iSBC 544
INTELLIGENT COMMUNICATIONS CONTROLLER BOARD HARDWARE REFERENCE MANUAL

Manual Order Number: 9800616A

Intel Corporation, 3065 Bowers Avenue, Santa Clara, California 95051

Printed in U.S.A./B79/0878/3.5K WHM

PREFACE

This manual provides general information, preparation for use, programming information, principles of operation, and service information for the iSBC 544 Intelligent Communications Controller Board. Supplementary information is provided in the following documents:

- Intel MULTIBUS Interfacing, Application Note AP-28.
- Intel MCS-85 User's Manual, Order No. 9800366.
- Intel 8080 Microcomputer Peripherals User's Manual, Order No. 98-364.
- Intel 8080/8085 Assembly Language Programming Manual, Order 98-301.

CONTENTS

intel

TABLES

ILLUSTRATIONS

CHAPTER 1
GENERAL INFORMATION

1-1. INTRODUCTION

The iSBC 544 Intelligent Communications Controller is a member of a complete line of Intel iSBC 80 system components. The iSBC 544 operates as an intelligent slave on the system, providing an expansion of system serial communications capability, including four fully programmable synchronous and asynchronous serial I/O channels with RS232C buffering. As an intelligent slave, the iSBC 544 employs its own 8085A CPU to handle all on-board processing. Baud rates, data formats, and interrupt priorities are individually software selectable for each channel. The iSBC 544 also includes 10 lines of buffered parallel I/O interface which provides compatibility with a Bell 801 Automatic Calling Unit.

The iSBC 544 is a self-contained communications processor that incorporates an 8085A CPU (for on board processing only), up to 16K bytes of dynamic RAM, 8K bytes of PROM, and the aforementioned I/O interface. The intelligent slave concept allows the iSBC 544 to unburden a system CPU by performing all communications related peripheral tasks without constant interruption of the CPU. This concept allows for maximum I/O throughput on the system, with a minimum amount of impact on the system bus. The iSBC 544 accepts commands from the master CPU, performs the necessary functions to complete the peripheral operation, interrupts the master CPU on completion and allows the transfer of data into or out of on-board memory.

The iSBC 544 is also capable of operating as a single board communications computer. In this mode, it can control a number of iSBC 534 Communication Expansion boards or other memory and I/O expansion boards. A list of the bus limitations of the iSBC 544 used in this mode can be found in Chapter 3.

1-2. DESCRIPTION

The iSBC 544 (figure 1-1) is designed to be plugged into a standard iSBC 604/614 Modular Backplane and Cardcage to interface directly with an Intel iSBC Single Board Computer or used with an Intel Microcomputer Development System. The iSBC 544 provides four serial I/O ports, one parallel I/O port, seven programmable timers, and eight interrupt inputs with programmable priority. Also provided is a dedicated on-board processor consisting of an 8085A CPU and memory.

1-3. SERIAL I/O PORTS

Each of the four serial I/O ports is fully RS232C plug compatible and is controlled and interfaced by an Intel 8251A USART (Universal Synchronous/Asynchronous Receiver/Transmitter) chip. Each USART is individually programmable for operation in most synchronous or asynchronous serial data transmission formats (including IBM Bi-Sync).

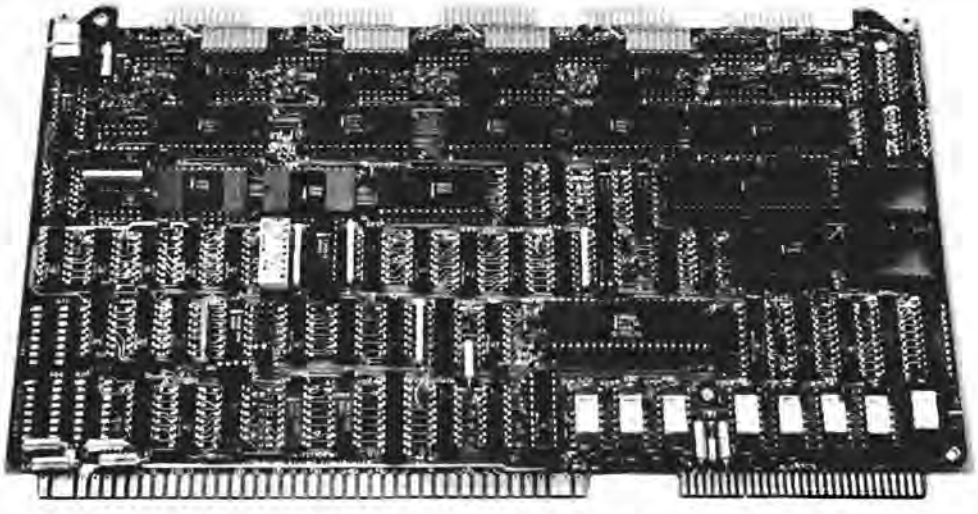


Figure 1-1. iSBC 544 Intelligent Communications Controller Board

In the synchronous mode the following are programmable:

a. Character length,

b. Sync character (or characters), and

c. Parity.

In the asynchronous mode the following are programmable:

a. Character length,

b. Baud rate factor (clock divide ratios of 1, I6, or 64),

c. Number of Stop bits, and

d. Parity.

In both the synchronous and asynchronous modes, each serial I/O port features half- or full-duplex, double-buffered transmit and receive capability. In addition, USART error detection circuits can check for parity, overrun, and framing errors. The USART transmit and receive clock rates are separately derived from one of five independently programmable Baud rate/time generators.

1-4. PARALLEL I/O PORT

The parallel I/O port has 10 buffered I/O lines controlled by an Intel 8155 Programmable Interface (PPI) chip. The parallel I/O port is directly compatible with an Automatic Calling Unit (ACU) such as the Bell Model 801, or equivalent, and can also be used for auxiliary functions. All signals are RS232C compatible, and the interface cable signal assignments meet RS366 specifications.

If the system application does not require an interface to an ACU, the parallel I/O port can be used for any general purpose or auxiliary parallel interface that is RS232C compatible.

1-5. PROGRAMMABLE TIMERS

One of the primary features of the iSBC 544 is flexible clock programming. The iSBC 544 has two Intel 8253 Programmable Interval Timer (PIT) chips that provide a total of six separate time/rate generators. All six are independently software-programmable, and can generate different Baud rate clock signals for each USART chip.

Four of the timers (BDG0-BDG3) are used as Baud rate generators; the fifth timer can be used as an auxiliary transmit or receive clock, and the sixth timer can be used to generate an interrupt.

In addition to the timers on the 8253 PITs, the iSBC 544 has a 14-bit timer located on the 8155 PPI which can be used for miscellaneous functions.

1-6. INTERRUPT FUNCTIONS

The iSBC 544 has the following interrupt sources:

a. Eight serial I/O interrupts serviced by an 8259 Programmable Interrupt Controller (PIC).

b. Flag Interrupt

c. Carrier Detect and Ring Indicator Interrupts

d. Multibus interrupts

e. Timer Interrupts

The 8259 PIC has eight input interrupt request lines. The PIC treats each true input signal condition as an interrupt request. After resolving the interrupt priority, the PIC issues a single interrupt request to the on-board 8085A CPU. The interrupt priorities of the PIC chip are independently programmable under software control. The programmable interrupt priority modes are:

a. Fully Nested Priority. Each interrupt request has a fixed priority: input 0 is highest, input 7 is lowest.

b. Auto-Rotating Priority. Each interrupt request has equal priority. Each level, after receiving service, becomes the lowest priority level until the next interrupt occurs.

c. Specific Priority. Software assigns lowest priority. Priority of all other levels is in numerical sequence based on lowest priority.

The Flag Interrupt allows any bus master to interrupt the iSBC 544 by writing into the base address of RAM memory. The flag interrupt is cleared when the on-board processor reads the base address. This flag provides a unique interrupt to each iSBC 544 in the system.

The Carrier Detect and Ring Indicator Interrupts allow the iSBC 544 to monitor the serial I/O ports and detect the loss of a carrier signal or the ringing of a telephone line respectively. These interrupts are detected by the on-board 8085A CPU, and available for interrogation through the 8155 PPI.

The iSBC 544 can generate an interrupt on the Multibus, and also receive an interrupt from the bus.

1-7. 8085A CPU

The 8085A CPU, which is the heart of the iSBC 544, performs on-board processing functions and generates the addresses and control signals required to access memory and I/O devices. The 8085A contains six 8-bit general purpose registers and an accumulator. The six general purpose registers may be addressed individually or in pairs, providing both single and double precision operations.

The 8085A has 5 prioritized interrupt inputs (TRAP, RST 7.5, RST 6.5, RST 5.5, and INTR) which generate unique memory addresses for interrupt handling routines. All interrupt inputs with the exception of TRAP may be masked via software.

1-8. PROM CONFIGURATION

The ROM/PROM on the iSBC 544 consists of either 4K or 8K bytes. Two sockets are provided for user installation of the PROM chips. Jumpers are provided for accommodation of different types of chips (Intel 2716 PROMs and 2316, or 2332 ROMs). Address block 0000-1FFF is reserved for PROM use only.

1-9. RAM CONFIGURATION

The iSBC 544 includes 16K of dynamic RAM implemented with eight Intel 2117 chips and an Intel 8202 Dynamic RAM Controller. Dual-port control logic allows the RAM to be accessed by either the on-board 8085A, or by another bus master. The RAM decode logic allows for extended Multibus addressing of up to 20 address lines. This allows bus masters with 20-bit addressing capability to partition the iSBC 544 into 16K segments in a 1-mega-byte address space. The on-board 8085A CPU, however, has only a 16-bit address capability, and limits the on-board RAM to a 64K address space.

The iSBC 544 also has 256 bytes of static RAM located on the Intel 8155 PPI. This memory is only accessible to the on-board 8085A CPU. The address block for the static RAM is 7F00-7FFF.

1-10. EQUIPMENT SUPPLIED

The following are supplied with the iSBC 544 Intelligent Communications Controller:

a. Schematic Drawing, dwg. no. 2001695

b. Assembly Drawing, dwg. no. 1001693

1-11. SPECIFICATIONS

Specifications for the iSBC 544 Intelligent Communications Controller are provided in table 1-1.

Table 1-1. Specifications

8085A CPU	
WORD SIZE	
Instruction:	8, 16, or 24 bits
Data:	8 bits
CYCLE TIME:	1.45 μsec ± 0.1% for fastest executable instruction; i.e., four clock cycles.
MEMORY CAPACITY	
On-Board ROM/PROM	4K or 8K of user installed ROM/PROM.
On-Board RAM (Dynamic)	16K of RAM. Integrity maintained during power failure with user-furnished batteries. (optional)
On-Board RAM (Static)	256 bytes of RAM:
MEMORY ADDRESSING:	
On-Board ROM/PROM	0000-1FFF
On-Board Static RAM	7F00-7FFF
On-Board RAM	16K: 8000-BFFF
On-Board RAM (System Access)	Jumpers allow board to act as slave memory for other bus masters. 16 or 20 bit addressing can be accommodated. Boundaries may be set on any 4K increment 00000-FF000, which is switch selectable. 4K, 8K, or 16K can be made available to the bus by switch selection.
CPU RAM ACCESS TIME:	450 nsec min, and 1100 nsec + off-board command duration max. CPU has priority over bus master access, however CPU request cannot abort bus master access in progress.

Table 1-1. Specifications (Cont'd.)

MINIMUM WAIT STATE CONSIDERATIONS

Type	On-Board Request		Off-Board Request (Master Mode)	
	Normal	Refresh	Normal	Refresh
I/O Read/Write	None	N/A	N/A	N/A
Memory Write (Dyn.)	1	2	2	3
Memory Read (Dyn.)	None	1	1	2
Memory Write (Stat.)	None	None	N/A	N/A
Memory Read (Stat.)	None	None	N/A	N/A

Note: Dyn.=Dynamic RAM
Stat.=Static RAM or PROM

INTERRUPTS:

8085A CPU includes five interrupt inputs, each of which vectors the processor to the following memory location for entry point to service routine:

Interrupt Input	Vector Address	Priority	Type
TRAP	24	1	Non-Maskable
RST 7.5	3C	2	Maskable
RST 6.5	34	3	Maskable
RST 5.5	2C	4	Maskable
INTR	Note	5	Maskable

Note: INTR input provided by 8259 PIC. See table 3-27 for vector addresses.

SERIAL COMMUNICATIONS

Synchronous: 5-, 6-, 7-, or 8-bit characters. Internal; 1 or 2 sync characters. Automatic sync insertion, parity and overrun error detection.

Asynchronous: 5-, 6-, 7-, or 8-bit characters. Break character generation and detection 1, 1½, or 2 stop bits. False start bit detection, parity, overrun and framing error detection.

Sample Baud Rate:

Frequency[1] (kHz, Software Selectable)	Baud Rate (Hz)[2]		
	Synchronous	Asynchronous	
		÷16	÷64
153.6	—	9600	2400
76.8	—	4800	1200
38.4	38400	2400	600
19.2	19200	1200	300
9.6	9600	600	150
4.8	4800	300	75
6.98	6980	—	110

Notes:

1. Frequency selected by I/O writes of appropriate 16-bit frequency factor to counter/timer Register.
2. Baud rates shown here are only a sample subset of possible software-programmable rates available. Any frequency from 18.75 Hz to 614.4 kHz may be generated utilizing on-board crystal oscillator and 16-bit Programmable Interval Timer (used here as frequency divider).

INTERVAL TIMER AND BAUD RATE GENERATOR

Input Frequency: On board 1.2288 MHz .1% crystal; 0.814 microsecond period, nominal or 1.8432 MHz ±.1% crystal; 0.542 microsecond period, nominal.

Table 1-1. Specifications (Cont'd.)

Output Frequencies: (at 1.2288 MHz)

Function	Single Timer		Dual Timers (Two Timers Cascaded)	
	Min	Max	Min	Max
Real-Time Interrupt Interval	1.63 μsec	53.3 msec	3.26 μsec	58.25 minutes
Rate Generator (Frequency)	18.75 Hz	614.4 kHz	0.00029 Hz	307.2 kHz

INTERFACE COMPATIBILITY

Serial I/O: EIA Standard RS232C signals provided and supported:

Carrier Detect	Receive Data
Clear to Send	Ring Indicator
Data Set Ready	Secondary Receive Data*
Data Terminal Ready	Secondary Transmit Data*
Request to Send	Transmit Clock
Receive Clock	Transmit Data
DTE Transmit	

* Optional if parallel port not used as ACU.

Parallel I/O: 4 input lines and 6 output lines; all signals compatible with EIA Standard RS232C. Directly compatible with Bell Model 801 Automatic Calling Unit.

System Bus: Compatible with Intel iSBC 80 Multibus.

I/O ADDRESSING: All communication to parallel and serial I/O ports, timers, and the interrupt controller is via read and write commands from the on-board 8085A CPU. Refer to table 3-2 for specific addresses.

COMPATIBLE CONNECTORS/CABLE: Refer to table 2-2 for compatible connector details.

POWER REQUIREMENTS:

	1	2	3	4
V_{CC} = +5V ± 5%	I_{CC} = 3.4 max.	I_{CC} = 3.3A max.	I_{CC} = 390mA max.	I_{CC} = 390mA max.
V_{DD} = +12V ± 5%	I_{DD} = 350mA max.	I_{DD} = 350 max.	I_{DD} = 176mA max.	I_{DD} = 20mA max.
V_{BB} = −5V ± 5%	I_{BB} = Note 5	I_{BB} = Note 5	I_{BB} = 5mA max.	I_{BB} = 5mA max.
V_{AA} = −12V ± 5%	I_{AA} = 200mA max.	I_{AA} = 200mA max.		

Notes:
1. Assuming two 2716 PROMs installed
2. No PROMs installed
3. For operational RAM only, for AUX power supply rating.
4. For RAM refresh only. Used for battery backup requirements. No RAM accessed.
5. V_{BB} is normally derived on board from V_{AA}. If V_{BB} supplied from bus, max requirement is 5mA.

ENVIRONMENTAL REQUIREMENTS

Operating Temperature: 0° to 55° (32° to 131°F).

Relative Humidity: To 90% without condensation.

PHYSICAL CHARACTERISTICS

Width: 30.48 cm (12.00 inches).
Depth: 17.15 cm (6.75 inches).
Thickness: 1.27 cm (0.50 inch).
Weight: 397 gm (14 ounces).

CHAPTER 2
PREPARATION FOR USE

2-1. INTRODUCTION

This chapter provides information for preparing the iSBC 544 Intelligent Communications Controller for use in the user-defined environment. This information includes unpacking and inspection; installation considerations; optional component installation; jumper configurations; multibus configuration; data set conversion; power fail/memory protect configuration; I/O cabling; and board installation.

2-2. UNPACKING AND INSPECTION

Inspect the shipping carton immediately upon receipt for evidence of mishandling during transit. If the shipping carton is severly damaged or waterstained, request that the carrier's agent is present when the carton is opened. If the carriers agent is not present when the carton is opened, and the contents of the carton are damaged, keep the carton and packing material for the agent's inspection.

For repairs to a product damaged in shipment, contact the Intel Technical Support Center (see paragraph 5-4) to obtain a Return Authorization Number and further instructions. A purchase order will be required to complete the repair. A copy of the purchase order should be submitted to the carrier with your claim.

It is suggested that salvageable shipping cartons and packing material be saved for future use in the event the product must be shipped.

2-3. INSTALLATION CONSIDERATIONS

The iSBC 544 Intelligent Communications Controller is designed for use as an "intellegent slave". It can be used any time the user desires to maximize I/O throughput with a minimum amount of impact on the system bus. The iSBC 544 is able to do this, because of its architecture which consists of a dedicated on-board 8085A CPU, dedicated on-board memory, and a variety of peripheral chips which perform such functions as format control, code conversions, data link control, error checking, and buffer management.

Important criteria for installing and interfacing the iSBC 544 in the above environment is presented in the following paragraphs.

2-4. USER FURNISHED COMPONENTS

Because the iSBC 544 can be used in a variety of applications, the user must purchase and install only those components which satisfy his particular needs. A list of components required to configure the iSBC 544 can be found in table 2-1. Table 2-2 is a list of the types and vendors of those connectors listed in table 2-1.

2-5. POWER REQUIREMENTS

The iSBC 544 requires +5V, +12V, and −12V power supply inputs. The currrents required from these supplies are listed in table 1-1.

Table 2-1. User Furnished and Installed Components

ITEM No.	ITEM	DESCRIPTION	USE
1	iSBC 604	Modular Backplane and Cardcage. Includes four slots with bus terminators.	Provides power input pins and Multibus signal interface between iSBC 544 and three additional boards in a multiple board system
2	iSBC 614	Modular Backplane and Cardcage. Includes four slots without bus terminators.	Provides four-board extensions of iSBC 604
3	Connector (mates with P1)	See Multibus Connector details in table 2-2.	Power inputs and Multibus signal interface. Not required if iSBC 544 installed in an iSBC 604/614.

Table 2-1. User Furnished and Installed Components (Cont'd.)

ITEM No.	ITEM	DESCRIPTION	USE
4	Connector (mates with P2)	See Auxiliary connector details in table 2-2.	Auxiliary backup battery inputs and associated memory protect functions
5	Connector (mates with J1, J2, J3, or J4)	See Serial I/O connector details in table 2-2.	Interfaces Serial I/O ports to Intel 8251A Programmable Communications Interface (USART)
6	Connector (mates with J5)	See parallel I/O connector details in table 2-2	Interface parrallel I/O port to Intel 8155
7	EPROM chips	Intel 2716 (2Kx8)	On-board UV erasable EPROM for program development and/or dedicated program use.
8	Capacitors	Four capacitors as required.	Rise time/noise capacitors for serial I/O port.
9	Jumpers	--	To connect optional power to serial I/O connectors J1-J4.
10	DIP Header Jumper	--	To convert data terminal interface to data set interface

Table 2-2. User Furnished Connector Details

FUNCTION	NO. OF PAIRS/ PINS	CENTERS (inches)	CONNECTOR TYPE	VENDOR	VENDOR PART NO.	INTEL PART NO.
Parallel/ Serial I/O Connector (J1-J5)	13/26	0.1	Flat Crimp	3M AMP ANSLY SAE	3462-0001 88106-1 609-2615 SD6726 Series	iSBC 955 Cable Set
Parallel/ Serial I/O Connector (J1-J5)	13/26	O.1	Soldered	TI AMP	H312113 1-583485-5	N/A
Parallel/ Serial I/O Connector (J1-J5)	13/26	0.1	Wire wrap[1]	TI	H311113	N/A
Multibus Connector (PI)	43/86	0.156	Soldered[1]	CDC[3] MICRO PLASTICS ARCO VIKING	VPB01E43D00A1 MP-0156-43-BW-4 AE443WP1 Less Ears 2VH43/1AV5	N/A

Table 2-2. User Furnished Connector Details (Cont'd.)

FUNCTION	NO. OF PAIRS/ PINS	CENTERS (inches)	CONNECTOR TYPE	VENDOR	VENDOR PART NO.	INTEL PART NO.
Multibus Connector (P1)	43/86	0.156	Wire wrap[1,2]	CDC[3] CDC[3] VIKING	VFB01E43D00A1 VPB01E43A00A1 2VH43/1AV5	MDS 985
Auxiliary Connector (P2)	30/60	0.1	Soldered[1]	TI VIKING	H312130 3VH30/1JN5	N/A
Auxiliary Connector (P2)	30/60	0.1	Wirewrap[1,2]	CDC[3] TI	VPB011B30A00A2 H311130	N/A

NOTES:
1. Connector heights are not guaranteed to conform to OEM packaging equipment.
2. Wire wrap pin lengths are not guaranteed to conform to OEM packaging equipment.
3. CDC VPB01... VPB02... VPB04... etc. are identical connectors with different electroplating thicknesses or metal surfaces.
4. Connector numbering convention may not agree with board connector numbers.

2-6. COOLING REQUIREMENTS

The iSBC 544 dissipates 275 gram-calories/minute (1.11 BTU/minute) and adequate circulation of air must be provided to prevent a temperature rise above 55°C (131°F). The System 80 enclosures and the Intellec System include fans to provide adequate intake and exhaust of ventilating air.

2-7. PHYSICAL DIMENSIONS

Physical dimensions of the iSBC 544 are as follows:
a. Width: 30.48cm (12.00 inches)
b. Height: 17.15cm (6.75 inches)
c. Thickness: 1.25cm (0.50 inch)

2-8. COMPONENT INSTALLATION

Instructions for installing the optional EPROMS, jumpers, and rise time/noise capacitors are given in the following paragraphs. When installing the optional chips; be sure to orient pin 1 of the chip adjacent to the white dot located near pin 1 of the associated IC socket. The grid location on figure 5-1 (parts location diagram) and figure 5-2 (schematic diagram) are specified for each user installed componet. Because the schematic diagram consists of nine sheets, grid references to figure 5-2 consist of four alphanumeric characters. For example reference 5ZB3 signifies sheet 5, zone B3.

2-9. EPROM CHIPS

Install the EPROM chips in IC sockets A35 and A51 (Refer to figure 5-1 zone C2 and figure 5-2 zone 4ZD3). Sockets A51 and A35 respectively, accommodate the low-order and high order addresses of the EPROM chip pair. For instance if two Intel 2716 EPROM's are installed, the chip installed in IC socket A51 is assigned addresses 0000-07FF and the chip installed in IC socket A35 is assigned addresses 0800-0FFF. The default (factory connected) jumpers are configured for Intel 2716 EPROMS. See Appendix D for 8K installation.

2-10 RISE TIME/NOISE CAPACITORS

Eye pads are provided so that rise time/noise capacitors may be installed as required on the individual serial I/O pins. The selection of capacitor values is at the option of the user and is normally a function of the particular environment. The location of these eye pads are as follows:

Capacitor	FIG 5-1	FIG 5-2
C5	D7	8ZD5
C9	D6	8ZB6
C13	D5	9ZD5
C17	D4	9ZB6

2-11 JUMPER CONFIGURATIONS

The iSBC 544 includes a variety of jumper-selectable options to allow the user to configure the board for his particular application. Table 2-3 summarizes these jumper selectable options and lists the grid reference locations of the jumpers as shown in figure 5-1 (part location diagram) and figure 5-2 (schematic diagram). The grid references for figure 5-2 are four alphanumeric characters long to denote sheet number and zone. For example grid reference 5ZB3 denotes sheet 5 zone B3.

Study table 2-3 carefully while making reference to figures 5-1 and 5-2. If the default (factory installed) jumper wiring is appropriate for a particular function, no further action is required for that function. If, however, a different configuration is required, remove the default jumper(s) and install the optional jumper(s) as specified . For most options, the information in table 2-3 is sufficient for proper configuration. Additional information, where necessary for clarity, is described in subsequent paragraphs.

2-12. PROM CONFIGURATION

Table 2-3 lists the jumper configuations for using 2716 PROM chips or Intel 2316 ROM chips. See Appendix D for use with Intel 2332 chips.

2-13. ON-BOARD RAM

The on-board 8085A has access to 16K of RAM starting at location 8000H. The addresses would be:

SIZE	LOCATIONS
16K	8000 - BFFF

Another bus master can access 4K, 8K, or 16K of iSBC 544 on-board RAM via the Multibus. The base address of this system accessable RAM is jumper and switch selectable as shown in table 2-3. It should be noted, that if the base address coming on the Multibus does not match the switch selectable base address, no RAM access will be allowed. This selection is performed by an Intel 3625-2 PROM located at A41 on the iSBC 544

Table 2-3 Jumper Selectable Options

FUNCTION	Fig 5-1 GRID REF	Fig 5-2 GRID REF	DESCRIPTION
PROM Configuration	C3	4ZC3	The following jumpers accomodate one of two types of PROM chips. Intel 2716 - 38-39, 40-41 Intel 2332 - See Appendix D.
PROM Size	C6	4ZB6	SWI Position 7 selects PROM size On (0) = 8K - See Appendix D. Off (1) = 4K
On-Board RAM (System Access)			The following describes the selection of on-board RAM by another system component.
	B6	4ZA7	*72-73, 74-75 - Selects lower 512K 73-74 - Selects upper 512K
	B6	4ZA6	51-52: 448-512K or 1024-1088K 53-54: 384-448K or 960-1024K 55-56: 320-384K or 832- 896K 57-58: 256-320K or 768- 832K Select 59-60: 192-256K or 704- 768K 64K 61-62: 128-192K or 640- 704K 63-64: 64-128K or 576- 640K *65-66: 0- 64K or 512- 576K NOTE Jumper selectable only on a 20 bit system.

Table 2-3 Jumper Selectable Options Continued

FUNCTION	Fig 5-1 GRID REF	Fig 5-2 GRID REF	DESCRIPTION
	C6	4ZB6	SW1 - Positions 1-4 select base address of the 4K, 8K, or 16K of RAM that is accessible by the system. (0=ON, 1=OFF) For example: Switch Setting 4321 — Base Address 0001 = 1000H 0010 = 2000H 0100 = 4000H 1101 = D000H (4K or 8K only) SW1- Positions 5-6 select RAM size as follows: Switch Setting 6 5 — RAM Size 0 0 — 4K 0 1 — 8K 1 0 — 16K 1 1 — NA Note If the base address you select does not allow for the RAM size you have selected, you will not be able to access RAM at all. This allows for the case where the iSBC 544 RAM is not used by the bus.
Bus Clock	B6	7ZD3	Jumper 76-77 to route Bus Clock signal BCLK/ to the Multibus. Only if this iSBC 544 is acting as bus master. Note The Frequency does not meet Multibus specifications.
Constant Clock	B6	7ZD3	Jumper 78-79 to route Constant Clock signal CCLK/ to the Multibus. Only if this iSBC 544 is acting as bus master. Note The frequency does not meet Multibus specifications.
Auxiliary Backup Power	C6	1ZC7 1ZC6	If auxiliary backup Power is employed to sustain memory during ac power outages, remove default jumpers *W12, *W13, and *W14
On-Board −5V Regulator	C6	1ZC6	The iSBC 544 requires a −5V AUX input to the on-board RAM chips. The −5V AUX input to the on-board RAM chips can be supplied by the on- board −5V regulator or by an auxiliary backup battery. (The on-board −5V regulator operates from the system −12V supply). If a system −5V supply is available disconnect default jumper W14 from between *A-B and connect it between B-C.
Timer Input Frequency (8253 PIT)	C7 C6	7ZD4 7ZB2	Input frequencies to 8253 Programmable Interval Timer counters are jumper selectable as follows: *Counter 0, 1, 2, and 3* (8251 Baud Rate Clocks).

Table 2-3. Jumper Selectable Options (Cont'd.)

FUNCTION	Fig 5-1 GRID REF	Fig 5-2 GRID REF	DESCRIPTION
			*30-29: 1.2288 MHz 30-31: 1.8432 MHz *Counter 4* (Secondary Baud Rate Clock). *30-29: 1.2288 MHz 30-31: 1.8432 MHz *Counter 5* (Interval Timer) *33-32: Output of Counter 4. 33-34: Same as Counters 0, 1, 2, 3, and 4. Jumper 33-32 effectively connects Counter 4 and Counter 5 in series in which the output of Counter 4 acts as the input clock for Counter 5. This allows for counting long time intervals (app. 1 hour total).
Timer Input Frequency (8155 Timer)	C3	7ZD4	Input frequencies to the 8155 Programmable Timer counter are jumper selectable as follows: *30-29: 1.2288 MHz 30-31: 1.8432 MHz
Priority Interrupts		2ZC6 2ZA7	There are a number of interrupts which can be interfaced to the on-board 8085A. An explanation of each of these interrupts can be found in paragraph 2-14. The interrupts are jumpered as follows:
Power Fail	B5		90-91: PFIN/ (Power Fail Interrupt) jumper to **TRAP** input on 8085A.
Timer Interrupt	C5		*49-50: TINT0 (Timer Interrupt 0) jumpered to **RST 7.5** input on 8085A.
Timer Interrupt	C7		*47-48: TINT1 (Timer Interrupt 1) jumpered to **RST 7.5** input on 8085A.
Flag Interrupt	C3		*43-44: FINT/ (Flag Interrupt) jumpered to **RST 5.5** input on 8085A.
Bus Interrupts (Input)	B6		81-82: INT0/ 81-83: INT1/ 81-84: INT2/ 81-85: INT3/ 81-86: INT4/ 81-87: INT5/ 81-88: INT6/ 81-89: INT7/ Bus Interrupts - jumpers one to **RST 5.5** and **SID** inputs on 8085A.
Bus Interrupt (output)	B6		80-82: INT0/ *80-83: INT1/ 80-84: INT2/ 80-85: INT3/

Table 2-3. Jumper Selectable Options (Cont'd.)

FUNCTION	Fig 5-1 GRID REF	Fig 5-2 GRID REF	DESCRIPTION
			80-86: INT4/ 80-87: INT5/ 80-88: INT6/ 80-89: INT7/ Jumpers *SOD* output of 8085A to one of the Bus Interrupt lines. SOD is 8085A's interrupt output to the Multibus.
Serial I/O Clocks (Baud Rate)			Jumper wires as required to connect inputs to Transmit Clock (TXC) and Receive Clock (RXC) of USART chips as follows (refer to paragraph 2-16).
	C7	8ZD6	**PORT 0** **TXC** **RXC** **SOURCE** *6-7 *2-4 BDG0 from PIT 0 1-2 BDG4 from PIT 1 5-6 2-4 XMIT CLK (external via J1) - 2-3 REC CLK (external via J1)
	C6	8ZA6	**PORT 1** **TXC** **RXC** **SOURCE** *13-14 *9-11 BDG1 from PIT 0 8-9 BDG4 from PIT 1 12-13 9-11 XMIT CLK (external via J2) - 10-9 REC CLK (external via J2)
	C5	9ZD6	PORT 2 TXC RXC SOURCE *20-21 *16-18 BDG2 from PIT 0 15-16 BDG4 from PIT 1 19-20 16-18 XMIT CLK (external via J3) - 17-16 REC CLK (external via J3)
	C4	9ZA6	**PORT 3** **TXC** **RXC** **SOURCE** *27-28 *23-25 BDG3 from PIT 1 22-23 BDG4 from PIT 1 26-27 23-25 XMIT CLK (external via J4) - 24-23 REC CLK (external via J4)
Serial I/O Port Interface			One 18-pin DIP header jumper assembly is supplied for each serial I/O port. These DIP header jumper assemblies allow the serial I/O ports to interface with RS232C devices as a data terminal (refer to paragraph 2-17).
	D7	8ZD3	Port 0 - W1
	D6	8ZB3	Port 1 - W2
	D5	9ZD3	Port 2 - W3
	D4	9ZB3	Port 3 - W4

Table 2-3. Jumper Selectable Options (Cont'd.)

FUNCTION	Fig 5-1 GRID REF	Fig 5-2 GRID REF	DESCRIPTION
TTY Adapter Interface Power			One 8-pin header jumper socket is supplied for each serial I/O port to supply power to a TTY Adapter. The jumper plugs are assigned as follows:
	C7	8ZC3	Port 0 - W5
	C6	8ZA3	Port 1 - W6
	C5	9ZC3	Port 2 - W7
	C4	9ZA3	Port 3 - W8
			Note The user must supply his own jumper header plugs.
Parallel I/O Port Outputs	D3	6ZD1 6ZC1	No optional jumpers
Master Mode	C6	4Z06	Set position 8 of S1 to on position.

*Default jumpers configured at the factory.

2-14. PRIORITY INTERRUPTS

Table 2-3 lists the source and destination of the interrupts which can be generated on the iSBC 544. For example, the FINT/ (Flag Interrupt) which signifies an off-board write to the RAM's base address generates an interrupt request to the RST 5.5 input on the 8085A.

There are two areas which require some explanation: the 8085A TRAP and RST 5.5, 6.5, and 7.5 interrupts.

The TRAP interrupt is useful for catastrophic errors such as power failure. On the iSBC 544, jumper 91-90 will allow the connection of PFIN/ (Power Failure) from the Multibus to the TRAP input on the on-board 8085A. The TRAP input is both level and edge sensitive. The TRAP interrupt has the highest priority, and can not be masked (disabled by the program).

RST 5.5, 6.5 and 7.5 interrupts cause the internal execution of an RST. These interrupts can be masked by the program. RST 7.5 is rising edge-sensitive, and RST 6.5 and 5.5 are high level-sensitive. These interrupts are default (factory connected) jumpered as shown in table 2-3.

2-15. COUNTER CLOCK FREQUENCY

The normal counter clock frequency is 1.2288 MHz. To change this frequency to 1.8432 MHz for greater timing flexibility, remove jumper 29-30 and connect jumper 30-31.

2-16. SERIAL I/O CLOCKS

Each of the two Programmable Interval Timers (PIT 0 and PIT 1) has three independent time/rate (Baud rate) generator sections as follows:

Timer	Counter	Output
PIT 0	0	BDG0
PIT 0	1	BDG1
PIT 0	2	BDG2
PIT 1	3	BDG3
PIT 1	4	BDG4
PIT 1	5	TINT1

There are four USART chips, one for each serial I/O port. Each USART chip, or serial I/O port, requires two clocks: a Transmit Clock (TXC) and a Receive Clock (RXC). These two clocks may be at the same frequency or at different frequencies.

The default (factory connected) clock for each serial I/O port is listed in table 2-3. Note that BDG0 serves as both TXC and RXC clock for Port 0, and that BDG1 through BDG3 serve as both the TXC and RXC clocks for Port 1 through Port 3 respectively.

Examination of table 2-3 shows that each port can accept inputs from five separate sources. Notice that each port can accept an externally supplied receive clock (REC CLK) and transmit clock (XMIT CLK). These clocks are input via the edge connector associated with each serial I/O port.

Clock signals BDG0 through BDG4 can be programmed for any integral submultiple of the iSBC 544 clock frequency (1.2288 MHz or 1.8432 MHz).

2-17. SERIAL I/O PORT INTERFACE

Each of the four serial I/O ports can be configured to accommodate RS232C devices. The iSBC 544 is supplied with four 18-pin DIP header jumper assemblies installed in sockets designated W1 through W4 to accommodate RS232C devices (refer to table 2-3). The jumper is set up for data terminal operation. To convert to data set operation see paragraph 2-21.

2-18. PARALLEL I/O PORT

The parallel I/O port has six parallel output lines and four parallel input lines that are compatible with the Bell Model 801 Automatic Calling Unit (ACU), or equivalent. The inputs and outputs of the parallel I/O port are controlled by an Intel 8155 Programmable Peripheral Interface (PPI) chip.

2-19. INPUT OPTIONS

Instead of the standard ACU input signals (PND, COS, DLO, and ACR), one input of each of the following pairs can be jumper-connected.

a. SRXD0 or CTS0/
b. SRXD1 or CTS1/
c. SRXD2 or CTS2/
d. SRXD3 or CTS3/

The four SRXD inputs are from the four serial I/O ports respectively: the CTS inputs from the USART chips can be monitored when jumpered as shown in table 2-3. IC A5 must be removed before CTS jumpers can be inserted.

2-20. OUTPUT OPTIONS

Instead of the standard ACU outputs (number bits NB1, NB2, NB4, NB8, CRQ, and DPR) the following outputs are available:

a. STXD0
b. STXD1
c. STXD2
d. STXD3

The four STXD outputs go to the four serial I/O ports, respectively. The Digit Present (DPR) and Call Request (CRQ) outputs are to an ACU.

2-21. DATA SET CONVERSION

Ports 0 through 3 are configured for data terminal operation in conjunction with an external *data set*. For certain applications, it may be necessary to convert one or more ports for data set operation in conjunction with an external *data terminal*. To convert to data set operation, proceed as follows:

a. Select port to be converted and remove associated 18-pin DIP header jumper assembly; e.g., for Port 0, remove DIP header jumper assembly from W1 (refer to table 2-3).
b. Wire a DIP header jumper assembly so that the following signals are reversed: (1) TXD and RXD, (2) RTS and CTS, and (3) DSR and DTR (see figure 2-1). Other signals may need to be reversed depending on the particular application.
c. Place reconfigured DIP header jumper assembly in the appropriate IC socket: W1 for Port 0, W2 for Port 1, W3 for Port 2, and W4 for Port 3.

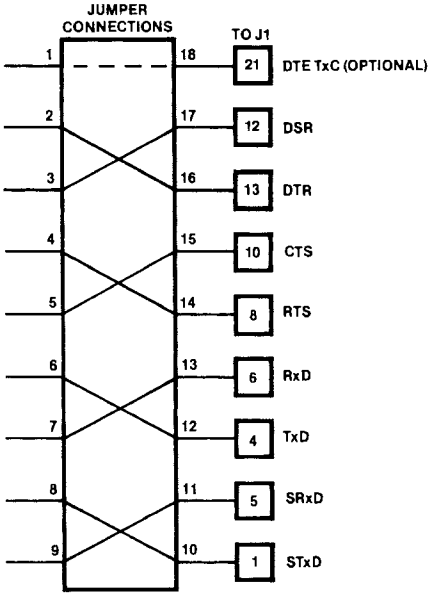


Figure 2-1. Reconfigured DIP Header Jumper Assembly for Data Set Operation.

2-22. MULTIBUS CONFIGURATION

For systems applications, the iSBC 544 is designed for installation in a standard Intel iSBC 604/614 Modular Backplane and Cardcage (refer to table 2-1 items 1 and 2). Alternatively, the iSBC 544 can be interfaced to a user-designed system backplane by means of an 86-pin connector (refer to table 2-1 item 3). Multibus signal characteristics and methods of implementing a serial or parallel priority resolution scheme for resolving bus contention in a multiple bus master system are described in the following paragraphs.

CAUTION

Always turn off the system power supply before installing the board in or removing the board from the backplane. Failure to observe this precaution can cause damage to the board.

2-23. SIGNAL CHARACTERISTICS

As shown in figure 1-1, connector P1 interfaces the iSBC 544 to the Multibus. Connector P1 pin assignments are listed in table 2-4 and descriptions of the signals are provided in table 2-5.

The dc characteristics of the iSBC 544 bus interface signals are provided in tables 2-6 and 2-7.

The ac characteristics of the iSBC 544 bus interface signals are provided in tables 2-8 and 2-9. Bus exchange and bus control timing is shown in figures 2-2 and 2-3 respectively.

Table 2-4. Multibus Connector P1 Pin Assignments

PIN*	SIGNAL	FUNCTION	PIN*	SIGNAL	FUNCTION
1	GND	Ground	44	ADRF/	Address bus
2	GND		45	ADRC/	
3	+5V	Power input	46	ADRD/	
4	+5V		47	ADRA/	
5	+5V		48	ADRB/	
6	+5V		49	ADR8/	
7	+12V		50	ADR9/	
8	+12V		51	ADR6/	
9	-5V		52	ADR7/	
10	-5V		53	ADR4/	
11	GND	Ground	54	ADR5/	
12	GND		55	ADR2/	
13	BCLK/	Bus Clock	56	ADR3/	
14	INIT/	System Initialize	57	ADR0/	
15		Reserved	58	ADR1/	
16			59		
17	BUSY/	Bus Busy	60		
18		Reserved	61		
19	MRDC/	Memory Read Command	62		
20	MWTC/	Memory Write Command			
21	IORC/	I/O Read Command	64		
22	IOWC/	I/O Write Command	65		
23	XACK/	Transfer Acknowledge	66		
24	INHI/	Inhibit RAM	67	DAT6/	Data bus
25			68	DAT7/	
26		Reserved	69	DAT4/	
27			70	DAT5/	
28	ADR10/	Extended Address Bus	71	DAT2/	
29		Reserved	72	DAT3/	
30	ADR11/	Extended Address Bus	73	DAT0/	
31	CCLK/	Constant Clock	74	DAT1/	
32	ADR12/	Extended Address Bus	75	GND	Ground
33		Reserved	76	GND	
34	ARD13/	Extended Address Bus	77		Reserved
35	INT6/	Interrupt request on level 6	78		
36	INT7/	Interrupt request on level 7	79	-12V	Power input
37	INT4/	Interrupt request on level 4	80	-12V	
38	INT5/	Interrupt request on level 5	81	+5V	
39	INT2/	Interrupt request on level 2	82	+5V	
40	INT3/	Interrupt request on level 3	83	+5V	
41	INT0/	Interrupt request on level 0	84	+5V	
42	INT1/	Interrupt request on level 1	85	GND	Ground
43	ADRE/	Address bus	86	GND	

*All odd-numbered pins (1,3,5...85) are on component side of the board. Pin 1 is the left-most pin when viewed from the component side of the board with the extractors at the top. All unassigned pins are reserved.

Table 2-5. Multibus Signal Functions

SIGNAL	FUNCTIONAL DESCRIPTION
ADR0/-ADRF/ ADR10/-ADR13/	*Address* . These 20 lines transmit the address of the memory location or I/O port to be accessed. ADRF/ is the most-significant bit except where ADR10/ through ADR13/ are used. ADR10/ through ADR13/ are transmitted only by those bus masters capable of addressing beyond 64K of memory. In this case, ADR13/ is the most-significant bit.
BCLK/	*Bus Clock* . Used to synchronize the bus contention logic on all bus masters. When generated by the iSBC 544, BCLK/ has a period of 180.84 nanoseconds (5.530 MHz) with a 35-65 percent duty cycle. (Does not conform to Multibus specifications).
BUSY/	*Bus Busy* . Indicates that the bus is in use and prevents all other bus masters from gaining control of the bus. BUSY/ is not synchronized with BCLK/ on the iSBC 544.
CCLK/	*Constant Clock* . Provides a clock signal of constant frequency for use by other system modules. When generated by the iSBC 544, CCLK/ has a period of 180.84 nanoseconds (5.530 MHz) with a 35-65 percent duty cycle. (Does not conform to Multibus Spec.)
DAT0/-DAT7/	*Data* . These eight bidirectional data lines transmit and receive data to and from the addressed memory location or I/O port. DAT7/ is the most-significant bit.
INH1/	*Inhibit RAM* . Prevents bus access to on-board RAM. Used to have PROM overlap iSBC 544 RAM.
INIT/	*Initialization* . Resets the entire system to a known internal state.
INT0/-INT7/	*Interrupt* . These eight lines are for inputting interrupt requests to the iSBC 544. INT0/ has the highest priority; INT7/ has the lowest priority. These lines may also be used to interface an interrupt signal from the iSBC 544 to the multibus.
IORC/	*I/O Read Command* . Indicates that the address of an I/O port is on the Multibus address lines and that the output of that port is to be read (placed) onto the data lines.
IOWC/	*I/O Write Command* . Indicates that the address of an I/O port is on the Multibus address lines and that the contents on the Multibus data lines are to be accepted by the addressed port.
MRDC/	*Memory Read Command* . Indicates that the address of a memory location is on the Multibus address lines and that the contents of that location are to be read (placed) on the Multibus data lines.
MWTC/	*Memory Write Command* . Indicates that the address of a memory location is on the Multibus address lines and that the contents on the Multibus data lines are to be written into that location.
XACK/	*Transfer Acknowledge* . Indicates that the addressed memory location or I/O port has completed the specified read or write operation. That is, data has been placed onto or accepted from the Multibus data lines.

