BSY* (Busy)	C	36	Assertion by the controller indicates that the controller has control of the interface bus and can't be interrupted.
ACK* (Acknowledge)	Z	38	Assertion by ZEBRA indicates data has been accepted or that data is ready to be transferred from ZEBRA to the controller.
RST* (Reset)	Z	40	Assertion by ZEBRA causes the controller to cease all operations and return to the IDLE condition. This signal is normally used during a power up sequence. A RESET during a write operation would cause incorrect data to be written on the disk. The RESET pulse should be at least 1 microsecond wide.

NOTE: Z = ZEBRA,
C = Controller,
* = Neg true or active low,
All odd pins GND.

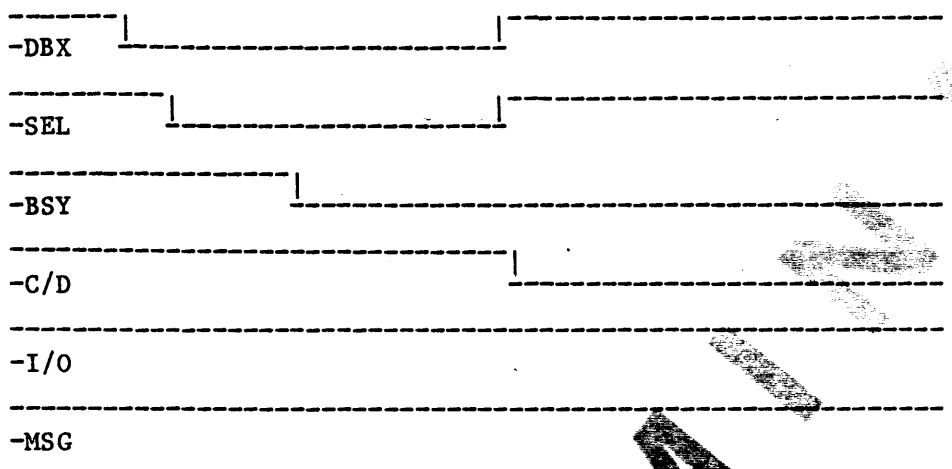
Table 3-1. SASI Signal Pin Assignments (J7) (Sheet 2 of 2)

Signal	Driven By	Pin	Description
MSG* (Message)	C	42	Assertion by the controller indicates that a status byte transfer has been accomplished. When MESSAGE is asserted, REQUEST will be asserted in order to transfer an 8-bit byte indicating the end of the operation. When the REQ/ACK handshake is complete, the controller will deassert all interface signal lines and return to the IDLE state.
SEL* (Select)	Z	44	Assertion by ZEBRA with the controller address bit (DB0) causes the controller (Initiator or Receiver) to be selected. The SELECT line must be deasserted after the controller asserts the BUSY line.
C/D* (Control/Data)	C	46	Assertion by the controller indicates that control or data information is to be transferred on the data bus. Deassertion of this line indicates that data information is to be transferred on the data bus. True indicates control.
REQ* (Request)	C	48	Assertion by the controller indicates that an 8-bit byte is to be transferred on the data bus. REQUEST is deasserted following assertion of the ACKNOWLEDGE line.
I/O* (Input/Output)	C	50	Assertion by the controller indicates that information will be transferred to the host from the controller. Deassertion indicates that information will be transferred to the controller from ZEBRA.

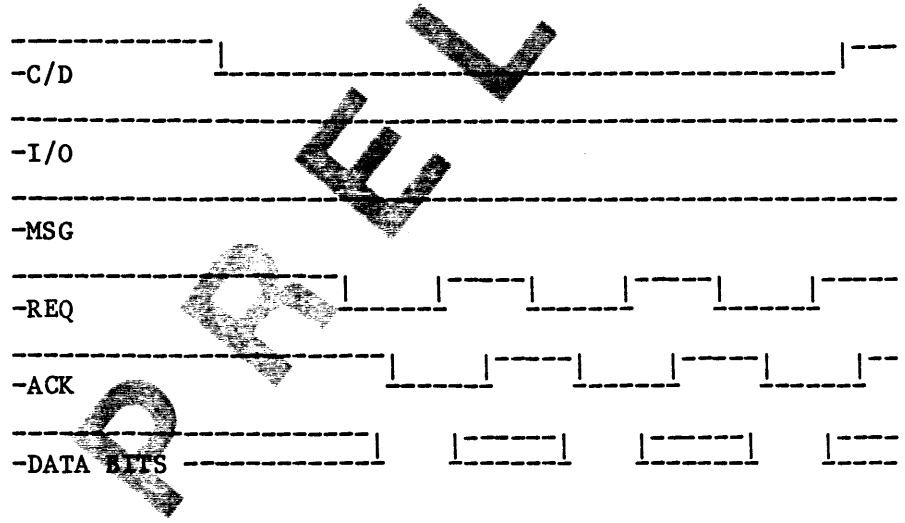
NOTE: Z = ZEBRA,
C = Controller,
* = Neg true or active low,
All odd pins GND.

3.2 SASI SELECTION SEQUENCE

To gain controller attention, the following Select sequence occurs. The host must first test BSY to determine if the controller is available. If BSY and all other I/O lines are deasserted, the host will assert one of the data lines (DBX = controller ID) and then assert SEL. The controller will then respond by asserting BSY. At this point, the host must deassert SEL and DBX. The controller responds to SEL deasserted by asserting C/D. I/O remains deasserted throughout the selection sequence. The relative timing of signals is illustrated below.



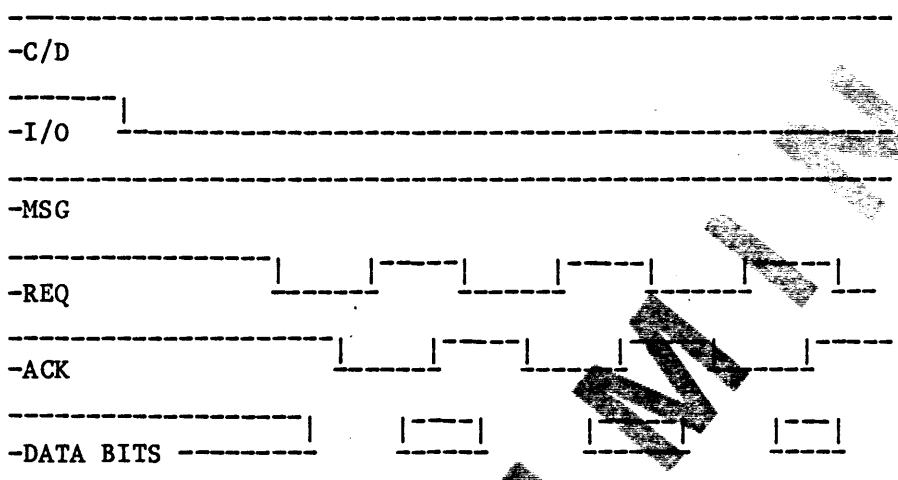
Following the Select sequence, the controller will assert Request. The host will then place the first byte of the device command field on the data bus. The host will then assert ACK. The controller will then respond by reading the byte on the data bus and then deassert REQ. The host must then deassert ACK to begin the next REQ/ACK handshake. The handshake continues until all bytes of the command have been transferred. SEL is deasserted and BSY is asserted throughout this sequence.



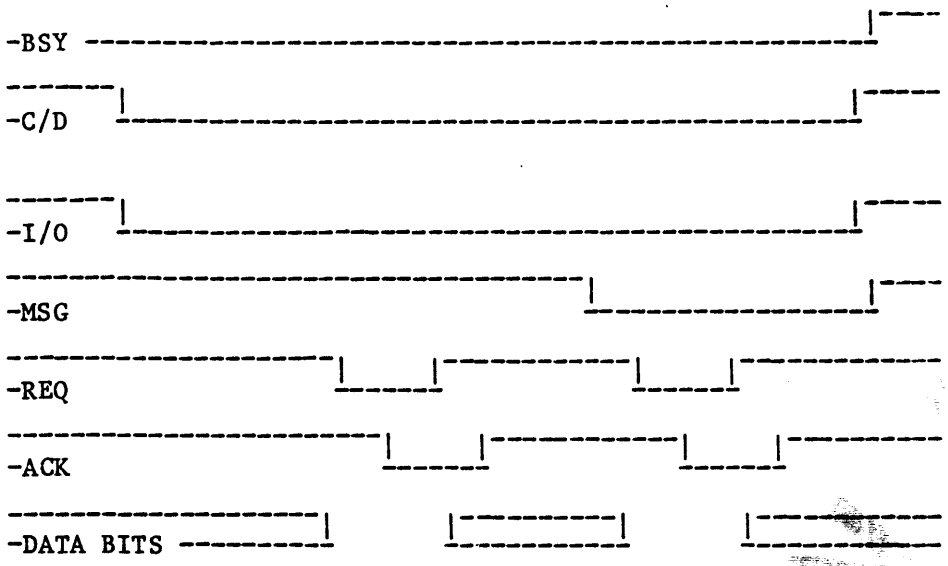
3.3 SASI DATA TRANSFER SEQUENCE

If the command sent to the controller involves a data transfer, the controller will deassert the C/D line to indicate a data transfer. If the data transfer is from the controller to the host (read data), the I/O line will be asserted. If the data transfer is from the host to the controller (write data), the I/O line will be deasserted. The controller will then assert the REQ line to request a byte transfer. The host responds by transferring a byte across the data bus and then asserts ACK. This handshake continues until all data bytes have been transferred. SEL is deasserted and BSY is asserted throughout this sequence.

Data to the Host



Following a command or data transfer, the controller will initiate a status byte and completion message transfer. When a status byte is required, the controller will assert C/D and I/O. The controller will then assert REQ. The host must then accept the status byte on the data bus and assert ACK. The controller will then deassert REQ and the host deasserts ACK. Following the status byte transfer, a completion message will be transferred to indicate operation complete. The controller will assert the MSG line, along with C/D and I/O, then assert REQ. The host accepts the completion message byte on the data bus and asserts ACK. The controller then responds by deasserting REQ and the host by deasserting ACK. At this point, BSY and all other controller signal lines will be deasserted and the controller will return to an IDLE state. SEL remains deasserted throughout this sequence.



Following the controller selection sequence, the controller will always request a Device Command Field (DCF) which is either 6 or 10 bytes in length. The first byte of the DCF must always contain the command. The remaining bytes specify the drive Logical Unit Number (LUN), logical sector address, number of sectors to be transferred, and a control byte.

The controller checks the validity of incoming DCFs. An error in the command structure will terminate the command and cause a status byte transfer to occur upon completion of the DCF transfer.

3.4 SASI/CPU INTERFACE

The SASI/CPU interface is located at base address $600060 and consists of four separately addressed byte-wide registers. They are as follows:

Offset from Base Address	Register Description Read	Write
+ $01	Reserved	Data Write Register
+ $03	Data Read Register	Reserved
+ $04	Reserved	SASI Control Line Command
+ $07	SASI Control Line Status	Reserved

3.4.1 DATA WRITE REGISTER

A write to this address puts the data on the SASI data lines if the SASI controller has asserted the proper control lines. A write to this register is normally done while the REQ* SASI line is asserted. The interface then automatically asserts the SASI ACK* line. The SASI controller then deasserts the REQ* line which causes the microprocessor to be 'DTACKed' and the write cycle completed.

If the SASI controller does not deassert REQ* within 50 usec, a timeout occurs, a bus error is sent to the microprocessor, and the SASI Bus Error Flag is set.

3.4.2 DATA READ REGISTER

A read of this address gets the data on the SASI data lines if the SASI controller has asserted the proper control lines. A read from this register is normally done while the REQ* SASI line is asserted. The interface then automatically asserts the SASI ACK* line. The SASI controller then deasserts the REQ* line which causes the microprocessor to be 'DTACKed' and the read cycle complete.

If the SASI controller does not deassert REQ* within 50 usec, a timeout occurs, a bus error is sent to the microprocessor, and the SASI Bus Error Flag is set.

3.4.3 SASI CONTROL LINE COMMAND

A write to this register is used to control the state of the SASI bus control lines and to reset the SASI Bus Error Flag. Two wait states are added to accesses of this register. The bits are defined as shown below:

Bit 00 - 1 = SASI bus line RST* asserted
0 = SASI bus line RST* deasserted

Bit 01 - 1 = SASI bus line SEL* asserted
0 = SASI bus line SEL* deasserted

Bit 02 - 1 = SASI bus line ATN* asserted
0 = SASI bus line ATN* deasserted

Bit 03 - 1 = Hold SASI Bus Error Flag in reset state
0 = Enable flag to detect SASI Bus Error condition

3.4.4 SASI CONTROL LINE STATUS

A read to this register is used to examine the state of the SASI bus control lines and the SASI Bus Error Flag. Two wait states are added to accesses of this register. The bits are defined as shown below:

Bit 00 - 1 = SASI bus line I/O*, data from SASI bus
0 = SASI bus line I/O*, data to SASI bus

Bit 01 - 1 = SASI bus line REQ* asserted
0 = SASI bus line REQ* deasserted

Bit 02 - 1 = SASI bus line BSY* asserted
0 = SASI bus line BSY* deasserted

Bit 03 - 1 = SASI bus line MSG* asserted
0 = SASI bus line MSG* deasserted

Bit 04 - 1 = SASI bus line C/D*, command phase
0 = SASI bus line C/D*, data phase

Bit 05 - 1 = SASI Bus Error Flag set
0 = SASI Bus Error Flag reset

The SASI bus interface interrupts the microprocessor at auto-vector level 1 whenever the SASI bus line REQ* is asserted.

4 memory management unit (MMU)

The ZEBRA 1700 Memory Management Unit (MMU) provides a segmented, paged memory management system to facilitate the effective use of memory under the XENIX Operating System.

The MMU board (Figure 4-1) consists of a MC68010L10 microprocessor and a two-level memory map. The MMU board must be mounted in J3 or J4 of the CPU board (see Figure 1-2) adjacent to the last memory board.

The MMU MC680010L10 microprocessor replaces the MC68000L10 (located on motherboard for the ZEBRA 1750) and drives the internal CPU bus directly.

A multi-tasking operating system needs a means to switch quickly between contexts. For example, assign CPU work on a new job while a previous job is suspended or waiting for a peripheral transfer to complete. This is accomplished by providing separate contexts (addressed through a Context Register) to point the CPU at a fresh set of segments and pages. Figure 4-2 illustrates a block diagram of the memory management system.

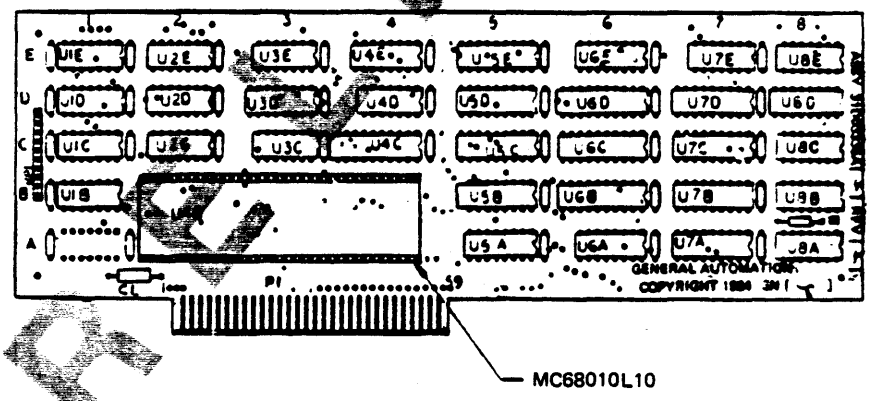


Figure 4-1. ZEBRA 1700 Memory Management Unit, 31C03335A

The MMU provides address translation, sharing, and memory allocation control for multiple processes executing on the CPU. The address space is divided into pages of 2K bytes each. The page address bits, A1-A10, pass through the translation process unmodified. Address bits A11-A20 are subject to translation. Bits A21-A23 are reserved for special system functions and take no part in the address translation. The maximum logical address space for a process on the CPU is 21 bits or 2MB. This 21-bit address is further extended with a 4-bit Context Register (referred to as the process or user number).

