iSBC® 546/547/548 HIGH PERFORMANCE TERMINAL CONTROLLERS HARDWARE REFERENCE MANUAL

Order Number: 122704-001

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

Additional copies of this manual or other Intel literature may be obtained from:

Literature Department
Intel Corporation
3065 Bowers Avenue
Santa Clara, CA 95051

The information in this document is subject to change without notice.

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

Intel Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in an Intel product. No other circuit patent licenses are implied.

Intel software products are copyrighted by and shall remain the property of Intel Corporation. Use, duplication or disclosure is subject to restrictions stated in Intel's software license, or as defined in ASPR 7-104.9(a)(9).

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

Intel Corporation makes no warranty for the use of its products and assumes no responsibility for any errors which may appear in this document nor does it make a commitment to update the information contained herein.

Intel retains the right to make changes to these specifications at any time, without notice.

Contact your local sales office to obtain the latest specifications before placing your order.

The following are trademarks of Intel Corporation and its affiliates and may be used only to identify Intel products:

Above	iLBX	iPDS	ONCE
BITBUS	im	iPSC	OpenNET
COMMputer	iMDDX	iRMX	Plug-A-Bubble
CREDIT	iMMX	iSBC	PROMPT
Data Pipeline	Insite	iSBX	Promware
GENIUS	Intel	iSDM	QueX
i	intel	iSXM	QUEST
i	intelBOS	Library Manager	Ripplemode
I²ICE	Intelevision	MCS	RMX/80
ICE	inteligent Identifier	Megachassis	RUPI
iCEL	inteligent Programming	MICROMAINFRAME	Seamless
iCS	Intellec	MULTIBUS	SLD
iDBP	Intellink	MULTICHANNEL	UPI
iDIS	iOSP	MULTIMODULE	VLSiCEL

MDS is an ordering code only and is not used as a product name or trademark. MDS® is a registered trademark of Mohawk Data Sciences Corporation.

*MULTIBUS is a patented Intel bus.

REV.	REVISION HISTORY	DATE
-001	Original Issue.	2/86

PREFACE

This manual provides information about the iSBC 548 and iSBC 547 Eight Channel Terminal Controllers and the iSBC 546 Terminal and Printer Controller. The iSBC 548 and iSBC 547 boards are functionally identical, but the iSBC 547 is a larger form factor (10" x 12") board with backpanel connectors on-board. The iSBC 546 is a four channel board with a clock calendar and a Centronix printer interface.

General information about all three boards is provided in Chapter 1. Chapter 2 provides a block diagrams and functional descriptions of the boards. Chapter 3 provides the information required to install the board. Programming information is provided in Chapter 4 as well as in Appendix A and B. Connector pin-out information for all boards is shown in Chapter 5. If you need to refer to the schematic diagrams see Chapter 6.

For reference purposes Appendix A provides jumper information for the boards. Appendix B covers the board firmware.

In addition to this manual you will need the following reference material (all are available from the Intel Literature Department, see page ii for address).

- Intel MULTIBUS Handbook, Order Number 210883
- Microsystem Components Handbook, Order Number 230843
- Serial Communications Controller Technical Manual, Order Number 230834.

CONTENTS

CONTENTS (continued)

TABLES (continued)

FIGURES

FIGURES (continued)

CHAPTER 1
GENERAL INFORMATION

1.1 INTRODUCTION

The iSBC 548, iSBC 547 and iSBC 546 are three single board terminal controllers to be used in the MULTIBUS I environment. The iSBC 548 and iSBC 547 are eight channel controllers. The iSBC 546 has four channels plus a line printer interface and clock/calendar.

The purpose of this chapter is to introduce you to all three boards. The remaining chapters will provide more detailed information on all the boards. This chapter gives a list of the key features, a brief description of each board and a list of specifications.

1.2 BOARD FEATURES

This section provides a brief list of key features of the iSBC 548 and iSBC 547 boards.

- Eight Mhz 80186 Microprocessors.
- Supports asynchronous RS232C interface in DTE configuration,on eight channels.
- 32K Byte dual-ported RAM, 96K Byte local RAM and supports up to 64K Byte EPROM sites populated with firmware (All Boards)
- Each serial channel supports transfer rates up to 19.2K Baud.
- Up to 96K Baud (per board) throughput rate (Special Character or Tandem Mode not used)
- Jumper selectable memory mapping
- Jumper selectable I/O mapping
- Jumper selectable MULTIBUS interrupts

- The iSBC 547 is a 10"x 12" form factor board with on-board backpanel connectors.

The iSBC 546 board differs from the iSBC 548 and iSBC 548 boards as follows:

- Four channels of RS232C instead of eight channels
- Line printer interface
- Clock calendar with battery back-up

1.3 BOARD DESCRIPTIONS

Sections 1.3.1, 1.3.2 and 1.3.3 provide general descriptions of the iSBC 548, iSBC 547 and iSBC 546 boards respectively. Figure 1-1 is a much simplified diagram for all three boards. Figures 1-2, 1-3 and 1-4 show the iSBC 548, iSBC 547 and iSBC 546 boards respectively.

1.3.1 iSBC 548 BOARD DESCRIPTION

The iSBC 548 board is a MULTIBUS based terminal controller. The board communicates with a MULTIBUS host as a slave board.

The board uses an Intel 80186 microprocessor, operating at 8 Mhz as its CPU. The 80186 controls eight serial channels sending data to or receiving data from the MULTIBUS host. The on-board 80186 gains the attention of the MULTIBUS host by generating an interrupt over the MULTIBUS interface to the host. A flag byte mechanism allows the MULTIBUS host to interrupt the board, to reset the board, or to reset an interrupt to the MULTIBUS host generated by the board.

The iSBC 548 board has four on-board 82530 Serial Communications Controllers (SCC). Each 82530 SCC contains two on-chip baud rate generators,allowing each channel to be independently programmed for separate baud rates. The maximum baud rate per channel is 19.2K Baud. Two 40-pin connectors can be attached to IBM PCAT compatible 9-pin connectors via ribbon cable.

The iSBC 548 board has four 64K x 4 DRAM (Dynamic RAM) devices, a total of 128 KBytes per board. The upper 32K Bytes can be addressed by other MULTIBUS boards.

The board also includes two 28-pin sockets. These sockets are populated with firmware EPROMs.

1.3.2 iSBC 547 BOARD DESCRIPTION

The iSBC 547 board is a terminal controller expansion to the Intel System 320. The board communicates with a MULTIBUS host as a slave board.

The board uses an Intel 80186 microprocessor, operating at 8 Mhz as its CPU. The 80186 controls eight serial channels sending data to or receiving data from the MULTIBUS host. The on-board 80186 gains the attention of the MULTIBUS host by generating an interrupt over the MULTIBUS interface to the host. A flag byte mechanism allows the MULTIBUS host to interrupt the board, to reset the board, or to reset an interrupt to the MULTIBUS host generated by the board.

The eight serial interfaces on the iSBC 547 board are through eight 9-pin connectors. The 9-pin connections are fully compatible with the IBM PCAT connections.

The iSBC 547 board has four on-board 82530 Serial Communications Controllers (SCC). Each 82530 SCC contains two on-chip baud rate generators,allowing each channel to be independently programmed for separate baud rates. The maximum baud rate per channel is 19.2K Baud.

The iSBC 547 board has four 64K x 4 DRAM (Dynamic RAM) devices, a total of 128 KBytes per board. The upper 32K Bytes can be addressed by other MULTIBUS boards.

The board also includes two 28-pin sockets. These sockets are populated with firmware EPROMs.

1.3.3 iSBC 546 BOARD DESCRIPTION

The iSBC 546 board is a terminal and line printer controller. The board communicates with a MULTIBUS host as a slave board.

The board uses an Intel 80186 microprocessor, operating at 8 Mhz as its CPU. The 80186 controls four serial channels, sending data to or receiving data from the MULTIBUS host, and a line printer interface. The on-board 80186 gains the attention of the MULTIBUS host by generating an interrupt over the MULTIBUS interface to the host. A flag byte mechanism allows the MULTIBUS host to interrupt the board, to reset the board, or to reset an interrupt to the MULTIBUS host generated by the board.

The four serial interfaces on the iSBC 546 board are through four 9-pin connectors. The 9-pin connections are fully compatible with the IBM PCAT connections.

The line printer interface is compatible with the IBM line printer interface.

The iSBC 546 board has two on-board 82530 Serial Communications Controllers (SCC). Each 82530 SCC contains two on-chip baud rate generators,allowing each channel to be independently programmed for separate baud rates. The maximum baud rate per channel is 19.2K Baud.

The iSBC 546 board has four 64K x 4 DRAM (Dynamic RAM) devices, a total of 128 KBytes per board. The upper 32K Bytes can be addressed by other MULTIBUS boards.

The board also includes two 28-pin sockets. These sockets are populated with firmware EPROMs.

A clock/calendar circuit, unique to the iSBC 546, is backed up by a non-rechargeable battery which keeps the clock/calendar operating for six months with all other power off.

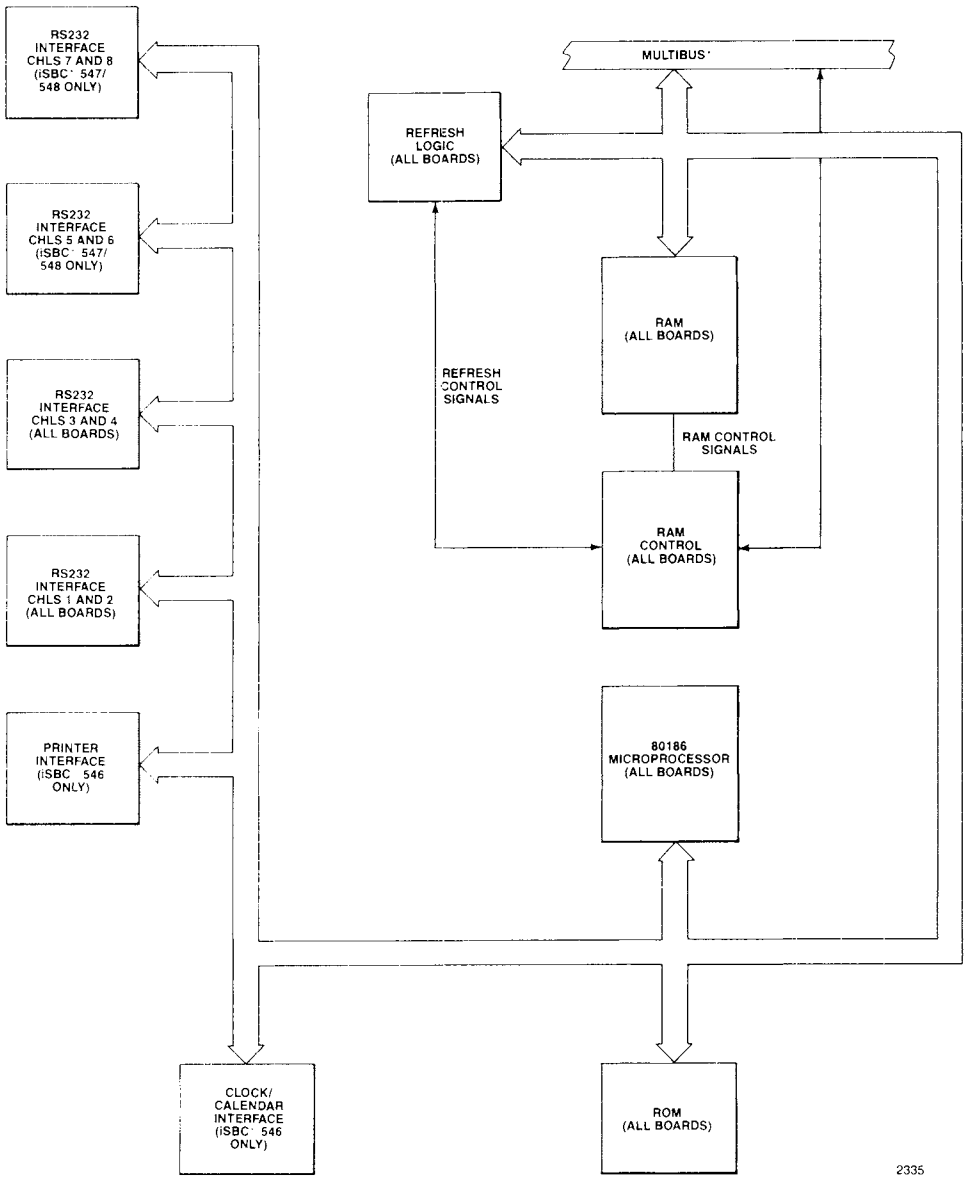


Figure 1-1. iSBC 546, iSBC 547 and iSBC 548 Boards, Block Diagram

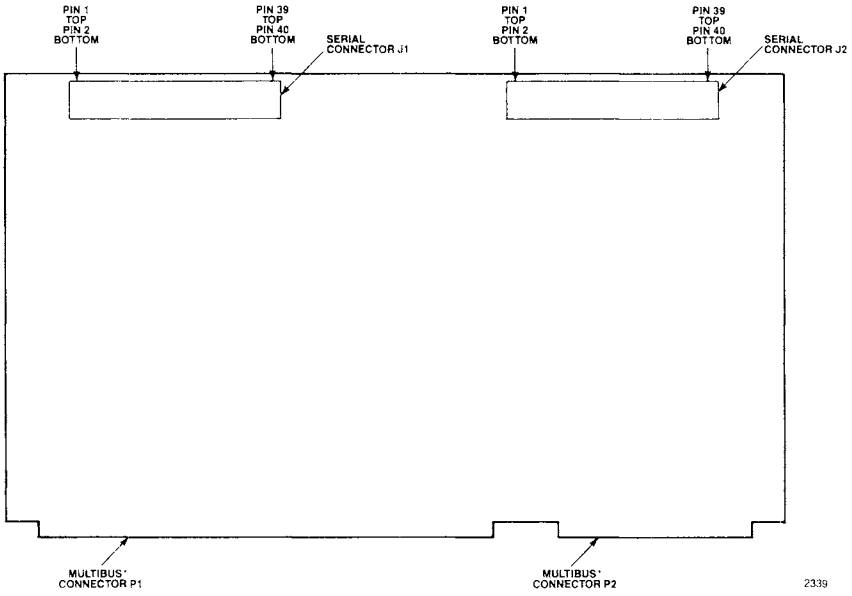


Figure 1-2. iSBC 548 High Performance Terminal Controller

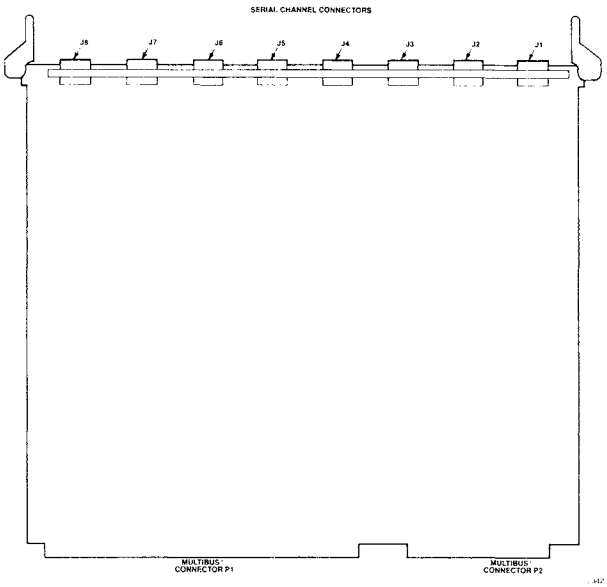


Figure 1-3. iSBC 547 High Performance Terminal Controller

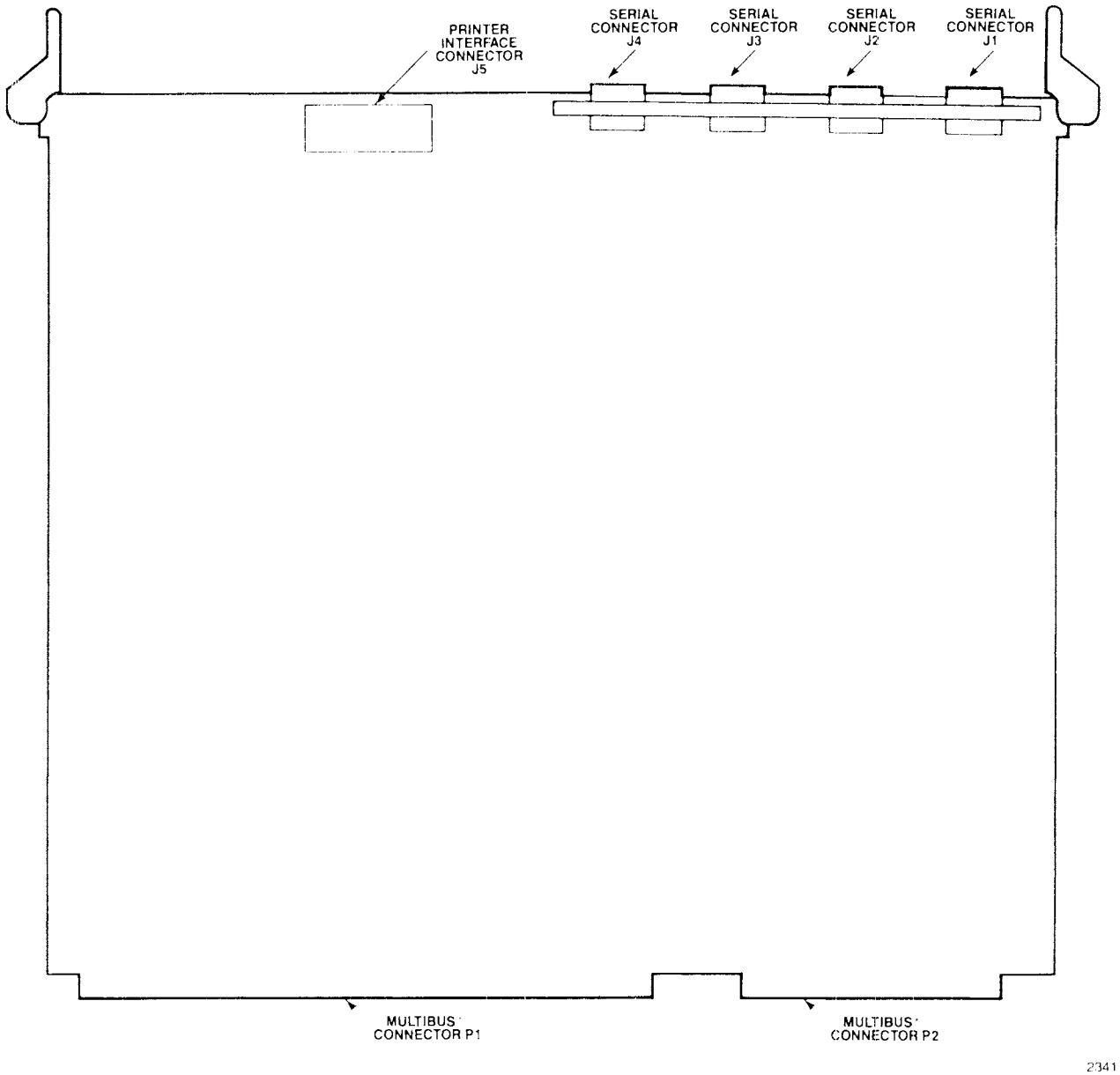


Figure 1-4. iSBC 548 High Performance Terminal Controller

1.4 SPECIFICATIONS

Table 1-1 summarizes the iSBC 546, iSBC 547 and iSBC 548 boards specifications.

Table 1-1. iSBC 546, iSBC 547, and iSBC 548 Specifications Summary

Board Performance (Transfer Rate)	
iSBC 547 and iSBC 548 Boards	Eight RS232C channels DTE configured. Maximum transfer rate per channel 19.2K Baud. Typical performance with firmware is 96K Baud.
iSBC 546	Four RS232C channels DTE configured. Maximum transfer rate per channel 19.2K Baud.
Interfaces	
iSBC 546 Board	MULTIBUS connectors P1 and P2. All MULTIBUS signals supported. The board at power-up requires an INIT pulse of at least 50 microseconds duration.
	Four RS232C channels, four 9-pin connectors.
	Line printer interface, one 25-pin connector. Interface is compatible with IBM PC Line Printer interface with the exception that AUTOFEED* and SELECT-INPUT signals are not supported.

Table 1-1. iSBC 546, iSBC 547, and iSBC 548 Specifications Summary (continued)

iSBC 547 Board		MULTIBUS connectors P1 and P2. All MULTIBUS signals supported. On power-up the board requires an INIT pulse of at least 50 microseconds duration. Eight RS232C channels eight 9-pin connectors.		
iSBC 548 Board		MULTIBUS connectors P1 and P2. All MULTIBUS signals supported. At power-up the board requires an INIT pulse of at least 50 microseconds duration. Eight RS232C channels, two 40-pin connectors.		
Electrical Requirements		iSBC 546	iSBC 547	iSBC 548
+5.00V ± 0.25V	(Max.)	3.260A	3.490A	3.490A
	(Typ.)	1.700A	1.870A	1.870A
+12.00V ± 0.60V	(Max.)	0.075A	0.150A	0.150A
	(Typ.)	0.390A	0.082A	0.082A
-12.00V ± 0.60V	(Max.)	0.069A	0.138A	0.138A
	(Typ.)	0.041	0.082A	0.082A
Environmental Characteristics				
Temperature		0 to 55 degrees C, minimum, 200 LFM of airflow		
Humidity		5% to 90%, non-condensing (25 to 55 degrees C)		

Table 1-1. iSBC 546, iSBC 547, and iSBC 548 Specifications Summary (continued)

Physical Dimensions	iSBC 546	iSBC 547	iSBC 548
Width	12.00 in (30.48 cm)	12.00 in (30.48 cm)	12.00 in (30.48 cm)
Length	10.00 in (25.40 cm)	10.00 in (25.40 cm)	7.00 in (17.78 cm)
Height (Including Components)	0.50 in (1.27 cm)	0.50 in (1.27 cm)	0.50 in (1.27 cm)

CHAPTER 2
BOARD OPERATION

2.1 INTRODUCTION

This chapter describes the operation of the three controller boards, the iSBC 546, the iSBC 547, and the iSBC 548. The iSBC 547 and iSBC 548 boards are functionally identical and their operation will be described jointly. The iSBC 546 board will be considered separately.

2.2 iSBC 547 AND iSBC 548 FUNCTIONAL DESCRIPTIONS

Figure 2-1 is a block diagram for the iSBC 547 and iSBC 548 boards. The boards are functionally identical and differ only in dimensions and in the type and number of serial interface connectors (eight 9-pin connectors for the iSBC 547 and two 40-pin connectors for the iSBC 548).

The iSBC 547 and iSBC 548 boards can not address the MULTIBUS interface, both are slave boards only. The interface to the MULTIBUS is through edge connectors P1 and P2.

Both boards use an Intel 80186 microprocessor, operating at 8 Mhz as their main processors. The 80186 has a 16 bit data bus and 16 bit internal architecture. The 80186 provides all bus controls without the need of a separate bus controller device.

The 80186 on the iSBC 547/548 controls eight serial channels sending data ,through them, from the MULTIBUS host or receiving data, through them, to the MULTIBUS host. Data transfer to and from the MULTIBUS is by use of a 32K Byte communication table (shared dual port memory) in the on-board dual-port RAM. The MULTIBUS host informs the on-board 80186 which serial channels are enabled. The 80186 then polls those channels continuously, looking for data from the MULTIBUS host, or the need to supply data to the MULTIBUS host.

The structure of the communication table is described in Appendix B, Section B.3.1 of this manual. The main blocks in the communication table in the on-board RAM are: a command queue (dynamic structures area), a status queue (static structures

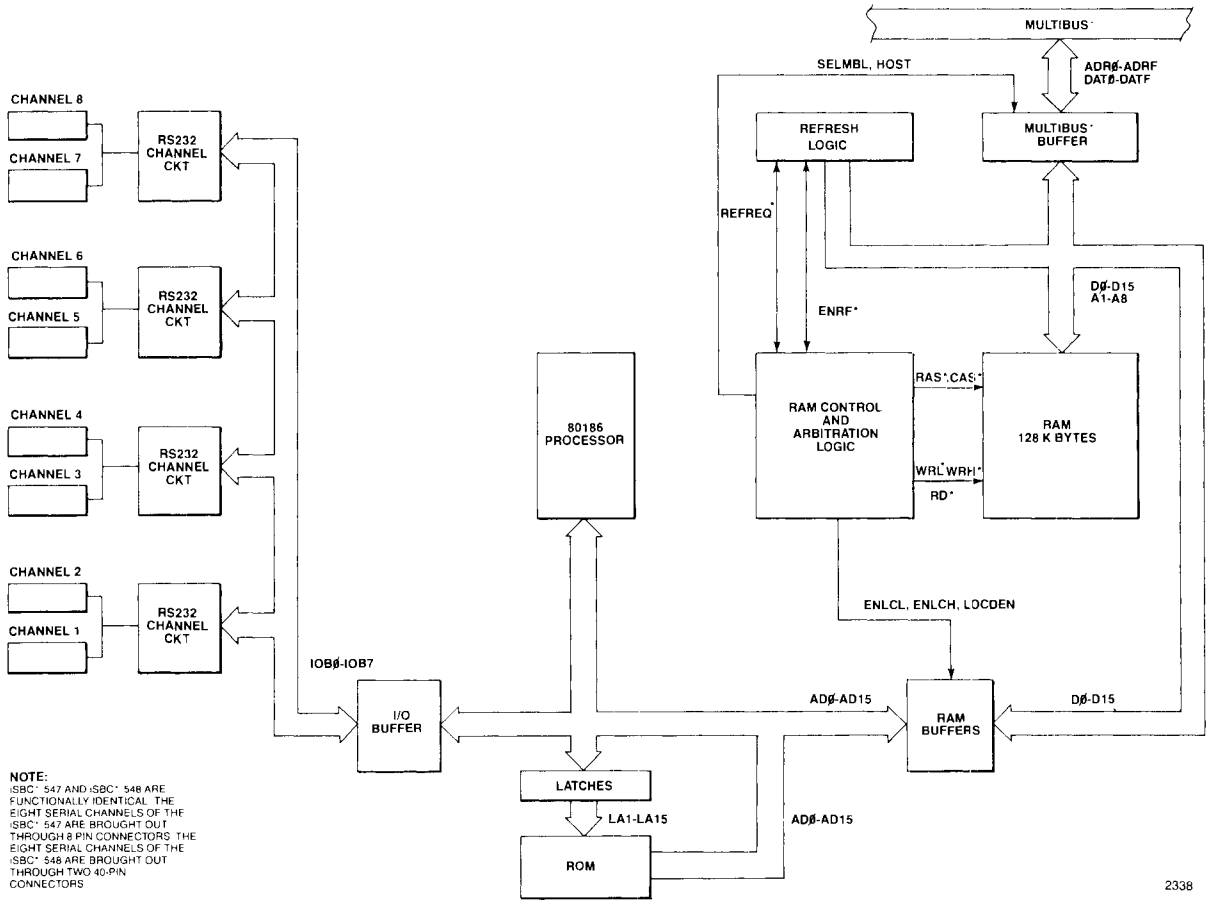


Figure 2-1. iSBC 547 and iSBC 548 Functional Block Diagram

area), a transmission area (transmit buffers), and a set of receive buffers. The MULTIBUS host gains the attention of the on-board 80186 to the command queue by a flag byte interrupt. The on-board 80186 gains the attention of the MULTIBUS host to the status queue by generating an interrupt over the MULTIBUS interface to the host. The interrupt line is jumper selectable as shown in Table A-2.

The flag byte mechanism allows the MULTIBUS host to interrupt the controller board, to reset the board, or to reset an interrupt to the MULTIBUS host generated by the board. The flag byte interrupt, sent by the MULTIBUS host to the controller board is an edge triggered input to the interrupt line of the on-board 80186. The flag byte is mapped to I/O space at a jumper selectable address (see Appendix A , Table A-2 of this manual). Interrupting the MULTIBUS host is done by writing data to an I/O port addressed through PCS5* (asterisk indicates signal is active low).

Each of the controller boards include two 28-pin sockets which are populated by two Intel 2764 EPROMs which contain the controller firmware. Appendix B of this manual describes the firmware in detail.

Although the controller boards are supplied with 2764 EPROMs the boards can support 27128 and 27256 EPROMs as well. The EPROM runs with zero wait states. The optional EPROMs must have access times of 250 ns or less. No jumper changes need be made when the different size EPROMs are used.

Each of the boards has four 64K x 4 DRAMs (Dynamic RAMs), a total of 128K Bytes of on-board RAM. The upper 32K Bytes of the on-board RAM can be addressed by other MULTIBUS boards as well as the on-board processor. The dual-port RAM can be seen from the MULTIBUS at several different starting addresses. The starting addresses are jumper selectable (see Table A-3 in Appendix A of this manual). The RAM operates with zero wait states.

The RAM is controlled with a PAL (Programmable Array Logic) device. The PAL generates all signals needed to control the RAM, arbitrate between the MULTIBUS host, the refresh logic and the 80186 and enables the address buffers as required. The on-board RAM is selected by the LCS (Lower Chip Select) signal generated by the on-board 80186. The memory arbiter allows refresh of the RAM even when the memory is locked.

RAM refresh uses a 1 Mhz output from Timer 1 of the on-board 80186. A divide by 15 counter causes a refresh request to be sent to the PAL arbiter every 15 microseconds. An eight bit counter addresses the RAM.

The serial channels of the controller boards are implemented in four 82530 Serial Communication Controller (SCC) chips. The baud rate clock for the serial channels is generated by the 82530 SCCs. Each channel has its own two on-chip baud rate generators, allowing each channel to be programmed separately. Chapter 4 of this manual describes baud rate programming.

The 82530 SCCs are selected by the PCS1* (Peripheral Chip Select) through PCS4* outputs of the on-board 80186. The DSR signals from the RS232 serial connectors are all tied to one input port decoded by the PCS0* line of the 80186.

2.3 iSBC 546 FUNCTIONAL DESCRIPTION

The iSBC 546 board, Figure 2-2, is similar to both the iSBC 547 and 548 boards. It differs primarily in that it has a line printer interface connector and associated circuitry, a clock/calendar circuit and supports only four serial channels.

The iSBC 546 processes data in the same manner as the other two boards; it has the same on-board RAM and controls it in same way as the other boards. The serial channels are controlled in the same manner as on the iSBC 547/548 boards except only two 82530 SCC devices are used.

The line printer interface is implemented through port A of an 8255A Programmable Peripheral Interface (PPI operated in strobed output mode). A PAL device controls timing and the line printer. Approximately two microseconds after data is written to port A the PAL generates a LP STB* (Line Printer Strobe) signal to the printer indicating data to the printer is valid. LP STB* stays active for one microsecond. When LP ACK (Line Printer Acknowledge) is returned by the printer it clears the port and allows more data to be sent.

The 8255A PPI is selected by the PCS3* signal generated by the on-board 80186. The PPI replaces one of the SCC devices in the I/O map for the controller boards.

The interface does not have RS232 lines 5 through 8, freeing four bits of the DSR port. These four lines are used for line printer status lines LP BUSY (Line Printer Busy), NO PAPER, FAULT and LP SELECT (Line Printer Select).

The line printer interface is compatible with the IBM line printer interface and with proper cabling interfaces to a Centronix line printer.

The clock calendar circuit uses a MM58167 clock chip and a 32.768 KHz crystal. The interface to the MM58167 uses the same PAL device as does the line printer interface. Port B of the 8255A device is used in both input or output strobed mode. PC4* and PC5* generated by the 8255A inform the PAL of either input or output mode. Coding of the two bits is as follows:

Function	PC4*	PC5*
Output to Clock Mode	1	0
Input From Clock Mode	0	1
Reset LP and Clock Interface	1	1
Reset Clock Interface Only	0	0

Whenever a new clock set is issued or a clock read is started PC4* and PC5* must be reset to 0,0 and the port set to the appropriate mode, input or output. Then PC4* and PC5* are programmed to the correct logic level and the hardware supplies the address to the clock by order,starting from milliseconds and all the way up to the clock internal RAM area. Only the first 16 addresses in the clock chip are addressable.

The PAL generates the control signals for the 8255A PPI. The data sent to the clock or received from the clock consists of eleven bytes.

The clock/calendar is backed-up by a non-rechargeable battery which insures at least six months operation with no off-board power. The battery back-up is jumper selectable.

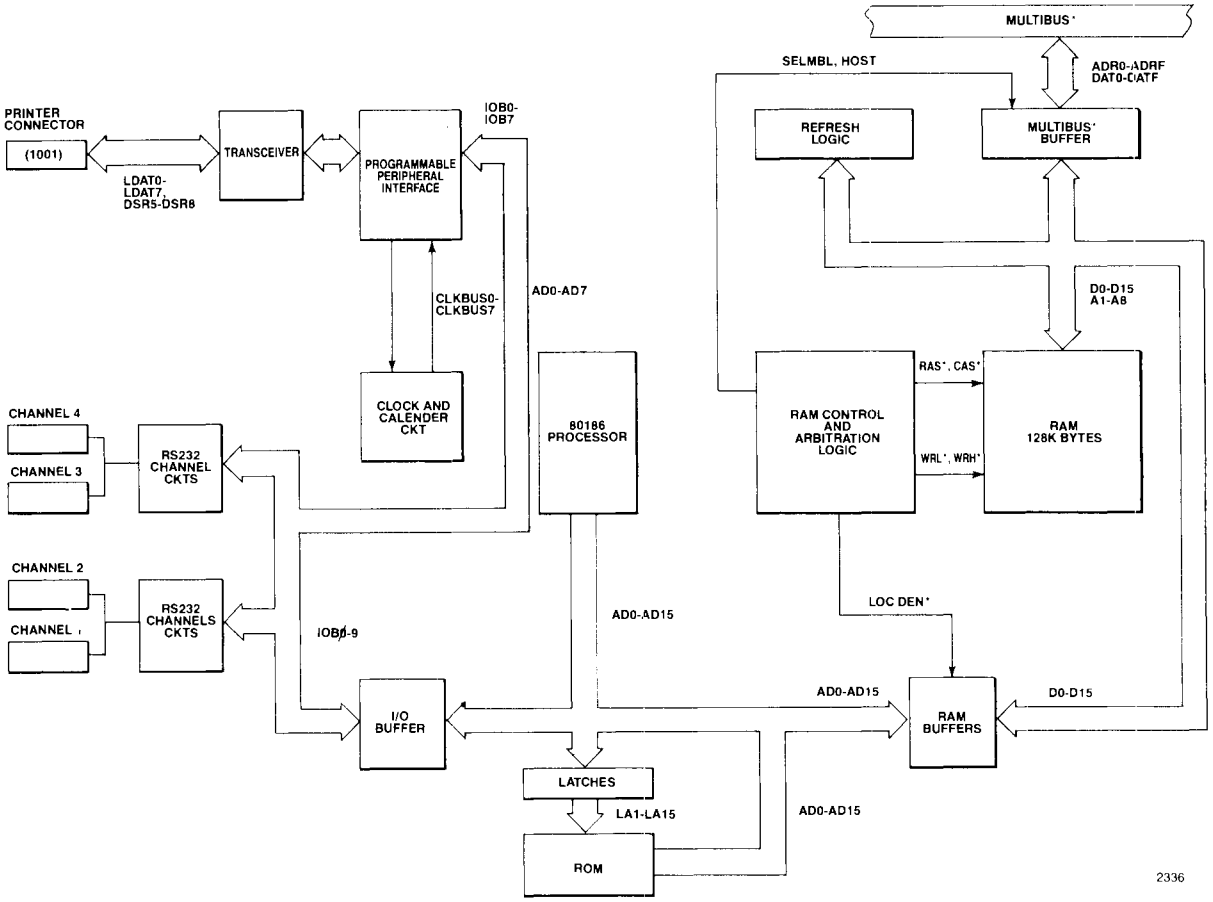


Figure 2-2. iSBC 546 Board Functional Block Diagram

CHAPTER 3
INSTALLATION

3.1 INTRODUCTION

This chapter explains how to receive, inspect and then install the iSBC 548, iSBC 547 and iSBC 546 boards. However, before installation you should read Chapter 4 Programming Considerations and Appendix A Jumper Information. Once you have set up the jumpers according to your system requirements proceed with the installation procedures in this chapter.