Table 2-6. iSBC 544 DC Characteristics - Slave Mode

SIGNAL	SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	MAX.	UNIT
ADR0/-ADR13/	V_I	Input Low Voltage	V_{CC} = 5.0V		0.8	V
	V_{IH}	Input High Voltage	V_{CC} = 5.0V	2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.4V		-0.47	mA
	I_{IH}	Input Current at High V	V_{IN} = 2.7V		200	μA
	*C_L	Capacitive Load			18	pF
DAT0/-DAT7/	V_{OL}	Output Low Voltage	I_{OL} = 50 mA		0.6	V
	V_{OH}	Output High Voltage	I_{OH} = -10 mA	2.4		V
	V_{IL}	Input Low Voltage			0.95	V
	V_{IH}	Input High Voltage		2.0		V
	I_I	Input Current at Low V	V_{IN} = 0.45 V		-0.25	mA
	I_L	Output Leakage High	V_O = 5.25 V		100	μA
	*C_L	Capacitive Load			18	pF
INH1/	V_{IL}	Input Low Voltage			0.8	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.5V		-2	mA
	I_{IH}	Input Current at High V	V_{IN} =2.2V		50	μA
	C_L	Capacitive Load				pF
INT0/-INT7/	V_{OL}	Output Low Voltage	I_{OL} = 16 mA		0.4	V
	V_O	Output High Voltage	OPEN COLLECTOR			
	*C_L	Capacitive Load			18	pF
INIT/	V_{IL}	Input Low Voltage			0.8	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.4V		-2.2	mA
	I_{IH}	Input Current at High V	V_{IN} = 2.4V		-80	μA
	C_L	Capacitive Load				pF
MRDC/ MWTC/	V_{IL}	Input Low Voltage			0.8	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.45V	-1.6		mA
	I_{IH}	Input Current at High V	V_{IN} = 2.4V	80		μA
	C_L	Capacitive Load				pF
RS232C Inputs	V_{TH}	Input High Threshold Voltage		1.75	2.25	V
	V_{TL}	Input Low Threshold Voltage		.75	1.25	V
	I	Input Current	V_I = +3V	+.43		mA
			V_{IN} = -3V	-.43		mA
RS232C Outputs	V_O	High Level Output Voltage		9.0		V
	V_O	Low Level Output Voltage		-9.0	-12.0	V
	I_O +	High Level SS Output Current		-6.0		mA
	I_{OS}-	Low Level SS Output Current		6.0	12.0	mA
XACK/	V_{OL}	Output Low Voltage	I_{OL} = 32mA		0.4	V
	V_{OH}	Output High Voltage	I_{OH} = -5.2mA	2.4		V
	I_{LH}	Output Leakage High	V_O = 2.4V		40	μA
	I_{LL}	Output Leakage Low	V_O = 0.4V		-40	μA
	*C_L	Capacitive Load			15	pF

*Note: Capacitive Loads are approximate.

Table 2-7. iSBC 544 DC Characteristics - Master Mode

SIGNAL	SYMBOL	PARAMETER	TEST CONDITIONS	MIN.	MAX.	UNIT
ADR0/-ADRF/	V_{OL}	Output Low Voltage	I_{OL} = 15mA		0.5	V
	V_{OH}	Output High Voltage	V_{OH} = −1mA	2.4		V
	I_{LH}	Output Leakage High	V_O = 4V		200	μA
	I_{LL}	Output Leakage Low	V_O = 0.45V		−0.52	mA
	C_L	Capacitive Load			18	pF
ADR10/-ADR13/	V_{OH}	Output High Voltage	I_{OH} = 87 μA	2.4		V
BCLK/	V_{OL}	Output Low Voltage	I_{OH} = 40 mA		0.7	V
	V_{OH}	Output High Voltage	I_{OH} = −2mA	2.7		V
BUSY/ (OPEN COLLECTOR)	V_{OL}	Output Low Voltage	I_{OL} = 40mA		0.7	V
	C_L	Capacitive Load				pF
CCLK/	V_{OL}	Output Low Voltage	I_{OL} = 40mA		0.7	V
	V_{OH}	Output High Voltage	I_{OH} = −2mA	2.7		V
	C_L	Capacitive Load				pF
DAT0/-DAT7/	V_{OL}	Output Low Voltage	I_{OL} = 50 mA		0.6	V
	V_{OH}	Output High Voltage	I_{OH} = −10 mA	2.4		V
	V_{IL}	Input Low Voltage			0.95	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.45V		−0.25	mA
	I_{LH}	Output Leakage High	V_O = 5.25V		100	μA
	I_{LL}	Output Leakage Low	V_O = 0.45V		−100	μA
	C_L	Capacitive Load			18	pF
INIT/ (SYSTEM RESET)	V_{OL}	Output Low Voltage	I_{OL} = 40mA		0.7	V
	V_{OH}	Output High Voltage	OPEN COLLECTOR			
	V_{IL}	Input Low Voltage			0.8	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = .4V		−2.2	mA
	I_{IH}	Input Current at High V	V_{IN} = 2.4V		−80	μA
	C_L	Capacitive Load				pF
INT0/-INT7/	V_{IL}	Input Low Voltage			0.8	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = .4V		−0.63	mA
	I_{IH}	Input Current at High V	V_{IN} = 2.4V		30	μA
	C_L	Capacitive Load				pF
MROC/, MWTC/ IORC/, IOWC/	V_{OL}	Output Low Voltage	I_{OL} = 32mA		0.4	V
	V_{OH}	Output High Voltage	V_{OH} = −5.2mA	2.4		V
	I_{LH}	Output Leakage High	V_O = 2.4V		60	μA
	I_{LL}	Output Leakage Low	V_O = 0.45V		0.44	mA
	C_L	Capacitive Load				pF
XACK/	V_{IL}	Input Low Voltage			0.8	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.4V		−0.44	mA
	I_{IH}	Input Current at High V	V_{IN} = 2.4V		60	μA
	C_L	Capacitive Load				pF

*Note: Capacitive Loads are approximate.

Table 2-8. iSBC 544 AC Characteristics - Slave Mode

Parameter	Minimum (nsec)**	Maximum (nsec)**	Description	Remarks
t_{AS}	50		Address setup to command	From Address to command
t_{DS}	−200		Write data setup to command	
*t_{ACK}		740	Command to transfer acknowledge time	
t_{AH}	0		Address hold time	
t_{DH}	0		Write data hold time	
t_{DH}	0		Read data hold time	
t_{TO}		60	Acknowledge turnoff delay	
*t_{ACC}		660	Access time to read data	
t_{IH}	50		Inhibit hold time from command trailing edge	
t_{IPW}	100		Inhibit Pulse Width	
*t_{CY}		940	Minimum cycle time	$t_{ACK} + t_{SEP}$
*t_{OB1}		1490	On-board Memory Cycle Delay	No Refresh
*t_{OB2}		1850	On-board Memory Cycle Delay	On board cycle following refresh.
*t_{RD}		435	Refresh delay time	
t_{RI}	11600	12,500	Refresh interval	128 row refresh
t_{IS}	−50		Inhibit setup to command	Blocks RAM cycle and t_{ACK}
t_{SEP}	200		Command Separation	

* When an asynchronous refresh cycle occurs, t_{RD} is added to these parameters, when on-board memory cycle occurs, $t_{OB1,2}$ is also added.

** Except where noted.

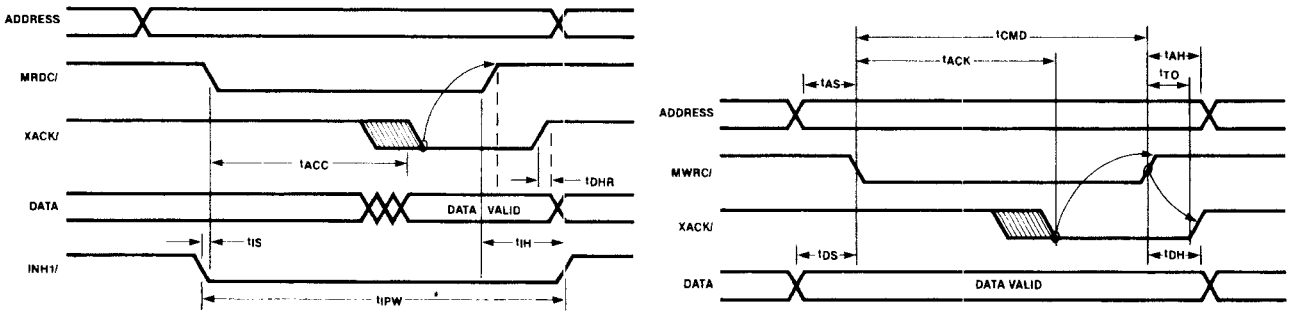


Figure 2-2. Bus Exchange Timing

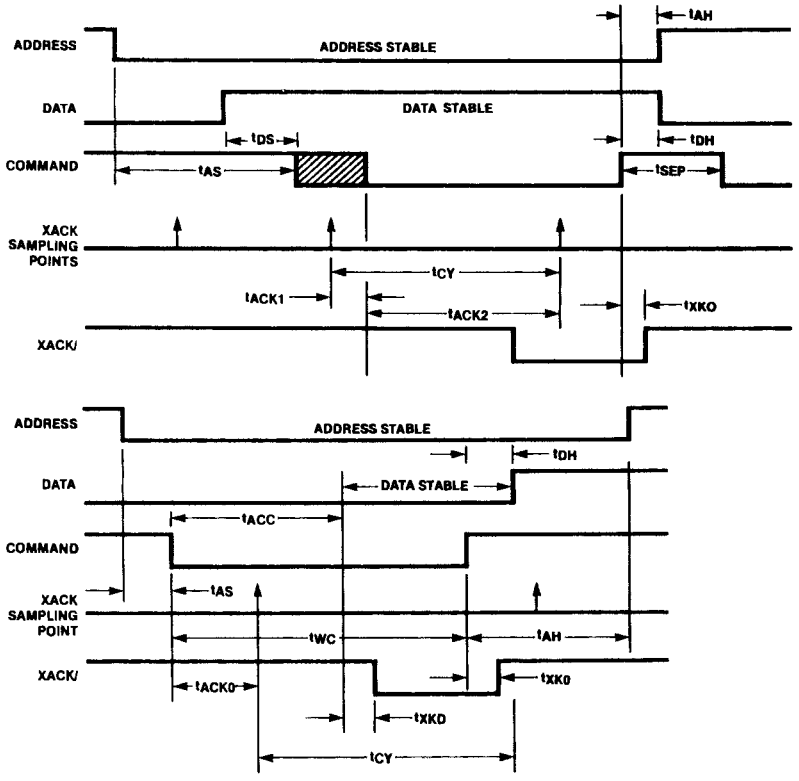


Figure 2-3. Bus Control Timing

Table 2-9. iSBC 544 AC Characteristics - Master Mode

Parameter	Write		IO Read		Mem Read		Description	Remarks
	Min (ns)	Max (ns)	Min (ns)	Max (ns)	Min (ns)	Max (ns)		
t_{AS}	330		200		50		Address Setup Time to Command	
t_{AH}	50		50		50		Address Hold Time	
t_{DS}	95						Data Setup Time to Command	
t_{DH}	50		0				Data Hold Time	
t_{ACK0}	-365	5	-135	135	50	310	First ACK Sampling Point of Current Cycle	Generates 0 Wait States
t_{ACK1}	-0	365	230	500	410	675	Second ACK Sampling Point of Current Cycle	Generates 1 Wait State
t_{ACK2}	360	730	590	860	770	1040	Third ACK Sampling Point of Current Cycle	Generates 2 Wait States
t_{CY}		362					ACK Sample Cycle Time	
t_{SEP}					200		Command Separation	
t_{WC}	280	450	430	550	593	775	Command Width	1
t_{ACC}				245		420	Read Access Time	1
t_{XKD}	0		0				XACK Delay From Valid Data or Write	
t_{XKO}	0	100	0	100	0	100	XACK Turn Off Delay	
t_{BCY}		181					Bus Clock cycle Time	544 Generator
t_{BW}	70	120					Bus Clock Low or High Periods	544 Generator
t_{INT}	3000						Initialization Width	After all voltages have stabilized

1 Assumes 0 Wait States. Each Wait State adds 362 ns

2 Read Command to next Read Command separation.

2-24. POWER FAIL/MEMORY PROTECT CONFIGURATION

If the Battery Backup feature is to be used a mating connector must be installed in the iSBC 604/614 Modular Cardcage and Backplane to accommodate the auxiliary connector P2 (refer to figure 1-1). Table 2-2 is a list of some of the 60-pin connectors that can be used for this purpose. Table 2-10 correlates the signals and pin numbers on the connector.

Procure the appropriate mating connector for P2 and secure it in place as follows:

a. Position holes in P2 mating connector over mounting holes that are in line with corresponding P1 mating connector.

b. From top of connector, insert two 0.5-inch /4-40 pan head screws down through connector and mounting holes.

c. Install a flat washer, lock washer, and star-type nut on each screw; then tighten the nuts.

When the mating connector for P2 is in place, wire the power fail signals to the appropriate pins of the connector as listed in table 2-8. In a typical system, these signals would be wired as follows:

a. Connect auxiliary signal common and returns for +5V, −5V, and +12V backup batteries to P2 pins 1 and 2.

b. Connect +5V battery input to P2 pins 3 and 4, −5V battery input to P2 pins 7 and 8 and +12V battery input to P2 pins 11 and 12. Remove jumpers W12, W13, and W14.

c. Connect MEM PROT/ input to P2 pin 20.

d. Connect PFIN/ input to P2 pin 19. To assign the PFIN/ input as the highest priority interrupt (8085A TRAP) connect jumper 90-91.

e. Connect HALT/ output at P2 pin 28 to external HALT indicator; which is typically a light- emitting diode (LED) mounted on the system enclosure.

f. Connect AUX RESET/ input to P2 pin 38. This signal is usually supplied by a momentary closure switch mounted on the system enclosure.

2-25. SERIAL I/O CABLING

The four serial I/O ports can be used with an RS232C device. Connection details for the devices are given in the following paragraphs. Compatible mating connectors for J1 through J4 are listed in table 1-1 (Specifications).

Table 2-10. Auxiliary Connector P2 Pin Assignments

Pin*	Signal	Definition
1	GND	Auxiliary common
2	GND	
3	+5V AUX	Auxiliary backup battery supply
4	+5V AUX	
7	−5V AUX	
8	−5V AUX	
11	+12V AUX	
12	+12V AUX	
19	PFI/	Power Fail Interrupt. This externally supplied signal is applied to the priority interrupt matrix. This signal should normally be jumpered to the 8085A microprocessor TRAP input.
20	MEM PROT/	Memory Protect. This externally supplied signal prevents access to RAM during battery backup operation.
28	HLT/	Halt. This output signal indicates that the 8085A microprocessor is halted.
38	AUX RESET/	Auxiliary Reset. This externally supplied signal initiates a power-up sequence; i.e., initializes the board and resets the entire system to a known internal state.

* All odd-numbered pins (1,3,5...59) are on component side of the board. Pin 1 is the left-most pin when viewed from the component side of the board with the extractors at the top.

Pin assignments and signal definitions for RS232C serial I/O communications are listed in table 2-11. Each of the four serial I/O ports is configured for data terminal operation. As described in paragraph 2-21, each port can alternatively be configured for data set operation by rewiring the DIP header jumper assembly.

The Intel iSBC 955 Cable Set consisting of two cable assemblies, is recommended for RS232C interfacing. One cable assembly consists of a 25-wire flat cable with a 26-pin PC edge connector at one end and an RS232C interfacing connector at the other end. The second cable assembly includes an RS232C connector at one end and has spade lugs at the other end; the spade lugs are used to interface to a teletypewriter. See Appendix B for ASR 33 TTY interface instructions. An iSBC-530 TTY Adapter is required to interface to a TTY.

For OEM applications where cables will be made for the iSBC 544, it is important to note that the mating connectors for J1 through J4 have one more pin (26) than an RS232C interface connector (25), which is used with a 25-wire flat cable. Consequently, when wiring the 26-pin mating connector, be sure that the cable makes contact with pins 1 and 2 of the mating connector, and not with pin 26.

Similarly, when installing the iSBC 544 with a 26-pin mating connector (J1 through J4), be sure that the connector is orientated properly on the serial I/O ports. If the connector is installed backward, no damage will occur but the I/O port will be inoperative.

NOTE

The numbers on the iSBC 544 card edge connector do not necessarily correspond with numbers on mating connectors.

Table 2-11. Connector J1-J4 RS232C Signal Interface

J1-J4[1] Pin	RS232C Pin	Signal Mnemonic	Definition
1	14	STXD	Secondary Transmit Data. Same as TXD except STXD is a secondary signal.
2	1	FGD	TTY Frame Ground. (Optional jumper plug)
3	15	XMIT CLK	Transmit Clock. External input clock signal for transmit data timing.
4	2	TXD	Transmit Data. Data transmitted from data terminal to data set.
5	16	SRXD	Secondary Receive Data. Same as RXD except SRXD is a secondary signal.
6	3	RXD	Receive Data. Data received by data terminal from data set.
7	17	REC CLK	Receive Clock. External input clock signal for receive data timing.
8	4	RTS	Request To Send. Control signal from data terminal to data set; sets data set in transmit mode.
9	18	—	Not used on iSBC 544.
10	5	CTS	Clear To Send. Control signal from data set to data terminal to indicate that data set is ready to transmit data; enables TXD output mode.
11	19	—	Not used in iSBC 544.
12	6	DSR	Data Set Ready. Indicates to data terminal that data set is connected to a communications channel; i.e., data set is not in Test, Talk, or Dial mode and timing and/or answer signals have been completed.
13	20	DTR	Data Terminal Ready. Indicates to data set that data terminal is ready to transmit or receive data.
14	7	SGD	Signal Ground.
15	21	—	Not used on iSBC 544.
16	8	CD	Carrier Detect. Signal from data set; indicates that data set is receiving a suitable signal.

Table 2-11. Connector J1-J4 RS232C Signal Interface (Cont'd.)

J1-J4[2] Pin	RS232C Pin	Signal Mnemonic	Definition
17	22	RI	Ring Indicator. Signal from data set; indicates that ringing signal has been received from a communications channel.
18	9	—	Not used for RS232C.
19	23	—	TTY Adapter PWR (−12V). (Optional jumper plug)
20	10	—	Not used for RS232C.
21	24	DTE TXC	Data Terminal Equipment Transmit Clock. Output from serial I/O port to data set to provide clock signal to transmitting signal converter.
22	11	—	TTY Adapter PWR (+12V). (Optional jumper plug)
23	25	—	TTY Adapter PWR (+5V). (Optional jumper plug)
24	12	—	Not used for RS232C.
25	N/C	SGD	Signal Ground.
26	13	—	Not used for RS232C.

NOTES:
1. J1-J4 pins 9, 11, 15, 18, 20, 24, and 26 are not used by iSBC 544.
2. Pin numbers refer to board connector pins only, they are not necessarily the same on the mating connectors.

Table 2-12. Connector J5 Parallel Output Signal Interface

J5 Pin	RS232C Pin	Signal Mnemonic	Description
1	14	NB1	Number Bit Lines. Binary Coded decimal (BCD) bits that indicate digits of number being called.
3	15	NB2	
5	16	NB4	
7	17	NB8	
4	2	DPR	Digit Present. Generated by data terminal; signal true indicates outputs NB1-NB8 have been set by data terminal and can be read by ACU.
6	3	ACR	Abandon Call; Retry. Generated by ACU to indicate that a call has failed (busy, no answer, dead line). Signal true suggests that if call has not been completed, it should probably be abandoned and retried later.
8	4	CRQ	Call Request. Generated by data terminal; signal true indicates a request for ACU to originate a call.
10	5	PND	Present Next Digit. Generated by ACU during dialing; signal true indicates ACU is ready to accept next digit output on NB lines. Signal false indicates data terminal must reset DPR output. PND will not be set true as long as DPR remains true. PND will be set true after data terminal resets DPR false after last digit on NB lines. PND will be true for duration of any call placed by ACU. PND will be false throughout all calls placed manually and throughout all incoming calls.
14	7	SGD	Signal Ground.
23	25		Not used by iSBC 544.
21	24		
16	8		

Table 2-12. Connector J5 Parallel Output Signal Interface (Cont'd.)

J5 Pin	RS232C Pin	Signal Mnemonic	Description
22	11	AUX 0	Auxiliary outputs; require jumper connection. (Refer to paragraph 2-20.)
24	12	AUX 1	
17	22	DLO	Data Line Occupied. Generated by ACU; signal true indicates to data terminal that data channel is in use.
26	13	COS	Call Origination Status. True indicates completed call.

Notes:

1. ACU is Automatic Calling Unit.
2 Pin numbers refer to board connector pins only, they are not necessarily the same on the mating connectors.

2-26. PARALLEL I/O CABLING

The parallel I/O port can be interfaced with the Intel iSBC 955 Cable Set described in paragraph 2-4. Pin assignments and signal definitions for the parallel I/O port are listed in table 2-12. Compatible mating connectors for J5 are listed in table 1-1 (Specifications).

2-27. BOARD INSTALLATION

CAUTION

Always turn off the computer system power supply before installing or removing the iSBC 544 board and before installing or removing device interface cables. Failure to take these precautions can result in damage to the board.

If an iSBC Single Board Computer based system, install the iSBC 544 in any slot that has not been wired for a dedicated function. In an Intellec Microcomputer Development System, install the iSBC 544 in any slot except slot 1 or 2. Attach the appropriate cable assemblies to connectors J1 through J5.

CHAPTER 3
PROGRAMMING INFORMATION

3-1. INTRODUCTION

The iSBC 544 Intelligent Communications Controller can operate in two modes:

a. Intelligent Slave Mode

b. Single Board Communications Computer

The first part of the this chapter will discuss the programming of the 544 as an intelligent slave, and the latter part will discuss the programming of the 544 as a bus master.

3-2. INTELLIGENT SLAVE CONCEPT

When the iSBC 544 operates as an intelligent slave, it can unburden a communications bound iSBC processor and perform such functions as format control, code conversions, data link control, error checking, data compression and buffer management. The 544 is capable of performing these functions, because of its architecture which consists of:

a. A dedicated 8085A CPU which controls the operation of the on-board memory and the on-board I/O devices.

b. Up to 8K bytes of ROM and 256 bytes of static RAM which are accessible only by the on board processor.

c. Dual port memory which is accessible from both the internal processor bus and the external system bus. This memory provides the primary means of communication between the on board processor and an external master.

d. Serial I/O which consists of four RS232C compatible I/O ports for interfacing with such things as data terminals, data sets, or other peripherals.

e. Parallel I/O which provides compatability with a Bell 801 Automatic Calling Unit (ACU), and additional lines for auxiliary control of external devices such as data sets and peripherals.

f. Interrupt control which provides the communication link between the on-board processor and the on-board I/O, and between the on-board processor and the system bus master.

With this architecture, the iSBC 544 can perform all the jobs unique to the I/O without constantly accessing the system bus and slowing down the master CPU. The 544 takes its direction from the master CPU, performs the necessary functions to interface with the external devices, controls the raw data transfer between I/O and memory, interrupts the bus master, and allows the bus master access to the on-board memory to retrieve the raw data or to enter raw data to be transferred to the external devices.

Operating in this manner, the iSBC 544 can maximize I/O throughput and minimize the impact on the system bus, which will allow for better throughput between the master CPU, system memory, and system I/O.

3-3. INTELLIGENT SLAVE PROGRAMMING

The programming of the iSBC 544 in the intelligent slave mode is separated into two parts: (1) system programming, and (2) local or on-board programming. The system programming concerns itself with the communication between the bus master (CPU) and the 544 (through the on-board 8085A). The local programming concerns itself with the on-board communication between the 8085A and the I/O devices. The following paragraphs describe these two aspects of the iSBC 544 programming.

3-4. SYSTEM PROGRAMMING

In the system programming environment, the iSBC 544 appears to be nothing more than an additional RAM memory module. The master CPU communicates with the iSBC 544 as if it were just an extension of system memory. Because the iSBC 544 is treated as memory by the system, the user is able to program into it a command structure which will allow the iSBC 544 to control its own I/O and memory operation. To enhance the programming of the iSBC 544, the user has been given some specific tools. The tools are; 1) the flag interrupt, 2) System bus RAM always mapped into on-board RAM at location 8000H. All iSBC 544's programmed with some firmware, 3) access to the bus interrupt lines.

The Flag Interrupt is generated anytime a write command is performed by an off-board CPU to the base address of the iSBC 544's RAM. This interrupt provides a means for the master CPU to notify the iSBC 544 that it wishes to establish a communication sequence. In systems with more than one intelligent slave, the flag interrupt provides a unique interrupt to each slave outside the normal eight bus interrupt lines.

The on-board RAM area that is accessible to both the master CPU and the on-board 8085A can be 4K, 8K or 16K, and can be located on any 4K boundry in the system. Whatever address is picked as the base address of the iSBC 544's RAM, is the address that will cause a flag interrupt when written into by the master CPU. This provides a unique interrupt to every intelligent slave on the system. Figure 3-1 shows the relationship of the on-board RAM to the system RAM.

The iSBC 544 can both exercise and respond to bus interrupts. With these tools in mind, the user can now develop his own command structure for the iSBC 544. The following paragraphs will give the user a possible approach to developing his own command structure for communicating with the iSBC 544.

3-5. COMMUNICATIONS AREA. The user must first reserve an area that can be set aside as a communications area between the master CPU and the on-board 8085A. For example this area could consist of the first 8 locations in the on-board RAM and would contain the information shown in figure 3-2.

The first byte in this communications area would be a command byte. This command byte could consist of a number of different simple commands such as:

a. Execute — which would cause the iSBC 544 to perform an instruction or series of instructions.

b. Reset — which would cause the iSBC 544 to reset all the peripheral devices by executing a series of codes.

c. Stop — which stops the execution of the instruction that the iSBC 544 is currently doing, and interrupts the master.

d. Test — would cause the iSBC 544 to write its status into the status byte(s).

or some complex command such as Transmit Data which would cause the iSBC 544 to transmit a block of data from one of its serial I/O ports. This command would require additional information such as what port, how much data, transfer speed, etc. The actual commands used will depend on the users specific application.

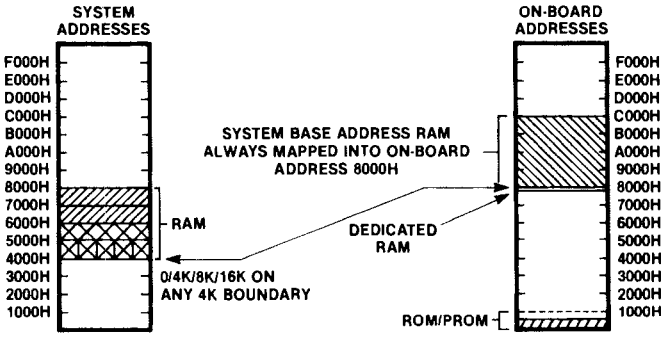


Figure 3-1. iSBC 544 Memory Addressing

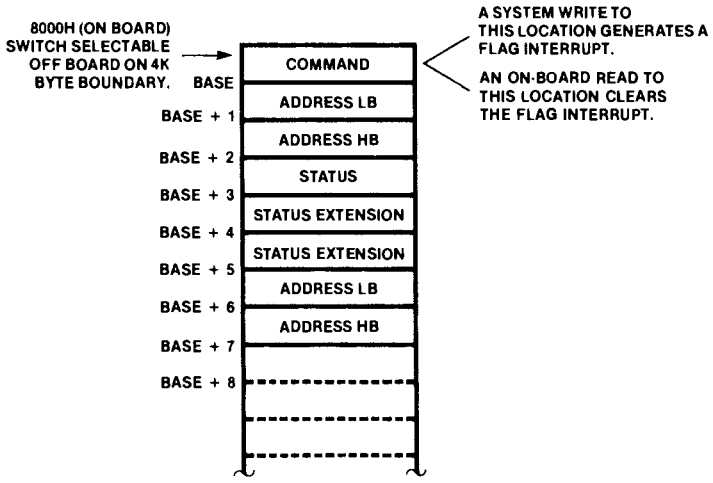


Figure 3-2. Communications Area

The next two bytes in the communications area would be the command address. This address could be the location where the iSBC 544 would find the additional information to perform the command. For example, in the case of the EXECUTE command this is location the CPU would jump to for the beginning of the instruction string.

The next byte in the communications area, would be the status byte. This byte is used by the on-board 8085A to communicate its status to the Master CPU. This status byte might contain such things as busy, done, error, etc. Typically, status cannot be communicated in one byte, so status extension bytes are necessary. These bytes could contain specific information such as the type of error, or the status of a particular port.

The next two locations are address locations which could be used to tell the off board processor where it can find the information it requested on a Receive command, or additional status.

It should be noted again, that this is simply an example of how a communications area could be set up. The actual number of locations used, and the contents of these locations is strictly up to the user.

3-6. COMMUNICATIONS PROTOCOL. When using the communications area, some type of control has to be exercised over the master CPU and the on-board 8085A so that they don't interfere with each other. This is done, by following some simple rules when setting up the software.

The first rule deals with the command byte. The off-board processor can only write a command byte into the base location of RAM if the location is zero. The on-board processor will zero the location after it reads it. The only time the on-board processor reads the command byte location, is when it gets a Flag Interrupt. This insures that the off-board processor has written a new command into this location. The Flag interrupt is cleared when the base address is read by the on-board processor.

The second set of rules deals with the status byte. The on-board processor can only write status when the status-byte location is zero. The status-byte location is zeroed by the system processor when it has read status.

3-7. COMMUNICATIONS PROGRAM SEQUENCE. The following discussion will explain the events necessary to execute a command sequence. The discussion will point out the steps that must be accomplished by both the system CPU, and by the on-board CPU. Figure 3-3 is a flow chart of this sequence.

The sequence starts with the system CPU checking the command byte location (Base) for zero. Finding it zero, the system CPU will write a command address into locations (Base + 1) and (Base + 2) and then a command into location (Base). This causes a Flag Interrupt to be generated.

The on-board processor detecting the Flag Interrupt reads the command byte location to determine what is to be done (this clears interrupt). It determines what to do by checking the command against the Command Interpreter which has been stored in memory. Once the type of command is determined, the command address is loaded and the command byte cleared.

The on-board processor then writes a status byte in the status byte location (Base + 3) to inform the system CPU that the command byte has been read, and that it can be executed. The on-board processor may interrupt the system CPU by way of a Multibus interrupt line to tell it to read the status byte. After generating the interrupt, the on-board processor starts executing the command. The system processor reads the status byte, clears it, and then readies itself to issue another command. The on-board processor will continue to execute the command, and will load additional status into the status byte location if necessary. At the completion of the command, the on-board processor will notify the system CPU by way of the interrupt line and status byte. This will cause the system CPU to jump out of its routine to determine what to do next, which might be to issue another command to the iSBC 544.

Again, this is only a typical sequence and will vary somewhat depending on the type of commands that the user decides to use.

3-8. COMMAND STRUCTURE. Commands like Execute can cause the iSBC 544 to execute a string of machine coded instructions. This creates no problem on simple commands, however to perform a complex function many bytes of code could be required. In order to simplify system programming of the iSBC 544, it is beneficial for the user to create a set of "macro-like" instructions which can perform a higher level function such as; set Baud rate generator X to XXXX, initialize USARTX to mode XXX, etc. The user could then write his program in a series of high level macro's which would require less bytes of memory and take up much less system time to transfer. To do this however, would require the user to have a library of these instructions stored in his PROMS. These instructions however, could be used over and over by the different programs that the iSBC 544 would be executing. Some of the types of high level instructions that could be used are shown throughout chapter three as programming examples.

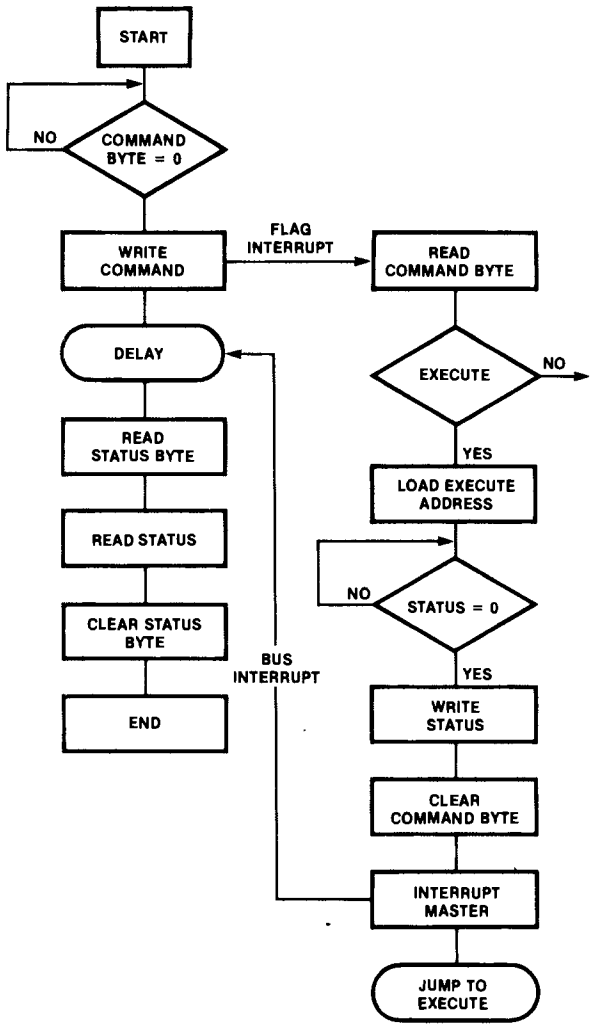


Figure 3-3. Communications Program Flow Chart

3-9. ON-BOARD PROGRAMMING

The on-board programming for the iSBC 544 will be presented by giving a detailed description of the programming of the following Intel chips:

a. Intel 8253 PIT (Programmable Interval Timer) that controls various frequency and timing functions.

b. Intel 8259 PIC (Programmable Interrupt Controller) that can handle up to eight vectored priority interrupts for the on-board microprocessor.

c. Intel 8155 Programmable Timer that supplies the on-board 8085A with an interval timer output.

d. Intel 8155 Programmable Peripheral Interface which controls the parallel I/O port.

e. Intel 8251A USART (Universal Synchronous/Asynchronous Receiver/Transmitter) that control the four serial I/O ports.

f. Intel 8085A Microprocessor interrupt capability only, will be discussed. The instruction set for the 8085A is included in Appendix A; a complete description of programming with Intel's assembly language is given in the *8080/8085A Assembly Language Programming Manual,* Manual Order No. 98-310.

In addition to the afore mentioned programming, this chapter also lists the on-board memory and I/O addresses and describes the effects of a system initialize command.

3-10. SYSTEM INITIALIZATION

When power is initially applied to the system, an initialize (INIT/) signal is automatically generated that clears the 8085A internal Program Counter, Instruction Register, and Interrupt Enable flip-flop and resets the Dual Port Logic, Flag Interrupt Logic, Master Mode Flop, and the 8155 Programmable Peripheral Interface. The 8155's I/O port is set to the input mode.

The initialize (INIT/) signal can also be generated by an auxiliary RESET switch. Depressing and releasing the RESET switch produces the same effect as the INIT/ signal described above.

The 8251A USART's must be initialized prior to processing. This is accomplished by programming the 8155 PPI. Refer to paragraph 3-31 for a detailed explanation of this procedure.

3-11. MEMORY ADDRESSING

The iSBC 544 which includes two IC sockets to accommodate up to 8K of user installable PROM is available with 16K of dynamic RAM and 256 bytes of static RAM. The iSBC 544 features a two-port RAM access arrangement in which the on- board RAM can be accessed by the on-board 8085A microprocessor or by a bus master board via the Multibus. The EPROM and the 256 byte static RAM can only be accessed by the on-board 8085A.

The on-board RAM can be accessed by a bus master that currently has control of the Multibus. It should be noted however, that even though the bus master may be accessing the iSBC 544 on-board memory, this does not lock out the on-board 8085A from accessing the on-board memory. In this situation, memory commands from the 8085A and the controlling bus master are interleaved. This, of course will impose 8085A wait states while the controlling bus master's memory command is being completed.

Addresses for EPROM and 8085A access of on-board RAM are provided in table 3-1. Note that the EPROM address space depends on the users configuration. For Multibus access the on-board RAM may be mapped into any 4K, 8K or 16K segment within the addressing constraints of the controlling bus master. For 16-bit Multibus addressing, the on-board RAM may be mapped into any 4K, 8K or 16K segment of the 64K byte address space. For 20-bit Multibus addressing, the on-board RAM may be mapped into any 4K, 8K, or 16K segment of the 1 megabyate address space. All memory must reside in the same 64K page.

When the 8085A is addressing *on-board* memory (RAM or EPROM), RMACK/(RAM Acknowledge) or IOACK/(I/O Acknowledge) is automatically generated to prevent imposing an 8085A wait state. When the 8085A is addressing *system* memory (only if iSBC 544 is Bus Master) it generates a Memory Read or Memory Write Command and waits for a Transfer Acknowledge (XACK/) to be received from the addressed memory device.

It should be noted in table 3-1 that it is possible to configure EPROM such as to create *illegal* addresses. If an illegal address is used in conjunction with a Memory Read Command to EPROM an IOACK/ (I/O Acknowledge) is generated as though the address was legal and the 8085A will continue executing the program. However, in this case, erroneous data will be returned.

3-12. I/O ADDRESSING

The on-board 8085A microprocessor communicates with the programmable chips through a sequence of I/O Read and I/O Write Commands. The I/O addresses for the different chips are shown in table 3-2.

Table 3-1. iSBC 544 On-Board Memory Addresses

Type	Configuration	Legal Address	Illegal Address
EPROM	One 2716 chip Two 2716 chips	0000 - 07FF 0000 - 0FFF	0800 -0FFF -
ROM	One 2332 chip Two 2332 chips	0000 - 0FFF 0000 - 1FFF	1000 - 1FFF
RAM	Eight 2117 chips	*8000 - BFFF	None
Static RAM	8155 chip	*7F00 - 7FFF	None

*Default (factory connected) jumper.

Table 3-2. I/O Address Assignments

I/O Address	Chip Select	Function
D0	8251 USART 0	Write: Data (Port 0) Read: Data (Port 0)
D1		Write: Mode or Command (Port 0) Read: Status (Port 0)
D2	8251 USART 1	Write: Data (Port 1) Read: Data (Port 1)
D3		Write: Mode or Command (Port 1) Read Status (Port 1)
D4	8251 USART 2	Write: Data (Port 2) Read: Data (Port 2)
D5		Write: Mode or Command (Port 2) Read: Status (Port 2)
D6	8251 USART 3	Write: Data (Port 3) Read: Data (Port 3)
D7		Write: Mode or Command (Port 3) Read: Status (Port 3)
D8	8253 PIT (#1)	Write: Counter 0 (Load Count + N) Read: Counter 0
D9		Write: Counter 1 (Load Count + N) Read: Counter 1
DA		Write: Counter 2 (Load Count + N) Read: Counter 2
DB		Write: Mode Word Read: None
DC	8253 PIT (#2)	Write: Counter 3 (Load Count + N) Read: Counter 3
DD		Write: Counter 4 (Load Count + N) Read: Counter 4
DE		Write: Counter 5 (Load Count + N) Read: Counter 5
DF		Write: Mode Word Read: None
E4	MASTER MODE FLOP	Set Master Mode
E5		Reset Master Mode
E6	8259 PIC	Write: ICW1, OCW2 and OCW3 Read: Status and Poll
E7		Write: ICW2, ICW3, and OCW1 (Mask) Read: OCW1 (Mask)
E8	8155 PPI	Write: Load Command Register Read: Status
E9		Write: Port A Read: Port A
EA		Write: Port B Read: Port B
EB		Write: Port C Read: Port C
EC	8155 PPI	Write: Load LSB of Count Length Read: Read LSB of Count Length
ED		Write: Load MSB + Mode Bits In Count Length Read: Read MSB + Mode Bits
EE		NOP
EF		NOP

3-13. 8253 PIT PROGRAMMING

The basic clock frequency for the programmable chips is supplied by a 22.1184-MHz crystal oscillator. This frequency is then divided by 12 and 18 to produce two jumper selectable clocks: 1.8432 MHz and 1.2288 MHz. These clocks are available for input to the counters on the 8253 PITs (Counters 0, 1, and 2 on the first 8253, and Counters 3, 4, and 5 on the 2nd 8253). The default (factory selected) and optional jumpers for selecting the clock inputs to the six counters are listed in table 2-3. The frequency of 1.2288 MHz is selected for compatability with the iSBC 534.

Default jumpers connect the outputs of the counters as shown in table 3-3.

Before programming the 8253 PITs, ascertain the input clock frequency and the output function of each of the six counters. These factors are determined and established by the user during the installation.

3-14. MODE CONTROL WORD COUNT

All counters must be initialized separately prior to their use. The initialization for each counter consists of two steps:

a. A mode control word (figure 3-4) is written to the control register for each individual counter.

Table 3-3. 8253 PIT Counter Outputs

Counter Output	Fig 5-2 Grid Ref	Function
Counter 0	8ZC6	BDG0 - Supplies clock input for TXC and RXC of 8251A USART 0.
Counter 1	8ZA6	BDG1 - Supplies clock input for TXC and RXC of 8251A USART 1.
Counter 2	9ZC6	BDG2 - Supplies clock input for TXC and RXC of 8251A USART 2.
Counter 3	9ZA6	BDG3 - Supplies clock input for TXC and RXC of 8251A USART 3.
Counter 4	7ZB1	BDG4 - Supplies clock input to Counter 5 to produce a long time interval counter, or split receiver clocks for the USARTS.
Counter 5	7ZB1	TINTI - Provides interval timer input to 8085A's RST 7.5 restart function.

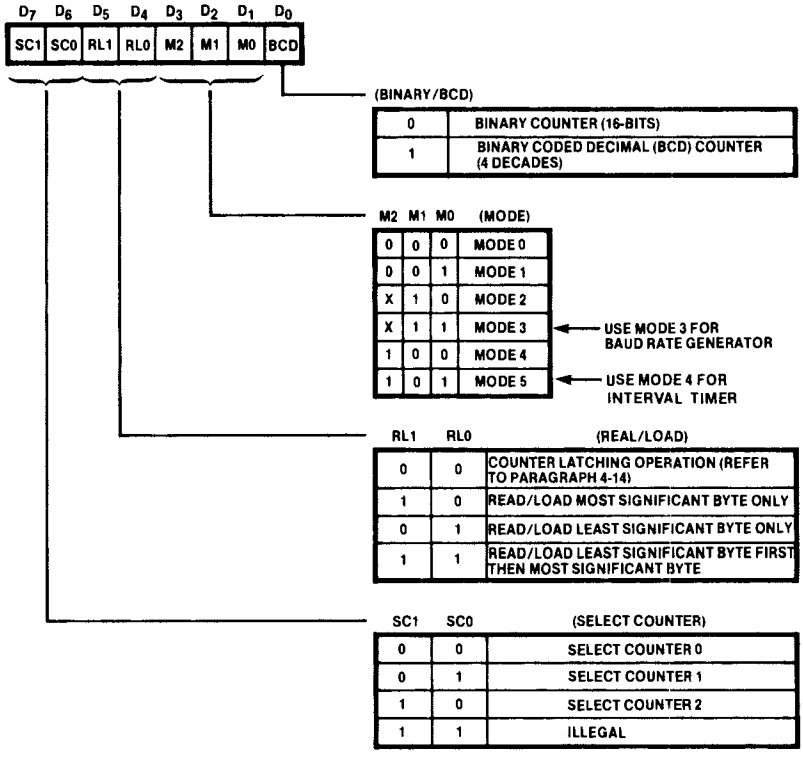


Figure 3-4. PIT Mode Control Word Format

b. A down-count number is loaded into each counter; number is in one or two 8-bit bytes as determined by mode control word.

The mode control word (figure 3-4) does the following:

a. Selects counter to be loaded.

b. Selects counter operating mode (either Mode 3 or Mode 4 is recommended for the iSBC 544).

c. Selects one of the following four counter read/load functions:
   (1) Counter latch (for stable read operation).
   (2) Read or load most-significant byte only.
   (3) Read or load least-significant byte only.
   (4) Read or load least-significant byte first, then most-significant byte.

d. Sets counter for either binary or BCD count.

The mode control word and the count register bytes for any given counter must be entered in the following sequence:

a. Mode control word.

b. Least-significant count register byte.

c. Most-significant count register byte.