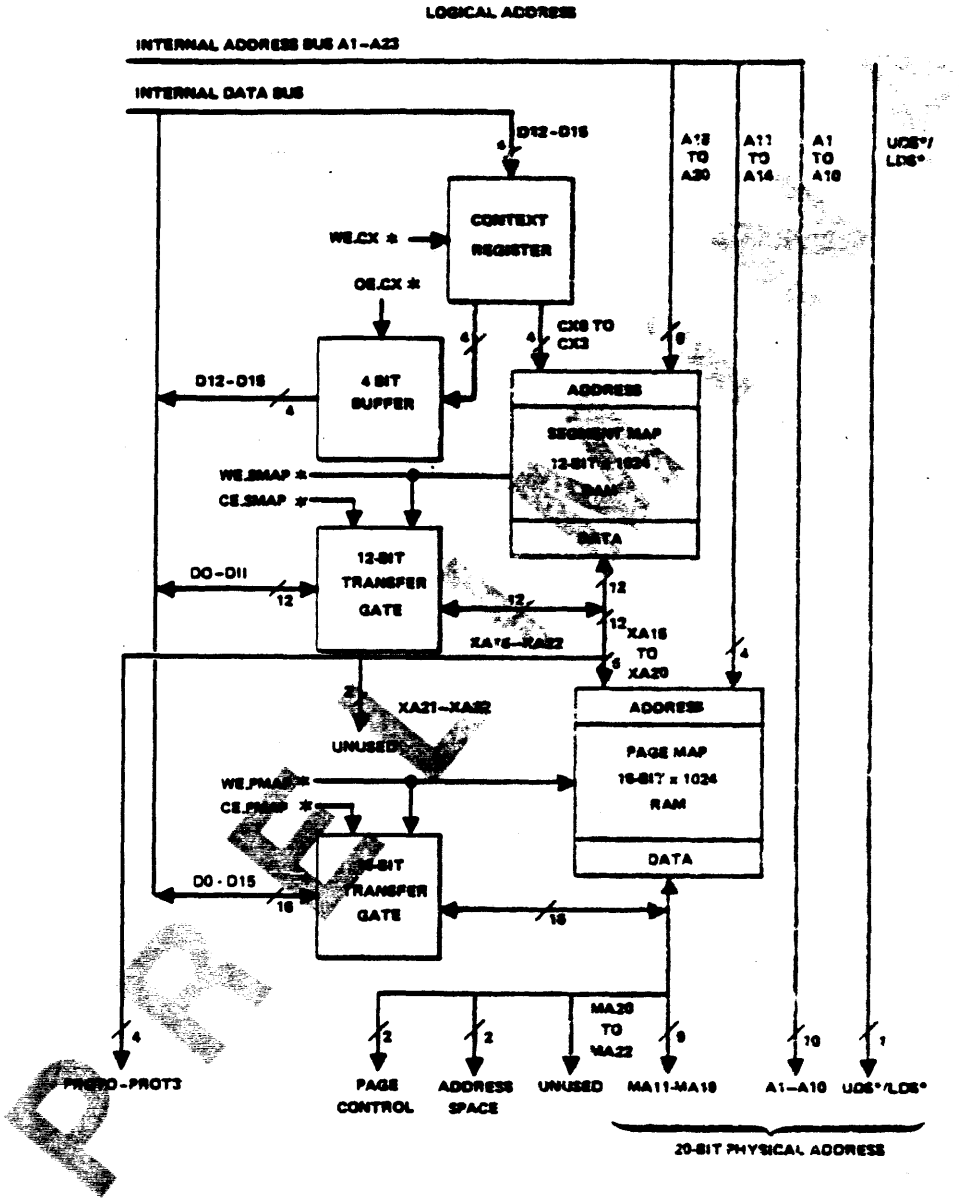


Figure 4-2. ZEBRA 1700 Memory Management Unit Block Diagram

A0 is not generated by the microprocessor. The microprocessor outputs Lower Data Strobe* (LDS*) and Upper Data Strobe* (UDS*) to select the appropriate byte or bytes from a 16-bit word addressed by A1-A23.

The 23-bit logical addresses, A1-A23, from the microprocessor are translated into 20-bit physical addresses in two stages. In the first stage, the logical address from the microprocessor is translated by the Segment Map look-up table into a virtual address, XA15-XA20. In the second stage, this virtual address is translated by the Page Map look-up table into a 19-bit physical address, A1-A10 and MA11-MA19. This 19-bit address is used together with the LDS* and UDS* to form a 20-bit physical address. This is the address that is then applied to the CPU bus on the motherboard.

Page control and address space control are provided at the Page Map level. Page access control consists of two bits which remember that a page has been referenced, used, and written to. Address space control determines in which physical address space a page is located and whether it references memory or input/output. Since no input/output addressing is done on card by the MMU, this designation is interpreted as an invalid page; in which case, a reference to a word in that page causes a page fault.

4.1 CONTEXT REGISTER

A system with multiple executing processes must be able to switch between processes without reloading state information relating to address translation for a particular process. The Context Register (see Figure 4-3) is a four-bit register, writable and readable under supervisor controls, that selects 1 of 16 unique sections of the Segment Map. This memory management method contains maps for 16 distinct process or user translations simultaneously.

4.2 SEGMENT MAP

The Segment Map is a 1024-entry table indexed by the four-bit Context Register and the six most significant bits of the logical address, A15-A20. The output of the Segment Map is six virtual address bits (XA15-XA20) and four protection bits (PROT 0 through PROT 3). Each context has up to 64 segments and each segment has individual protection attributes. Segments may be kept private to a process or shared with other processes. The six-bit virtual address from a segment entry refers to a block of 16 consecutive page entries in the 1024-entry Page Map. A segment can be as large as 32K bytes (by using all 16 of the associated Page Map entries) or as small as 2K bytes (by invalidating unused page entries in the Page Map). By linking consecutive segment map entries, a process can have a single address space of 2MB.

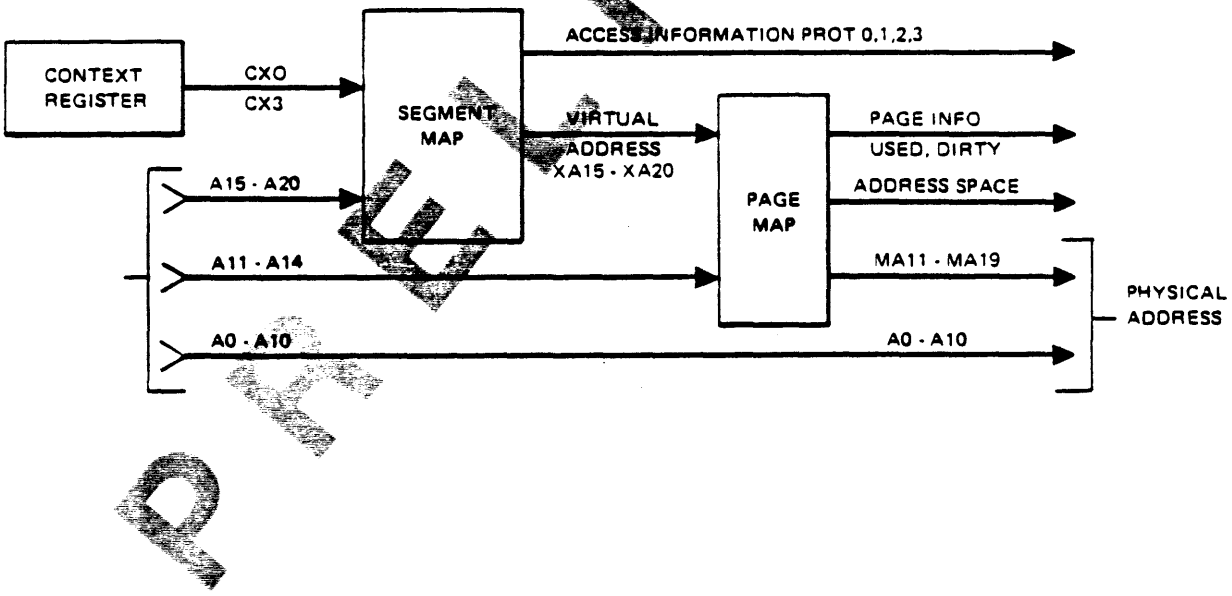


Figure 4-3. ZEBRA 1700 Memory Management Unit Address Translation

4.3 SEGMENT LEVEL PROTECTION

Protection is associated with the Segment Map. Four protection bits (PROT 0 through PROT 3) are provided to disallow read, write, and execute access to two levels (the system level and the user level). Refer to Table 4-1.

Full access is denoted rwxrwx where the first rwx applies to system access and the second rwx to user access.

Table 4-1. Segment Level Protection Attributes

Protection Code	Protection Bits PR3	PR2	PR1	PR0	Access Allowed System*	User*
0	0	0	0	0	- - -	- - -
1	0	0	0	1	- - x	- - -
2	0	0	1	0	r - -	- - -
3	0	0	1	1	r - x	- - -
4	0	1	0	0	r w -	- - -
5	0	1	0	1	r w x	- - -
6	0	1	1	0	r - -	r - -
7	0	1	1	1	r w -	r - -
8	1	0	0	0	r - -	r w -
9	1	0	0	1	r w -	r w -
10	1	0	1	0	r w -	r - x
11	1	0	1	1	r w -	r w x
12	1	1	0	0	r - x	r - x
13	1	1	0	1	r w x	r - x
14	1	1	1	0	r w x	- - x
15	1	1	1	1	r w x	r w x

*r = Read
w = Write
x = Execute
- = Attribute not enabled

4.4 PAGE MAP

In the Page Map, the 6-bit virtual address from the Segment Map and the next four logical address bits (A11-A14) from the microprocessor are translated into a physical address and a physical address space. Each segment virtual address refers to a block of 16 consecutive page entries in the Page Map.

The output of the Page Map is the upper ten bits of the physical address which is linked with the lower eleven bits of the logical address to form a 21-bit Physical Address.

In addition to determining the upper ten bits of the physical address, a page entry also determines to which physical address space the address belongs. By setting the address space control bits appropriately, a page may be declared to be in one of these address spaces:

Page Definition Bits	Page Type
00b	Valid memory page
01b	Invalid memory page
10b	Reserved code
11b	Reserved code

Access to pages marked invalid or reserved will cause a bus error to be sent to the microprocessor.

Each page map entry has two bits of page access control information. The referenced bit indicates that this page has been referenced:

1. Data read reference
2. Data write reference
3. Execute reference

The modified bit indicates that this page has been written to. These bits are automatically updated on every valid mapped reference. These bits are intended for future use in virtual memory systems.

4.5 MEMORY MAP ADDRESSING

The Memory Map can be accessed only from supervisor mode. Any access from user mode of the map resources causes a bus error. Table 4-2 illustrates the method of addressing and modifying the contents of the memory map registers.

Table 4-2. Memory Map Register Access

Address	Attribute	Register	Data Bus Bit Specification 15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
$A00000	r/w	Page Map	DI	US	PD 01	PD 00	na	na	MA 20	MA 19	MA 18	MA 17	MA 16	MA 15	MA 14	MA 13	MA 12	MA 11
$C00000	w	Segment Map	na	na	na	na	PR 03	PR 02	PR 01	PR 00	na	na	XA 20	XA 19	XA 18	XA 17	XA 16	XA 15
$C00000	r	Seg Map Cntx Reg	CX 03	CX 02	CX 01	CX 00	PR 03	PR 02	PR 01	PR 00	na	na	XA 20	XA 19	XA 18	XA 17	XA 16	XA 15
$E00000	w	Context Register	CX 03	CX 02	CX 01	CX 00	na	na	na	na	na	na	na	na	na	na	na	na
$E00000	r	na	na	na	na	na	na	na	na	na	na	na	na	na	na	na	na	na

NOTES

1. The 64 segments for each context are accessed at intervals of $8000 for each entry from the base address of $C00000. The context register must be set to the desired context.

2. The 1024 page entries are addressed at intervals of $800 from the base address of $A00000.

3. Attribute r = Read
   w = Write
   r/w = Read or Write

4. DI = Dirty bit (1 = dirty)
   US = Used bit (1 = used)

5. PDnn = Page definition bits
   MAnn = Translated physical memory address
   PRnn = Protection bits
   XAnn = Translated virtual segment address
   CXnn = Context register bits

5 Winchester/cartridge subsystems

The ZEBRA 1700/1750 can be equipped with one of two Winchester/Cartridge subsystems (see Figures 1-1 and 1-2). The standard version employs the 5-1/4" Winchester and a removable cartridge disk (CD) drive. The other version provides the Winchester, but replaces the cartridge disk with a 1/4" cartridge tape (CT). Both versions are connected to the CPU/SASI (Shugart Associates System Interface) bus by an OMTI controller.

The ZEBRA 1700/1750 CPU/SASI bus can be cabled to drive the stand-alone Expansion Enclosure, a system option which can be used in conjunction with either version described above. The Expansion Enclosure is a separate unit, of the same dimensions as the ZEBRA 1700/1750 cabinet.

5.1 OMTI CONTROLLER

The OMTI controller implements SCSI (Small Computer System Interface), the proposed ANSI standard. SCSI provides ZEBRA with an Input/Output (I/O) device independence, allowing disk drives, tape drives, and future devices to be added without special drives or special modification to system hardware and software.

5.1.1 OMTI SIGNAL LINES

The signal lines between ZEBRA and OMTI are carried by the 50-line SASI bus. See Section 3, Table 3-1.

5.1.2 OMTI COMMANDS

ZEBRA initiates an OMTI command sequence by first selecting the OMTI controller which will then request a DCF (Device Command Field). The DCF is 6 or 10 bytes in length, the first byte being the command. The remaining bytes specify the drive LUN (logical unit number), logical sector address, and the number of sectors to be transferred. The ZEBRA OMTI commands are described in OMTI literature (see FOREWORD).

In summary, the ZEBRA OMTI commands are:

Command	Hex Code
SENSE STATUS	00
RECALIBRATE	01
REQUEST SENSE	03
FORMAT DRIVE	04
CHECK TRACK FORMAT	05
FORMAT TRACK	06
FORMAT BAD TRACK	07
READ DATA	08
WRITE DATA	0A
SEEK	0B
ASSIGN ALTERNATE TRACK	0E
CHANGE CARTRIDGE	1B
COPY	20
BACKUP	22*
RESTORE	23*
BACKUP WITH HEADER	24*
RESTORE CONTINUE	26*
RESET TAPE	44*
VERIFY TAPE	45*
READ SENSE	46*
WRITE HEADER	48*
READ HEADER	49*
WRITE BLOCKS	4A*
READ BLOCKS	4B*
DEFINE FLEXIBLE DISK FORMAT	C0
ASSIGN DISK PARAMETERS	C2
RAM DIAGNOSTICS	E0
WRITE ECC	E1
READ ID	E2
READ DATA BUFFER	EC
WRITE DATA BUFFER	EF

*Tape Commands

5.1.3 OMTI CONTROLLER

Two versions of the OMTI controller are provided for the driving of a disk and a cartridge disk (OMTI 5100) or a disk and a cartridge tape (OMTI 5300). Figures 5-1 and 5-2 show a layout of each board with identification of cables, connectors, jumper positions and their function.

5.1.4 INTERFACE CONNECTIONS

The following listings summarize pin assignments and jumper positions for the OMTI controllers. For further detail, refer to OMTI manuals identified in the FOREWORD of this manual.

5.1.4.1 5-1/4" Winchester Disk Drive Cable

The following listing defines the pin assignments for the 5-1/4" Winchester disk drives supported.

Control Cable (21C01659A):

	Pin #		Fixed	Removable
GND	1	2	Head Select 3/MSI	Change Cartridge
	3	4	Head Select 2	
	5	6	Write Gate	
	7	8	Seek Complete	
	9	10	Track 000	
	11	12	Write Fault	
	13	14	Head Select 0	
	15	16	Reserved	Sector Pulse
	17	18	Head Select 1	
	19	20	Index	
	21	22	Ready	
	23	24	Step	
	25	26	Drive Select 1	
	27	28	Drive Select 2	
	29	30	Drive Select 3	
	31	32	Drive Select 4	
GND	33	34	Direction Select	

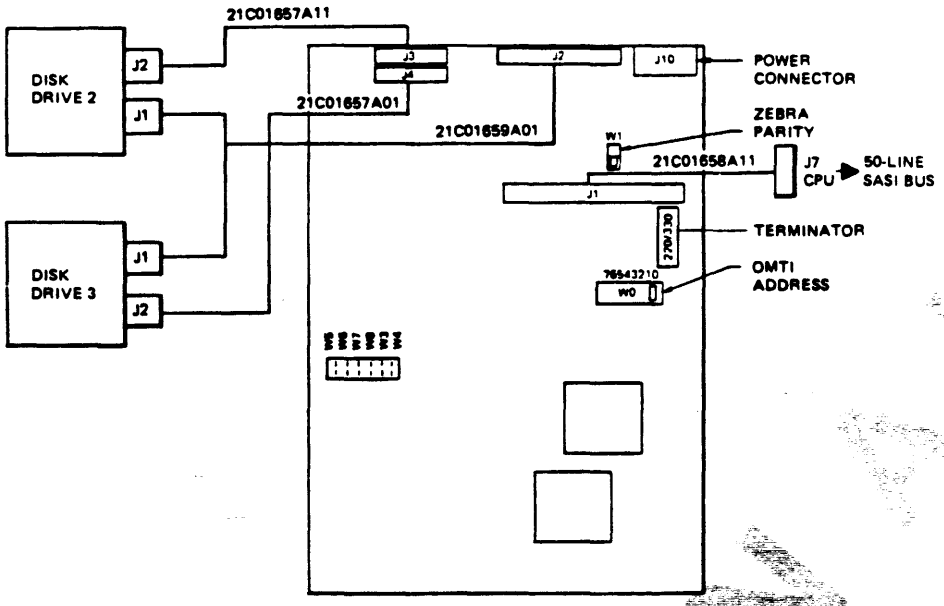


Figure 5-1. OMTI 5100 (31A08004A05) Interconnection

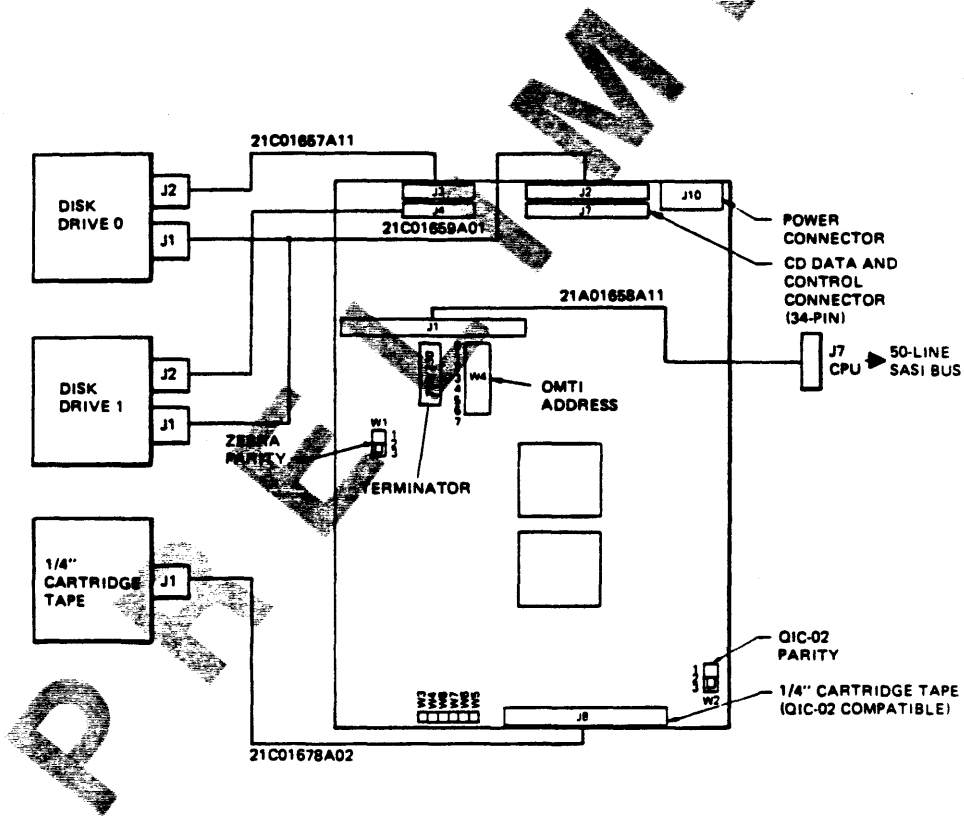


Figure 5-2. OMTI 5300 (31A08004A02) Interconnection

Data Cable (21C01657A):