3.2 UNPACKING AND INSPECTION

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

United States customers can obtain service and repair assistance by contacting the Intel product service hotline in Phoenix, Arizona (see Chapter 6 for more information). Customers outside the United States should contact their sales source (Intel sales office or authorized distributor) for service information and repair assistance.

3.3 COMPATIBLE EQUIPMENT

The iSBC 548 can be installed in any MULTIBUS Compatible chassis.

The iSBC 547 board serves as a terminal controller expansion to the Intel System 320.

The iSBC 546 is part of the basic Intel System 320.

3.4 INSTALLATION CONSIDERATIONS

The following sections describe some of the installation consideration for the three boards.

THe iSBC 548, 547, and 546 boards can be configured to reside in 32 different address locations (see Table A-4) in the MULTIBUS address space. The board's flag byte address (wake-up address) is jumper selectable (see Table A-3) with eight options available in the MULTIBUS address space. The iSBC 548 and iS4H In the most ideal mult each controller board (iSBC 548, 547 or 546) would have different I/O mapping, different memory mapping and different interrupt lines. Under these conditions up to eight controller boards can be used in a system.

In a system application where more than eight controller boards are required the boards are grouped so that several boards share the same I/O address and the same interrupt line. The boards however cannot share the same address space.

As an example, if a system has one unused interrupt line, two unused I/O address lines, in the 8A0 through 8A7 range, and 20 unused address locations in the range the controller boards can be configured to (see Table A-4), than 20 different controller boards can be installed in the system. The boards will share the same interrupt line and use either one or two I/O addresses.

3.4.1 CONNECTOR CONFIGURATIONS

On all three boards connectors P1 and P2 are the MULTIBUS connectors. Pin assignments for each connector are provided in Table 5-1 and Table 5-3 respectively. The location of each connector on each board is shown in Figures 3-1, 3-2, and 3-3. Table 5-1 and Table 5-3 respectively.

On the iSBC 548 board connectors J1 and J2 are the serial I/O connectors (see Table 5-6 for pin assignments).

On the iSBC 547 board connectors J1 through J8 are the serial I/O connectors (see Table 5-5 for pin assignments).

On the iSBC 546 board connectors J1 through J4 are the serial I/O connectors (see Table 5-4 for pin assignments). Connector J5 is the printer interface connector (see Table 5-7 for pin assignments and Table 5-8 for signal descriptions).

3.4.2 BATTERY BACKUP

In order to use the battery backup for the clock/calendar on the iSBC 546 board the jumper between E30 and E31 must be installed by the user. In the default condition (as delivered from the factory) the backup battery is installed but the jumper is not.

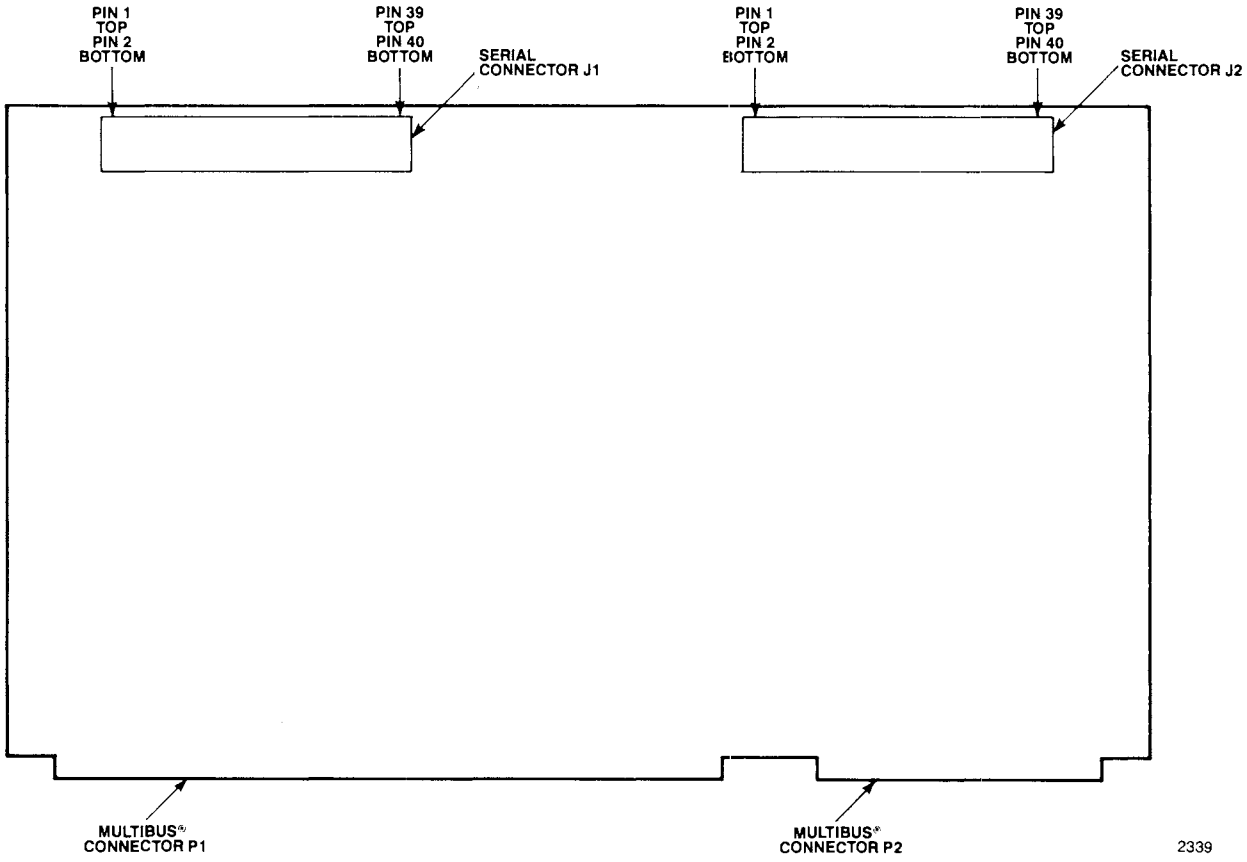


Figure 3-1. iSBC 548 Board Connector Locations

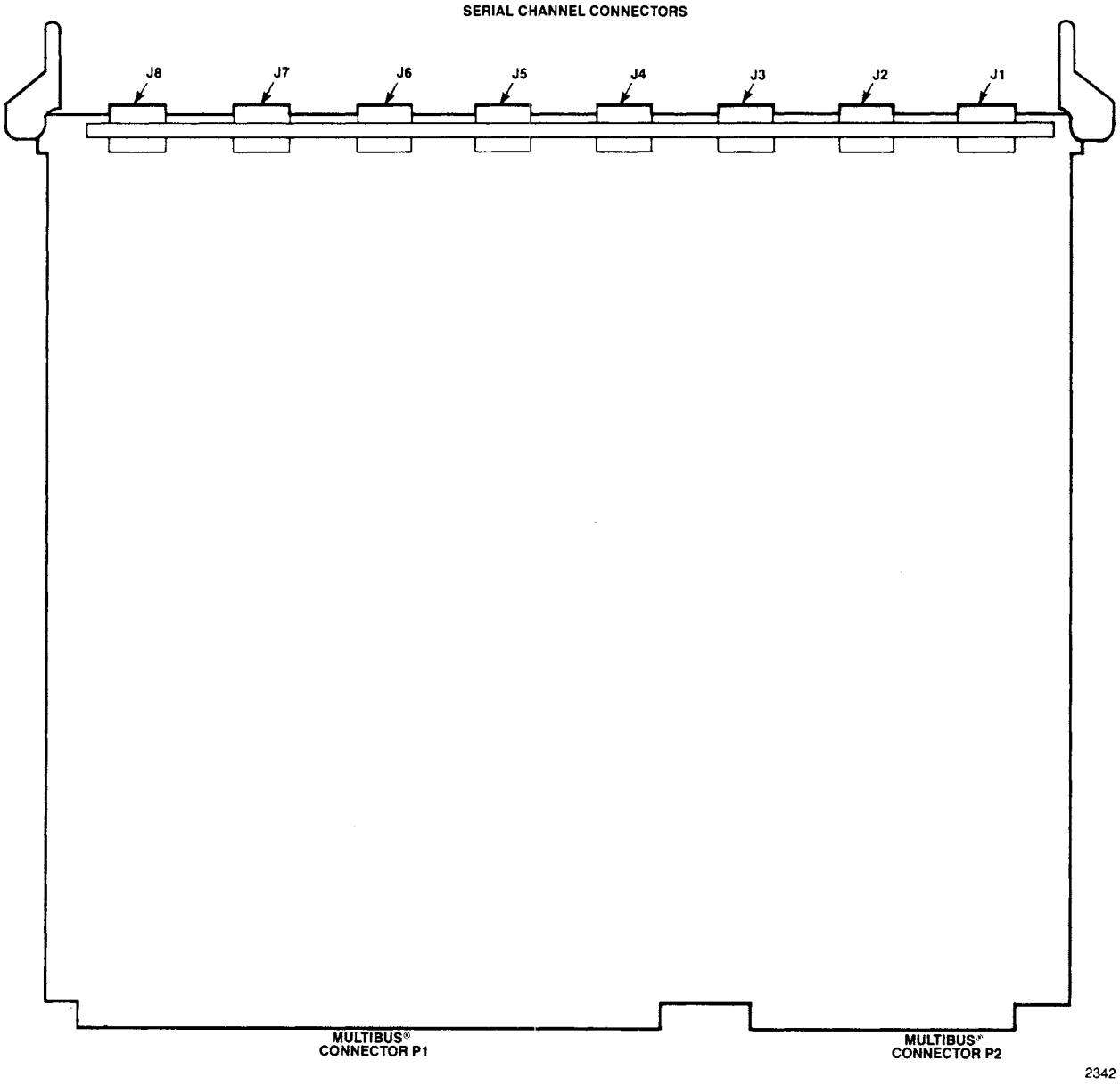


Figure 3-2. iSBC 547 Board Connector Locations

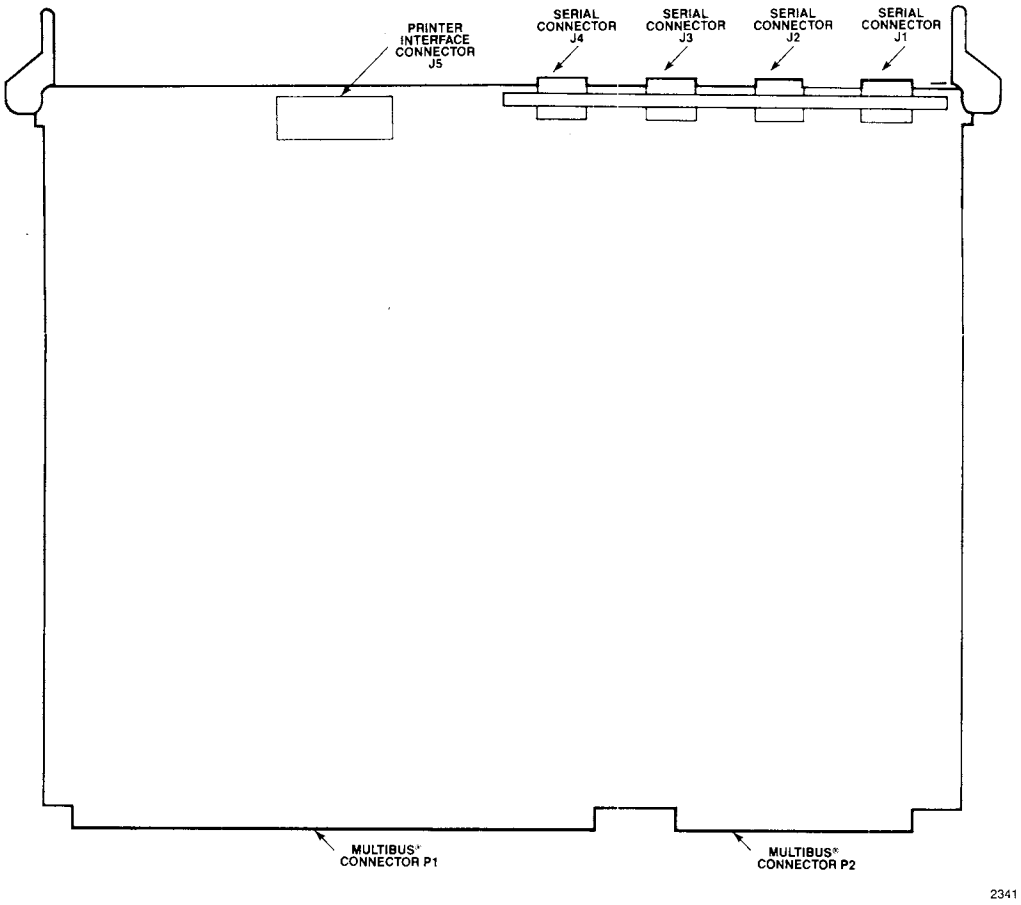


Figure 3-3. iSBC 546 Board Connector Locations

3.4.3 CABLING

The iSBC 548 board requires two flat 40 conductor cables to connect to the back panel. These cables can be acquired from Intel as part of the Intel 310 Cable Kit or can be fabricated by the user. Table 3-1 summarizes the recommended cable and connector part numbers for the iSBC 548 board. Figure 3-4 shows the cable construction. Table 3-2 lists the pin to pin wiring for the cable shown in Figure 3-4.

The iSBC 546 and iSBC 547 boards do not require cables. Connection is made directly on the card edge.

Table 3-1. Recommended Cables and Connectors

Connector	Manufacturer	Part Number
40 Pin or	3M	3417-6000 (without strain relief)
40 Pin or	3M	3417-6040 (with strain relief)
40 Pin or	T&B Ansley	609-4000M (without strain relief)
40 Pin	T&B Ansley	609-4001M (with strain relief
9 Pin	T&B Ansley	609-9P-ML (metal shroud, male)

Table 3-2. Pin to Pin Wiring List

40 Pin Connector	P4	P3	40 Pin Connector	P2	P1
1	5	-	19	5	-
2	9	-	20	9	-
3	4	-	21	4	-
4	8	-	22	8	-
5	3	-	23	3	-
6	7	-	24	7	-
7	2	-	25	2	-
8	6	-	26	6	-
9	1	-	27	1	-
10	-	5	28	-	5
11	-	9	29	-	9
12	-	4	30	-	4
13	-	8	31	-	8
14	-	3	32	-	3
15	-	7	33	-	7
16	-	2	34	-	2
17	-	6	35	-	6
18	-	1	36	-	1

Pins 37 through 40 of 40 pin connector not used. P1 through P4 are 9-pin connectors.

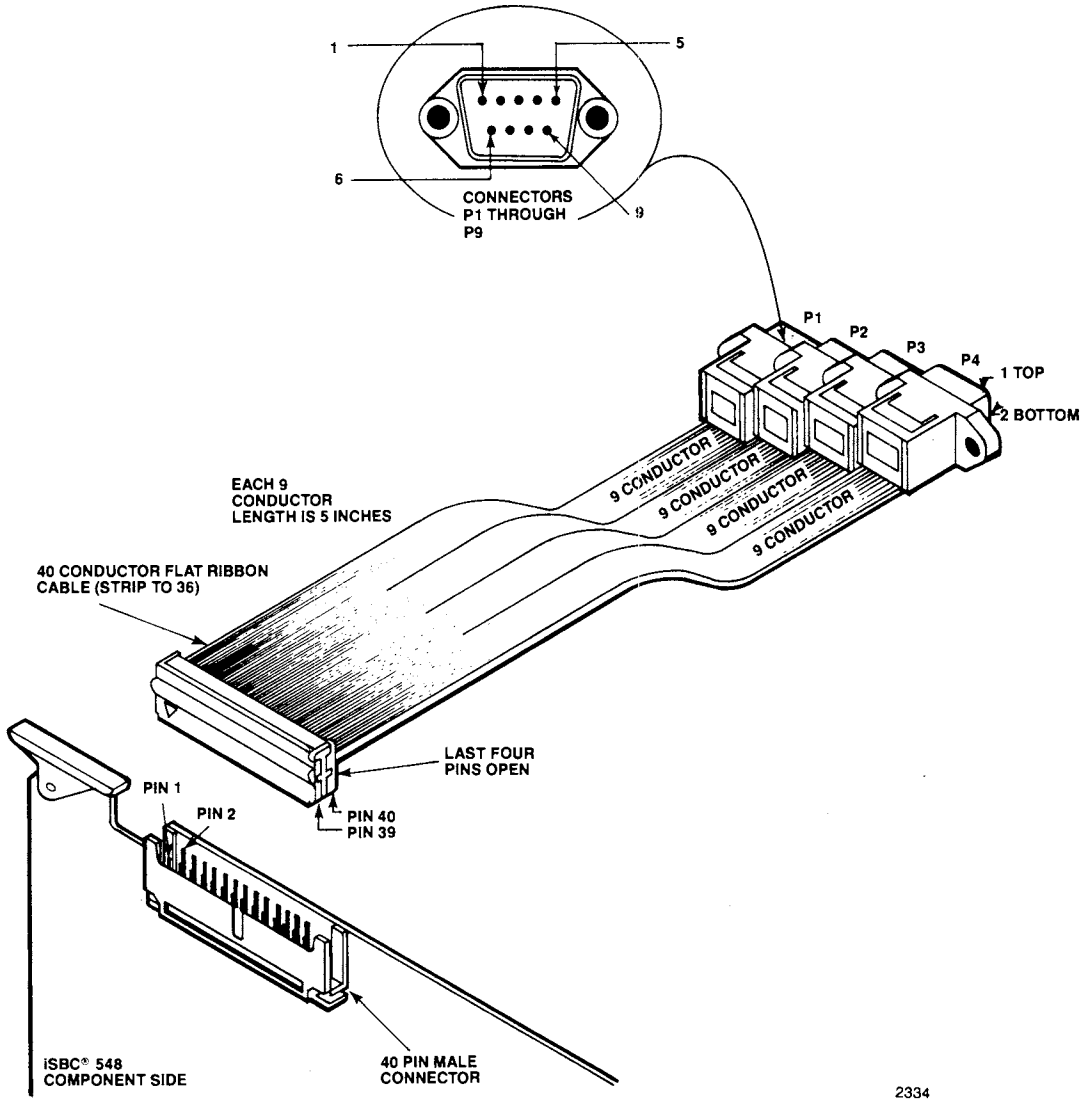


Figure 3-4. iSBC 548 RS232C Cable Construction

3.5 INSTALLATION PROCEDURE

The following is a general procedure for installing the terminal controller boards.

1. Check Appendix A for the jumper configuration.

2. Ensure that power to your system is turned off.

3. For the iSBC 548 board install the I/O cables to the 40 pin connectors.

4. Install the terminal controller board into the appropriate slot in your cardcage. Ensure that connectors P1 and P2 are fully seated in the cardcage.

CHAPTER 4
PROGRAMMING CONSIDERATIONS

4.1 INTRODUCTION

This chapter describes the programming considerations applicable to the users of the iSBC 546, iSBC 547 and iSBC 548 boards. This information can be used by a user wishing to run his own software on the boards, using the download feature.

4.2 JUMPERS

Appendix A of this manual locates the various jumpers (for all three controller boards) and describes their functions. The user should reference this appendix to verify that the required jumpers have been installed by the factory (the default condition) or to install his own configuration.

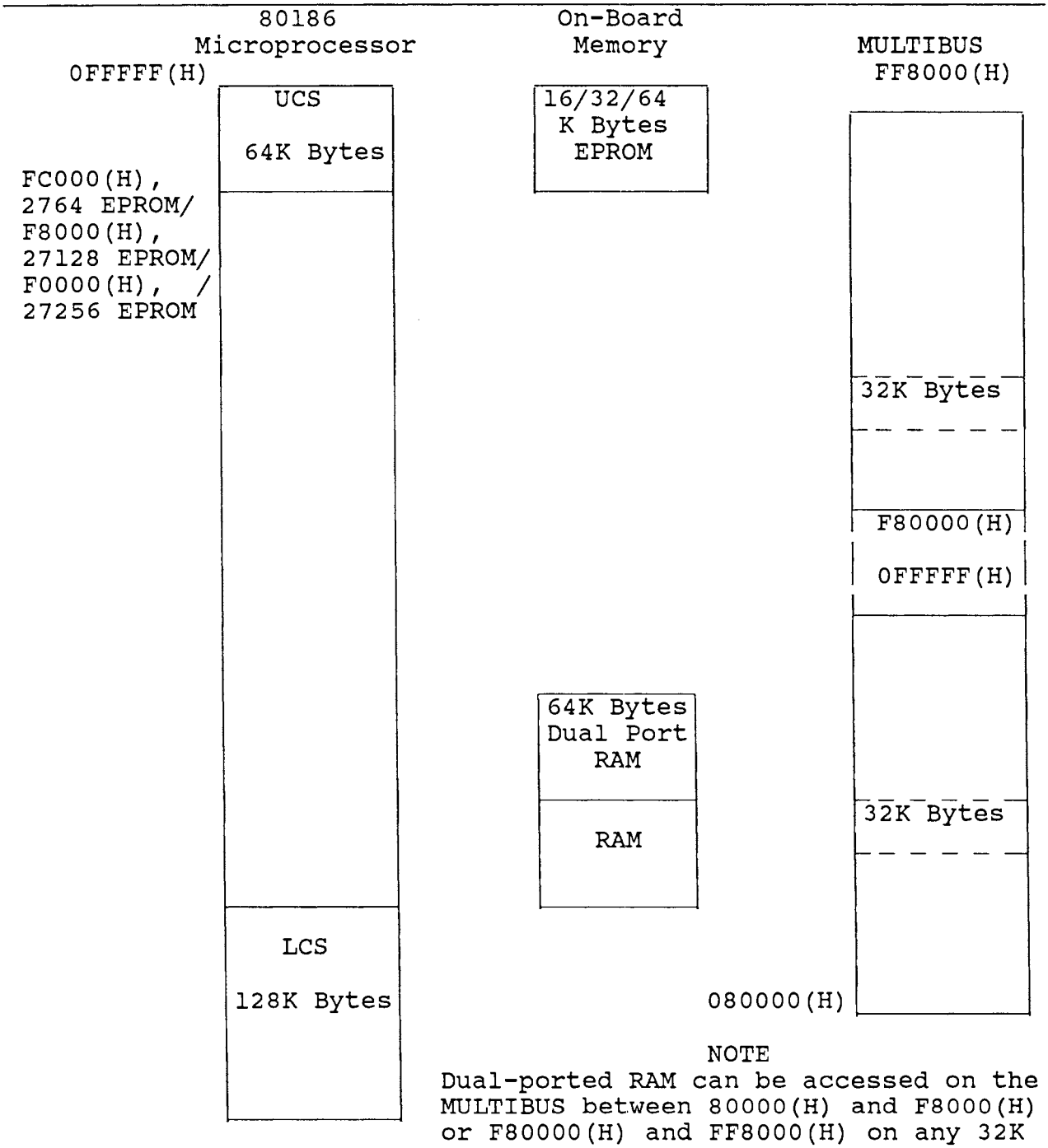
4.3 ADDRESSING

Figure 4-1 is a memory map for the iSBC 546/547/548 controllers.

The controller boards include two 28 pin sockets that can support either 2764, 27128 or 27256 EPROMs. Decoding of this memory portion is done by the 80186 processors UCS (Upper Chip Select) signal. Because of the different EPROMs capacities the starting addresses for this memory portion will vary as follows:

EPROM	Memory Size	Starting Address
2764	16K	FC000(H)
27128	32K	F8000(H)
27256	64K	F0000(H)

There are four 64K x 4 DRAMS on each controller board, a total of 128K Bytes. The upper 32K Bytes can be addressed by other MULTIBUS



NOTE
Dual-ported RAM can be accessed on the MULTIBUS between 80000(H) and F8000(H) or F80000(H) and FF8000(H) on any 32K boundary.

Figure 4-1. iSBC 546/547/548 Boards Memory Map

master boards. The dual-ported RAM can be addressed from the MULTIBUS interface at any 32K boundary starting between 80000(H) and F8000(H) or between F80000 and FF8000. The starting address is jumper determined see Appendix A). For the iSBC 546 board the default starting address is 0FA0000(H). For the iSBC 547 and 548 boards the default starting address is 0F90000(H).

4.4 PROGRAMMING CONSIDERATIONS

Sections 4.4.1 through 4.4.3 discuss the programming considerations for the three controller boards

4.4.1 FIRMWARE

The firmware for the controller boards is described in detail in Appendix B of this manual. The following paragraphs provide a brief description of firmware operation.

The 80186 microprocessors on the iSBC 547 and iSBC 548 boards control eight serial data channels. The 80186 on the iSBC 546 controls four serial data channels. The data received from the channel is communicated to the MULTIBUS host and the data transmitted to the channel is received from the MULTIBUS host. The MULTIBUS host informs the controller's 80186 which channels to enable and which not. The 80186 continuously polls the enabled channels looking for data or the request for data.

On the iSBC 546 board the line printer channel and clock/calendar are treated like serial channels.

4.4.2 80186 PROCESSOR PROGRAMMING CONSIDERATIONS

When programming the controller's 80186 microprocessor the following guidelines should be followed:

1. The LCS (Lower Chip Select) should be programmed for 128K Byte size and zero wait states.

2. The UCS (Upper Chip Select) should be programmed for 64K Byte size and zero wait states.

3. The PCS (Peripheral Chip Select) should be I/O mapped and configured as follows:

PCS	Function
0	Selects DSR port. PCS0 is not to to be used for for an output.
1	Selects serial ports 1 and 2.
2	Selects serial ports 3 and 4.
3	Selects serial ports 5 and 6 on iSBC 547 and 548 boards and line printer interface and clock/calendar on the iSBC 546 board.
4	Selects serial ports 7 and 8 (iSBC 547 and 548 only)
5	Sets MULTIBUS interrupt port when used as an output. PCS5 is not to be used as an input.

One wait state should be used for the PCS lines.

If the PCS lines base address is 0(H) then the I/O map will be as follows:

Address	Port	Type
0000 0000 0XXX XXXX	DSR Port	I
0000 0000 1XXX X000	Serial Line 2, control	I/O
0000 0000 1XXX X010	Serial Line 2, data	I/O
0000 0000 1XXX X100	Serial Line 1, control	I/O
0000 0000 1XXX X110	Serial Line 1, data	I/O

0000 0001 0XXX X000	Serial Line 4, control	I/O
0000 0001 0XXX X010	Serial Line 4, data	I/O
0000 0001 0XXX X100	Serial Line 3, control	I/O
0000 0001 0XXX X110	Serial Line 3, data	I/O
0000 0001 1XXX X000	Serial Line 6, control or Line Printer	I/O O
0000 0001 1XXX X010	Serial Line 6, data or clock/calendar	I/O
0000 0001 1XXX X100	Serial Line 5, control or Line Printer and clock/ calendar controls	I/O
0000 0001 1XXX X110	Serial Line 5 data or 8255 control	I/O O
0000 0010 0XXX X000	Serial Line 8, control	I/O
0000 0010 0XXX X010	Serial Line 8, data	I/O
0000 0010 0XXX 0100	Serial Line 7, control	I/O
0000 0010 0XXX 0110	Serial Line 7, data	I/O
0000 0010 1XXX XXXX	MULTIBUS Interrupt	O

In the RAM case EXTERNAL RDY overrides INTERNAL RDY. If INTERNAL RDY is active but EXTERNAL RDY is not, a wait state must be inserted.

The A2 address line selects between serial channels on the same components. When A2 equals 0 the port with the larger number is selected.

The 80186 address mapping I/O should be programmed as follows:

Port	Address	Data
UMCS (Upper Memory Chip Select)	0FFA0(H)	0F038(H)
LMCS (Lower Memory Chip Select)	0FFA2(H)	1FF8(H)
PACS (Peripheral Chip Select)	0FFA4(H)	0039(H)
MPCS (Mid-Range Peripheral Chip Select)	0FFA8(H)	80B9(H)

4. Timer 1 is programmed for a 1 Mhz output. Its mode control (I/O address 5E(H)) should be written with 0C003(H) and the count register (I/O address 5A(H)) should be written with 00001(H).

5. The interrupt controller should have only one external interrupt. INT1 from the flag byte activates interrupt 13 routine.

   Except for software interrupts there are only two timer interrupts available, timers 0 and 2 can be used by the firmware.

4.4.3 8255 PROGRAMMING

Programming considerations for the 8255 Programmable Peripheral Interface (PPI) are as follows:

The 8255 PPI control word (address 186(H)) should be programmed 0A4(H) when the clock is to be set, and 0A6(H) when the clock is to be read. To set PC4 and PC5 to desired levels, single bit addressing should be used.

To determine if data from the clock is available bit 0 of the input port 184(H) should be checked. If bit 0 is 1 data is available.

To determine if the clock or line printer are ready for more data, port 184(H) bits 0 (for clock) and 3 (for line printer) should be read. A 1 for either bit indicate a readiness for more data.

4.4.4 DSR PORT

The DSR port control word format for each controller board is shown below:

D7	D6	D5	D4	D3	D2	D1	D0
Fault	Line Printer Select	No Paper	Line Printer Busy	DSR4	DSR3	DSR2	DSR1

iSBC 546 Board

D7	D6	D5	D4	D3	D2	D1	D0
DSR8	DSR7	DSR6	DSR5	DSR4	DSR3	DSR2	DSR1

iSBC 547 and iSBC 548 Boards

4.5 BAUD RATE PROGRAMMING (ALL BOARDS)

To program the baud rate of a specific channel a time constant must be written to its time constant register. The time constant is calculated as follows:

$$\text{Time Constant} = \frac{\text{Clock}}{32 \times \text{Baud Rate}} - 2$$

Where: Clock = 4.9152 Mhz

Baud rates and their corresponding time constants are as follows:

Baud Rate	Time Constant (Decimal)
19,200	6
9,600	14
4,800	30
2,400	62
1,200	126
600	254
300	510

CHAPTER 5
INTERFACING INFORMATION

5.1 INTRODUCTION

This chapter provides pin assignments for all connector interfaces of the iSBC 546, iSBC 547 and iSBC 548 boards.

5.2 MULTIBUS INFORMATION

All three boards connect to the MULTIBUS interface through board connectors P1 and P2. Table 5-1 lists MULTIBUS connector P1 pin assignments, Table 5-2 describes the functions of the P1 signals. Table 5-3 lists MULTIBUS connector P2 pin assignments.

Table 5-1. MULTIBUS Connector P1 Pin Assignments

(Component Side)			(Circuit Side)		
Pin	Mneumonic	Description	Pin	Mnemonic	Description
1	GND	Signal GND	2	GND	Signal GND
3	+5V	+5 Vdc	4	+5V	+5 Vdc
5	+5V	+5 Vdc	6	+5V	+5 Vdc
7	+12V	+12 Vdc	8	+12V	+12 Vdc
9		Reserved	10		Reserved
11	GND	Signal GND	12	GND	Signal GND
13			14	INIT	Initialize
15			16		
17			18		
19	MRDC*	Mem Read Cmd	20	MWTC*	Mem Write Cmd
21			22	IOWC*	I/O Write Cmd
23	XACK*	XFER Ack	24	INH1*	Inhibit 1
25	LOCK*	Bus Lock	26		Reserved
27	BHEN*	Byte High En	28	ADR10*	
29			30	ADR11*	Address Bus
31			32	ADR12*	
33			34	ADR13*	

Table 5-1. MULTIBUS Connector P1 Pin Assignments (continued)

(Component Side)			(Circuit Side)		
Pin	Mneumonic	Description	Pin	Mnemonic	Description
35	INT6*	Parallel	36	INT7*	Parallel
37	INT4*	Interrupt	38	INT5*	Interrupt
39	INT2*	Requests	40	INT3*	Requests
41	INT0*		42	INT1*	
43	ADRE*		44	ADRF*	
45	ADRC*		46	ADRD*	
47	ADRA*		48	ADRB*	
49	ADR8*	Address Bus	50	ADR9*	Address Bus
51	ADR6*		52	ADR7*	
53	ADR4*		54	ADR5*	
55	ADR2*		56	ADR3*	
57	ADR0*		58	ADR1*	
59	DATE*		60	DATF*	
61	DATC*		62	DATD*	
63	DATA*		64	DATB*	
65	DAT8*		66	DAT9*	
67	DAT6*	Data Bus	68	DAT7*	Data Bus
69	DAT4*		70	DAT5*	
71	DAT2*		72	DAT3*	
73	DAT0*		74	DAT1*	
75	GND	Signal GND	76	GND	Signal GND
77		Reserved	78		Reserved
79	-12V	-12 Vdc	80	-12V	-12 Vdc
81	+5V	+5 Vdc	82	+5V	+5 Vdc
83	+5V	+5 Vdc	84	+5V	+5 Vdc
85	GND	Signal GND	86	GND	Signal GND

Signals not shown are not used in this application

Table 5-2. MULTIBUS Connector P1 Signal Descriptions

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. ADR13 is the most significant address bit.
DAT0* - DATF*	Data. These 16 bidirectional data lines transmit and receive data to and from the addressed memory location or I/O port. DATF* is the most significant bit.
INH1*	Inhibit RAM. For system application, allows the RAM addresses to be overlaid by another RAM or ROM in the system.
INIT*	Initialize. This signal resets the entire system to a known internal state. The iSBC 546, iSBC 547 and iSBC 548 boards are slave boards and will never generate INIT*. These boards require an INIT* pulse of 50 microseconds or longer for proper operation.
IOWC*	I/O Write. Indicates the address of an I/O port is on the MULTIBUS interface address lines and that the contents on the MULTIBUS interface data lines are to be accepted by the addressed port.
LOCK*	Lock. When the MULTIBUS master accesses the on-board dual port RAM and activates LOCK* the on-board resources are locked out by the dual port RAM until the MULTIBUS master removes LOCK*.
MRDC*	Memory Read Command. Indicates that a memory location address is on the MULTIBUS interface address lines and that the contents of that location are to be read on the MULTIBUS interface data lines.

Table 5-2. MULTIBUS Connector P1 Signal Descriptions (continued)

Signal	Functional Description
MWTC*	Memory Write Command. Indicates that a memory location address is on the MULTIBUS interface address lines and that the contents on the MULTIBUS interface data lines are to be written into that location.
XACK*	Transfer Acknowledge. Indicates to the bus. master that the read or write operation is completed by the generating device and that valid data is available on the MULTIBUS interface.

Table 5-3. Connector P2 Pin Assignments

(Component Side)			(Circuit Side)		
Pin	Mnemonic	Description	Pin	Mnemonic	Description
1			2		
3			4		
5			6		
7			8		
9			10		
11			12		
13			14		
15			16		
17			18		
19			20		
21			22		
23			24		
25			26		
27			28		
29			30		
31			32		
33			34		
35			36		
37			38		
39			40		
41			42		
43			44		
45			46		
47			48		
49			50		
51			52		
53			54		
55	ADR16*	Address	56	ADR17*	Address
57	ADR14*	Bus	58	ADR15*	Bus
59			60		

Note: 1. If address lines ADR14 through ADR17 are not used in specific system applications they are held high at connector P2, by the iSBC 546/547/548 boards.