As long as the above procedure is followed *for each counter*, the chip can be programmed in any convenient sequence. For example, mode control words first can be loaded into each of three counters per chip, followed by the least significant byte, etc. Figure 3-5 shows the two programming sequences described above.

Since all counters in the PIT chip are downcounters, the value loaded in the count registers is decremented. Loading all zeroes into a count register results in a maximum count of 2^{16} for binary numbers or 10^4 for BCD numbers.

When a selected count register is to be loaded, it *must* be loaded with the number of bytes programmed in the mode control word. One or two bytes can be loaded, depending on the appropriate down count. These two bytes can be programmed at any time following the mode control word, as long as the correct number of bytes is loaded in order.

The count mode selected in the control word controls the counter output. As shown in figure 3-4, the PIT chip can operate in any of six modes; however, the iSBC 544 normally uses only Mode 3 and Mode 4 as follows:

a. Mode 3: Square wave generator. Mode 3, which is the primary operating mode used in the iSBC 544, is used for generating Baud rate clock signals. In this mode, the counter output remains high until one-half of the count value in the count register has been decremented (for even numbers). The output then goes low for the other half of the count. If the value in the count register is odd, the counter output is high for (N + 1)/2 counts, and low for (N −1)/2 counts.

b. Mode 4: Software triggered strobe. After the mode is set, the output will be high. When the count is loaded, the counter will begin counting. On terminal count, the output will go low for one input clock period, then will go high again. If the count register is reloaded between output pulses the present period will not be affected, but the subsequent period will reflect the new value. Reloading the counter register will restart counting beginning with the new number.

NOTE

Mode 4 can only be used on Counter 5 (output = TINT1).

PROGRAMMING FORMAT

Step		
1		Mode Control Word Counter n
2	LSB	Count Register Byte Counter n
3	MSB	Count Register Byte Counter n

ALTERNATE PROGRAMMING FORMAT

Step		
1		Mode Control Word Counter 0
2		Mode Control Word Counter 1
3		Mode Control Word Counter 2
4	LSB	Counter Register Byte Counter 1
5	MSB	Count Register Byte Counter 1
6	LSB	Count Register Byte Counter 2
7	MSB	Count Register Byte Counter 2
8	LSB	Count Register Byte Counter 0
9	MSB	Count Register Byte Counter 0

Figure 3-5. PIT Programming Sequence Examples

3-15. ADDRESSING

As listed in table 3-2, each PIT uses four consecutive I/O addresses (D8 through DB for PIT /1 and DC through DF for PIT /2). The first three addresses for each PIT are used in loading and reading the count in the three counters on each chip. The fourth address on each chip is used in writing the mode control word to the desired counters.

3-16. INITIALIZATION

To initialize the PIT chips, peform the following:

a. Write mode control word for PIT 0 Counter 0 to DB. Note that *all* mode control words for PIT 0 are written to DB since mode control word must specify which counter is being programmed. (Refer to figure 3-4.) Table 3-4 provides a sample subroutine for writing mode control words to the six counters comprising PIT 0 and PIT 1.

b. Assuming mode control word has selected a 2-byte load, load least-significant byte of count into Counter 0 at D8. (Count value to be loaded is described in paragraphs 3-11 through 3-13.) Table 3-5 provides a sample subroutine for loading 2-byte count value.

c. Load most-significant byte of count into Counter 0 at D8.

NOTE

Be sure to enter the downcount in two bytes if the counter was programmed for a two-byte entry in the mode control word. Similarly, enter the downcount value in BCD if the counter was so programmed.

d. Repeat steps a, b, c and d for PIT 0 Counters 1 and 2, and for PIT 1 counters as necessary. Refer to table 3-2 for I/O addresses.

3-17. OPERATION

The following paragraphs describe operating procedures for a counter read, clock frequency divide/ratio selection, and interrupt timer count selection.

3-18. COUNTER READ. For Mode 3 operation, there usually is no requirement to reset or read the counters; however, it is possible to do so at any time. If a count register is reloaded during counting in Mode 3, the new value is reflected immediately following the output transition of the current count. For Mode 4 (interrupt on terminal count), reloading during counting has the following results:

a. Loading first byte stops current count.

Table 3-4. Typical PIT Control Word Subroutine

```
;INTTMR INITIALIZES INTERVAL TIMERS PIT 0 AND PIT 1
;FOUR OF THE COUNTERS ARE INITIALIZED AS BAUD RATE GENERATORS.
;THE OTHER TWO COUNTERS ARE SET UP AS INTERRUPT TIMERS.
;ALL SIX COUNTERS ARE SET UP FOR 16-BIT BINARY OPERATION.
;USES-NOTHING; DESTROYS-A

          PUBLIC    INTTMR
INTTMR:   MVI       A,36H          ;MODE 3 CONTROL WORD FOR COUNTERS 0 & 3
          OUT       0DBH
          OUT       0DFH
          MVI       A,76H          ;MODE 3 CONTROL WORD FOR COUNTER 1
          OUT       0DBH
          MVI       A,0B6H         ;MODE 3 CONTROL WORD FOR COUNTER 2
          OUT       0DBH
          MIV       A,78H          ;MODE 4 CONTROL WORD FOR COUNTER 4
          OUT       0DFH
          MIV       A,0B8H         ;MODE 4 CONTROL WORD FOR COUNTER 5
          OUT       0DFH
          RET

          END
```

Table 3-5. Typical PIT Count Value Load Subroutine

```
;LOAD0 LOADS COUNTER 0 FROM D & E. D IS MSB, E IS LSB.
;USES D,E; DESTROYS-A

          PUBLIC    LOAD0
LOAD0:    MOV       A,E            ;GET LSB
          OUT       0D8H
          MOV       A,D            ;GET MSB
          OUT       0D8H
          RET

          END
```

b. Loading second byte starts new count.

If desired, it is possible to read the count register during the down count. The recommended procedure is to use a mode control word to latch the contents of the count register; this ensures that the count reading is accurate and stable. The latched value of the count can then be read by the main processor.

NOTE

If a counter is read during the down count, it is mandatory to complete the read procedure; that is, if two bytes were programmed to the counter, then two bytes *must* be read before any other operations are performed with that counter.

A typical Counter read subroutine is given in table 3-6.

a. Write counter register latch control word (figure 3-6) to DB (PIT 0) or to DF (PIT 1), as appropriate. Control word specifies desired counter and selects counter latching operation.

b. Perform a read operation of desired counter, refer to table 3-2 for counter addresses.

NOTE

Be sure to read one or two bytes, whichever was specified in the initialization mode control word. For two bytes, read in the order specified.

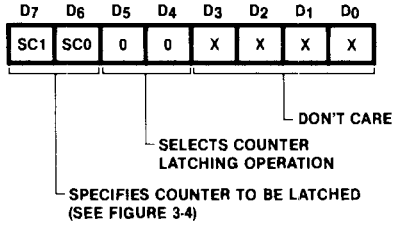


Figure 3-6. PIT Counter Register Latch Control Word Format

3-19. CLOCK FREQUENCY/DIVIDE RATIO SELECTION

The internal clock output-default jumper is connected so that the clock output frequency is 1.2288 MHz. The jumper can be changed so that output frequency is 1.8432 MHz. (Refer to paragraph 2-15.) This clock signal is divided by the counters in the two PIT chips to generate signals BDG0 through BDG5. The default wiring, in turn, connects signals BDG0 through BDG3 to the USART transmit clock (TXC) and receive clock (RXC) inputs as shown in table 2-3.

Each counter must be programmed with a down count number, of count value N. When count value N is loaded into a PIT counter, it becomes the clock divisor. To derive N for either synchronous or asynchronous operation, use the procedures described in the following paragraphs.

3-20. SYNCHRONOUS MODE

In the synchronous mode, the TXC and/or RXC rates equal the Baud rate. Therefore, the count value is determined by

$$N = C/B$$

where N is the count value,
B is the desired Baud rate, and
C is 1.2288 MHz (or 1.8432 MHz), the internal clock freuqency.

Thus, for a 4800 Baud rate, the required count value (N) is:

$$N = \frac{1.2288 \times 10^6}{4800} = 256.$$

If the binary equivalent of count value N = 256 is loaded into Counter 0 of PIT 0, then the output frequency of BDG0 (for USART 0) is 4800 Hz, which is the desired clock rate for synchronous mode operation.

Table 3-6. Typical PIT Counter Read Subroutine

```
;READ5 READS COUNTER 5 ON-THE-FLY INTO D & E. MSB IN D. LSB IN E.
;USES NOTHING; DESTROYS-A,D,E

            PUBLIC      READ5

READ5:      MVI         A,80H       ;MODE WORD FOR LATCHING COUNTER 5 VALUE
            OUT         0DFH
            IN          0DEH        ;LSB OF COUNTER
            MOV         E,A
            IN          0DEH        ;MSB OF COUNTER
            MOV         D,A
            RET

            END
```

3-21. ASYNCHRONOUS MODE. In the asynchronous mode, the TXC and/or RXC rates equal the Baud rate times of the following multipliers: X1, X16, or X64. Therefore, the count value is determined by

$$N = C/BM$$

where N is the count value,
B is the desired Baud rate,
M is the Baud rate multiplier (1, 16, or 64), and
C is 1.2288 MHz (or 1.8432 MHz), the internal clock frequency.

Thus, for a 4800 Baud rate, with a Baud rate multiplier of 16 the required count value (N) is

$$N = \frac{1.2288 \times 10^6}{4800 \times 16} = 16.$$

Table 3-7. PIT Count Value Vs Rate Multiplier for Each Baud Rate

Baud Rate (B)	*Count Value (N) FOR		
	M = 1	M = 16	M = 64
75	16384	1024	256
110	11171	698	175
150	8192	512	128
300	4096	256	64
600	2048	128	32
1200	1024	64	16
2400	512	32	8
4800	256	16	4
9600	128	8	2
19200	64	4	
38400	32	2	

* Count Values (N) assume clock is 1.2288 MHz. Multiply Count Values (N) by 1.5 for 1.8432 MHz. Count Values (N) and Rate Multipliers (M) are in decimal.

If the binary equivalent of count value N = 16 is loaded into Counter 1 of PIT 0, then the output frequency of BDG1 (for USART 1) is 4800 x 16 Hz, which is the desired clock rate for asynchronous mode operation. Count values (N) versus rate multiplier (M) for each Baud rate are listed in table 3-7.

NOTE

During initialization, be sure to load the count value (N) into the appropriate PIT counter and the Baud rate multiplier (M) into the USART.

3-22. RATE GENERATOR/ INTERVAL TIMER

Table 3-8 shows the maximum and minimum timer intervals when Counters 4 and 5 of PIT 1 are connected in parallel or series. These counters generate signals BDG4 and TINT1, which can be used either as auxiliary clock counters or to generate interrupt intervals.

3-23. INTERRUPT TIMER

To program an interval timer for an interrupt on terminal count, program the appropriate PIT for the correct operating mode (Mode 4) in the control word. Then load the count value (N), which is derived by

$$N = TC$$

where N is the timer count value
T is the desired interrupt time interval in seconds, and
C is the internal clock frequency (Hz).

Table 3-9 shows the count value (N) required for several time intervals (T) that can be generated for outputs BDG4 and TINT1.

Table 3-8. PIT Rate Generator Frequencies and Timer Intervals

Function	Single Timer[1] (BDG0 Thru BDG4 + TINT1)		Dual Timer[1] (BDG4 and TINT1 In Series)		Single Timer[2] (BDG0 Thru BDG4 + TINT1)		Dual Timer[2] (BDG4 and TINT1 In Series)	
	Minimum	Maximum	Minimum	Maximum	Minimum	Maximum	Minimum	Maximum
Rate Generator (Frequency)	18.75 Hz	614.4 kHz	0.00029 Hz	307.2 kHz	28.125 Hz	921.6 Hz	.00044 Hz	460.8 KHz
Real-Time Interrupt (Interval)	1.63 μsec	53.3 msec	3.26 μsec	58.25 minutes	1.09 μsec	35.5 msec	2.17 μsec	38.83 minutes

Notes:
1. Assuming a 1.2288 MHz clock input.
2. Assuming a 1.8432 MHz clock input.

Table 3-9. PIT Time Intervals Vs Timer Counts

T	N*
10 μsec	12
100 μsec	123
1 msec	1229
10 msec	12288
50 msec	61440

* Count Values (N) assume clock is 1.2288 MHz. For 1.8432 MHz multiply Count Value (N) by 1.5. Count values (N) are in decimal.

3-24. 8259 PIC PROGRAMMING

The 8259 Programmable Interrupt Controller (PIC) performs the function of an interrupt manager on the iSBC 544. It monitors the interrupt requests from eight separate sources. When one or more of the interrupt requests are active (true), the PIC determines the following:

a. Which input signal has the highest priority.

b. Whether the input signal has a higher priority than the interrupt presently being serviced by the processor. If so, the interrupt being serviced is interrupted; if not, the input signal is held for later output.

c. Whether the interrupt input bit is masked.

Thus the basic functions of the PIC are (1) to resolve the priority of interrupt requests and (2) issue a single interrupt request to the on-board 8085A based on the priority. The output of the 8259 PIC is applied directly to the INTR input at the 8085A microprocessor. (Refer to paragraph 2-14.)

3-25. INTERRUPT PRIORITY MODES

The PIC has two modes for resolving the priority of interrupt inputs: (1) fully nested mode and (2) rotating mode. The rotating mode has two variations: (1) auto-rotating and specific rotating.

3-26. FULLY NESTED MODE. In this mode the PIC input signals are assigned priority from 0 through 7. The PIC operates in this mode unless specifically programmed otherwise. Interrupt IR0 has the highest priority, IR7 has the lowest priority. When an interrupt is acknowledged, the highest priority request is available to the 8085A. Lower priority interrupts are inhibited; higher priority interrupts will be able to generate an interrupt that will be acknowledged if the 8085A has enabled its own interrupt input through its software. The End-Of-Interrupt (EOI) command from the 8085A is required to reset the PIC for the next interrupt.

3-27. AUTO-ROTATING MODE. In this mode the interrupt priority rotates. Once an interrupt on a given input is serviced, that interrupt assumes the lowest priority. Thus, if there are a number of simultaneous interrupts, the priority will rotate among the interrupts in numerical order. For example, if interrupts IR4 and IR6 request service simultaneously, IR4 will receive the highest priority. After service, the priority level rotates so that IR4 has the lowest priority and IR5 assumes the highest priority. In the worst case, seven other interrupts are serviced before IR4 again has the highest priority. Of course, if IR4 is the only request, it is serviced promptly. In the Auto-Rotating Mode, priority shifts when the PIC chip receives an EOI command.

3-28. SPECIFIC ROTATING MODE. In this mode the software can change interrupt priority by specifying the bottom priority, which automatically sets the highest priority. For example, if IR5 is assigned the bottom priority, IR6 assumes the highest priority. In the specific rotating mode, the priority can be rotated by writing a Specific Rotate at EOI (SEOI) command to the PIC chip. This command contains the BCD code of the interrupt being serviced; that interrupt is reset as the bottom priority. In addition, the bottom priority interrupt can be fixed at any time by writing a command word to the PIC chip.

3-29. INTERRUPT MASK

One or more of the eight interrupt request inputs can be individually masked during the PIC initialization or at any subsequent time. If an interrupt is masked while it is being serviced, lower priority interrupts are inhibited. There are two ways to enable the lower priority interrupts:

a. Write an End-of-Interrupt (EOI) command.

b. Set the Special Mask Mode.

The Special Mask Mode is useful when one or more interrupts are masked. If for any reason an input is masked while it is being serviced, the lower priority interrupts are disabled. However, it is possible to enable the lower priority interrupt with the Special Mask Mode. In this mode, the lower priority lines are enabled until the Special Mask Mode is reset. Higher priorities are not affected.

3-30. STATUS READ

Interrupt request inputs are handled by the following two internal PIC registers:

a. Interrupt Request Register (IRR), which stores all interrupt levels that are requesting service.

b. In-Service Register (ISR), which stores all interrupt levels that are being serviced.

Either register can be read by writing a suitable command word and then performing a read operation.

3-31. INITIALIZATION COMMAND WORDS

The eight interrupt service routines that are called by the PIC have eight addresses equally spaced in memory that can be programmed at intervals of four or eight bytes (see tables 3-28). Interrupt service routines thus occupy a 32 or 64-byte block respectively, of memory. The address format for device interrupt service routines is shown in figure 3-7.

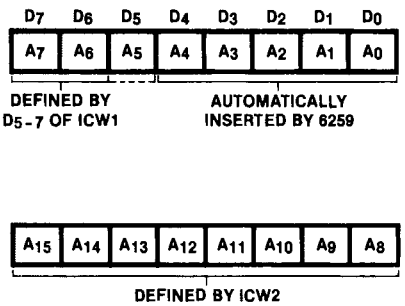


Figure 3-7. PIC Interrupt Routine Addresses

Bits 0-4 are automatically inserted by the 8259, while bits 6-15 are programmed by Initialization Command Words ICW1 and ICW2. Bit 5 is dependent on the address interval. If the interval is eight bits, bit 5 is automatically inserted by the PIC. If the interval is four bits, bit 5 is programmed in ICW1. Therefore, the 32 byte or 64 byte block of addresses reserved for interrupt service routines can be located anywhere in the available memory space. Table 3-10 shows the address format inserted by the PIC for each device.

Initialization of the 8259 PIC consists of writing two or three 8-bit Initialization Command Words as shown in figure 3-8. Since there are no slave PICs, the initialization for the one PIC consists of writing two Initialization Command Words as follows:

a. The first Initialization Command Word (ICW1) consists of the following:
   1) Bits 5-7 specify the most-significant bits of the lower address byte of the interrupt service routine.
   2) Bit 2 specifies the address interval.
   3) Bit 3 specifies whether or not there are slave (cascaded) PICs. Since there are no slave PICs, set bit 1 = 1.
   4) Bits 0, 3, and 4 identify the word as an ICW1.

b. The second word (ICW2) specifies the upper-byte (bits 8-15) of the interrupt service routine.

3-32. OPERATION COMMAND WORDS

After being initialized, the PIC can be programmed at any time for various interrupt modes. The Operation Command Word (OCW) formats are shown in figure 3-9 and discussed in paragraphs 3-27.

3-33. ADDRESSING

The PIC uses two consecutive addresses for writing to and reading internal registers. Address functions pertinent to programming are identified in table 3-2.

3-34. INITIALIZATION

To initialize the PIC proceed as follows (table 3-11 provides a typical initialization subroutine):

a. Disable system interrupts by executing a DI (Disable Interrupts) instruction.

Table 3-10. PIC Device Address Insertion

	Lower Routine Address Byte															
	Interval = 4								Interval = 8							
	Lower Memory Routine Address															
	D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
IR7	A7	A6	A5	1	1	1	0	0	A7	A6	1	1	1	0	0	0
IR6	A7	A6	A5	1	1	0	0	0	A7	A6	1	1	0	0	0	0
IR5	A7	A6	A5	1	0	1	0	0	A7	A6	1	0	1	0	0	0
IR4	A7	A6	A5	1	0	0	0	0	A7	A6	1	0	0	0	0	0
IR3	A7	A6	A5	0	1	1	0	0	A7	A6	0	1	1	0	0	0
IR2	A7	A6	A5	0	1	0	0	0	A7	A6	0	1	0	0	0	0
IR1	A7	A6	A5	0	0	1	0	0	A7	A6	0	0	1	0	0	0
IR0	A7	A6	A5	0	0	0	0	0	A7	A6	0	0	0	0	0	0

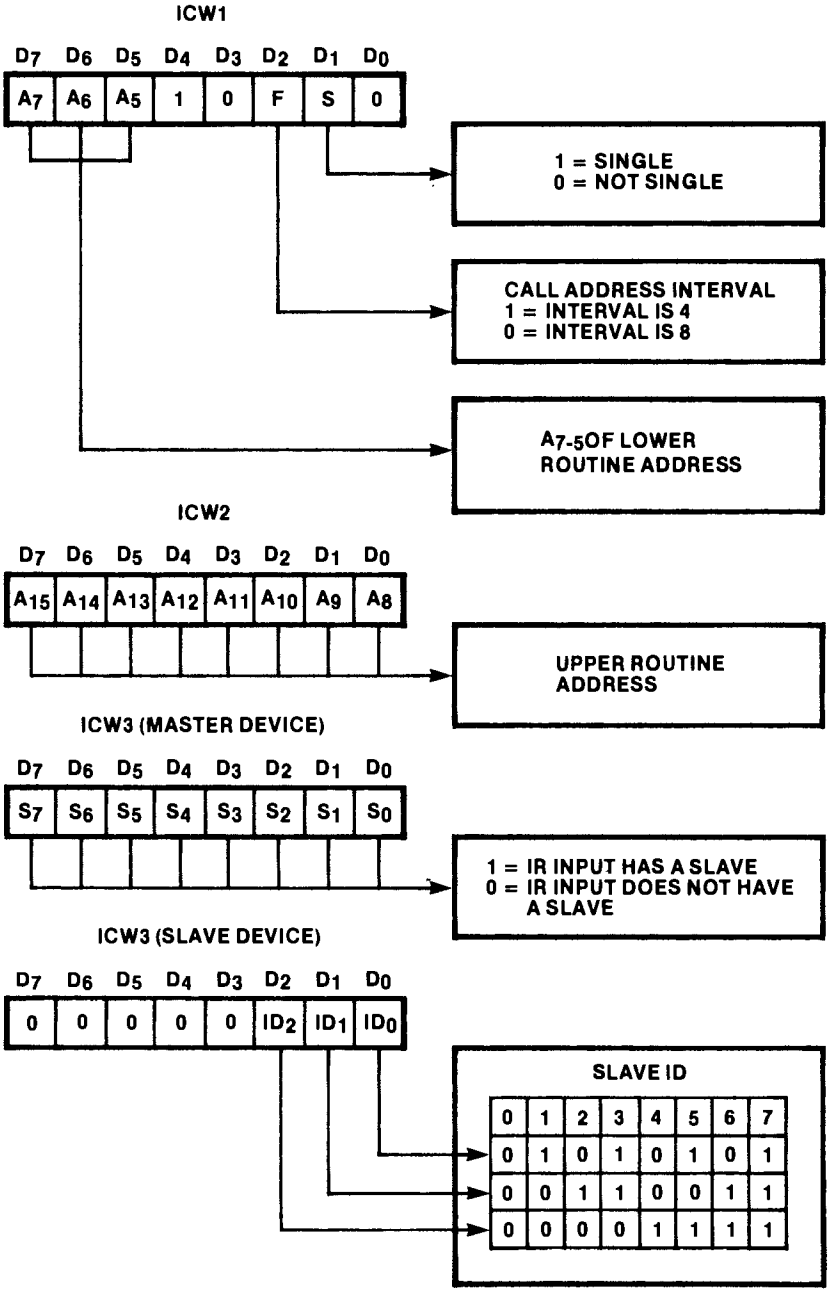


Figure 3-8. PIC Initialization Command Word Formats

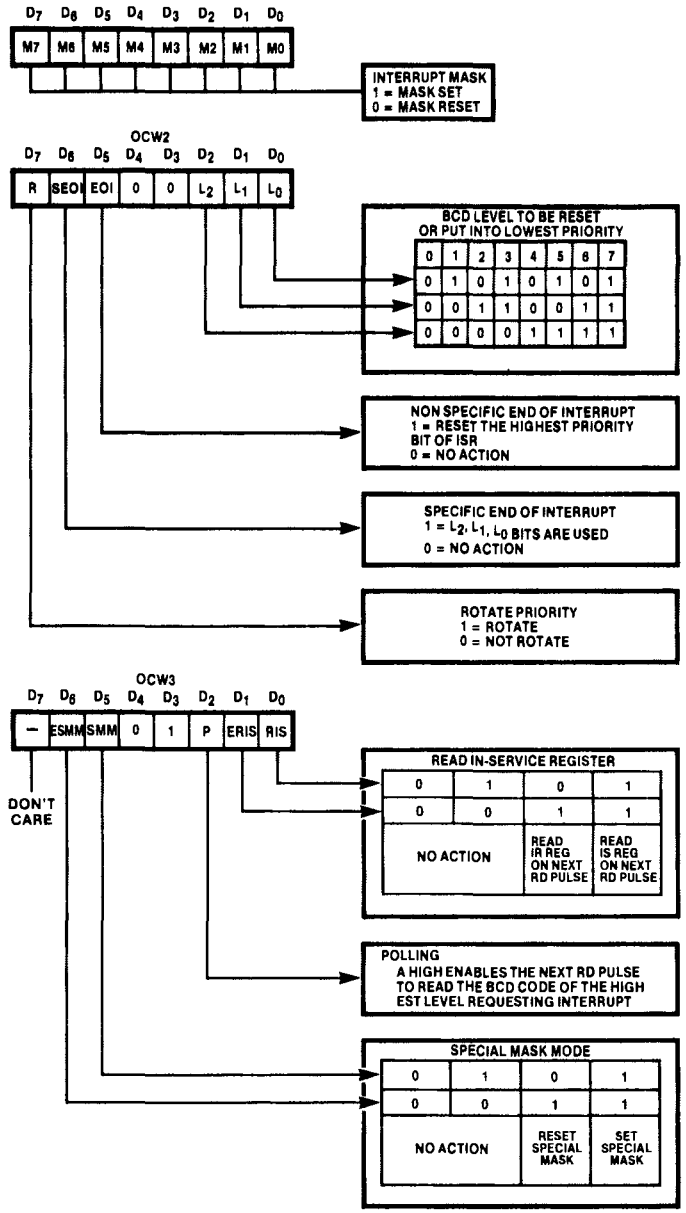


Figure 3-9. PIC Operation Control Word Formats

b. Write ICW1 to E6.

c. Write ICW2 to E7.

d. Enable system interrupts by executing an EI (Enable Interrupts) instruction.

NOTE

The PIC chip operates in the fully nested mode after the initialization sequence without requiring any Operation Control Word (OCW), however the mask register must be properly initialized.

3-35. OPERATION

After initialization, the PIC can be programmed at any time for the following operations:

a. Auto-rotating priority.

b. Specific rotating priority.

c. Status read of Interrupt Request Register (IRR).

d. Status read of In-Service Register (ISR).

e. Interrupt mask bits set, reset, or read.

f. Special mask mode set or reset.

Table 3-12 lists the details of the PIC programming operations. Note that an End-Of-Interrupt (EOI) or a Special-End-Of-Interrupt (SEOI) command is required at the end of each interrupt service routine to reset the ISR. The EOI command is used in the fully nested, polled, and auto-rotating priority modes and the SEOI command, which specifies the bit to be reset, is used in the specific rotating priority mode. Table 3-13 through 3-17 provide typical subroutines for the following:

a. Read IRR (table 3-13)

b. Read ISR (table 3-14)

c. Set mask register (table 3-15)

d. Read mask register (table 3-16)

e. Issue EOI command (table 3-17)

Table 3-11. Typical PIC Initialization Subroutine

```
;INT59 INITIALIZES THE INTERRUPT CONTROLLER ON THE SBC-544.
;STANDARD VECTORED INTERRUPTS ARE USED WITH AN 8-BYTE SPACING.
;IR0 VECTORS TO 40H, IR7 VECTORS TO 78H.ALL INTERRUPTS ARE MASKED
;BY THIS ROUTINE AND INTERRUPTS ARE DISABLED.
;USES-NOTHING DESTROYS-A

        PUBLIC  INT59
        EXTRN   BASAD

INT59;  DI                      ; DISABLE INTERRUPTS
        MVI     A,01010010B     ; ICWIC INSTRUCTION
        OUT     0E6H
        MVI     A,0             ; ICW2
        OUT     0E7H
        MVI     A,0FFH          ; MASK ALL INTERRUPTS
        OUT     0E7H
        RET

        END
```

Table 3-12. PIC Operation Procedures

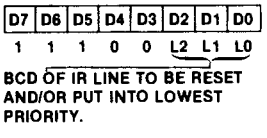
Operation	Procedure
Auto-Rotating Priority Mode	To set: In OCW2, write a rotate Priority at EOI command (A0H) to E6. Terminate interrupt and rotate priority: In OCW2, write EOI command (20H) to E6.
Specific Rotating Priority Mode	To set: In OCW2, write a Rotate Priority at SEOI command in the following format to E6: 

Table 3-12. PIC Operation Procedures (Cont'd.)

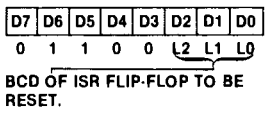
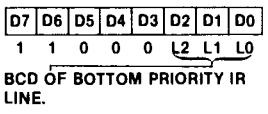
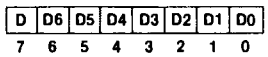
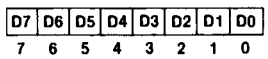
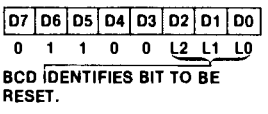
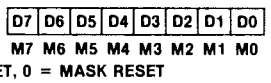
Operation	Procedure
	To terminate interrupt and rotate priority: In OCW2, write an SEOI command in the following format to E6:  To rotate priority without EOI: In OCW2, write a command word in the following format to E6: 
Interrupt Request Register (IRR) Status	The IRR stores a "1" in the associated bit for each IR input line that is requesting an interrupt. To read the IRR (refer to footnote): (1) Write OAH to E6. (2) Read E6. Status Format is: 
In-Service Register (ISR) Status	The ISR stores a "1" in the associated bit for priority inputs that are being serviced. The ISR is updated when an EOI command is issued. To read the ISR (refer to footnote): (1) Write OBH to E6. (2) Read E6. Status format is:  Be sure to reset ISR bit at end-of-interrupt when in the following modes: Auto-Rotating (both types) and Special Mask. To reset ISR in OCW2, write: 
Interrupt Mask Register	To set mask bits in OCW1, write the following mask byte to E7:  To read mask bits, read E7.
Special Mask Mode	The Special Mask Mode enables desired bits that have been previously masked; lower priority bits are also enabled. To set, write 68H to E6. To reset, write 48H to E6.
NOTE: If previous operation was addressed to same register, it is not necessary to rewrite the OCW.	

Table 3-13. Typical PIC Interrupt Request Register Read Subroutine

```
;RR0 READS INTERRUPT REQUEST REG
;USES-NOTHING; DESTROYS-A

                PUBLIC          RR0

RR0:            MVI             A,0AH           ;OCW3 RR INSTRUCTION TO PIC
                OUT             0E6H
                IN              0E6H
                RET

                END
```

Table 3-14. Typical PIC In-Service Register Read Subroutine

```
;RIS READS IN-SERVICE REGISTER OF PIC
;USES-NOTHING; DESTROYS-A

                PUBLIC          RIS0

RIS:            MVI             A,0BH           ;OCW3 RIS INSTRUCTION TO PIC
                OUT             0E6H
                IN              0E6H
                RET

                END
```

Table 3-15. Typical PIC Set Mask Register Subroutine

```
;SMASK STORES A REG INTO MASK REG OF PIC
;A ONE MASKS OUT AN INTERRUPT, A ZERO ENABLES IT
;USES-A; DESTROYS-NOTHING

                PUBLIC  SMASK

SMASK           OUT     0E7H
                RET

                END
```

Table 3-16. Typical PIC Mask Register Read Subroutine

```
;RMASK READS MASK REG OF PIC INTO A REG
;USES-NOTHING; DESTROYS-A

                        PUBLIC          RMASK

RMASK:                  IN              0E7H
                        RET

                        END
```

Table 3-17. Typical PIC End-of-Interrupt Command Subroutine

```
;EOI ISSUES END-OF-INTERRUPT TO PIC
;USES-NOTHING; DESTROYS-A

            PUBLIC      EOI

EOI0:       MVI         A,20H       ;NON-SPECIFIC EOI
            OUT         0E6H
            RET

            END
```

3-36. 8155 PROGRAMMABLE PERIPHERAL INTERFACE AND TIMER

The Intel 8155 is made up of the following operational areas:

a. 256 x 8 Static RAM.

b. 14 bit Timer.

c. Two 8-bit I/O ports (one input and one output) and one 6-bit I/O input port.

3-37. 8155 I/O PORT PROGRAMMING

The parallel I/O port, which is controlled by the Intel 8155 PPI chip, is designed to interface directly with a Bell Model 801 Automatic Calling Unit (ACU). The PPI chip has two 8 bit parallel ports (Port A which is an output port, and Port B which is an input port) and a 6-bit parallel input port (Port C). The following table is a list of the signals interfaced to the PPI chip.

Port A Outputs	Port B Inputs	Port C Inputs
NB1 or STXD0	RI0 (Port 0)	PND or SRXD0
NB2 or STXD1	RI1 (Port 1)	COS or SRXD1
NB4 or STXD2	RI2 (Port 2)	DLO or SRXD2
NB8 or STXD3	RI3 (Port 3)	ACR or SRXD3
PGRST (For 8251 USARTS)	CD0 (Port 0)	FINT
Reset for INT FLOPS	CD1 (Port 1)	PFS
CRQ	CD2 (Port 2)	
DPR	CD3 (Port 3)	

NOTES

1. RI and CD signals are connected to interrupt circuit.
2. RI and CD are from serial I/O ports.

3-38. ADDRESSING OF I/O PORTS. The I/O section of the 8155 consists of a Command/Status Register (C/S) and one register for each of the three I/O ports. Addresses for these four registers are provided in table 3-2.

3-39. INITIALIZATION. The 8155 is reset when power is initially applied to the system. However an initialize routine must be generated for the 8155 in order to set up the ACU interface, and to remove the reset condition from the USARTS and from the RINT and PINT interrupt flops. Table 3-18 is a typical initialize routine.

3-40. COMMAND REGISTER FORMAT. The Command Register consists of eight 1-bit latches. Bits 0-3 define the mode of Port A, B, and C, bits 4 and 5 enable or disable Port A and B interrupts, and bits 6 and 7 are used for the Timer portion of the 8155. Figure 3-10 shows the Command Register format.

The Command Register can be altered at any time by performing an I/O write command to port E8. Table 3-19 is a typical Command Register Load routine.

3-41. STATUS REGISTER FORMAT. The Status Register consists of seven 1-bit latches. Bit 0-5 define the status of the ports, and bit 6 defines the status of the timer. The Status Register format is shown in figure 3-11.

The contents of the Status Register can be obtained at any time by performing an I/O read to port E8.

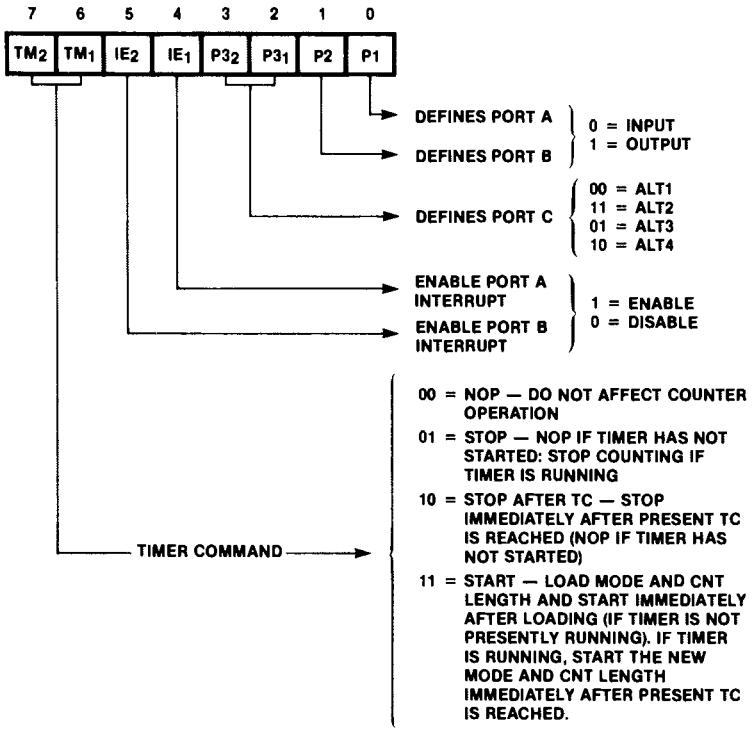


Figure 3-10. Command Register Format

Table 3-18. Typical 8155 Initialize Routine

```
;INTAUX INITIALIZES THE 8155 PARALLEL I/O
;PORT CHIP. THE DATA IN REG C IS PUT OUT
;ON PORT A. THE TIMER IS STOPPED.

                PUBLIC      INTAUX

INTAUX:         MVI         A,41H       ;PORT A OUT; PORTS B + C
                                         IN; STOP TIMER
                OUT         0E8H
                MOV         A,C         ;GET PORT A DATA
                OUT         0E9H
                RET

                END
```

NOTE

Register C must be loaded with the correct bit configuration to set up the ACU interface and to reset the USARTS, and the Ring Indicator and Carrier Detect Interrupt flops. A typical bit configuration would be C0H.

Table 3-19. Typical Command Register Load Routine

```
;INTAUX INITIALIZES THE 8155 PARALLEL I/O PORT CHIP. THE DATA IN REG
;C IS PUT OUT ON PORT A. THE TIMER IS STOPPED.

            PUBLIC      INTAUX
            EXTRN       BASAD

INTAUX;     MVI         A,41H       ;PORT A OUT; PORTS B&C IN; STOP TIMER
            OUT         E8H
            MOV         A,C         ;GET PORT A DATA
            OUT         E9H
            RET

            END
```

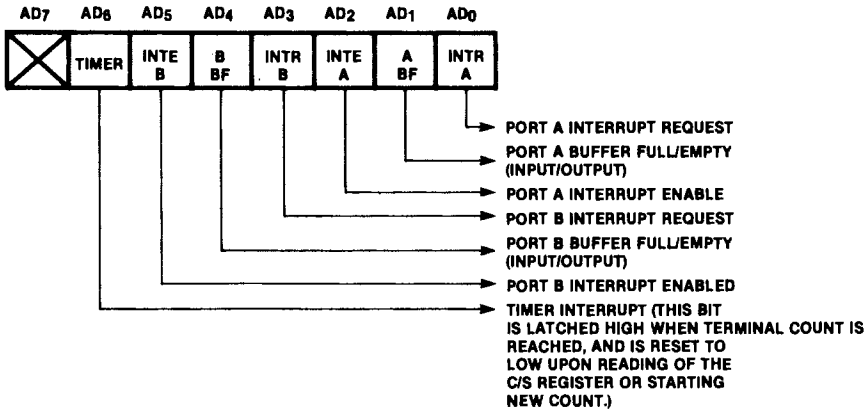


Figure 3-11. Status Register Format

3-42. PORT A PROGRAMMING

Port A is an output port, and is written to by the CPU. Data is written to Port A, or a call is placed with the ACU, by performing a write to E9.

The status of the previous bits written to Port A can be obtained by performing a read to E9.

Figure 3-12 shows the bit definitions for Port A.

3-43. PORT B AND C PROGRAMMING

Data from Port B and Port C is read by performing a read of EA and EB, respectively. Bit definitions for Ports B and C are given in figure 3-13.

Typical routines for programming Ports A, B, and C are shown in table 3-21.

3-44. 8155 TIMER PROGRAMMING

The 8155 Timer is a 14-bit down counter that counts the timer input pulses (1.2288 MHz) and provides an output of a square wave or pulse when terminal count (TC) is reached. The timer output (TINTO) is connected by a jumper to the RST 7.5 inut on the on-board 8085A CPU. By connecting the timer output to the RST 7.5 input of the CPU, the CPU can be interrupt driven at an interval desired for serial I/O communications. The count lengths required for various baud rates are given in table 3-21.

3-45. ADDRESSING. The 8155 Timer has two I/O addresses associated with it. One address (EC) is for the low order byte (least-significant bits of count

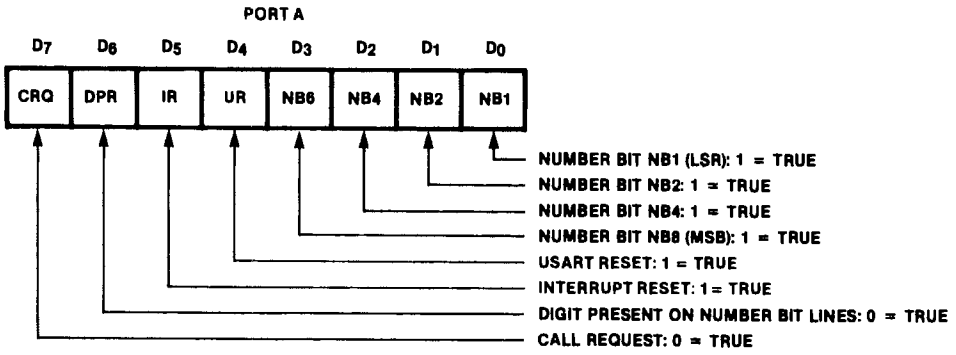


Figure 3-12. PPI Port A Bit Definitions

Table 3-20. Typical I/O Port Programming Routines

```
;AOUT OUTPUTS THE DATA IN REG C TO PORT A OF THE 8155.
;USES-C DESTROYS-A

            PUBLIC      AOUT

AOUT:       MOV         A,C         ;GET DATA
            OUT         0E8H
            RET

            END

;AIN READS PORT A OF THE 8155 INTO REG A.
;USES-NOTHING DESTROYS-A

            PUBLIC      AIN

AIN:        IN          0E8H
            RET

            END

;BIN READS PORT B OF THE 8155 INTO REG A.
;USES-NOTHING DESTROYS-A

            PUBLIC      BIN

BIN:        IN          0EAH
            RET

            END

;CIN READS PORT C OF THE 8155 INTO REG A.
;USES-NOTHING DESTROYS-A.

            PUBLIC      CIN
            EXTRN       BASAD

CIN:        IN          0EBH
            RET

            END
```

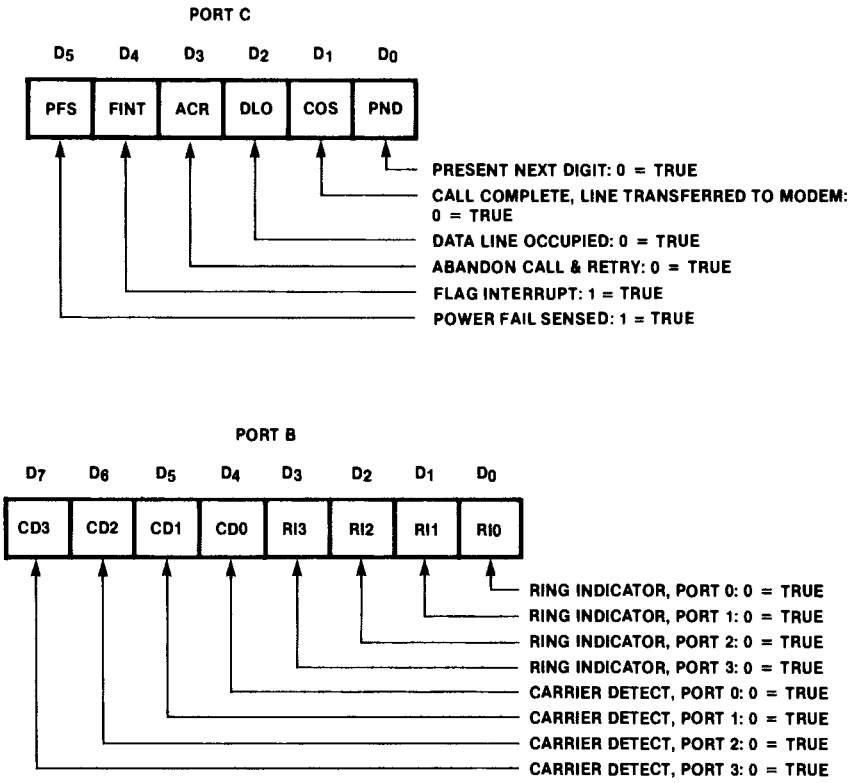


Figure 3-13. Port B and C Bit Definitions

length), and the other address (ED) is for the high order byte (most-significant bits of count length and timer mode). Figure 3-14 shows the timer format.

3-46. COUNT LENGTH REGISTER LOADING AND READING. The timers I/O addresses serve a dual purpose. During an I/O Write operation, the count length (bits 0-13) and mode (bits 14-15) are loaded into the 16 bit Count Length Register; during an I/O Read operation, the present count (the count at the time of the I/O Read operation) and the mode bits are read. To ensure that the correct count is read, it is preferable to stop counting, read the counter, and then reload the counter and continue counting.