Pin #	Fixed	Removable
1	Drive Selected	
2	Ground	
3	Reserved	
4	Ground	
5	Reserved	Write Protected
6	Ground	
7	Reserved	
8	Ground	
9	Reserved	Cartridge Changed
10	Reserved	
11	Ground	
12	Ground	
13	+MFM Write Data	
14	-MFM Write Data	
15	Ground	
16	Ground	
17	+MFM Read Data	
18	-MFM Read Data	
19	Ground	
20	Ground	

5.1.4.2 ZEBRA SASI Cables (CPU to OMTI)

Refer to Section 3.0.

5.1.4.3 Power Cable Connector

Power is applied to the controller via J10, a 4-pin AMP connector.

J10:

NC	Pin 1
NC	Pin 2
+ 5RTN	Pin 3
+ 5VDC	Pin 4

5.1.4.4 1/4" Cartridge Tape Cable (21C01678A)

The following listing defines the pin assignments for 1/4" Cartridge Tape drives. The QIC-02 Standard specifies the hardware device interface for 1/4" cartridge tape drives. ZEBRA 1700/1750 systems using 1/4" cartridge tape utilize tape drives with QIC-02 interface which format the data on a cartridge according to the QIC-24 standard.

Tape Cable (J8):

Pin #			Signal
GND	1	2	Not Used
GND	3	4	Not Used
GND	5	6	Not Used
GND	7	8	Not Used
GND	9	10	-Parity
GND	11	12	-Data Bit 7
GND	13	14	-Data Bit 6
GND	15	16	-Data Bit 5
GND	17	18	-Data Bit 4
GND	19	20	-Data Bit 3
GND	21	22	-Data Bit 2
GND	23	24	-Data Bit 1
GND	25	26	-Data Bit 0
GND	27	28	-On Line
GND	29	30	-Request
GND	31	32	-Reset
GND	33	34	-Transfer
GND	35	36	-Acknowledge
GND	37	38	-Ready
GND	39	40	-Exception
GND	41	42	-Direction
GND	43	44	Not Used
GND	45	46	Not Used
GND	47	48	Not Used
GND	49	50	Not Used

5.1.4.5 Jumper Settings

The OMTI controllers allow the user to select (with jumper) various controller functions (see Figures 5-1 and 5-2). These functions are as follows:

OMTI 5300

W4	0 Jumpered 1 through 7 Open	Controller Address = 0
W1	Pins 2 - 3 Jumpered	Parity Disabled
W2	Pins 2 - 3 Jumpered	QIC-02 Parity Disabled
W3	Jumpered	Disk Sector Size: 256 Bytes
W4	Open	Disk Sector Size: 256 Bytes
W5	Open	Disk, LUN 0
W6	Open	Disk, LUN 1
W7	Open	Disk, LUN 2
W8	Open	Disk, LUN 3

OMTI 5100

W0	0 Jumpered 1 through 7 Open	Controller Address = 0
W1	Pins 2 - 3 Jumpered	Parity Disabled
W3	Jumpered	Disk Sector Size: 256 Bytes
W4	Open	Disk Sector Size: 256 Bytes
W5	Open	Disk, LUN 0
W6	Open	Disk, LUN 1
W7	Open	Disk, LUN 2
W8	Open	Disk, LUN 3

5.11.4.6 Default Addresses

Upon power-on or any reset command, the controller defaults to the following addresses.

Winchester Disk Drives:

Step Pulse Width = 9 microseconds	(Hex 09)
Step Pulse Period = 3.0 milliseconds	(Hex 3C)
Step Mode = 0	(Hex 00)
Number of Heads = 4	(Hex 03)
Maximum Cylinder Address HI = 0	(Hex 00)
Maximum Cylinder Address LO = 153	(Hex 98)
Reduced Write Current = 128	(Hex 80)
If 256 Bytes per Sector; Sectors per Track = 32	(Hex 1F)
If 512 Bytes per Sector; Sectors per Track = 17	(Hex 10)
If 1024 Bytes per Sector; Sectors per Track = 9	(Hex 08)
If 128 Bytes per Sector; Sectors per Track = 55	(Hex 36)

Cartridge Disk Drives:

Step Pulse Width = 2 microseconds	(Hex 02)
Step Pulse Interval = 7 milliseconds	(Hex 07)
Maximum Cylinder Address = 80 cylinders	(Hex 4F)
Head Settling Delay = 22 milliseconds	(Hex 16)
Head Select Delay = 205 microseconds	(Hex CD)
Drive Select Delay = 0 milliseconds	(Hex 00)
Write Gate Delay = 1.1 milliseconds	(Hex 0B)

5.2 PERIPHERAL ERROR REPORTING

Fatal error within disk, cartridge disk, or cartridge tape will be displayed with:

FATAL {media} ERROR, LUN {drive:error}, ADDRESS {n}

where:

media =	DISK, or CT (cartridge tape), or CD (cartridge disk)
drive:error =	0:nn for disk, nn for error code 1:nn for CD, nn for error code 3:nn for CT, nn for error code
n =	record address number where the error was sensed

Table 5-1 provides a summary and description of hexadecimal error codes.

Table 5-1. Error Code Summary (Sense Byte 1)

Code (Hex)	Description
00	No error
01	No index signal
02	No seek complete
03	Write fault
04	Drive not ready
05	Drive not selected
06	No track zero found
07	Multiple drives selected
08	Not used
09	Cartridge changed
0A to 0C	Not used
0D	Operation in progress
0E to 0F	Not used
10	Tape exception (tape only)
11 and 91	Uncorrectable error in data field
12 and 92	Not used
13 and 93	No address mark in data field
14 and 94	No record found
15 and 95	Seek error
16 and 96	Not used
17 and 97	Write protected
18 and 98	Correctable ECC error
19 and 99	Bad track flag set
1A and 9A	Incorrect interleave factor
1B and 9B	Not used
1C and 9C	Unable to read alternate track data
1D and 9D	Not used
1E and 9E	Illegal direct access to alternate track
1F	Tape drive failure (tape only)
20	Invalid command
21	Illegal parameters
22	Illegal function for drive type
23	Volume overflow
24 to 2F	Not used
30	Power up diagnostic error
31	FDC 765 error (5200/5204/5400 only)
32 through 90	Not used
A0 through FF	Not used

6 memory and I/O expansion

The ZEBRA CPU (Figure 2-1) contains five 60-pin connectors for the mounting of additional memory and Input/Output (I/O) expansion options.

6.1 MEMORY EXPANSION

The memory expansion bus is a connection to the 128KB Random-Access Memory (RAM) array located on the CPU board. This connection is made through J4 and J5, 60-pin connectors. These connectors are used to drive a maximum of two boards (31C03364A) containing the additional RAM. The first memory expansion slot (J5) accommodates a 128KB or 384KB RAM board, while the second memory slot (J4) can be configured with a 512KB RAM board. Slot J5 must be filled before memory is added to J4.

Memory addresses, timing, and control signals are received by the memory card (over the J4 and J5 connectors) from the CPU. The row and column addresses are generated on the CPU board using a pair of 8-bit dynamic memory drivers. The drivers use microprocessor address lines A1-A8 to develop the row addresses and memory address MA lines MA1-MA16, together with A9 and A10, to develop the column addresses. The logic is shown in Figure 6-1.

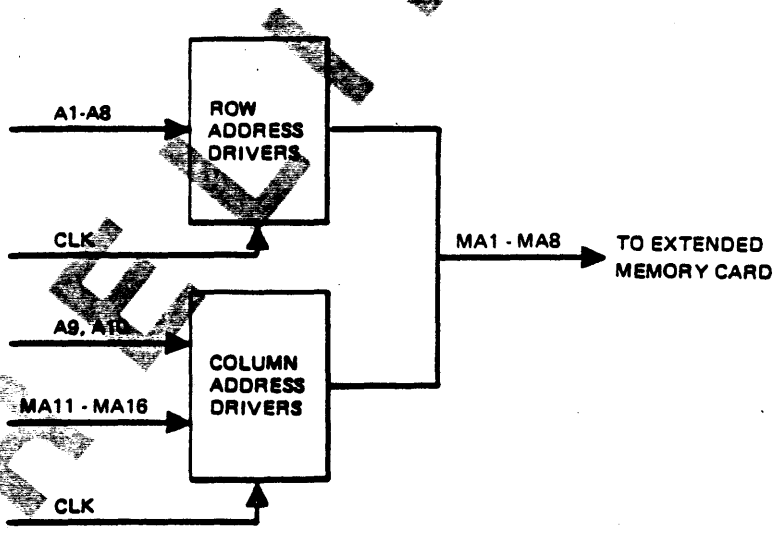


Figure 6-1. Dynamic Memory Row/Column Multiplexing Logic

The outputs of the memory drivers are multiplexed onto MA lines MA1-MA8 and output to the memory card. The memory address lines are buffered and output to six memory banks; each bank is organized as a 64 x 18 bit array. Each byte has a parity bit appended, so that the actual address space is organized as 384 16-bit words. Separate row address strobes are input from the CPU for the upper and lower bytes in each word, while the column address strobe and write-enable signals are common to both bytes. The active column address strobe is determined by memory map address line MA17-MA19 input to a decoder, which is controlled by CLK*. The write-enable signal is input from the CPU. The memory bank control and address signals are shown in Figure 6-2.

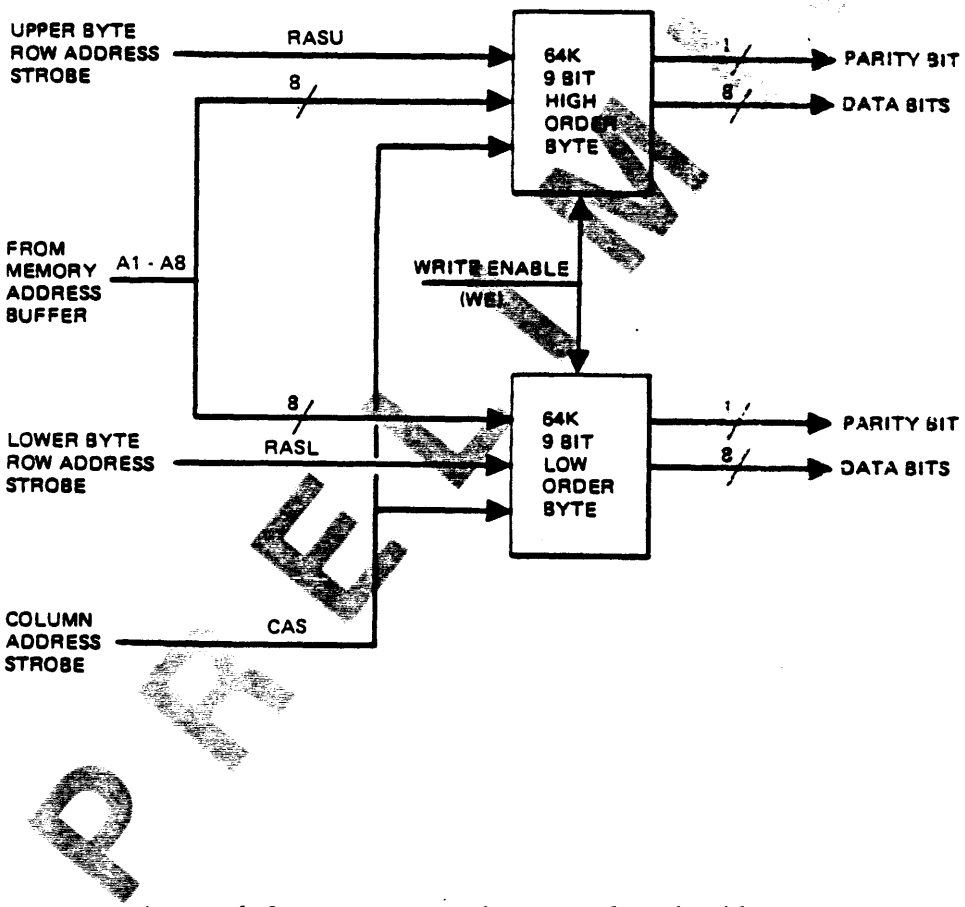


Figure 6-2. Memory Bank Control and Address Signals

6.1.1 PIN ASSIGNMENTS - J4, J5

The pin assignments for the memory expansion bus are shown in Table 6-1.

Table 6-1. Pin Assignments

90C03364A

Pin No.	Signal Mnemonic J4	J5	Pin No.	Signal Mnemonic J4	J5
1	GND	GND	31	A3	A3
2	GND	GND	32	A2	A2
3		(A)	33	(A)	GND
4		(B)	34	MD7	MD7
5	MD15	MD15	35	MD5	MD5
6		(C)	36	MD6	MD6
7	MD13	MD13	37	LDS*	LDS*
8	MD14	MD14	38	(B)	GND
9	MD12	MD12	39	(C)	+5V/GND
10	MPUI*	MPUI*	40	MD4	MD4
11	MD9	MD9	41	UDS*	UDS*
12	MD11	MD11	42	A01	A01
13	MD8	MD8	43	MD1	MD1
14	MD10	MD10	44	MD3	MD3
15	A8	A8	45	MD0	MD0
16	A16	A16	46	MD2	MD2
17	A15	A15	47	MOE*	MOE*
18	A12	A12	48	LMD*	LMD*
19	A11	A11	49	MPLO*	MPLO*
20	A7	A7	50	CASE*	CASE*
21	A6	A6	51	MPUO*	MPUO*
22	A14	A14	52	A19	A19
23	A13	A13	53	A17	A17
24	A10	A10	54	A18	A18
25	A9	A9	55	WRT*	WRT*
26	A5	A5	56	MPLI*	MPLI*
27	REN*	REN*	57	+5V	+5V
28	CEN*	CEN*	58	+5V	+5V
29	A20	A20	59	N/C	N/C
30	A4	A4	60	+5V	+5V

(A) J4-33 -> J5-3 (GND)
(B) J4-38 -> J5-4 (GND)
(C) J4-39 -> J5-6 (OPEN)

6.1.2 MEMORY EXPANSION BOARDS

The ZEBRA 1700/1750 memory can be expanded from a minimum total memory of 128KB to a maximum total of 2MB. Two versions of the memory board are available:

31C03364A Memory Board with up to 512KB
31C03378A Memory Board with up to 1MB.

6.1.2.1 512KB Board, 31C03364A

Figure 6-3 is a layout of this memory board showing the maximum (eight) memory chips, each containing 64KB. The dash number for each board that is available for ZEBRA 1700/1750 and for the earlier ZEBRA 700/750 models are summarized in Table 6-2. This table also identifies the required status of cuts for JP2 and JP3, and the required jumpers for JP1. Figure 6-4 is a logic diagram showing cut and jumper positions.