2. Signals not shown are not used in this application.

5.3 SERIAL INTERFACES

All three boards, iSBC 546, 547 and 548 have RS232C serial interface connectors. The serial interface connectors associated with each board are shown below:

Board	Connectors
iSBC 546	Four 9 pin connectors, J1 through J4
iSBC 547	Eight 9 pin connectors, J1 through J8
iSBC 548	Two 40 pin connectors, J1 and J2

Pin assignments for the iSBC 546 board connectors are shown in Table 5-4. Table 5-5 shows the pin assignments for the iSBC 547 boards serial interface connectors and Table 5-6 shows the pin assignments for the iSBC 548 boards serial interface connectors.

Table 5-4. Serial Connectors Pin Assignments, iSBC 546 Board

Connector J1			Connector J2		
Pin	Mnemonic	Description		Mnemonic	Description
1	CD1	Carrier Detect	1	CD2	See
2	RXD1	Receive Data	2	RXD2	Description
3	TXD1	Transmit Data	3	TXD2	Connector J1
4	DTR1	Data Terminal Rdy	4	DTR2	
5	GND	Ground	5	GND	
6	DSR1	Data Set Ready	6	DSR2	
7	RTS1	Request to Send	7	RTS2	
8	CTS1	Clear to Send	8	CTS2	
9	RI1	Ring Indicator	9	RI2	

Note: 1. Number at the end of the mnemonic indicates channel.

Table 5-4. Serial Connectors Pin Assignments, iSBC 546 Board (continued)

Connector J3			Connector J4		
Pin	Mnemonic	Description		Mnemonic	Description
1	CD3	Carrier Detect	1	CD4	See
2	RXD3	Receive Data	2	RXD4	Description
3	TXD3	Transmit Data	3	TXD4	Connector J3
4	DTR3	Data Terminal Rdy	4	DTR4	
5	GND	Ground	5	GND	
6	DSR3	Data Set Ready	6	DSR4	
7	RTS3	Request to Send	7	RTS4	
8	CTS3	Clear to Send	8	CTS4	
9	RI3	Ring Indicator	9	RI4	

Note: 1. Number at the end of the mnemonic indicates channel.

Table 5-5. Serial Connectors Pin Assignments, iSBC 547 Board

Connector J1			Connector J2		
Pin	Mnemonic	Description		Mnemonic	Description
1	CD1	Carrier Detect	1	CD2	See
2	RXD1	Receive Data	2	RXD2	Description
3	TXD1	Transmit Data	3	TXD2	Connector J1
4	DTR1	Data terminal Rdy	4	DTR2	
5	GND	Ground	5	GND	
6	DSR1	Data Set Ready	6	DSR2	
7	RTS1	Request to Send	7	RTS2	
8	CTS1	Clear to Send	8	CTS2	
9	RI1	Ring Indicator	9	RI2	
Connector J3			Connector J4		
Pin	Mnemonic	Description		Mnemonic	Description
1	CD3	See	1	CD4	See
2	RXD3	Description	2	RXD4	Description
3	TXD3	Connector J1	3	TXD4	Connector J1
4	DTR3		4	DTR4	

Note: 1. Number at the end of the mnemonic indicates channel.

Table 5-5. Serial Connectors Pin Assignments, iSBC 547 Board (continued)

Connector J3			Connector J4		
Pin	Mnemonic	Description		Mnemonic	Description
5	GND	See	5	GND	See
6	DSR3	Description	6	DSR4	Description
7	RTS3	Connector J1	7	RTS4	Connector J1
8	CTS3		8	CTS4	
9	RI3		9	RI4	
Connector J5			Connector J6		
Pin	Mnemonic	Description		Mnemonic	Description
1	CD5	See	1	CD6	See
2	RXD5	Description	2	RXD6	Description
3	TXD5	Connector J1	3	TXD6	Connector J1
4	DTR5		4	DTR6	
5	GND		5	GND	
6	DSR5		6	DSR6	
7	RTS5		7	RTS6	
8	CTS5		8	CTS6	
9	RI5		9	RI6	
Connector J7			Connector J8		
Pin	Mnemonic	Description		Mnemonic	Description
1	CD7	See	1	CD8	See
2	RXD7	Description	2	RXD8	Description
3	TXD7	Connector J1	3	TXD8	Connector J1
4	DTR7		4	DTR8	
5	GND		5	GND	
6	DSR7		6	DSR8	
7	RTS7		7	RTS8	
8	CTS7		8	CTS8	
9	RI7		9	RI8	

Note: 1. Number at the end of the mnemonic indicates channel.

Table 5-6. Serial Connectors Pin Assignments, iSBC 548 Board

Connector J1			RS232C
Pin	Mnemonic	Description	Pin
1	GND	Ground	1
2	RI8	Ring Indicator,Ch8	22
3	DTR8	Data Term Rdy,Ch8	20
4	CTS8	Clear to Send,Ch8	5
5	TXD8	Transmit Data,Ch8	2
6	RTS8	Req to Send,Ch8	4
7	RXD8	Receive Data,Ch8	3
8	DSR8	Data Set Rdy,Ch8	6
9	CD8	Carrier Detect,Ch8	8
10	GND	Ground	1
11	RI7	Ring Indicator,Ch7	22
12	DTR7	Data Term Rdy,Ch7	20
13	CTS7	Clear to Send,Ch7	5
14	TXD7	Transmit Data,Ch7	2
15	RTS7	Req to Send,Ch7	4
16	RXD7	Receive Data,Ch7	3
17	DSR7	Data Set Rdy,Ch7	6
18	CD7	Carrier Detect,Ch7	8
19	GND	Ground	1
20	RI6	Ring indicator,Ch6	22
21	DTR6	Data Term Rdy,Ch6	20
22	CTS6	Clear to Send,Ch6	5
23	TXD6	Transmit Data,Ch6	2
24	RTS6	Req to Send,Ch6	4
25	RXD6	Receive Data,Ch6	3
26	DSR6	Data Set Rdy,Ch6	6
27	CD6	Carrier Detect,Ch6	8
28	GND	Ground	1
29	RI5	Ring Indicator,Ch5	22
30	DTR5	Data Term Rdy,Ch5	20
31	CTS5	Clear to Send,Ch5	5
32	TXD5	Transmit Data,Ch5	2
33	RTS5	Req to Send,Ch5	4
34	RXD5	Receive Data,Ch5	3
35	DSR5	Data Set Rdy,Ch5	6
36	CD5	Carrier Detect,Ch5	8
37	-		1
38	-		
39	-		
40	-		

Table 5-6. Serial Connectors Pin Assignments, iSBC 548 Board

Connector J2			RS232C
Pin	Mnemonic	Description	Pin
1	GND	Ground	1
2	RI4	Ring Indicator,Ch4	22
3	DTR4	Data Term Rdy,Ch4	20
4	CTS4	Clear to Send,Ch4	5
5	TXD4	Transmit Data,Ch4	2
6	RTS4	Req to Send,Ch4	4
7	RXD4	Receive Data,Ch4	3
8	DSR4	Data Set Rdy,Ch4	6
9	CD4	Carrier Detect,Ch4	8
10	GND	Ground	1
11	RI3	Ring Indicator,Ch3	22
12	DTR3	Data Term Rdy,Ch3	20
13	CTS3	Clear to Send,Ch3	5
14	TXD3	Transmit Data,Ch3	2
15	RTS3	Req to Send,Ch3	4
16	RXD3	Receive Data,Ch3	3
17	DSR3	Data Set Rdy,Ch3	6
18	CD3	Carrier Detect,Ch3	8
19	GND	Ground	1
20	RI2	Ring Indicator,Ch2	22
21	DTR2	Data Term Rdy,Ch2	20
22	CTS2	Clear to Send,CH2	5
23	TXD2	Transmit Data,Ch2	2
24	RTS2	Req to Send,Ch2	4
25	RXD2	Receive Data,Ch2	3
26	DSR2	Data Set Rdy,Ch2	6
27	CD2	Carrier Detect,Ch2	8
28	GND	Ground	1
29	RI1	Ring Indicator,Ch1	22
30	DTR1	Data Term Rdy,Ch1	20
31	CTS1	Clear to Send,Ch1	5
32	TXD1	Transmit Data,Ch1	2
33	RTS1	Req to Send,Ch1	4
34	RXD1	Receive Data,Ch1	3
35	DSR1	Data Set Rdy,Ch1	6
36	CD1	Carrier Detect,CH1	8
37	-		
38	-		
39	-		
40	-		

5.4 PRINTER INTERFACE (iSBC 546 ONLY)

The iSBC 546 board has a line printer interface connector (J5). Table 5-7 shows the pin assignments for the connector. Table 5-8 describes the function of the printer interface connector signals.

Table 5-7 Printer Interface Connector J5 Pin Assignments

Pin	Mnemonic	Description
1	LP STB*	Line Printer Strobe
2	LDAT0	Line Printer Data Bit 0
3	LDAT1	Line Printer Data Bit 1
4	LDAT2	Line Printer Data Bit 2
5	LDAT3	Line Printer Data Bit 3
6	LDAT4	Line Printer Data Bit 4
7	LDAT5	Line Printer Data Bit 5
8	LDAT6	Line Printer Data Bit 6
9	LDAT7	Line Printer Data Bit 7
10	LP ACK	Line Printer Acknowledge
11	LP BUSY	Line Printer Busy
12	NO PAPER	No Paper
13	LP SELECT	Line Printer Select
14	-	Not Used
15	FAULT	Fault
16	LP RST	Line Printer Reset
17	-	Not Used
18	GND	Ground
19	GND	Ground
20	GND	Ground
21	GND	Ground
22	GND	Ground
23	GND	Ground
24	GND	Ground
25	GND	Ground

Table 5-8. Connector J5 Signal Descriptions

Signal	Functional Description
LP STB*	Line Printer Strobe. This signal is sent to to the line printer and causes the printer to strobe the data on the data lines (LDAT0 through LDAT7) into the printer.
LPDAT0 through LPDAT7	Data Bus. This is the data bus between the iSBC 546 board and the line printer. The contents of the data bus are strobed into the line printer by LP STB*.
LP ACK*	Line Printer Acknowledge. The line printer activates this signal to indicate it has accepted the data strobed off the data lines.
LP BUSY	Line Printer Busy. This signal is activated by the line printer to indicate it is busy and cannot accept more data.
NO PAPER	No Paper. This signal from the printer indicates it is out of paper.
LP SELECT	Line Printer Select. This signal is activated by the line printer to indicate it is ready for use.
FAULT	Fault. This signal from the printer indicates a problem has developed which will prevent further printer operation.
LP RST	Line Printer Reset. This signal is generated by the iSBC 546 board to reset the line printer.

CHAPTER 6
SERVICE ASSISTANCE INFORMATION

6.1 INTRODUCTION

This chapter provides a list of service diagrams and service and repair assistance instructions for the iSBC 548, iSBC 547, and iSBC 546 boards.

6.2 SERVICE AND REPAIR ASSISTANCE

Intel Customer Support Service Engineering provides both a Return Replacement Authorization (RRA) and Direct Return Authorization (DRA) service.

The RRA service provides replacement of a defective board. Return the defective board to Intel, freight prepaid, and Intel will replace the board with a new serial number board. This service is not offered on all products. It is subject to board availability, and is available to customers in non-service areas. Intel expects to ship 90% of these products within 48 hours of receiving the defective board.

The DRA service provides repair work. Return the defective board to Intel, freight prepaid, and Intel will repair, test and update the board, with all mandatory Engineering Change Orders. The boards serial number will not change. Normal turn-around time is four to six weeks.

Determine which service fits your needs, RRA or DRA. Before calling Customer Support Service (Refer to Figure 6-1 for the telephone number in your area) have the following information ready:

1. Part and serial number of the board.
2. Purchase order number, needed for repair and shipping charges.
3. If it is a warranty repair, proof of purchase is required. Purchase must have been within 90 days of the service request. Without proof of purchase date services will be billed at the current rate.

4. Your shipping and billing address.

5. Your Intel contact and your telephone number.

In correspondence with Customer Support Engineering, reference the authorization number on the packing slip, the purchase order, and other related documents.

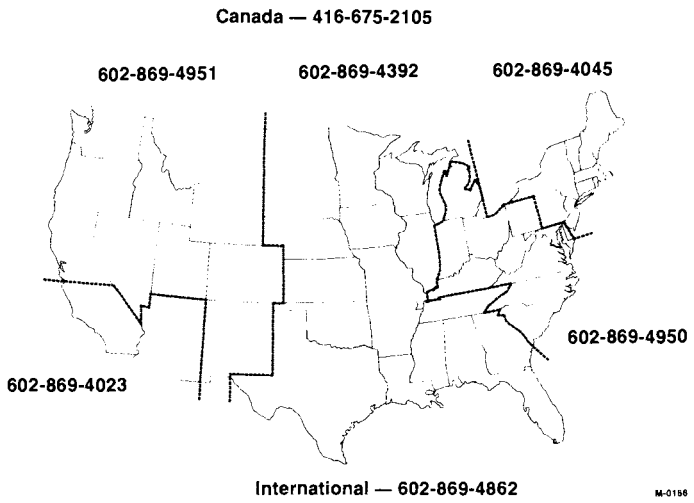


Figure 6-1. Territorial Service Telephone Numbers

Before shipping remove all user modifications. Protect the product from damage in transit as follows:

1. Boards should be placed in anti-static bags, and then in padded shipping bags. Large items should be wrapped in anti-static material.

2. Allow room in the box for protective padding, e.g. flow pack, foam etc.

3. Write the return authorization number on the outside of the box, and label the box "FRAGILE".

4. Damage sustained due to the lack of compliance safe return packaging could result in extra repair charges.

5. Forward the board and all correspondence to:

   Intel Corporation
   Customer Support Marketing Ad.
   Billing Department DV-1-704A
   2402 W. Beardsley Road
   Phoenix, Arizona 85027
   Authorization *_____

6.3 SERVICE DIAGRAMS

Figure 6-2 is the schematic diagram for the iSBC 548 board. Figure 6-3 is the schematic diagram for the iSBC 547 board and Figure 6-4 is the schematic diagram for the iSBC 546 board.

On the schematic diagrams a signal mnemonic followed by an asterisk indicates a signal active in the low state. Conversely a signal mnemonic without an asterisk indicates a signal active in the high state.

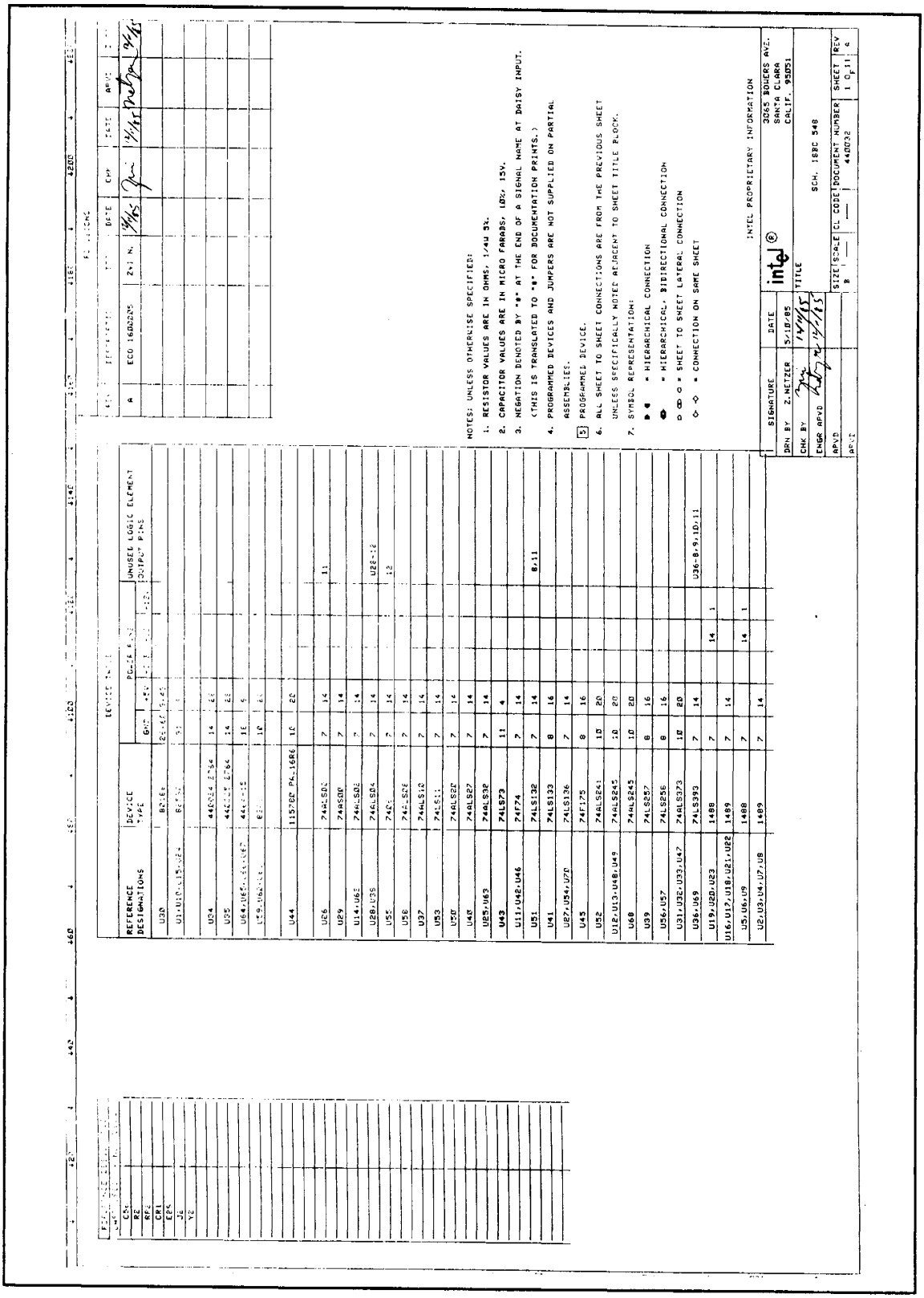


Figure 6-2 iSBC 548 Schematic Diagram (Sheet 1 of 11)

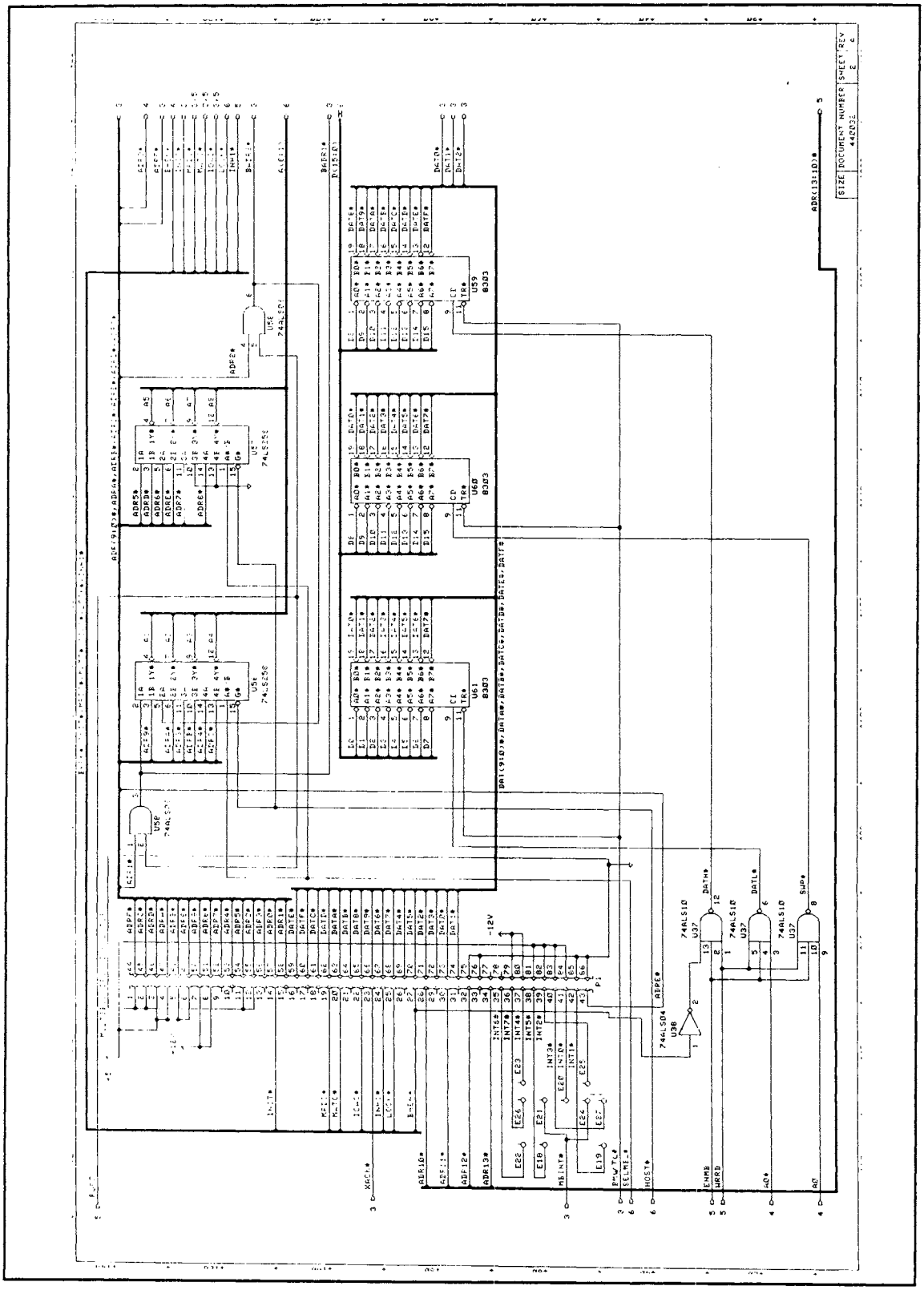


Figure 6-2 iSBC 548 Schematic Diagram (Sheet 2 of 11)

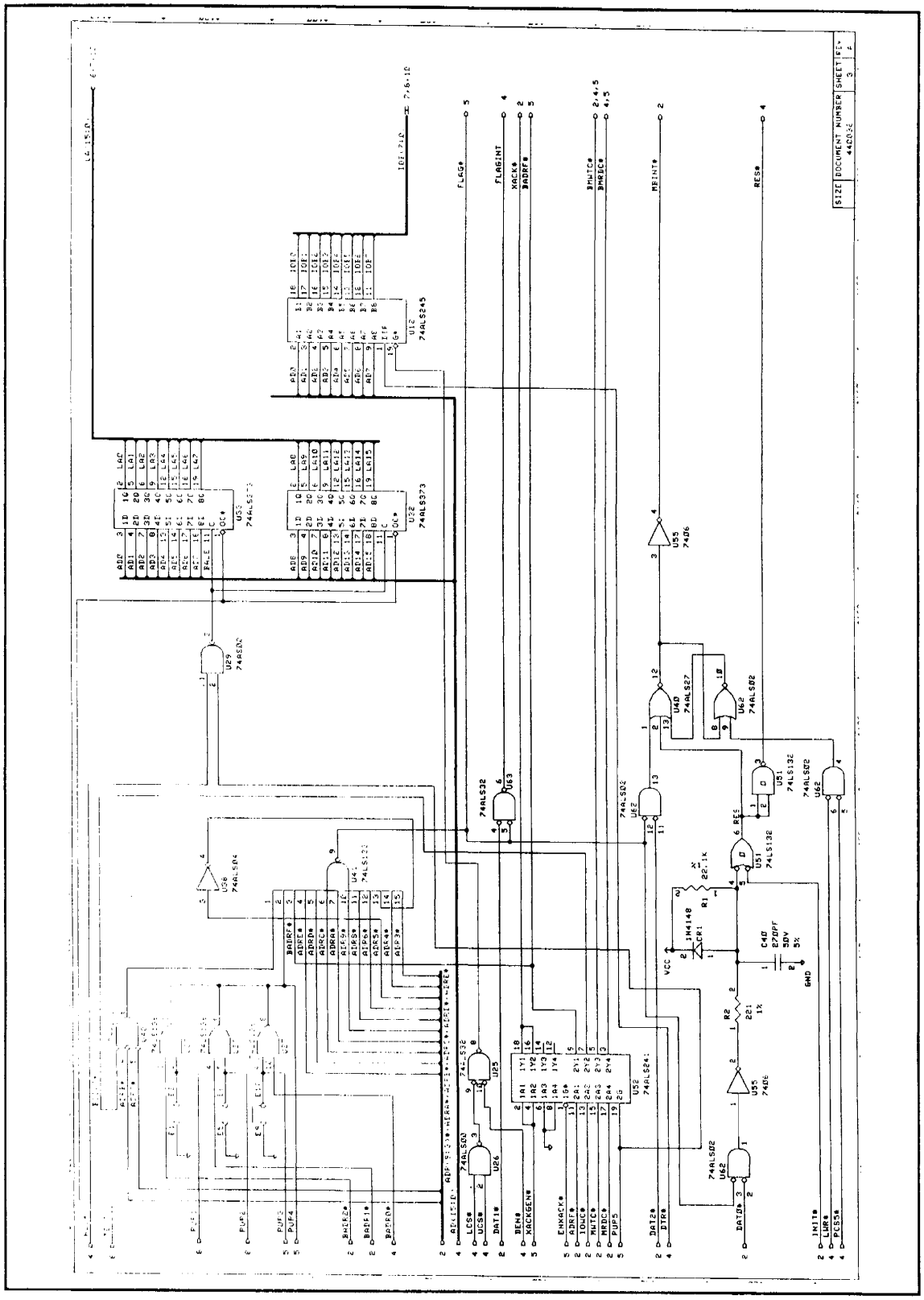


Figure 6-2 iSBC 548 Schematic Diagram (Sheet 3 of 11)

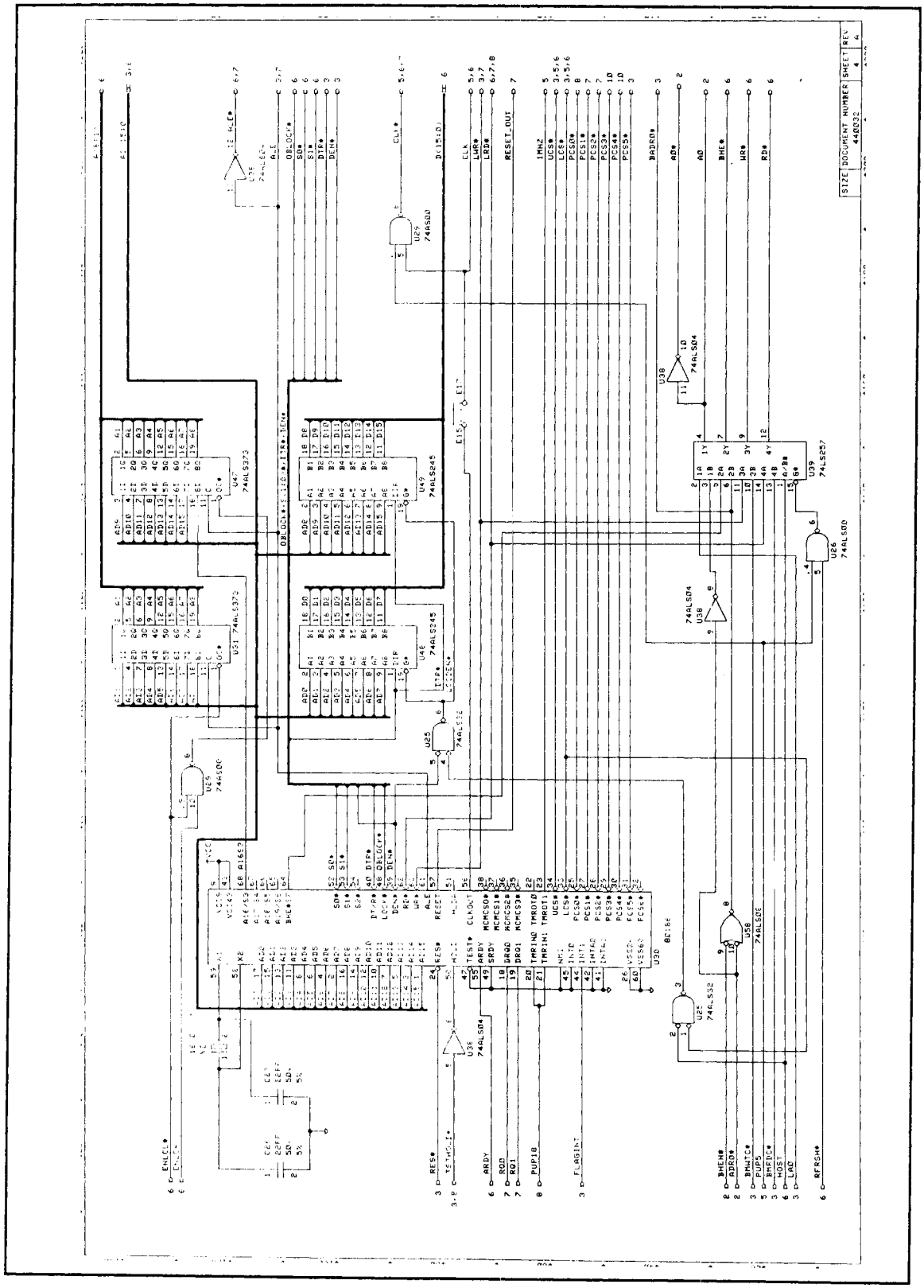


Figure 6-2 iSBC 548 Schematic Diagram (Sheet 4 of 11)

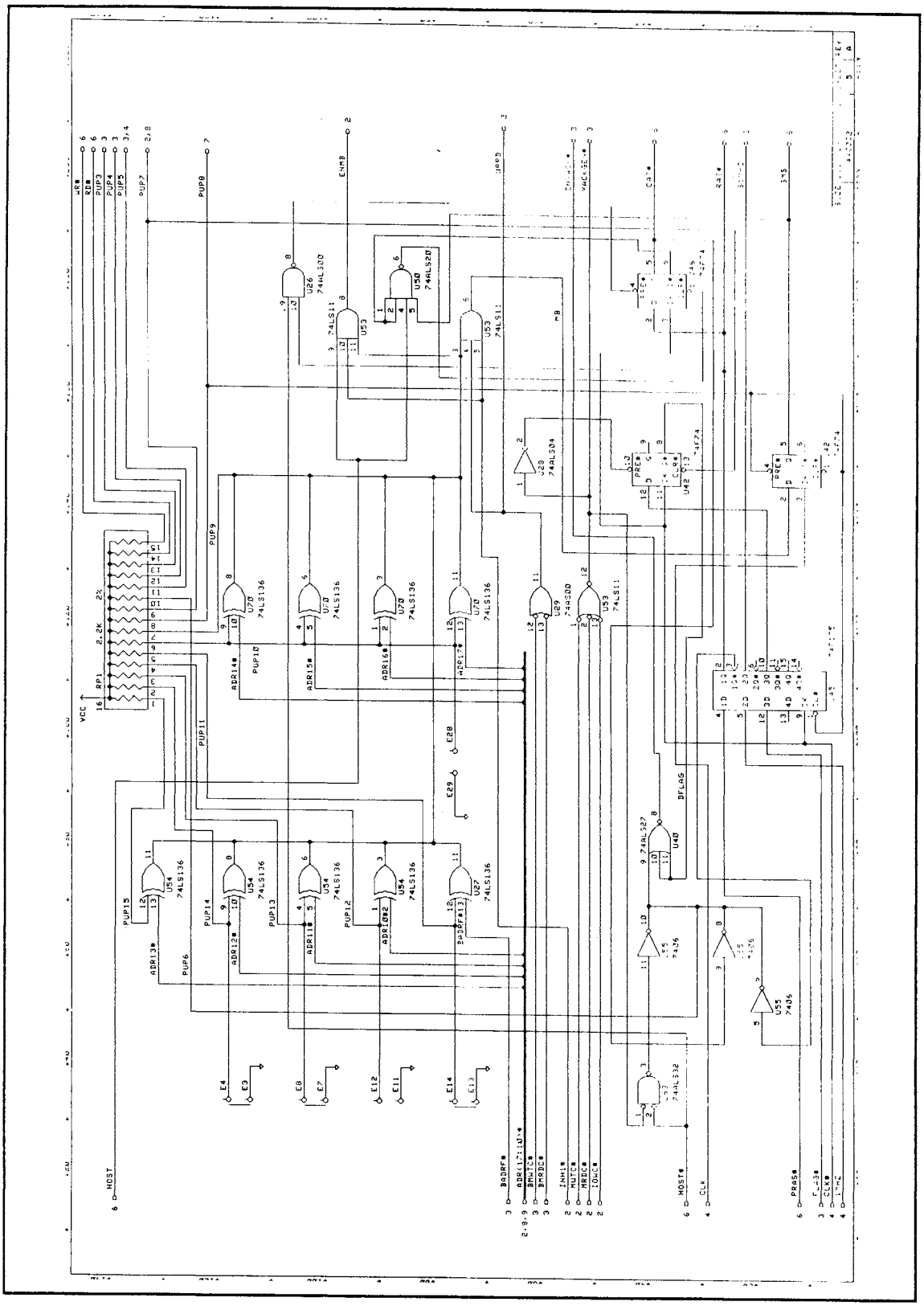


Figure 6-2 iSBC 548 Schematic Diagram (Sheet 5 of 11)

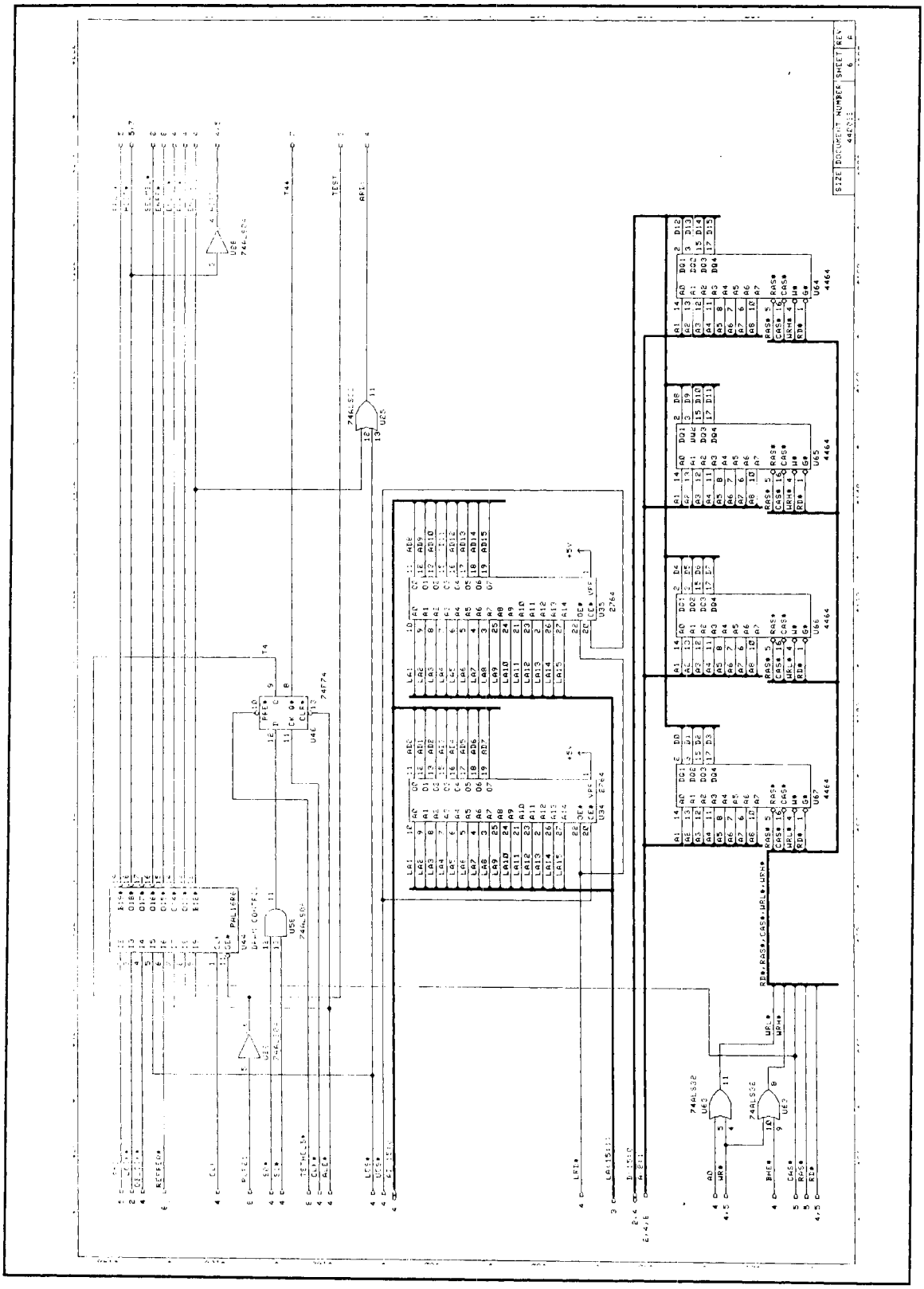


Figure 6-2 iSBC 548 Schematic Diagram (Sheet 6 of 11)