Table 3-21. Baud Rates Vs Count Lengths

Baud Rate	Decimal Count
4800	256
2400	512
1200	1024
600	2048
110	11171
75	16384

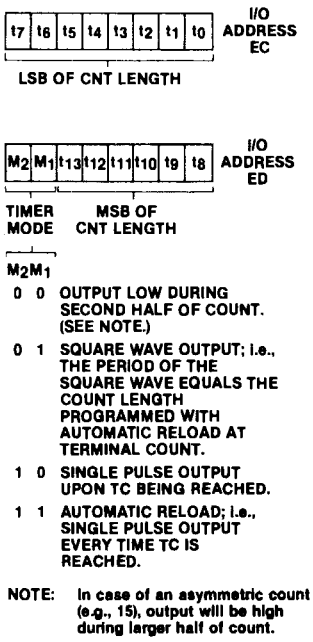


Figure 3-14. Timer Format

3-47. 8155 TIMER OPERATION. To program the timer, first stop the counter, then load the Count Length Register, one byte at a time, by performing I/O Write routines to the two timer addresses (EC and ED). Bits 0-13 will specify the length of the next count, and bits 14-15 will specify the timer output mode. There are two modes to choose from on the iSBC 544:

1. Single pulse upon TC being reached.

2. Repetitive single pulses each time TC is reached and automatic reload of counter when TC is reached, until instructed to stop by a new command loaded into the Command/Status Register.

Bits 6-7 of the Command/Status Register are used to start and stop the counter. See Figure 3-10 for a description of the different commands.

Table 3-22 shows a typical 8155 Timer load and count routine.

3-48. 8251A USART PROGRAMMING

Each of the four serial I/O ports is controlled by an Intel 8251A USART chip. The USART converts parallel output data into virtually any serial output data format (including IBM Bi-Sync) for half- or full-duplex operation. The USART also converts serial input data into parallel data format.

Prior to starting transmitting or receiving data, the USART must be loaded with a set of control words. These control words, which define the complete functional operation of the USART, must immediately follow a reset (internal or external) operation. The control words are either a Mode instruction or a Command instruction.

3-49. MODE INSTRUCTION FORMAT

The Mode instruction word defines the general characteristics of the USART and must follow a reset operation (internal or external). Once the Mode instruction word has been written into the USART, sync characters or command instructions may be inserted. The Mode instruction word defines the following:

a. For Sync Mode:
   (1) Character length
   (2) Parity enable
   (3) Even/odd parity generation and check
   (4) External sync detect (not supported by iSBC 544)
   (5) Single or double character sync

b. For Async Mode:
   (1) Baud rate factor (X1, X16, or X64)
   (2) Character length
   (3) Parity enable
   (4) Even/odd parity generation and check
   (5) Number of stop bits

Instruction word and data transmission formats for synchronous and asynchronous modes are shown in figures 3-15 through 3-18.

Table 3-22. Typical 8155 Timer Routine

```
MAIN PROGRAM - INITIALIZES THE 8155 COUNTER AND STARTS THE
COUNT BEFORE CONTINUING WITH ITS OTHER ROUTINES. THIS PROGRAM
SETS THE TIME TO COUNT 122 TIMER-IN PULSES BEFORE OUT-
PUTTING A TIMER PULSE THAT WILL GENERATE AN INTERRUPT.
USING MODE 3, THE TIMER WILL AUTOMATICALLY RELOAD
AND BEGIN ANOTHER COUNTDOWN.

MVI     A,7CH
OUT     0ECH        ; OUTPUT LSB OF COUNT LENGTH
MVI     A,0C0H
OUT     0EDH        ; OUTPUT MSB AND TIMER MODE
MVI     A,18H
SIM                 ; UNMASK RESTART INTERRUPTS
EI                  ; ENABLE PROCESSOR INTERRUPTS
MVI     A,0C0H
OUT     0E8H        ; START TIMER COUNTDOWN
  .
  .
  .
  .           MAIN PROGRAM CONTINUES
  .
  .

END MAIN PROGRAM
```

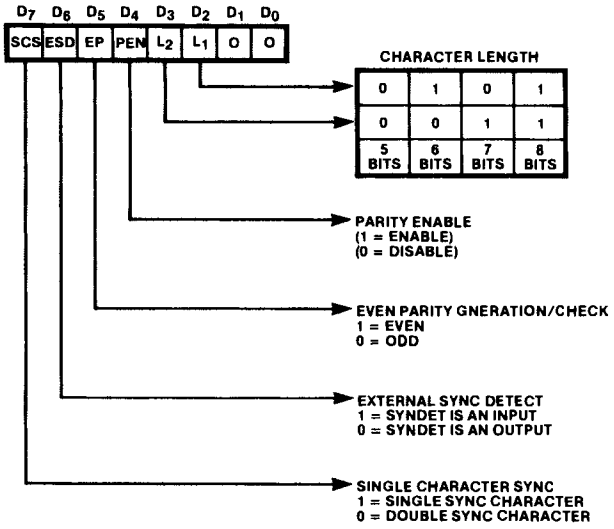


Figure 3-15. Synchronous Mode Instruction Word Format

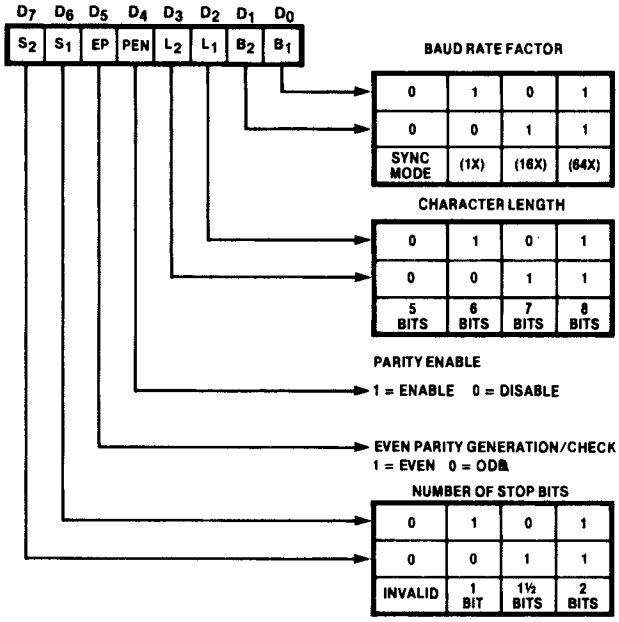


Figure 3-17. Asynchronous Mode Instruction Word Format

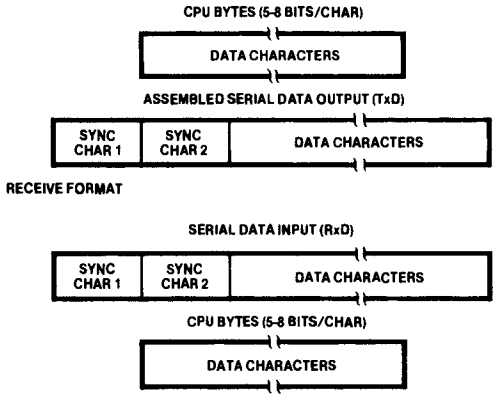


Figure 3-16. Synchronous Mode Transmission Format

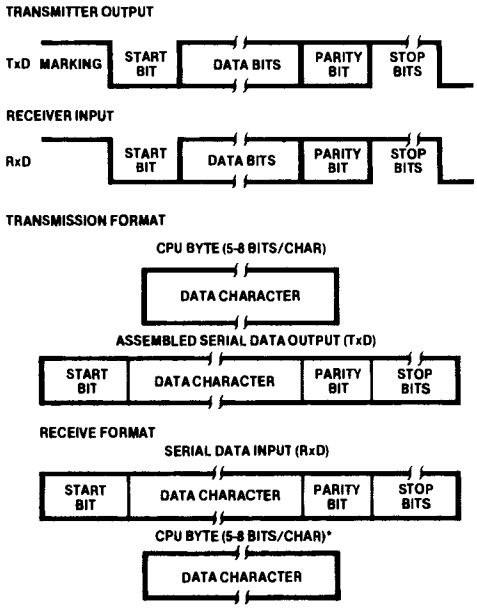


Figure 3-18. Asynchronous Mode Transmission Format

3-50. SYNC CHARACTERS

Sync characters are written to each USART in the synchronous mode only. The USART can be programmed for either one or two sync characters; the format of the sync characters is at the option of the programmer.

3-51. COMMAND INSTRUCTION FORMAT

The Command instruction word shown in figure 3-19 controls the operation of the addressed USART. A Command instruction must follow the mode and/or sync words. Once the Command instruction is written, data can be transmitted or received by the USART.

It is not necessary for a Command instruction to precede each data transaction; only those transmissions that require a change in the Command instruction. An example is a change in the transmit enable or receive enable bits. Command instructions can be written to the USART at any time after one or more data operations.

After initialization, always read the chip status and check for the TXRDY bit prior to writing either data or command words to the USART. This ensures that any prior input is not overwritten and lost. Note that issuing a Command instruction with bit 6 (IR) set will return the USART to the Mode instruction format.

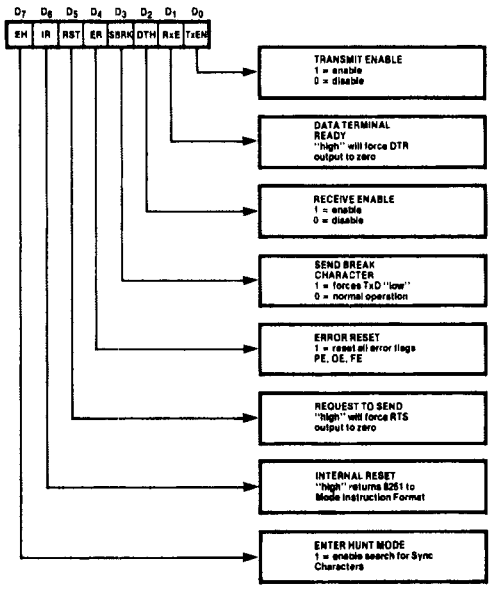


Figure 3-19. USART Command Instruction Word Format

3-52. RESET

To change the Mode instruction word, the USART must receive a Reset command. The next word written to the USART after a Reset command is assumed to be a Mode instruction. similarly, for sync mode, the next word after a Mode instruction is assumed to be one or two sync characters. All control words written into the USART after the Mode instruction (and/or the sync character) are assumed to be Command instruction.

Note that an individual USART can be reset by bit 6 (IR) in the Command word; all four USARTs can be reset by a board reset or by a system reset operation.

3-53. ADDRESSING

Each of the four USART chips uses two consecutive addresses. The lower of the two addresses is used to read and write I/O data; the upper address is used to write mode and command words and to read the USART status. (Refer to table 3-2.)

3-54. INITIALIZATION

A typical USART initialization and I/O data sequence is presented in figure 3-20. Each USART chip is initialized in four steps:

a. Reset USART to Mode instruction format.

b. Write Mode instruction word. One function of mode word is to specify synchronous or asynchronous operation.

c. If synchronous mode is selected, write one or two sync characters as required.

d. Write Command instruction word.

To avoid spurious interrupts during USART initialization, disable the corresponding USART interrupts. This can be done by either masking the appropriate interrupt request inputs at the PIC or by disabling the 8085A microprocessor interrupts by executing a DI instruction.

a. Individual USART Reset
   (1) Write Command instruction to desired USART (D1). Command instruction word must have bit 6 set (IR1); all other bits are immaterial.

NOTE

The reset procedure should be used only if the USART has been completely initialized, or the initialization procedure has reached the point that the USART is ready to receive a Command word. For example, if the reset

```
;RSTU PERFORMS A HARDWARE RESET ON THE FOUR USARTS
;USES-NOTHING DESTROYS-A

            PUBLIC      RSTU

RSTU:       IN          0E9H    ;READ CURRENT DATA ON PORT A
            ORI         10H     ;SET BIT 4 TO PERFORM RESET
            OUT         0E9H
            ANI         0EFH    ;RESET BIT 4
            OUT         0E9H
            RET

            END
```

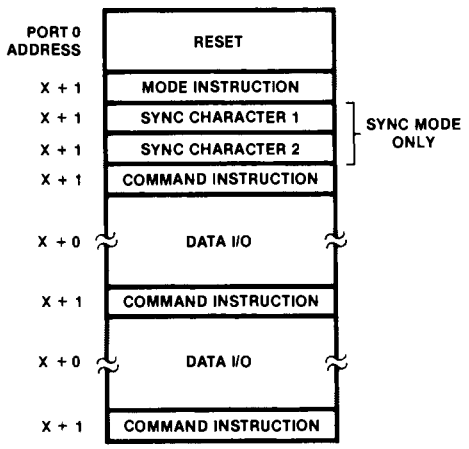
Table 3-23. Typical USART Mode or Command Instruction Subroutine

```
;CMD2 OUTPUTS CONTROL WORD TO USART 2 (PORT 2)
;USES-A, STAT2; DESTROYS-NOTHING

            PUBLIC  CMD2
            EXTRN   STAT2

CMD2:       PUSH    PSW
LP:         CALL    STAT2
            ANI     1           ;CHECK TXRDY
            JZ      LP          ;TXRDY MUST BE TRUE
            POP     PSW
            OUT     0D5H        ;ENTER HERE FOR INITIALIZATION
            RET

            END
```



*THE SECOND SYNC CHARACTER IS SKIPPED IF MODE INSTRUCTION HAS PROGRAMMED USART TO SINGLE CHARACTER INTERNAL SYNC MODE. BOTH SYNC CHARACTERS ARE SKIPPED IF MODE INSTRUCTION HAS PROGRAMMED USART TO ASYNC MODE.

Figure 3-20. Typical USART Initialization and Data I/O Sequence

command is written when the initialization sequence calls for a sync character, then subsequent programming will be in error.

(2) Repeat step (2) for remaining USART chips to be initialized. Addresses D3, D5, D7.

b. The following is a typical procedure used to reset all the USART chips:

First, reset the USART chips individually or reset all USART chips simultaneously by setting the resetting bit 4 of the 8155. At the top of the page is a typical reset routine.

Next, write a Mode instruction word to the desired USART. (See figures 3-15 through 3-18.) A typical subroutine for writing both Mode and Command instructions to USART 2 (Port 2) is given in table 3-23.

If the USART is programmed for the synchronous mode, write one or two sync characters depending on the transmission format.

Finally, write a Command instruction word to the desired USART. Refer to figure 3-19 and table 3-23.

3-55. OPERATION

Normal operating procedures use data I/O read and write, status read, and Command instruction write operations. Programming and addressing procedures for the above are summarized in following paragraphs.

Prior to any operating change, a new command word must be written with command bits changed as appropriate. (Refer to figure 3-19 and table 3-23.)

3-56. DATA INPUT/OUTPUT. For data receive or transmit operations perform a read or write, respectively, to the desired USART. Tables 3-24 and 3-25 show examples of typical character read and write subroutines for USART 1.

During normal transmit operation, each USART generates a Transmit Ready (TXRDY) signal that indicates that the USART is ready to accept a data character for transmission. TXRDY is automatically reset when the 8085A loads a character into the USART.

Similarly, during normal receive operation, each USART generates a Receive Ready (RXRDY) signal that indicates that a character has been received and is ready for input to the 8085A. RXRDY is automatically reset when a character is read by the 8085A.

The TXRDY and RXRDY outputs of each USART are connected to the Programmable Interrupt Controller chip (PIC) which resolves priority in case of simultaneous inputs and generates an interrupt for the 8085A. TXRDY and RXRDY are also available in the status word. (Refer to paragraph 3-50.)

3-57. STATUS READ. The 8085A can determine the status of a serial I/O port by issuing an I/O Read Command to the upper address of the appropriate USART chip. The format of the status word is shown in figure 3-21. A typical status read subroutine for Port 0 is given in table 3-26.

Table 3-24. Typical USART Data Character Read Subroutine

```
;RX1 READS DATA CHARACTER FROM USART 1 (PORT 1)
;USES-STAT1; DESTROYS-A, FLAGS

        PUBLIC  RX1,RXA1
        EXTRN   STAT1

RX1:    CALL    STAT1
        ANI     2               ;CHECK FOR RXRDY
        JZ      RX1
RXA1:   IN      0D2H            ;ENTER HERE IF RXRDY IS TRUE
        RET

        END
```

Table 3-25. Typical USART Data Character Write Subroutine

```
;TX1 WRITES DATA CHARACTER FROM REG A TO USART 1 (PORT 1)
;USES-STAT1; DESTROYS-A,FLAGS

        PUBLIC  TX1,TXA1
        EXTRN   STAT1

TX1:    PUSH    PSW             ;SAVE DATA
TX11:   CALL    STAT1
        ANI     1               ;CHECK FOR TXRDY
        JZ      TX11
        POP     PSW
TXA1:   OUT     0D2H            ;ENTER HERE IF TXRDY IS TRUE
        RET

        END
```

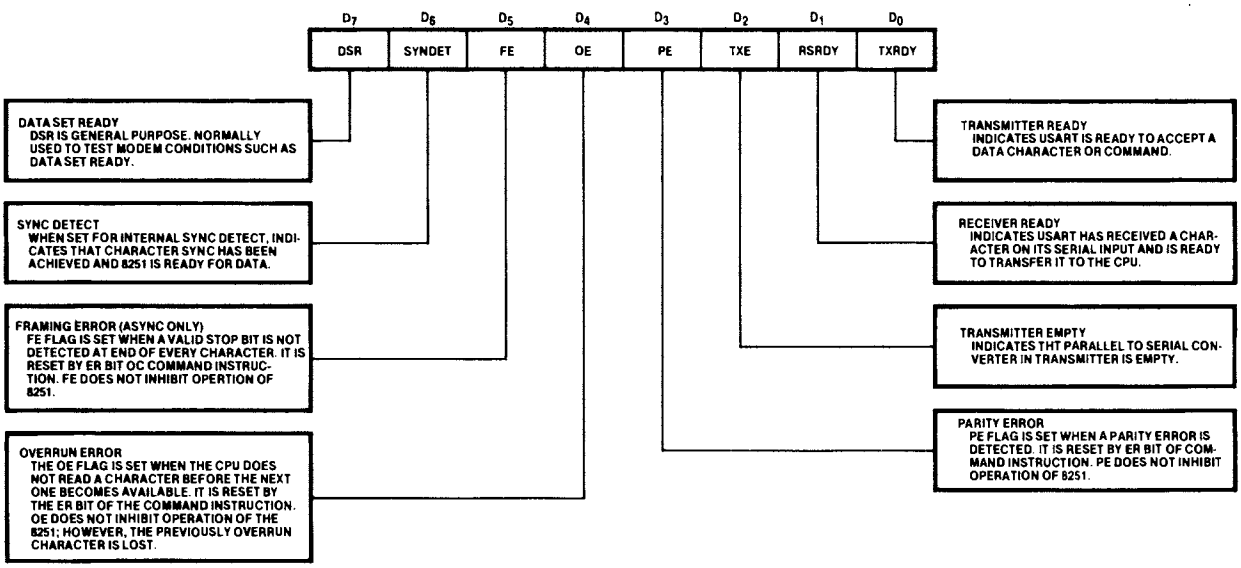


Figure 3-21. USART Status Read Format

Table 3-26. Typical USART Status Read Subroutine

```
;STAT0 READS STATUS FROM USART 0 (PORT 0)
;DESTROYS-A

            PUBLIC      STAT0

STAT0:
            IN          0D1H
            RET

            END
```

3-58. 8085A INTERRUPT HANDLING

Included in the 8085A CPU are five interrupt inputs: TRAP, RST 7.5, RST 6.5, RST 5.5, and INTR. The three "restart" interrupts (RST 7.5, RST 6.5, and RST 5.5) are maskable; the TRAP is also a "restart" interrupt, but is non-maskable.

The RST 7.5, RST 6.5, and RST 5.5 interrupts cause the internal execution of an RST instruction if the interrupts are enabled and if the interrupt mask is not set by a previously executed SIM instruction. The non-maskable TRAP interrupt causes the execution of an RST instruction independent of the state of the interrupt enable or interrupt mask. The priority and vector location of each of the restart interrupts are given in table 3-27.

On the iSBC 544, the interrupt inputs of the on-board 8085A CPU have been default connected as follows:

TRAP = not connected (Jumper available for PFIN/ – Power Fail Interrupt).

RST 7.5 = connected to the Timer outputs of the 8155 (TINT0 and Counter Five of the 8253 (TINT1).

RST 6.5 = connected to the serial port Ring Indicators (RINT) and the serial port Carrier Detect signals (PINT).

RST 5.5 = connected to Flag Interrupt (FINT) and optionally to one of the interrupt lines from the Multibus (INT0/ – INT 7/).

INTR = connected to the output of the 8259 PIC which monitors the interrupts from the 8251 USARTS.

Table 3-27. Interrupt Vector Memory Locations

Interrupt		Vector Location	Priority
TRAP		24	Highest
Timer	RST 7.5	3C	2nd
Ring Indicator	RST 6.5	34	3rd
Carrier Detect	RST 6.5	34	3rd
Flag Interrupt	RST 5.5	2C	4th
RXRDY0	INTR	40	Lowest
TXRDY0	INTR	48	Lowest
RXRDY1	INTR	50	Lowest
TXRDY1	INTR	58	Lowest
RXRDY2	INTR	60	Lowest
TXRDY2	INTR	68	Lowest
RXRDY3	INTR	70	Lowest
TXRDY3	INTR	78	Lowest

Note:
1. These are suggested addresses-specified during initialization of the 8259.

3-59. TRAP INTERRUPT

There are special considerations that must be made when the TRAP interrupt is used. The fact that the TRAP interrupt is non-maskable can present problems in at least two areas.

Interrupt driven systems often contain parameters that must be modified only within critical regions. A critical region can be roughly defined as a section of code that once begun must complete execution before it or another critical region that corresponds to the same system parameter(s) can be executed. A TRAP interrupt handler cannot safely alter such parameters either directly or indirectly by causing the execution of procedures or tasks that may alter such parameters.

If the hardware generates a TRAP interrupt on power up or power fail, the system must be able to process the TRAP interrupt before it is completely initialized. It should also take into account that an interrupt routine that runs with interrupts disabled can still be interrupted by a TRAP. Power Fail Sense status can be read by inputting bit 5 of Port C of the 8155.

Because of these considerations, it is recommended that the TRAP interrupt only be used for system start-up and/or catastrophic error handling such as a power failure.

It should be noted that TRAP does not destroy a previously established interrupt enable status. Executing the first RIM instruction following a TRAP interrupt yields the interrupt enable status as it was before the TRAP occurred. Following this first *mandatory* RIM instruction, subsequently executed RIM instructions provide current interrupt enable status.

3-60. RST 7.5, 6.5, AND 5.5 INPUTS

These interrupts can be individually masked by a SIM instruction and can thus be prevented from interrupting the processor. The priorities shown in table 3-27 do not take into account the priority of a routine that was started by a higher priority interrupt. An RST 6.5 interrupt can interrupt an RST 7.5 routine if the interrupts are *re-enabled* before the end of the RST 7.5 routine.

The RST 7.5 interrupt is *rising edge sensitive* and can be set by a pulse; the request once set by the pulse will be remembered until the request is serviced or reset by a SIM instruction. The RST 6.5 and 5.5 interrupts are *high level sensitive* and, in order to be recognized must remain high. Table 3-28 is a typical routine showing the RST 7.5 input being triggered by the 8155 Timer (TINT0).

Table 3-28. Typical RST 7.5 Interrupt Routine

```
FUNCTION: INTR75
DESCRIPTION: WHEN A RST 7.5 IS GENERATED BY THE 8155 (DETERMINED BY
             THE MAIN PROGRAM), THIS ROUTINE STORES THE STATUS WORD
             AND THEN INCREMENTS A COUNT VARIABLE, IT MAY THEN CALL
             A REAL TIME COUNT AND DISPLAY ROUTINE WHICH INCREMENTS
             VARIABLES FOR MILLISECONDS, SECONDS, ETC., AT THE PROPER TIME.
             FINALLY, THE INTERRUPT ROUTINE WILL RESET RST 7.5, RESTORE
             PROCESSOR STATUS WORD, AND ENABLE INTERRUPTS BEFORE RE-
             TURNING TO THE MAIN PROGRAM.

        ORG     003CH

        COUNT   EQU     00      ; SET TIMER DELAY

INTR75:
        CALL    SAVE            ; SAVE STATE OF MACHINE
        LXI     H,COUNT         ; LOAD H&L WITH LOCATION OF COUNT
        INR     M               ; INCREMENT THE COUNT
        CALL    RTC             ; CALL REAL TIME COUNT AND DISPLAY ROUTINE
        MVI     A,10H             (NOT SHOWN)
        SIM                     ; RESET RST 7.5
        CALL    RSTORE          ; RESTORE STATE OF MACHINE
        EI
        RET
```

Table 3-29. PINT and RINT Flop Reset Routine

```
;RSTINT PERFORMS A HARDWARE RESET ON THE RING INDICATOR AND
;CARRIER DETECT EDGE-SENSE FLIP-FLOPS
;USES-NOTHING DESTROYS-A

        PUBLIC  RSTINT

RSTINT: IN      0E9H            ; READ CURRENT DATA ON PORT A
        ORI     20H             ; SET BIT 5 TO PERFORM RESET
        OUT     0E9H
        ANI     0DFH            ; RESET BIT 5
        OUT     0E9H
        RET

        END
```

3-61. INTERRUPTS HANDLED BY RST 7.5, RST 6.5, and RST 5.5

The RST 7.5 input to the 8085A CPU, handles the two timer interrupts from the 8155 (TINT0) and from the 8253 (TINT1).

The RST 6.5 input to the 8085A CPU handles the Ring Indicator Interrupt (RINT), which is used to detect such things as a telephone ringing on one of the serial ports and the Carrier Detect Interrupt (PINT), which is used to detect the loss of a carrier signal on one of the serial I/O ports. If one of these interrupts occur, the 8155 PPI must be interrogated to determine which port caused the interrupt. The source of the interrupt is determined by reading Port B of the 8155. To determine if there is a second Carrier Detect interrupt pending, we would reset the RINT and PINT edge detect flip flops via the 8155, and then read Port B again. The RINT and PINT interrupts are reset by setting and resetting bit 5 of Port A of the 8155.

The RST 5.5 input to the 8085A CPU handles the Flag Interrupt (FINT) and the Multibus Interrupt lines. To determine which interrupt set RST 5.5, look at bit 7 of the accumulator (8085A) after executing a RIM instruction. This bit will be set by the SID line of the 8085A if a Multibus interrupt occurred.

The Flag Interrupt monitors the status of the RAM's base address. If an off-board write occurs into the base address, FINT will set. This can be used to indicate to the iSBC 544, that a command has been given it by one of the bus masters. The Flag Interrupt flop is reset by an on-board read of the RAM's base address. This Flag Interrupt provides a unique interrupt for each iSBC 544 on the system. The status of Flag Interrupt can be determined by inputting bit 4 of Port C of the 8155.

3-62. INTR INTERRUPT

The INTR interrupt from the 8259 PIC has the lowest priority and is sampled only during the last clock cycle of a given instruction. When INTR is active, the Program Counter (PC) is inhibited and three bytes, timed by INTA/ pulses, are transferred from the PIC to the 8085A.

a. Byte 1 = Call instruction (11001101)

b. Byte 2 = Lower byte or routine address.

c. Byte 3 = Upper byte of routine address.

The 16-bit routine address must be on a 32-byte boundary or a 64-byte boundary as determined by the Initialization Control Words previously written to the PIC. (Refer to paragraph 3-26.)

The INTR is enabled or disabled by the program. It is disabled by "RESET", and immediately after an interrupt is accepted.

3-63. 8085A INTERRUPT GENERATION

In addition to receiving interrupts from many sources, the 8085A is also capable of generating an interrupt and sending it out on the Multibus. The 8085A uses its SOD line to generate this interrupt.

The 8085A sets the SOD output by executing a SIM instruction with bit 6 and bit 7 of the accumulator set to 1's.

3-64. MASTER MODE

The iSBC 544 Intelligent Communications Controller can also be used as a bus master. In this role, it could be used to control a number of iSBC 534 Communication Expansion boards or other memory and I/O expansion boards.

When using the iSBC 544 as a bus master, there are some limiting factors such as:

a. The iSBC 544 can not operate on the Multibus with other masters, because it does not have the necessary bus contention logic.

b. A non-standard clock period.

c. An unsynchronized BUSY function that stays set as long as the 544 is in master mode, which locks out other bus masters.

The iSBC 544 does have an I/O read and I/O write line, and a memory read and memory write line. The limitations mentioned, must be considered before attempting to use the iSBC 544 as a controlling bus master.

The iSBC 544 can be put into the master mode in one of two ways:

a. Turn on the master mode switch (S1 position number 8) located at B6 or C6 on figure 5-1.

b. Program the iSBC 544 to operate in the master mode, by performing an I/O write command to I/O address E4H.

Both of these methods will set the master mode flop, allowing the iSBC 544 to operate as a bus master.

The iSBC 544 can be taken out of master mode (assuming the master mode switch is not set) by a system reset or by doing an I/O write to address E5.

CHAPTER 4
PRINCIPLES OF OPERATION

4-1. INTRODUCTION

This chapter provides a functional description and a circuit analysis of the iSBC 544 Intelligent Communications Controller. Figures 4-1 and 4-2, located at the end of the chapter, are simplified block diagrams that illustrate the functional interface between the on-board 8085A and its associated I/O devices and the on-board and system interface to the dynamic RAM located on the iSBC 544.

4-2. FUNCTIONAL DESCRIPTION

The following paragraphs give a brief description of the functional blocks that make up the iSBC 544. An operational circuit analysis is given starting with paragraph 4-14.

4-3. CLOCK CIRCUITS

The clock circuit for the iSBC 544 consists of crystal Y1 (22.1184 MHz), A15, A26, and A17. This circuit produces five frequencies, four (1.2288 MHz, 1.8432 MHz, 2.4576 MHz, and 22.1184 MHz) which can be used for all on-board clock timing except the 8085A CPU, and one (5.5296 MHz) which is used as an input to the 8085A CPU and that can be jumpered to the Multibus for use as Bus Clock (BCLK/) and Constant Clock (CCLK/).

NOTE

The 5.5296 MHz used for BCLK/ and CCLK/ is a non-standard frequency.

The clock circuit also generates a power up Reset signal to initialize the system to a known internal state. The Reset signal can also be generated by a input signal supplied via auxiliary connector P2.

4-4. 8085A CENTRAL PROCESSOR UNIT

The 8085A CPU, which is the heart of the iSBC 544, performs on-board processing functions and generates the addresses and control signals required to access memory and I/O devices. The CPU multiplexes the 8-bit data bus and the lower eight bits of the address bus. During the first part of the machine cycle, the lower eight address bits on the address/data bus are strobed into latch A65 by the Address Latch Enable (ALE) signal; the outputs of A65 are combined with the upper eight bits of the address to form the 16-bit address bus. During the remainder of the machine cycle; AD0-AD7 pins of the CPU are used for data input/outputs. A detailed description of the Intel 8085A can be found in the MCS-85 User's Manual (98-366C).

4-5. INTERVAL TIMER AND BAUD RATE GENERATORS

The Interval Timer and Baud Rate Generators consist of two Intel 8253 Programmable Interval Timer chips (A27, A28). Each of the PIT chips has three independent counters, each of which are separately programmable.

Four of the six independent counters (BDG0-BDG3) are connected by default jumpers to the four USART chips. The remaining two counters serve as auxiliary timers or rate generators. A jumper matrix (pins 32-34) can be used to connect the timers in series. The default jumper connection is from 32 to 33 which connects the output of Counter 4 (BDG4) to the clock input of Counter 5 (TINT1), thereby creating a long interval timer (approximately 1 hr). Counter 4 (BDG4) can also be used as a secondary baud rate clock for the USARTs.

In addition to the 8253 Timers, there is also a timer on the 8155 PPI. It is a 14 bit binary counter/timer that can be used as an interval timer by the on-board 8085A CPU.

4-6. SERIAL I/O PORTS

The iSBC 544 has four completely independent I/O Ports which are controlled by 8251A USARTs (A18-A21). Each port provides either full or half duplex communications with modems, data sets, or other serial devices. Each serial port converts parallel format data into serial format for transmission, or converts serial data into a parallel format for use by the system. The 8251A USART chips can support virtually any serial data technique currently in use, including IBM Bi-Sync.

4-7. PARALLEL I/O PORT

The parallel I/O port, which is controlled by an Intel 8155 Programmable Peripheral Interface chip (A22), is designed to support automatic calling units, such as the Bell Model 801, and certain other input or output signals that are not supported by serial I/O Ports

0 through 3. The 8155 can also be used as parallel I/O for controlling other RS232C devices. The 8155 PPI has two eight bit ports (PA and PB) and one six bit port (PC), which are programmed to operate in the basic I/O mode.

Ports B and C are used as input ports, and port A is used as an output port. Port C supports either an Automatic Calling Unit (ACU) or secondary receive data signals (SRXD0-3). Port B receives carrier detect (CD) and ring indicator (RI) signals for examination by the CPU. Port A supports either an Automatic Calling Unit (ACU) or secondary transmit data signals (STXD0-3). Port A bits 4 and 5 are used to reset the 8251A USARTs and the carrier detect and ring indicator interrupt flops respectively.

4-8. INTERRUPT CONTROL

The interrupt control portion of the iSBC 544 consists of an 8259 Peripheral Interrupt Controller (A29), the Flag Interrupt Flop (A30), the Carrier Detect and Ring Indicator Interrupt Flops (A23, A24, and A44) and the five interrupt inputs on the 8085A CPU. The priority of the interrupts on the iSBC 544 has been predetermined by the way that they are default connected to the 8085A. A detailed explanation of how the 8085A handles each one of these interrupts may be found in paragraph 3-58.

4-9. PROM CONFIGURATION

The ROM/PROM on the iSBC 544 consists of either 4K or 8K bytes. Two sockets A35 and A51 are provided for user installation of the PROM chips. Jumpers are provided to accommodate the various ROM/PROM chips. Address block 0000-1FFF is reserved for ROM/PROM use only.

4-10. RAM CONFIGURATION

The iSBC 544 includes 16K of dynamic RAM implemented with eight Intel 2117 chips and an Intel 8202 Dynamic RAM Controller (A62). Dual-port control logic allows this RAM to be accessed by either the on-board 8085A, or by another master CPU. The dual-port logic provides a reservation scheme for the iSBC 544's RAM. In its normal default state, the dual-port logic has selected the on-board processor to have memory. If the off-board master CPU wants to access memory, it must cause the dual-port logic to set up in a way that will reserve memory for it.

The RAM decode logic allows for extended Multibus addressing of up to 20 address lines. This allows bus masters with 20-bit addressing capability to partition the iSBC 544 into any 4K, 8K, or 16K segment in a 1-megabyte address space. The on-board 8085A CPU, however, has only a 16-bit address capability, which limits it to a 64K address space.

The iSBC 544 employs a network made up of a switch (S1) and a PROM decoder (A41) to determine if the Multibus address is a valid address for this board, and to transform it into an address within the range of the on-board RAM (8000H - BFFFH), for use by the Flag Interrupt logic. The iSBC 544 also has 256 bytes of static RAM located on the Intel 8155 PPI (A22). This memory is only accessible to the on-board 8085A. The address block for the static RAM is 7F00-7FFF.

4-11. BUS INTERFACE

The iSBC 544's Multibus interface consists of bi-directional address bus drivers (A73, A74, A75), bi-directional data bus drivers (A76, A77), memory read/write lines, and I/O read/write lines. The iSBC 544 has no Multibus control logic, because when it operates as a master, it can be the only master on the Multibus.

4-12. DUAL PORT CONTROL

The dual port control logic allows the iSBC 544 to function as a slave RAM device when operating in the "intelligent slave mode." The dual port logic allows access to the on-board RAM by the on-board 8085A or by some Multibus master. The dual port logic reserves the memory for the on-board 8085A if a request is imminent.

4-13. MASTER MODE

The iSBC 544 has two modes of operation; intelligent slave, and master mode. Its normal mode is the intelligent slave mode. The master mode flop (A44) can be set by a switch (S1) or by an I/O write command to address E4. Once master mode has been set, the iSBC 544 will perform as a bus master. It should be noted, however, that the iSBC cannot function as a dual master because it does not contain the Multibus control logic. The iSBC 544 could be used to control a number of slave iSBC 534 Communication Expansion boards.

The iSBC 544 prevents any use of the Multibus by other masters, by keeping the bus busy as long as it is in Master Mode.

4-14. CIRCUIT ANALYSIS

Figure 5-2 is a schematic diagram of the iSBC 544. The schematic diagram consists of 9 sheets, each of

which includes grid coordinates. Signals that traverse from one sheet to another are assigned grid coordinates at both the signal source and signal destination. For example, the grid coordinates 5ZD1 locate a signal source (or signal destination) on sheet 5 zone D1.

Both active-high and active-low signals are used. A signal mnemonic that ends with a virgule (e.g. DAT7/) denotes that the signal is active low (≤0.8V). Conversely, a signal mnemonic without a virgule (e.g. ALE) denotes that the signal is active high (≥2.0V).

4-15. INITIALIZATION.

When power is applied in a start-up sequence, the contents of the 8085A CPU program counter, instruction register, and interrupt enable flip-flop are subject to random factors and cannot be predicted. For this reason, a power up sequence is used to set the CPU, interrupt flops, and I/O ports to a known internal state.

When power is initially applied to the iSBC 544, capacitor C3 (7ZC7) begins to charge through resistor R1. The charge developed across C3 is sensed by a Schmitt trigger, which is internal to Clock Generator A15. The Schmitt trigger converts the slow transition at pin 2 into a clean, fast rising synchronized PURST output of 3 seconds at pin 1. The PURST signal is inverted by A72 and A54 (2ZD7) to produce the Initialize signal INIT/. The INIT/ signal clears the 8085A CPU program counter, instruction register, and interrupt enable flip-flop; resets the parallel I/O port to the input mode; resets the serial I/O ports to the idle mode via the 8155; resets the Dual Port logic (3ZB8); resets the Flag Interrupt flop (5ZA1); resets the Master Mode flop (6ZA8); and resets the PINT and RINT Interrupt flops (6ZC5). The INIT/ signal can also be used as a system reset by way of jumper 71-70 to the Multibus.

The initialization sequence described above can be triggered by an AUX RESET signal from connector P2 (7ZC7), or from INIT/ on the Multibus.

4-16. CLOCK CIRCUITS (Sheet 7)

All the on-board clock signals are generated by the 8224 Clock Generator (A15). The 8224 produces two frequencies 22.1184 MHz, and 2.4576 MHz. The 2.4576 MHz signal is divided by A26 to produce the iSBC 534 compatible frequency of 1.2288 MHz which is the default frequency used for the 8253 PIT's, and for the 8155 Timer. The 2.4576 MHz signal (ϕ2TTL) is also used for the internal device timing of the 8251 USARTs.

The 22.1184 MHz frequency is divided by four and divided by twelve (A26 ad A17) to produce frequencies of 5.5296 MHz and 1.8432 MHz. The 5.5296 MHz frequency is used as the input frequency for the 8085A CPU and can also be used for BCLK/ and CCLK/ by jumpering the 5.5296 MHz signal to the Multibus. It should be noted that BCLK/ and CCLK/ in this case would not be standard clock frequencies. The 1.8432 MHz frequency is used as an optional jumper selectable input for the 8253 PITs and for the 8155 Timer.

4-17. 8085A CPU TIMING

The 8085A CPU internally divides the 5.5296 MHz clock input by two to develop the timing requirements for the various time dependent functions. These functions are described in the following paragraphs.

4-18. INSTRUCTION TIMING. The execution of any program consists of read and write operations, where each operation transfers one byte of data between the CPU and memory or between the CPU and an I/O device. Although the CPU can vary the address, data, type, and sequence of operations, it is capable of performing only a basic read or write operation. With the exception of a few control lines, such as Address Latch Enable (ALE), these read and write operations are the only communication necessary between the CPU and the other components to execute any instruction.

An *instruction cycle* is the time required to fetch and execute an instruction. During the fetch phase, the selected instruction (consisting of up to three bytes) is read from memory and stored in the operating registers of the CPU. During the execution phase, the instruction is decoded by the CPU and translated into specific processing activities.

Each instruction cycle consists of up to five machine cycles. A *machine cycle* is required each time the CPU accesses memory or an I/O device. The fetch phase requires one machine cycle for each byte to be fetched. Some instructions do not require any machine cycles other than those necessary to fetch the instructions from memory; other instructions, however, require an additional machine cycle(s) to write or read data to or from memory or I/O devices.

Every instruction cycle has at least one reference to memory during which time the instruction is fetched. An instruction cycle must always have a fetch, even if the execution of that instruction requires no reference to memory. The first machine cycle in every instruction cycle is therefore a fetch, and beyond that there are no specific rules. For instance, the IN (in-

put) and OUT (output) instructions each require three machine cycles: fetch (to obtain the instruction), memory read (to obtain the I/O address of the device), and an input or output machine cycle (to complete the transfer).

Each machine cycle consists of a minimum of three and a maximum of six states designated T_1 through T_6. A *state* is the smallest unit of processing activity and is defined as the interval between two successive falling edges of the CPU clock. Each state (or CPU clock cycle) has a duration of 362 nanoseconds (derived by dividing the 5.5296 MHz frequency by two).

Every machine cycle normally consists of three T-states with the exception of an opcode fetch, which consists of either four or six T-states. The actual number of states required to execute any instruction depends on the instruction being executed, the particular machine cycle within the instruction cycle, and the number of *wait* states inserted into the machine cycle. The wait state is initiated when the READY input to the CPU is pulsed low.

There are no wait states imposed when the CPU is addressing on-board PROM or on-board I/O, or static RAM. There will be wait states imposed, however, in the following operations:

a. When addressing on-board RAM and a refresh cycle is in progress.

b. When addressing on-board RAM and a bus master is currently reading or writing on-board RAM.

c. When operating in the master mode and addressing off-board I/O or memory.

d When writing to the dynamic RAM.

Figure 4-3 is presented to show the relationship between an instruction cycle, machine cycle, and T-state. This example shows the execution of a Store Accumulator Direct (STA) instruction involving on-board memory. Notice that for this instruction the opcode fetch (machine cycle M_1) requires four T-states and the remaining three cycles each require three T-states.

The opcode fetch is the only machine cycle that requires more than three T-states. This is because the CPU must interpret the requirements of the opcode fetched during T_1 through T_3 before it can decide what must be done in the remaining T-state(s).

There are seven types of machine cycles, each of which can be differentiated by the states of three CPU status lines (IO/$\overline{M}$, S0, and S1) and three CPU control lines ($\overline{RD}$, $\overline{WR}$, and $\overline{INTA}$). Table 4-1 lists the states of the CPU status and control lines during each of the seven machine cycles.

Table 4-1. CPU Status and Control Lines

Machine Cycle	Status			Control		
	IO/M	S0	S1	$\overline{RD}$	$\overline{WR}$	$\overline{INTA}$
Opcode Fetch	0	1	1	0	1	1
Memory Read	0	0	1	0	1	1
Memory Write	0	1	0	1	0	1
I/O Read	1	0	1	0	1	1
I/O Write	1	1	0	1	0	1
INTR Acknowledge	1	1	1	1	1	0
Bus Idle	X	X	X	1	1	1
Halt	TS	0	0	TS	TS	1

4-19. OPCODE FETCH TIMING. Figure 4-4 shows the timing relationship of a typical opcode fetch machine cycle. At the beginning of T_1 of *every* machine cycle, the CPU performs the following:

a. Pulls IO/$\overline{M}$ low to signify that the machine cycle is a memory reference operation.

b. Drives status lines S0 and S1 high to identify the machine cycle as an opcode fetch.

c. Places high-order bits (PCH) of program counter onto address lines A8-A15. These address bits will remain true until at least T_4.

d. Places low-order bits (PCL) of program counter onto address/data lines AD0-AD7. These address bits will remain true for only one clock cycle, after which AD0-AD7 go to their high- impedance state as indicated by the dashed line in figure 4-3.

e. Activates the Address Latch Enable (ALE) signal.

At the beginning of T_2, the CPU pulls the RD/ line low to enable the addressed memory device. The device will then drive the AD0-AD7 lines. After a period of time, as determined by the access time of the addressed memory device, valid data (the DCX instruction in this example) will be present on the D0-D7 lines. During T_3 the CPU loads the data on the D0-D7 lines into its instruction register and drives RD/ high, disabling the addressed memory device. During T_4 the CPU decodes the opcode and decides whether or not to enter T_5 on the next clock cycle or start a new machine cycle and enter T_1. In the case of the DCX instruction, the CPU will enter T_5 and then T_6 before beginning a new machine cycle.