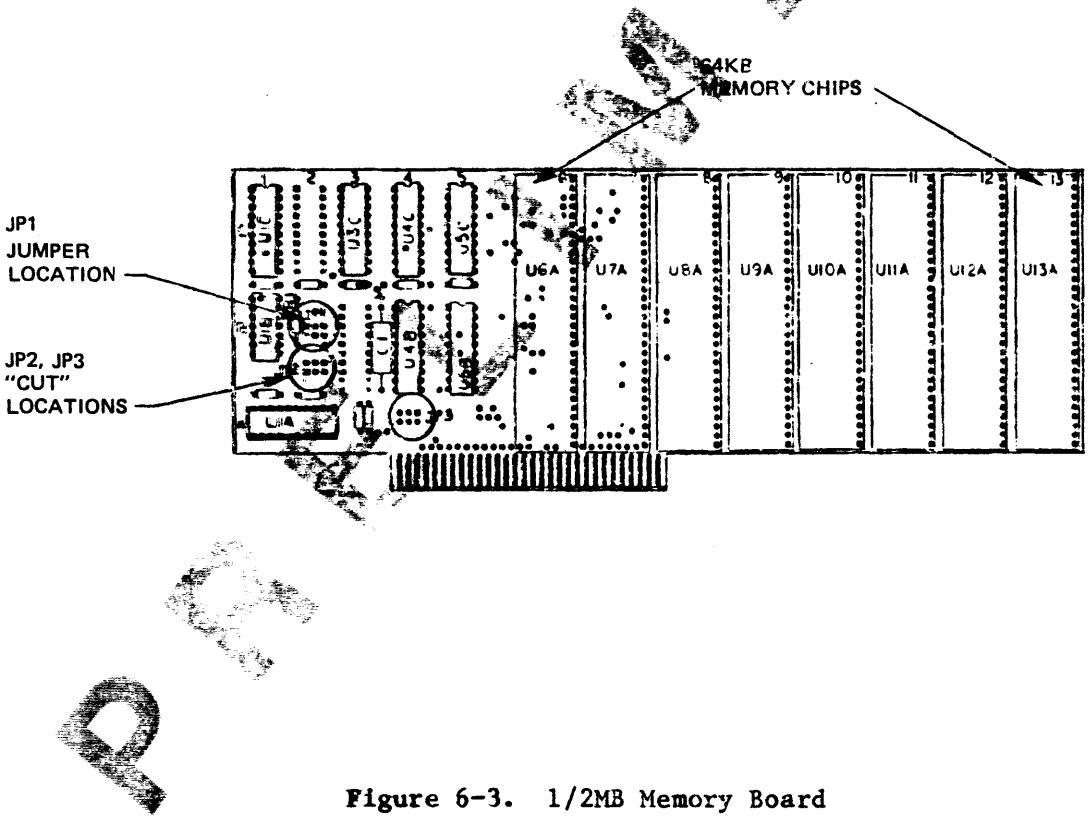


Figure 6-3. 1/2MB Memory Board

Table 6-2. Memory Board (31C03364A) Cuts/Jumpers

Dash Number	ZEBRA Model	Memory Expansion	Cuts*	Jumpers
01	1700/1750	128KB	None	None
11	1700/1750	384KB	None	None
21	1700/1750	512KB	None	None
31	700/750	128KB	JP2 1-2, 3-4, 5-6 JP3 1-2, 3-4, 5-6	JP1 1-2, 3-4, 5-6
41	700/750	384KB	JP2 1-2, 3-4, 5-6 JP3 1-2, 3-4, 5-6	JP1 1-2, 3-4, 5-6

*"Cuts" means cutting the jumpers listed and shown schematically in Figure 6-4.

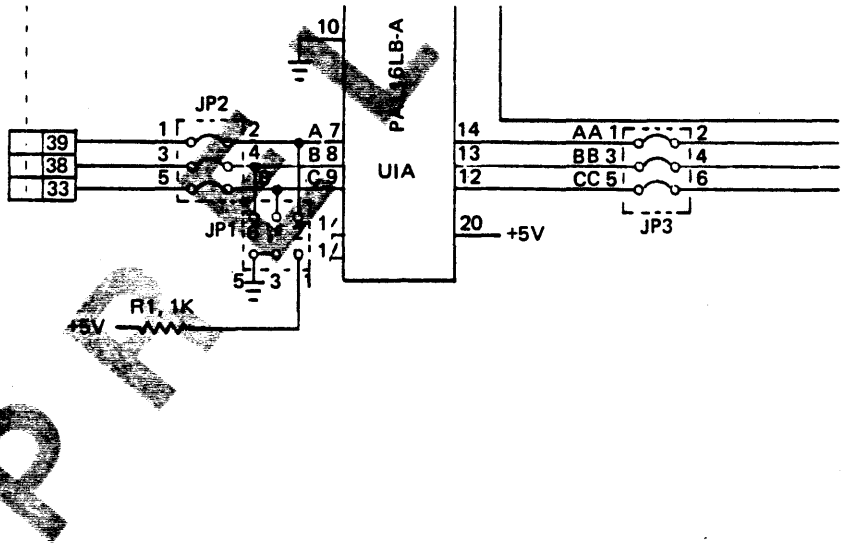


Figure 6-4. 512KB Board, Cuts/Jumpers

6.1.2.2 1MB Board, 31C03378A

Figure 6-5 is a layout of the 1MB memory board showing the maximum (36) memory chips, each containing 256K bits. The dash number for each board that is available for ZEBRA 1700/1750 and for the earlier ZEBRA 700/750 models are summarized in Table 6-3. This table also calls out the required cuts for JP2, 3, 4, and required jumpers for JP1. Figure 6-6 illustrates the cut and jumper locations.

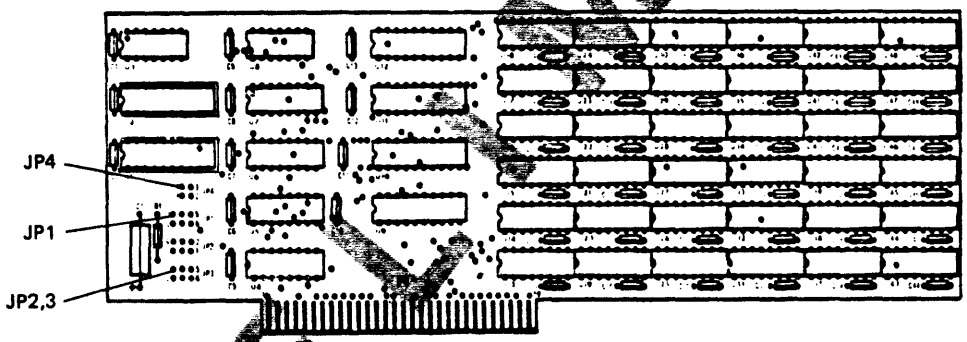


Figure 6-5. 1MB Memory Board

Table 6-3. Memory Board (31C03378A) Cuts/Jumpers

Dash Number	ZEBRA Model	Memory Expansion	Cuts	Jumpers
01	1700/1750	1MB	None	None
11	1700/1750	512KB	None	None
21	1700/1750	128KB	JP4 1-2	JP4 3-4
31	700/750	512KB	JP2 1-2, 3-4, 5-6 JP3 1-2, 3-4, 5-6	JP1 1-2, 3-4, 5-6
41	700/750	128KB	JP2 1-2, 3-4, 5-6 JP3 1-2, 3-4, 5-6	JP1 1-2, 3-4, 5-6

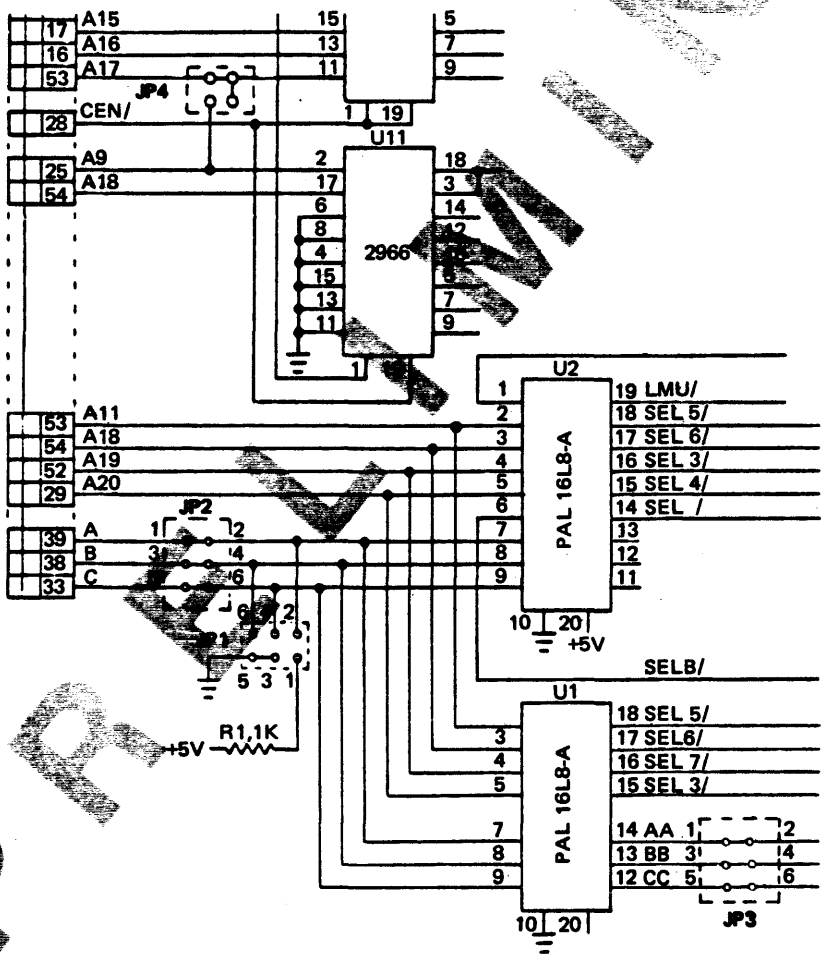


Figure 6-6. 1MB Board, Cuts/Jumpers

6.2 I/O EXPANSION BUS

The I/O Expansion Bus is an extension of the microprocesor bus on the main CPU board. Access to this bus is made via three 60-pin printed circuit board connectors, J1 through J3. The signals on these connectors (see Table 6-4) can be used to add I/O device controllers to the system.

Table 6-4. I/O Expansion Bus, J1 through J3 Pin Assignments

Pin No.	Signal Mnemonic	Pin No.	Signal Mnemonic
1	GND	31	CPUCLK
2	GND	32	A21
3	A01	33	D04
4	A07	34	A22
5	A04	35	BR*
6	A08	36	A23
7	A03	37	FC2
8	A09	38	D15
9	A05	39	BG*
10	A10	40	D14
11	See (1)	41	DTACK*
12	A11	42	D13
13	See (2)	43	WRT*
14	A12	44	D12
15	IPL2*	45	LDS*
16	A13	46	D11
17	BERR*	47	ULS*
18	A14	48	D10
19	A02	49	AS*
20	A15	50	D09
21	A06	51	D00
22	A16	52	D08
23	INT2*	53	D01
24	A17	54	D07
25	IRST*	55	D02
26	A18	56	D06
27	HALT*	57	D03
28	A19	58	D05
29	INT6*	59	-5V
30	A20	60	+5V

(1) J1 & J2 = SINT J3 = IPL0*
(2) J1 & J2 = CGND J3 = IPL1*

6.3 LOCAL AREA NETWORK (LAN) CONTROLLER

The LAN controller is a single board (31C03323A) option with the ZEBRA 1700/1750 series.

Figure 6-7 is a layout of the ZEBRA 1700/1750 LAN printed circuit board showing the location of major components, jumpers, and switches. Pin connections for the LAN connector are identical to those described in Table 6-4. For a complete description of major LAN components (COM9026, COM9032), refer to the Standard Microsystems Corporation literature (see Foreword).

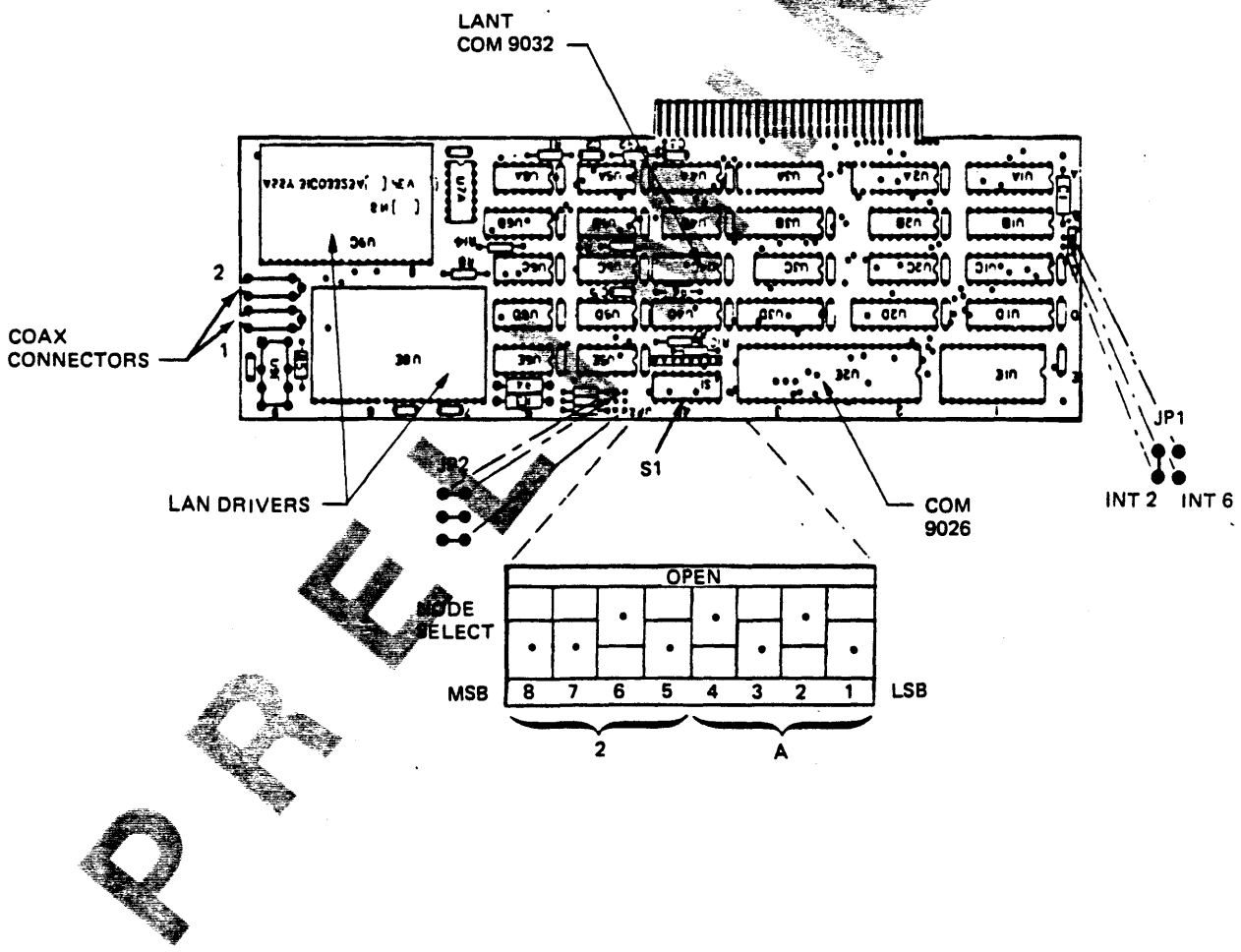


Figure 6-7. ZEBRA 1700/1750 LAN Board

6.3.1 NODE SELECT

When equipped with LAN, the ZEBRA 1700/1750 can communicate with other ZEBRAs equipped with LAN. To do this, each must be a uniquely identified node within the network. This is accomplished by setting the 8-position switch (S1, Figure 2-3) at an assigned number (determined by the user) from 1 to 255. As an example, the enlarged view of S1 shows a node identification setting with switches 1, 3, 5, 7, and 8 in closed position, meaning that the node number is set at 2A hex or 42 decimal. See Table 6-5 for a list of LAN addresses and switch settings.

6.3.2 JUMPERS

JP1 Jumper is installed (as shown in Figure 6-7) to select Interrupt Level 2 INT2*.

JP2 Jumpers are installed (as shown in Figure 6-7) to set at high (+5V) COM9026 extended timeout functions (ET1, ET2) and Echo Diagnostic (ECHO) enable. These jumpers are only removed for diagnostics of the LAN board.