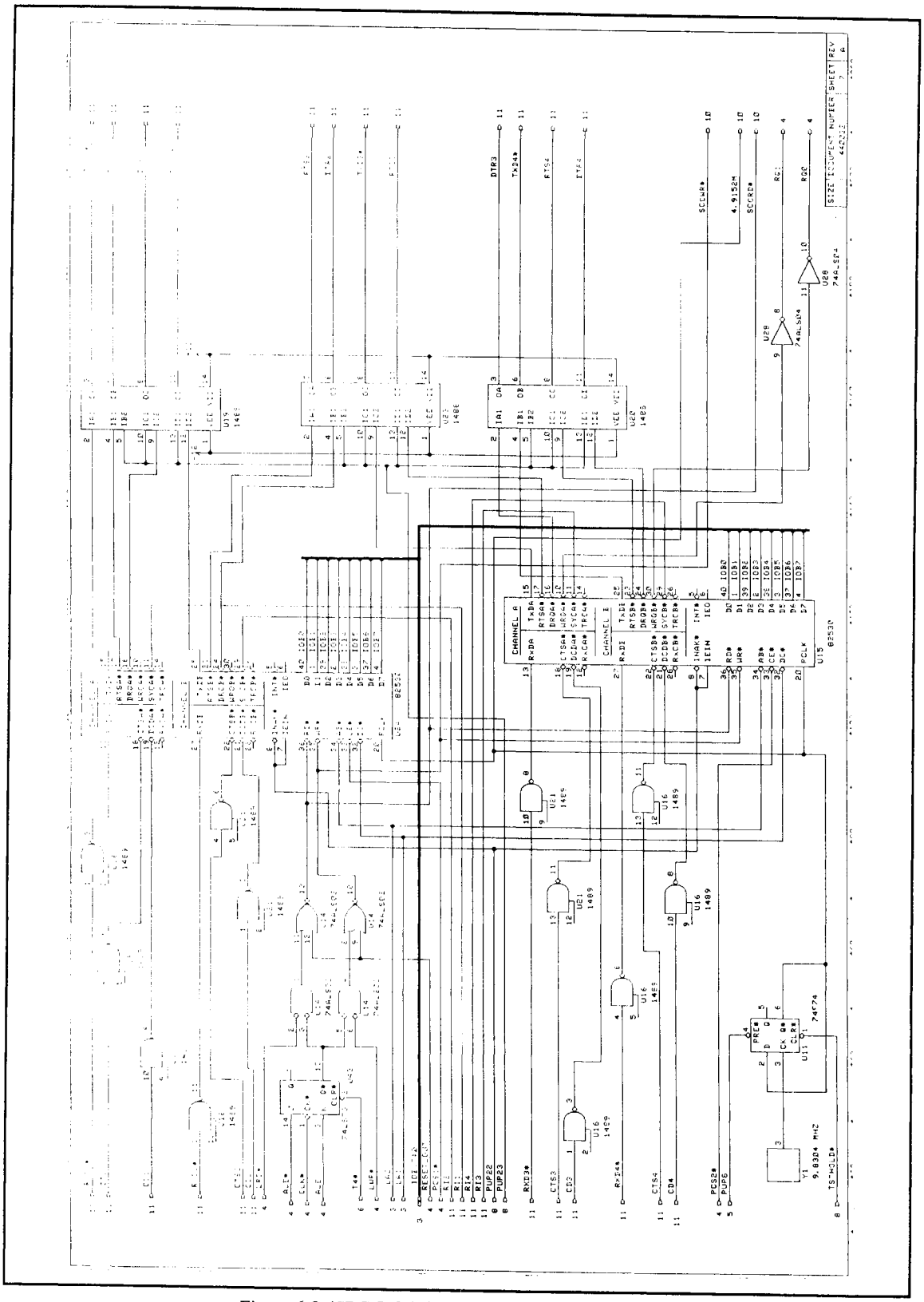


Figure 6-2 iSBC 548 Schematic Diagram (Sheet 7 of 11)

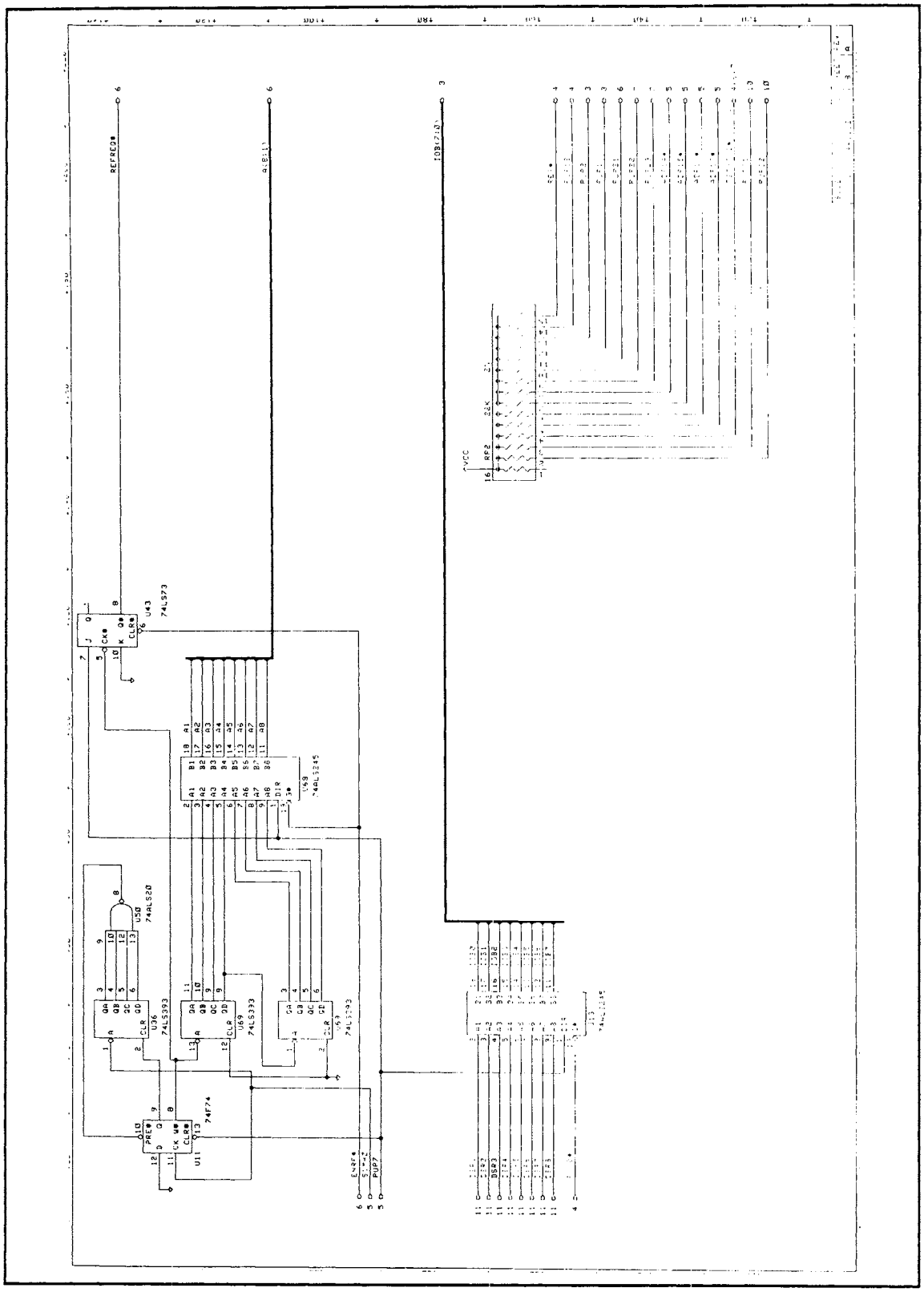


Figure 6-2 iSBC 548 Schematic Diagram (Sheet 8 of 11)

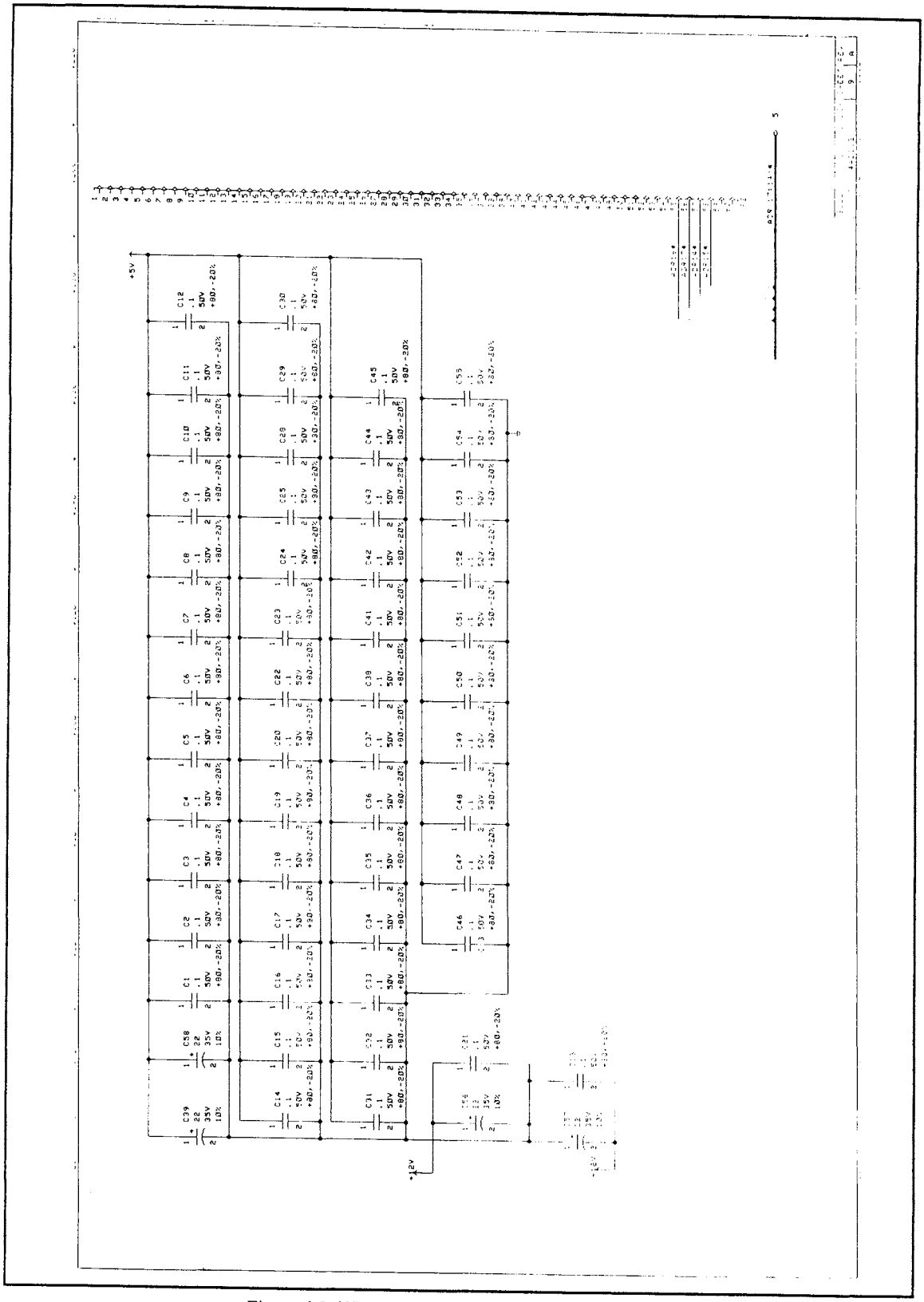


Figure 6-2 iSBC 548 Schematic Diagram (Sheet 9 of 11)

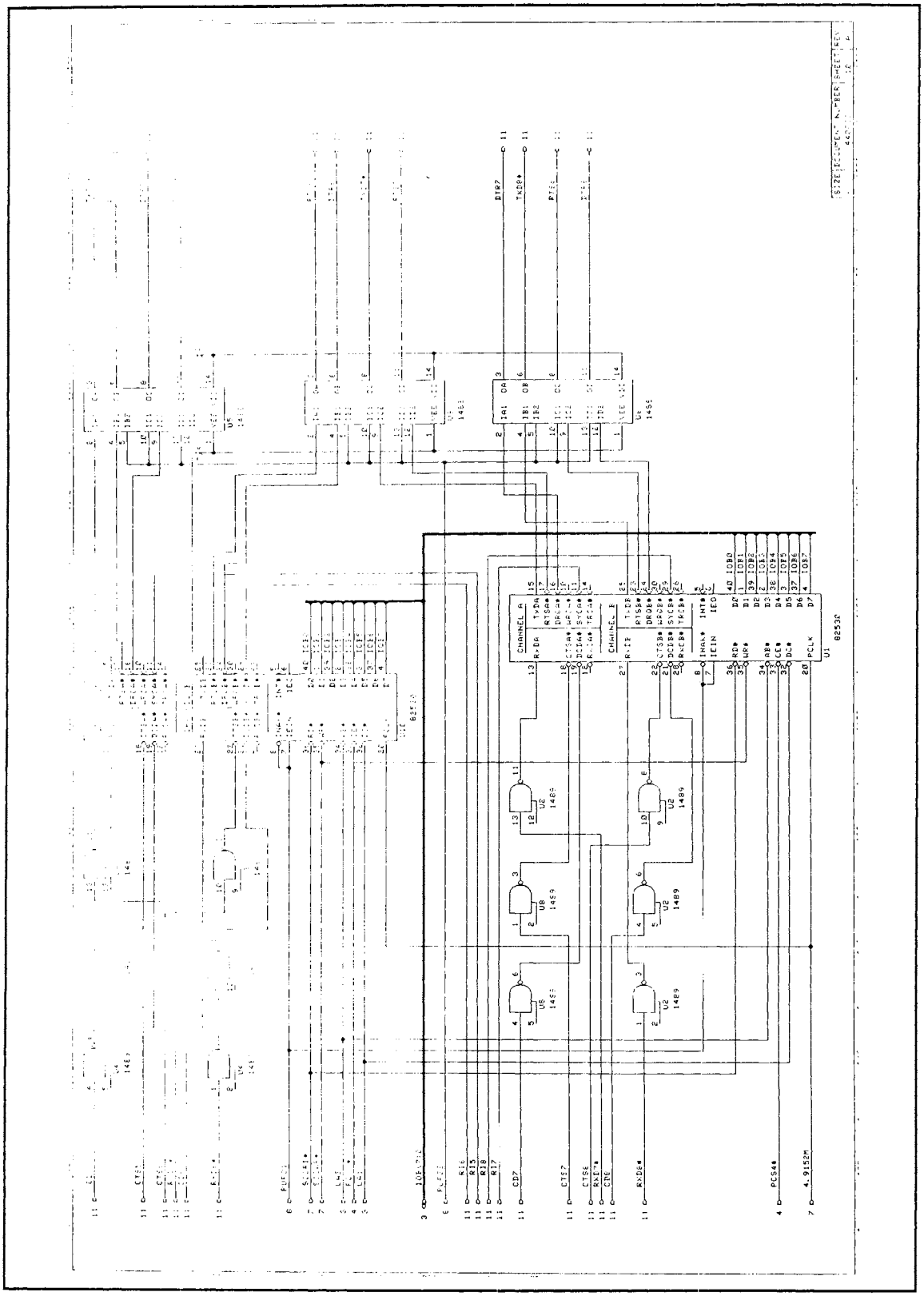


Figure 6-2 iSBC 548 Schematic Diagram (Sheet 10 of 11)

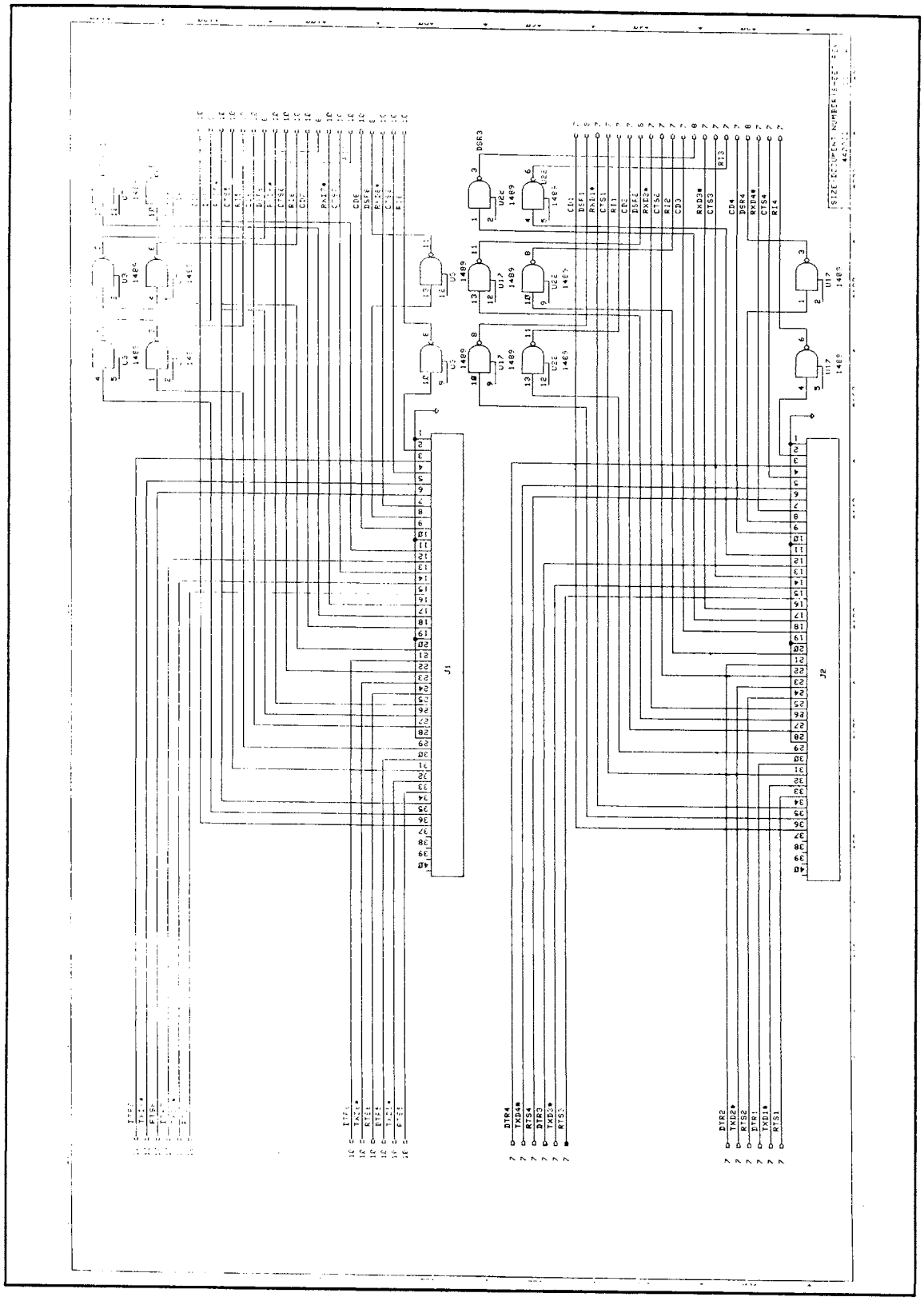


Figure 6-2 iSBC 548 Schematic Diagram (Sheet 11 of 11)

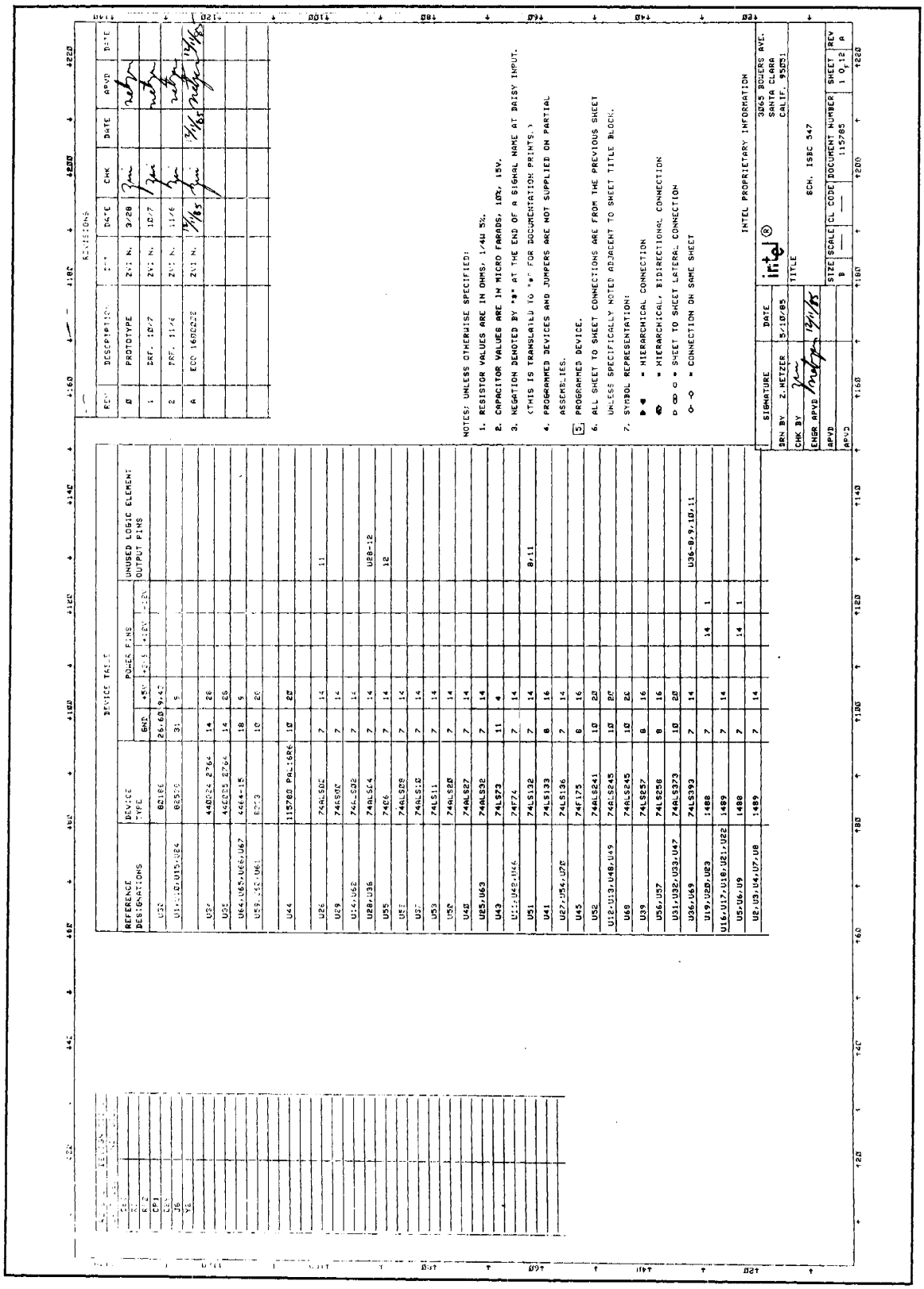


Figure 6-3 iSBC 547 Schematic Diagram (Sheet 1 of 12)

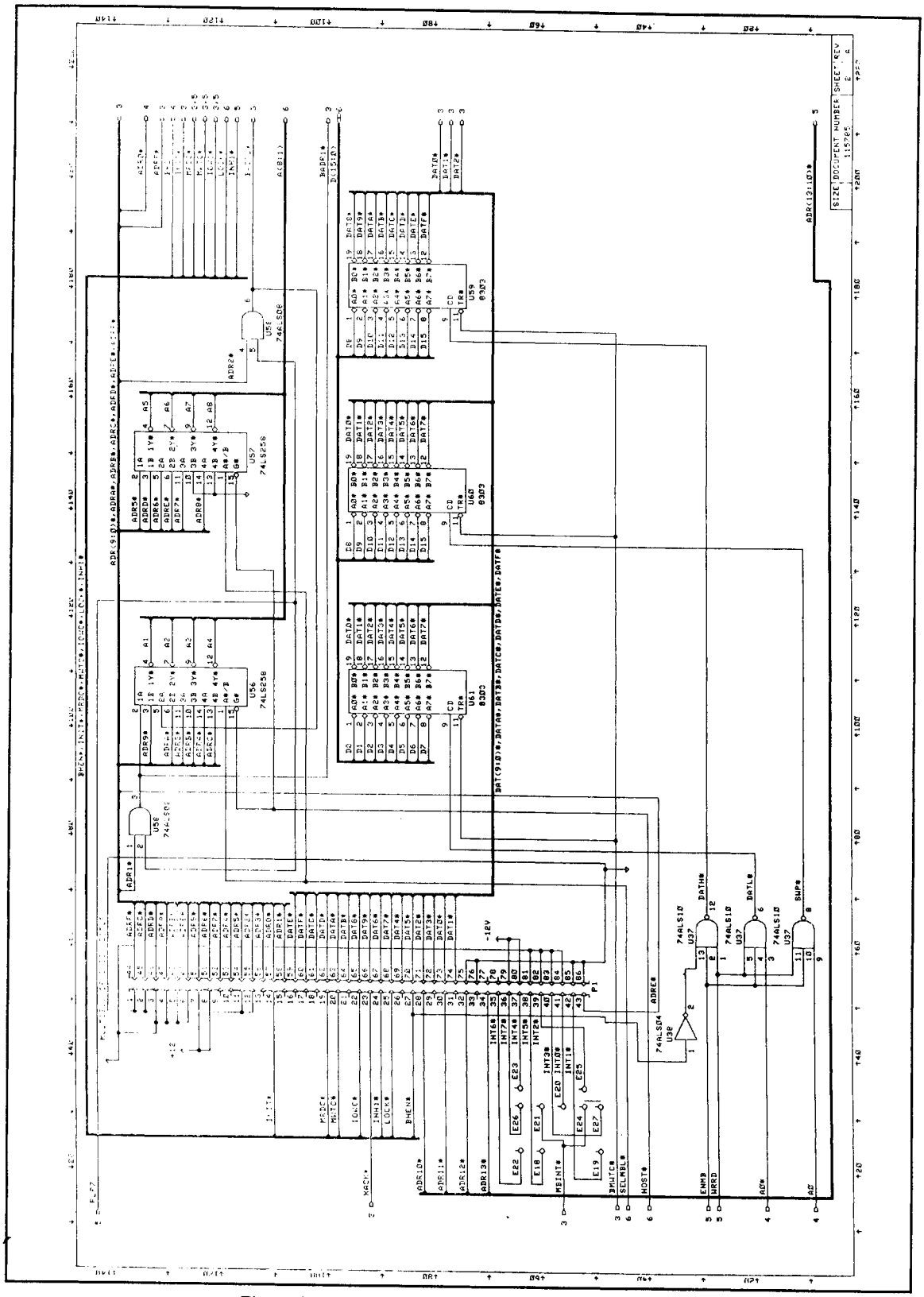


Figure 6-3 iSBC 547 Schematic Diagram (Sheet 2 of 12)

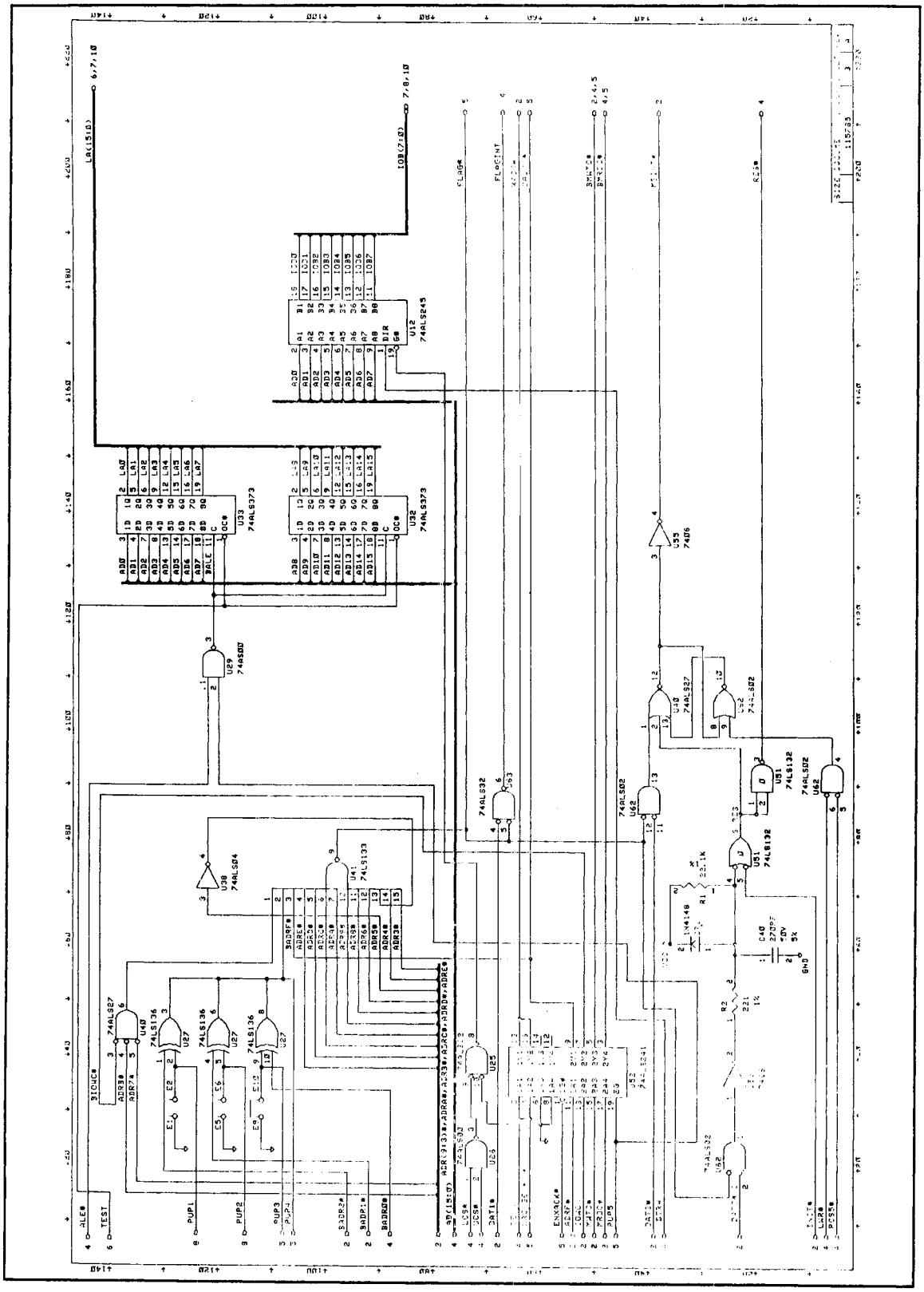


Figure 6-3 iSBC 547 Schematic Diagram (Sheet 3 of 12)

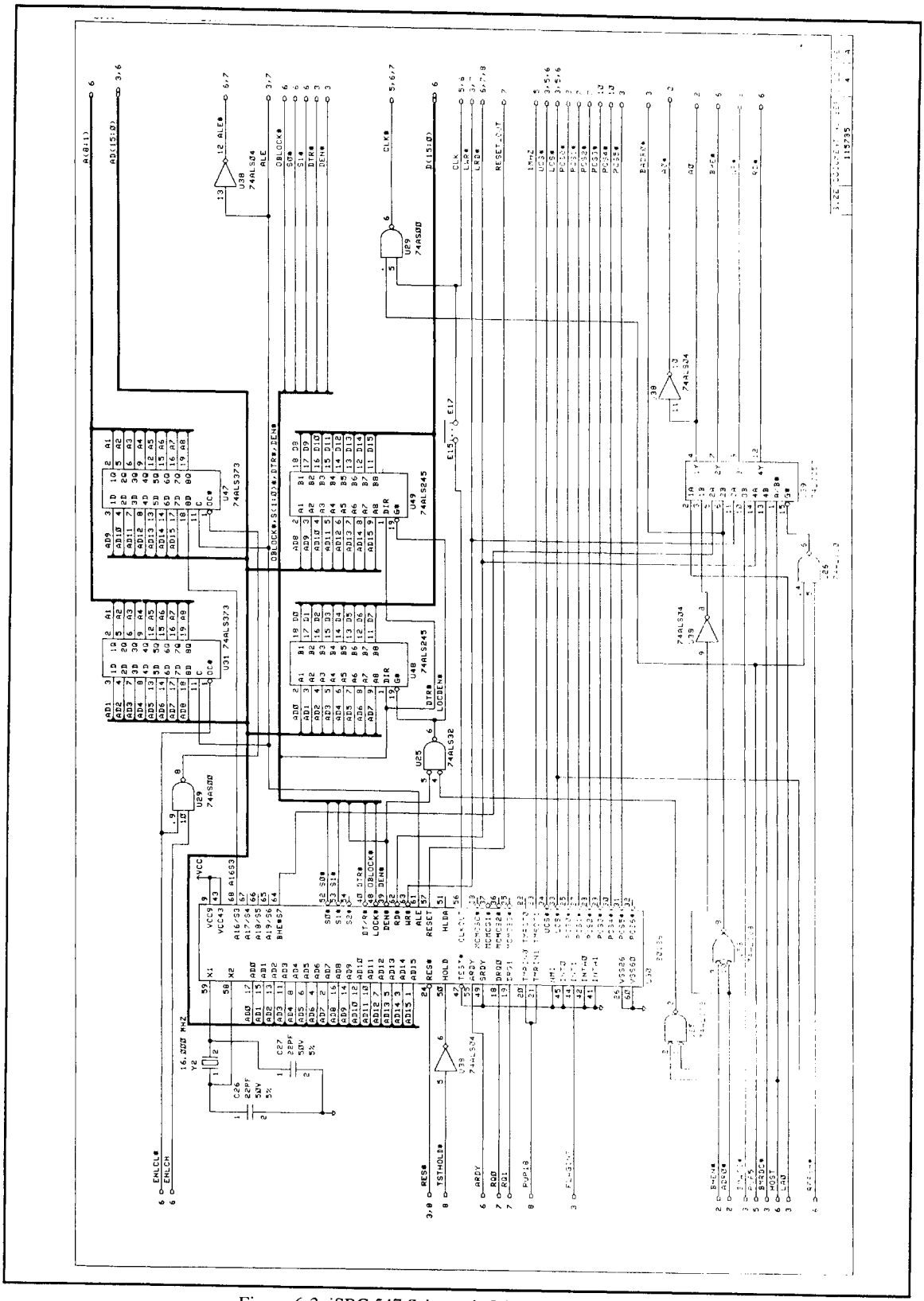


Figure 6-3 iSBC 547 Schematic Diagram (Sheet 4 of 12)