Figure 4-5 is identical to figure 4-4 with one exception, which is the use of the READY input to the CPU. As shown in figure 4-5; the CPU examines the state of the READY input during T_2. If the READY input is high, the CPU will proceed to T_3 as shown in figure 4-4. If the READY input is low, however, the CPU will enter the T_{wait} state and stay there until READY goes high. When READY goes high, the CPU will exit the T_{wait} state and enter T_3. The external effect of using the READY input is to preserve

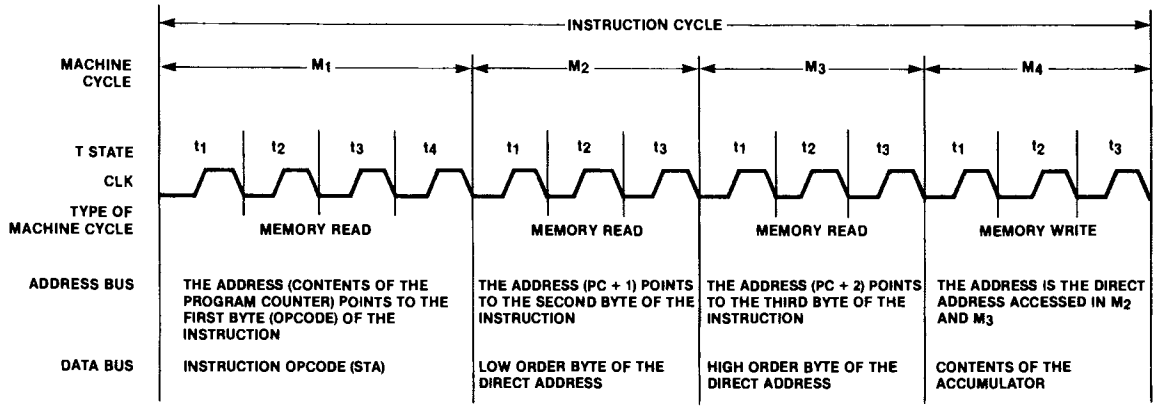


Figure 4-3. Typical CPU Instruction Cycle

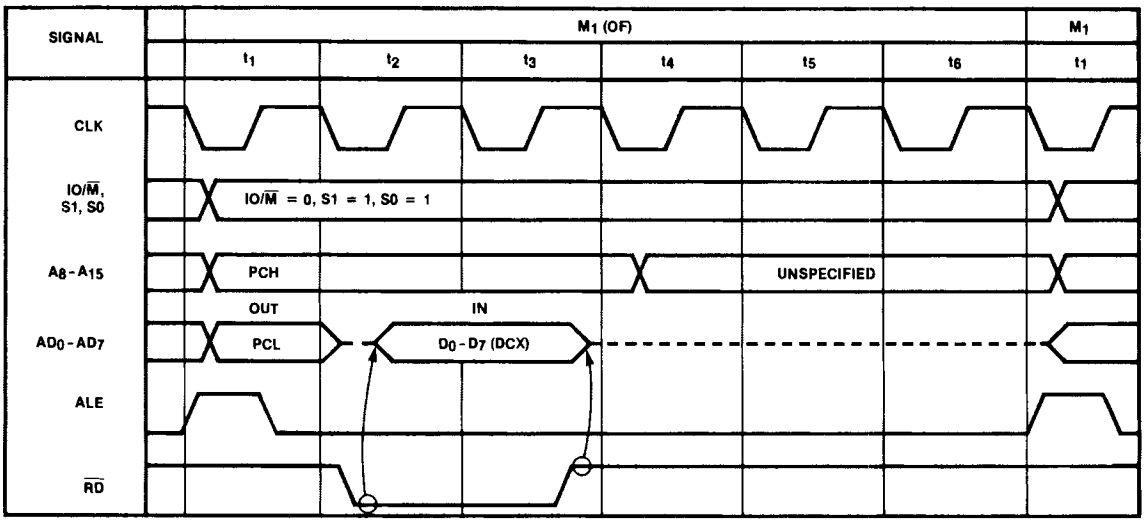


Figure 4-4. Opcode Fetch Machine Cycle (No Wait)

the exact state of the CPU signals at the end of T_3 for an integral number of clock periods before finishing the machine cycle. This stretching of the system timing, in effect, increases the allowable access time for memory or I/O devices. By inserting T_{wait} states, the CPU can accommodate slower memory or slower I/O devices. It should be noted, however, that access to the on-board PROM and I/O ports does not impose a T_{wait} state. However as mentioned previously, T_{wait} states are imposed in certain instances when accessing on-board RAM. T_{wait} states are also imposed when the iSBC 544 is in the master mode and is accessing off-board memory or I/O.

4-20. MEMORY READ TIMING. Figure 4-6 shows the timing of two successive memory read machine cycles, the first without a T_{wait} state and the second with a one T_{wait} state. Disregarding the states of the S0 and S1 lines, the timing during T_1 through T_3 is identical with the opcode fetch machine cycle shown in figure 4-2. The major difference between the opcode fetch and memory read cycles is that an opcode fetch machine cycle requires four or six T-states whereas the memory-read machine cycle requires only three T-states. One minor difference between the cycles is that the memory address used for the opcode fetch cycle is always the contents of the program counter (PC), which points to the current instruction; the address used for a memory read cycle can be one of several origins. Also, the data read from memory is placed into the appropriate register instead of the instruction register. Note that a T_{wait} state is not imposed during a read of on-board PROM.

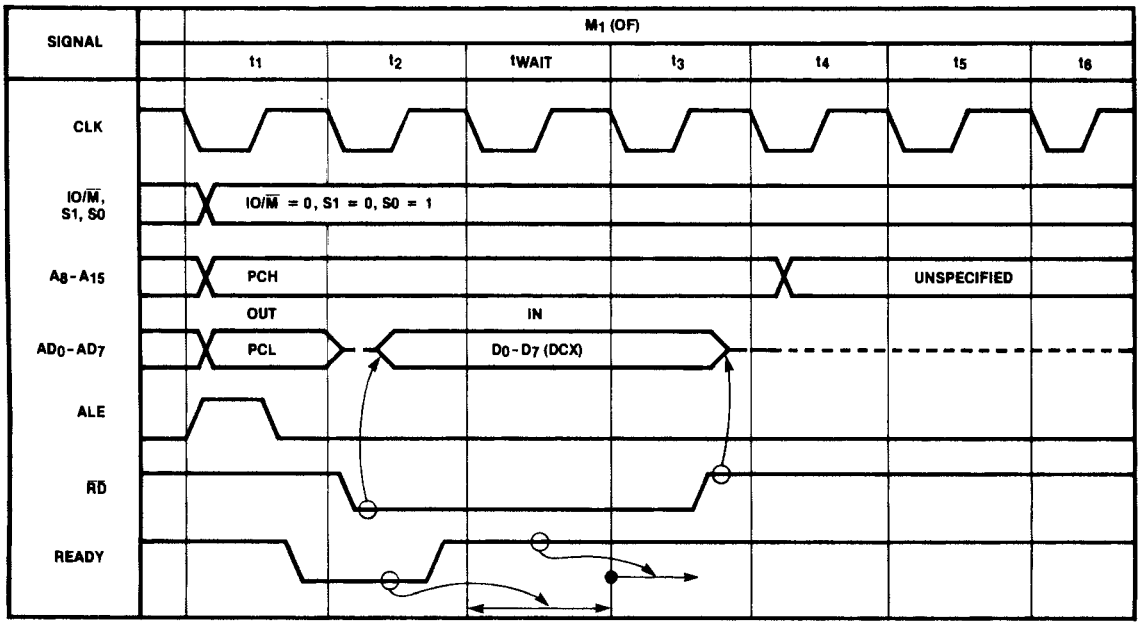


Figure 4-5. Opcode Fetch Machine Cycle (With Wait)

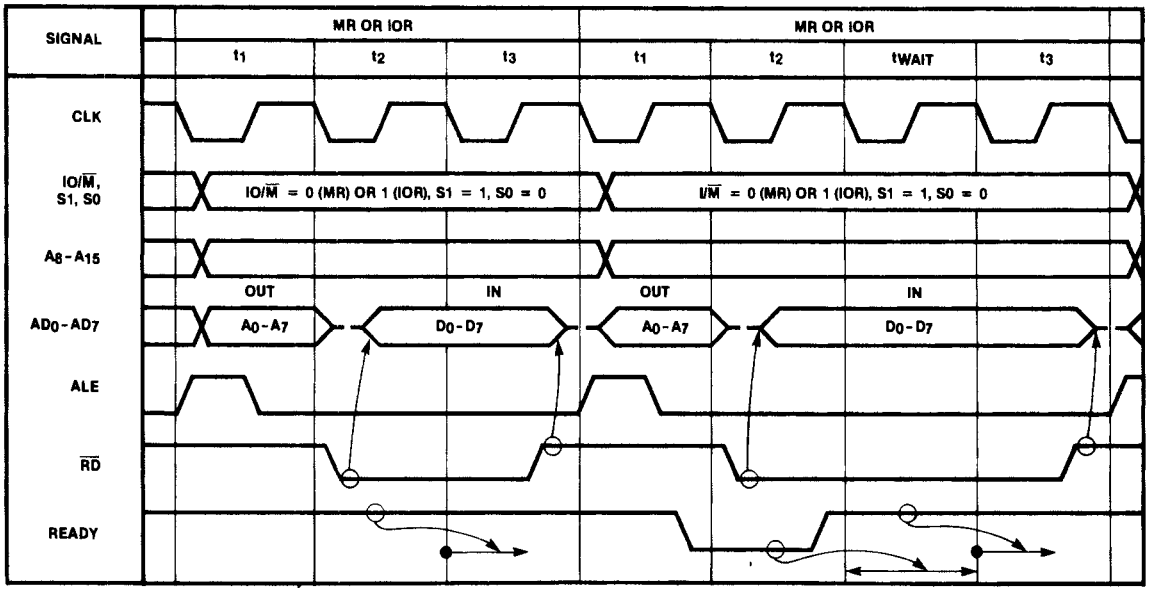


Figure 4-6. Memory Read (or I/O Read) Machine Cycle

4-21. I/O READ TIMING. Figure 4-6 also illustrates the timing of two successive I/O read machine cycles, the first without a T_{wait} state and the second with one T_{wait} state. With the exception of the IO/$\overline{M}$ status signal, the timing of a memory read cycle and an I/O read cycle is identical. For an I/O read, IO/$\overline{M}$ is driven high to identify that the current machine cycle is referencing an I/O port. One other minor exception is that the address used for an I/O read cycle is derived from the second byte of an IN instruction; this address is duplicated onto both the A8-A15 and AD0-AD7 lines. The data read from the I/O port, specified by the IN instruction, is always placed in the accumulator. Note that a T_{wait}

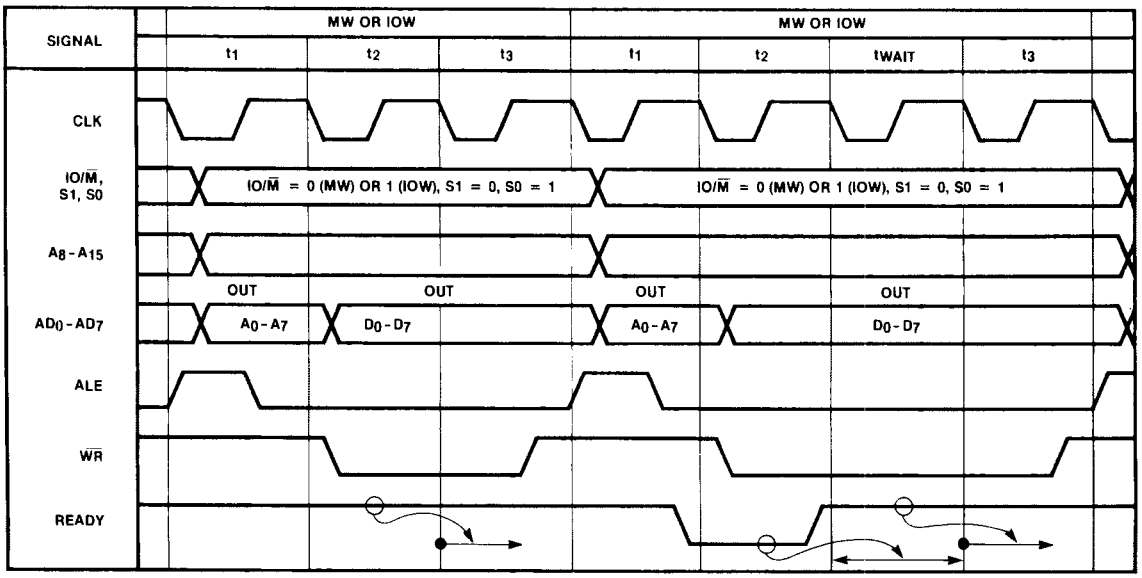


Figure 4-7. Memory Write (or I/O Write) Machine Cycles

is not imposed during the access of on-board I/O devices; T_{wait} states are imposed during the access of system I/O device via the Multibus.

4-22. MEMORY WRITE TIMING. Figure 4-7 shows the timing of two successive memory write machine cycles, the first without a T_{wait} state. Again, disregarding the states of the S0 and S1 lines, the timing during T_1 is identical to the timing of an opcode fetch, memory read, and I/O read cycles. The difference occurs, however, at the end of T_1. For instance, in a memory read cycle the AD0-AD7 lines are disabled (high impedance) at the beginning of T_2 in anticipation of the returned data. In a memory write cycle, the AD0-AD7 lines are not disabled and the data to be written into memory is placed on these lines at the beginning of T_2. The Write (WR/) line is driven low at this time to enable the addressed memory device. During T_2 the READY input is checked to determine if a T_{wait} state is required. If the READY input is low, T_{wait} states are inserted until READY goes high. During T_3, the WR/ line is driven high to disable the addressed memory device and terminate the memory write operation. Note that the contents on the address and data lines do not change until the next T_1 state.

4-23. I/O WRITE TIMING. Figure 4-7 also illustrates the timing of the two successive I/O write machine cycles, the first without a T_{wait} state and the second with one T_{wait}. With the exception of the IO/$\overline{M}$ status signal, the timing of a memory write cycle and an I/O write cycle are identical.

4-24. INTERRUPT ACKNOWLEDGE TIMING. Figure 4-8 shows the CPU timing in response to the INTR input being driven high by the 8259 PIC. It is assumed that the CPU interrupt enable flip- flop has been set by a previously executed Enable Interrupt instruction. The status of the TRAP, RST 7.5, RST 6.5, RST 5.5, and INTR inputs are sampled during CLK of the T-state immediately preceding T_1 of the M1 machine cycle. If INTR was the only valid interrupt, the CPU would clear its interrupt enable flip-flop and enter the Interrupt Acknowledge (INA) machine cycle. With two exceptions, the INA machine cycle is identical to the Opcode Fetch (OF) machine cycle. The first exception is that IO/$\overline{M}$ = 1, which signifies that the opcode fetched will be from an I/O device. The second exception is that $\overline{INTA}$ is asserted instead of $\overline{RD}$. Although the contents of the CPU program counter is sent out on the address lines, the address lines are ignored.

When INTA is asserted, the PIC provides a CALL instruction which causes the CPU to push the contents of the program counter onto the stack before jumping to a new location. After receiving the CALL opcode, the CPU performs a second INTA machine cycle (M_2) to access the second byte of the CALL instruction from the PIC. The timing of M_2 is identical with M_1 except that M_2 has three T-states. M_2 is followed by M_3 to access the third byte of the CALL instruction. When all three bytes have been received, the CPU executes the instruction. The CPU inhibits the incrementing of the program counter during the three INTA cycles so that the correct program counter value can be pushed onto the stack during M_4 and M_5.

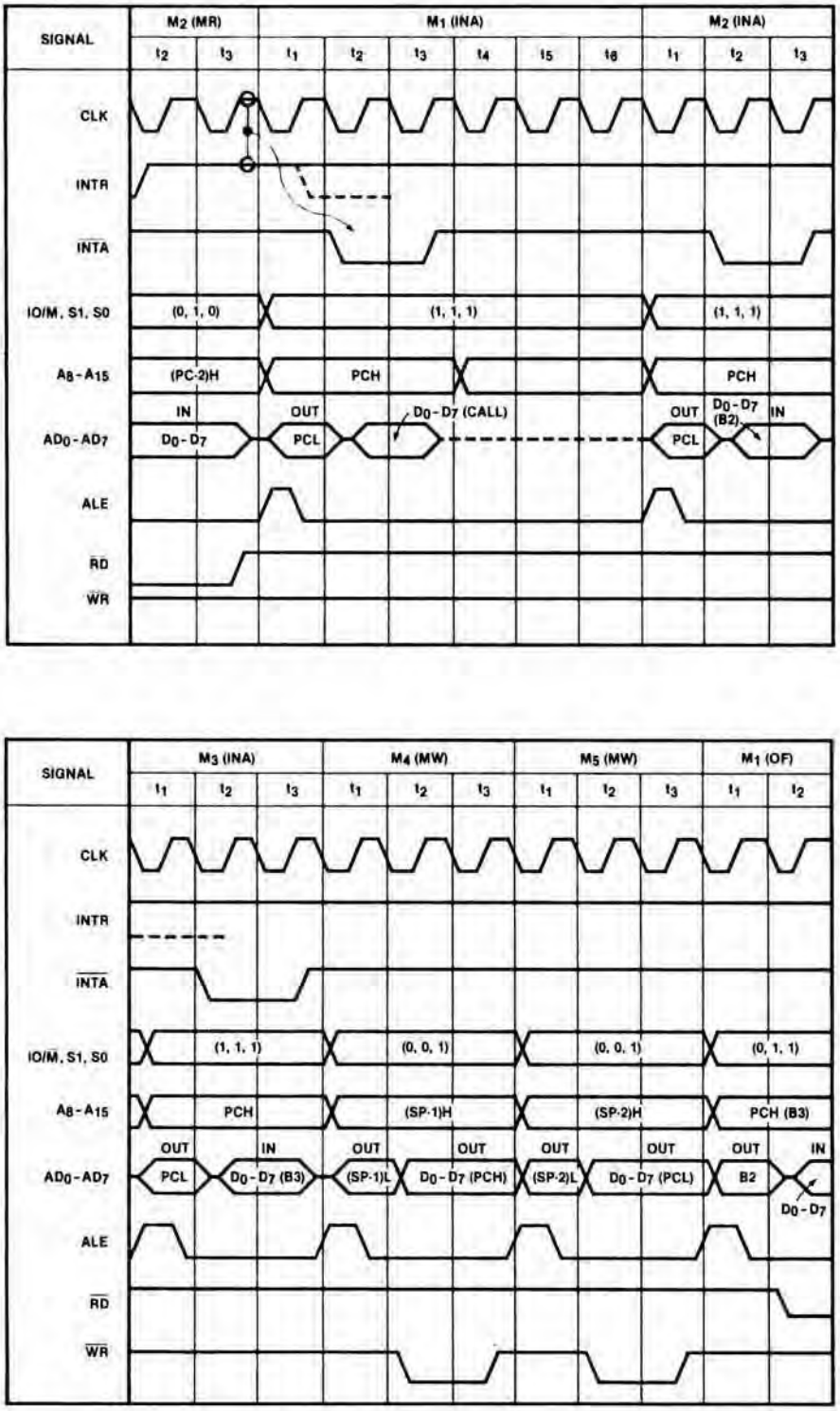


Figure 4-8. Interrupt Acknowledge Machine Cycles

During M_4 and M_5, the CPU performs Memory Write (MW) machine cycles to write (push) the contents of the program counter onto the top of the stack. The CPU then places the two bytes accessed during M_2 and M_3 into the upper and lower bytes of the program counter. This has the same effect as jumping the execution of the program to the location specified by the CALL instruction.

After the interrupt service routine is executed, the CPU sets the interrupt enable flip-flop, pops the stack and loads it into the program counter, and resumes system operation at the point of interrupt.

4-25. ADDRESS BUS (SHEET 2)

Figure 4-9 is a block diagram of the iSBC 544 Address Bus. The iSBC address bus is also shown in figures 4-1 and 4-2 as a weighted line. The upper eight bits of the address bus are fed by A8-A15 of the 8085A. The lower eight bits of the address bus are fed by AD0-AD7 (8085A Data Bus) through latch A65. AD0-AD7 are used during T_1 of a machine cycle to output the lower eight bits of an address. During T_2 and T_3 of the machine cycle AD0-AD7 input or output data. The 16 address lines PAB8-PABF from the 8085A and PAB0-PAB7 from latch A65 are distributed as follows:

a. PAB0-PABB to the PROM chips A35 and A51 (4ZD3).

b. PAB8-PABF to IO/M Address Decode PROM (A50) for chip select and buffer control logic (4ZB3).

c. PAB0-PABF to RAM Address Buffers (A63 and A64), RAD0-RADF from the RAM Address Buffers to the RAM Controller A62 (5ZD6).

4-26. DATA BUS (SHEET 2)

Figure 4-10 is a block diagram of the data bus. The data bus (AD0-AD7) is used for outputting and inputting data to/from the 8085A CPU. The data bus is distributed as follows:

a. PDB0-PDB7 to/from RAM Transmitter/Receiver A66 (5ZD2).

b. PDB0-PDB7 to/from 8155 PPI A22 (6ZA8).

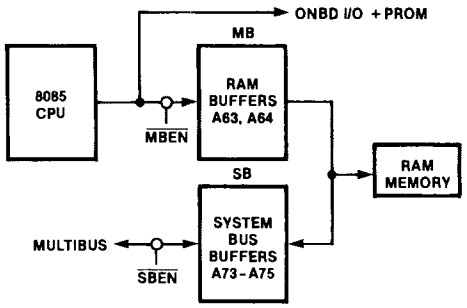
c. PDB0-PDB7 to/from I/O Transmitter/Receiver A67 (2ZC2).

d. IODB0-IODB7 to/from PROM chips A35 and A51 (4ZD1).

e. IODB0-IODB7 to/from 8259 PIC A29 (7ZB8).

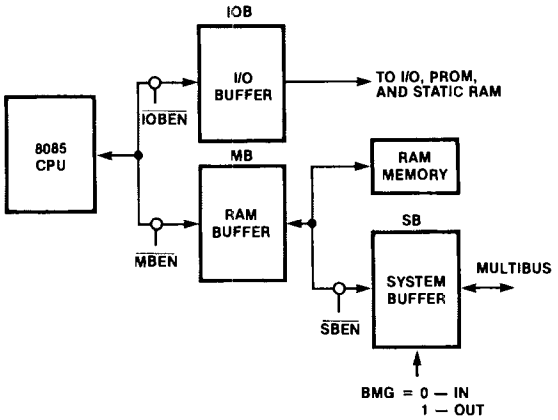
f. IODB0-IODB7 to/from 8251A USARTS A18 and A19 (8ZB8) and A20, A21 (9ZB8).

g. IODB0-IODB7 to/from 8253 PITs A27 and A28 (7ZC2).



$\overline{MBEN} = \overline{(ONBD\ I/O + PROM) \cdot (SEL\ ONBD)}$
$\overline{SBEN} = \overline{SELECT\ ONBD + BUS\ MASTER\ GATE\ (BMG)}$

Figure 4-9. Address Bus and Buffers



$\overline{IOBEN} = \overline{IOBCS \cdot CMD + INTA}$
$\overline{MBEN} = \overline{ONBD\ I/O + PROM \cdot SEL\ ONBD}$
$\overline{SBEN} = \overline{SELECT\ ONBD + BUS\ MASTER\ GATE\ (BMG)}$

Figure 4-10. Data Bus and Buffers

4-27. READ/WRITE COMMAND GENERATION

Figure 4-11 is a block diagram of the Command and Acknowledge logic. The Read/Write Command logic is divided into memory commands and I/O commands. All of these commands are derived from the 8085A's $\overline{RD}$ and $\overline{WR}$ outputs, its IO/$\overline{M}$ output, and from the status lines S0 and S1 shown in table 4-1. These signals are used, so that the on-board 8085A can avoid wait states, which would occur if the commands were not decoded.

4-28. I/O COMMANDS (SHEET 2). The iSBC 544's I/O signals, IOR/, IOW/ and IOCMD (2ZB2) are derived by gating the CPU IO/$\overline{M}$, RD/ and WR/ outputs. The I/O Command (IOCMD) is used by the

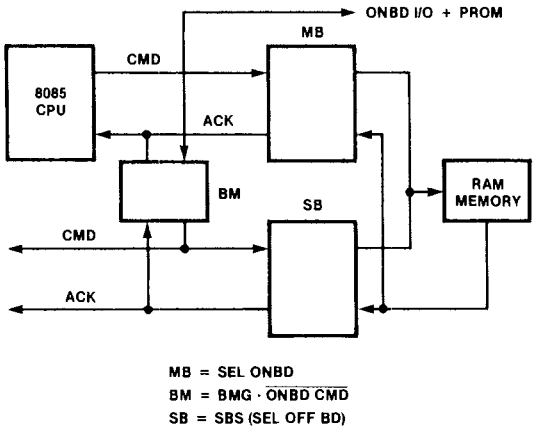


Figure 4-11. Command and Acknowledge Logic

IO/M Decode PROM for Chip select and for I/O bus control (4ZC8), and by the 8155 PPI to select the I/O portion of the chip (6ZB8).

The I/O Read Command (IOR/) is used to generate an off-board I/O Read in the master mode, and by the 8259 PIC, 8251A USARTs and 8253 PITs to perform an I/O read.

The I/O Write Command (IOW/) is used by the 8259, and 8253 PITs, and the 8251A USARTs to perform an I/O write, and is used to control the Master Mode flop if it is an I/O write to address E4 or E5 (6ZA6).

4-29. MEMORY COMMANDS (SHEET 2). The memory command signals on the iSBC 544 consist of, RD/ and WR/ directly from the 8085A, Memory Write (MW/), Advanced Memory Read (AMR), and Advance Command (ACMD/).

The RD/ and WR/ signals are used by the 8155 PPI. The RD/ signal is also used as a chip select on PROM chips A35 and A51. The MW/ signal is used to write data into the dynamic RAM.

The Advance Commands are generated because of timing restraints. In order for the 8085A to continue in its cycles when it sees a read or write command, the addressed device must return a READY to the CPU. If this doesn't occur, the 8085A will go into a wait state. The normal commands RD/ and WR/ are generated just after the READY sample point and therefore would cause wait states to occur if they were used. To avoid this problem, the iSBC 544 uses the status indicators (S0 and S1) to generate what are called Advance Commands (AMR/, AMR, and ACMD/). Figure 4-12 shows the relationship of normal commands and advance commands. An advance memory write is not used, because the data is not available soon enough to be used by the RAM. Therefore one wait state will be incurred, on memory writes.

Decoder A46 and latch A45 (2ZB4) are used to generate the Advance commands. The two chips decode the status of S1 and S0 from the CPU to

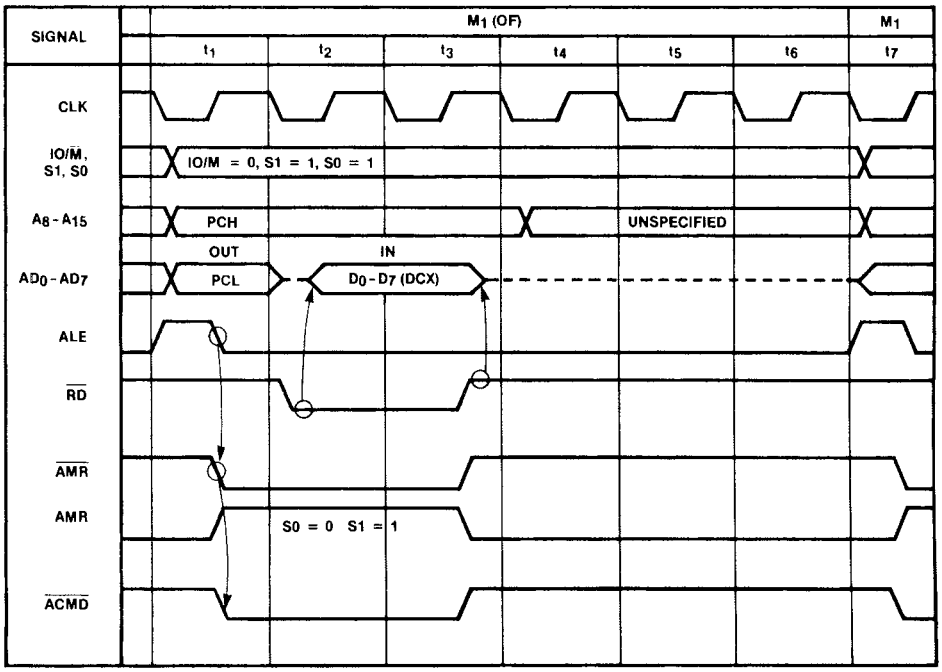


Figure 4-12. Advance Command Signals

create AMR, AMR/, and ACMD/. The ACMD/ function is used by the Dual Port logic to reserve the memory for the on-board 8085A processor.

There is one other set of commands generated by the iSBC 544, called Qualified Commands. These commands are only used when the iSBC 544 is operating in the master mode. The two commands (QMW/ and QIOW/) are necessary because when the 544 goes out to the Multibus, data must be stable for 50 nanoseconds before it gets a command. To accomplish this, the 544 qualifies the commands with a clock pulse through A43 (2ZA4) and produces the qualified Commands.

4-30. DUAL PORT CONTROL LOGIC

The Dual Port Control logic (figure 5-2, sheet 3) allows the on-board RAM to be shared by the on-board 8085A CPU and by another bus master via the Multibus. The iSBC 544 acts as a "slave" RAM device to other bus masters. When accessing its on-board RAM, the on-board CPU has priority over any attempt to access the on-board RAM via the Multibus. In this situation, the bus access is held off until the CPU has completed its particular read or write operation. When a bus access is in progress, the Dual Port logic holds off any subsequent CPU request until the bus access is terminated.

4-31. BUS MASTER RAM ACCESS. The Dual Port Logic is controlled by the three flip-flops shown at the bottom of sheet 3. The normal state of the flip-flops after reset, selects the on-board CPU to have access to memory. If the bus master wants access to memory, it must cause all three flip-flops to toggle. The first flip flop (A37-5) will set if the following conditions are true, and a clock pulse occurs:

a. Previous RAM access is over (ACK/ asserted).

b. The on-board processor does not want memory (ALE/ and CSM0/ high).

c. There is a request from the bus (MWTC/ or MRDC/) and the address is within the iSBC 544's RAM.

The second flip-flop (A36-9) will set if the following conditions are met:

a. Flop A37-5 is set.

b. The on-board processor still does not want access (CSM0/ or ACMD/ high).

c. The address is within the iSBC 544's RAM. This gate is necessary, because of the amount of time it takes for the address to get through the address decode logic.

When flop A36-9 sets, it latches itself through A70. The output of the flop activates System Bus Enable (SBEN/) which gates the off-board address into the inputs of the 8202 RAM Controller (5ZD6) and System Bus Direction Control (SBDC/) which allows the data from the Multibus into the RAM (5ZC3). It also disables MBEN/ which prevents an on-board address and on-board data from getting to the RAM.

When the last flip-flop in the string sets (A36-6), the negation output goes low activating the Command Buffer A60 (3ZD3) and allowing the bus master's command (MWTC/ or MRDC/) to be gated to the RAM controller (5ZC6). The negation output also activates Dynamic RAM Chip Select (DRCS/) which selects the RAM controller.

After the RAM accepts the command, it will send out an acknowledge (SACK/) which will allow the flip-flops to start to reset. The MEMACK/ signal from the RAM controller is gated out to the Multibus (through A71) to generate XACK/ (Transfer Acknowledge) which notifies the bus master that its command has been acknowledged.

The acknowledge signal (SACK/) starts the clearing process for the dual-port flip-flops. When SACK/ occurs, the output of A53 (3ZB6) will go low disabling the input to A37 (pin 2). The next clock pulse will clear A37 (pin 5). When XACK/ occurs, the bus master will drop its command. The output of A38-6 will go low and the output of A54-2 will go high. The termination of the command also causes SACK/ to go high (via the 8202, A62). With both inputs to A57 true, its output will go high. Consequently, on the next clock pulse, A37-9 will set. This removes the preset from A36 (pin 10). A36 (pin 9) will reset on the next clock pulse, which will allow A36 (pin 6) to clear on the following pulse. The flops are now back in the normal state, which is set up to favor the on-board CPU's RAM requests.

4-32. ON-BOARD CPU RAM ACCESS. As previously mentioned, the Dual Port Control logic flip-flops are normally in a state which favors an on-board access. When the on-board processor wants to access RAM, it generates a read command (AMR/) or a write command (MW/) which is gated to the 8202 RAM controller through the Command Buffer A60 (3ZD3). The gating is controlled by A38 which looks at CSMO/ (On-Board Memory Chip Select) and the negation side of flip-flop A36-6 which is normally high. The output of A36 (3ZA3) also generates DRCS/ which selects the on-board RAM through the 8202 RAM Controller. MBEN/ (Memory Bus Enable) which is normally low allows the 8085A CPU address (4ZC7) and data (5ZD2) to be gated to the RAM.

After the RAM receives the command, it generates an acknowledge (SACK/) which is gated to the CPU Ready logic by A40 (3ZC2). This tells the CPU that

the RAM has received the command and a refresh is not in progress, which allows the CPU to continue its cycling.

4-33. OFF BOARD MEMORY REQUEST

When operating in Master Mode, the iSBC 544 is capable of generating commands to off-board memory and off-board I/O (such as an iSBC 534 I/O Expansion board).

The off-board commands for memory AMR/ (Advance Memory Read) and QMW/ (Qualified Memory Write), are gated on to the Multibus by OFF BD REQ/. OFF BD REQ/ (3ZB1) says that the iSBC 544 is in Master Mode (MASMD/) and the 8085A address is not for the on-board RAM (ONBD/). These commands go out on the bus where they are processed by the addressed external board.

When the off-board memory processes the command, it sends back an acknowledge XACK/ (3ZA1) which is used by the 8085A CPU's READY logic. XACK/ is anded with OFF BD REQ/ to generate READY on the input to the 8085A. Allowing the 8085A to terminate the command and go on with the next command.

4-34. I/O OPERATION

The following paragraphs describe on-board and system I/O operations. The actual functions performed by specific read and write commands to on-board I/O devices are described in Chapter 3.

4-35. ON-BOARD I/O OPERATION. During an on-board I/O operation, the address of the selected device is sent out on the upper 8 bits of the 8085A CPU address bus. The upper 8 bits (PAB8-PABF) are applied to PROM Decoder A50 (4ZB3). The address selected in the PROM will contain the necessary bit configuration to activate the chip and the corresponding I/O control lines. For example, if the I/O command is to the 8259 PIC the CPU would address E6. Addressing E6 in the PROM will produce a bit configuration of 1D (00011101). The lower four bits are applied to the 8205 Decoder A32 (4ZB2) where they select output 5. Output 5 (CSIO/) is the chip select for the 8259 PIC chip A29 (7ZB7). The upper 3 bits (000) activate the I/O functions IOBA/ (I/O Bus Allow), IOACK/ (I/O Acknowledge), and ONBD/ (On-Board Address) which are used in the execution of the I/O command. A complete list of all Decode PROM (A50) locations is found in Appendix C.

After the I/O device has been selected, specific functions for the chips are selected by PAB0 and PAB1 (Processor Address Bits 0 and 1). For example addresses D8, D9, DA, and DB all select the 8253 PIT chip, but different counters on the chip. (Refer to table 3-2 for a further breakdown.)

Along with selecting the I/O chip, the 8085A also generates the following signals, IOCMD/ (I/O Command), IOR/ (I/O Read), and IOW/ (I/O Write). IOCMD/ is used by the Decode PROM (A50) to activate the I/O control signals, and by the 8155 PPI to select the I/O portion of the chip (6ZC2). IOR/ is used by the 8253 PIT, the 8259 PIC, and by the 8251A USART's to perform a read operation. The IOR/ signal is also used to feed the Multibus when the iSBC 544 is in the Master Mode and wants to perform and external I/O read (3ZD7). IOW/ is used on the same chips as IOR/ to allow them to perform a write operation, and is used by the iSBC 544 to control the Master Mode flop.

After the I/O address has been decoded, and it is determined that the CPU is executing an I/O command, IOACK/ (I/O Acknowledge is generated by A50 (4ZB1) and drives the 8085A CPU's READY line. This allows the CPU to proceed to the next machine cycle and finish executing the I/O command.

4-36. SYSTEM I/O OPERATION. Address bits PAB8-PABF are decoded by Decode PROM A50 described in paragraph 4-35. If the address is not for an on-board I/O device, ONBD/ remains false. If the iSBC 544 is in the Master Mode, MAMSD true and ONBD/ false AND together at A39 (3ZB2) to produce OFF BD REQ/ (Off Board Request). OFF BD REQ/ gates the I/O command (IOR or QIOW) on to the Multibus (3ZD7) to be transferred to an external device. The off-board device acknowledges the command and sends back XACK/ true via the Multibus (3ZA2), to drive the 8085A CPU's READY line.

The data transfer is controlled by SBDC/ (System Bus Data Control). On an off-board I/O read operation, SBDC/ would be true allowing the data (DAT0/-DAT7/) from the Multibus to be gated through the Bi-directional Bus Drivers (5ZC2) on to the processor data bus (PDB0-PDB7). On an off-board I/O write, SBDC/ would be false, allowing the processor data bus (PDB0-PDB7) to be gated out to the Multibus (DAT0/-DAT7).

4-37. ROM/PROM OPERATION

The two ROM/PROM chips are installed by the user in IC sockets A35/A51 (4ZD3). Memory addresses 0000 - 1FFF are reserved exclusively for ROM/PROM; the actual occupied memory space depends on the ROM/PROM chips used:

Chip Size	Chip Addresses In	
	A35	A51
2K x 8	0000 - 07FF	0800 - 0FFF
4K x 8	0000 - 0FFF	1000 - 1FFF

When the 8085A CPU is addressing ROM, bits PAB0- PABB are applied directly to the PROM chips, and bits PAB8-PABF are applied to the PROM Decoder chip A50 (4ZB3). If the address is within the range of the iSBC 544's ROM, the output bit configuration from A50 will select output 00 (CSPR0/) or 01 (CSPR1/) of 8205 A33 (4ZC2), and cause IOACK/ (I/O Acknowledge) to be true. CSPR0/ and CSPR1 are used to power up the 2716 chips. These signals and RD/ from the 8085A CPU activate the PROM chip which contains the location that is being addressed by PAB0-PABB. IOACK/ drives the CPU READY line, signifying the acknowledgement of the command by the ROM.

The data outputs from the ROM (IODB0/-IODB7/) are gated through the Bi-Directional Data Buffer A67 (2ZC2) by CPU status signal S1 (S1=1 for Read) to the CPU.

4-38. RAM OPERATION

As described in paragraph 4-30, the Dual Port Control logic allows the on-board RAM facilities to be shared by the on-board CPU and by another bus master via the Multibus. The following paragraphs describe the RAM Control and the overall operation of how the RAM is addressed for read/write operation.

4-39. RAM CONTROLLER. All address and control inputs to the on-board RAM are supplied by RAM Controller A62 (5ZD6). The 8202 RAM Controller provides a RAS refresh timing cycle to dynamic RAM chips A79-A86. Default jumper 68-67 holds the TREF (RAM Refresh) signal low, allowing the RAM Controller to operate in the automatic refresh mode. In the automatic refresh mode, a read or write request can be delayed if a refresh cycle is in progress.

The RAM Controller when enabled with a low input to the PCS/pin, multiplexes the address to the RAM chips. Low order address bits A0-A6 are presented at the RAM input pins and RAS/ is driven low at the beginning of the first memory clock cycle. High-order address bits A7-A13 are presented at the RAM input pins and CAS/ is driven low during the second memory clock cycle.

The RAM Controller also examines its RD/ and WRT/ inputs. If WR/ is low, the RAM Controller drives its WE/ output low just before CAS/, then high to provide a WRITE signal to RAM.

When the memory cycle begins, the RAM Controller drives its SACK/ output low (delayed until XACK/ if refresh in progress), and when the cycle is complete (i.e., data is valid), drives its XACK/ output low. The SACK/ and XACK/ outputs go high when the RD/ or WRT/ input goes high.

4-40. ON-BOARD READ/WRITE OPERATION. The 8085A CPU initiates a RAM operation by generating a read command (AMR/) or a write command (MW/). These commands are qualified by the Dual Port Control logic (Sheet 3) and fed to the inputs of the 8202 RAM Controller as DRRD/ (Dynamic RAM Read) and DRWR/ (Dynamic RAM Write). The RAM Controller chip is selected by DRCS/ (3ZC1) which says that the command was for the RAM (CSM0/) and we are not in battery backup operation (MEMORY PROTECT/ false).

During a RAM read, the address from the 8085A (PAB0-PABF) is gated into the RAM Address Buffers A63 and A64 (4ZD6) by MBEN/. The output of the buffers (RAD0/-RADF/) is applied to the address inputs of the 8202 RAM Controller, where it is multiplexed to the RAM. DRRD/ being true causes the 8202 to perform a read. The data from the RAM chips is latched by A78 (5ZC3) and transferred to the 8085A CPU data bus by MBEN/ from the Dual Port Control logic. The RAM acknowledge (RMACK/) is generated by SACK/ from the 8202 qualified by the Dual Port Control logic. RMACK/ drives the 8085A CPU's READY line.

During a RAM write, the data from the 8085A CPU is gated through A66 (5ZD2) to the RAM chips. With DRWR/ true on the input of the RAM Controller, the 8202 gates out WE/, a write operation is performed and data is written in to the location specified by the RAM address (RAD0/-RADF/).

Reference paragraph 4-18 to determine the possible influence of wait states on the RAM read or write operations.

4-41. BUS MASTER READ/WRITE OPERATION. Another bus master on the system can access the iSBC 544's RAM. In order to do this, the bus master must gain control of the RAM through the Dual Port Control logic, as explained in paragraph 4-31.

Assuming no on-board CPU access of RAM is in progress, address bits ADR10/-ADR13/ from the Multibus are decoded by A56 (4ZA6) to select a 64K page of memory. If bits ADR10/-ADR13/ are selecting the 64K page allocated to this board, the correct output of A56 will be jumpered to the input of function BSAD/ (Board Address). Bits ADRC/-ADRF/ from the Multibus are compared to the Base Address switches (4ZB6) by Decode PROM A41 to determine if the address is within the 4K boundary established as the base address for this board. If the comparison is true, and if the address is within the RAM size (determined by positions 6 and 7 of S1), the other input to BSAD/ will be satisfied and the bus master access will be allowed.

Address bits ADR0/-ADRB/ and ATRC/-ATRE/ are gated through the Bi-Directional Bus Drivers A73-A75 (5Z-7) by SBEN/ (System Bus Enable). When DRWR/ (Dynamic RAM Write) or DRRD/ (Dynamic RAM READ) becomes true on the inputs to the RAM Controller (5ZD6), the address in RAM is either read or written. The data is transferred into or out of the RAM (depending on read or write) through the Bi-Directional Bus Drivers A76 and A77 and onto or off of the Multibus (5ZC2).

The RAM Controller generates XACK/ which is transferred out on the Multibus, to acknowledge execution of the bus masters command.

There is one area that needs further explanation. Decode PROM A41 (4ZB5) is used for address comparison, and also for address transformation. The PROM has been programmed to transform any incoming address to an address starting at 8000H. The on- board base address of RAM is 8000, so regardless of the base address on the Multibus and the switches, the address seen by the RAM must be equal to or greater than 8000. Appendix C shows the outputs of the Decode PROM for different address inputs from the Multibus. This transformation is necessary so that the Flag Interrupt logic can generate an interrupt unique to the iSBC 544.

4-42. INTERRUPT OPERATION

The following paragraphs describe the interrupt logic areas of the iSBC 544 Intelligent Communications controller.

4-43. MULTIBUS INTERRUPTS. The on-board 8085A can be interrupted by a bus master by way of the Multibus. One of eight interrupts INT0/-INT7/ (2ZA7) can be jumpered to the RST 5.5 input on the CPU.

The 8085A can generate an interrupt request to another bus master. The SOD output of the 8085A CPU is jumpered to one of the eight Multibus interrupt lines INT0/-INT7/ (2ZA7).

4-44. FLAG INTERRUPT. The Flag Interrupt is used as a communication device between the bus masters and iSBC 544 on the system. The Flag Interrupt flop A30 (5ZB2) is set when a write (DRWR/) is performed by a bus master into the base address of the on-board RAM. The output of the Flag Interrupt flop (FINT/) feeds to the RST 5.5 input of the CPU (2ZC8). The FINT output of the flop feeds to the C port of the 8155 PPI (6ZC8) where it can be read.

The Flag Interrupt flop is cleared when a read (DRRD/) is performed on the base location of the RAM by the on-board 8085A CPU.

4-45. CARRIER DETECT INTERRUPT. The Carrier Detect Interrupt flops A23 and A24 (6ZC5) monitor the carrier detect signals on the serial ports. If one of the carriers (CD0-CD3) drops out, one of the four flops would set. When the flop sets, the negation side of the flop goes low setting PINT (Carrier Detect Interrupt). PINT drives the RST 6.5 input to the CPU. The Carrier Detect signals also feed to port B on the 8155 PPI (6ZC7) where they can be interrogated.