HEX LSB (DIP 1-4)

HEX MSB (DIP 5-8)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0000	0001	0002	0003	0004	0005	0006	0007	0008	0009	0010	0011	0012	0013	0014	0015
1	0016	0017	0018	0019	0020	0021	0022	0023	0024	0025	0026	0027	0028	0029	0030	0031
2	0032	0033	0034	0035	0036	0037	0038	0039	0040	0041	0042	0043	0044	0045	0046	0047
3	0048	0049	0050	0051	0052	0053	0054	0055	0056	0057	0058	0059	0060	0061	0062	0063
4	0064	0065	0066	0067	0068	0069	0070	0071	0072	0073	0074	0075	0076	0077	0078	0079
5	0080	0081	0082	0083	0084	0085	0086	0087	0088	0089	0090	0091	0092	0093	0094	0095
6	0096	0097	0098	0099	0100	0101	0102	0103	0104	0105	0106	0107	0108	0109	0110	0111
7	0112	0113	0114	0115	0116	0117	0118	0119	0120	0121	0122	0123	0124	0125	0126	0127
8	0128	0129	0130	0131	0132	0133	0134	0135	0136	0137	0138	0139	0140	0141	0142	0143
9	0144	0145	0146	0147	0148	0149	0150	0151	0152	0153	0154	0155	0156	0157	0158	0159
A	0160	0161	0162	0163	0164	0165	0166	0167	0168	0169	0170	0171	0172	0173	0174	0175
B	0176	0177	0178	0179	0180	0181	0182	0183	0184	0185	0186	0187	0188	0189	0190	0191
C	0192	0193	0194	0195	0196	0197	0198	0199	0200	0201	0202	0203	0204	0205	0206	0207
D	0208	0209	0210	0211	0212	0213	0214	0215	0216	0217	0218	0219	0220	0221	0222	0223
E	0224	0225	0226	0227	0228	0229	0230	0231	0232	0233	0234	0235	0236	0237	0238	0239
F	0240	0241	0242	0243	0244	0245	0246	0247	0248	0249	0250	0251	0252	0253	0254	0255

DECIMAL VALUES

Example: Hex 2A = 0042 Decimal

7 power requirements

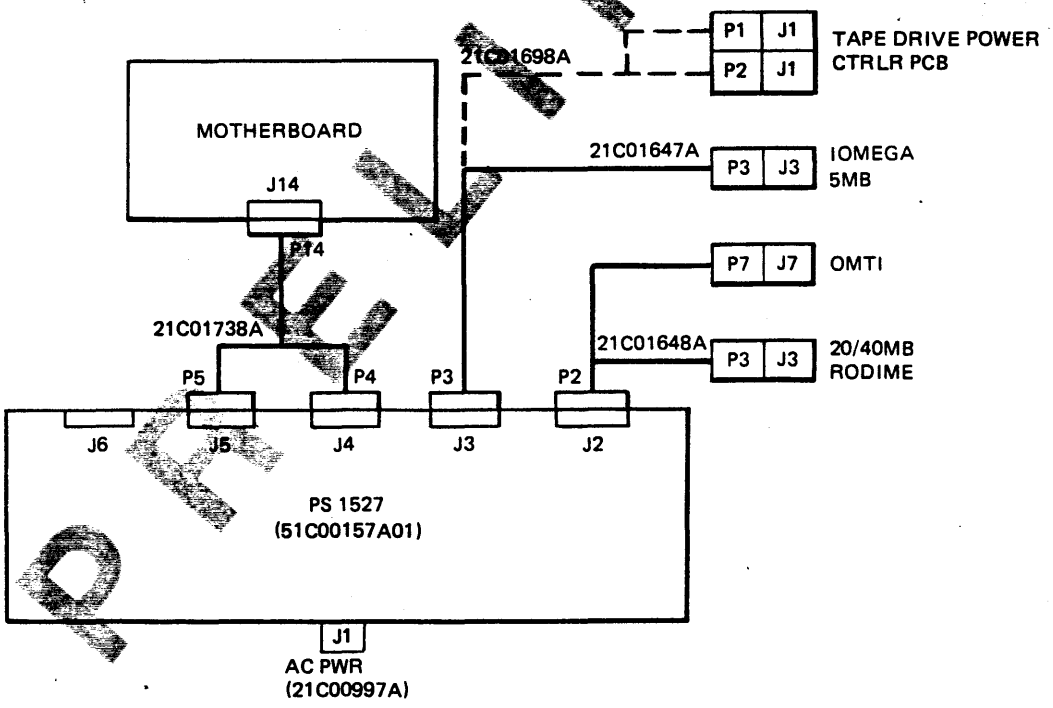
7.1 AC POWER REQUIREMENTS

The ZEBRA 1700/1750 is designed to operate over two AC voltage ranges: 90 VAC to 130 VAC and 180 VAC to 260 VAC. The allowable frequency range is 50Hz to 60Hz. The listing below shows the AC current requirements for these different configurations:

Input Voltage Range	Static AC Current	Peak AC Current
100 VAC to 120 VAC	1.1 Amp	3.0 Amp
220 VAC to 240 VAC	.6 Amp	1.5 Amp

7.2 DC POWER REQUIREMENTS

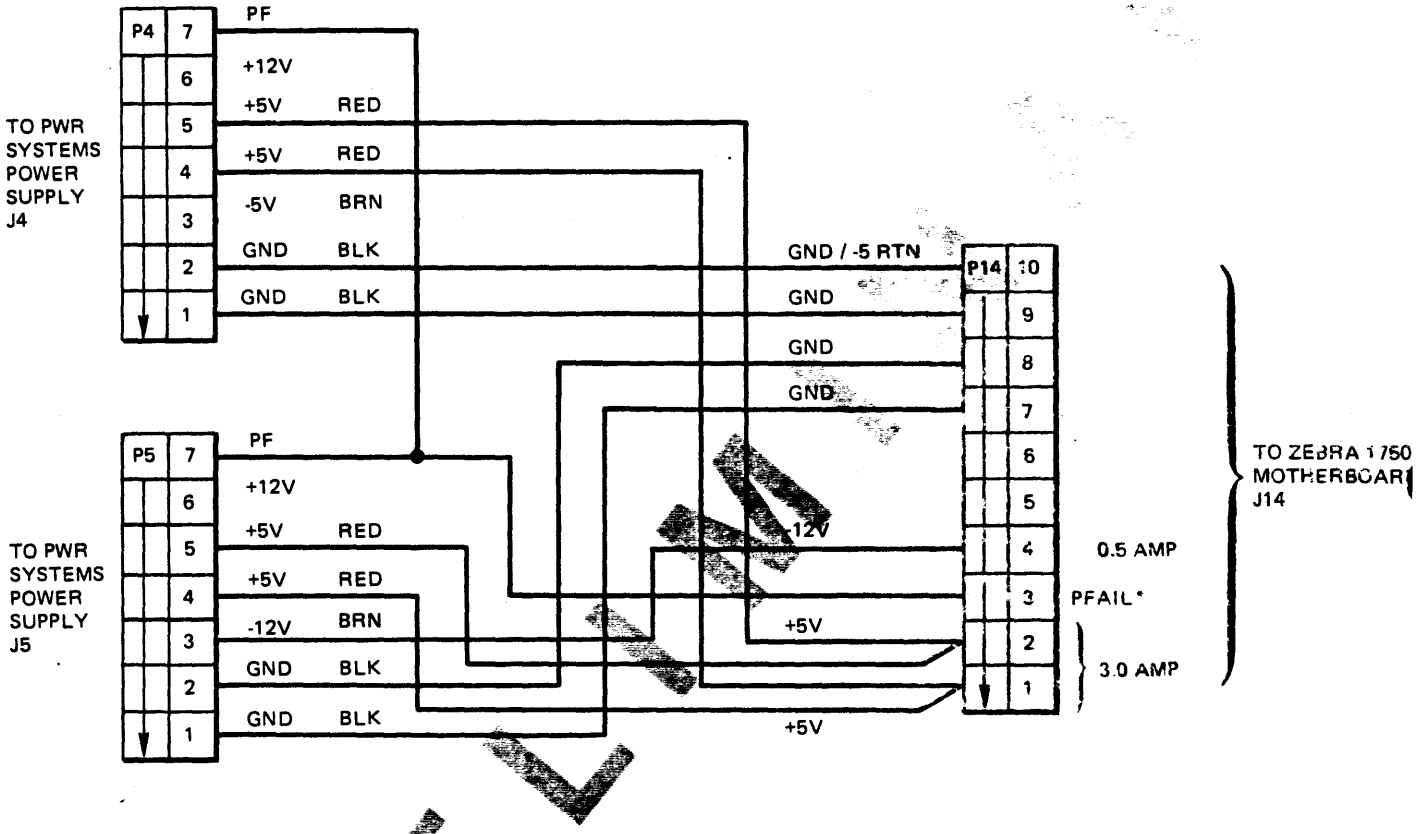
ZEBRA 1700/1750 power is provided by the Power Systems unit, PS1527. This is a switching power supply providing three voltages: +5VDC, -12VDC, and +12VDC. These voltages are distributed by cables to subsystems as shown below:



7.2.1 POWER CABLE CONNECTIONS

Pin assignments for each of the foregoing cables are summarized below.

7.2.1.1 Motherboard Cable (21C01738A)



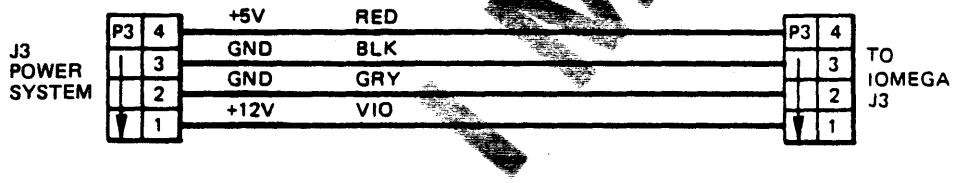
The ZEBRA 1700 MMU board increases the +5VDC current by 1.1 amp to 4.1 amp. Table 7-1 summarizes the DC current requirements for each I/O device of the ZEBRA 1750 PICK System. For the ZEBRA 1700 XENIX system, 1.1 amp should be added to the +5VDC current for the MMU board.

Table 7-1. DC Current Requirement

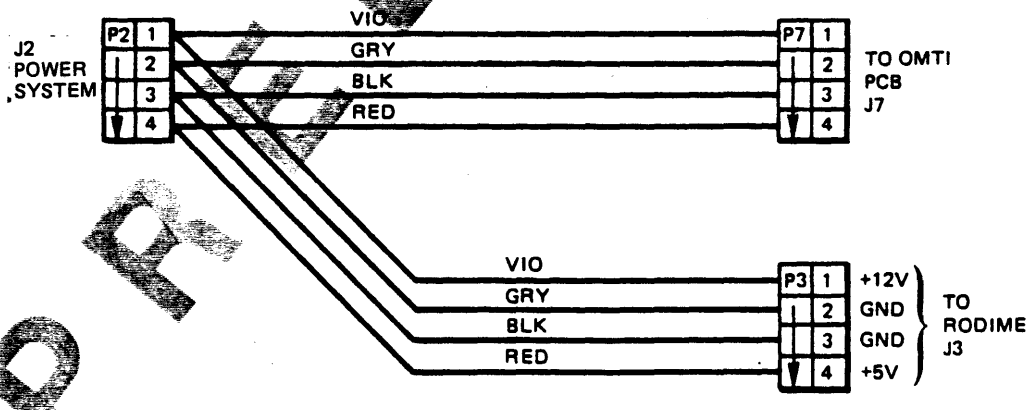
Device	+5VDC	-12VDC	+12VDC
Standard Disk Subsystem:			
Motherboard (1750)	*3.00 Amp	.10 Amp	.00 Amp
OMTI 5100	1.00 Amp	.00 Amp	.00 Amp
RODIME Winchester Drive	.65 Amp	.00 Amp	2.00 Amp
IOMEGA Beta 5	1.70 Amp	.00 Amp	1.00 Amp
Total	6.35 Amp	.10 Amp	3.00 Amp
Optional Disk/Tape Subsystem:			
Motherboard (1750)	*3.00 Amp	.10 Amp	.00 Amp
OMTI 5300	1.50 Amp	.00 Amp	.00 Amp
RODIME Winchester Drive	.65 Amp	.00 Amp	2.00 Amp
1/4" Tape	0.60 Amp	---	1.60 Amp
Total	5.75 Amp	.10 Amp	3.60 Amp

*For ZEBRA 1700, add 1.1 amp.

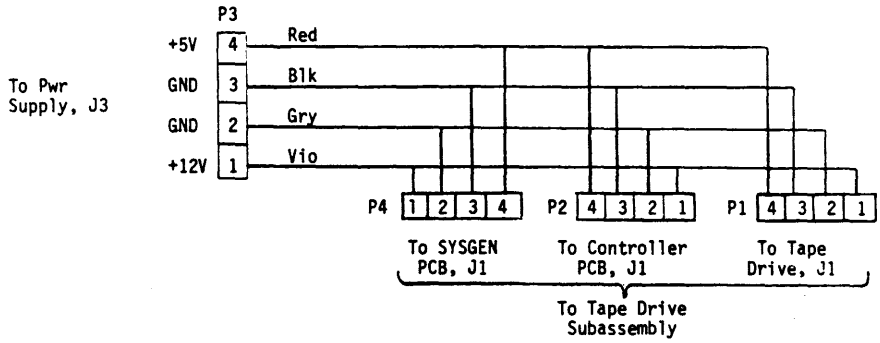
7.2.1.2 IOMEGA Cable (21C01647A)



7.2.1.3 OMTI/RODIME Cable (21C01648A)



7.2.1.4 ARCHIVE 1/4" CT Cable (21C01698A)



7.3 ENVIRONMENTAL CONSIDERATIONS

Storage Temperature:	$-8^{o}F$ to $126^{o}F$
Operating Temperature:	$55^{o}F$ to $85^{o}F$, $70^{o}F$ optimum
Maximum Temperature Gradient:	$25^{o}F$ per hour
Maximum Altitude:	7,000 feet
Relative Humidity:	20% to 80%, noncondensing

Air Cleanliness (for IOMEGA):

Particle Size (micron)	Particles/m 3
<1.0	4 x 10 7
<1.5	4 x 10 6
<5.0	4 x 10 5

7.4 POWER FAIL

The system power supply is designed to detect a power fail. When a power-fail condition exists, the PFAIL* line on the DC power connector on the motherboard (J14) is asserted by the power supply. The state of this signal can be read directly through the printer status register (see Section 2.6.3). When the PFAIL* line goes from the deasserted to the asserted state, the Power Fail Flag is set (this flag can also be examined through the printer status register). Whenever this flag is set, a level 6 auto-vector interrupt of the microprocessor occurs. This flag can be cleared by writing a $0001 (word) to address $200000.

ZEBRA 1700/1750 firmware executive A

A.1 INTRODUCTION

The ZEBRA 1700/1750 Series CPU board is equipped with EPROMs containing the Firmware Executive. The Firmware Executive has a set of commands to aid the user in providing system backup, system testing, and diagnostics.

NOTE

Command descriptions and display examples in the following text are based upon Executive EPROMs 70A01680A01, A02, and Diagnostic EPROMs 70A01681A01, A02.

Section A.2 covers the steps to be taken for startup of ZEBRA. Section A.3 describes all of the Executive commands in an alpha sequence, and Section A.4 provides a description of all subcommands under Diagnostics.

The following basic rules apply to the use of Executive commands:

- Keywords may be entered in upper or lowercase, but will be displayed in uppercase.
- Bracketed portions of command formats indicate optional arguments. In most cases, these arguments may be entered in any order. A qualifier (byte, word, or long word) is indicated with a '.q' which will default to word size. Only long word size is valid for address registers or register-addressed memory.
- Spaces shown must be entered, but the number of spaces does not matter as long as it is at least one.
- Multiple commands may be entered on one line when separated by a semicolon (;). Interactive commands may only appear at the end of a multiple command line.
- The backspace key performs a destructive back-space. The console bell will ring if an attempt is made to backspace past the beginning of input.
- The rubout/delete key performs a destructive backspace to the beginning of the input line.

- Commands will be listed in the order the command processor searches for them.

- In commands referencing a SASI device, the DU (Device/Unit) specification is in the form 'device,unit'. 'Device' is the address of the particular SASI controller and 'unit' is the Logical Unit Number (LUN) of the device attached to that controller. Devices may have a number from 0 to 7 (default 0). The range of LUNs depends upon the controller. If only one number is entered, it is assumed to be the unit number for device zero. The DU specification must follow the device specification (e.g., DISK, CD, CT).

A.2 EXECUTIVE INITIALIZATION

Following POWER ON and RESET, the Executive is initialized and a sign-on message is displayed:

```
GENERAL AUTOMATION EXECUTIVE - P/N XXXX-X
XXX KBYTES RAM

Enter BOOT, BACKUP or RESTORE

Ok, _
```

At this time, the user has the option to boot the system, back up the system (binary), or restore the system via a binary tape. To boot the system, the user types in "BOOT [CR]" following the Ok, prompt. If the operator does not make a keyboard entry within 30 seconds following "Ok," an automatic boot will take place, with a display of:

```
Ok, BOOT [CR]
ZEBRA HARD DISK LOADER

LOADING AND VERIFYING PICK MONITOR

PICK MONITOR LOADED AND VERIFIED

 xxxK MEMORY
   x  COMM LINES

OPTIONS(X,F,B)-
```

Following the successful load of PICK, the operator will be prompted to select an option.

During startup and normal operation, the operator will enter X. This will initialize the system and begin the COLDSTART procedure. Options F and B are provided for use by the system operator. These options are summarized as follows:

Option

F Loads entire system from a FILE-SAVE tape or cartridge disk containing the data and dictionary files. Prompt: "(m)ag tape or (c)artridge" will be given. User should enter "m" or "c" as appropriate and mount tape or cartridge with blocksize of 4000 bytes.

B Will return back to the Ok, prompt where the Utilities and Diagnostics can be selected (see Section 6.0).