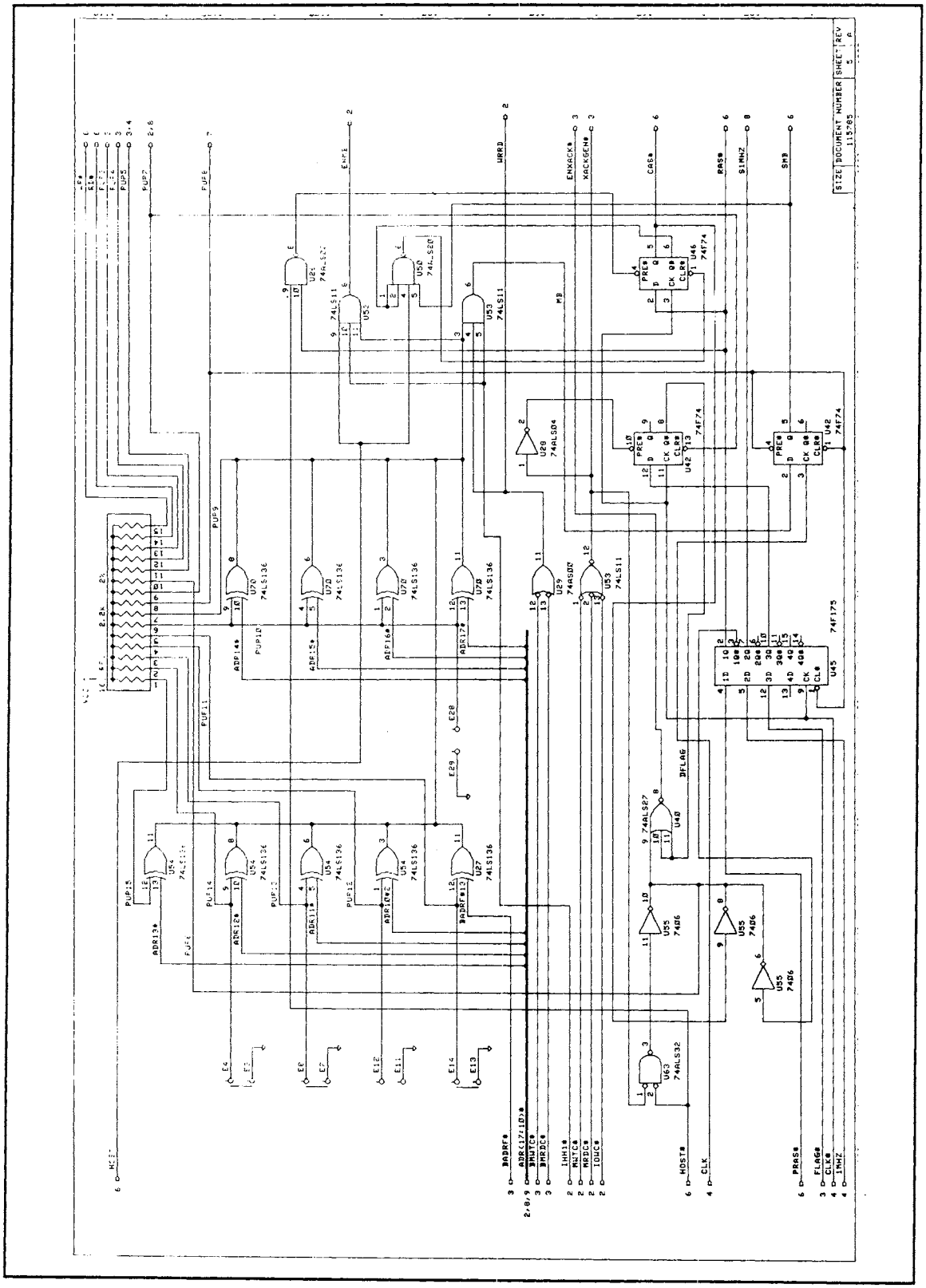


Figure 6-3 iSBC 547 Schematic Diagram (Sheet 5 of 12)

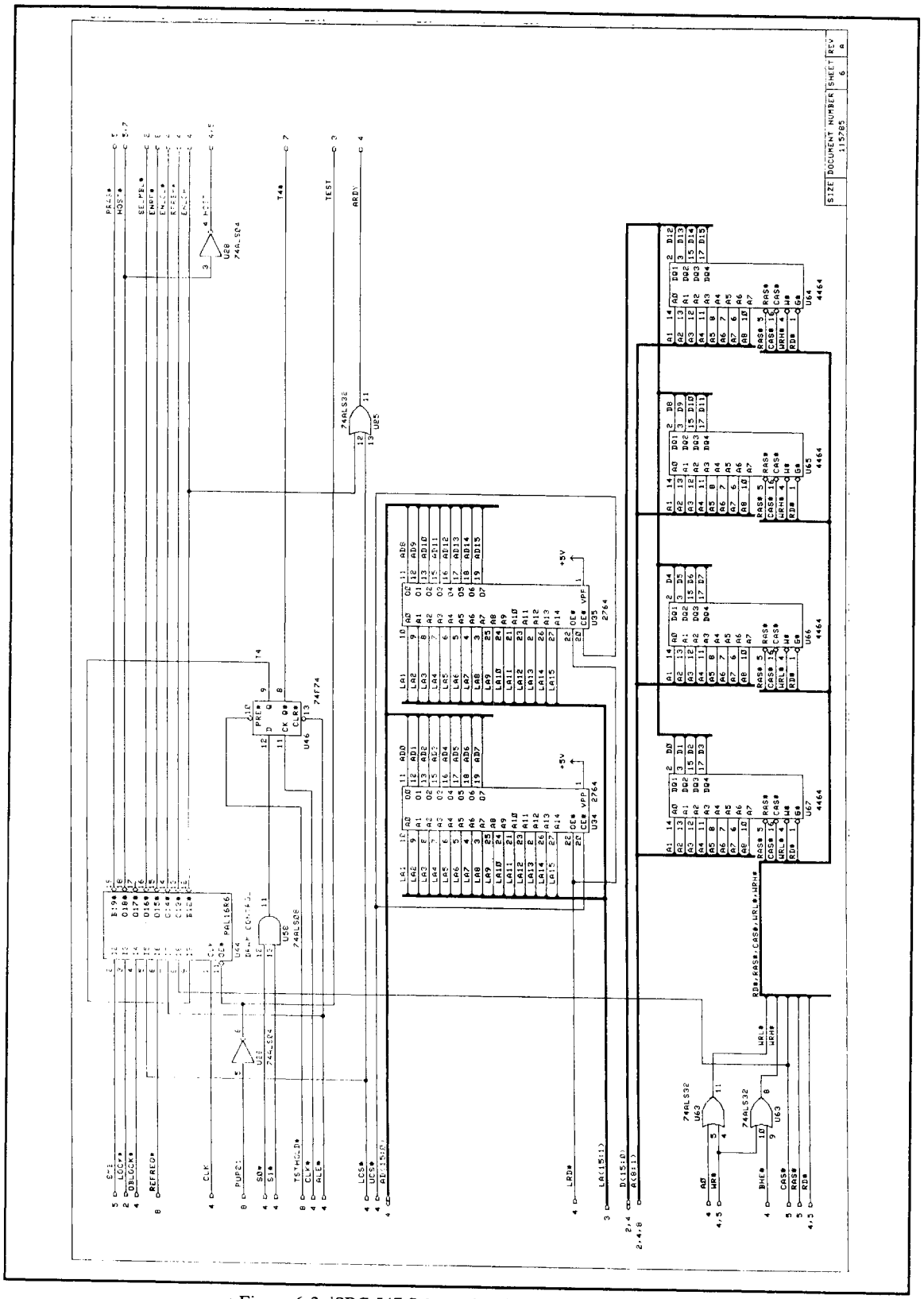


Figure 6-3 iSBC 547 Schematic Diagram (Sheet 6 of 12)

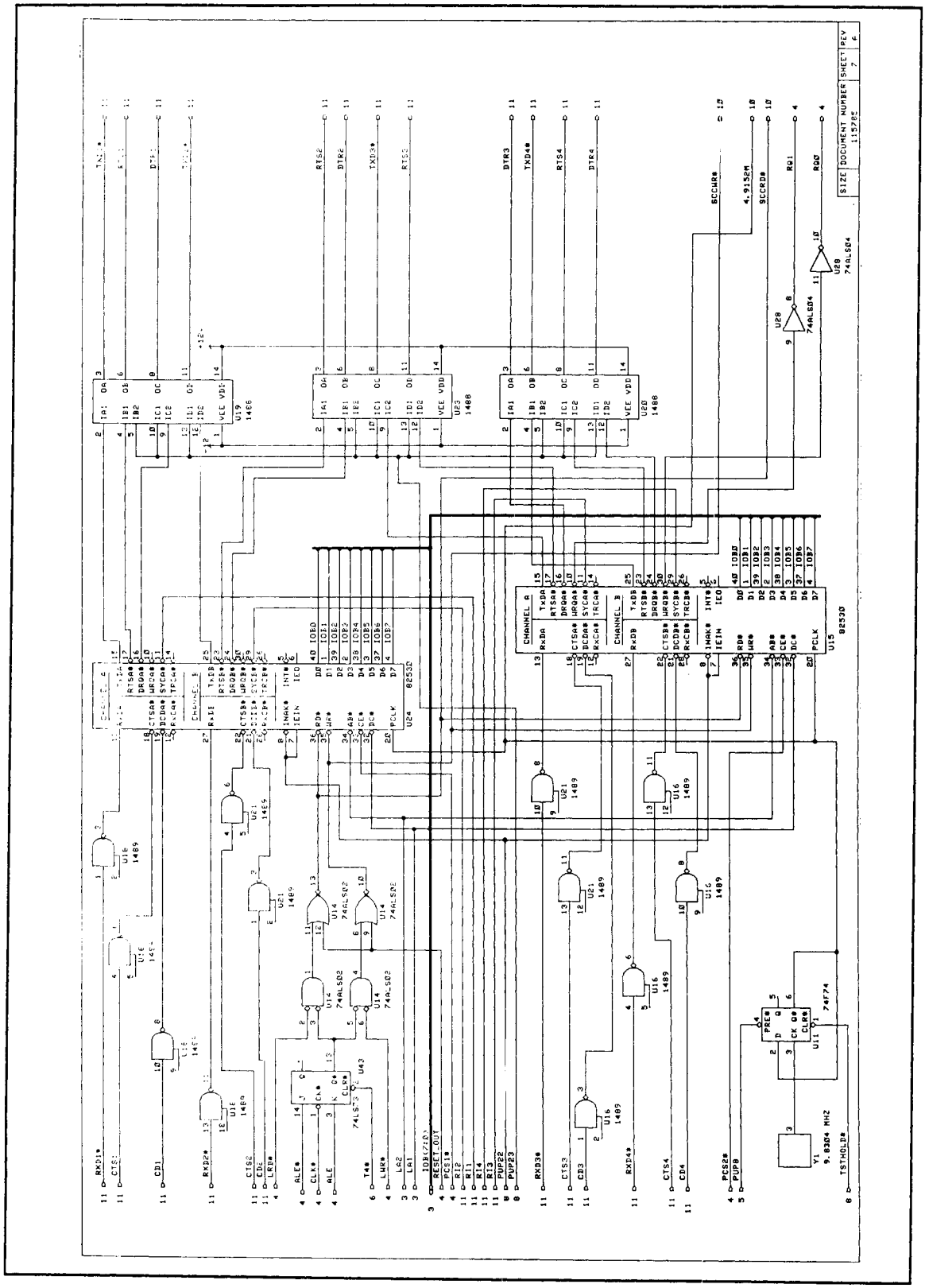


Figure 6-3 iSBC 547 Schematic Diagram (Sheet 7 of 12)

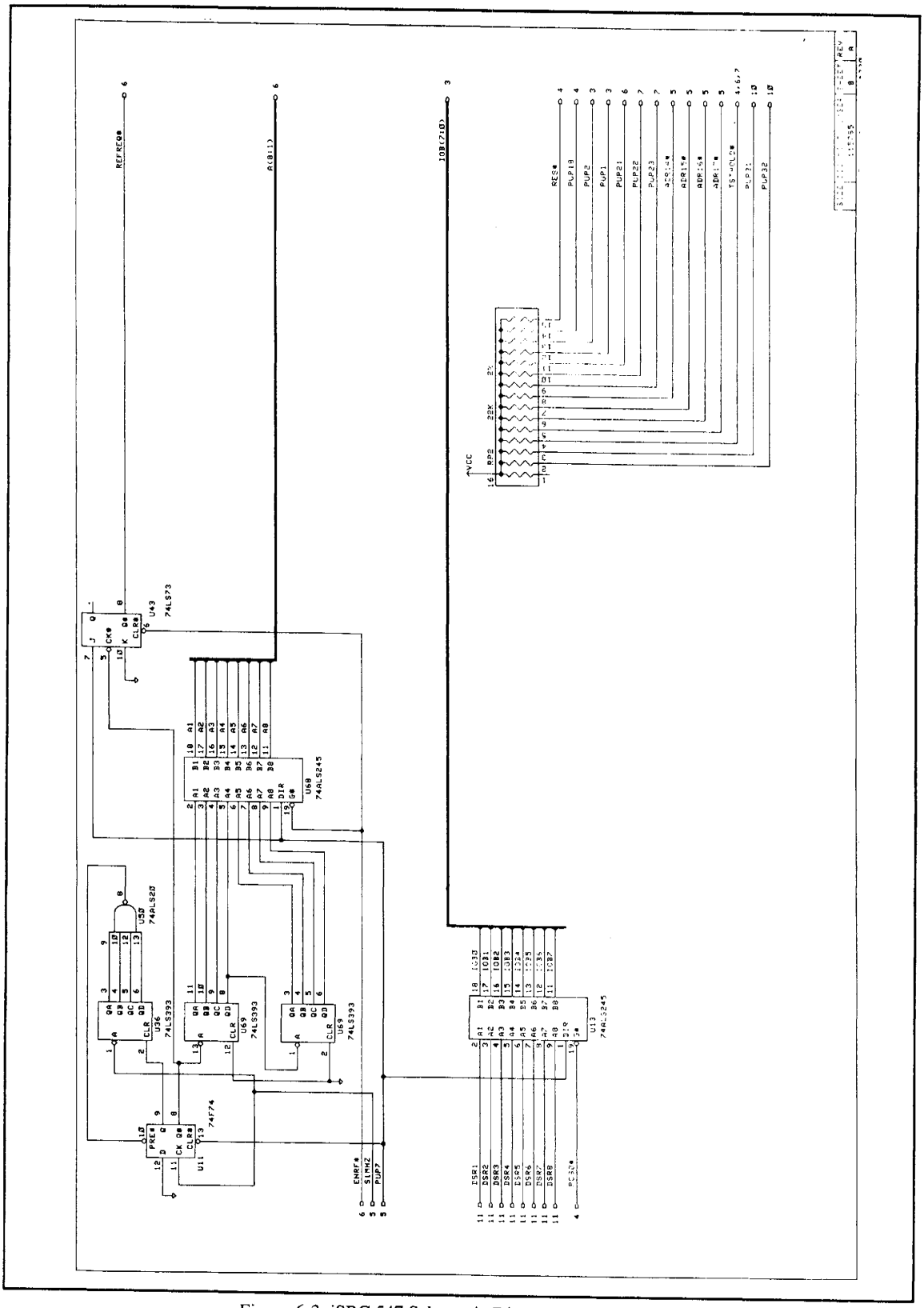


Figure 6-3 iSBC 547 Schematic Diagram (Sheet 8 of 12)

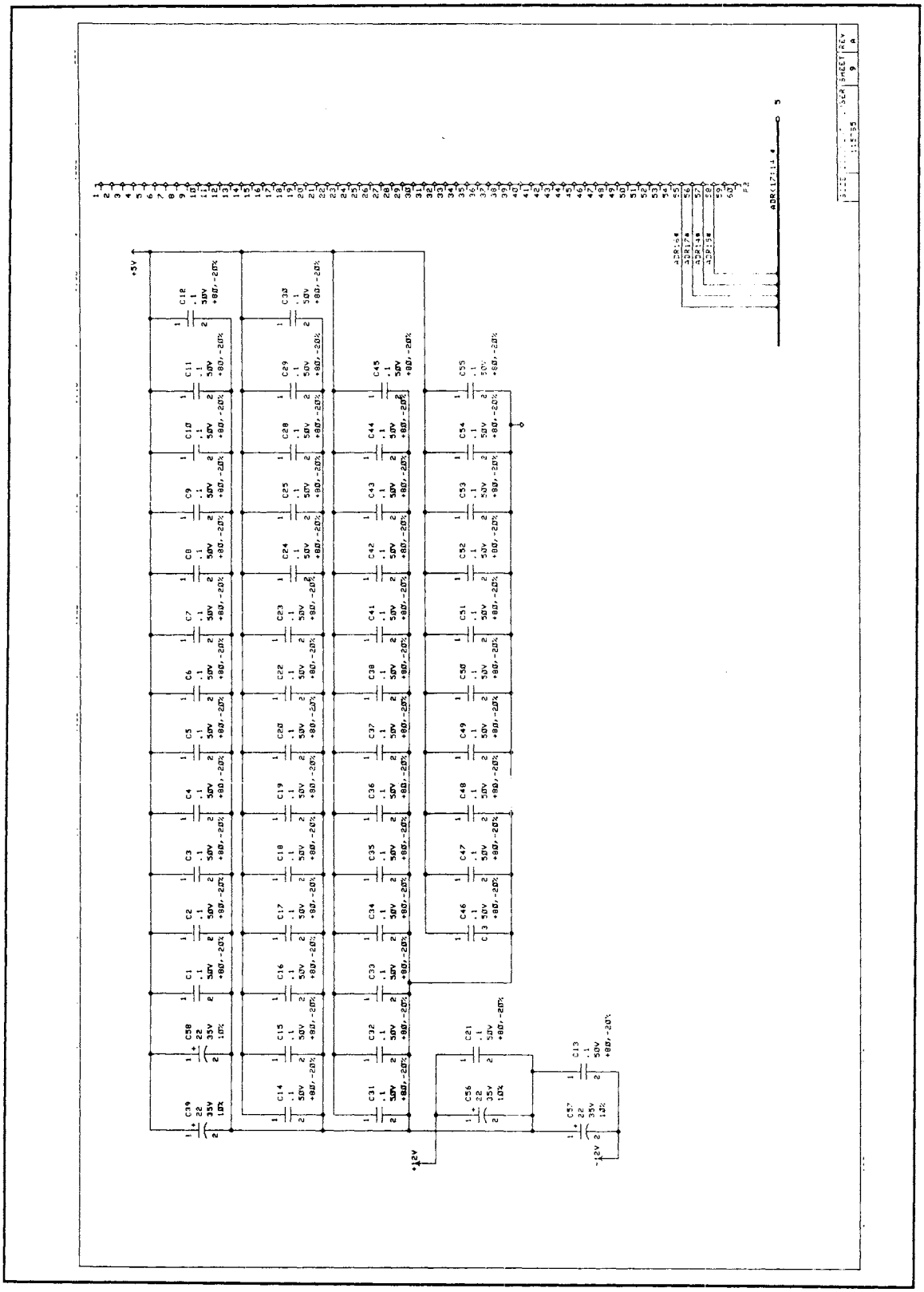


Figure 6-3 iSBC 547 Schematic Diagram (Sheet 9 of 12)

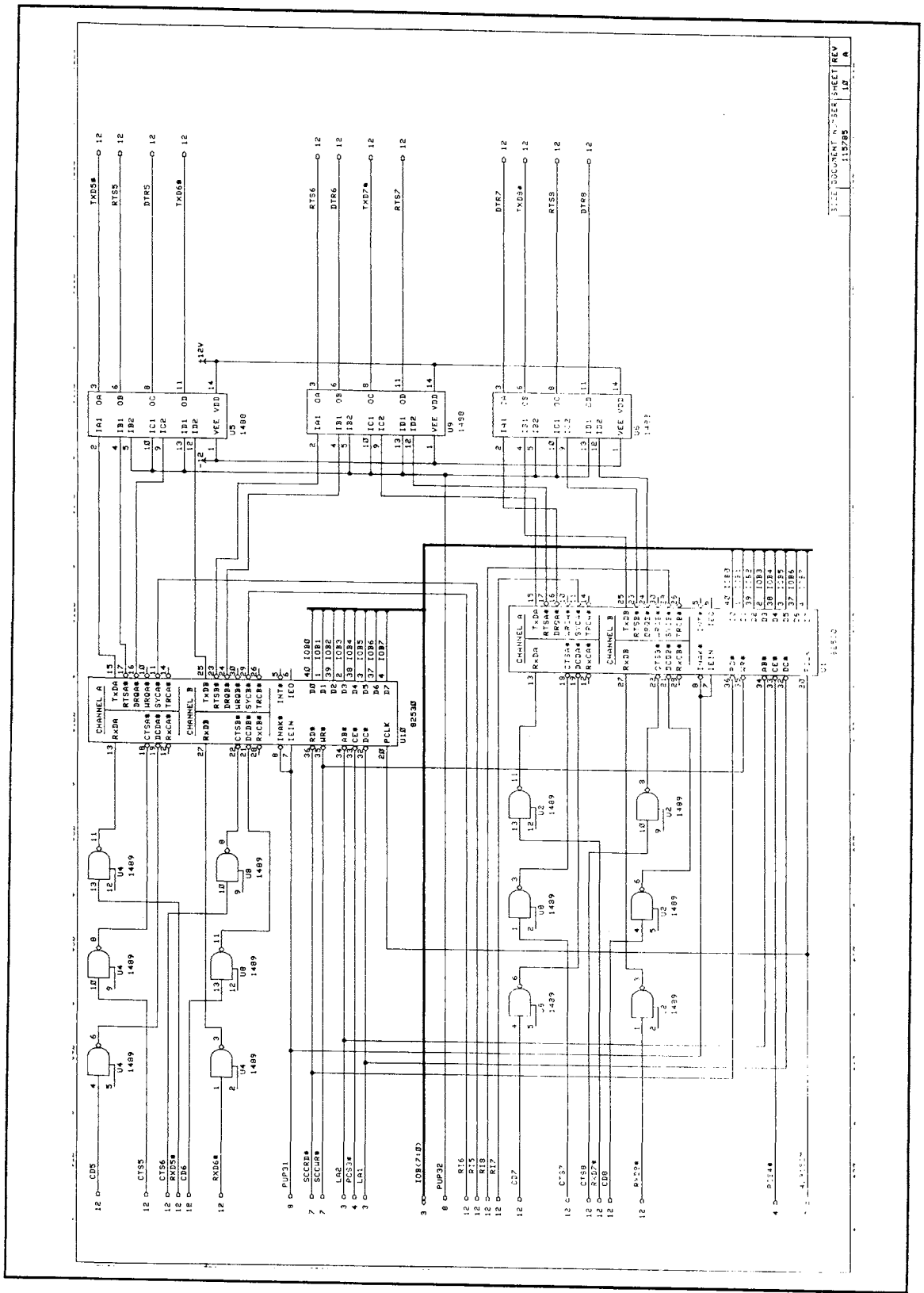


Figure 6-3 iSBC 547 Schematic Diagram (Sheet 10 of 12)

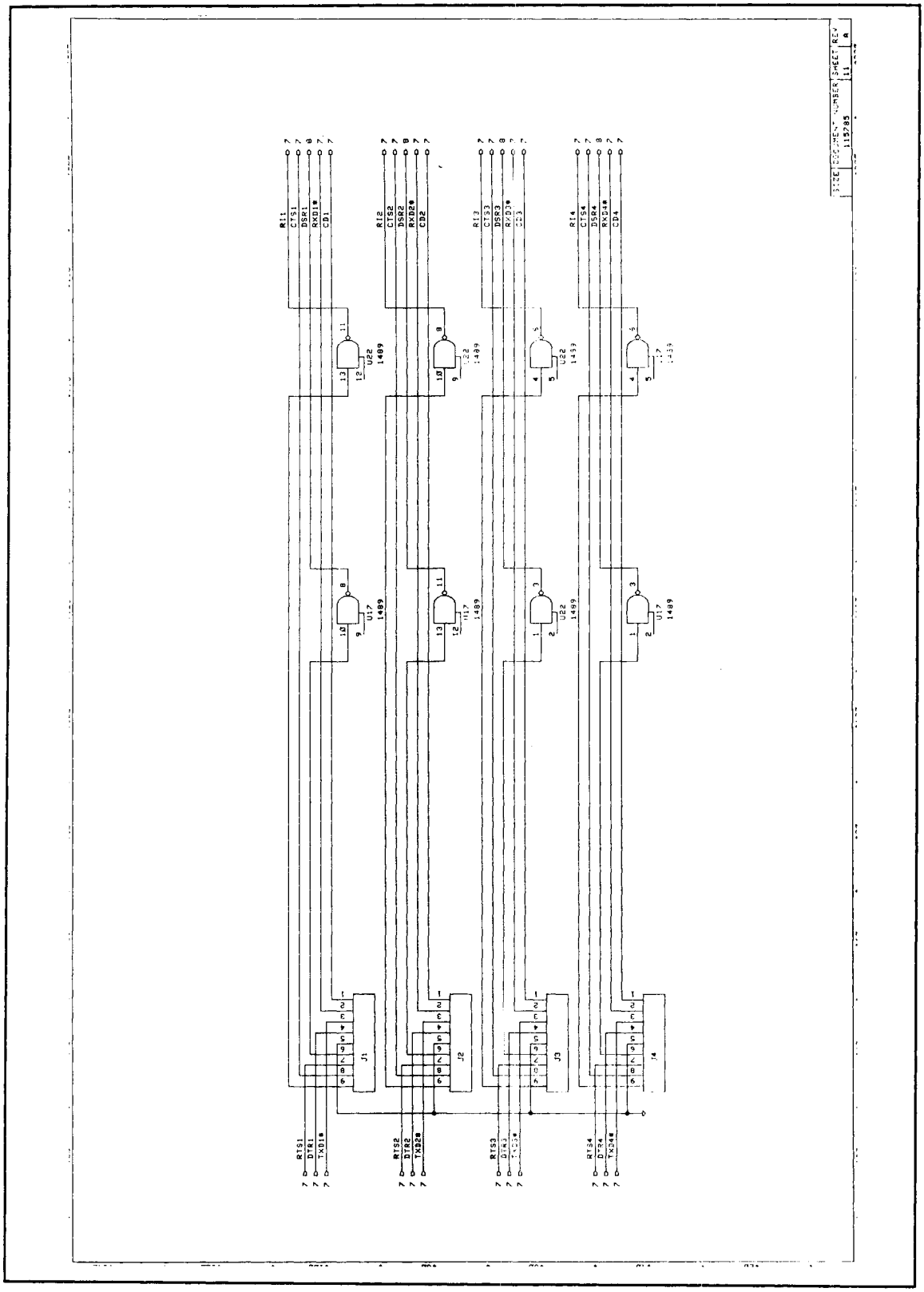


Figure 6-3 iSBC 547 Schematic Diagram (Sheet 11 of 12)

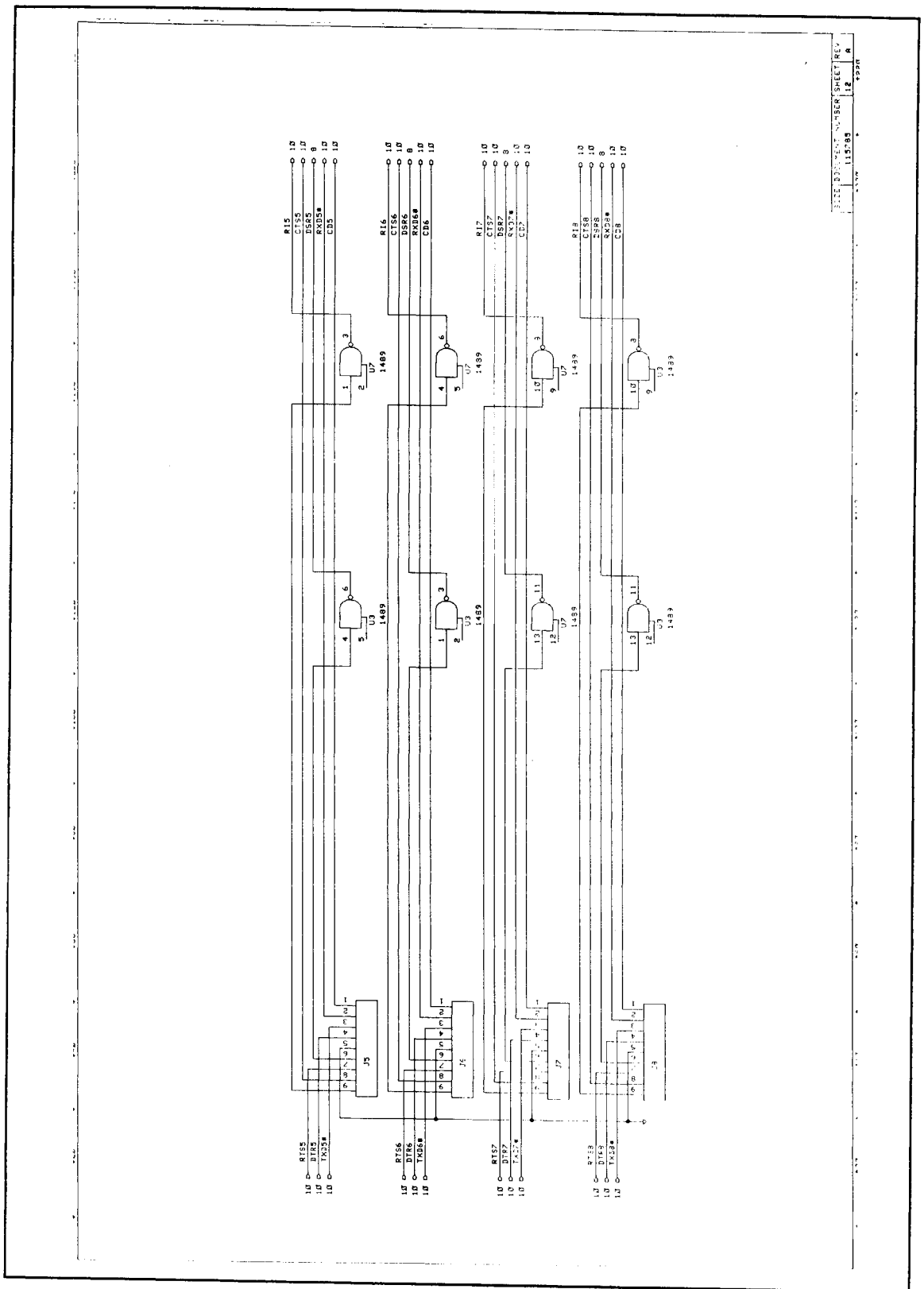


Figure 6-3 iSBC 547 Schematic Diagram (Sheet 12 of 12)

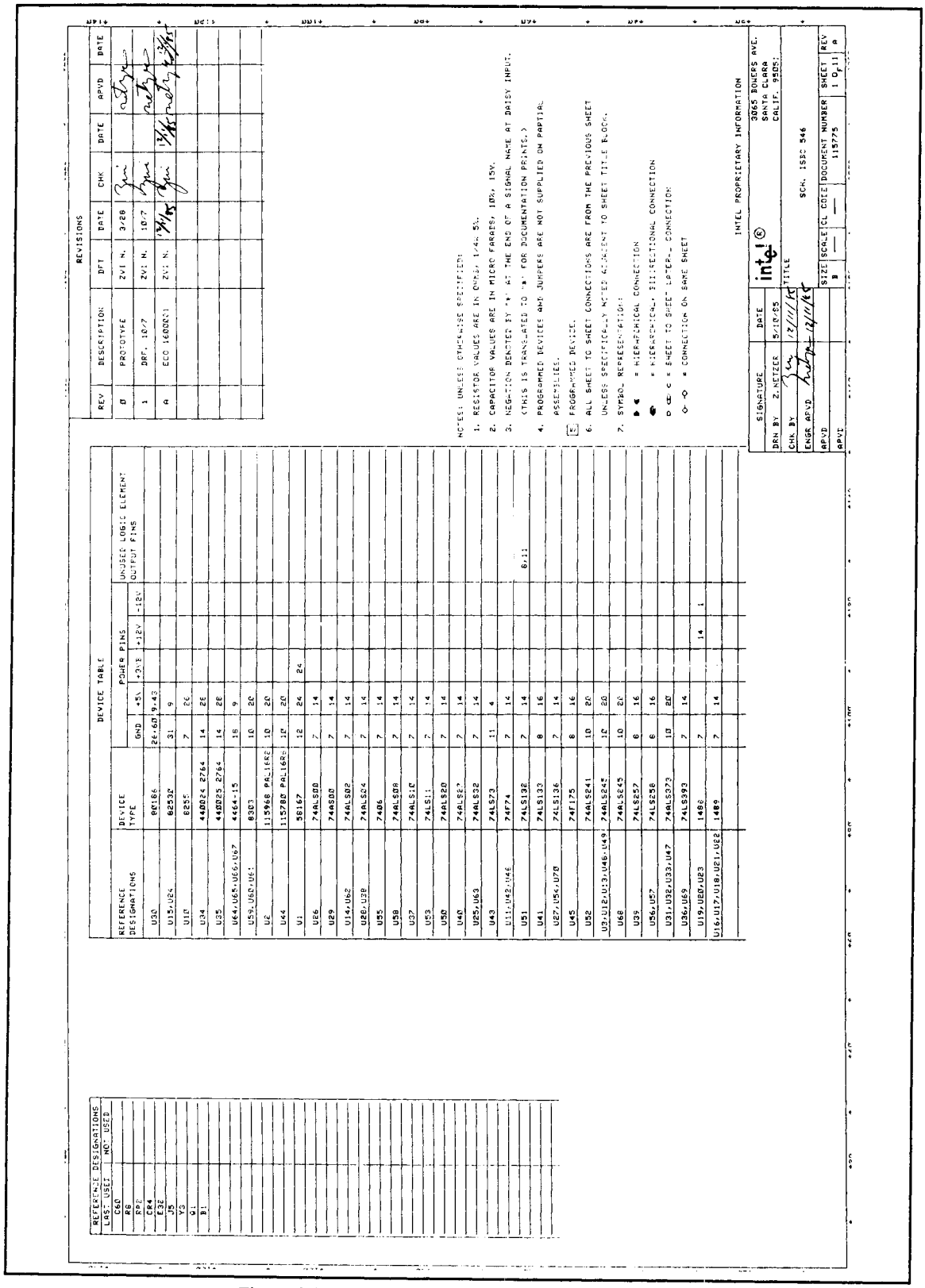


Figure 6-4 iSBC 546 Schematic Diagram (Sheet 1 of 11)