4-46. RING INDICATOR INTERRUPT. The Ring Indicator Interrupt flop A44 (6ZB5) monitors ring lines (RI0-RI3) from the serial ports. If a ring occurs on one of the lines, flop A44 sets and sets RINT (Ring Interrupt). RINT drives the RST 6.5 input to the CPU. The ring lines RI0-RI3 (6ZC8) also feed to port B on the 8155 PPI where they can be interrogated.

4-47. 8259 INTERRUPT CONTROLLER. The 8259 PIC A29 (7ZA7) monitors the interrupt lines from the 8251A USARTS. The output of the 8259 drives the INTR line on the 8085A CPU.

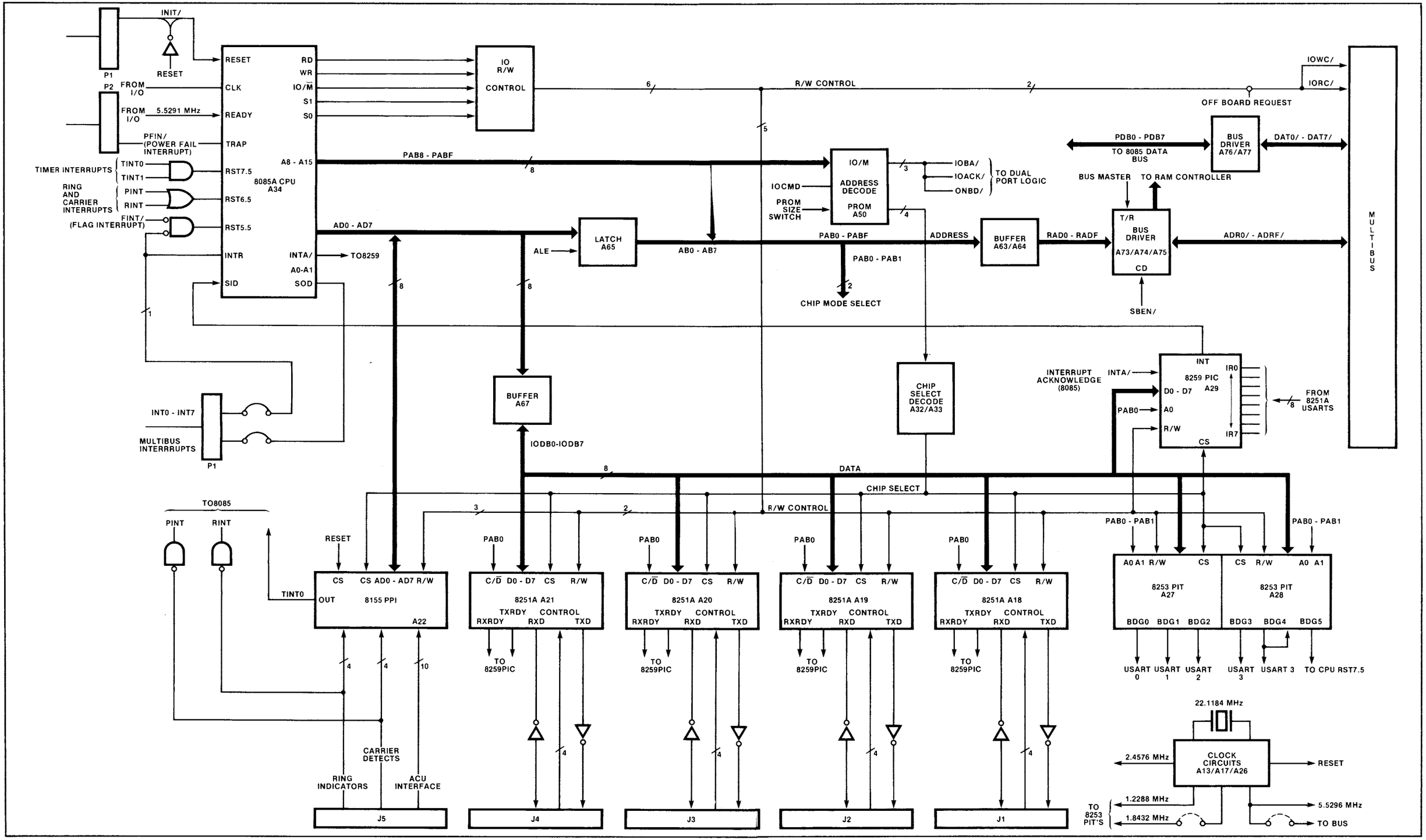


Figure 4-1. iSBC 544 Input/Output and Interrupt Block Diagram

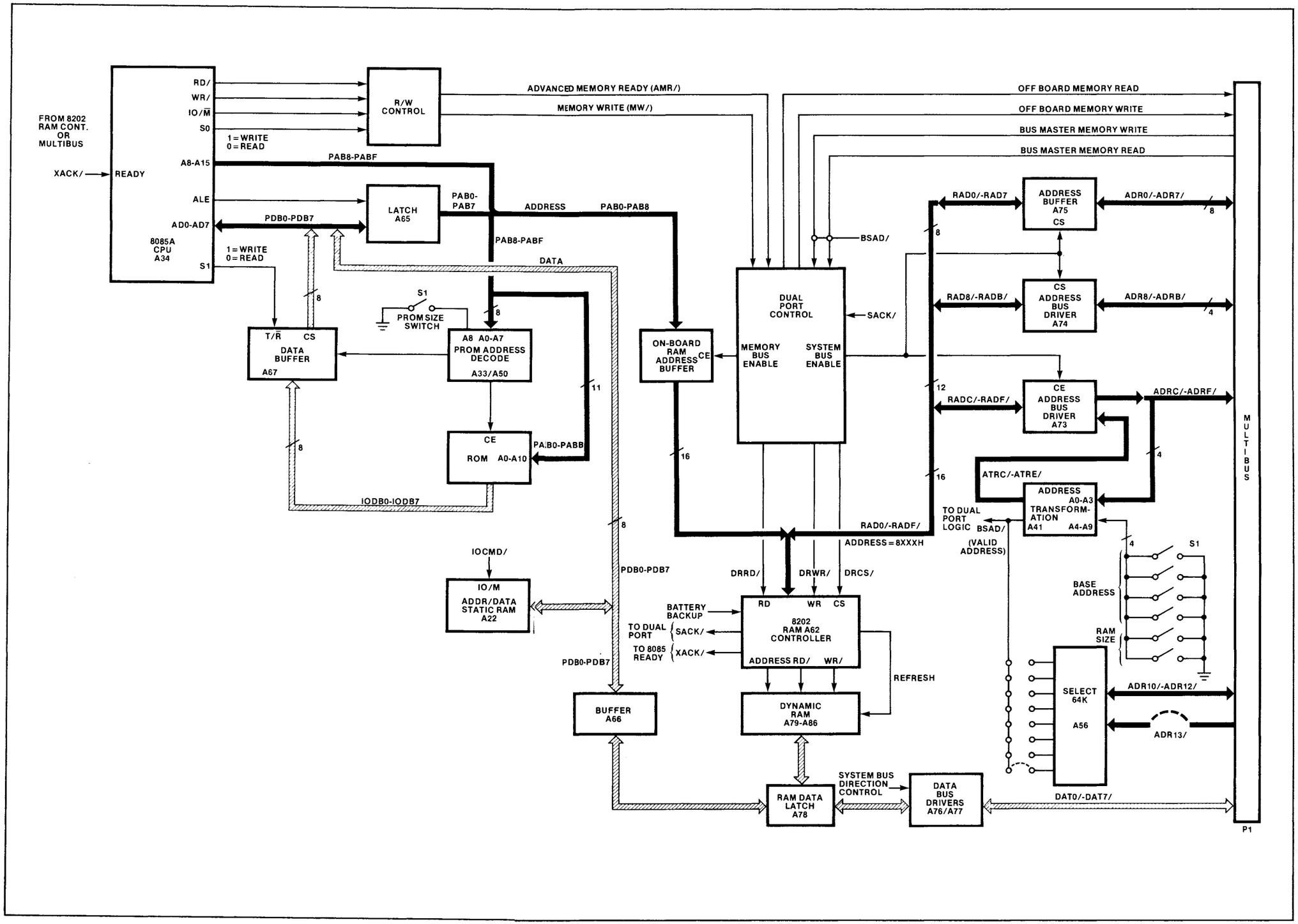


Figure 4-2. iSBC 544 Memory Block Diagram

CHAPTER 5
SERVICE INFORMATION

5-1. INTRODUCTION

This chapter provides a list of replaceable parts, service diagrams, and service and repair assistance instructions for the iSBC 544 Intelligent Communications Controller Board.

5-2. REPLACEABLE PARTS

Table 5-1 provides a list of replaceable parts for the iSBC 544. Table 5-2 identifies and locates the manufacturers specified in MFR CODE column in table 5-1. Intel parts that are available on the open market are listed in the MFR CODE column as "COML"; every effort should be made to procure these parts from a local (commerical) distributor.

5-3. SERVICE DIAGRAMS

The iSBC 544 parts location diagram and schematic diagram are provided in figures 5-1 and 5-2, respectively. On the schematic diagram, a signal mnemonic that ends with a slash (e.g., IOWC/) is active low. Conversely, a signal mnemonic without a slash (e.g., IOC) is active high.

5-4. SERVICE AND REPAIR ASSISTANCE

United States customers can obtain service and repair assistance from Intel by contacting the MCD Technical Support Center in Santa Clara, California at one of the following numbers:

Telephone:

From Alaska or Hawaii call –
(408) 987-8080

From locations within California call toll free –
(800) 672-3507

From all other U.S. locations call toll free –
(800) 538-8014

TWX: 910-338-0026

TELEX: 34-6372

Always contact the MCD Technical Support Center before returning a product to Intel for service or repair. You will be given a "Repair Authorization Number", shipping instructions, and other important information which will help Intel provide you with fast, efficient service. If the product is being returned because of damage sustained during shipment from Intel, or if the product is out of warranty, a purchase order is necessary in order for the MCD Technical Support Center to initiate the repair.

In preparing the product for shipment to the MCD Technical Support Center, use the original factory packaging material, if available. If the original packaging is not available, wrap the product in a cushioning material such as Air Cap TH-240 (or equivalent) manufactured by the Sealed Air Corporation, Hawthorne, N.J., and enclose in a heavy-duty corrugated shipping carton. Seal the carton securely, mark it "FRAGILE" to ensure careful handling, and ship it to the address specified by MCD Technical Support Center personnel.

NOTE

Customers outside of the United States should contact their sales source (Intel Sales Office or Authorized Intel Distributor) for directions on obtaining service or repair assistance.

Table 5-1. Replaceable Parts

Reference Designation	Description	MFR. Part No.	MFR. Code	QTY.
A1-A5, A11, A12, A16	IC, 1489A Quad Line Receiver	SN75189AN	TI	8
A6-A10, A14	IC, 1488 Quad Line Driver	SN75188AN	TI	6
A13	IC, 74LS20 Dual 4 Input Positive-Nand Gate	SN74LS20N	TI	1
A15	IC, 8224 Clock Generator	Intel 8224	Intel	1
A17	IC, 7492A Divide-By-Twelve Counter	SN7492AN	TI	1
A18-A21	IC, 8251A Serial I/O Interface	Intel 8251A	Intel	4
A22	IC, 8155 Programmable Peripheral Interface	Intel 8155	Intel	1
A23, A24, A43	IC, 74LS74 Dual D-Type Positive Trigger Flip-Flop	SN74LS74N	TI	3
A25, A38	IC, 74S00 Quad 2-Input Pos. NAND-Gate	SN74S00N	TI	2

Table 5-1. Replaceable Parts (Cont'd.)

Reference Designation	Description	MFR. Part No.	MFR. Code	QTY.
A26, A36, A37	IC, 74S74 Dual D-Type Flip-Flop	SN74S74N	TI	3
A27, A28	IC, 8253 Programmable Interval Timer	Intel 8253	Intel	2
A29	IC, 8259 Programmable Interrupt Ctrlr.	Intel 8259	Intel	1
A30	IC, 7470 Gated J-K Flip-Flop	SN7470N	TI	1
A31, A59	IC, 74LS04 Hex-Inverter	SN74LS04N	TI	2
A32, A33, A56	IC, 8205 1-of-8 Binary Decoder	Intel 8205	Intel	3
A34	IC, 8085A 8-Bit Microprocessor	Intel 8085A	Intel	1
A35, A50, A51	Socket, 24 Pin DIP		TI	3
A39	IC, 74LS00 Quad 2-Input Positive NAND-Gate	SN74LS00N	TI	1
A40	IC, 74S32 Quad 2-Input OR-Gate	SN74S32N	TI	1
A41	IC, 3625-2 1K x 4 PROM	9100124	Intel	1
A42, A48, A70	IC, 74LS32 Quad 2-Input OR-Gate	SN74LS32N	TI	3
A44	IC, 74LS109 Dual J-K Flip-Flop	SN74LS109N	TI	1
A45	IC, 74S175 Quad D-Type Flip-Flop	SN74S175N	TI	1
A46	IC, 74LS157 Data Selector/Multiplexer	SN74LS157N	TI	1
A47, A57	IC, 74LS08 Quad 2-Input Positve-AND Gates	SN74LS08N	TI	2
A49, A54	IC, 74S04 Hex Inverter	SN74S04N	TI	2
A50	IC, 3628 8K Bipolar PROM	9100127	Intel	1
A52	Not Used			
A53	IC, 74S11 3-Input Positive-AND Gates	SN74S11N	TI	1
A55	IC, 74S20 Dual 4-Input Positive-NAND Gates	SN74S20N	TI	1
A58	IC, 74S02 Quad 2-Input Positive-NOR Gates	SN74S02N	TI	1
A60, A71	IC, DM 8097 Tri-State Buffer	DM 8097	NAT	2
A61	IC, 74S133 13-Input Positive-NAND Gates	SN74S133N	TI	1
A62	IC, 8202 RAM Controller	Intel 8202	Intel	1
A63, A64	IC, 74LS240 Octal Buffers Line Driver/Receiver	SN74LS240N	TI	2
A65, A78	IC, 74LS373 Octal D-Type Latches	SN74LS373N	TI	2
A66, A67, A75	IC, DP 8304 Bi-Directional Transmitter/Receiver	DP 8304	NAT	3
A68	Not Used			
A69	Not Used			
A72	IC, 7406 Hex Inverters Buffer/Drivers	SN7406N	TI	1
A73, A74	IC, 8216 Bi-Directional Bus Driver	Intel 8216	Intel	2
A76, A77	IC, Intel 8226 Bi-Directional Bus Driver	Intel 8226	Intel	2
A79-A86	IC, Intel 2117-4 16K RAM	Intel 2117-4	Intel	8
CR1	Diode, IN4002	OBD	COML	1
C1, C4, C6-8, C10-12, C14-16, C18-21, 26, 27, 33-60, 62-70, 72-80, 87, 89, 95, 97, 99, 101	Capacitor, fxd., 1μF, +80, −20%, 50v	OBD	COML	69
C2	Capacitor, Mica, 10pF, ±5%, 500v	OBD	COML	1
C3	Capacitor, fxd., 10μF, ±10%, 20v	OBD	COML	1
C22, C23-25, C28-31, C88, C96 C100, C102	Capacitor, fxd., 01μF, +80, −20%, 50v	OBD	COML	12
C61	Capacitor, fxd., .001μF, ±20%, 50v	OBD	COML	1
C81, 86, 90, 94, 98, 103	Capacitor, fxd., .33μF, +80, −20%, 50v	OBD	COML	6
C71, 82-85, 91, 93	Capacitor, fxd., 22μF, ±10%, 15v	OBD	COML	7
C92	Capacitor, fxd., 4.7μF, ±10%, 10v	OBD	COML	1
R1	Resistor, fxd., comp., 100K ohm, ±5%, ¼w	OBD	COML	1
R2, R8	Resistor, fxd., comp., 5.1K ohm, ±%, ¼w	OBD	COML	2
R3	Resistor, fxd., comp., 10K ohm, ±5%, ¼w	OBD	COML	1
R4, R6, R7, R9	Resistor, fxd., comp., 1K ohm, ±%, ¼w	OBD	COML	4
R5	Resistor, fxd., comp., 430 ohm, ±5%, ¼w	OBD	COML	1
RP1, RP3	Resistor, pack, 8-pin 1K ohm	OBD	COML	2
RP2, RP4, RP6	Resistor, pack, 8-pin 10K ohm	OBD	COML	3
RP5, RP7	Resistor, pack, 8-pin 22K ohm	OBD	COML	2
S1	Switch 8 position, DIP	OBD	COML	1
VR1	IC, LM320LZ-5, Voltage Regulator	LM320LZ-5	NAT, MOT	1
Y1	Crystal, 22.1184 MHz fundamental	HW3	CTS	1

Table 5-1. Replaceable Parts (Cont'd.)

Reference Designation	Description	MFR. Part No.	MFR. Code	QTY.
W1-W4, A41	Socket, DIP, 18-pin	C93-18-02	TI	5
A86	Socket, DIP, 16-pin	C93-16-02	TI	1
W5-W8	Socket, DIP, 8-pin	C93-8-02	TI	4
A78	Socket, DIP, 20-pin	C93-20-02	TI	1
A62, A34	Socket, DIP, 40-pin	C93-40-02	TI	2
	Extractor, Card	S-203	SCA	2
	Pins, Wirewrap	OBD	COML	75

Table 5-2. List of Manufacturers' Codes

MFR. Code	Manufacturer	Address	MFR. Code	Manufacturer	Address
INTEL	Intel Corporation	Santa Clara, CA	AMP	AMP, Incorporated	Harrisburg, PA
TI	Texas Instruments	Dallas, TX	SCA	Scanbe, Incorporated	El Monte, CA
CTS	CTS Corporation	Elkhart, IN	COML	Any Commercial Source; Order by Description	
NAT	National Semiconductor Corporation	Santa Clara, CA	MOT	Motorola Inc.	Pheonix, Ariz.

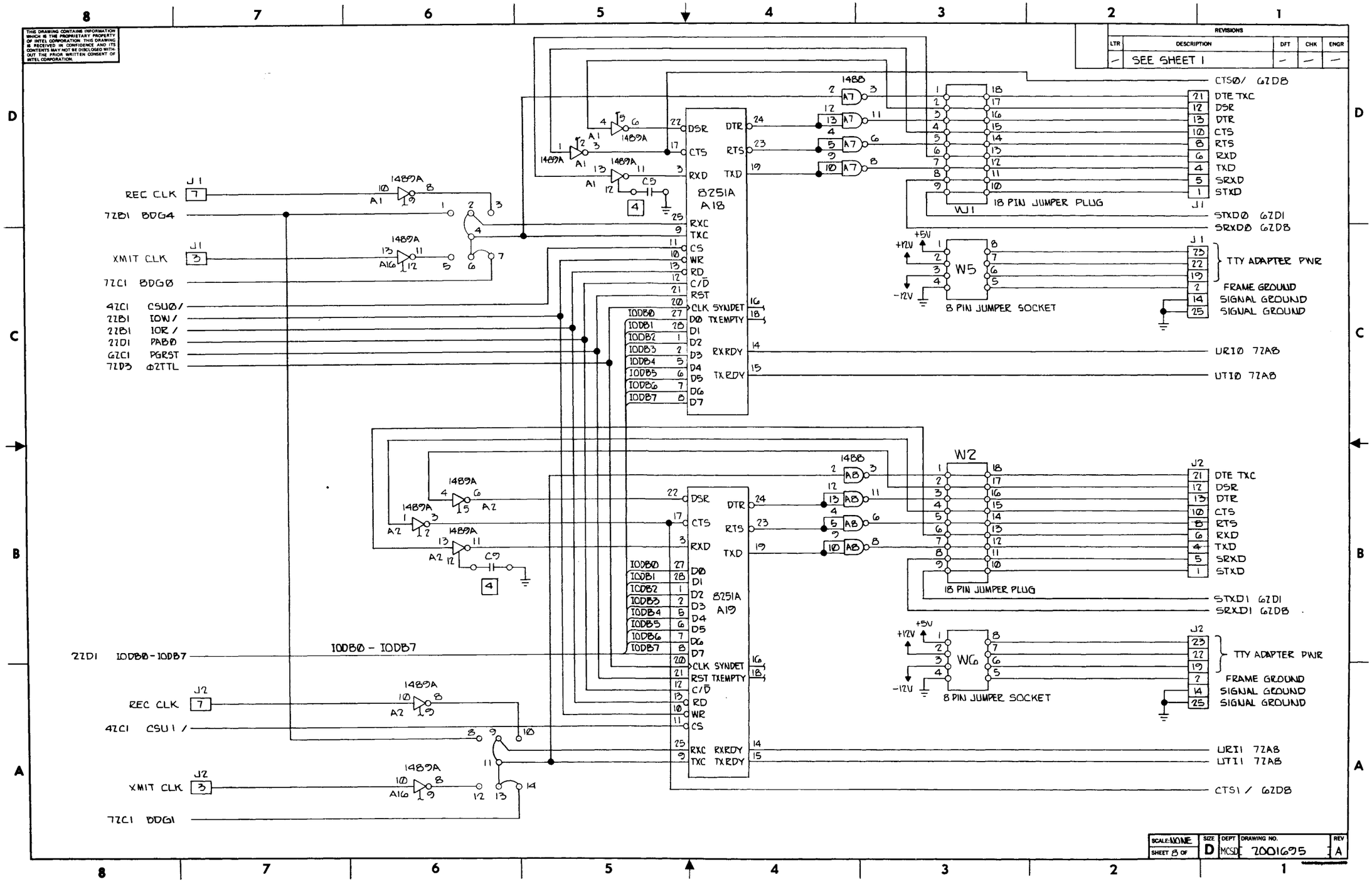


Figure 5-2. iSBC 544 Schematic Diagram (Sheet 8 of 9)

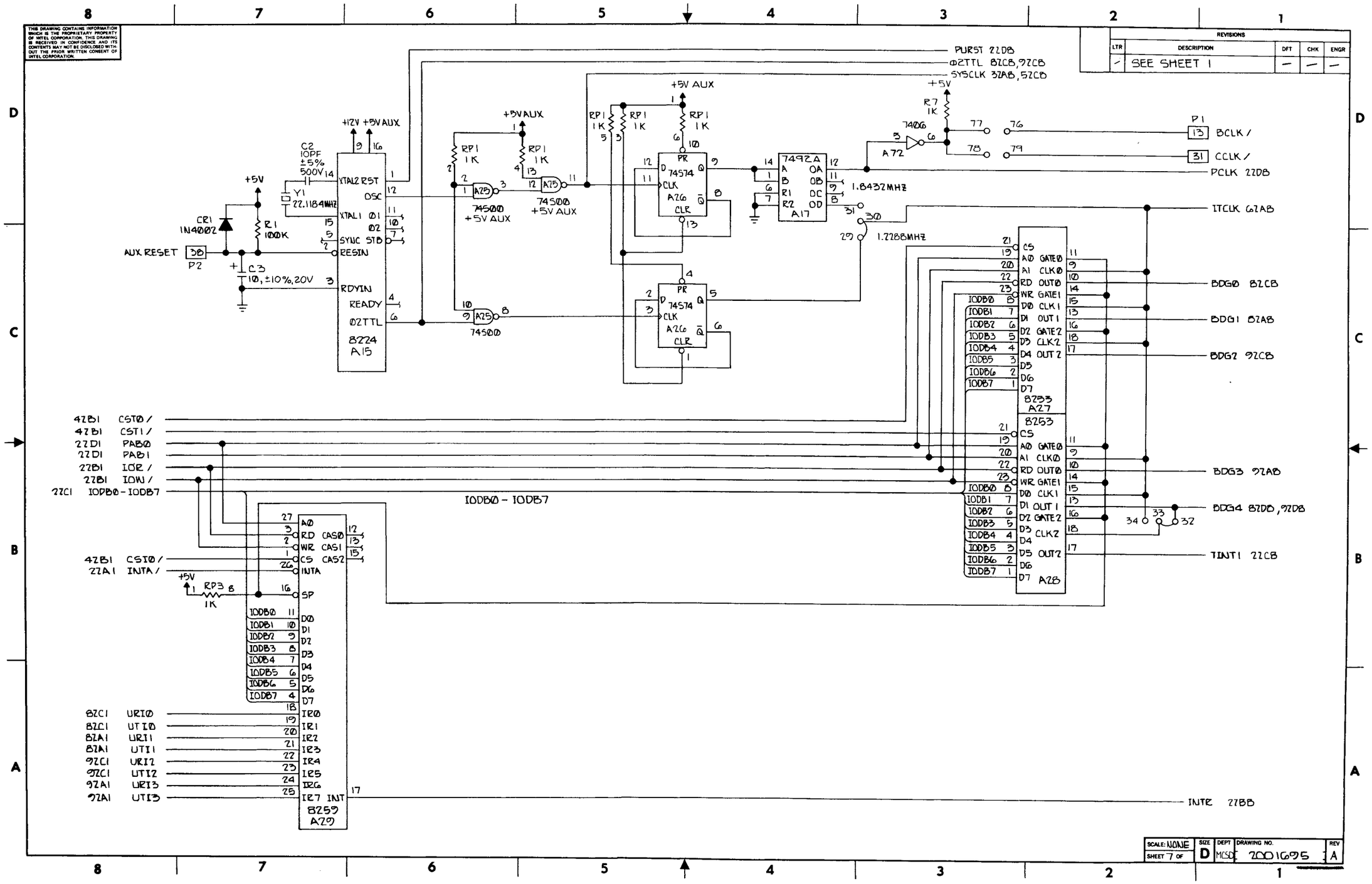


Figure 5-2. iSBC 544 Schematic Diagram (Sheet 7 of 9)

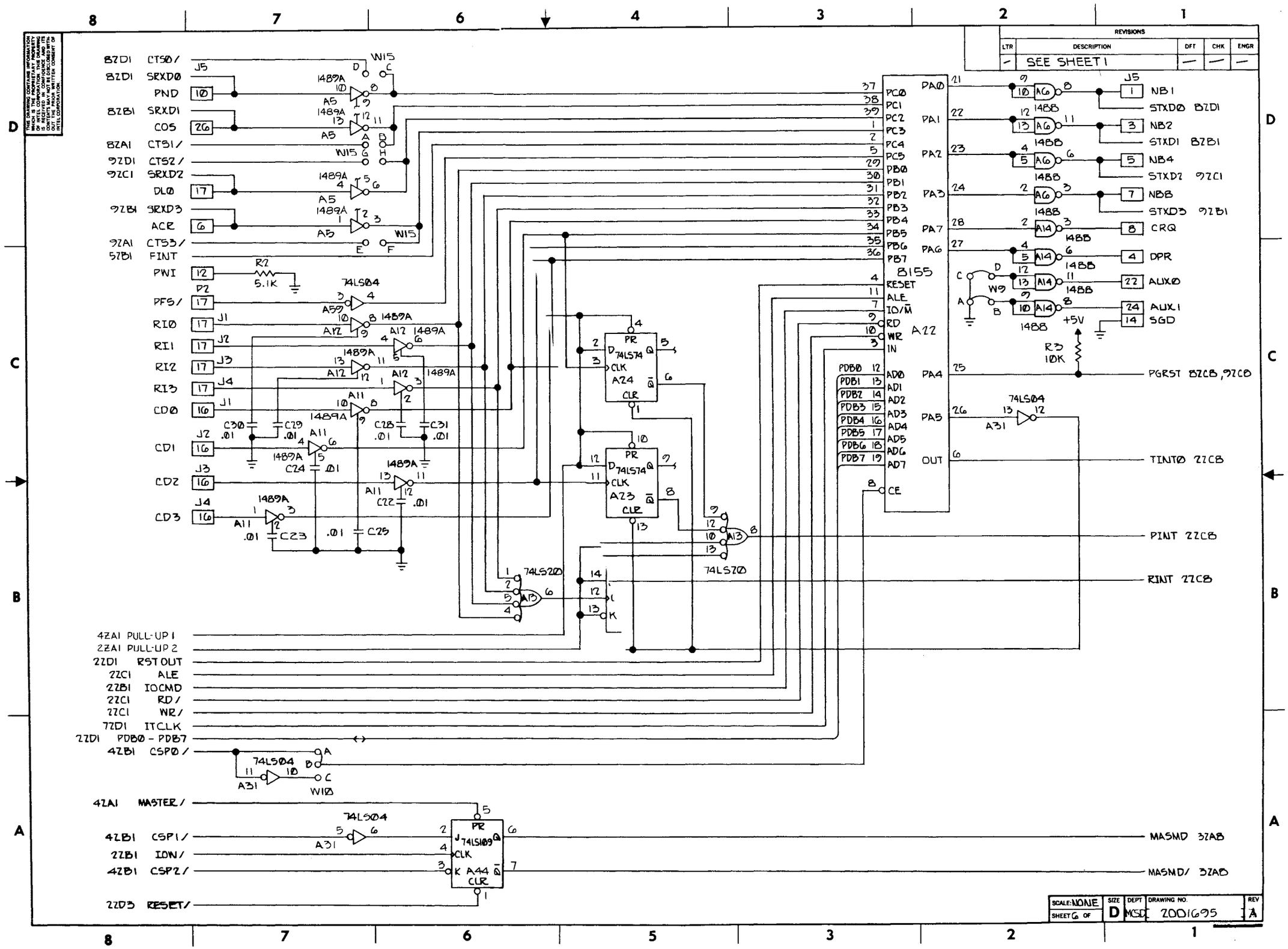


Figure 5-2. iSBC 544 Schematic Diagram (Sheet 6 of 9)

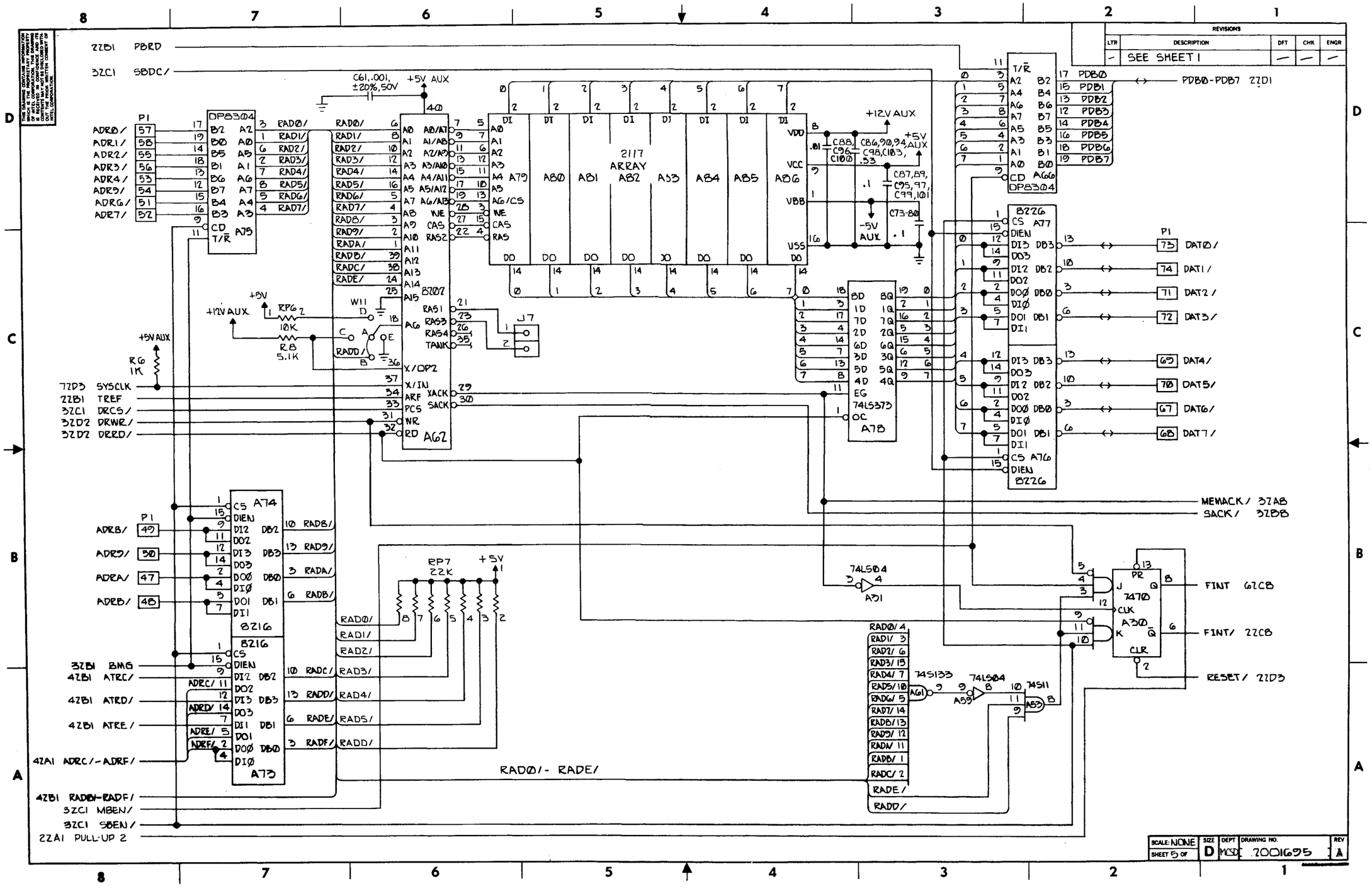


Figure 5-2. iSBC 544 Schematic Diagram (Sheet 5 of 9)

REVISIONS				
LTR	DESCRIPTION	DFT	CHK	ENGR
–	SEE SHEET 1	–	–	–

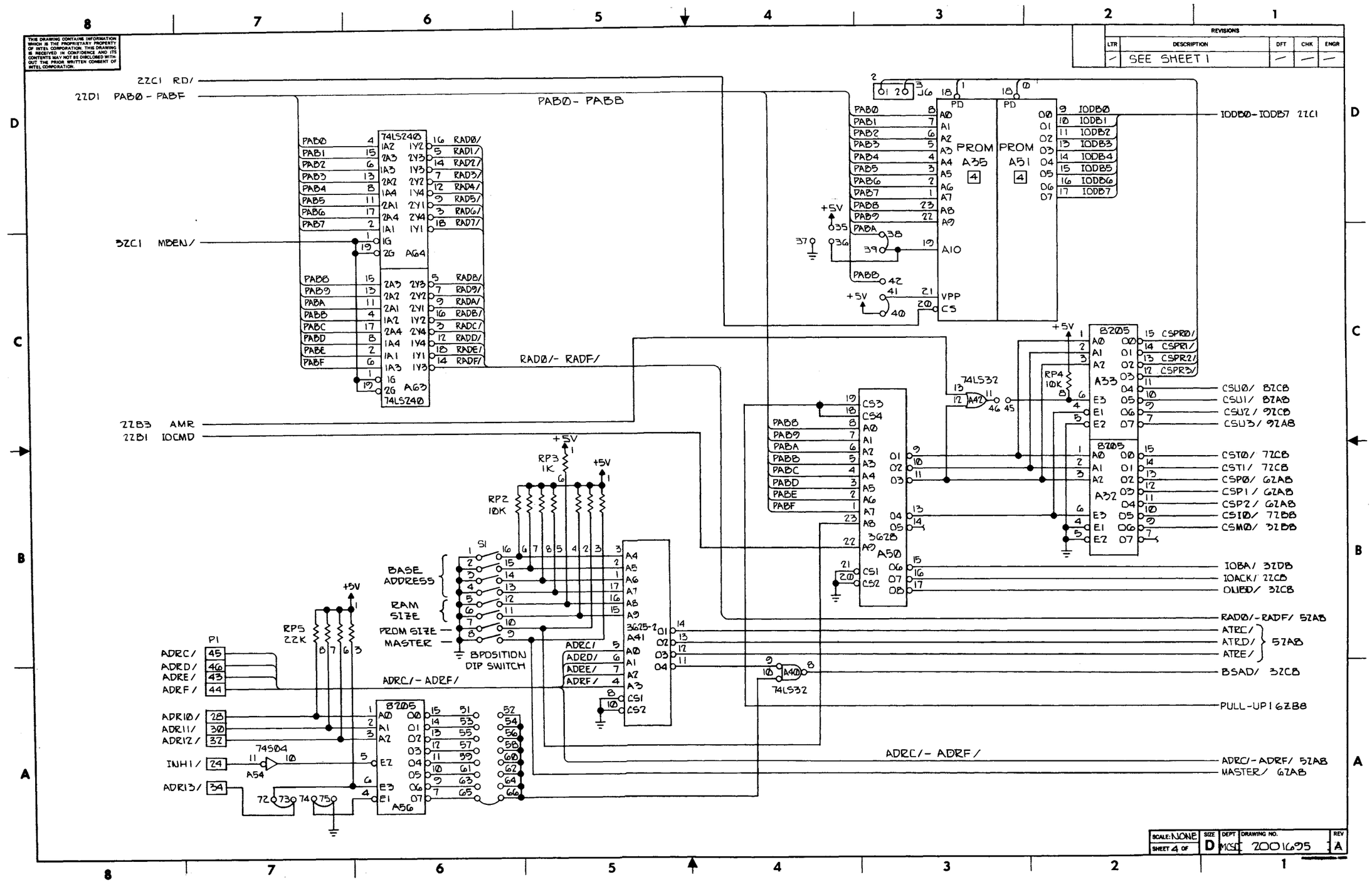


Figure 5-2. iSBC 544 Schematic Diagram (Sheet 4 of 9)

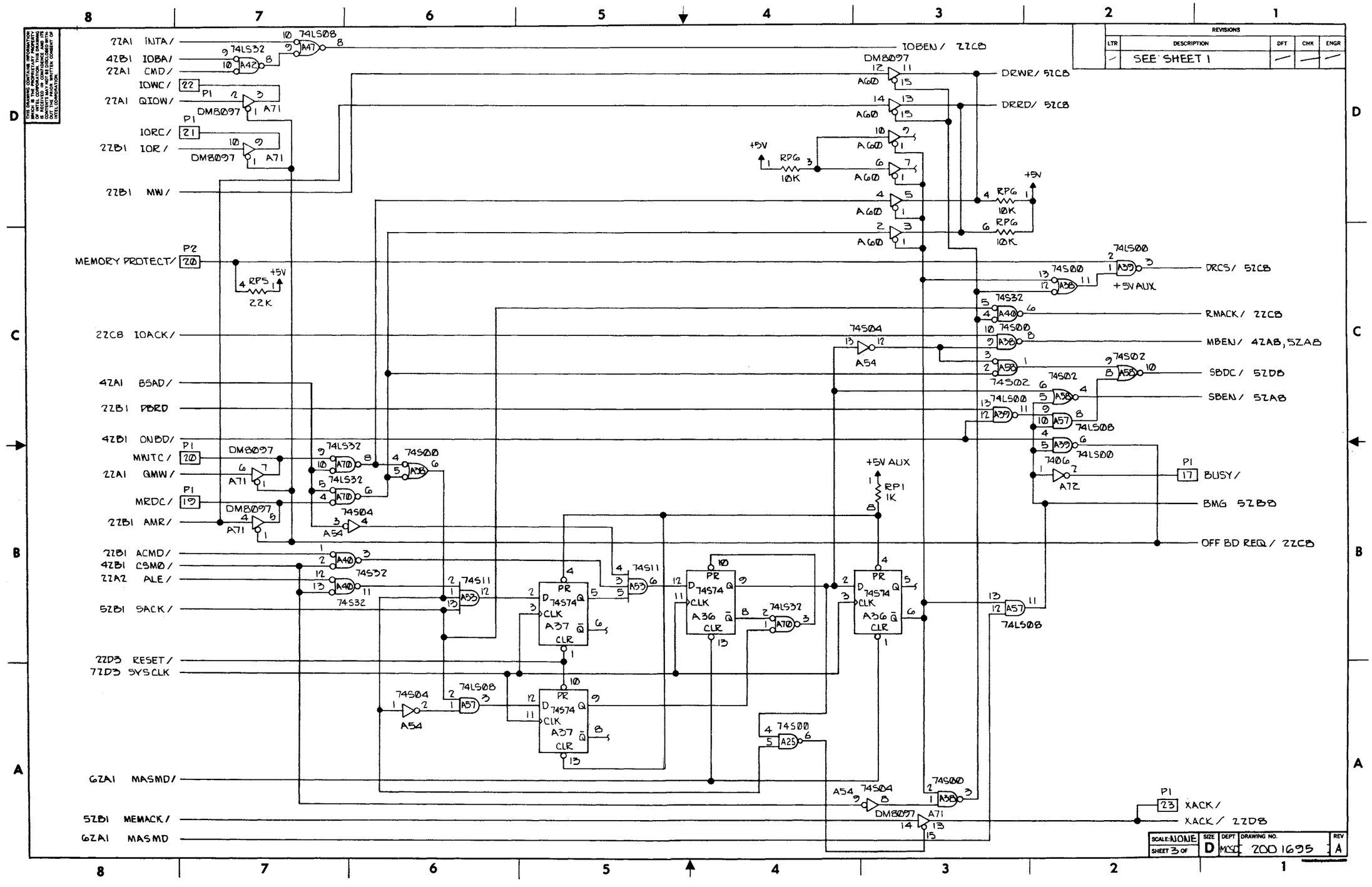


Figure 5-2. iSBC 544 Schematic Diagram (Sheet 3 of 9)