The entry of "X" will result in IPL from disk, automatic logon and display:

```
SPOOLER STARTED
LINKING WORKSPACE FOR LINE 0



<<< R80   GENERAL AUTOMATION  REV: 3.0 C/L 6 >>>
<<< HH:MM:SS     ZEBRA          DD MMM YYY    >>>


 THIS IS THE COLD-START PROCEDURE


 NOW CLEANING UP ACC FILE


TAPE ATTACHED BLOCK SIZE: 4000

HH:MM:SS  DD MMM YYYY

TIME = HH:MM:SS    [CR]
DATE = DD MMM YYYY [CR]

 NOW VERIFYING THE SYSTEM

[341] ZEBRA PICK R80 3.0  SYSTEM VERIFIED!!!

< CONNECT TIME= XXX MINS.; CPU= XXX UNITS; LPTR PAGES= 0    >
< LOGGED OFF AT HH:MM:SS           ON DD MMM YYY            >


LOGON TO THE GENERAL AUTOMATION PICK ZEBRA 1750 REV:3.0 C/L 6 AT HH:MM:SS
PLEASE ENTER YOUR ACCOUNT NAME >_
```

(Refer to PICK Operator Guide for detail on subsequent steps.)

Your ZEBRA is now ready for use under the PICK Operating System. For complete guidance in using PICK, refer to the PICK set of manuals delivered with your system.

A.3 SYSTEM EXECUTIVE

The System Executive Firmware contains routines that enable the user to modify, dump, or display memory; modify, dump, or display hardware registers, etc. In addition, the user will have this facility to BACKUP, RESTORE, FORMAT, or RELOAD (BOOT) the physical disk in the event the PICK OS is corrupted or the disk drive had to be replaced. The following sections describe each command available within the System Executive.

A.3.1 ? (OR COMMAND OR HELP) COMMAND

The System Executive Firmware menu is accessed by typing in ?, or Command, or Help at the Ok, prompt, as follows:

```
Ok, ? [CR]

  ? BOOT BACKUP COMMANDS DUMP DIAGNOSTIC ERASE FORMAT HELP LOAD
    RESTORE RESET RETENSION REWIND SAVE
```

The full command names are listed in uppercase from left to right, top to bottom in the order they are searched for. This allows the determination of the minimum abbreviation for a command. The letter C may be used for COMMAND; the letter H for HELP. The short form for other commands will be noted in the following sections with the minimum characters underlined.

The complete menu of commands will be displayed by entry of ?? following Ok, prompt:

```
Ok, ?? [CR]

  ?? BOOT BACKUP BREAKPOINT COMMANDS CONNECT CONSOLE CONTEXT DUMP
     DIAGNOSTIC ERASE FORM FORMAT FILL GOTO HELP HOST INPUT LOAD
     MEMORY MATCH MOVE OUTPUT PAGE PRINTER RESTORE REGISTER RESE RESET
     RETENSION REWIND SAVE STEP SEGMENT SRECORD SYSTEM TRACE
```

A.3.2 BOOT COMMAND

BOOT uses the same format as LOAD (Section A.3.19), but will immediately execute the loaded code upon completion.

A.3.3 BACKUP COMMAND

The BACKUP command provides for image backup of disk to another disk (DISK), 1/4" cartridge tape (CT) or cartridge disk (CD). If removable media is used, the user will be prompted with the following message:

```
Mount Cartridge N (y/n): _
```

N is the cartridge number to mount; this number will initiate at 1 and progress to subsequent numbers (2, 3, etc.) as necessary for the removable media being used. Backup will not exceed the size of the disk, no matter what the FOR count specified is. For either fixed or cartridge disk, the first eight sectors will be skipped, as they contain information particular to that disk. If an error occurs, the error code and logical address will be displayed and the Executive reentered.

A.3.3.1 1/4" Cartridge Tape BACKUP

The BACKUP command format for cartridge tape is as follows:

```
BACKUP [unit] CT [unit] [DU device, lun] [FROM sector] [FOR sectors]
```

The unit has a default value of 0 for the source (disk) and 0 for the destination (tape). The FROM sector address (relative sector number) has a default of 8 (beginning of disk to backup). The FOR sectors is the number of sectors to backup. The default number of sectors is the size of the disk.

A.3.3.2 Fixed or Cartridge Disk BACKUP

The BACKUP command format for fixed disk or cartridge disk is as follows:

```
BACKUP [unit] CD [unit] [DU device, lun] [FROM sector] [TO sector]
[FOR sectors]
BACKUP [unit] DISK [unit] [DU device, lun] [FROM sector] [TO sector]
[FOR sectors]
```

The unit has a default value of 0 for the source (disk) and 0 for the destination (disk or cartridge disk). The FROM sector address (relative sector number) has a default of 8 (beginning of disk to backup). The FOR sectors is the number of sectors to backup. The default number of sectors is the size of the disk.

A.3.4 BREAKPOINT COMMAND

Up to eight conditional breakpoints may be set at any one time. Breakpoints are numbered 0 through 7. Breakpoints remain set until cleared. When continuing program execution from a breakpoint, the breakpoint will be ignored and the first instruction executed, after which the breakpoint is enabled. A breakpoint without a condition will always be recognized except as previously noted. A conditional breakpoint will be recognized only if the condition is met. The condition may be based on a register or memory byte, word, or long-word value and is logically tested according to the condition, which may be:

=	Equals	>	Greater Than
<>	Not Equal	>=	Greater Than or Equal
<	Less Than	&	Bits Set (Value is an 'AND' Mask)
<=	Less Than or Equal	!&	Bits Cleared (Value is an 'XOR/AND' Mask)

Recognized registers are D0-D7, A0-A7, and US. Memory may be specified as either absolute or register addressed (@register). The size of the value for testing may be byte, word, or long word (word default). In the following examples, the pound sign (#) indicates the condition operator. Breakpoints may be displayed or cleared by the following format:

```
BREAKPOINT [number] [CLEAR]
```

If no number is entered, all breakpoints are included. If the CLEAR keyword is not entered, the status of the breakpoint(s) is displayed as follows:

```
number AT address
number AT address REGISTER register#value.q
number AT address MEMORY @register#value.q
number AT address MEMORY address#value.q
number AT address *** OVERWRITTEN ***
```

In the last example, the breakpoint could not be verified because something overwrote the breakpoint instruction at the indicated address. If a breakpoint is not set, nothing is displayed for it. Conditional values are always displayed as long-word size, no matter what the qualifier indicates.

Breakpoints and their conditions may be set with the following format:

```
BREAKPOINT number [AT address]
BREAKPOINT number [AT address] REGISTER register#value[.q]
BREAKPOINT number [AT address] MEMORY @register#value[.q]
BREAKPOINT number [AT address] MEMORY address#value[.q]
```

The condition may be altered without entering the address if the breakpoint is already defined. If an address is entered, the condition, if any, must also be entered. If a breakpoint is set where one already exists, the new breakpoint and condition will replace the old ones.

When a breakpoint is recognized, the registers specified by the FORM command are displayed.

A.3.5 COMMAND (?, HELP) COMMAND

See Section A.3.1 for an explanation of this command.

A.3.6 CONNECT COMMAND

The CONNECT command provides for transparent communication between the console and host ports. An optional termination string may be specified. Termination will occur when the termination string or an enabled break condition is received from the console or the host. The entire termination string is sent to the destination before termination. The command format is as follows:

CONNECT [terminator]

A.3.7 CONSOLE COMMAND

The CONSOLE, PRINTER, and HOST commands all use the same format. The console is assumed to be a terminal from which commands will be issued and to which status is displayed. These commands allow the user to alter the physical port used and its characteristics from those set at initialization time. These commands may be extended to multiple lines by ending a line with the and sign (&). The command format is:

```
CONSOLE [options]
```

If no options are entered, the present options are displayed. An example of the display format follows.

```
Unit 0, 9600 Baud, 8 Bits/Char, No Parity, 1 Stop Bit,
Full Duplex, Break Key Disabled, Control ´C´ Enabled,
Break On Error Disabled, Upper/Lower Case, Hardware
Handshake Disabled, Software Handshake Enabled, Send XOFF
At 192 Characters, Send XON At 128 Characters, Copy Output
To Unit 1
```

Options not entered are not altered. Multiple options are separated by at least one space. The valid options are:

```
BAUD=number (decimal number up to 19200)
BITS=number (5 to 8 bits per character)
BREAK=number (0=Break Key disabled, 1=enabled)
CONTROL=number (0=Control C Break disabled, 1=enabled)
DUPLEX=number (0=full, 1=half)
ERROR=number (0=Break On Error disabled, 1=enabled)
HARD=number (0=Hardware Handshake disabled, 1=enabled)
INPUT=unit (unit to copy input to, negative disables)
OUTPUT=unit (unit to copy output to, negative disables)
PARITY=number (0=none, 1=odd, 2=even)
SOFT=number (0=Software Handshake disabled, 1=enabled)
STOP=number (0=1 bit, 1=1-1/2 bits, 2=2 bits)
UNIT=number (any valid Port number starting from zero)
XON=number (number of input characters for receiver XON)
XOFF=number (number of input characters for receiver XOFF)
```

Setting the baud rate, bits per character, parity, number of stop bits, or hardware handshake resets the unit. Note that input and output cannot be copied from at the same time. The default setting after initialization depends upon the application. Generally, production versions have the break key and break on error disabled as well as hardware handshake; control ´C´ is enabled as well as software handshake. Baud rate will be 9600 baud, 1 stop bit, no parity, eight bits per character and full-duplex mode.

A.3.8 CONTEXT COMMAND

The CONTEXT command allows examination and modification of the Context Register. It is valid only for ZEBRA 1700 systems containing an MMU board. The Context Register is a four-bit register that selects 1 of 16 unique sections of the Segment Map. If no hexadecimal context is entered, the current context is displayed and a new context may be entered or, if nothing is entered (return only), no change is made. If a context is entered when the command is entered, the context is changed but not displayed. The command format is as follows:

CONTEXT [number]

A.3.9 DUMP COMMAND

The DUMP command will display in hex and ASCII format 16 bytes per line. The hex format may be either byte, word, or long word (default word). If the terminating condition is not entered, 16 lines (256 bytes) are displayed followed by a colon (:) each time the return key is entered. Any other character followed by a return will terminate the command. The terminating condition may be either FOR a count or UNTIL an address. The command format is as follows:

DUMP[.q]address
DUMP[.q]address FOR count
DUMP[.q]address UNTIL address

An example of a word display line format follows:

00001000 4455 4D50 2057 4F52 4420 4C49 4E45 0D0C DUMP WORD LINE..

Periods are displayed for the non-printing ASCII codes. The display always consists of 16 bytes of data per line.

A.3.10 DIAGNOSTIC COMMAND

The DIAGNOSTIC command invokes the diagnostics package. Refer to Section A.4, ZEBRA Diagnostics, for menu and descriptions of diagnostics. The command format is as follows:

DIAGNOSTIC

A.3.11 ERASE COMMAND

The ERASE command will erase data from a 1/4" cartridge tape. The tape will then be positioned at BOT. The command format is as follows:

ERASE [CT]

This command is valid only for cartridge tape.

A.3.12 FORM COMMAND

The FORM command specifies the register(s) and/or the register content (value) to be displayed for a TRACE, STEP, or BREAKPOINT. The arguments may be in any order; registers are specified as D0-D7, A0-A7, US, SR, and PC. A value pointed to by a register is obtained by placing an at sign (@) before the register name (i.e., @PC displays the instruction word) and may have a qualifier (i.e., @A0.B). A slash (/) may be used to force a new line. All other characters are displayed without interpretation. If no arguments are entered, the present form is displayed. The arguments must be separated by at least one space. The command format is as follows:

FORM [format]

In the next example, the PC, instruction word, status word, and stack pointer are set for display on the first line and registers D0, A0, and A1 on the second line.

```
Ok, FORM PC  <@PC>  SR  A7/  D0  A0  @A1.B
```

When, for example, a trace is in progress, the following display would occur:

```
TRACE - PC=00021002  <@PC=4E71>  SR=2704  A7=00000F4C
D0=274E310D  A0=00000F52  @A1=E8
```

The default form set by initialization is as follows:

```
PC  @PC  SR
```

A.3.13 FORMAT COMMAND

The FORMAT command is used to configure and format the selected disk drive connected to the OMTI disk controller. The command has two formats, one for new disk types (where MODEL number is not available) and one for known disk types, with MODEL numbers assigned.

1. New Disk Type:

```
FORMAT device [DISK CD] [unit] [WIDTH number] [PERIOD number]
  [MODE number] HEADS number CYLINDERS number [REDUCE number]
  TYPE number SECTORS number [INTERLEAVE number]
```

where:

[unit]	is default 0
[WIDTH]	is step pulse width (microseconds), 0 to FF; default 2
[PERIOD]	is step period (microseconds), 0 to FF; default 1 (50 microseconds)
[MODE]	is step mode from 0 to 2; default 0 (buffer stepping)
HEADS	is number of heads may be 1 to 7F
CYLINDERS	is number of cylinders may be 1 to 7FFF
[REDUCE]	is cylinder number at which to reduce write current, 0 to FF; default 0, no reduced current
TYPE	is manufacturer ID code, 0 to 7
SECTORS	is number of sectors per track, 1 to 7F
INTERLEAVE	must be less than number of sectors; default 2.

2. Known Disk Type:

```
FORMAT device unit MODEL number [modifiers]
```

where:

device unit	is physical drive number (0 or 1)
number	is disk model type (0 to 5)
[modifier]	is a modifier of SECTORS, HEADS or CYLINDERS parameters

Format example:

```
FORMAT DISK DU 1,I MODEL x
Disk Configured, Proceed With Format (y/<n>: y
(R)ecover or (I)nitialize Defect Map (<r>/i): [CR]
(L)ist or (E)dit Defect Map (<l>/e):
Defect At Cylinder:  10 Head 0 HEX = A 0
(E)xit or (R)eformat Disk (e/r): e
```

The disk model numbers have been assigned as follows:

Model	Manufacturer	Sectors	Heads	Cylinders	Capacity
0	IOMEGA BETA-5	52	1	394(7)	5,151,744
1	TOSHIBA	32	10	830(16)	
2	RODIME 202	32	4	320(6)	10,289,152
3	RODIME 202E	32	4	640(12)	20,578,304
4	RODIME 204E	32	8	640(12)	41,156,608

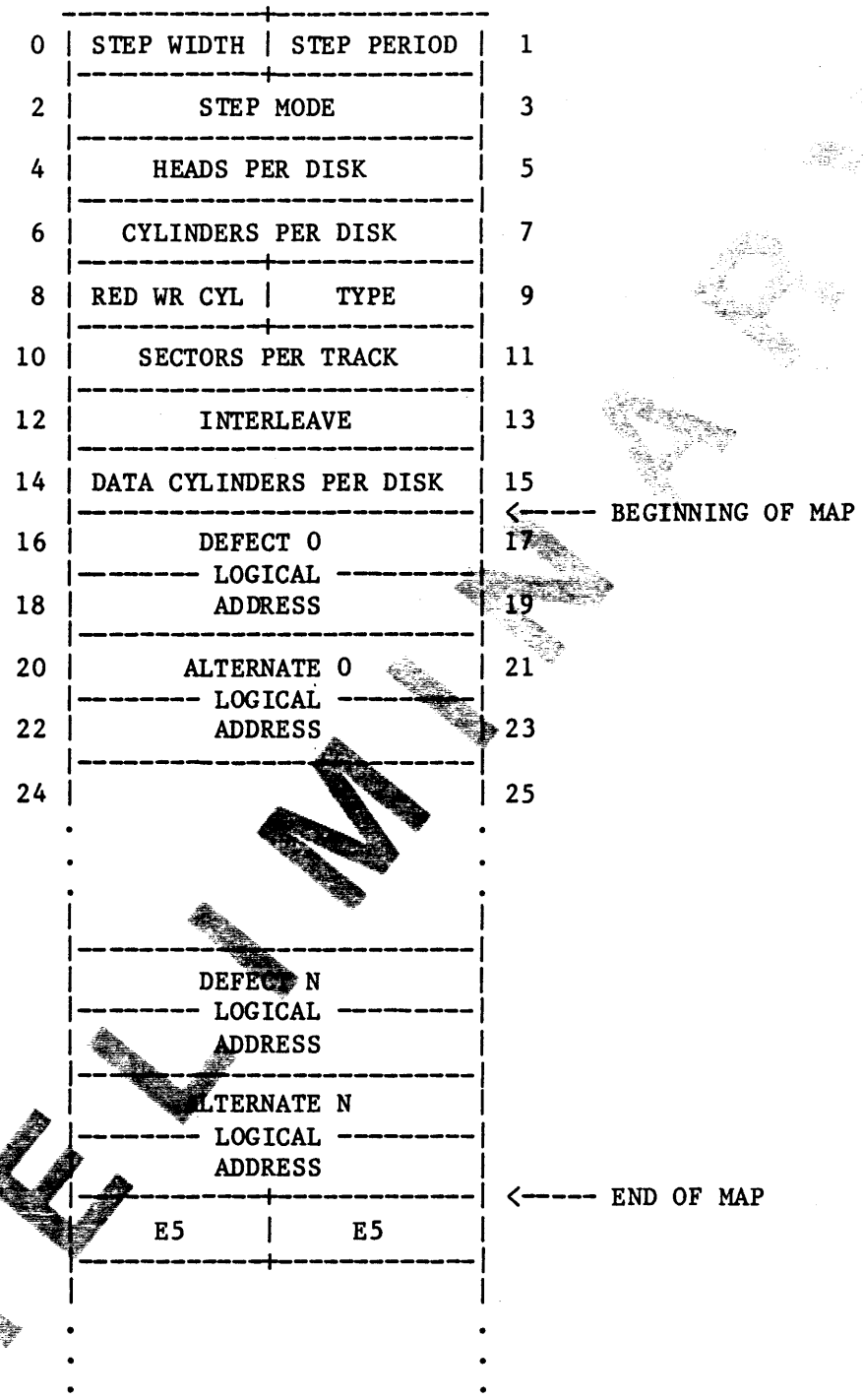
Defective track alternates are mapped to the last cylinders on the disk. The number of reserved cylinders appears in parentheses after the number of cylinders. The maximum number of data cylinders is the total cylinders minus the reserved cylinders. The number of reserved cylinders is 2% of the number of cylinders (rounded down) or 254 tracks, whichever is less.