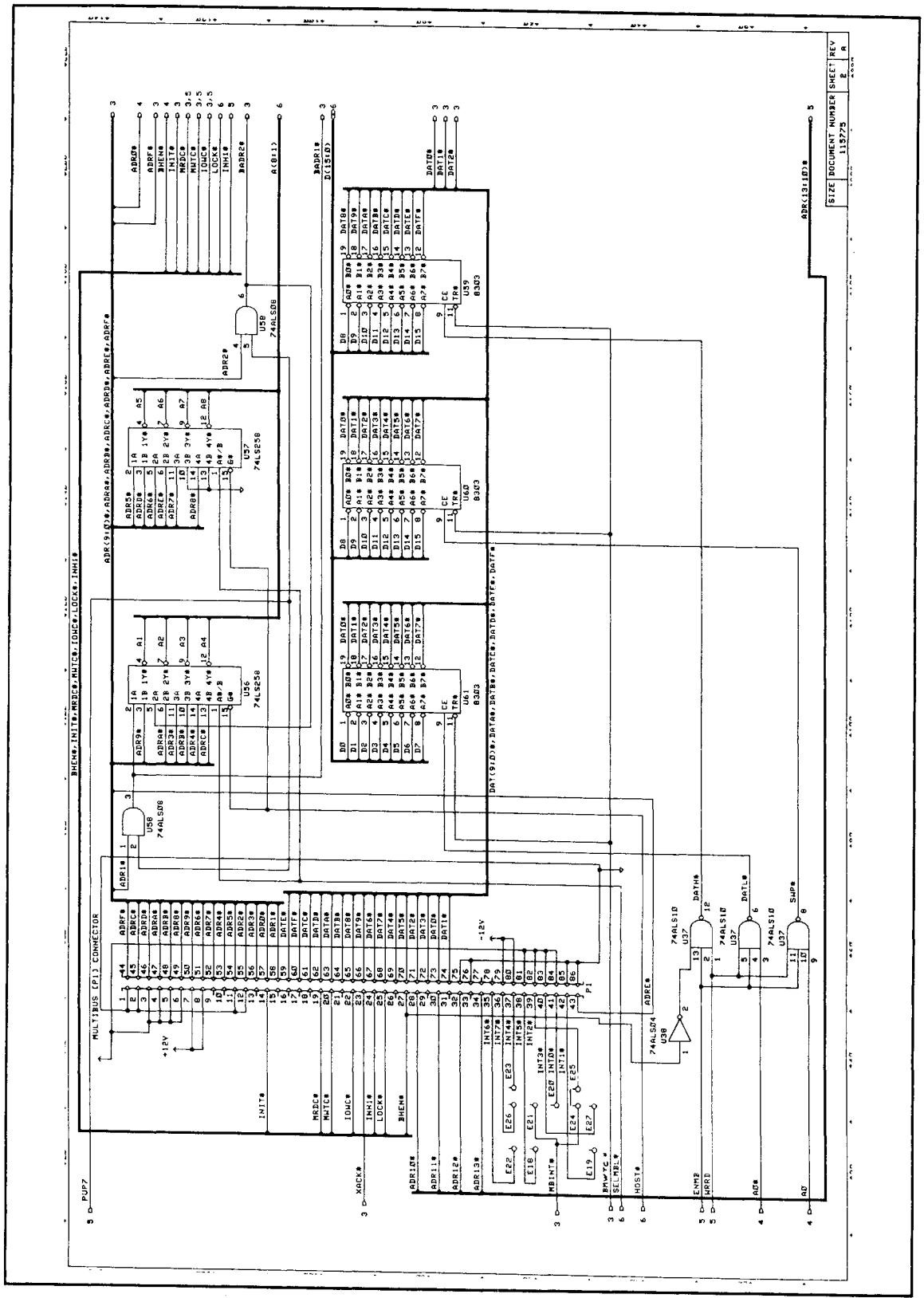


Figure 6-4 iSBC 546 Schematic Diagram (Sheet 2 of 11)

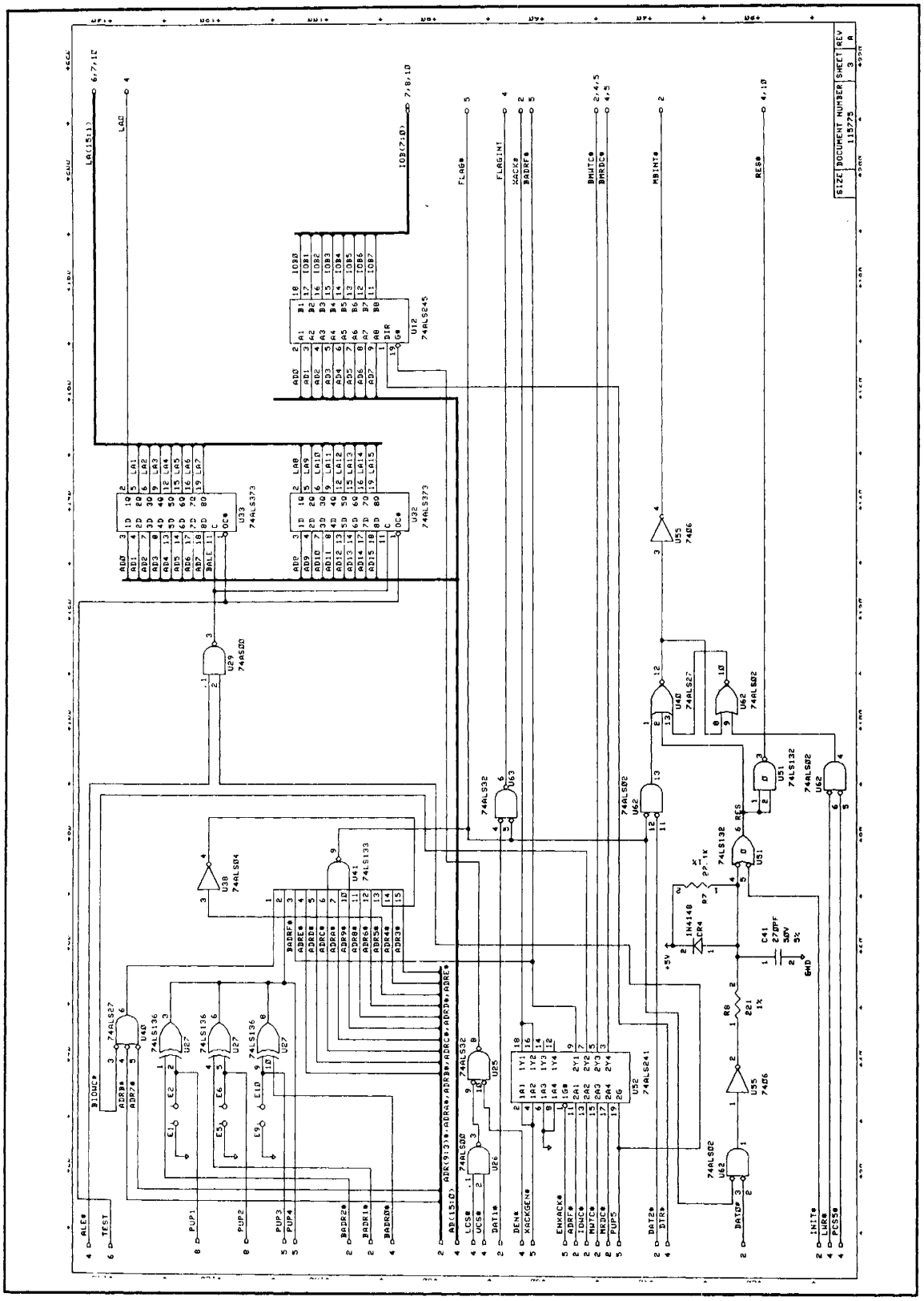


Figure 6-4 iSBC 546 Schematic Diagram (Sheet 3 of 11)

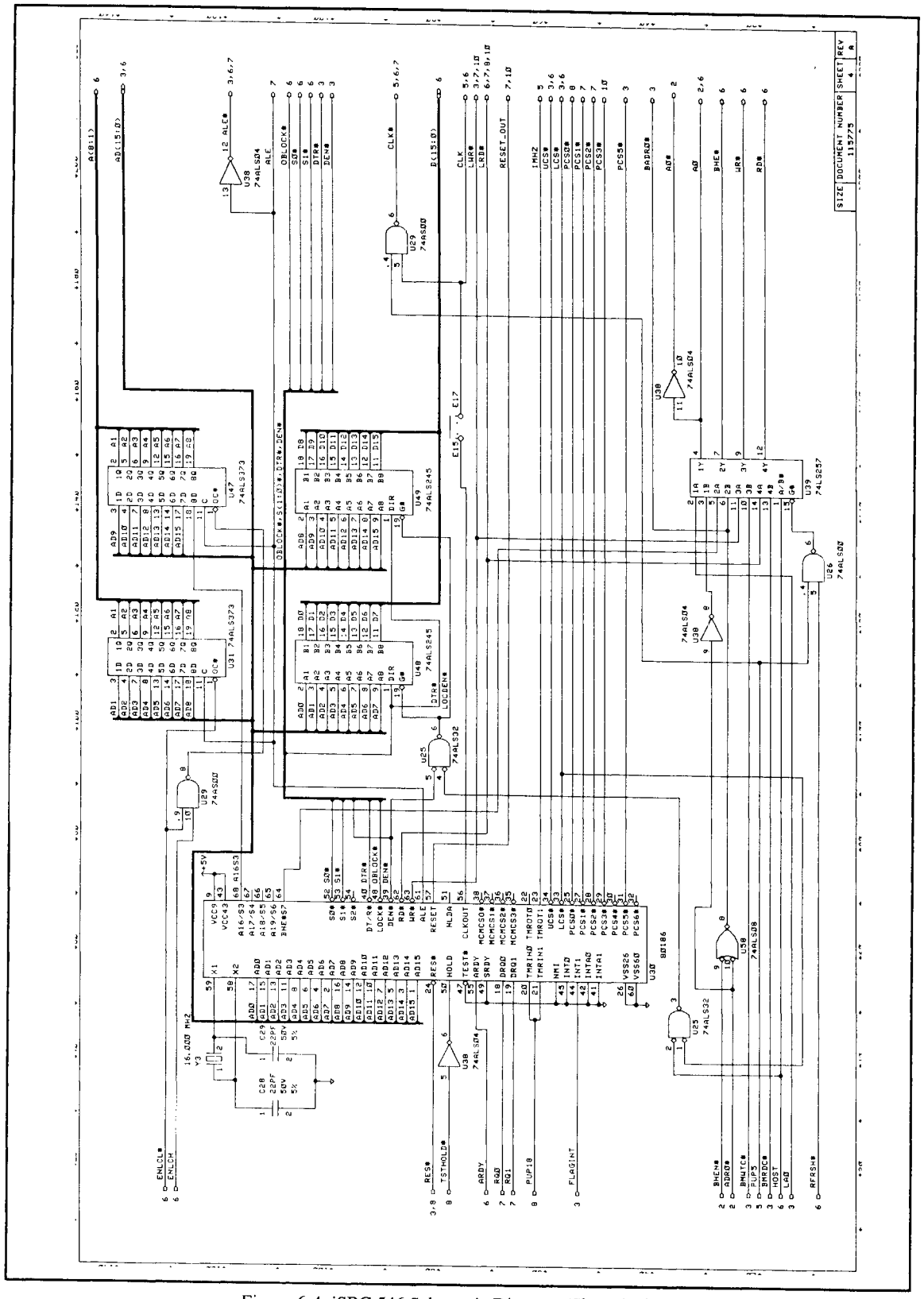


Figure 6-4 iSBC 546 Schematic Diagram (Sheet 4 of 11)

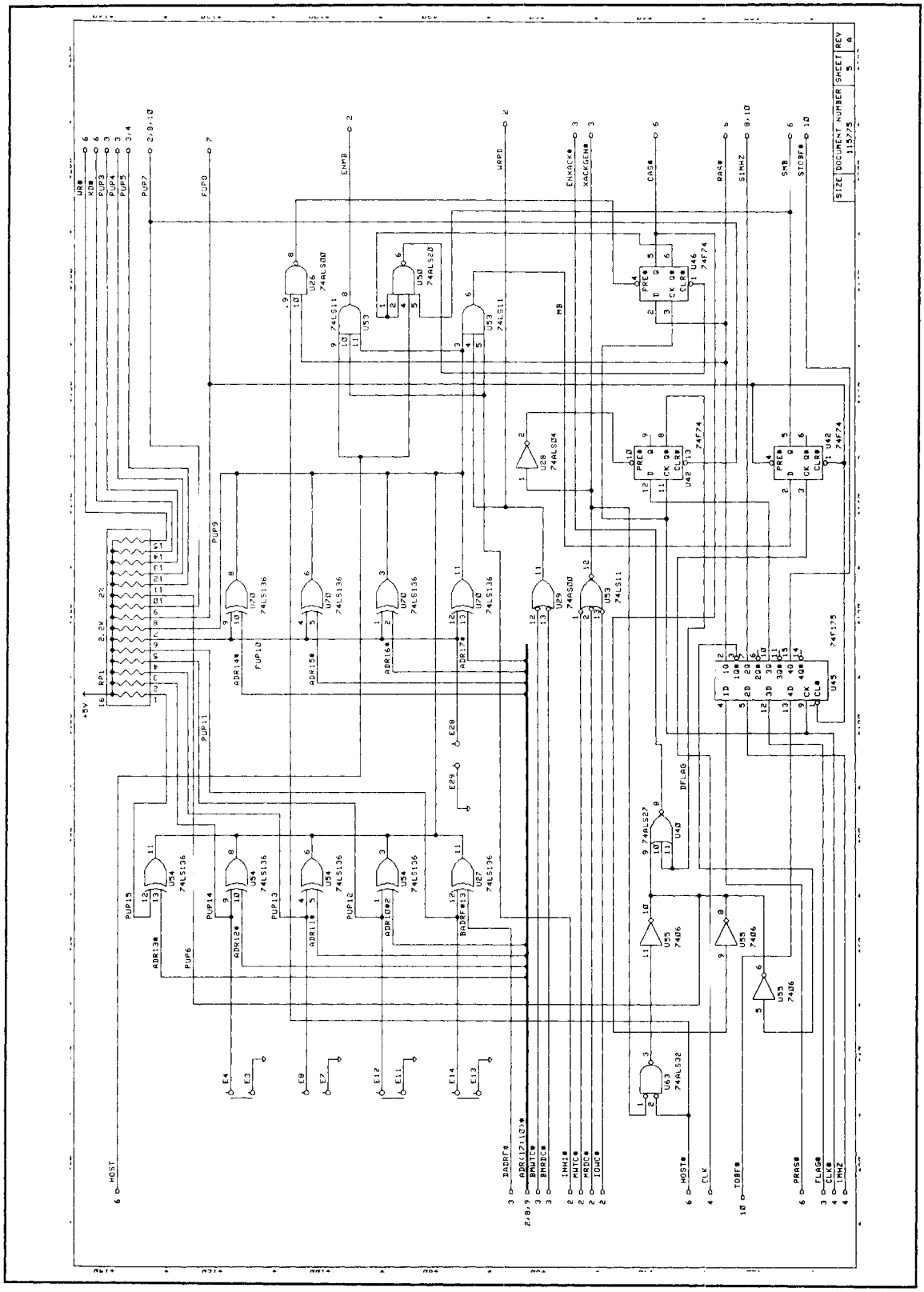


Figure 6-4 iSBC 546 Schematic Diagram (Sheet 5 of 11)

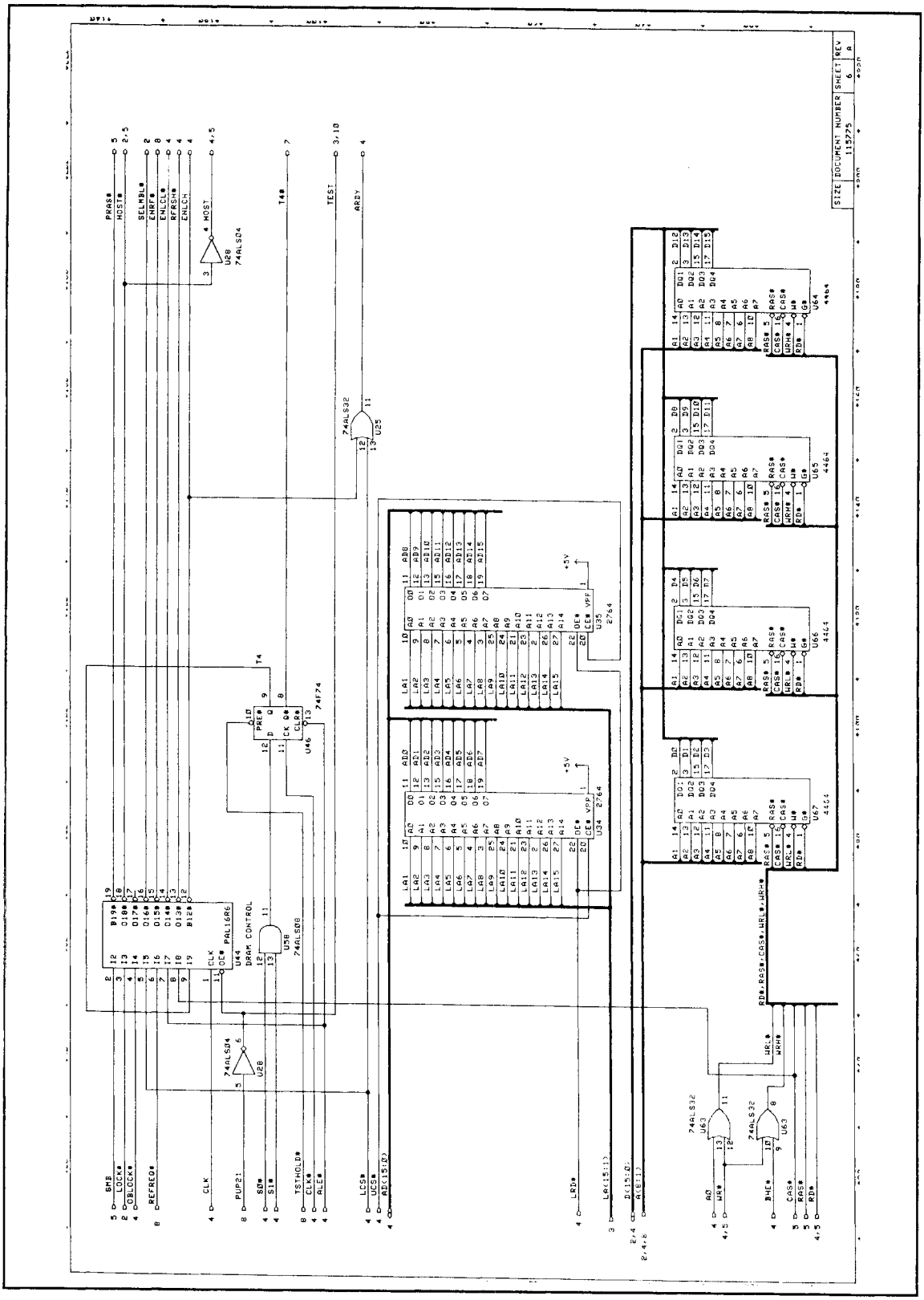


Figure 6-4 iSBC 546 Schematic Diagram (Sheet 6 of 11)

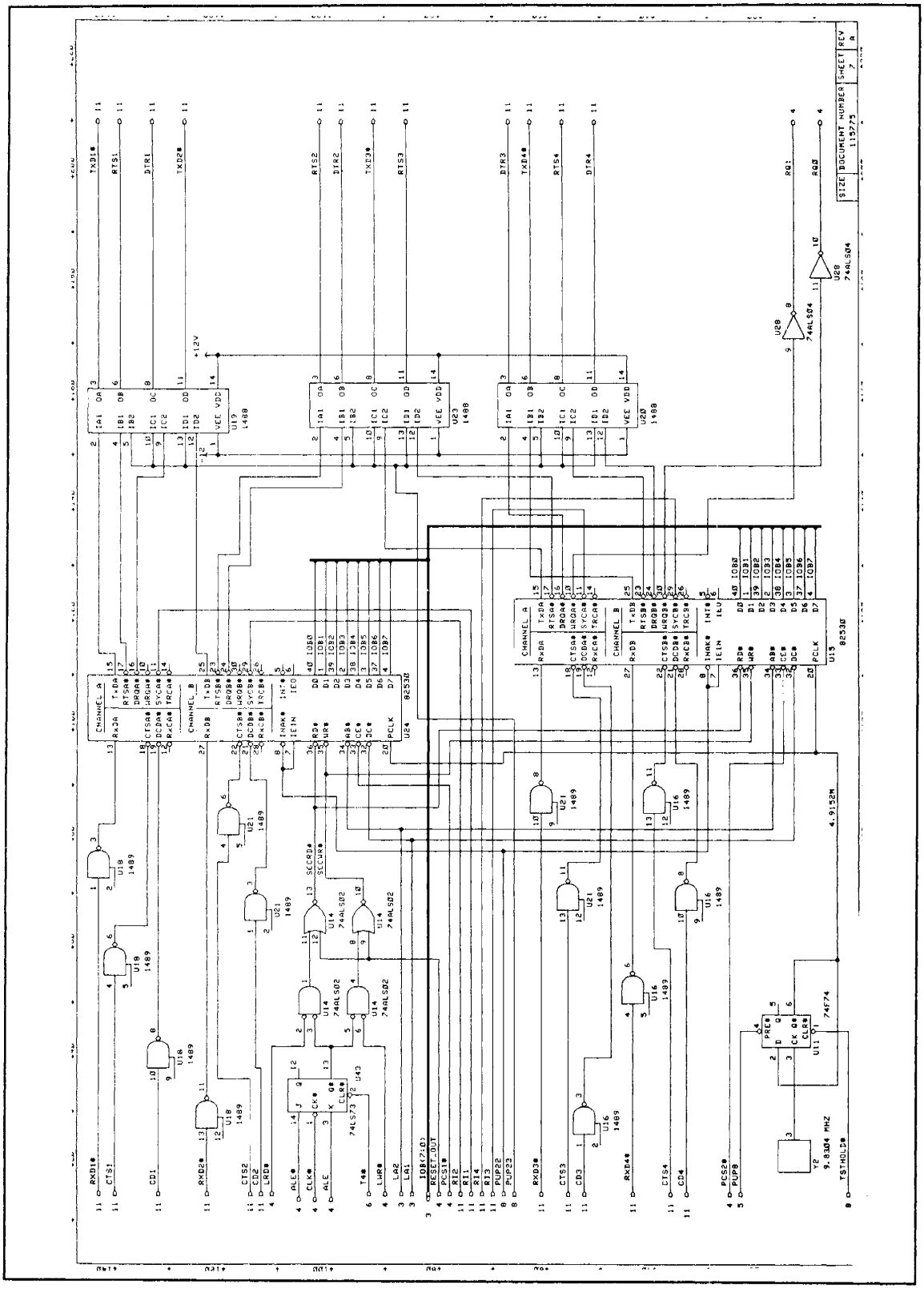


Figure 6-4 iSBC 546 Schematic Diagram (Sheet 7 of 11)

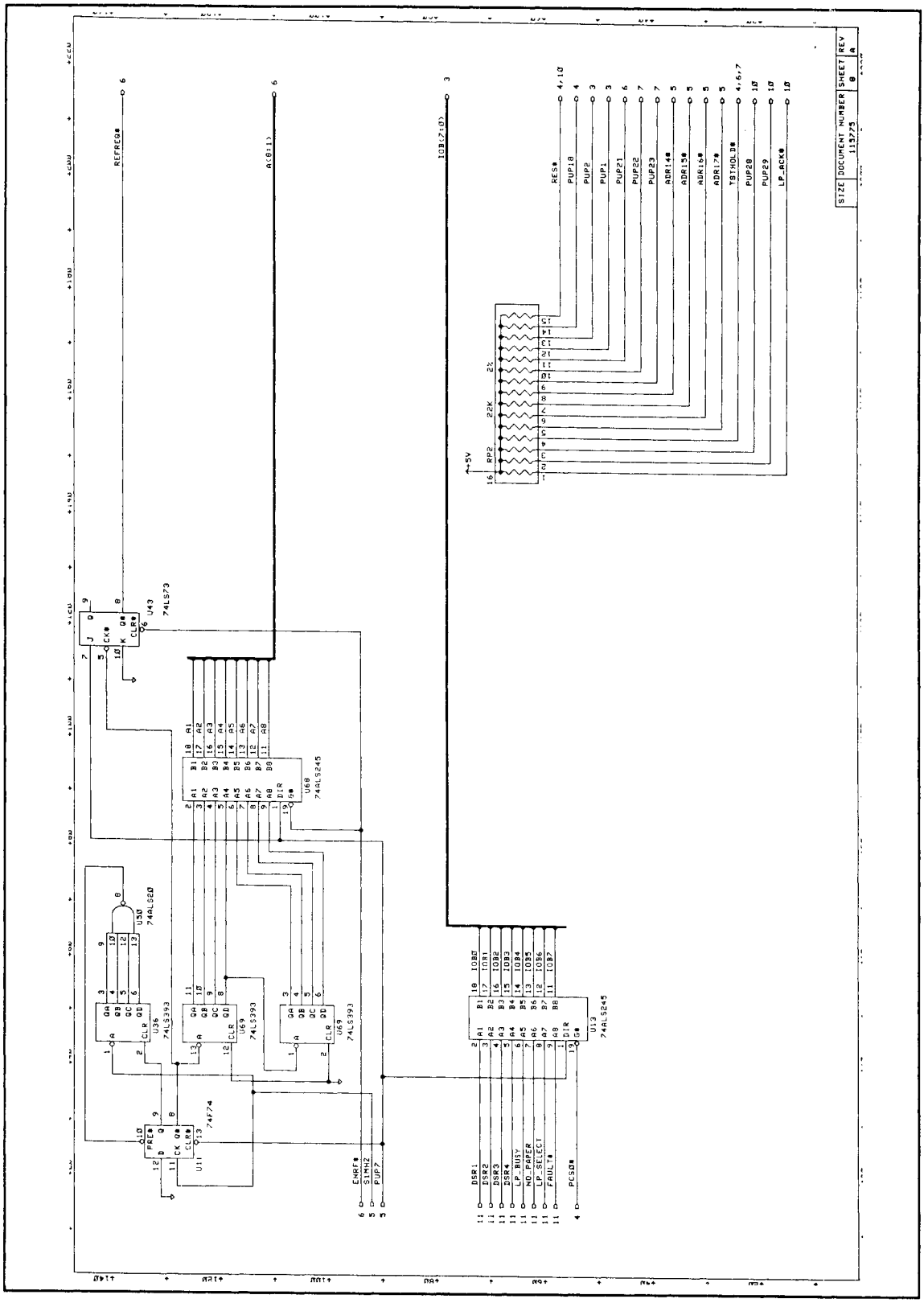


Figure 6-4 iSBC 546 Schematic Diagram (Sheet 8 of 11)

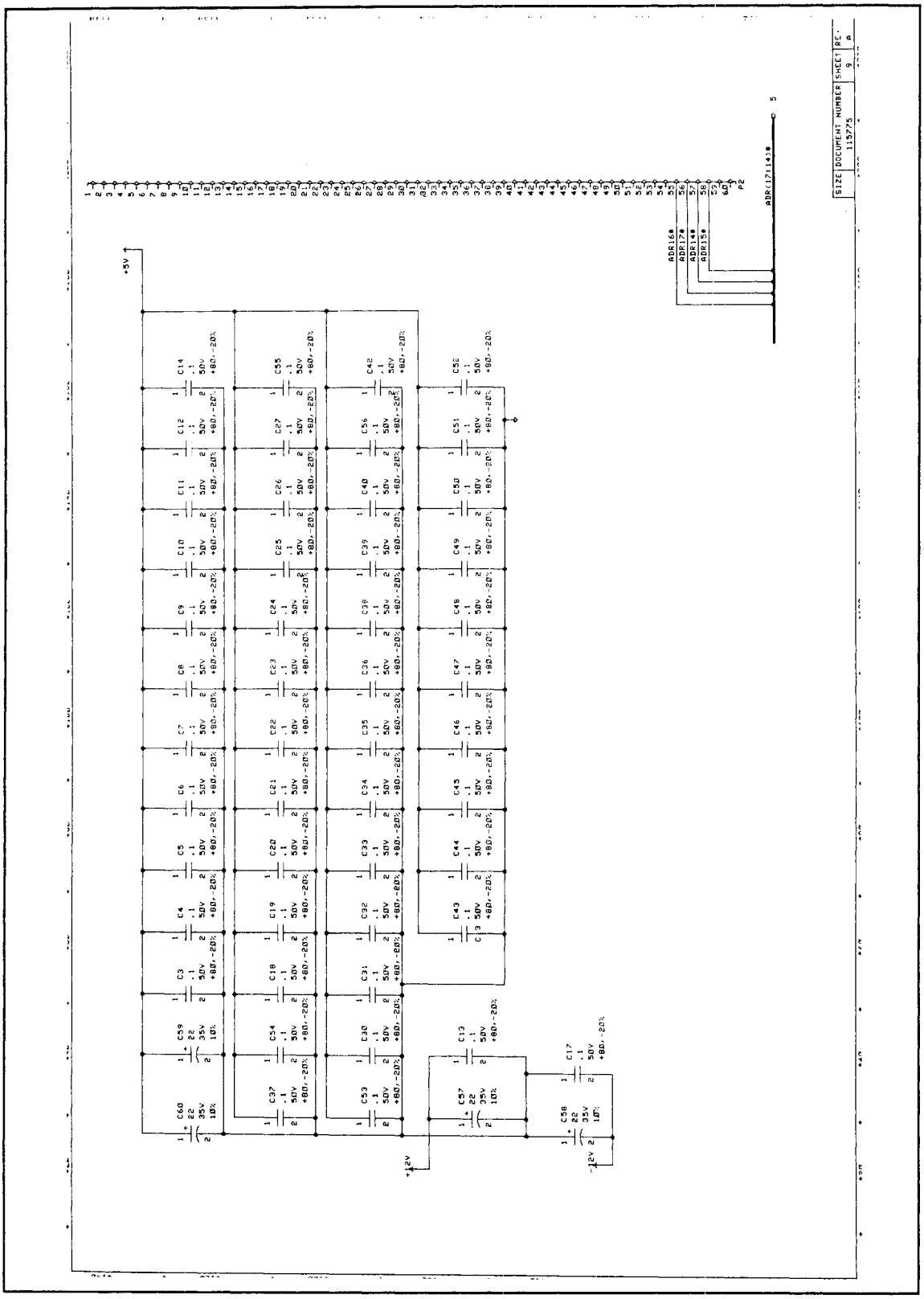


Figure 6-4 iSBC 546 Schematic Diagram (Sheet 9 of 11)

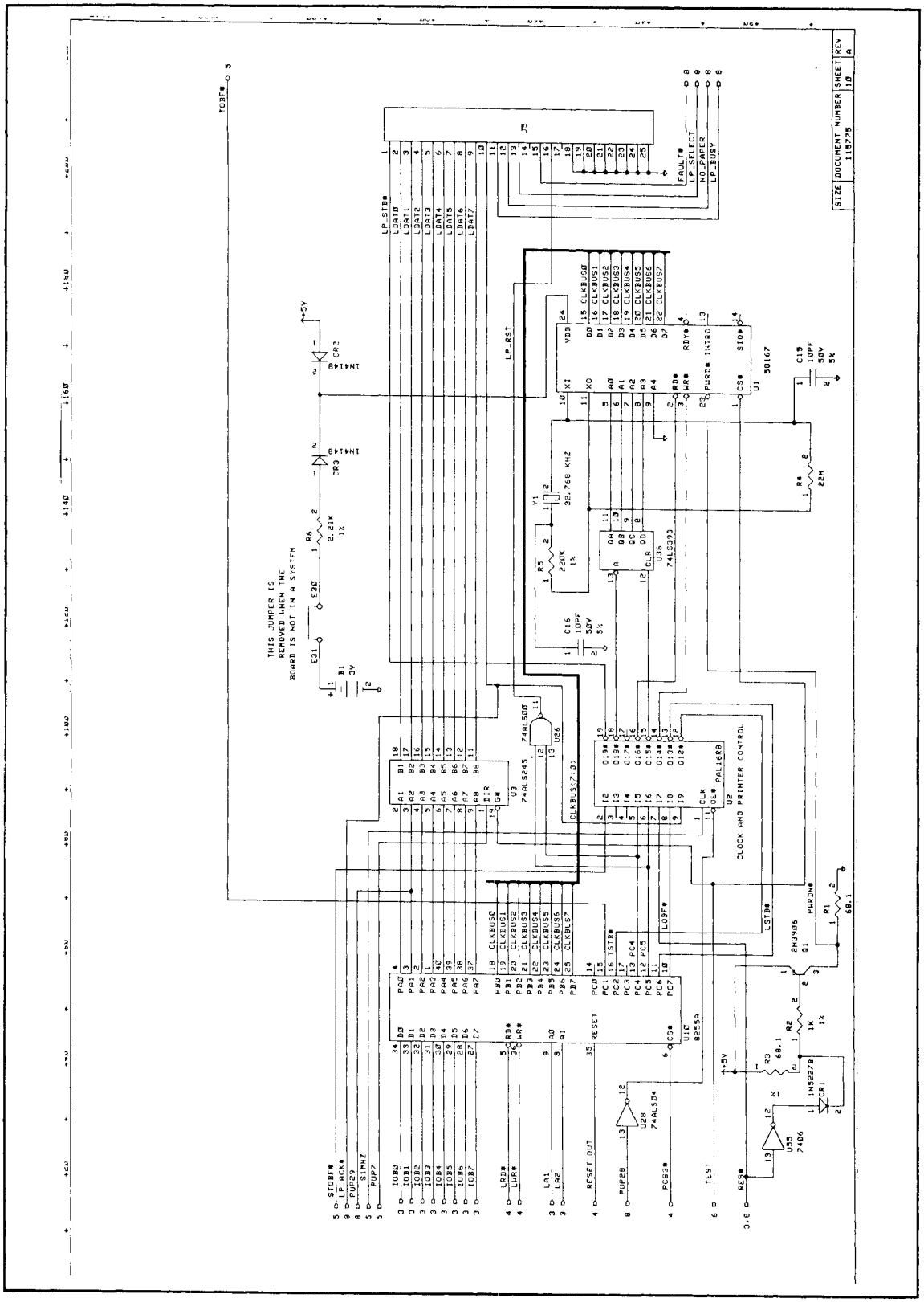


Figure 6-4 iSBC 546 Schematic Diagram (Sheet 10 of 11)

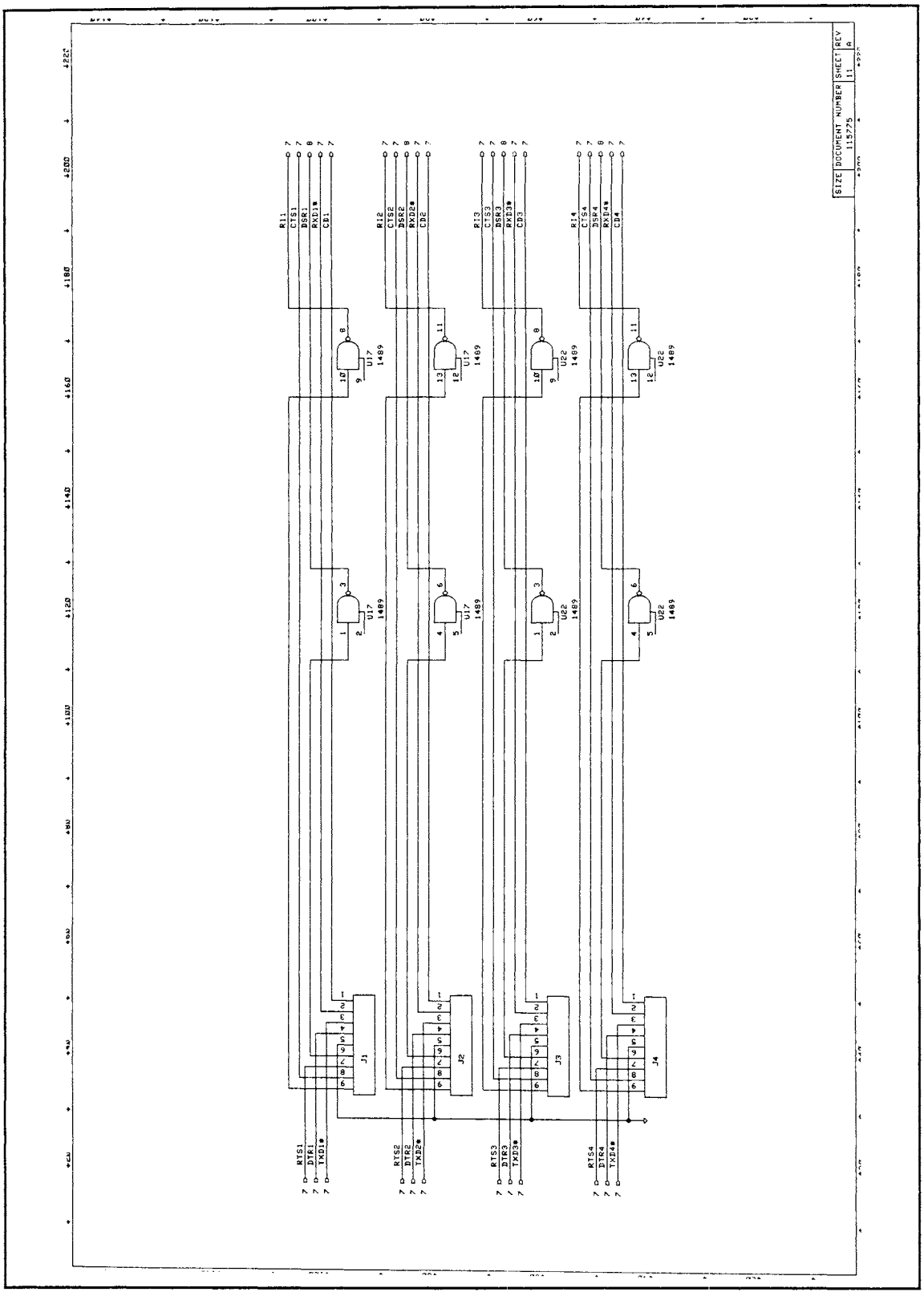


Figure 6-4 iSBC 546 Schematic Diagram (Sheet 11 of 11)

APPENDIX A
JUMPER INFORMATION

A.1 INTRODUCTION

This appendix provides jumper information for the three controller boards, iSBC 546, iSBC 547 and iSBC 548. The controller boards leave the factory in a specific configuration called the default configuration. Table A-1 lists all stake pin combinations (on which jumpers can be installed) for the iSBC 546 board. Table A-2 does the same for the iSBC 547 and iSBC 548 boards. A "Yes" in the Default column of Table A-1 or Table A-2 indicates the default jumpers installed by the factory. Sections A-3 through A-6 provide more detailed information about the jumpers. Figures A-1, A-2 and A-3 show the location of the stake pins on each of the boards

Table A-1. Jumper Combinations iSBC 546 Board

Jumper	Default	Function
E1 - E2	No	Flag Byte Address Jumper
E3 - E4	Yes	Dual Port RAM Address Jumper
E5 - E6	No	Flag Byte Address Jumper
E7 - E8	No	Dual Port RAM Address Jumper
E9 - E10	No	Flag Byte Address Jumper
E11 - E12	Yes	Dual Port RAM Address Jumper
E13 - E14	Yes	Dual Port RAM Address Jumper
E15 - E17	Yes	80186 Clockout Jumper (Removed only during factory test)
E19 - E24	No	Makes INT1* the MULTIBUS Interrupt when installed.
E25 - E24	Yes	Makes INT2* the MULTIBUS Interrupt when installed.