REVISIONS				
LTR	DESCRIPTION	DFT	CHK	ENGR
–	SEE SHEET 1	–	–	–

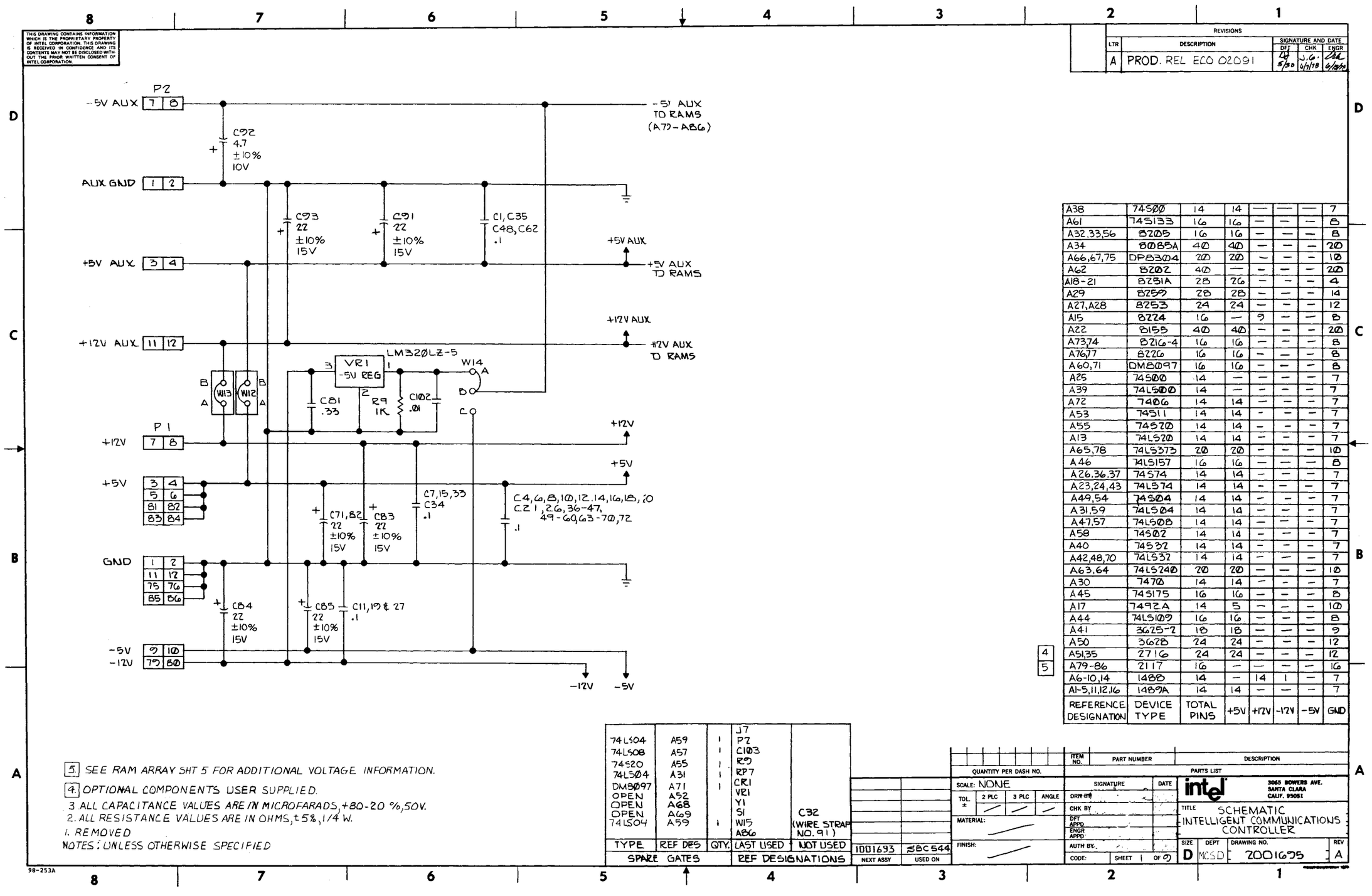
RESET/ 3ZBB,5ZA1,6ZAB

RSTOUT 6ZBB
PAB0-PABF 4ZDB
PDB0-PDB7 5ZD1, 6ZAB

7ZD3 PURST
INIT/
RP5 22K
RP3 1K
74Ø6 A72
74S04 A54
+5V
RS 430Ω
7ZD1 PCLK
3ZA1 XACK/
3ZB1 OFF BD REQ/
3ZC8,4ZB1 IOACK/
3ZC1 RMACK/
3ZD3 IOBEN/
74LS32 A70
74S7Ø A55
PZ
PFIN/
74LS04 A57
RP6 10K
RP4 10K
RP4 10K
74LS08 A47
6ZC1 TINTØ
7ZB1 TINT1
6ZB1 PINT
6ZB1 RINT
5ZB1 FINT/
74LS32 A48
74LS00 A39
7ZA1 INTR
RP5 22K
8085A A34
RESETIN
HOLD
X1
X2
READY
TRAP
RST7.5
RST6.5
RST5.5
INTR
SID
SØ
INTA
CLK OUT
SOD
RESETØ
AB A9 A10 A11 A12 A13 A14 A15
PABB PAB9 PABA PABB PABC PABD PABE PABF
ALE
HLDA
AD0 AD1 AD2 AD3 AD4 AD5 AD6 AD7
PDB0 PDB1 PDB2 PDB3 PDB4 PDB5 PDB6 PDB7
S1
WR
RD
IO/M
74S04 A49
PABB-PABF
PDB0 – PDB7
74LS373 AG5
DP8304 AG7
PDB0 PDB1 PDB2 PDB3 PDB4 PDB5 PDB6 PDB7
8D 6D 5D 4D 3D 7D 2D 1D EG OC
8Q 6Q 5Q 4Q 3Q 7Q 2Q 1Q
19 PAB0 15 PAB1 12 PAB2 9 PAB3 6 PAB4 16 PAB5 5 PAB6 2 PAB7
B2 B4 B6 B7 B5 B3 B1 B0
A2 A4 A6 A7 A5 A3 A1 A0
3 IODB0 5 IODB1 7 IODB2 8 IODB3 6 IODB4 4 IODB5 2 IODB6 1 IODB7
CD T/R
IODB0-IODB7 4ZD1, 7ZBB 8ZBB, 9ZBB
ALE 6ZBB
WR/ 6ZBB
RD/ 6ZBB, 4ZDB
IOCMD 4ZCB, 6ZDB
IOR/ 3ZD8, 7ZBB, 8ZCB, 9ZCB
PBRD 3ZCB, 5ZD8
IOW/ 6ZAB, 7ZBB, 8ZCB, 9ZCB
MW/ 3ZD8
AMR/ 3ZBB
AMR 4ZCB
ACMD/ 3ZBB
74S175 A45
74S02 A50
74S06 A72
PZ
28 HALT/
RP6 10K
74LS04 A59
67 68 69
TREF 5ZCB
INTA/ 3ZD8, 7ZBB
RP3, 1K
PULL-UP 2, 6ZB8, 5ZA8
CMD/ 3ZD8
74LS08 A47
74LS74 A43
74LS32 A47
QMW/ 3ZBB
QIOW/ 3ZD8
ALE/ 3ZBB
STB
SEL 1A 2A 3A 3B 1A 1B 4A 4B
1Y 2Y 3Y 4Y
74LS157 A46
1D 2D 3D 4D CLR CLK 1Q 2Q 3Q 4Q
INTØ/ INT1/ INT2/ INT3/ INT4/ INT5/ INT6/ INT7/
P1 41 42 39 40 37 38 35 36
B2 B3 B4 B5 B6 B7 B8 B9
B0 B1

SCALE: NONE	SIZE D	DEPT MCSD	DRAWING NO. 2001695	REV A
SHEET 2 OF				

Figure 5-2. iSBC 544 Schematic Diagram (Sheet 2 of 9)



REFERENCE DESIGNATION	DEVICE TYPE	TOTAL PINS	+5V	+12V	-12V	-5V	GND
A38	74S00	14	14	—	—	—	7
A61	74S133	16	16	—	—	—	8
A32,33,56	8205	16	16	—	—	—	8
A34	8085A	40	40	—	—	—	20
A66,67,75	DP8304	20	20	—	—	—	10
A62	8202	40	—	—	—	—	20
A18-21	8251A	28	26	—	—	—	4
A29	8259	28	28	—	—	—	14
A27,A28	8253	24	24	—	—	—	12
A15	8224	16	—	9	—	—	8
A22	8155	40	40	—	—	—	20
A73,74	8216-4	16	16	—	—	—	8
A76,77	8226	16	16	—	—	—	8
A60,71	DM8097	16	16	—	—	—	8
A25	74S00	14	—	—	—	—	7
A39	74LS00	14	—	—	—	—	7
A72	7406	14	14	—	—	—	7
A53	74S11	14	14	—	—	—	7
A55	74S20	14	14	—	—	—	7
A13	74LS20	14	14	—	—	—	7
A65,78	74LS373	20	20	—	—	—	10
A46	74LS157	16	16	—	—	—	8
A26,36,37	74S74	14	14	—	—	—	7
A23,24,43	74LS74	14	14	—	—	—	7
A49,54	74S04	14	14	—	—	—	7
A31,59	74LS04	14	14	—	—	—	7
A47,57	74LS08	14	14	—	—	—	7
A58	74S02	14	14	—	—	—	7
A40	74S32	14	14	—	—	—	7
A42,48,70	74LS32	14	14	—	—	—	7
A63,64	74LS240	20	20	—	—	—	10
A30	7470	14	14	—	—	—	7
A45	74S175	16	16	—	—	—	8
A17	7492A	14	5	—	—	—	10
A44	74LS109	16	16	—	—	—	8
A41	3625-2	18	18	—	—	—	9
A50	3628	24	24	—	—	—	12
A51,35	2716	24	24	—	—	—	12
A79-86	2117	16	—	—	—	—	16
A6-10,14	1488	14	—	14	1	—	7
A1-5,11,12,16	1489A	14	14	—	—	—	7

Figure 5-2. iSBC 544 Schematic Diagram (Sheet 1 of 9)

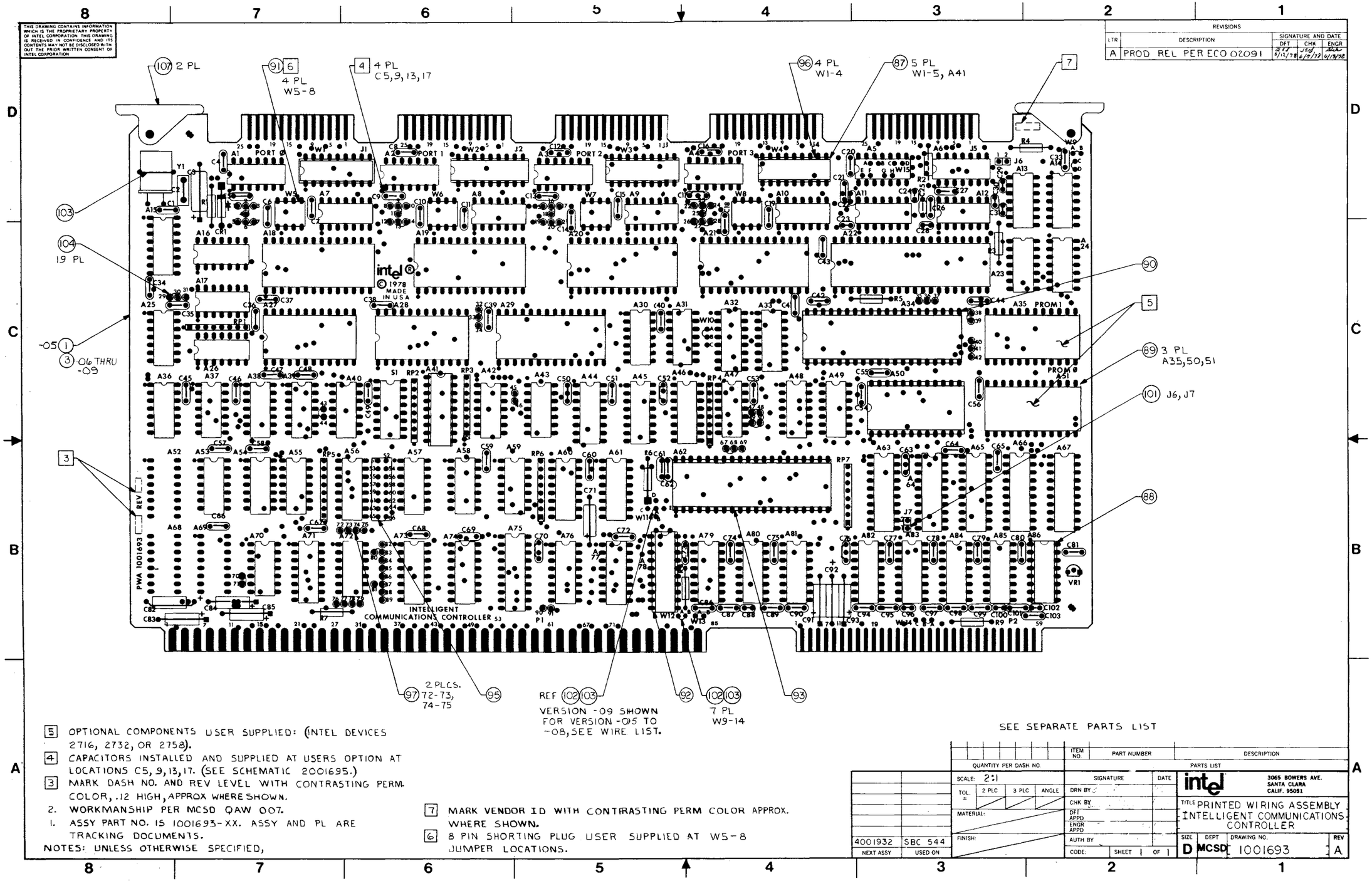
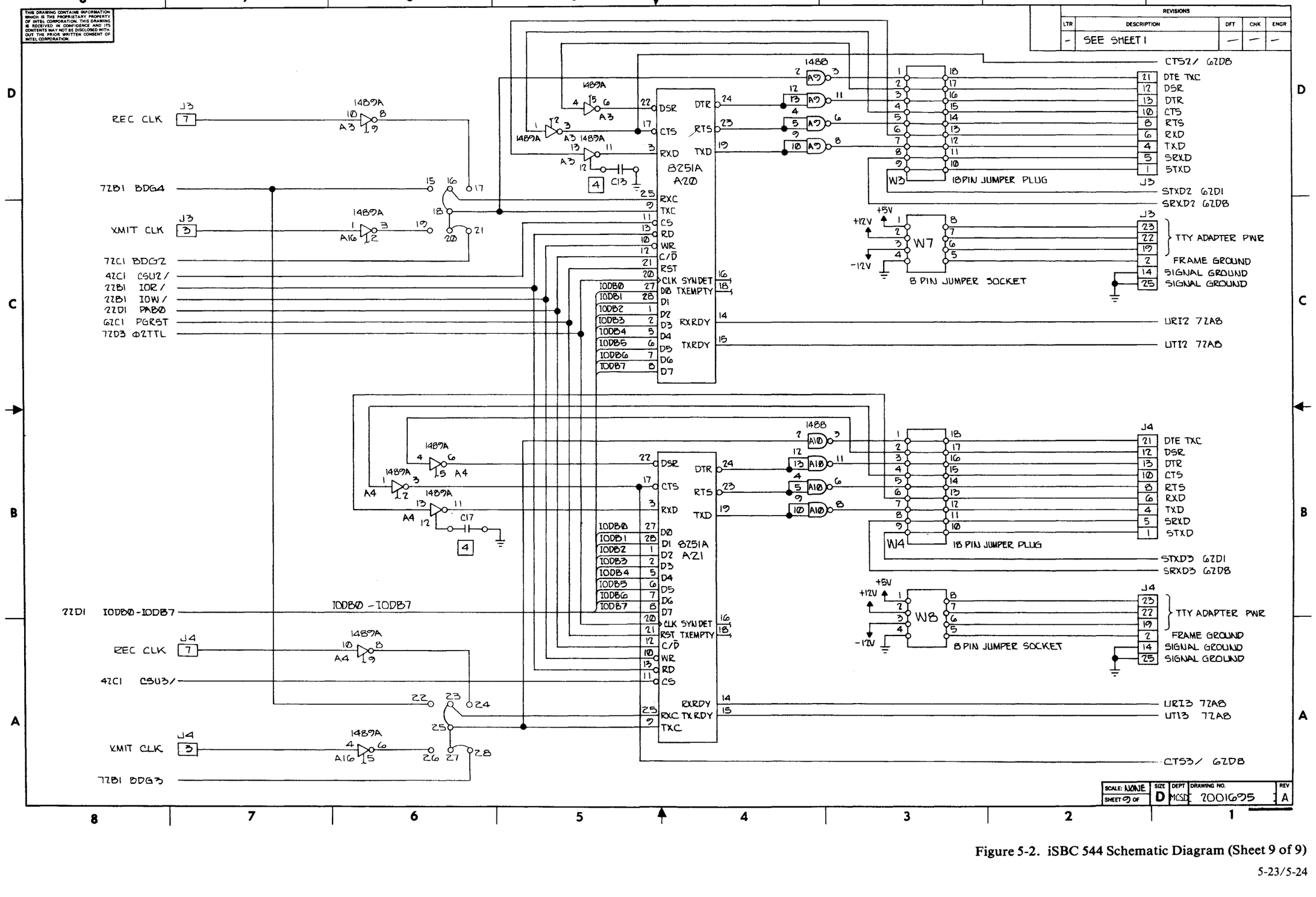


Figure 5-1. iSBC 544 Parts Location Diagram

REVISIONS				
LTR	DESCRIPTION	DFT	CHK	ENGR
-	SEE SHEET 1	—	—	—



SCALE: NONE	SIZE	DEPT	DRAWING NO.	REV
SHEET 9 OF	D	MCSD	7001695	A

Figure 5-2. iSBC 544 Schematic Diagram (Sheet 9 of 9)

APPENDIX A
8085 INSTRUCTION SET

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

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

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

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

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

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

THE 8085 INSTRUCTION SET

The 8085 instruction set includes five different types of instructions:

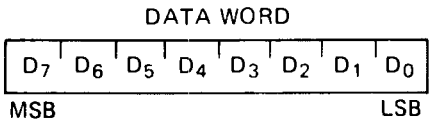
- **Data Transfer Group**—move data between registers or between memory and registers
- **Arithmetic Group** – add, subtract, increment or decrement data in registers or in memory
- **Logical Group** – AND, OR, EXCLUSIVE-OR, compare, rotate or complement data in registers or in memory
- **Branch Group**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

Instruction and Data Formats:

Memory for the 8085 is organized into 8-bit quantities, called Bytes. Each byte has a unique 16-bit binary address corresponding to its sequential position in memory.

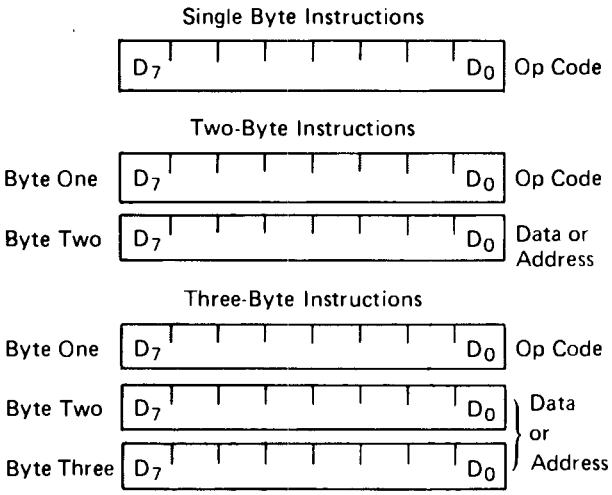
The 8085 can directly address up to 65,536 bytes of memory, which may consist of both read-only memory (ROM) elements and random-access memory (RAM) elements (read/write memory).

Data in the 8085 is stored in the form of 8-bit binary integers:



When a register or data word contains a binary number, it is necessary to establish the order in which the bits of the number are written. In the Intel 8085, BIT 0 is referred to as the **Least Significant Bit (LSB)**, and BIT 7 (of an 8 bit number) is referred to as the **Most Significant Bit (MSB)**.

The 8085 program instructions may be one, two or three bytes in length. Multiple byte instructions must be stored in successive memory locations; the address of the first byte is always used as the address of the instructions. The exact instruction format will depend on the particular operation to be executed.



Addressing Modes:

Often the data that is to be operated on is stored in memory. When multi-byte numeric data is used, the data, like instructions, is stored in successive memory locations, with the least significant byte first, followed by increasingly significant bytes. The 8085 has four different modes for addressing data stored in memory or in registers:

- Direct – Bytes 2 and 3 of the instruction contain the exact memory address of the data item (the low-order bits of the address are in byte 2, the high-order bits in byte 3).
- Register – The instruction specifies the register or register-pair in which the data is located.
- Register Indirect – The instruction specifies a register-pair which contains the memory address where the data is located (the high-order bits of the address are in the first register of the pair, the low-order bits in the second).
- Immediate – The instruction contains the data itself. This is either an 8-bit quantity or a 16-bit quantity (least significant byte first, most significant byte second).

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

- Direct – The branch instruction contains the address of the next instruction to be executed. (Except for the 'RST' instruction, byte 2 contains the low-order address and byte 3 the high-order address.)
- Register indirect – The branch instruction indicates a register-pair which contains the address of the next instruction to be executed. (The high-order bits of the address are in the first register of the pair, the low-order bits in the second.)

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

Condition Flags:

There are five condition flags associated with the execution of instructions on the 8085. They are Zero, Sign, Parity, Carry, and Auxiliary Carry, and are each represented by a 1-bit register in the CPU. A flag is "set" by forcing the bit to 1; "reset" by forcing the bit to 0.

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

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

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

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

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

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

Symbols and Abbreviations:

The following symbols and abbreviations are used in the subsequent description of the 8085 instructions:

SYMBOLS	MEANING
accumulator	Register A
addr	16-bit address quantity
data	8-bit data quantity
data 16	16-bit data quantity
byte 2	The second byte of the instruction
byte 3	The third byte of the instruction
port	8-bit address of an I/O device
r,r1,r2	One of the registers A,B,C,D,E,H,L
DDD,SSS	The bit pattern designating one of the registers A,B,C,D,E,H,L (DDD=destination, SSS=source):

DDD or SSS	REGISTER NAME
111	A
000	B
001	C
010	D
011	E
100	H
101	L

rp	One of the register pairs: B represents the B,C pair with B as the high-order register and C as the low-order register; D represents the D,E pair with D as the high-order register and E as the low-order register; H represents the H,L pair with H as the high-order register and L as the low-order register; SP represents the 16-bit stack pointer register.
RP	The bit pattern designating one of the register pairs B,D,H,SP:

RP	REGISTER PAIR
00	B-C
01	D-E
10	H-L
11	SP

rh	The first (high-order) register of a designated register pair.
rl	The second (low-order) register of a designated register pair.
PC	16-bit program counter register (PCH and PCL are used to refer to the high-order and low-order 8 bits respectively).
SP	16-bit stack pointer register (SPH and SPL are used to refer to the high-order and low-order 8 bits respectively).
r_m	Bit m of the register r (bits are number 7 through 0 from left to right).
Z,S,P,CY,AC	The condition flags: Zero, Sign, Parity, Carry, and Auxiliary Carry, respectively.
( )	The contents of the memory location or registers enclosed in the parentheses.
←	"Is transferred to"
∧	Logical AND
∀	Exclusive OR
∨	Inclusive OR
+	Addition
−	Two's complement subtraction
*	Multiplication
←→	"Is exchanged with"
—	The one's complement (e.g., $(\overline{A})$)
n	The restart number 0 through 7
NNN	The binary representation 000 through 111 for restart number 0 through 7 respectively.

Description Format:

The following pages provide a detailed description of the instruction set of the 8085. Each instruction is described in the following manner:

1. The MCS 85™ macro assembler format, consisting of the instruction mnemonic and operand fields, is printed in **BOLDFACE** on the left side of the first line.
2. The name of the instruction is enclosed in parenthesis on the right side of the first line.
3. The next line(s) contain a symbolic description of the operation of the instruction.
4. This is followed by a narative description of the operation of the instruction.
5. The following line(s) contain the binary fields and patterns that comprise the machine instruction.

6. The last four lines contain incidental information about the execution of the instruction. The number of machine cycles and states required to execute the instruction are listed first. If the instruction has two possible execution times, as in a Conditional Jump, both times will be listed, separated by a slash. Next, any significant data addressing modes (see Page A-2) are listed. The last line lists any of the five Flags that are affected by the execution of the instruction.

Data Transfer Group:

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

MOV r1, r2 (Move Register)

(r1) ← (r2)

The content of register r2 is moved to register r1.

0	1	D	D	D	S	S	S

Cycles: 1
States: 4
Addressing: register
Flags: none

MOV r, M (Move from memory)

(r) ← ((H) (L))

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

0	1	D	D	D	1	1	0

Cycles: 2
States: 7
Addressing: reg. indirect
Flags: none

MOV M, r (Move to memory)

((H) (L)) ← (r)

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

0	1	1	1	0	S	S	S

Cycles: 2
States: 7
Addressing: reg. indirect
Flags: none

MVI r, data (Move Immediate)

(r) ← (byte 2)

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

0	0	D	D	D	1	1	0
data							

Cycles: 2
States: 7
Addressing: immediate
Flags: none

MVI M, data (Move to memory immediate)

((H) (L)) ← (byte 2)

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

0	0	1	1	0	1	1	0
data							

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

LXI rp, data 16 (Load register pair immediate)

(rh) ← (byte 3),

(rl) ← (byte 2)

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

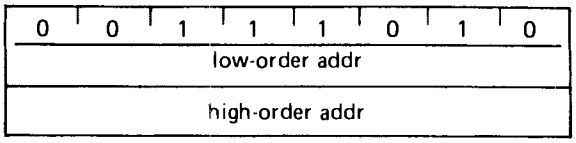
0	0	R	P	0	0	0	1
low-order data							
high-order data							

Cycles: 3
States: 10
Addressing: immediate
Flags: none

LDA addr (Load Accumulator direct)

(A) ← ((byte 3)(byte 2))

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

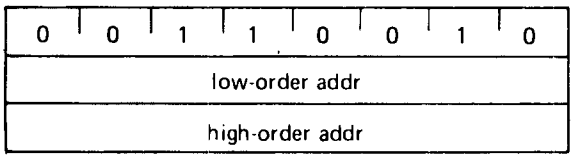


Cycles: 4
States: 13
Addressing: direct
Flags: none

STA addr (Store Accumulator direct)

((byte 3)(byte 2)) ← (A)

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



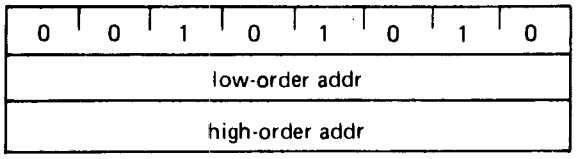
Cycles: 4
States: 13
Addressing: direct
Flags: none

LHLD addr (Load H and L direct)

(L) ← ((byte 3)(byte 2))

(H) ← ((byte 3)(byte 2) + 1)

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



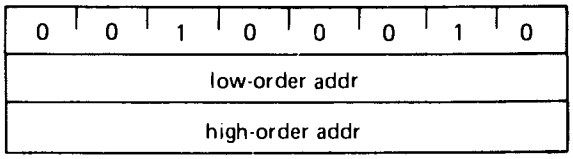
Cycles: 5
States: 16
Addressing: direct
Flags: none

SHLD addr (Store H and L direct)

((byte 3)(byte 2)) ← (L)

((byte 3)(byte 2) + 1) ← (H)

The content of register L is moved to the memory location whose address is specified in byte 2 and byte 3. The content of register H is moved to the succeeding memory location.

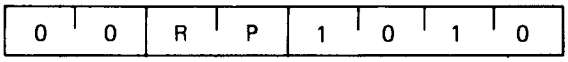


Cycles: 5
States: 16
Addressing: direct
Flags: none

LDAX rp (Load accumulator indirect)

(A) ← ((rp))

The content of the memory location, whose address is in the register pair rp, is moved to register A. Note: only register pairs rp=B (registers B and C) or rp=D (registers D and E) may be specified.

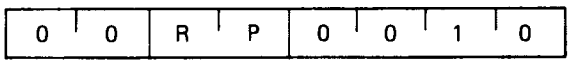


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: none

STAX rp (Store accumulator indirect)

((rp)) ← (A)

The content of register A is moved to the memory location whose address is in the register pair rp. Note: only register pairs rp=B (registers B and C) or rp=D (registers D and E) may be specified.



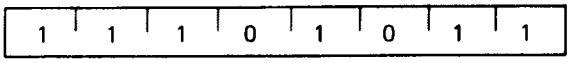
Cycles: 2
States: 7
Addressing: reg. indirect
Flags: none

XCHG (Exchange H and L with D and E)

(H) ←→ (D)

(L) ←→ (E)

The contents of registers H and L are exchanged with the contents of registers D and E.



Cycles: 1
States: 4
Addressing: register
Flags: none

Arithmetic Group:

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

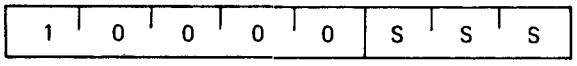
Unless indicated otherwise, all instructions in this group affect the Zero, Sign, Parity, Carry, and Auxiliary Carry flags according to the standard rules.

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

ADD r (Add Register)

(A) ← (A) + (r)

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

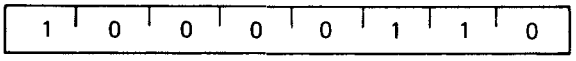


Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

ADD M (Add memory)

(A) ← (A) + ((H) (L))

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

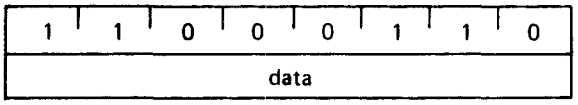


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

ADI data (Add immediate)

(A) ← (A) + (byte 2)

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

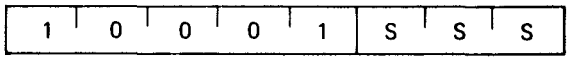


Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

ADC r (Add Register with carry)

(A) ← (A) + (r) + (CY)

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

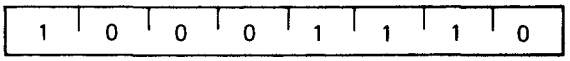


Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

ADC M (Add memory with carry)

(A) ← (A) + ((H) (L)) + (CY)

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

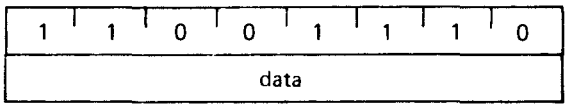


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

ACI data (Add immediate with carry)

(A) ← (A) + (byte 2) + (CY)

The content of the second byte of the instruction and the content of the CY flag are added to the contents of the accumulator. The result is placed in the accumulator.

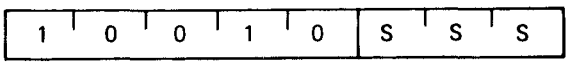


Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

SUB r (Subtract Register)

(A) ← (A) − (r)

The content of register r is subtracted from the content of the accumulator. The result is placed in the accumulator.

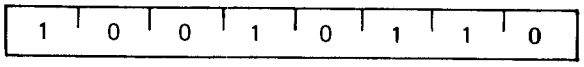


Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

SUB M (Subtract memory)

(A) ← (A) – ((H) (L))

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

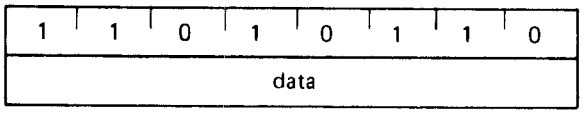


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

SUI data (Subtract immediate)

(A) ← (A) – (byte 2)

The content of the second byte of the instruction is subtracted from the content of the accumulator. The result is placed in the accumulator.

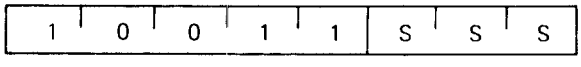


Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

SBB r (Subtract Register with borrow)

(A) ← (A) – (r) – (CY)

The content of register r and the content of the CY flag are both subtracted from the accumulator. The result is placed in the accumulator.

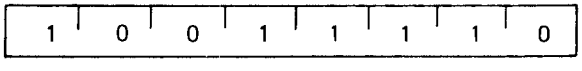


Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

SBB M (Subtract memory with borrow)

(A) ← (A) – ((H) (L)) – (CY)

The content of the memory location whose address is contained in the H and L registers and the content of the CY flag are both subtracted from the accumulator. The result is placed in the accumulator.

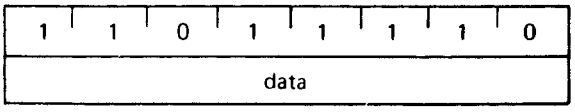


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

SBI data (Subtract immediate with borrow)

(A) ← (A) – (byte 2) – (CY)

The contents of the second byte of the instruction and the contents of the CY flag are both subtracted from the accumulator. The result is placed in the accumulator.

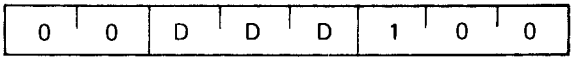


Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

INR r (Increment Register)

(r) ← (r) + 1

The content of register r is incremented by one. Note: All condition flags **except CY** are affected.

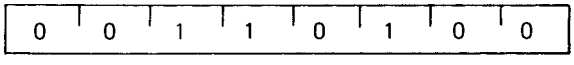


Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,AC

INR M (Increment memory)

((H) (L)) ← ((H) (L)) + 1

The content of the memory location whose address is contained in the H and L registers is incremented by one. Note: All condition flags **except CY** are affected.

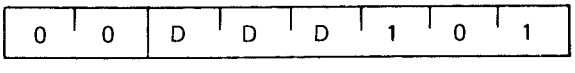


Cycles: 3
States: 10
Addressing: reg. indirect
Flags: Z,S,P,AC

DCR r (Decrement Register)

(r) ← (r) – 1

The content of register r is decremented by one. Note: All condition flags **except CY** are affected.

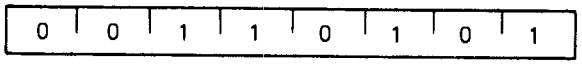


Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,AC

DCR M (Decrement memory)

((H) (L)) ← ((H) (L)) – 1

The content of the memory location whose address is contained in the H and L registers is decremented by one. Note: All condition flags **except CY** are affected.



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

INX rp (Increment register pair)

(rh) (rl) ← (rh) (rl) + 1

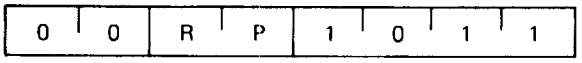
The content of the register pair rp is incremented by one. Note: **No condition flags are affected.**

Cycles: 1
States: 6
Addressing: register
Flags: none

DCX rp (Decrement register pair)

(rh) (rl) ← (rh) (rl) – 1

The content of the register pair rp is decremented by one. Note: **No condition flags are affected.**

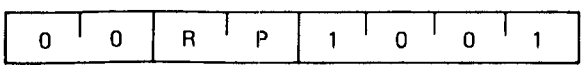


Cycles: 1
States: 6
Addressing: register
Flags: none

DAD rp (Add register pair to H and L)

(H) (L) ← (H) (L) + (rh) (rl)

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



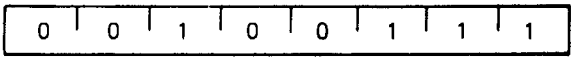
Cycles: 3
States: 10
Addressing: register
Flags: CY

DAA (Decimal Adjust Accumulator)

The eight-bit number in the accumulator is adjusted to form two four-bit Binary-Coded-Decimal digits by the following process:

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

NOTE: All flags are affected.



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

Logical Group:

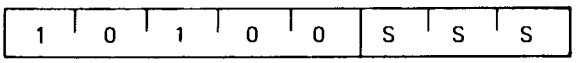
This group of instructions performs logical (Boolean) operations on data in registers and memory and on condition flags.

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

ANA r (AND Register)

(A) ← (A) ∧ (r)

The content of register r is logically anded with the content of the accumulator. The result is placed in the accumulator. **The CY flag is cleared and AC is set.**

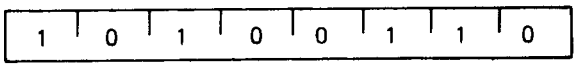


Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

ANA M (AND memory)

(A) ← (A) ∧ ((H) (L))

The contents of the memory location whose address is contained in the H and L registers is logically anded with the content of the accumulator. The result is placed in the accumulator. **The CY flag is cleared and AC is set.**

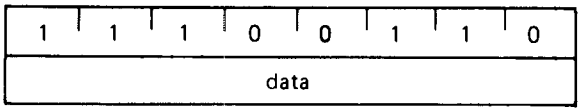


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

ANI data (AND immediate)

(A) ← (A) ∧ (byte 2)

The content of the second byte of the instruction is logically anded with the contents of the accumulator. The result is placed in the accumulator. **The CY flag is cleared and AC is set.**

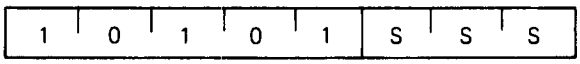


Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

XRA r (Exclusive OR Register)

(A) ← (A) ∀ (r)

The content of register r is exclusive-or'd with the content of the accumulator. The result is placed in the accumulator. **The CY and AC flags are cleared.**

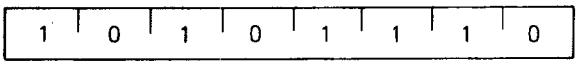


Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

XRA M (Exclusive OR Memory)

(A) ← (A) ∀ ((H) (L))

The content of the memory location whose address is contained in the H and L registers is exclusive-OR'd with the content of the accumulator. The result is placed in the accumulator. **The CY and AC flags are cleared.**

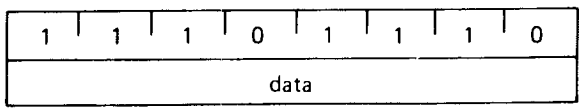


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

XRI data (Exclusive OR immediate)

(A) ← (A) ∀ (byte 2)

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

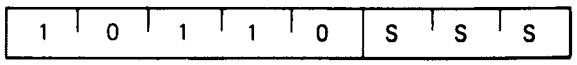


Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

ORA r (OR Register)

(A) ← (A) V (r)

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

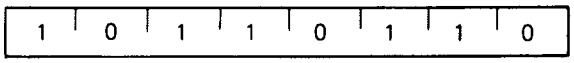


Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

ORA M (OR memory)

(A) ← (A) V ((H) (L))

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

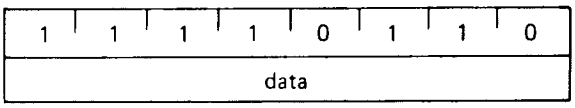


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

ORI data (OR Immediate)

(A) ← (A) V (byte 2)

The content of the second byte of the instruction is inclusive-OR'd with the content of the accumulator. The result is placed in the accumulator. **The CY and AC flags are cleared.**

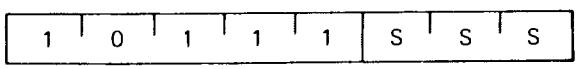


Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

CMP r (Compare Register)

(A) − (r)

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

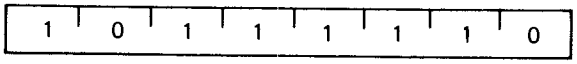


Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

CMP M (Compare memory)

(A) − ((H) (L))

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

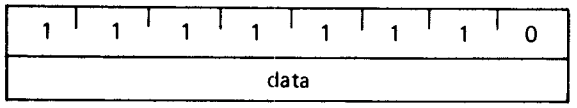


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

CPI data (Compare immediate)

(A) − (byte 2)

The content of the second byte of the instruction is subtracted from the accumulator. The condition flags are set by the result of the subtraction. The Z flag is set to 1 if (A) = (byte 2). The CY flag is set to 1 if (A) < (byte 2).



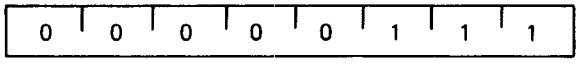
Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

RLC (Rotate left)

$(A_{n+1}) \leftarrow (A_n)$; $(A_0) \leftarrow (A_7)$

$(CY) \leftarrow (A_7)$

The content of the accumulator is rotated left one position. The low order bit and the CY flag are both set to the value shifted out of the high order bit position. **Only the CY flag is affected.**



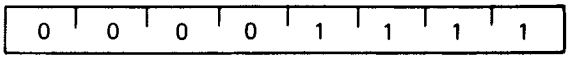
Cycles: 1
States: 4
Flags: CY

RRC (Rotate right)

$(A_n) \leftarrow (A_{n-1})$; $(A_7) \leftarrow (A_0)$

$(CY) \leftarrow (A_0)$

The content of the accumulator is rotated right one position. The high order bit and the CY flag are both set to the value shifted out of the low order bit position. **Only the CY flag is affected.**



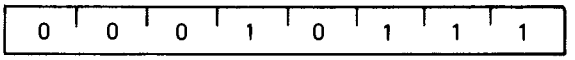
Cycles: 1
States: 4
Flags: CY

RAL (Rotate left through carry)

$(A_{n+1}) \leftarrow (A_n)$; $(CY) \leftarrow (A_7)$

$(A_0) \leftarrow (CY)$

The content of the accumulator is rotated left one position through the CY flag. The low order bit is set equal to the CY flag and the CY flag is set to the value shifted out of the high order bit. **Only the CY flag is affected.**



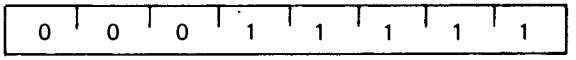
Cycles: 1
States: 4
Flags: CY

RAR (Rotate right through carry)

$(A_n) \leftarrow (A_{n+1})$; $(CY) \leftarrow (A_0)$

$(A_7) \leftarrow (CY)$

The content of the accumulator is rotated right one position through the CY flag. The high order bit is set to the CY flag and the CY flag is set to the value shifted out of the low order bit. **Only the CY flag is affected.**

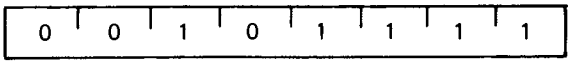


Cycles: 1
States: 4
Flags: CY

CMA (Complement accumulator)

$(A) \leftarrow (\overline{A})$

The contents of the accumulator are complemented (zero bits become 1, one bits become 0). **No flags are affected.**

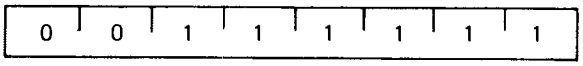


Cycles: 1
States: 4
Flags: none

CMC (Complement carry)

(CY) ← $(\overline{CY})$

The CY flag is complemented. **No other flags are affected.**



Cycles: 1
States: 4
Flags: CY

STC (Set carry)

(CY) ← 1

The CY flag is set to 1. **No other flags are affected.**

Cycles: 1
States: 4
Flags: CY

Branch Group:

This group of instructions alter normal sequential program flow.

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

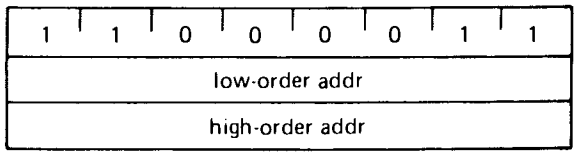
The two types of branch instructions are unconditional and conditional. Unconditional transfers simply perform the specified operation on register PC (the program counter). Conditional transfers examine the status of one of the four processor flags to determine if the specified branch is to be executed.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

CONDITION	CCC
NZ – not zero (Z = 0)	000
Z – zero (Z = 1)	001
NC – no carry (CY = 0)	010
C – carry (CY = 1)	011
PO – parity odd (P = 0)	100
PE – parity even (P = 1)	101
P – plus (S = 0)	110
M – minus (S = 1)	111

JMP addr (Jump)

(PC) ← (byte 3) (byte 2)

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



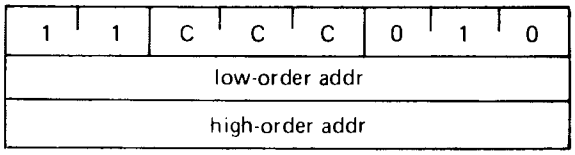
Cycles: 3
States: 10
Addressing: immediate
Flags: none

Jcondition addr (Conditional jump)

If (CCC),

(PC) ← (byte 3) (byte 2)

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



Cycles: **2/3**
States: **7/10**
Addressing: immediate
Flags: none

CALL addr (Call)

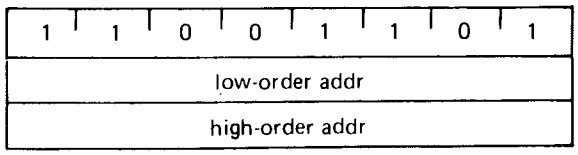
((SP) – 1) ← (PCH)

((SP) – 2) ← (PCL)

(SP) ← (SP) – 2

(PC) ← (byte 3) (byte 2)

The high-order eight bits of the next instruction address are moved to the memory location whose address is one less than the content of register SP. The low-order eight bits of the next instruction address are moved to the memory location whose address is two less than the content of register SP. The content of register SP is decremented by 2. Control is transferred to the instruction whose address is specified in byte 3 and byte 2 of the current instruction.

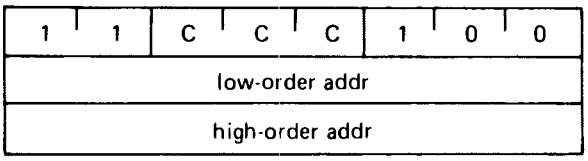


Cycles: 5
States: **18**
Addressing: immediate/reg. indirect
Flags: none

Ccondition addr (Condition call)

If (CCC),
(SP) − 1) ← (PCH)
((SP) − 2) ← (PCL)
(SP) ← (SP) − 2
(PC) ← (byte 3) (byte 2)

If the specified condition is true, the actions specified in the CALL instruction (see above) are performed; otherwise, control continues sequentially.

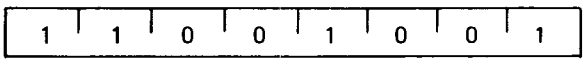


Cycles: 2/5
States: 9/18
Addressing: immediate/reg. indirect
Flags: none

RET (Return)

(PCL) ← ((SP));
(PCH) ← ((SP) + 1);
(SP) ← (SP) + 2;

The content of the memory location whose address is specified in register SP is moved to the low-order eight bits of register PC. The content of the memory location whose address is one more than the content of register SP is moved to the high-order eight bits of register PC. The content of register SP is incremented by 2.

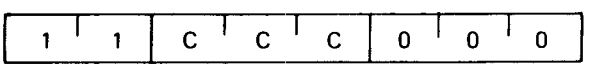


Cycles: 3
States: 10
Addressing: reg. indirect
Flags: none

Rcondition (Conditional return)

If (CCC),
(PCL) ← ((SP))
(PCH) ← ((SP) + 1)
(SP) ← (SP) + 2

If the specified condition is true, the actions specified in the RET instruction (see above) are performed; otherwise, control continues sequentially.



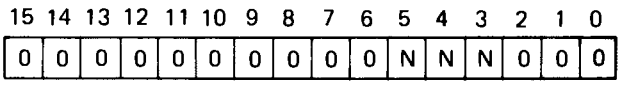
Cycles: 1/3
States: 6/12
Addressing: reg. indirect
Flags: none

RST n (Restart)

((SP) − 1) ← (PCH)
((SP) − 2) ← (PCL)
(SP) ← (SP) − 2
(PC) ← 8 * (NNN)

The high-order eight bits of the next instruction address are moved to the memory location whose address is one less than the content of register SP. The low-order eight bits of the next instruction address are moved to the memory location whose address is two less than the content of register SP. The content of register SP is decremented by two. Control is transferred to the instruction whose address is eight times the content of NNN.

Cycles: 3
States: 12
Addressing: reg. indirect
Flags: none

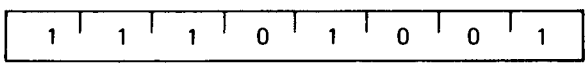


Program Counter After Restart

PCHL (Jump H and L indirect – move H and L to PC)

(PCH) ← (H)
(PCL) ← (L)

The content of register H is moved to the high-order eight bits of register PC. The content of register L is moved to the low-order eight bits of register PC.