If the model is specified first, its characteristics may be used for defaults and new values substituted.

During formatting, the defective head and track numbers are displayed. When formatting is completed, the number of defective tracks is displayed and the user is prompted to enter additional tracks from the manufacturer's defect list. The format of the response is defined as follows:

```
Add Defect (Head, Cylinder):  [!] head cylinder
```

The user responds with the head and cylinder number of the additional defective track. If the numbers are preceded with an exclamation point (!), they are taken to be decimal. A null input will terminate the defect list and all defective tracks are then remapped. After mapping, a 2048-byte configuration table/defect map are written into the first sector on track 0, head 0. The format of the disk record is defined on the following page.



A.3.14 FILL COMMAND

The FILL command may be used to fill memory with a byte, word, or long-word value. The default size is word and the default value is zero. The terminating condition may be either FOR a count or UNTIL an address. The command format is as follows:

```
FILL[.q] address [WITH VALUE] FOR count
FILL[.q] address [WITH VALUE] UNTIL address
```

A.3.15 GOTO COMMAND

The GOTO command is used to initiate a user program under Executive control. If the program has not already been run and terminated by the Executive normally, the address must be specified. Otherwise, the state at the time of the program break is restored. After reset, the Executive initializes the user registers to all ones (1's), the supervisor and user stack pointers to the top of local memory, and the status register with 2700 hex. The command format is as follows:

```
GOTO [address]
```

A.3.16 HELP COMMAND

See Section A.3.1 for an explanation of the HELP command.

A.3.17 HOST COMMAND

The HOST command works exactly like the CONSOLE command already described. The host is used for the CONNECT and RECORD commands. The command format is as follows:

```
HOST [options]
```

A.3.18 INPUT COMMAND

The INPUT command is used to read the contents of an address assumed to be an I/O device. The size of the input may be byte, word, or long word. The address entered is biased by the IOBASE value set at assembly time and may be taken to be absolute by preceding the address with an exclamation mark (!). The command format is as follows:

```
INPUT[.q] [!]address
```

This is an interactive command. The address and value of the input register are displayed followed by a colon (:). If only a return is entered, the next register and value are displayed. If a slash (/) is entered, the previous register and value are displayed. If a comma (,) is entered, the same register and value are displayed. If a period (.) is entered, the command terminates. An example of the display format follows.

```
Ok, INPUT.B 0
  001F0000=27:
  001F0001=80: /
  001F0000=FF: ,
  001F0000=27: .
Ok,
```

Note that the IOBASE set at assembly time in the example was 1F0000 (hex).

A.3.19 LOAD COMMAND

The LOAD command is used to read in from disk or cartridge tape data into memory. The command format is as follows:

```
LOAD [DISK] [unit] [DU device, lun] [TO address] [FROM address]
  [FOR sectors]
LOAD CD [unit] [DU device, lun] [TO address] [FROM address] [FOR sectors]
LOAD CT [unit] [DU device, lun] [TO address] [FOR sectors]
```

The default load device is the disk (default unit 0). The default cartridge tape or disk unit is 0. The TO address has a default of 8000 (hex). The address is set in the user's "PC" register so that the start of the loaded memory may be jumped to with a simple GOTO command. The FROM sector address has a default of 8 and is applicable only to disk. The FOR sectors is the number of sectors to load with default of 2.

NOTE: The BOOT command format is identical to the above, with display of "BOOT" rather than "LOAD". BOOT will immediately execute following completion of the LOAD.

A.3.20 MEMORY COMMAND

The MEMORY command is used to display and modify memory data anywhere in the address space in byte, word, or long-word format. If the address is not entered, zero (0) is assumed. The default size is word. The command format is as follows:

```
MEMORY[.q] [address]
```

This is an interactive command. The address and value are displayed and the processor waits for user input. If no value is entered, the location is not changed and the next address and value are displayed. If a slash (/) is entered without a value or after a value, the previous address and value are displayed. If a comma (,) is entered without a value or after a value, the present address and value are redisplayed. If a period (.) is entered without a value or after a value, the command terminates. If an at sign (@) is entered without a value or after a value, the long word at the present address is the next address. A qualifier may follow the at sign. An example of the display format follows:

```
Ok, MEMORY.L 2000
  00002000=001F0800: 001F0000
  00002004=142135F9: /
  00002000=001F0000: ,
  00002000=001F0000: @.B
  001F0000=17: .
Ok,
```

A.3.21 MATCH COMMAND

The MATCH command is used to search for a byte, word, or long word in memory. If the value is preceded by an exclamation mark (!), the search is for values other than the one specified (e.g., mismatch). The search may be FOR a count or UNTIL an address. The command format is as follows:

```
MATCH[.q] [!]value [FROM] address FOR count
MATCH[.q] [!]value [FROM] address UNTIL address
```

Each time a match is found, its address is displayed as shown below.

```
Ok, MATCH.B 20 FROM 2000 UNTIL 2020
  MATCH AT 2012
  MATCH AT 2017
Ok,
```

After 20 matches have been displayed, the user is prompted with a colon (:) and the next 20 matches are displayed each time the return key is pressed.

A.3.22 MOVE COMMAND

The MOVE command is used to move a block of memory from one area to another. The move is either from the start or end of the memory block, depending on any overlap. This means that a move will always move the source data, never moving data it has moved before. The move may be FOR a count or UNTIL a source address. The command format is as follows:

```
MOVE [FROM] address [TO] address FOR count
MOVE [FROM] address [TO] address UNTIL address
```

A.3.23 OUTPUT COMMAND

The OUTPUT command is used to write a value to an address assumed to be an I/O device. The size of the output may be byte, word, or long word. The address entered is biased by the IOBASE value set at assembly time and may be taken to be absolute by preceding the address with an exclamation mark (!). Note that the content is not read first, which would make it useless in some circumstances. The command format is as follows:

```
OUTPUT[.q] [!]address
```

This is an interactive command. The address of the output register is displayed and the processor waits for user input. If no value is entered, the register is not written to and the next register address is displayed. If a slash (/) is entered without a value or after a value, the previous register is displayed. If a comma (,) is entered without a value or after a value, the same register is displayed. If a period (.) is entered without a value or after a value, the command terminates. An example of the display format follows.

```
Ok, OUTPUT.B !1F0000
  001F0000:
  001F0001: 1/
  001F0000: 30,
  001F0000: 20.
Ok,
```

A.3.24 PAGE COMMAND

The PAGE command allows examination and modification of the Page Map. It is valid only for ZEBRA 1700 system with MMU board. If the page number is not part of the command, page zero (0) is assumed. Each page is displayed and the processor then waits for console input. If a value is entered, that value is stored and the next page is displayed. If nothing is entered (return only), no change is made to the data. If a period (.) is entered, the processor is exited. If a period is entered after a value, the processor is exited after the new value is stored. If a slash (/) is entered, the previous page is displayed. If a slash is entered after a value, the previous page is displayed after the value is stored. If a comma (,) is entered, the same page is displayed. If a comma is entered after a value, the page is displayed after the value is entered. An example of the command and its effects follows:

```
Ok, PAGE 3DE
  Page 03F=003F, L=01F800, XX, Multibus Memory, P=01F800:
  Page 040=1000, L=020000, XX, Invalid Memory, P=000000:
Ok,
```

In the above example, the address and contents of the page are displayed. The 'L=' displays the associated logical memory address. Note that this logical address actually corresponds to the segment map physical address. In the first two lines, the 'XX' indicates that the 'Used' and 'Dirty' bits are false, but in the third line they are true (11), indicating that that page has been written to. Next, the type of memory assignment is displayed. Finally, the 'P=' displays the physical address of that memory assignment.

A.3.25 PRINTER COMMAND

The PRINTER command works exactly like the CONSOLE command already described. There are no standard default uses for the printer. The command format is as follows:

```
PRINTER [options]
```

A.3.26 RESTORE COMMAND

The RESTORE command provides for image restore of disk from another disk (DISK), 1/4" cartridge tape (CT) or cartridge disk (CD). If removable media is used, the user will be prompted with the following message:

```
Mount Cartridge N (y/n): _
```

N is the cartridge number to mount; this number will initiate at 1 and progress to subsequent numbers (2, 3, etc.) as necessary for the removable media being used. Restore will not exceed the size of the disk, no matter what the FOR count specified is. For either fixed or cartridge disk, the first eight sectors will be skipped, as they contain information particular to that disk. If an error occurs, the error code and logical address will be displayed and the Executive reentered.

A.3.26.1 1/4" Cartridge Tape RESTORE

The RESTORE command format for cartridge tape is as follows:

```
RESTORE [unit] CT [unit] [DU device, lun] [TO sector] [FOR sectors]
```

The unit has a default value of 0 for the destination (disk) and 0 for the source (tape). The TO sector address (relative sector number) has a default of 8 (beginning of disk to restore). The FOR sectors is the number of sectors to restore. The default number of sectors is the size of the disk.

A.3.26.2 Fixed and Cartridge Disk RESTORE

The RESTORE command format for fixed or cartridge disk is as follows:

```
RESTORE [unit] CD [unit] [DU device, lun] [FROM sector] [TO sector]
  [FOR sectors]
RESTORE [unit] DISK [unit] [DU device, lun] [FROM sector] [TO sector]
  [FOR sectors]
```

The unit has a default value of 0 for the destination (disk) and 0 for the source. The TO sector address (relative sector number) has a default of 8 (beginning of disk to restore). The FOR sectors is the number of sectors to restore. The default number of sectors is the size of the disk.

A.3.27 REGISTER COMMAND

The REGISTER command is used to display or alter the contents of the user registers. If no register is entered, all registers and exception information are displayed and the command terminates. If a register name is entered, that is the first register displayed for modification. The command format is as follows:

REGISTER [register]

This is an interactive command. The register and value are displayed and the processor waits for user input. If no value is entered, the register is not changed and the next register and value are displayed. If a slash (/) is entered without a value or after a value, the previous register and value are displayed. If a period (.) is entered without a value or after a value, the command terminates. If an at sign (@) is entered without a value or after a value, the long-word value in the register is used as the next address in the MEMORY mode. A qualifier may follow the at sign. Exception information cannot be altered. If the last register (PC) or first register (D0) is reached, wraparound is performed. An example of the display format follows:

```
Ok, REGISTER D0
  D0=FFFFFFFF: 0/
  PC=00002000: @.W
  00002000=4E71: .
Ok,
```

The registers displayed when no register name is entered include any special registers as a result of an exception. The registers displayed also depend on the processor type (68000 or 68010). The following is a list of all register mnemonics:

D0-D7	32-Bit Data Registers
A0-A6	32-Bit Address Registers
A7	32-Bit Supervisor Stack Pointer
US	32-Bit User Stack Pointer
SR	16-Bit Status Register
PC	32-Bit Program Counter
VO	16-Bit Vector Offset (68010 only)
SS	16-Bit Special Status Register (Bus/Address Exception only)
AA	32-Bit Access Address(Bus/Address Exception only)
OB	16-Bit Output Buffer (68010 Bus/Address Exception only)
IB	16-Bit Input Buffer (68010 Bus/Address Exception only)
IR	16-Bit Instruction Register (Bus/Address Exception only)

A.3.28 RESET COMMAND

The RESET command reinitializes the Executive. This is different from a hardware reset. The purpose of this command is to reset the serial ports and Executive variables without destroying the contents of memory. The command may not be abbreviated. The command format is as follows:

RESET [DT] [PORTS] [VECTORS]

The DT option resets only the disk/tape (not 1/2" tape) subsystem. The PORTS option resets the Executive serial ports. The VECTORS option reinitializes the exception vector table. If no options are specified, a complete reset is performed.

A.3.29 RETENSION COMMAND

The RETENSION command will retension the 1/4" cartridge tape. The tape will then be positioned at BOT. The command format is as follows:

RETENSION [CT]

This command is valid only for cartridge tape.

A.3.30 REWIND COMMAND

The REWIND command will rewind the 1/4" cartridge tape. The tape will then be positioned at BOT. The command format is:

REWIND [CT]

This command is valid only for cartridge tape.

A.3.31 SAVE COMMAND

The SAVE command is used to write out to disk or cartridge tape data from memory. The command format is as follows:

SAVE [DISK] [unit] [DU device, lun] [FROM address] [TO sector] [FOR sectors]
SAVE CD [unit] [DU device, lun] [FROM address] [TO sector] [FOR sectors]
SAVE CT [unit] [DU device, lun] [FROM address] [FOR sectors]

The default load device is the disk (default unit 0). The default cartridge tape or disk unit is 0. The FROM address has a default of 8000 (hex). The TO sector address has a default of zero and is applicable only to disk. The FOR sectors is the number of sectors to save.

A.3.32 STEP COMMAND

The STEP command is used to initiate a user program under Executive control. If the program has not already been run and terminated by the Executive normally, the address must be specified. Otherwise, the state at the time of the program break is restored. After reset, the Executive initializes the user registers to all ones, the supervisor and user stack pointers to the top of local memory and the status register with 2700 hex. If no termination condition is entered, the program is stepped one instruction at a time and continues when only a return is entered. If any character and a return is entered, the command terminates. Termination conditions may be FOR a count or UNTIL an address. After each instruction is executed, the data specified by the FORM command is displayed. The command format is as follows:

```
STEP [address]
STEP [address] FOR count
STEP [address] UNTIL address
```

A.3.33 SEGMENT COMMAND

The SEGMENT command allows examination and modification of the Segment Map. It is valid only for ZEBRA 1700 system with MMU board. If the segment number is not part of the command, segment zero (0) is assumed. Each segment is displayed and the processor then waits for console input. If a value is entered, that value is stored and the next segment is displayed. If nothing is entered (return only), no change is made to the data. If a period (.) is entered, the processor is exited. If a period is entered after a value, the processor is exited after the new value is stored. If a slash (/) is entered, the previous segment is displayed. If a slash is entered after a value, the previous segment is displayed after the value is stored. If a comma (,) is entered, the same segment is displayed after the value is entered. An example of the command and its effects follows:

```
Ok, SEGMENT 20
  Segment 20=F20, L=100000, S=rwx, U=rwx, P=100000:
  Segment 21=F21, L=108000, S=rwx, U=rwx, P=108000:
  Segment 22=F22, L=110000, S=rwx, U=rwx, P=110000:
Ok,
```

In the above example, the address and contents of the segment are displayed. Note that the address and contents are for the present context. The 'L=' displays the associated logical memory address. The 'S=' and 'U=' display the protection mode for the supervisor and user mode respectively (r = read access, w = write access, and x = execution access). Finally, the 'P=' displays the physical address that drives the page map.

A.3.34 SRECORD COMMAND

The SRECORD command is used to load, via the host unit, standard Motorola 'S' record format absolute load modules. All Motorola record types are recognized. An error will cause an error message to be displayed, but loading will continue until terminated by the end record or a user break. The command format is as follows:

SRECORD [text]

All three address size formats (16, 24, and 32) are recognized. Further, the head record, record count record, and end record cause a message to be printed to the console as shown in the following example:

```
Ok, SRECORD cat main.sav
  Header Record Received:  HDR
  Records Read:  625
  End Record, Bytes Read:  12542, Start Address:  00020000
Ok,
```

Note that the string 'cat main.sav' is sent to the host in order to initiate the transfer. This particular example is typical of a XENIX host. The start address is stored in the user's program counter (PC).

A.3.35 SYSTEM COMMAND

The SYSTEM command works the same as the REGISTER command with no arguments. The register values displayed are those at the last exception when in the Executive mode (user program not running). If there has been no exception, nothing is displayed. Registers may not be altered. This function is mainly used to determine the cause of Executive faults. The command format is as follows:

SYSTEM

A.3.36 TRACE COMMAND

The TRACE command is used to initiate a user program under Executive control. If the program has not already been run and terminated by the Executive normally, the address must be specified. Otherwise, the state at the time of the program break is restored. After reset, the Executive initializes the user registers to all ones, the supervisor and user stack pointers to the top of local memory and the status register with 2700 hex. After each instruction is executed, the data specified by the FORM command is displayed. The command format is as follows:

TRACE [address]

A.4 ZEBRA DIAGNOSTICS

A.4.1 INTRODUCTION

ZEBRA 1700/1750 diagnostic programs are based on a menu-driven system with formatted input/output at the console. The programs are designed to minimize user training by using menus and self-explanatory prompts wherever possible. In addition, default responses are available in almost all cases and are indicated by bracketing them with greater-than and less-than signs (i.e., <default value>). Additionally, valid ranges of responses are given if not self-evident, generally assuming a minimum value of zero or one as appropriate (i.e., <=value). The type of response may be a single character, a string of characters (leading blanks ignored, blank delimiter or carriage return terminator), or a value in either decimal format (16 bits maximum) or hexadecimal format (indicated in the prompt by the dollar sign ($), which is not entered as part of the user response). Format control characters are designed for the Dialogue™ 80 Terminal. Other terminals may be used; however, display formatting for those terminals is not provided.