Table A-1. Jumper Combinations iSBC 546 Board (continued)

Jumper	Default	Function
E27 - E24	No	Makes INT3* the MULTIBUS Interrupt when installed.
E23 - E24	No	Makes INT4* the MULTIBUS Interrupt when when installed.
E28 - E29	No	Dual Port RAM Address Jumper, installed to select mapping in the lower MByte, not installed to select mapping in the upper MByte.
E30 - E31	No	Selects Battery Back-up for clock/calendar circuit.

Table A-2. Jumper Combinations iSBC 547/548 Boards

Jumper	Default	Function
E1 - E2	No	Flag Byte Address Jumper
E3 - E4	Yes	Dual Port RAM Address Jumper
E5 - E6	No	Flag Byte Address Jumper
E7 - E8	Yes	Dual Port RAM Address Jumper
E9 - E10	Yes	Flag Byte Address Jumper
E11 - E12	No	Dual Port RAM Address Jumper
E13 - E14	Yes	Dual Port RAM Address Jumper
E15 - E17	Yes	80186 Clockout Jumper (Removed only during factory test)
E18 - E21	No	Makes INT5* the MULTIBUS Interrupt when installed.
E19 - E24	No	Makes INT1* the MULTIBUS Interrupt when installed.
E20 - E21	No	Makes INT0* the MULTIBUS Interrupt when installed.
E22 - E21	No	Makes INT6* the MULTIBUS Interrupt when installed.
E23 - E24	No	Makes INT4* the MULTIBUS Interrupt when installed.
E25 - E24	No	Makes INT2* the MULTIBUS Interrupt when installed.

Table A-2. Jumper Combinations iSBC 547/548 Boards (continued)

Jumper	Default	Function
E26 - E21	No	Makes INT7* the MULTIBUS Interrupt when installed.
E27 - E24	Yes	Makes INT3* the MULTIBUS Interrupt when installed.
E28 - E29	No	Dual Port RAM Address Jumper, installed to select mapping in the lower MByte, not installed to select mapping in the upper MByte.

A-2 FLAG BYTE ADDRESS JUMPERS

I/O mapping of the flag byte is a jumper configurable option on the three controller boards. Table A-3 shows the jumpers and configurations available

Table A-3. Flag Byte Address Options and Jumpers

Flag Byte Addresses	Jumpers		
	E1 - E2	E5 - E6	E9 - E10
8A0(H)	X	X	X
8A1(H)	X	X	-
8A2(H)	X	-	X
8A3(H)	X	-	-
8A4(H)	-	X	X
8A5(H)	-	X	-
8A6(H)*	-	-	X
8A7(H)**	-	-	-

X = Jumper installed
\- = Jumper not installed
* = Default flag byte address for iSBC 547 and iSBC 548
** = Default flag byte address for iSBC 546

A.3 MULTIBUS INTERRUPT JUMPERS

The selection of which MULTIBUS Interrupt is used to interrupt the host is jumper selectable. A list of interrupts and there associated jumpers (the jumper installed selects its interrupt) is shown below:

Interrupt	Jumper
INT0*	E20 - E21 Selectable on iSBC 547/548 only
INT1*	E19 - E24 Selectable on all boards
INT2*	E25 - E24 Default installation iSBC 546 Selectable on all boards
INT3*	E27 - E24 Default installation iSBC 547 and iSBC 548 Selectable on all boards
INT4*	E23 - E24 Selectable on all boards
INT5*	E18 - E21 Selectable on iSBC 547/548 only
INT6*	E22 - E21 Selectable on iSBC 547/548 only
INT7*	E26 - E21 Selectable on iSBC 547/548 only

A.4 MEMORY MAPPING JUMPERS

Memory mapping of the DRAM is a jumper configurable option on all three controller boards. The jumper combinations and the addresses they select are shown in Table A-4. The jumpers and addresses are identical on all boards.

Table A-4. Memory Map Jumpers and Addresses

Addresses	Jumpers				
	E28 - E29	E3 - E4	E7 - E8	E11 - E12	E13 - E14
080000(H)	X	X	X	X	X
088000(H)	X	X	X	X	-
090000(H)	X	X	X	-	X
098000(H)	X	X	X	-	-
0A0000(H)	X	X	-	X	X
0A8000(H)	X	X	-	X	-
0B0000(H)	X	X	-	-	X
0B8000(H)	X	X	-	-	-
0C0000(H)	X	-	X	X	X
0C8000(H)	X	-	X	X	-
0D0000(H)	X	-	X	-	X
0D8000(H)	X	-	X	-	-
0E0000(H)	X	-	-	X	X
0E8000(H)	X	-	-	X	-
0F0000(H)	X	-	-	-	X
0F8000(H)	X	-	-	-	-
F80000(H)	-	X	X	X	X
F88000(H)	-	X	X	X	-
F90000(H)*	-	X	X	-	X
F98000(H)	-	X	X	-	-
FA0000(H)**	-	X	-	X	X
FA8000(H)	-	X	-	X	-
FB0000(H)	-	X	-	-	X
FB8000(H)	-	X	-	-	-
FC0000(H)	-	-	X	X	X
FC8000(H)	-	-	X	X	-
FD0000(H)	-	-	X	-	X
FD8000(H)	-	-	X	-	-
FE0000(H)	-	-	-	X	X
FE8000(H)	-	-	-	X	-
FF0000(H)	-	-	-	-	X
FF8000(H)	-	-	-	-	-

* Default address for the iSBC 547 and iSBC 548 boards.
** Default address for the iSBC 546 board.
X = Jumper installed.
- = Jumper not installed.

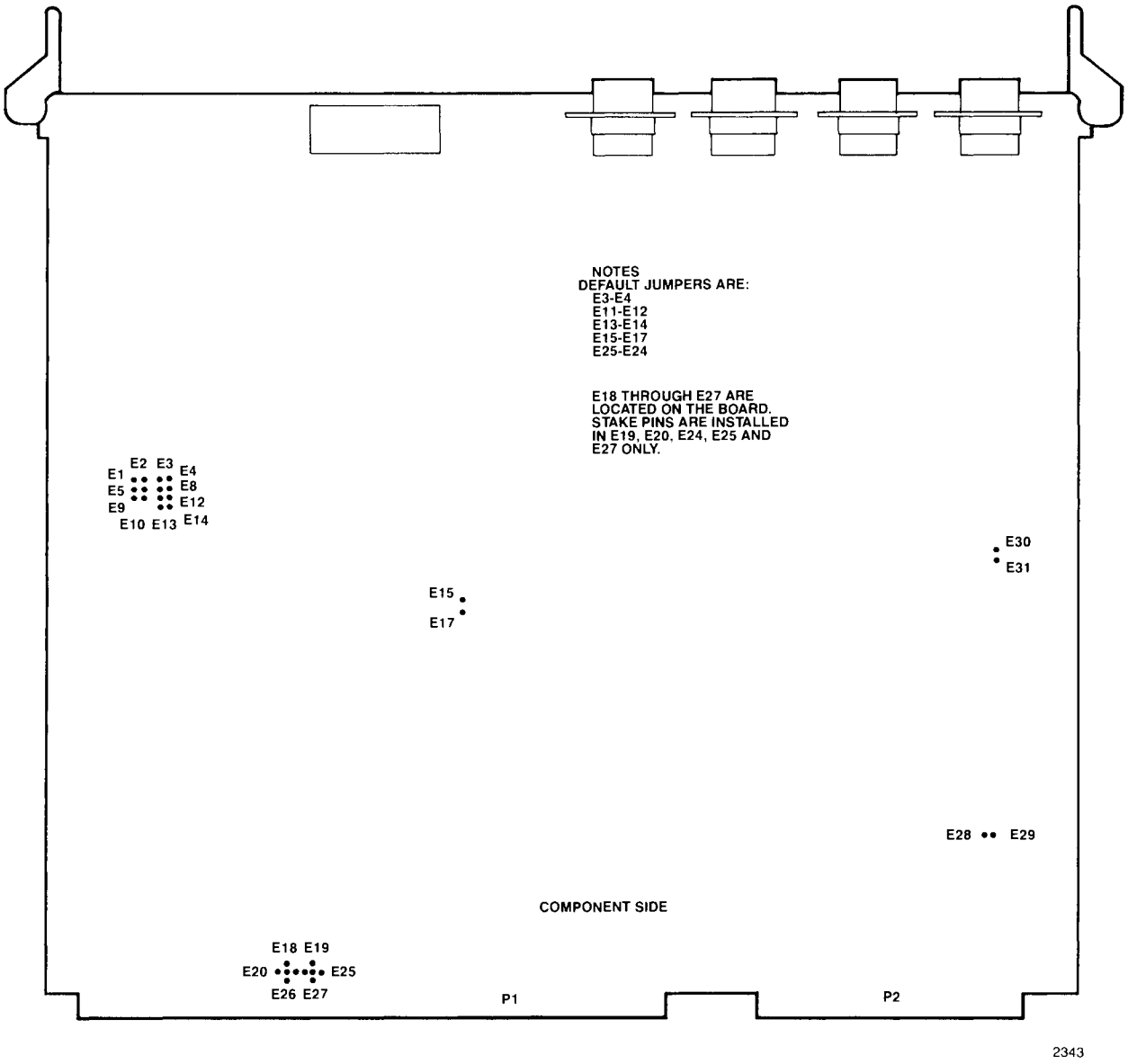


Figure A-1. iSBC 546 Board Jumper Location

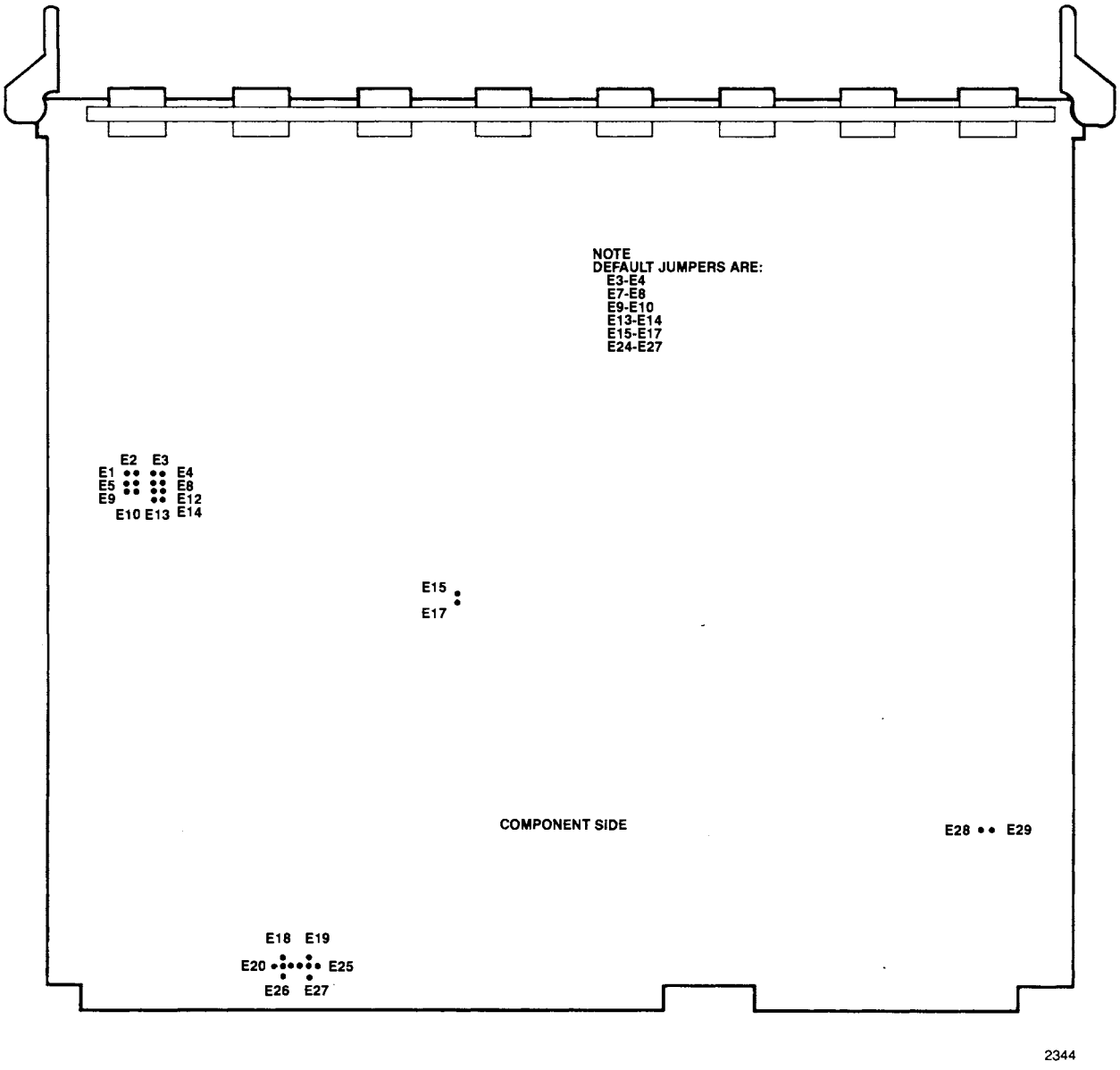


Figure A-2. iSBC 547 Board Jumper Location

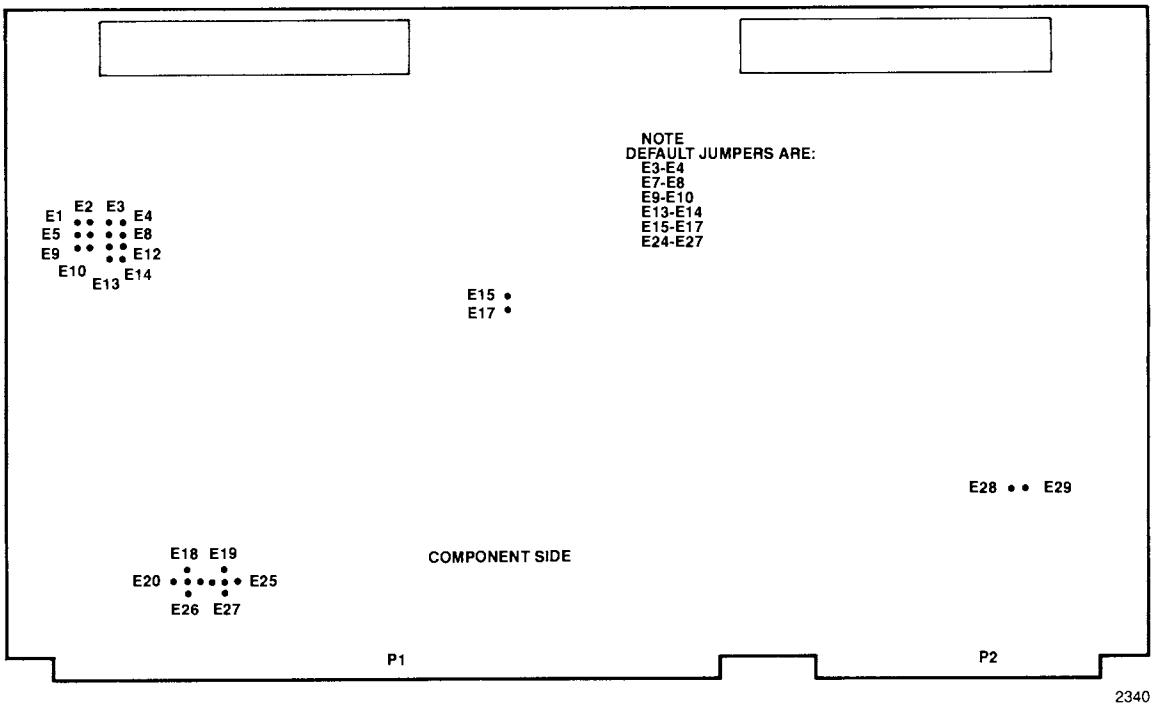


Figure A-3. iSBC 548 Board Jumper Location

APPENDIX B
FIRMWARE

B.1 INTRODUCTION

This appendix describes the user commands for the communication firmware supplied with the iSBC 546, iSBC 547 and iSBC 548 boards. The firmware makes the boards into terminal controllers. The primary features of the firmware are listed and briefly summarized in Table B-1.

B.2 FIRMWARE OVERVIEW

NOTE
Throughout this appendix the word controllers indicates all three boards, iSBC 546, iSBC 547 and iSBC 548. References to individual boards will include the model number.

The iSBC 546/547/548 firmware is released as a set of two Intel EPROMs. The firmware makes the three boards into intelligent terminal controllers which can provide a MULTIBUS host CPU with either four (iSBC 546) or eight (iSBC 547 and iSBC 548) asynchronous serial channels.

MULTIBUS host CPUs view the iSBC 546/547/548 boards as slave peripheral controllers. The host and the controllers communicate via shared data structures and a message passing scheme implemented in the controllers on-board dual ported RAM. The host CPU signals the controller with the hardware I/O mapped flag byte mechanism on the controllers. The controllers signal the host CPU by requesting an interrupt on a jumper selectable MULTIBUS interrupt line.

Table B-1. iSBC 546/547/548 Firmware Features

Feature	Description
Asynchronous Serial Channel Support	The firmware supports the serial channels in asynchronous mode Parameters such as baud rate, parity generation, parity checking and character length can be programmed independently for each channel.
Block Data Transfer	The firmware relieves the MULTIBUS host CPU of one character at a time interrupt processing. The board accepts blocks of data for transmission and interrupts the processor only when the entire block is transmitted.
Modem Control	The firmware provides software control of the Data Terminal Ready (DTR) line on all channels. Transitions on the Carrier Detect (CD) line are sensed and reported to the host CPU. Request to Send (RTS) is continuously asserted. In the default mode the transmitter and receiver are enabled independently of the state of the Clear to Send (CTS) and CD modem signals respectively. A special command from the host instructs the controller boards to make CTS and CD gating signals for transmission and reception respectively. RI and DSR signal transitions are reported to the host if the host so instructs, otherwise they are not reported.
Tandem Mode Support	The firmware provides a flow control facility to synchronize a remote source that may be transmitting so fast that the

Table B-1. iSBC 546/547/548 Firmware Features (continued)

Feature	Description
	controller (iSBC 546/547/548) may exhaust its receive buffer space for that channel. The controller transmits an XOFF character when the number of characters in its receive buffer exceeds a threshold value and transmits an XON character when the buffer drains below its threshold.
Tandem Response Mode	In this mode the controllers will suspend all transmissions to a line if an XOFF was received from this line, and will resume only upon receipt of XON.
Automatic Baud Rate Recognition	The firmware provides a capability to detect the baud rate of an agent connected to a serial channel. The remote agent must transmit a maximum of four ASCII "U" CHARACTERS. The detected baud rate must be 19200, 9600, 4800, 2400, 1200, 600, 300, or 150.
Download and Execute Capability	The firmware provides a capability for the host CPU to load code anywhere in the lowest 128K Byte space of the controller (except in the DYNAMIC STRUCTURE and QUEUE areas) and for the controller to start (with the exception of code) at any address in this address space.
Power Up Confidence	The firmware executes a sequence of simple tests to establish that crucial components on the boards are functional.

B.2.1 FIRMWARE OPERATION

For the MULTIBUS host to input commands to a controller board in it's system it must do the following:

> NOTE
> The commands, messages queues and procedures discussed in this section are described in detail in Section B.3

1. Load the commands into the IN-QUEUE, starting at the first location in the queue or in the location immediately after the last location used.

2. Update the IN-QUEUE TAIL in the Dynamic Structure to show the current number of commands in the queue.

3. Send a flag interrupt (write 2 to the I/O address) to the controller board.

When the controller receives the interrupt it scans the commands in the IN-QUEUE and executes them. The controller then updates the IN-QUEUE-HEAD to indicate the number of commands it read and executed.

If several controller boards are sharing the same I/O address they all detect the same interrupt. When the IN-QUEUE is scanned,the controllers that find no new commands ignore the interrupt and return to their states before the interrupt.

When a controller board sends a message to the host it writes the message into the OUT-QUEUE and updates the OUT-QUEUE TAIL to indicate the number of messages in the OUT-QUEUE. After the updating the controller sends an interrupt to the host. When the host receives the interrupt it scans the OUT-QUEUE of all controller boards in the system sharing the interrupt line, and reads the available messages. The host then updates the OUT-QUEUE-HEAD to indicate to the controller that it has read the messages. The host then resets the interrupt line by writing a 4 to the I/O address of the controller board. All controller boards sharing the same I/O address reset their interrupts together. Controller boards should share the same I/O addresses only if the share the same interrupt lines.

B.2.2 RECOMMENDATIONS FOR HIGH PERFORMANCE

To maximize controller board performance the following factors should be considered:

1. Lines that are not used should be disabled.

2. Use of special options (TANDEM Mode, SPECIAL CHAR Mode and AUTO BAUD Mode, until the baud rate is found) slow board performance greatly. These options should not be used unless necessary.

3. The host should not clear it's buffer immediately after receipt of the INPUT AVAILABLE message. By not clearing the buffer immediately the number of interrupts from the controller board will be reduced. Both host and controller performance will be improved. No new INPUT AVAILABLE message for this line will be received until the CLEAR BUFFER command is sent. The data will be received by the controller but will not be reported until the CLEAR BUFFER command is received.

4. The CD line on the serial input should not be allowed to float. If the line is allowed to float false reports of CD DETECT and CD LOST will occur.

B.3 FUNCTIONAL ARCHITECTURE

A host CPU communicates with the controller boards via a shared data structures and a simple message passing scheme implemented in the dual port RAM on the controller boards. Inter-processor signalling is accomplished by using the hardware I/O mapped wakeup byte on the controllers and requesting an interrupt to the host on a MULTIBUS interrupt line.

Section B.3.1 describes the structures in dual po ted memory. A description of the messages exchanged by the host CPU and the controllers follows in Section B.3.2. Section B.3.3 details the implementation of the message passing scheme. Section B.3.4 details the power-up confidence tests.

B.3.1 STRUCTURES OF DUAL PORTED RAM

Sections B.3.1.1 through B.3.1.6 describe the layout of the data structures in dual port memory. The addresses are given in decimal notation and are relative to the start address which is mapped to the MULTIBUS. To the controllers 18000H is the start address.

Figure B-1 shows the memory layout used.

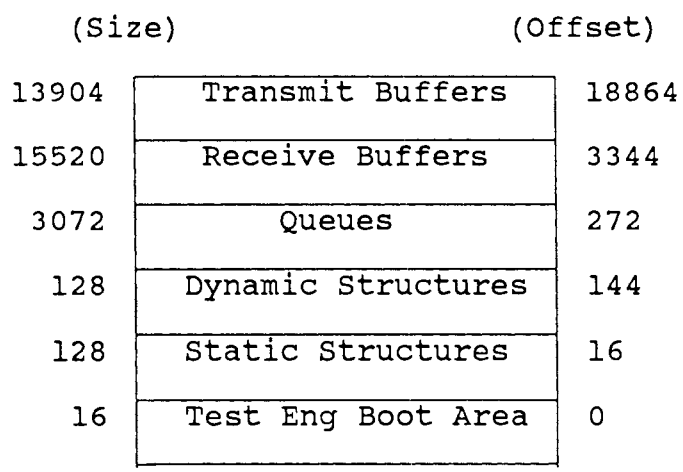


Figure B-1. Layout of Shared (Dual Port) Memory

B.3.1.1 Test Engineering Boot Area

This area provides an interface for test programs to run on the board bypassing all normal firmware initialization. On rest, the firmware waits for a minimum of 250 ms for the 12 byte ASCII pattern RIGHTNOWGOTO to be loaded into the first 12 bytes of this area. If the pattern is loaded within 250 ms of reset the firmware executes a far jump to the address specified by a 32-bit 8086 style pointer (16-bit offset plus 16-bit selector) in the next four bytes. If the pattern is not loaded within the 250 ms the firmware continues with its normal initialization. Figure B-2 shows the layout of the Test Engineering Boot Area.

Before the 250 ms wait, the firmware performs no initialization other than setting the internal I/O in the on-board 80186.

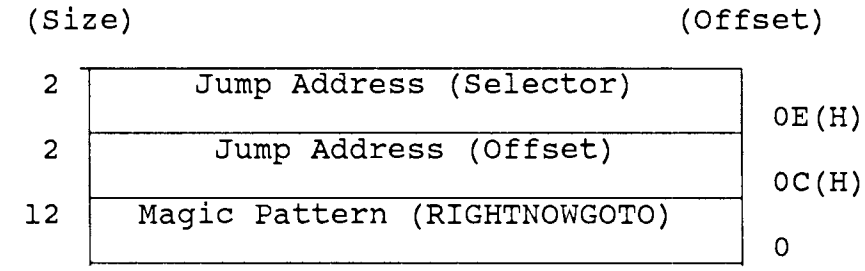


Figure B-2. Test Engineering Boot Area Layout

B.3.1.2 Static Structures

Figure B-3 details the Static Structures area. This area is set by the firmware and must only be read, not modified by the host CPU. After completing its initialization sequence on reset, the firmware sets the following values in this area:

Board Type

This value set to 02H indicates an iSBC 547 or 548 board. This value set to 03H indicates an iSBC 548 board. To use an iSBC 188/48 driver with this firmware requires that it be modified to recognize the new board types (iSBC 546/547/548.)

Version

This value indicates the version of the firmware. The version Vxy is represented by the value (x * 16) (x * 16) + y

Completion Flag

This flag is set to 0FFH when the initialization is completed. This will occur within 10 milliseconds if the board is functional.

Confidence Test Result (Read by the Host)

This value is set to 0FFH if all confidence test succeed during initialization. Otherwise the value indicates the test that failed.

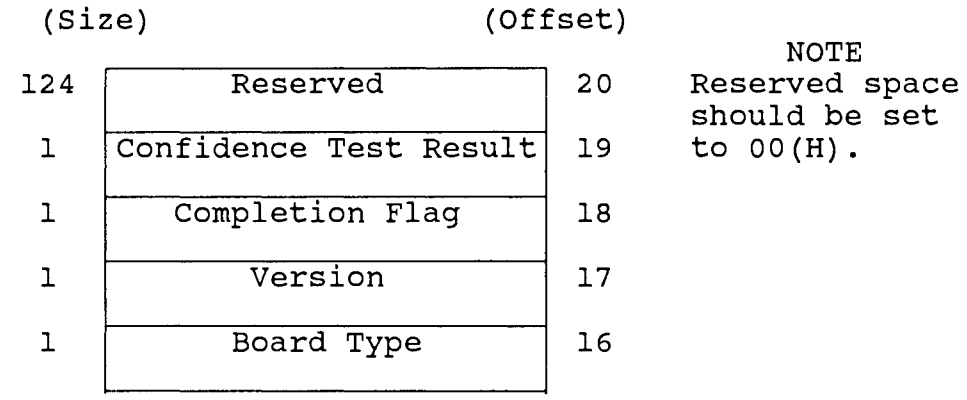


Figure B-3. Static Structure Area Layout

B.3.1.3 Dynamic Structures

The message passing scheme utilized for inter-processor communication is implemented as two circular queues in shared memory. One queue (the OUT queue) is used for messages going from the controllers to the host CPU. The other queue (the IN queue) is for messages going from the host CPU to the controllers. The Dynamic Structures area contains the variables that control the queueing mechanism. Figure B-4 details the layout of the Dynamic Structures area, the semantics are described in Section B.3.3.

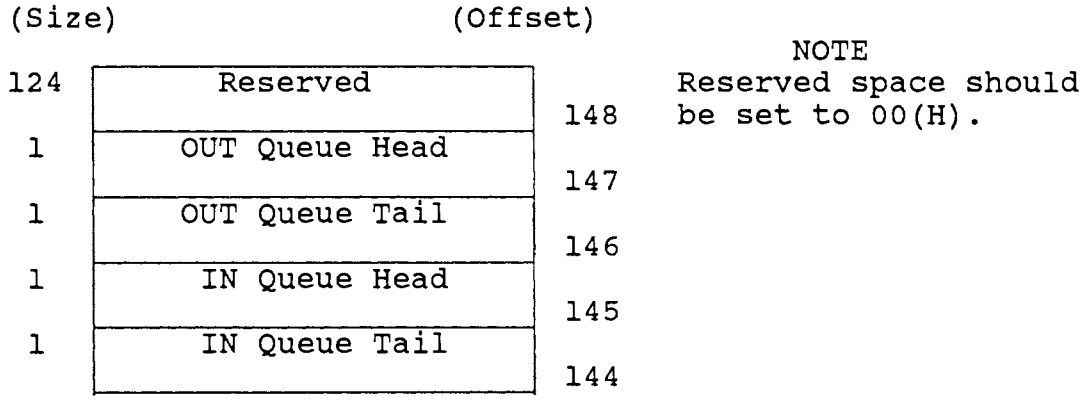


Figure B-4. Dynamic Structure Layout

B.3.1.4 Queue

The Queue area contains the actual contents of the inter-processor message passing queues. It is divided into two equal regions as shown in Figure B.5. One region is the IN queue the other is the OUT queue. Section B.3.3 presents more detailed information on the Queue area.

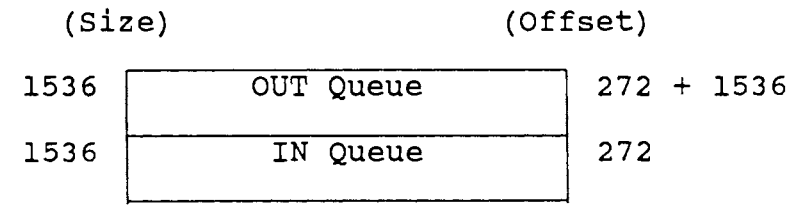


Figure B-5. Layout of Queue Area

B.3.1.5 Receive Buffers

The Receive Buffers area is divided into eight receive buffers for the eight serial channels supported. Each buffer is 1940 bytes long. The buffer for line i starts at offset 3344 + (i X 1940) for $0 \leq i \leq 7$.

B.3.1.6 Transmit Buffers

The Transmit Buffers area is a relatively unstructured area that may be used by the host CPU for allocating transmit buffers or for any other purpose. This area is managed by the host CPU and is not modified by the controllers. The Transmit buffer starts at an offset from the beginning of dual-ported memory 18864.

B.3.2 INTER-PROCESSOR MESSAGES

This section describes the message formats and protocol used for communication between the host CPU and the controllers. Section B.3.2.1 describes the messages sent by the host CPU to the controllers. Section B.3.2.2 describes the messages sent by the controllers to the host CPU. Section B.3.3 describes the implementation of the message passing scheme.

All messages have a fixed length of 16 bytes. Several messages have fields labelled "Reserved". It is recommended that these fields be set to zero for compatibility with future products.

The messages described, later identify serial channels by "line numbers". Line numbers 1 through 8 correspond to serial channels 1 through 8.

B.3.2.1 Host CPU to Controller Messages

These sections (B.3.2.1 through B.3.2.21) describes messages used by user level software running on the host CPU to communicate with the controller.

Host CPU to iSBC Controller Messages

B.3.2.1.1 INITIALIZE. This message is used by the host CPU to initialize the controllers.

This message must be the first message sent to the board after reset unless a download and execute function is to be performed,in which case the Download command must be the first message. If the Initialize message is not the first message after a reset, it is ignored.

After the Initialize message is processed by the controller all lines are disabled. Each line has to be independently enabled with an Enable Line command before it can be used. The only line specific commands that can be directed to a disabled line are Configure Line and Enable Line.

The controllers return an Initialize Complete message containing a bit map of the lines determined to be valid. This number will be 6 for the iSBC 546 board or 8 for the iSBC 547 and iSBC 548 boards. Subsequent line specific commands must be directed to those lines noted as valid. The message format is shown in Figure B-6.

Message Format

0	01H
1	Reserved
2	Reserved
3	Reserved
4	Reserved
	:
15	Reserved

NOTE
Reserved space should be set to 00(H).

Response — An Initialization Complete command is returned. No indication of MULTIMODULE present is returned to the driver.