Cycles: 1
States: 6
Addressing: register
Flags: none

Stack, I/O, and Machine Control Group:

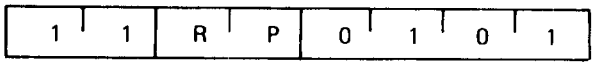
This group of instructions performs I/O, manipulates the Stack, and alters internal control flags.

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

PUSH rp (Push)

((SP) − 1) ← (rh)
((SP) − 2) ← (rl)
(SP) ← (SP) − 2

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



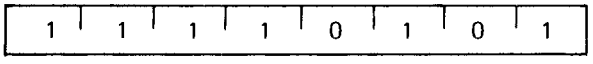
Cycles: 3
States: 12
Addressing: reg. indirect
Flags: none

PUSH PSW (Push processor status word)

((SP) − 1) ← (A)
$((SP) - 2)_0$ ← (CY), $((SP) - 2)_1$ ← X
$((SP) - 2)_2$ ← (P), $((SP) - 2)_3$ ← X
$((SP) - 2)_4$ ← (AC), $((SP) - 2)_5$ ← X
$((SP) - 2)_6$ ← (Z), $((SP) - 2)_7$ ← (S)
(SP) ← (SP) − 2

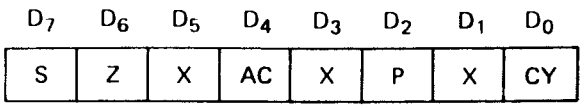
X: Undefined.

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



Cycles: 3
States: 12
Addressing: reg. indirect
Flags: none

FLAG WORD



X: undefined

POP rp (Pop)

(rl) ← ((SP))
(rh) ← ((SP) + 1)
(SP) ← (SP) + 2

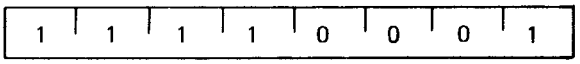
The content of the memory location, whose address is specified by the content of register SP, is moved to the low-order register of register pair rp. The content of the memory location, whose address is one more than the content of register SP, is moved to the high-order register of register pair rp.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

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

POP PSW (Pop processor status word)

(CY) ← $((SP))_0$
(P) ← $((SP))_2$
(AC) ← $((SP))_4$
(Z) ← $((SP))_6$
(S) ← $((SP))_7$
(A) ← ((SP) + 1)
(SP) ← (SP) + 2

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

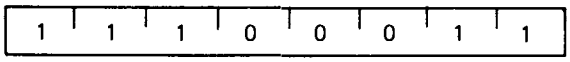


Cycles: 3
States: 10
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

XTHL (Exchange stack top with H and L)

(L) ⟷ ((SP))
(H) ⟷ ((SP) + 1)

The content of the L register is exchanged with the content of the memory location whose address is specified by the content of register SP. The content of the H register is exchanged with the content of the memory location whose address is one more than the content of register SP.

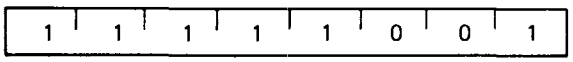


Cycles: 5
States: 16
Addressing: reg. indirect
Flags: none

SPHL (Move HL to SP)

(SP) ← (H) (L)

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

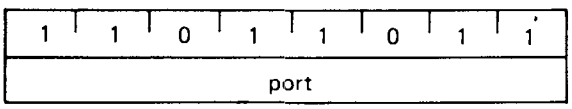


Cycles: 1
States: 6
Addressing: register
Flags: none

IN port (Input)

(A) ← (data)

The data placed on the eight bit bi-directional data bus by the specified port is moved to register A.

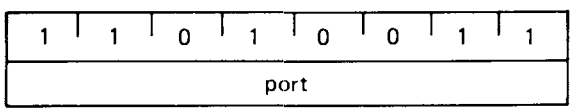


Cycles: 3
States: 10
Addressing: direct
Flags: none

OUT port (Output)

(data) ← (A)

The content of register A is placed on the eight bit bi-directional data bus for transmission to the specified port.



Cycles: 3
States: 10
Addressing: direct
Flags: none

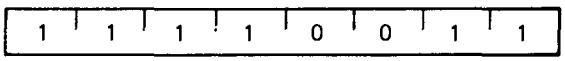
EI (Enable interrupts)

The interrupt system is enabled **following the execution of the next instruction.**

Cycles: 1
States: 4
Flags: none

DI (Disable interrupts)

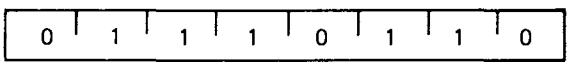
The interrupt system is disabled **immediately following the execution of the DI instruction.**



Cycles: 1
States: 4
Flags: none

HLT (Halt)

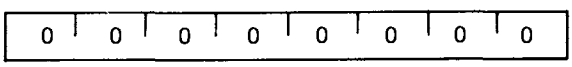
The processor is stopped. The registers and flags are unaffected.



Cycles: 1
States: 5
Flags: none

NOP (No op)

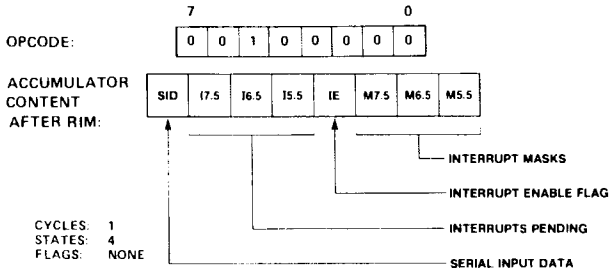
No operation is performed. The registers and flags are unaffected.



Cycles: 1
States: 4
Flags: none

RIM (Read Interrupt Mask)

After the execution of the RIM instruction, the accumulator is loaded with the restart interrupt masks, any pending interrupts, and the contents of the serial input data line (SID).



SIM (Set Interrupt Masks)

During execution of the SIM instruction, the contents of the accumulator will be used in programming the restart interrupt masks. Bits 0-2 will set/reset the mask bit for RST 5.5, 6.5, 7.5 of the interrupt mask register, if bit 3 is 1 ("set"). Bit 3 is a "Mask Set Enable" control.

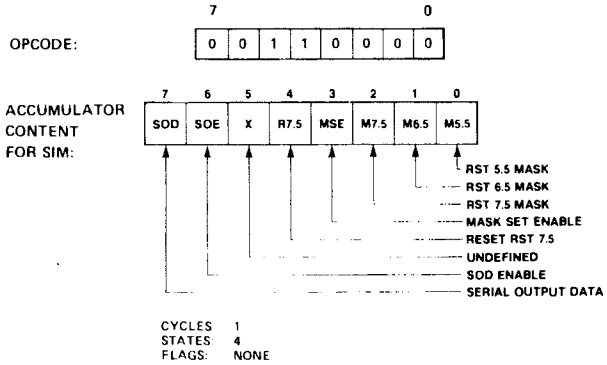
Setting the mask (i.e. masked bit = 1) **disables** the corresponding interrupt.

	Set	Reset
RST 5.5 MASK	if bit 0 = 1	if bit 0 = 0
RST 6.5 MASK	bit 1 = 1	bit 1 = 0
RST 7.5 MASK	bit 2 = 1	bit 2 = 0

RST 7.5 (edge trigger enable) internal request flip flop will be reset if bit 4 of the accumulator = 1; regardless of whether RST 7.5 is masked or not.

$\overline{\text{RESET IN}}$ input (pin 36) will set all RST MASKs, and reset/disable all interrupts.

SIM can, also, load the SOD output latch. Accumulator bit 7 is loaded into the SOD latch if bit 6 is set. The latch is unaffected if bit 6 is a zero. $\overline{\text{RESET IN}}$ input sets the SOD latch to zero.



INSTRUCTION SET

Summary of Processor Instructions

Mnemonic	Description	Instruction Code[1] D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Clock[2] Cycles
$MOV_{r1,r2}$	Move register to register	0	1	D	D	D	S	S	S	5
MOV M, r	Move register to memory	0	1	1	1	0	S	S	S	7
MOV r, M	Move memory to register	0	1	D	D	D	1	1	0	7
HLT	Halt	0	1	1	1	0	1	1	0	7
MVI r	Move immediate register	0	0	D	D	D	1	1	0	7
MVI M	Move immediate memory	0	0	1	1	0	1	1	0	10
INR r	Increment register	0	0	D	D	D	1	0	0	5
DCR r	Decrement register	0	0	D	D	D	1	0	1	5
INR M	Increment memory	0	0	1	1	0	1	0	0	10
DCR M	Decrement memory	0	0	1	1	0	1	0	1	10
ADD r	Add register to A	1	0	0	0	0	S	S	S	4
ADC r	Add register to A with carry	1	0	0	0	1	S	S	S	4
SUB r	Subtract register from A	1	0	0	1	0	S	S	S	4
SBB r	Subtract register from A with borrow	1	0	0	1	1	S	S	S	4
ANA r	And register with A	1	0	1	0	0	S	S	S	4
XRA r	Exclusive Or register with A	1	0	1	0	1	S	S	S	4
ORA r	Or register with A	1	0	1	1	0	S	S	S	4
CMP r	Compare register with A	1	0	1	1	1	S	S	S	4
ADD M	Add memory to A	1	0	0	0	0	1	1	0	7
ADC M	Add memory to A with carry	1	0	0	0	1	1	1	0	7
SUB M	Subtract memory from A	1	0	0	1	0	1	1	0	7
SBB M	Subtract memory from A with borrow	1	0	0	1	1	1	1	0	7
ANA M	And memory with A	1	0	1	0	0	1	1	0	7
XRA M	Exclusive Or memory with A	1	0	1	0	1	1	1	0	7
ORA M	Or memory with A	1	0	1	1	0	1	1	0	7
CMP M	Compare memory with A	1	0	1	1	1	1	1	0	7
ADI	Add immediate to A	1	1	0	0	0	1	1	0	7
ACI	Add immediate to A with carry	1	1	0	0	1	1	1	0	7
SUI	Subtract immediate from A	1	1	0	1	0	1	1	0	7
SBI	Subtract immediate from A with borrow	1	1	0	1	1	1	1	0	7
ANI	And immediate with A	1	1	1	0	0	1	1	0	7
XRI	Exclusive Or immediate with A	1	1	1	0	1	1	1	0	7
ORI	Or immediate with A	1	1	1	1	0	1	1	0	7
CPI	Compare immediate with A	1	1	1	1	1	1	1	0	7
RLC	Rotate A left	0	0	0	0	0	1	1	1	4
RRC	Rotate A right	0	0	0	0	1	1	1	1	4
RAL	Rotate A left through carry	0	0	0	1	0	1	1	1	4
RAR	Rotate A right through carry	0	0	0	1	1	1	1	1	4
JMP	Jump unconditional	1	1	0	0	0	0	1	1	10
JC	Jump on carry	1	1	0	1	1	0	1	0	10
JNC	Jump on no carry	1	1	0	1	0	0	1	0	10
JZ	Jump on zero	1	1	0	0	1	0	1	0	10
JNZ	Jump on no zero	1	1	0	0	0	0	1	0	10
JP	Jump on positive	1	1	1	1	0	0	1	0	10
JM	Jump on minus	1	1	1	1	1	0	1	0	10
JPE	Jump on parity even	1	1	1	0	1	0	1	0	10
JPO	Jump on parity odd	1	1	1	0	0	0	1	0	10
CALL	Call unconditional	1	1	0	0	1	1	0	1	17
CC	Call on carry	1	1	0	1	1	1	0	0	11/17
CNC	Call on no carry	1	1	0	1	0	1	0	0	11/17
CZ	Call on zero	1	1	0	0	1	1	0	0	11/17
CNZ	Call on no zero	1	1	0	0	0	1	0	0	11/17
CP	Call on positive	1	1	1	1	0	1	0	0	11/17
CM	Call on minus	1	1	1	1	1	1	0	0	11/17
CPE	Call on parity even	1	1	1	0	1	1	0	0	11/17
CPO	Call on parity odd	1	1	1	0	0	1	0	0	11/17
RET	Return	1	1	0	0	1	0	0	1	10
RC	Return on carry	1	1	0	1	1	0	0	0	5/11
RNC	Return on no carry	1	1	0	1	0	0	0	0	5/11
RZ	Return on zero	1	1	0	0	1	0	0	0	5/11
RNZ	Return on no zero	1	1	0	0	0	0	0	0	5/11
RP	Return on positive	1	1	1	1	0	0	0	0	5/11
RM	Return on minus	1	1	1	1	1	0	0	0	5/11
RPE	Return on parity even	1	1	1	0	1	0	0	0	5/11
RPO	Return on parity odd	1	1	1	0	0	0	0	0	5/11
RST	Restart	1	1	A	A	A	1	1	1	11
IN	Input	1	1	0	1	1	0	1	1	10
OUT	Output	1	1	0	1	0	0	1	1	10
LXI B	Load immediate register Pair B & C	0	0	0	0	0	0	0	1	10
LXI D	Load immediate register Pair D & E	0	0	0	1	0	0	0	1	10
LXI H	Load immediate register Pair H & L	0	0	1	0	0	0	0	1	10
LXI SP	Load immediate stack pointer	0	0	1	1	0	0	0	1	10
PUSH B	Push register Pair B & C on stack	1	1	0	0	0	1	0	1	11
PUSH D	Push register Pair D & E on stack	1	1	0	1	0	1	0	1	11
PUSH H	Push register Pair H & L on stack	1	1	1	0	0	1	0	1	11
PUSH PSW	Push A and Flags on stack	1	1	1	1	0	1	0	1	11
POP B	Pop register pair B & C off stack	1	1	0	0	0	0	0	1	10
POP D	Pop register pair D & E off stack	1	1	0	1	0	0	0	1	10
POP H	Pop register pair H & L off stack	1	1	1	0	0	0	0	1	10
POP PSW	Pop A and Flags off stack	1	1	1	1	0	0	0	1	10
STA	Store A direct	0	0	1	1	0	0	1	0	13
LDA	Load A direct	0	0	1	1	1	0	1	0	13
XCHG	Exchange D & E, H & L Registers	1	1	1	0	1	0	1	1	4
XTHL	Exchange top of stack, H & L	1	1	1	0	0	0	1	1	18
SPHL	H & L to stack pointer	1	1	1	1	1	0	0	1	5
PCHL	H & L to program counter	1	1	1	0	1	0	0	1	5
DAD B	Add B & C to H & L	0	0	0	0	1	0	0	1	10
DAD D	Add D & E to H & L	0	0	0	1	1	0	0	1	10
DAD H	Add H & L to H & L	0	0	1	0	1	0	0	1	10
DAD SP	Add stack pointer to H & L	0	0	1	1	1	0	0	1	10
STAX B	Store A indirect	0	0	0	0	0	0	1	0	7
STAX D	Store A indirect	0	0	0	1	0	0	1	0	7
LDAX B	Load A indirect	0	0	0	0	1	0	1	0	7
LDAX D	Load A indirect	0	0	0	1	1	0	1	0	7
INX B	Increment B & C registers	0	0	0	0	0	0	1	1	5
INX D	Increment D & E registers	0	0	0	1	0	0	1	1	5
INX H	Increment H & L registers	0	0	1	0	0	0	1	1	5
INX SP	Increment stack pointer	0	0	1	1	0	0	1	1	5
DCX B	Decrement B & C	0	0	0	0	1	0	1	1	5
DCX D	Decrement D & E	0	0	0	1	1	0	1	1	5
DCX H	Decrement H & L	0	0	1	0	1	0	1	1	5
DCX SP	Decrement stack pointer	0	0	1	1	1	0	1	1	5
CMA	Complement A	0	0	1	0	1	1	1	1	4
STC	Set carry	0	0	1	1	0	1	1	1	4
CMC	Complement carry	0	0	1	1	1	1	1	1	4
DAA	Decimal adjust A	0	0	1	0	0	1	1	1	4
SHLD	Store H & L direct	0	0	1	0	0	0	1	0	16
LHLD	Load H & L direct	0	0	1	0	1	0	1	0	16
EI	Enable Interrupts	1	1	1	1	1	0	1	1	4
DI	Disable interrupt	1	1	1	1	0	0	1	1	4
NOP	No-operation	0	0	0	0	0	0	0	0	4
RIM	Read Interrupt Mask	0	0	1	0	0	0	0	0	4
SIM	Set Interrupt Mask	0	0	1	1	0	0	0	0	4

NOTES: 1. DDD or SSS – 000 B – 001 C – 010 D – 011 E – 100 H – 101 L – 110 Memory – 111 A.
2. Two possible cycle times, (5/11) indicate instruction cycles dependent on condition flags.

APPENDIX B
TELETYPEWRITER MODIFICATIONS

B-1. INTRODUCTION

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

B-2. INTERNAL MODIFICATIONS

WARNING

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

Remove the top cover and modify the teletypewriter as follows:

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

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

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

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

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

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

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

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

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

B-3. EXTERNAL CONNECTIONS

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

B-4. SBC 530 TTY ADAPTER

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

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

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

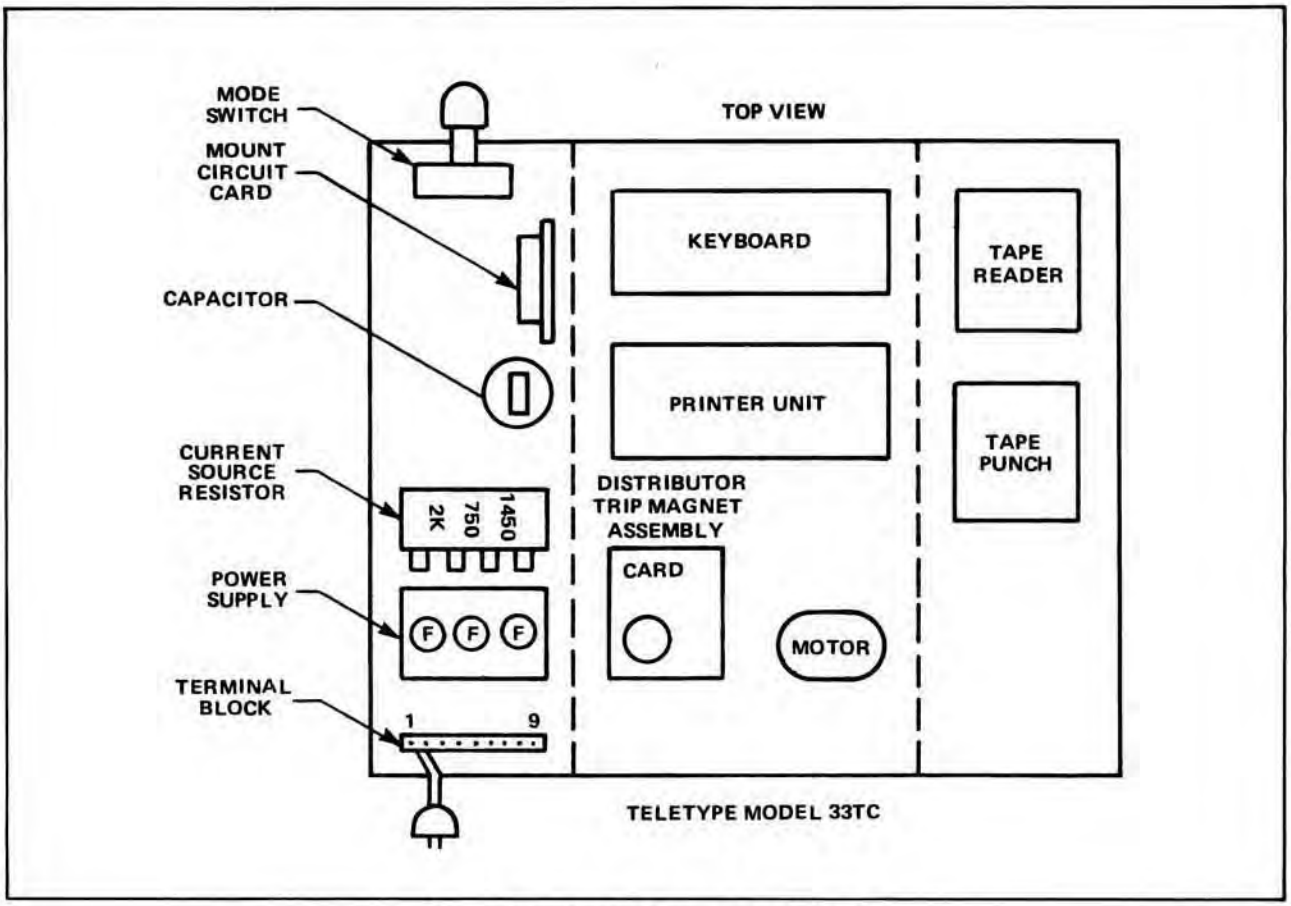


Figure B-1. Teletype Component Layout

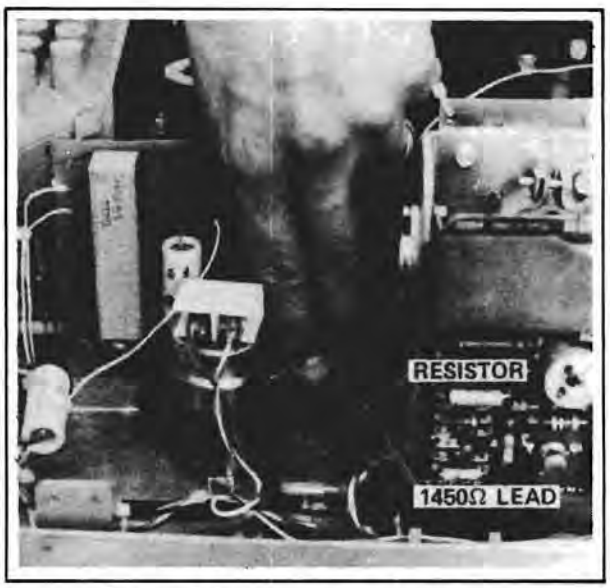


Figure B-2. Current Source Resistor

Figure B-3. Terminal Block

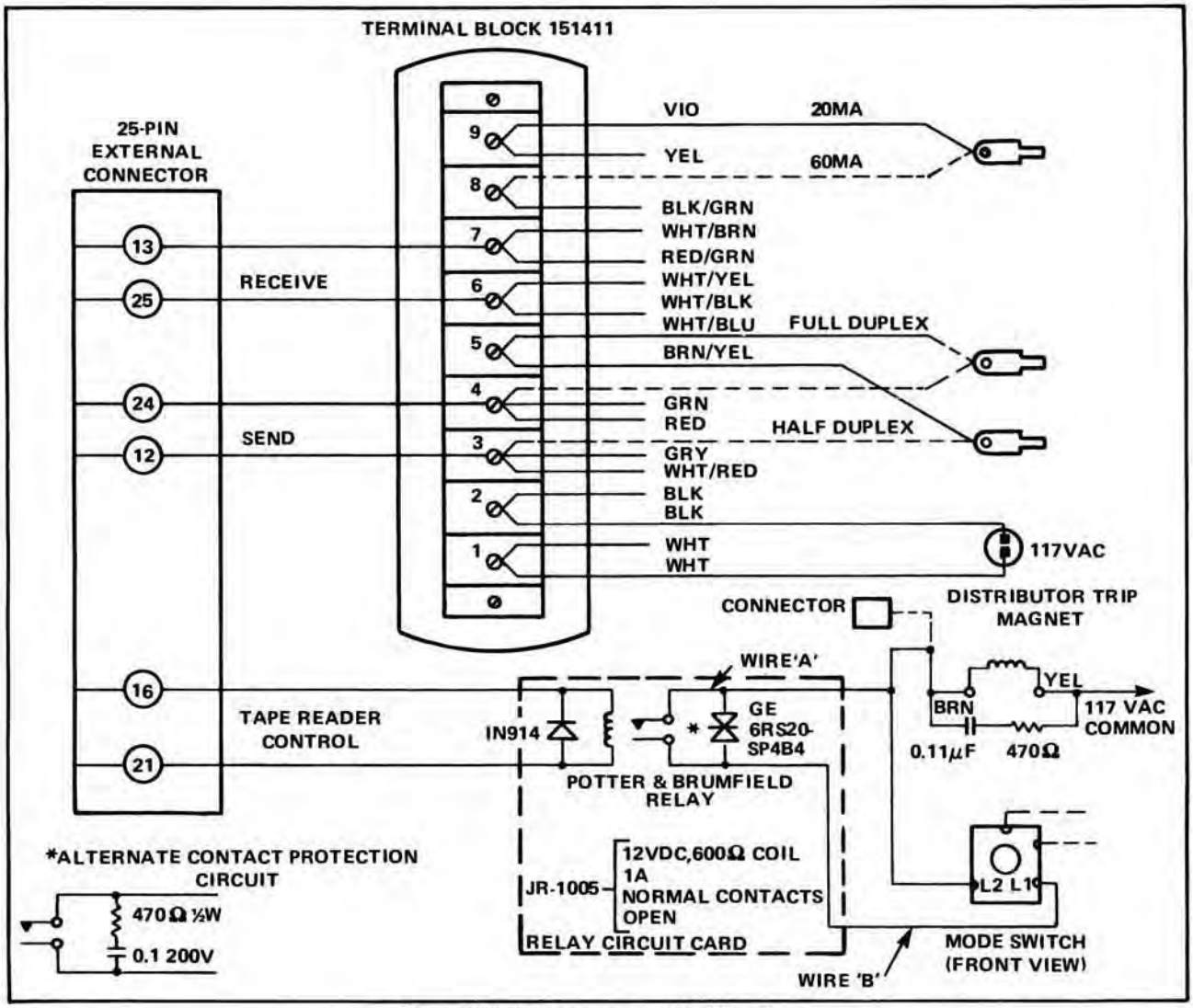


Figure B-4. Teletypewriter Modifications

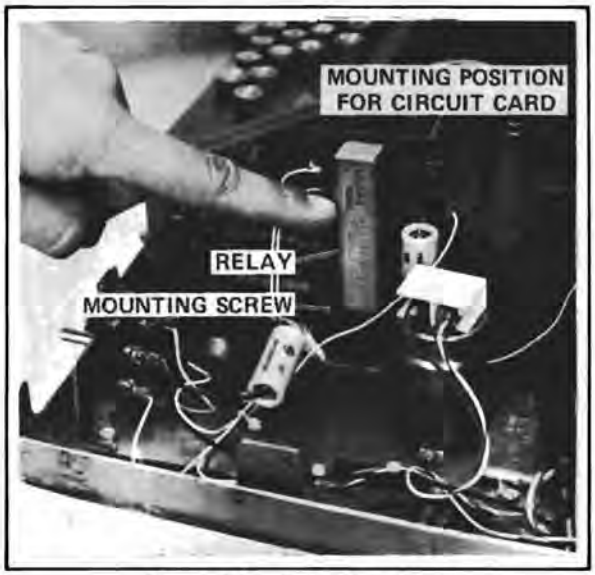


Figure B-5. Relay Circuit

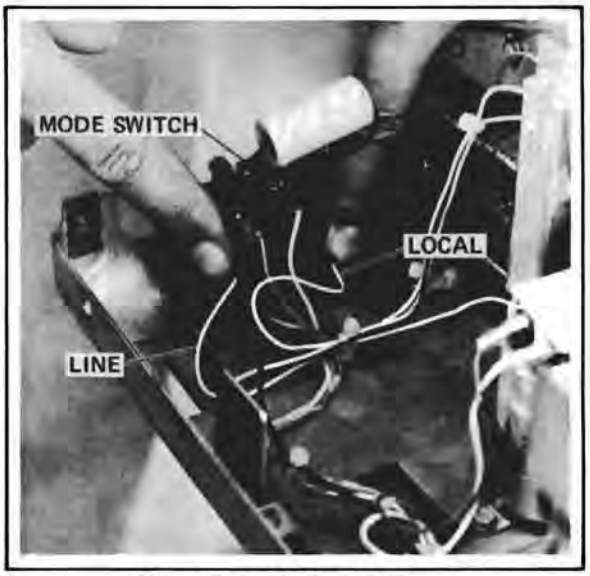


Figure B-6. Mode Switch

Figure B-7. Distributor Trip Magnet

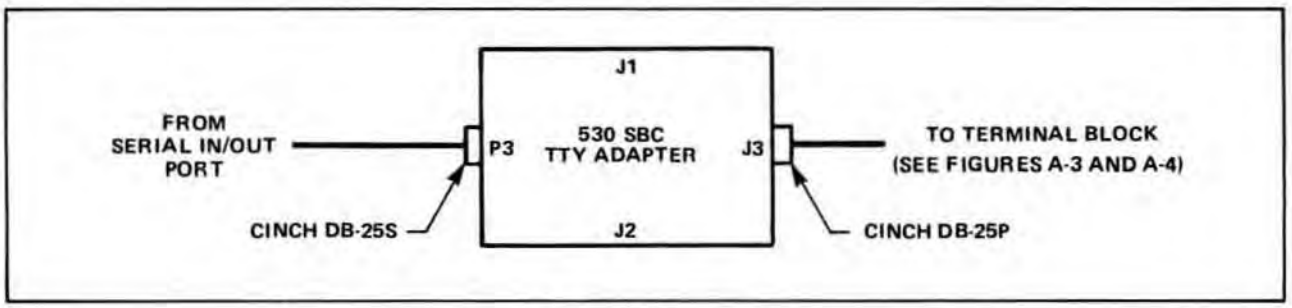


Figure B-8. TTY Adapter Cabling

APPENDIX C
CUSTOM PROGRAMMED PROMS

C-1. INTRODUCTION

This appendix provides information about two custom programmed PROMS used on the iSBC 544. One is used for on-board Chip Select and Address Selection (A50), and the other is used for Address Transformation (A41)and off-board Address Selection. The components can be referenced on figure 5-2, Sheet 4 while reading this information.

C-2. CHIP SELECT PROM

The SBC-544 has two blocks of RAM, up to 2 EPROMS, and 8 Peripheral Chips. These functional blocks inter-connect with several on-board buses. It is the job of this PROM to select the proper block, send control signals to the Bus Enable logic, and generate an acknowledge if required.

There are several versions of this PROM which are used to configure the SBC-544 in various forms. These PROMS differ only in the size of and location of the I/O, RAM, and EPROM blocks.

This version is for 16K RAM and 2K or 4K of EPROM.

The PROM used is an Intel 3628, 1K by 8 Bipolar PROM. As such, it has 10 address inputs and 8 outputs. Eight of the inputs (PROM address lines A0 thru A7) connect to the upper 8 CPU address lines so that the PROM may select the proper chips. PROM address input A9 is the IO/M signal from the 8085A to tell the PROM if the current access is to memory or IO. Input A8 is connected to an option switch (S1 section 7) which allows the user to select 1 of 2 possible sizes for the EPROM block.

There are 15 chip select lines required. These outputs are encoded on the lower 4 bits of the output (01, 02, 03, & 04). This encoded output is decoded by 2 1-of-8 decoders connected as a 1-of-16 decoder. Table C-1 shows the relation of encoded chip select output and the function enable. Output 08 is used to indicate that the current access is to on-board I/O or memory. This signal is active-low. Output of 07 indicates that the current access is an access to on-board I/O or EPROM (active-low). This output is used to generate an acknowledge to the CPU. Output 06 indicates that the current access should take place over the I/O data bus (also active-low). Output 05 is not used. Table C-2 shows the relation of input addresses to outputs. For all addresses not shown, the outputs are inactive (all ones).

Table C-1. Chip Select Coding

CS Output	Function Enabled
0	EPROM 0
1	EPROM 1
2	Reserved
3	Reserved
4	USART 0
5	USART 1 } 8251As
6	USART 2
7	USART 3
8	TIMER 0 } 8253 PITs
9	TIMER 1
A	Parallel Port/Static RAM } 8155 PPI
B	Set Master Mode
C	Reset Master Mode }
D	Interrupt Control } 8259 PIC
E	Dynamic RAM
F	None

Table C-2. Chip Select Addressing

Address	IO/M	CS Outputs	IO Bus	IO Ack	On Bd
0000-07FF	M	00	X	X	X
0800-0FFF	M	1	X	X	X
7F00-7FFF	M	A	X	X	X
8000-8FFF	M	E	X	-	X
D0	IO	4	X	X	X
D1	IO	4	X	X	X
D2	IO	5	X	X	X
D3	IO	5	X	X	X
D4	IO	6	X	X	X

Table C-2. Chip Select Addressing (Cont'd.)

Address	IO/M	CS Outputs	IO Bus	IO Ack	On Bd
D5	IO	6	X	X	X
D6	IO	7	X	X	X
D7	IO	7	X	X	X
D8	IO	8	X	X	X
D9	IO	8	X	X	X
DA	IO	8	X	X	X
DB	IO	8	X	X	X
DC	IO	9	X	X	X
DD	IO	9	X	X	X
DE	IO	9	X	X	X
DF	IO	9	X	X	X
E4	IO	B	X	X	X
E5	IO	C	X	X	X
E6	IO	D	X	X	X
E7	IO	D	X	X	X
E8	IO	A	-	X	X
E9	IO	A	-	X	X
EA	IO	A	-	X	X
EB	IO	A	-	X	X
EC	IO	A	-	X	X
ED	IO	A	-	X	X
EE	IO	A	-	X	X
EF	IO	A	-	X	X

Note: Memory sizes shown are max. An "X" means the signal is active.

Table C-4. Chip Select Decode PROM Outputs (2K of ROM)

Page 0 A=0 IO/M=M Option Switch=On

B=CPU Address (ADRC-F)	C=CPU Address (ADR8-B)															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	10	10	10	10	11	11	11	11	FF	FF	FF	FF	FF	FF	FF	FF
1	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
2	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
3	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
6	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
7	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	3A
8	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E
9	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E
A	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E
B	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E
C	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
D	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
E	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
F	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

For presentation here, the PROM is broken up into 4 256-byte pages. IO/$\overline{M}$ (PROM input A9) breaks the PROM into 2 halves, and the option switch (PROM input A8) breaks each half into 2 sections. Table C-3 shows the relation of inputs to PROM pages.

Table C-3. PROM Page Partitioning

IO/M	S1-Position	PROM Page
M	ON	0
M	OFF	1
IO	ON	2
IO	OFF	3

C-3. CHIP SELECT PROM OUTPUTS

Tables C-4 through C-6 show the possible PROM outputs under specified conditions. For presentation here, the PROM address is split into 3 bytes. The page number (A) specifies the upper 2 bits of address (A8 & A9). The left hand column of each table (B) specifies the middle 4 bits of address (A4, A5, A6, & A7). The top row of each table specifies the lower 4 bit of address (A0, A1, A2, & A3).

The PROM address way be found by combining bytes "A", "B", & "C". For example: a particular entry is on page 2, row 5, column 9. The address within the PROM is 259 hex.

Table C-5. Chip Select Decode PROM Outputs (4K of ROM)

Page 1 A=1 IO/M=M Option Switch=Off

B = CPU Address (ADRC-F)	C = CPU Address (ADR8-B)															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	10	10	10	10	10	10	10	10	11	11	11	11	11	11	11	11
1	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
2	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
3	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
6	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
7	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	3A
8	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E
9	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E
A	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E
B	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E	7E
C	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
D	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
E	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
F	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

Table C-6. Chip Select Decode PROM Outputs (I/O Chips)

Page 2 or 3 A=2 or 3 IO/M=I/O Option Switch=On or Off

B = CPU Address (ADRC-F)	C = CPU Address (ADR8-B)															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
1	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
2	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
3	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
4	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
5	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
6	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
7	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
8	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
9	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
A	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
B	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
C	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
D	14	14	15	15	16	16	17	17	18	18	18	18	19	19	19	19
E	FF	FF	FF	FF	1B	1C	1D	1D	3A	3A	3A	3A	3A	3A	3A	3A
F	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

C-4. ADDRESS TRANSFORMATION PROM

The SBC-544 has up to 16K of dual port RAM which may be accessed by both the on-board processor and the bus. The base address of RAM on-board is fixed. The base address of RAM from the bus may be set on any 4K boundry. It is the job of this PROM to "transform" the bus address to the proper on-board address for correct dual-port access.

The base address as seen from the bus is set by 4 switches: sections 1 thru 4 of S1. These switches may be set in any one of 16 patterns, corresponding to the 16 4K boundries in a 64K address space. Table C-7 shows how these switches are set.

Table C-7. RAM Base Address

0=Switch On, 1=Switch Off

S1				Base
4	3	2	1	Address
0	0	0	0	0000H
0	0	0	1	1000H
0	0	1	0	2000H
0	0	1	1	3000H
0	1	0	0	4000H
0	1	0	1	5000H
0	1	1	0	6000H
0	1	1	1	7000H
1	0	0	0	8000H
1	0	0	1	9000H
1	0	1	0	A000H
1	0	1	1	B000H
1	1	0	0	C000H
1	1	0	1	D000H
1	1	1	0	E000H
1	1	1	1	F000H

The size of RAM available to the bus may also be specified. Two additional switches enable 4K, 8K, or 16K to the bus. See table C-8.

Table C-8. RAM Size

C=Switch On, 1=Switch Off

S1		RAM
6	5	Size
0	0	4K
0	1	8K
1	0	16K
1	1	N/A

RAM may also be totally disabled from the bus by setting the base address to F000H and the RAM size to 16K.

The PROM used is an Intel 3625-2, 1K by 4 bipolar PROM. As such, it has 10 address inputs and 4 outputs. Four of the inputs, (PROM address lines A0, A1, A2, A3) connect directly to bus address lines ADRC/, ADRD/, ADRE/, ADRF/. These lines tell the PROM the current address on the bus so that the PROM may determine if that address is for the on-board RAM. Since the address lines on the bus are inverted, the address represented by the inputs will be the complement of the actual input. Four more address inputs (PROM lines A4, A5, A6, A7) connect to the base address switches to tell the PROM the desired starting point of RAM. The final two inputs (PROM lines A8 and A9) connect to the RAM size switches to determine the amount of RAM on the bus.

One of the PROM outputs (04) is the RAM select line. This output is negative-true, that is, it goes low when an on-board RAM location is selected.

The other three outputs are the transformed address lines to the RAM. These lines are also negative-true. Output 01=ATRC/, 02=ATRD/, 03=ATRE/. Note that since 16K is the maximum memory size, ADRF is not needed.

For explanation purposes, the PROM is broken up into 4 256-byte pages. Each page corresponds to one setting of the RAM size switches. Table C-9 shows the relation between switch settings and the PROM page selected.

Table C-9. RAM Size-PROM Page

0=Switch On, 1=Switch Off

S1		RAM	PROM	PROM Address
6	5	Size	Page	Range
0	0	4K	0	0000H-0FFH
0	1	8K	1	100H-1FFH
1	0	16K	2	200H-2FFH
1	1	N/A	N/A	N/A

C-5. ADDRESS TRANSFORMATION PROM OUTPUTS

Tables C-10 through C-12 show the contents of the Address Transformation PROM for different size RAMS. For explanation purposes, the PROM address is split into 3 bytes. The page number (A) specifies the upper 2 bits of the address (A8 & A9). The left hand column of each table (B) specifies the middle 4 bits of the address (A4, A5, A6, and A7). The top row of each table specifies the lower 4 bits of the address (A0, A1, A2, and A3). The PROM address may be found by combining bytes A, B, and C. For example: a particular entry is on page 2, row 5, column 9. The address within the PROM is 259 Hex.

Table C-10. Address Transformation PROM Output (4K RAM)

Page 0 A=0 RAM Size=4K

B = Base Address Switches	C = Bus Address (Inverted)															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	7
1	F	F	F	F	F	F	F	F	F	F	F	F	F	F	7	F
2	F	F	F	F	F	F	F	F	F	F	F	F	F	7	F	F
3	F	F	F	F	F	F	F	F	F	F	F	F	7	F	F	F
4	F	F	F	F	F	F	F	F	F	F	F	7	F	F	F	F
5	F	F	F	F	F	F	F	F	F	F	7	F	F	F	F	F
6	F	F	F	F	F	F	F	F	F	7	F	F	F	F	F	F
7	F	F	F	F	F	F	F	F	7	F	F	F	F	F	F	F
8	F	F	F	F	F	F	F	7	F	F	F	F	F	F	F	F
9	F	F	F	F	F	F	7	F	F	F	F	F	F	F	F	F
A	F	F	F	F	F	7	F	F	F	F	F	F	F	F	F	F
B	F	F	F	F	7	F	F	F	F	F	F	F	F	F	F	F
C	F	F	F	7	F	F	F	F	F	F	F	F	F	F	F	F
D	F	F	7	F	F	F	F	F	F	F	F	F	F	F	F	F
E	F	7	F	F	F	F	F	F	F	F	F	F	F	F	F	F
F	7	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F

Table C-11. Address Transformation PROM Output (8K RAM)

Page 1 A=1 RAM Size=8K

B = Base Address Switches	C = Bus Address (Inverted)															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	F	F	F	F	F	F	F	F	F	F	F	F	F	F	6	7
1	F	F	F	F	F	F	F	F	F	F	F	F	F	6	7	F
2	F	F	F	F	F	F	F	F	F	F	F	F	6	7	F	F
3	F	F	F	F	F	F	F	F	F	F	F	6	7	F	F	F
4	F	F	F	F	F	F	F	F	F	F	6	7	F	F	F	F
5	F	F	F	F	F	F	F	F	F	6	7	F	F	F	F	F
6	F	F	F	F	F	F	F	F	6	7	F	F	F	F	F	F
7	F	F	F	F	F	F	F	6	7	F	F	F	F	F	F	F
8	F	F	F	F	F	F	6	7	F	F	F	F	F	F	F	F
9	F	F	F	F	F	6	7	F	F	F	F	F	F	F	F	F
A	F	F	F	F	6	7	F	F	F	F	F	F	F	F	F	F
B	F	F	F	6	7	F	F	F	F	F	F	F	F	F	F	F
C	F	F	6	7	F	F	F	F	F	F	F	F	F	F	F	F
D	F	6	7	F	F	F	F	F	F	F	F	F	F	F	F	F
E	6	7	F	F	F	F	F	F	F	F	F	F	F	F	F	F
F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F

Table C-12. Address Transformation PROM Output (16K RAM)

Page 2 A=2 RAM Size=16K

B = Base Address Switches	C = Bus Address (Inverted)															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	F	F	F	F	F	F	F	F	F	F	F	F	4	5	6	7
1	F	F	F	F	F	F	F	F	F	F	F	4	5	6	7	F
2	F	F	F	F	F	F	F	F	F	F	4	5	6	7	F	F
3	F	F	F	F	F	F	F	F	F	4	5	6	7	F	F	F
4	F	F	F	F	F	F	F	F	4	5	6	7	F	F	F	F
5	F	F	F	F	F	F	F	4	5	6	7	F	F	F	F	F
6	F	F	F	F	F	F	4	5	6	7	F	F	F	F	F	F
7	F	F	F	F	F	4	5	6	7	F	F	F	F	F	F	F
8	F	F	F	F	4	5	6	7	F	F	F	F	F	F	F	F
9	F	F	F	4	5	6	7	F	F	F	F	F	F	F	F	F
A	F	F	4	5	6	7	F	F	F	F	F	F	F	F	F	F
B	F	4	5	6	7	F	F	F	F	F	F	F	F	F	F	F
C	4	5	6	7	F	F	F	F	F	F	F	F	F	F	F	F
D	5	6	7	F	F	F	F	F	F	F	F	F	F	F	F	F
E	6	7	F	F	F	F	F	F	F	F	F	F	F	F	F	F
F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F

APPENDIX D
8K ROM CONVERSION

The following procedure will allow for the use of Intel 2332 4K x 8 ROMs on the iSBC 544 Intelligent Communications Controller. Reference figure 5-2 for the location of the components.

1. Remove jumper 40-41.
2. Install jumper 41-42.
3. Install jumper 45-46.
4. Remove decode PROM A50. Reprogram the following locations. (The original PROM may be used by transfering all of the PROM contents into a PROM Programmer memory, changing the specified locations, and reprogramming the same PROM with the new data.)
   a. Program locations 000 to 00FH with 10H.
   b. Program locations 010H to 01FH with 11H.
5. Reinstall PROM A50.
6. Switch SW1 (position 7) is now defined as follows:

   On (0) = 8K
   Off (1) = 4K

With SW1 (position 7) in the on position, ROM 0 (A51) covers addresses 0000H-0FFFH; and ROM 1 (A35) covers addresses 1000H-1FFFH. With SW1 (position 7) in the off position, the operation is as before allowing 4K of PROM.

REQUEST FOR READER'S COMMENTS

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

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

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

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

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

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

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

NAME _____ DATE _____
TITLE _____
COMPANY NAME/DEPARTMENT _____
ADDRESS _____
CITY _____ STATE _____ ZIP CODE _____

Please check here if you require a written reply. ☐

WE'D LIKE YOUR COMMENTS . . .

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

First Class
Permit No. 1040
Santa Clara, CA

BUSINESS REPLY MAIL

No Postage Stamp Necessary if Mailed in U.S.A.

Postage will be paid by:

Intel Corporation
3065 Bowers Avenue
Santa Clara, CA 95051

Attention: Technical Publications