A.4.2 BASIC ZEBRA REQUIREMENTS

The minimum hardware requirements for the operation of Diagnostics are a working CPU board and a terminal with necessary cables. The terminal must be a Dialogue 80 or equivalent. Diagnostics will not be available if the CPU board is not fully operational. It is also necessary that the on-board Firmware Executive has been run and that the memory maps and device initialization remain as set by the Executive. This minimum system provides the resident features of the package. As the program is still under development, some functions listed in the menus may not be available. The system detects this and notifies the user.

Normal or abnormal termination of Diagnostics will return control to the Executive. Exit from Diagnostics to the Executive is either by 'QUIT'ing or by generating a break with the Dialogue 80 CTRL-BREAK keyboard command.

Diagnostics is entered with the Executive command:

```
DIAGNOSTIC
```

followed by display of

```
GENERAL AUTOMATION DIAGNOSTICS - P/N XXX-X

Formatted Output (<y>/n): y
```

A "y", entered by the operator, indicates a Dialogue terminal or the equivalent. The descriptions of the display operations are for the Dialogue terminal. Other terminals will be spaced down three lines for a clear display and exhibit other minor differences in the display format.

A.4.3 MAIN MENU

Following the clearing of the terminal screen, the main menu is displayed:

```
        SYSTEM DIAGNOSTICS

    1.  QUIT
    2.  CPU TESTS
    3.  CPU DIAGNOSTICS
    4.  SASI TESTS
    5.  SASI DIAGNOSTICS
    6.  SERIAL TESTS
    7.  PARALLEL PRINTER TESTS
    8.  LOCAL AREA NETWORK TESTS
```

The user is then prompted for a program number:

```
    Enter program number (<1>): _
```

Note that the default (in this case, program number one (QUIT)) is indicated by bracketing it with greater-than and less-than signs. This is standard practice for all prompts. All prompts terminate with a colon (:) and user input will be ignored until this character appears. (Note: One character is buffered, but will not be echoed until the prompt is displayed.)

The following sections describe the operation of each function available from the main menu and related submenus.

A.4.3.1 Quit

This function returns the user to the Executive.

A.4.3.2 CPU Tests

This menu item is called with:

Enter program number (<1>): 2 CR

Following clearance of the screen, the CPU Tests submenu is displayed:

CPU TESTS

1. QUIT
2. LOCAL MEMORY
3. SEGMENT MAP
4. PAGE MAP
5. ALL

This submenu provides access to test functions that will continuously test various CPU hardware blocks in order to verify the hardware and test its reliability over an extended period of time.

Quit - This function returns the user to the main menu.

Local Memory - This function first checks the parity circuit (if there is less than one megabyte of memory present), then scans for any unmapped local memory; if unmapped memory is detected, the operator is notified. This function then continuously, destructively, exercises local memory by writing a rotating bit pattern and its complement, in word increments, through a rotating address field throughout all selected local RAM memory, with the exception of areas reserved by the Executive. The amount of memory tested is displayed. The user is given the option of halting on errors and selecting the address range. The test is terminated by entering any character from the keyboard.

Segment Map* - This function continuously, non-destructively, maps two megabytes of logical address space into two megabytes of physical address space for each context proceeding from context 0 to context 15, and within each context from segment map entry 0 to segment map entry 63. The user is given the option of halting on error and looping on failure. The test is terminated by entering any character from the keyboard.

Page Map* - This function continuously, non-destructively, maps all pages as non-accessed, non-used, local memory. The user is given the option of halting on error and looping on failure. The test is terminated by entering any character from the keyboard.

All - This function continuously runs all the above tests in sequence until terminated by any character from the keyboard or by halting on an error if that option has been selected by the user.

*This test is usable only on the ZEBRA 1700.

A.4.3.3 CPU Diagnostics

This menu item is called with:

Enter program number (<1>): 3 CR

Following clearance of the screen, the CPU Diagnostics submenu is displayed:

CPU DIAGNOSTICS

1. QUIT
2. LOCAL MEMORY
3. SEGMENT MAP
4. PAGE MAP
5. ALL

This submenu provides access to diagnostic functions that will test various CPU hardware blocks for the purpose of troubleshooting.

Quit - This function returns the user to the main menu.

Local Memory - This function first checks the parity circuit (if there is less than one megabyte of memory present), then scans for any unmapped local memory. If unmapped memory is detected, the operator is notified. This function then, destructively, verifies local memory by writing a rotating bit pattern and its complement, in word increments, through a rotating address field throughout all selected local RAM memory, with the exception of areas reserved by the Executive. The user is given the option of selecting the address range. The amount of memory tested is displayed upon completion of the diagnostic.

Segment Map* - This function non-destructively verifies segment mapping capability by mapping two megabytes of logical address space into two megabytes of physical address space for each context proceeding from context 0 to context 15, and within each context from segment map entry 0 to segment map entry 63. The user is given the option of looping on failure.

Page Map* - This function performs a rudimentary non-destructive verification of the page map by mapping all pages as non-accessed, non-used, local memory. The user is given the option of looping on failure.

All - This function performs all the above diagnostic functions in sequence.

*This test is usable only on the ZEBRA 1700.

A.4.3.4 SASI Tests

This submenu provides access to test functions that will continuously test various SASI devices to verify the hardware and test its reliability over an extended period of time. This menu item is called with:

```
Enter program number (<1>):  4 CR
```

This is followed by display of:

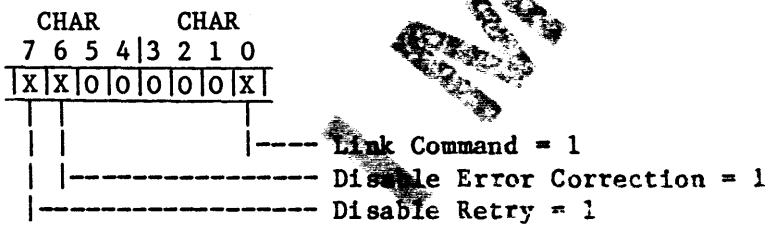
```
Enter Device Number (<0>-7):
Enter Logical Unit Number (<0>-3):
Enter Control Field Byte (<$00>):
Halt on Error (y/<n>):
```

where:

Device Number is the physical unit number (0 through 7; 0 default) of the SASI controller. Device Number 7 is reserved for the SYSGEN tape controller.

Logical Unit Number (LUN) specifies a unit (0 through 3; 0 default) controlled by the physical unit specified by Device Number.

Control Field Byte - This byte specifies three test conditions, or any combination of these three conditions. Bits and characters within the Control Field byte are assigned as follows:



Disable Retry ($80) - Should a valid DATA error occur with the RETRY bit of the CONTROL BYTE enabled, the controller will attempt to read the sector up to four times. If this attempt is unsuccessful, the controller will recalibrate the drive and try to read again. After four unsuccessful "recalibrate/read" sequences, the controller will report the error to the host.

Disable Error Correction ($40) - Should a CORRECTABLE DATA ERROR occur with the ERROR CORRECTION bit enabled, the controller will correct the data prior to transferring the data to the host.

Link Command ($01) - If the LINK COMMAND bit of the CONTROL BYTE is enabled and the previous command was executed without an error, the controller will request the next command.

Combinations of the foregoing test conditions can be requested by entry of hexadecimal numbers $00 (default, no test), $C0, $C1, $41, $81.

Any of the parameters described above can be modified with submenu item 8, Change Parameters. This submenu is displayed following the entry of parameters:

SASI TESTS

1. QUIT
2. OMTI CONTROLLER
3. DISK SEEK/READ
4. DISK WRITE/READ
5. TAPE WRITE/READ
6. TAPE WRITE
7. TAPE READ
8. CHANGE PARAMETERS

Quit - This function returns the user to the main menu.

OMTI Controller - This function first commands the OMTI controller to perform its own RAM diagnostic and then proceeds to perform a loop of writing an incrementing pattern to the buffer memory and then reading it back and comparing it.

Disk Seek/Read - This function recalibrates the disk and then performs a random read on all logical addresses selected.

Disk Write/Read - This function prompts the user for the starting logical address and sector count. The default starts at 8 for 248 sectors, which is a reserved area on disk. Each sector is filled with its logical address and then read for confirmation, sequentially.

Tape Write/Read - This function erases the tape, writes a block with its block number, writes a filemark, rewinds, spaces to the block it wrote, reads the block, spaces to the end of tape (append) and then continues with the block and filemark writes, etc.

Tape Write - This function prompts the user for the tape block count (default all of tape), erases the tape and writes tape with the specified byte pattern (D5 hex default).

Tape Read - This function prompts the user for the tape block count (default all of tape), rewinds the tape and reads and compares the tape with the specified byte pattern (D5 hex default).

Change Parameters - To be determined.

If an error occurs during OMTI test, the appropriate status is displayed and the user is prompted to enter a return before returning the appropriate menu, unless loop on error or no halt on error has been set. Looping or testing ceases when a key is pressed. Errors that can be sensed are: SASI returned status (sense status), SASI bus error (host interface timeout) or a SASI phase error which displays the expected and received phase state. All messages are ASCII text with appropriate hex values.

A.4.3.5 SASI Diagnostics

This submenu provides access to diagnostic functions that will test various SASI devices for the purpose of troubleshooting. This menu item is called with:

```
Enter program number (<1>):  5 CR
```

This is followed by display of: (see Section A.4.3.4 for an explanation of these entries)

```
Enter Device Number (<0>-7):
Enter Logical Unit Number (<0>-3):
Enter Control Field Byte (<$00>):
Halt on Error (y/<n>):
```

Following entry of the parameters, the SASI Diagnostics submenu is displayed:

```
        SASI DIAGNOSTICS

 1.  QUIT                          13.  COPY COMMAND (TAPE)
 2.  SASI COMMAND                  14.  ERASE TAPE CARTRIDGE
 3.  SENSE STATUS                  15.  COPY COMMAND (DISK)
 4.  RECALIBRATE                   16.  SCAN DATA COMMANDS
 5.  REWIND                        17.  ASSIGN DISK PARAMETERS
 6.  REQUEST SENSE                 18.  RAM DIAGNOSTICS
 7.  READ DATA                     19.  WRITE ECC
 8.  SET DISK BLOCK SIZE           20.  READ IDENTIFIER
 9.  WRITE DATA                    21.  REQUEST LOGOUT
10.  SEEK                          22.  READ DATA BUFFER
11.  WRITE FILEMARK                23.  WRITE DATA BUFFER
12.  SPACE (TAPE POSITIONING)      24.  CHANGE PARAMETERS
```

Quit - This function returns the user to the main menu.

If an error occurs during program execution, the appropriate status is displayed and the user is prompted to enter a return before returning to the appropriate menu, unless an error has been set. Looping ceases when a key is pressed. Errors that can be sensed are: SASI returned status (sense status), SASI bus error (host interface timeout) or a SASI phase error which displays the expected and received phase state. All messages are ASCII test with appropriate hex values.

SASI Command - This function allows the issuance of any command to the SASI device. It will request the parameter block bytes from the user, with a zero default, for the appropriate number of bytes depending upon the command class. The lun and control fields are automatically set. If a data-out phase is detected, the user will be asked to supply the required amount of data, in the same fashion as the WRITE DATA command used in the ADES package. If a data-in phase is detected, the date will be displayed, in the same fashion as the READ DATA command used in the ADES package.

Sense Status - This function senses status and displays it to the user.

Recalibrate - This function recalibrates the drive. (Disk only.)

Rewind - This function rewinds the tape. (Tape only.)

Request Sense - This function requests sense status and displays it to the user.

Read Data - This function prompts the user for the starting logical address and the sector/block count. The data read is displayed to the user.

Set Disk Block Size - This function prompts the user for the disk block (sector) size.

Write Data - This function prompts the user for the starting logical address, the sector/block count and the data to be written.

Seek - This function prompts the user for the seek logical address. (Disk only.)

Write Filemark - This function writes a filemark on the tape. (Tape only.)

Space (Tape Positioning) - This function prompts the user for the space type (block, filemark, append). (Tape only.)

Copy Command (TAPE) - This function prompts the user for the function (backup, restore), the disk device number and logical unit number, the number of disk sectors and the disk logical address. The tape is assumed to be at device 7, logical unit number 0.

Erase Tape Cartridge - This function erases the tape cartridge.

Copy Command (DISK) - This function prompts the user for the source logical address, the sector count, the destination logical unit number and destination logical address. The source logical unit number is the one specified when the diagnostics menu was entered.

Scan Data Commands - This function prompts the user for the scan type (equal, high or equal, low or equal), the logical address and sector count. (Disk only.)

Assign Disk Parameters - This function prompts the user for the step pulse width, step period, step mode, number of heads, cylinder address, reduced write current cylinder, device type and sectors per track. (Disk only.)

RAM Diagnostics - This function causes the SASI device to perform the pattern test on the sector buffer. (Disk only.)

Write ECC - This function prompts the user for the logical address, the data to be written and the four ECC bytes. (Disk only.)

Read Identifier - This function displays the user for the logical address and displays the cylinder number, head number, flags and sector number. (Disk only.)

Request Logout - This function displays the last retry and permanent error counts. (Disk only.)

Read Data Buffer - This function displays the devices data buffer. (Disk only.)

Write Data Buffer - This function prompts the user for data to be written to the devices data buffer. (Disk only.)

Change Parameters - To be determined.

A.4.3.6 Serial Tests

This submenu provides access to test functions that will continuously test a selected serial port to verify the hardware and test its reliability over an extended period of time. This menu is called with:

Enter program number (<1>): 6 CR

Following clearance of the screen, the Serial Tests submenu is displayed:

SERIAL TESTS

1. QUIT
2. ECHO TEST
3. CONTINUOUS TRANSMIT

Quit - This function returns the user to the main menu.

Echo Test - In this function, a character received by the test port is retransmitted to both the test and the console ports. Error status is reported to the console. The test is terminated upon receipt of an ESC (hex 1B) character.

At startup, the operator is asked for the port to test, baud rate, character length, number of stop bits, and the types of parity (none, odd, or even). This test is status driven.

Continuous Transmit - This function transmits the entire printable ASCII character set to the test port. The test is terminated by pressing any key on the console port.

At startup, the operator is asked for the port to test, baud rate, character length, number of stop bits, and the type of parity (none, odd, or even). This test is interrupt driven.

A.4.3.7 Parallel Printer Tests

This submenu provides access to a test function that will continuously exercise the parallel printer port, in order to verify the hardware and test its reliability over an extended period of time. This menu item is called with:

Enter program number (<1>): 7 CR

Following clearance of the screen, the Parallel Printer Tests submenu is displayed:

PARALLEL PRINTER TESTS

1. QUIT
2. CONTINUOUS PRINT

Quit - This function returns the user to the main menu.

Continuous Print - This function continuously transmits the entire printable ASCII character set parallel printer port. The test is terminated by pressing any key on the console keyboard.

A.4.3.8 Local Area Network Tests

This submenu provides access to a test function that will continuously test the LAN in order to verify the hardware and test its reliability over an extended period of time. This menu item is called with:

Enter program number (<1>): 8 CR

Following clearance of the screen, the Local Area Network Tests submenu is displayed:

LOCAL AREA NETWORK TESTS

1. QUIT
2. RAM TEST

Quit - This function returns the user to the main menu.

RAM Test - This function writes and read/compares the LAN memory. The test is terminated by pressing any key on the console keyboard.

A.4.4 EXTENDED DIAGNOSTICS

Certain diagnostic testing of ZEBRA 1700/1750 can now be carried out in an off-line manner from disk, cartridge disk (CD), or cartridge tape (CT). The loading for each of these is accomplished with:

- Boot from A to 8200
- Boot CD
- Boot CT

Following the selected boot operation, the following display will take place.

```
    GENERAL AUTOMATION EXTENDED DIAGNOSTICS - REV. 1.1

    Loaded - Enter "Extend" to Execute


        EXTENDED DIAGNOSTICS

    1.  QUIT
    2.  SERIAL TESTS
    3.  PARALLEL PRINTER TESTS
    4.  LOCAL AREA NETWORK TESTS

Enter program number (<1>):

    Hit ESC Key to Terminate Test
    Hit Any Key to Stop
    Enter Port to Test:
    Enter Desired Baud Rate (<9600>):
    Enter Character Length (<8>):
    Enter Number of Stop Bits (<2>):
    Parity (<n>):
    Odd or <Even>:


        SERIAL PORTS TESTS

    1.  QUIT
    2.  ECHO TEST
    3.  CONTINUOUS TRANSMIT
```

```
    PARALLEL PRINTER PORT TEST

Hit Any Key to Stop
Enter Number of Columns (<80>):
Enter Number of Lines (<55>):

1.  QUIT
2.  CONTINUOUS PRINT


    LOCAL AREA NETWORK TESTS

1.  QUIT
2.  RAM TEST
```

For a description of each menu item, refer to the previous text (Sections A.4.3.6, A.4.3.7, and A.4.3.8) covering each test.