Figure B-6. Initialize Message Format

Host CPU to iSBC Controller Messages

B.3.2.1.2 ENABLE. This command is used by the host CPU to enable a serial channel. Once enabled, other commands can be directed to the channel.

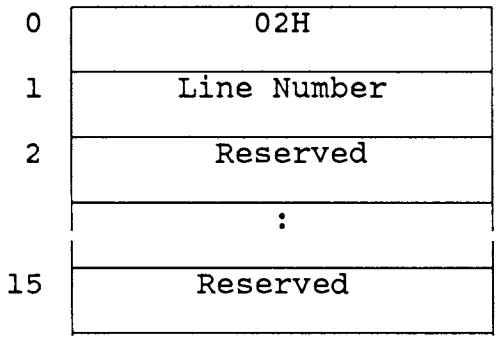
The firmware enables the serial channel's receiver and transmitter only on this command. The RTS and DTR modem control signals are asserted and cleared respectively.

An Enable Line command, received when the line is already enabled is ignored by the controllers.

An Enable Command to the line printer causes a reset pulse to be issued to the line printer.

The Enable message format is shown in Figure B-7.

Message Format



NOTE
Reserved space should be set to 00(H).

Line Number The serial channel being enabled

Response None

Figure B-7. Enable Message Format

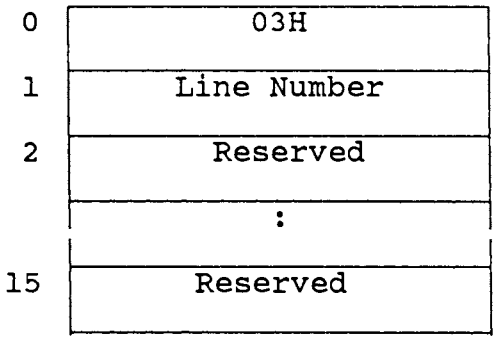
Host CPU to iSBC Controller Messages

B.3.2.1.3 DISABLE. This command is used to disable a serial channel. In the disabled state, the serial channel's receiver and transmitter are disabled and the DTR and RTS modem control lines are cleared.

It is recommended that this command be used when the line is quiescent, as it clears the state of the channel, with any pending output being cancelled and any received characters discarded. Further, if a pending transmit operation is cancelled in this process no Transmit Complete message is returned.

The Disable message format is shown in Figure B-8.

Message Format



NOTE
Reserved space should be set to 00(H).

Line Number The serial channel being enabled

Response None

Figure B-8. Disable Message Format

Host CPU to iSBC Controller Messages

B.3.2.1.4 CONFIGURE. This command is used to set the parameters of a serial channel. It is recommended that this command be used when the line is in the quiescent state, as the command clears the channel, clearing any pending output and discarding any received characters. Further, if a pending transmit operation is cancelled in this process no Transmit Complete message is returned. This command is accepted when the line is disabled.

The Configure message format is shown in Figure B-9.

Message Format

0	04H
1	Line Number
2	Baud Rate
3	
4	Link Parameters
5	Line Discipline
6	Mode
7	Tandem High Water Mark
8	Tandem Low Water Mark
9	Signal Spl Char High Water Mark
10	XON Character
11	XOFF Character
12	Special Character
13	Special Character
14	Special Character
15	Special Character

Figure B-9. Configure Message Format

A description of the format blocks shown in Figure B-9 is provided below:

Line Number — The number of the serial channels being configured

Baud Rate — The desired baud rate (both transmit and receive). The actual baud rate obtained can be computed using the following:

Actual Baud Rate = 153,600/(count + 2)
where count = trunc(153600/Requested Baud Rate) -2

The highest baud rate that can be specified is therefore 76,800.

A baud rate of zero has special significance. It instructs the controller to place the line in an automatic baud rate recognition mode. In this mode the controller attempts to sense the baud rate of an agent connected to the serial channel. The remote agent is required to transmit a maximum of four ASCII "U" characters before its baud rate is determined. The remote agent must be set to transmit at one of the following standard baud rates - 19200, 9600, 4800, 2400, 1200, 600, 300, or 150.

Once the baud rate is sensed an Autobaud Complete message is returned to the host.

Link Parameters — The parameters to be used on the physical link:

Bit 1-0: Parity
- 00 - No parity
- 10 - Even parity
- 11 - Odd parity

Bit 3-2 Character length
- 00 - 6 bits/character
- 10 - 7 bits/character
- 11 - 8 bits/character

Bit 5-4 Number of Stop Bits
- 00 - 1 Stop Bit
- 01 - 1 1/2 Stop Bits
- 10 - 2 Stop bits

Bit 7-6 Reserved

If parity is enabled, an additional bit position , beyond those specified in the Character Length control is added to the transmitted data and expected in received data. The received parity bit is transferred to the CPU as part of the data unless 8 bits/character is selected. If a parity error is detected on input, the character is discarded.

In the 6 and 7 bits/character modes unused bit positions in transmit data are ignored. Unused bits in receive data are set to 1. If a framing error is detected on input, the character is returned as an 8-bit null (00H).

Line Discipline

This block is assigned for future firmware implementations which support more complex functions. This block is set to 01H for this application.

Mode

Used to set special modes:

Bit 0: Tandem Mode Enable
0 - Tandem Mode Off
1 - Tandem Mode On

Bit 1: Signal Special Character Mode Enable
0 - Signal Special Character Mode Off
1 - Signal Special Character Mode On

Bit 2: Tandem Response Mode
0 - Tandem Response Mode Off
1 - Tandem Response Mode On

Bits 3 -7: Reserved

Tandem Mode provides a mechanism for the controllers to throttle a remote transmitter that could potentially cause the controller to run out of receive buffer space. On receiving a character, if the number of characters in the receive buffer is greater than or equal to the Tandem High Water Mark an XOFF character is immediately transmitted on the

same channel, if an XOFF was transmitted. When the receive buffer drains to a value equal to the Tandem Low Water Mark an XON character is transmitted to allow the remote source to continue transmitting.

When in the Tandem Response Mode the controllers will suspend transmission to a line if an XOFF signal was received from that line. Transmission will resume upon receipt of XON from that line.

The Signal on Special Character mode facilitates the the expeditious handling of interrupt characters. A common problem with buffered terminal controllers is that when there is substantial type ahead, interrupt characters are buffered with the data on the controller. Consequently, the host does not see the interrupt character until all the data characters preceding it have been copied out of the controller.

The Signal on Character mode provides a solution to this problem. If a special character is received and there are more than some specified number of characters in the receive buffer a Special Character Received message is sent to the host. The character is then stored in the receive buffer to mark the position of the interrupt in the input stream. The set of (up to four) special characters is user specified. The comparison of a received character to the characters making up this set is restricted to the data portion defined by the specified character length.

Tandem High Water Mark — The high water mark used for Tandem mode is eight times this value.

Tandem Low Water Mark — The low water mark used for Tandem mode is eight times this value.

Signal Special Character High Water Mark — The high water mark used in the Signal Special Character mode is eight times this value.

XON Character	The XON character used in Tandem mode.
XOFF Character	The XOFF character used in Tandem mode.
Special Character	The characters forming the special character set in the Signal on Special Character mode.
Defaults	Line parameters are set to the following defaults on reset: 9600 baud 7 bits/character 1 stop bit Even parity Tandem mode OFF Signal on special Character mode OFF Tandem Response mode OFF Note that all parameters must be specified any time Configure message is used.
Response	An Autobaud Complete message is returned if automatic baud recognition is requested.

Host CPU to iSBC Controller Messages

B.3.2.1.5 TRANSMIT BUFFER. This message is used to initiate the transmission of a sequence of characters on a serial channel.

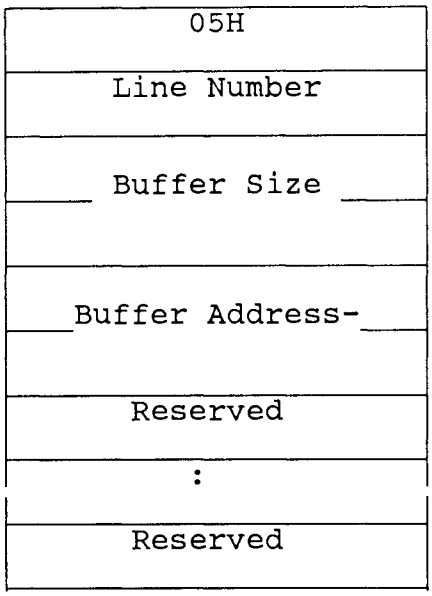
After the entire transmission completes, a Transmit Complete message is returned to the host. A transmit command is ignored if the line is transmitting a break or has not been enabled or has been placed in an automatic baud rate recognition mode by a previously issued Configure command.

A transmission once initiated can be suspended with a Suspend Transmit message and aborted with an Abort Transmit message.

A transmission to the clock/calendar line must be 11 data bytes as described in section B.3.2.5.

Figure B-10 shows the Transmit message.

Message Format



NOTE
Reserved space should be set to 00(H).

Line Number — The serial channel on which transmission is initiated.

Buffer Size — The size of the transmit buffer in bytes.

Buffer Add. — The 16-bit offset + 16384 (for product compatibility) at the beginning of the transmit buffer in the dual ported address space of the controllers. The host CPU fills the transmit buffer by writing into it directly prior to issuing this command. In the 6 and 7 bits/character formats the data must appear in the least significant positions of an 8-bit byte. Unused bit positions are ignored.

Response — A Transmit Complete message is returned after the ENTIRE sequence of bytes contained in the transmit buffer has been transmitted.

Figure B-10. Transmit Buffer Message Format

Host CPU to iSBC Controller Messages

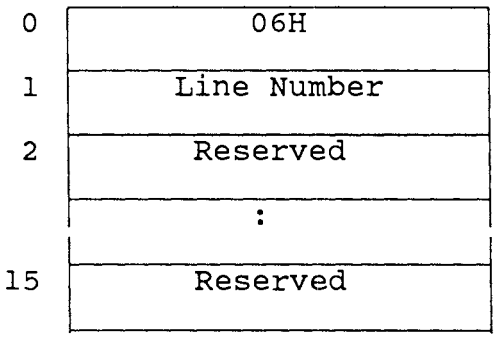
B.3.2.1.6 ABORT TRANSMIT. This message is used to abort a transmission already in progress.

This function is expected to be useful for operating system drivers to implement the purge transmit buffer operation commonly requested with an "O" character.

If a transmission is not in progress on the line, the command is ignored. Otherwise, a Transmit Complete message is returned. If there are multiple outstanding transmit requests for the line, only the current (the one issued the earliest) is aborted.

Figure B-11 shows the Abort Transmit message format.

Message Format



NOTE
Reserved space should be set to 00(H).

Line Number — The serial channel on which transmission is aborted.

Response — A Transmit Complete message is returned if a transmission was indeed aborted.

Figure B-11. Abort Transmit Message Format

Host CPU to iSBC Controller Messages

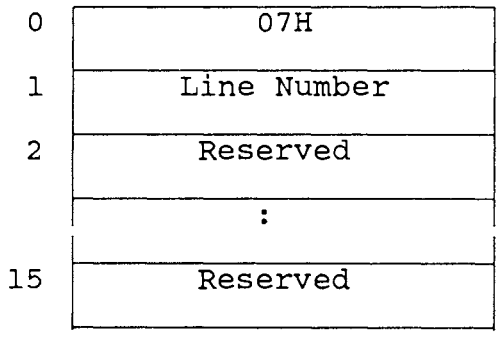
B.3.2.1.7 SUSPEND TRANSMIT. This command is used to suspend (rather than abort) a transmission on the line.

If there is no transmission in progress on the particular serial channel, the command is ignored. If the transmission is already suspended, the command is again ignored. A suspended transmission can be resumed with a subsequent Resume Transmit command.

If there are multiple outstanding transmit requests for the line the line will remain suspended until a Resume Transmit command is issued.

Figure B-12 shows the Suspend Transmit message format.

Message Format



NOTE
Reserved space should be set to 00(H).

Line Number — The serial channel on which transmission is suspended.

Response — None.

Figure B-12. Suspend Transmit Message Format

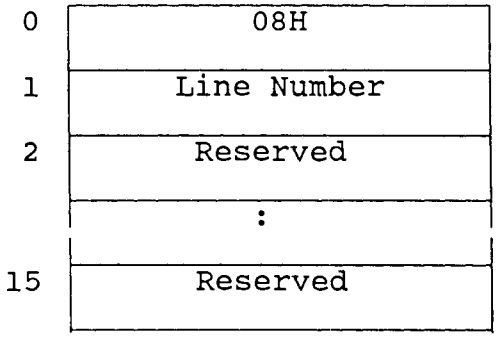
Host CPU to iSBC Controller Messages

B.3.2.1.8 RESUME TRANSMIT. This command is used to resume a previously suspended transmission.

If there is no transmission in progress on the line or if the transmission is not suspended, the command is ignored.

Figure B-13 shows the Resume Transmit message format.

Message Format



NOTE
Reserved space should be set to 00(H).

Line Number — The serial channel on which transmission is resumed.

Response — None.

Figure B-13. Resume Transmit Message Format

Host CPU to iSBC Controller Messages

B.3.2.1.9 ASSERT DTR. This command is used to assert the Data Terminal Ready (DTR) modem signal on a serial channel.

If the Carrier Detect modem signal is asserted for the line when this command is received, the controllers return a Carrier Detect message even though an OFF to ON transition was not sensed on the Carrier Detect signal. Thus, the MULTIBUS host can maintain a state variable following the Carrier Detect modem signal by toggling the variable when subsequent Carrier Detect and Carrier Loss messages are received.

Figure B-14 shows the Assert DTR message format.

Message Format

0	09H
1	Line Number
2	Reserved
	:
15	Reserved

NOTE
Reserved space should be set to 00(H).

Line Number — The serial channel on which DTR is asserted.

Response — A Carrier Detect message is returned if the Carrier Detect is asserted when the command is received.

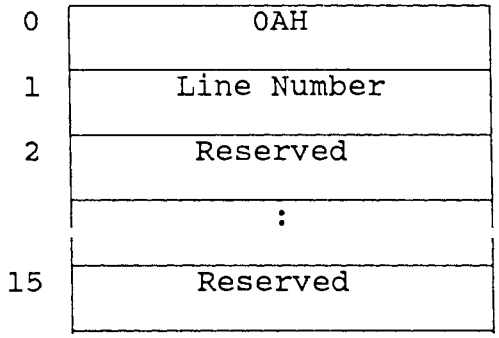
Figure B-14. Assert DTR Message Format

Host CPU to iSBC Controller Messages

B.3.2.1.10 SET CTS AND CD GATES. This command causes the controller's specified line not to transmit unless CTS is active and not to receive unless CD is active.

Figure B-15 shows the command format.

Message Format



NOTE
Reserved space should be set to 00(H).

Line Number — The serial channel on which the CTS and CD gates are set.

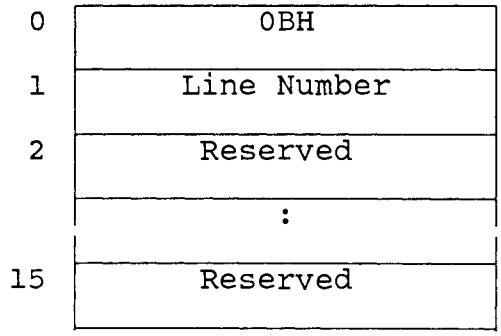
Figure B-15. Set CTS and CD Gates Message Format

Host CPU to iSBC Controller Messages

B.3.2.1.11 CLEAR CTS AND CD GATES. This command causes the controllers to transmit and receive on the specified line regardless of CTS and CD. This is the default condition after reset.

Figure B-16 shows the command format.

Message Format



NOTE
Reserved space should be set to 00(H).

Line Number — The serial channel on which the CTS and CD gates are cleared.

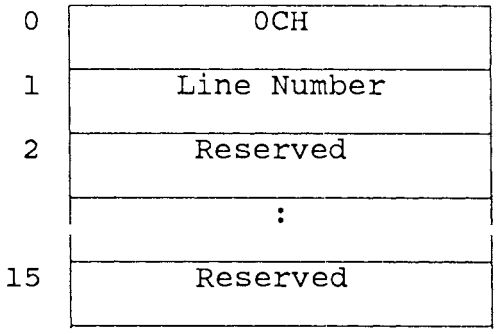
Figure B-16. Clear CTS and CD Gates Message Format

Host CPU to iSBC Controller Messages

B.3.2.1.12 SET DSR REPORT. This command causes the controllers to report changes of the DSR signal on the specified line.

Figure B-17 shows the command format.

Message Format



NOTE
Reserved space should be set to 00(H).

Line Number The serial channel on which DSR Report is set.

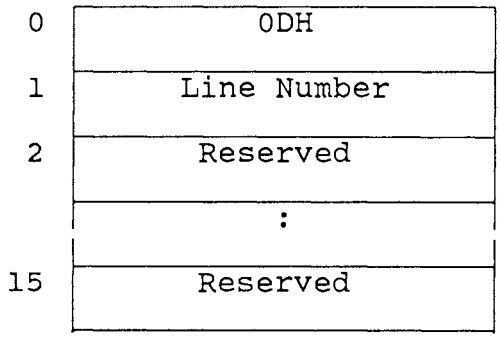
Figure B-17. Set DSR Report Message Format

Host CPU to iSBC Controller Messages

B.3.2.1.13 CLEAR DSR REPORT. This command cancels the previous request to report DSR changes on the specified line. This is the default condition after reset.

Figure B-18 shows the command format.

Message Format



NOTE
Reserved space should be set to 00(H).

Line Number — The serial channel on which DSR Report is cleared.

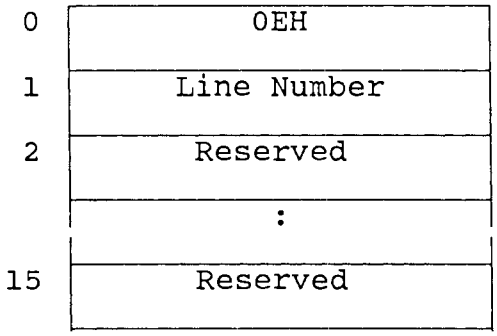
Figure B-18. Clear DSR Report Message Format

Host CPU to iSBC Controller Messages

B.3.2.1.14 SET RI REPORT. This command causes the controllers to report changes of the RI signal on the specified line.

Figure B-19 shows the command format.

Message Format



NOTE
Reserved space should be set to 00(H).

Line Number — The serial channel on which RI Report is set.

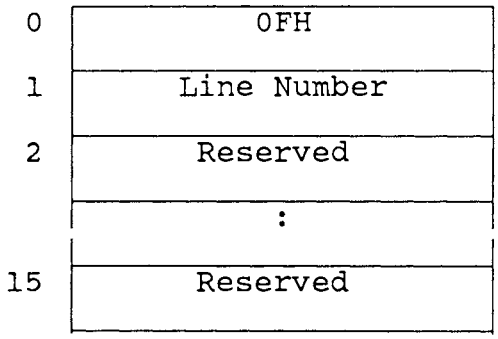
Figure B-19. Set RI Report Message Format

Host CPU to iSBC Controller Messages

B.3.2.1.15 CLEAR RI REPORT. This command cancels the previous request to report RI changes for the specified line.

Figure Figure B-20 shows the command format.

Message Format



NOTE
Reserved space should be set to 00(H).

Line Number — The serial channel on which RI Report is cleared.

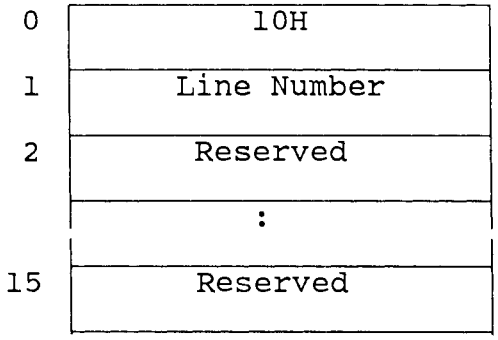
Figure B-20. Clear RI Report Message Format

Host CPU to iSBC Controller Messages

B.3.2.1.16 CLEAR DTR. This command is used to clear the Data Terminal ready modem signal on a serial channel.

Figure B-21 shows the Clear DTR message format.

Message Format



NOTE
Reserved space should be set to 00(H).

Line Number The serial channel on which DTR is cleared .

Response None

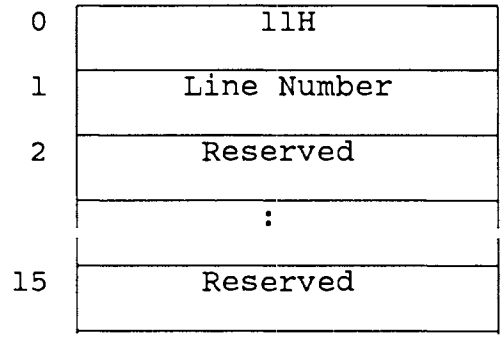
Figure B-21. Clear DTR Message Format

Host CPU to iSBC Controller Messages

B.3.2.1.17 SET BREAK. This command is used to force the transmission of continuous zeros (i.e. hold the line in a continuous spacing condition) on a serial channel. This command is ignored if a transmission is in progress on the line. Once a Set Break command is issued, it must be followed by a Clear Break command before any transmit Buffer commands are issued on the particular line.

Figure B-22 shows the Set Break message format.

Message Format



NOTE
Reserved space should be set to 00(H).

Line Number — The serial channel on which break is transmitted.

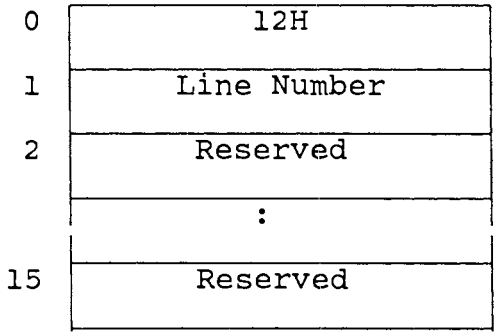
Response — None

Figure B-22. Set Break Message Format

Host CPU to iSBC Controller Messages

B.3.2.1.18 CLEAR BREAK. This command is used to clear a transmit break condition on a line caused by a previous Send Break command. Figure B-23 shows the Clear Break message format.

Message Format



NOTE
Reserved space should be set to 00(H).

Line Number The serial channel on which break is cleared.

Response None

Figure B-23. Clear Break Message Format

This page intentionally left blank

Host CPU to iSBC Controller Messages

B.3.2.1.19 DOWNLOAD. This command is used to download code into any location in the controllers address space. This allows the flexibility to load code into RAM not visible to the MULTIBUS. The Execute command can then be used to transfer control of the loaded code.

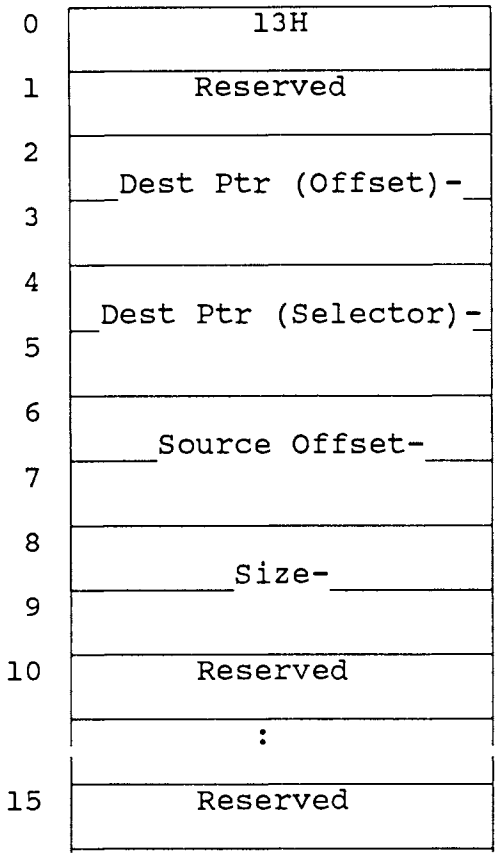
This command (or sequence of commands) is only allowed directly after a reset. After the download completes a Download Complete message is returned to the MULTIBUS host.

Since the Download command and subsequent Execute command use the normal message interface, care must be taken not to overwrite memory used to implement the queues or the lower 16K of local memory where the firmware maintains its data structures. Further, copying to nonexistent memory may hang up the processor.

The Test program boot mechanism may present an alternative to the use of this command. With the Test program boot mechanism the message to be dowloaded must be downloaded within 250 ms after a reset. Using the Download message still requires a reset but there is no time limitation.

Figure B-24 shows the Download command format.

Message Format



NOTE
Reserved space should be set to 00(H).

Dest Ptr — The 32-bit 8086 style pointer (offset + selector) to the location in the address space of the controller where the code is to be loaded.

Source Offset — The 16-bit offset in the controllers dual-ported RAM from where the controller is to copy the code to the destination address plus 16384. The code must be loaded into this area prior to issuing this command.

Size — The size in bytes of the code that is copied from dual-port RAM to the destination address.

Response — A Down-load Complete message is returned to the host when the operation completes.

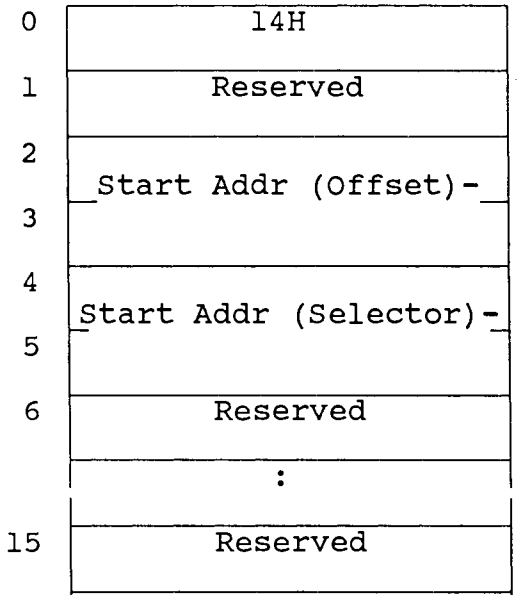
Figure B-24. Download Message Format

Host CPU to iSBC Controller Messages

B.3.2.1.20 EXECUTE. This command is used to transfer control to previously down-loaded code. The Execute command must be preceded by one or more Download commands.

Figure B-25 shows the Execute command message format.

Message Format



NOTE
Reserved space should be set to 00(H).

Start Address — The 32-bit 8086 style (offset + selector) start address of the down-loaded code. The firmware executes a far jump to this address.

Response — None.

Figure B-25. Execute Command Message Format

Host CPU to iSBC Controller Messages

B.3.2.1.21 CLEAR RECEIVE BUFFER. This command is used to respond to an Input Available message from the controllers. The Input Available message contains an address and a count describing a buffer where received data characters have been accumulated. The Clear Receive Buffer message is used to inform the controller of the number of characters the MULTIBUS host CPU has copied out of the receive buffer so that the controller can release the corresponding buffer space.

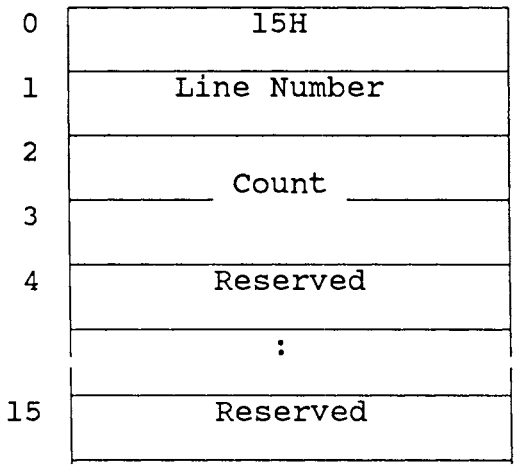
The Clear Receive Buffer message also serves an important synchronization function. The controllers ensure that at most, one Input Available message per line is pending. That is, it issues an Input Available message on a particular line only after any previously issued Input Available message has been acknowledged with a clear Receive Buffer message. In this manner, the host CPU can exercise flow control by delaying Clear Receive Buffer messages.

A Clear Receive Buffer message that is received when there is no outstanding Input Available message is ignored by the controllers.

This command has a special use in the clock/calendar and line printer interfaces (on the iSBC 546 board). When issued to these lines with count zero it is a request for input.

Figure B-26 shows the Clear Receive Buffer message format.

Message Format



NOTE
Reserved space should be set to 00(H).

Line Number — The serial channel from which characters have been cleared.

Count — The number of characters copied out of the receive buffer. The count can be 0. The count must not exceed the count specified in the corresponding Input Available message.

Response — None.

This message clears the board to send an Input Available message immediately, if the receive buffer is not empty.

Figure B-26. Clear Receive Buffer Command Message Format

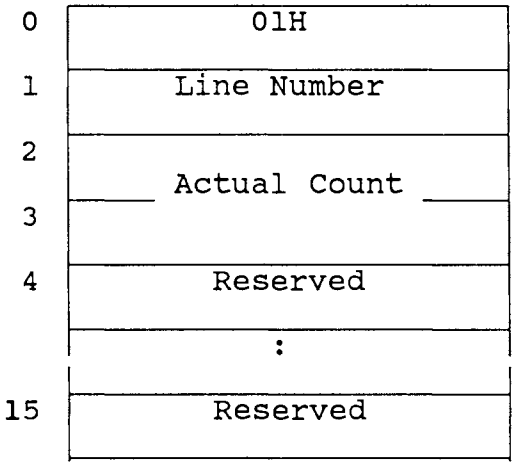
B.3.2.2 Controller To Host CPU Messages

Sections B.3.2.2.1 through B.3.2.2.12 describe messages sent between the controllers and the MULTIBUS host CPU.

B.3.2.2.1 TRANSMIT COMPLETE. This message is sent by the controllers to indicate the completion of a Transmit Buffer command previously issued by the host CPU. The message returns the actual number of characters transmitted. This message also clears the host CPU to request another Transmit Buffer operation.

Table B-27 shows the Transmit Complete message format.

Message Format



NOTE
Reserved space should be set to 00(H).

Line Number	The serial channel on which a previously issued Transmit Buffer command has completed.
Actual Count	The actual number of characters transmitted. Thi may be different from the number of characters specified in the Transmit Buffer request if the transmit operation was cancelled by the Abort Transmit command.
Response	None

Figure B-27. Transmit Complete Message Format

---

iSBC Controller to Host CPU Messages

B.3.2.2.2 INPUT AVAILABLE. This message is sent by the controllers to inform the host CPU of pending received characters.

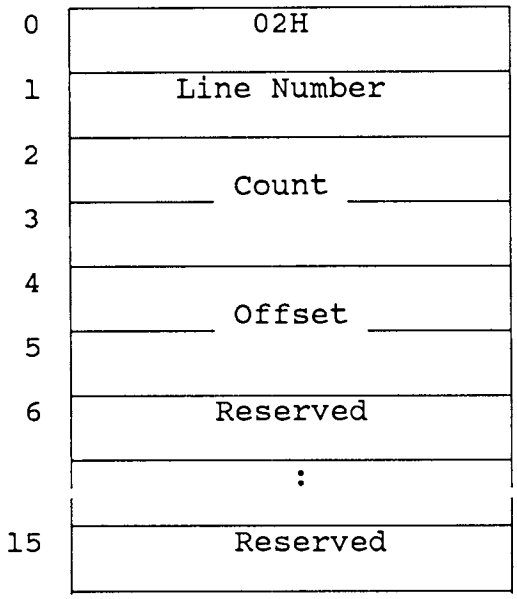
The host CPU must copy the received data into its own buffers and then signal the controllers to release their buffer space by sending it a Clear Receive Buffer message.

The controllers will not issue any further Input Available messages for the channel until the host CPU responds with a Clear Receive Buffer message. Thus some measure of flow control can be exercised by the host CPU.

Note, the controllers discard characters received with parity errors and replaces characters received with frame errors with eight bit nulls (00H). The received parity bit is transferred to the CPU unless eight bits/character is selected. In the 6 and 7 bits/character formats unused bit positions are set to 1.

Figure B-28 shows the Input Available message format.

Message Format



NOTE
Reserved space should be set to 00(H).

Line Number — The serial channel on which input has been received.

Offset — The 16-bit offset from the base of the controller memory to the beginning of the area where the received characters have been accumulated. The firmware adds 16384 bytes to maintain compatibility with other Intel products.

Count — The number of characters available starting at the above offset. Note that the receive buffer for each line is organized as a circular queue and the host CPU must account for any wrap-around implied by the offset and count.

Response — The host CPU must respond with a Clear Receive Buffer Message.

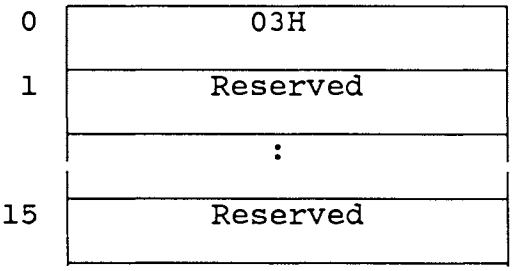
Figure B-28. Input Available Message Format

iSBC Controller to Host CPU Messages

B.3.2.2.3 DOWNLOAD COMPLETE. This message is sent by the controllers to inform the host CPU of the completion of a previously issued Download command. This message also clears the host CPU to issue another Download command or an Execute command.

Figure B-29 shows the Download Complete message format.

Message Format



NOTE
Reserved space should be set to 00(H).

Response None

Figure B-29. Download Complete Message Format

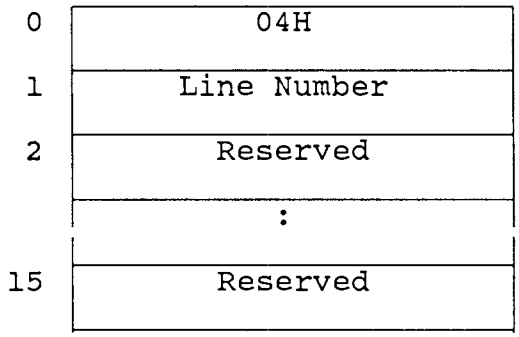
iSBC Controller to Host CPU Messages

B.3.2.2.4 CARRIER DETECT. The controllers send the host CPU this message when an OFF to ON transition is detected on the Carrier Detect Modem line.

If the Carrier Detect modem signal is asserted for the line when the Assert DTR command is received, the controllers return a Carrier Detect message even though an OFF to ON transition was not sensed on the Carrier Detect signal. Thus the host can maintain a state variable following the Carrier detect modem signal by toggling the variable when subsequent Carrier Detect and Carrier Loss messages are received.

Figure B-30 shows the Carrier Detect message format.

Message Format



NOTE
Reserved space should be set to 00(H).

Line Number — The serial channel on which the carrier was detected.

Response — None

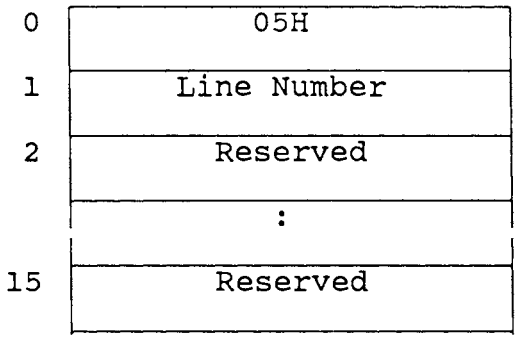
Figure B-30. Carrier Detect Message Format

iSBC Controller to Host CPU Messages

B.3.2.2.5 CARRIER LOSS. The controllers send the host CPU this message when an ON to OFF transition is detected on the Carrier Detect Modem line.

Figure B-31 shows the Carrier Loss message format.

Message Format



NOTE
Reserved space should be set to 00(H).

Line Number — The serial channel on which the carrier was lost.

Response — None

Figure B-31. Carrier Loss Message Format

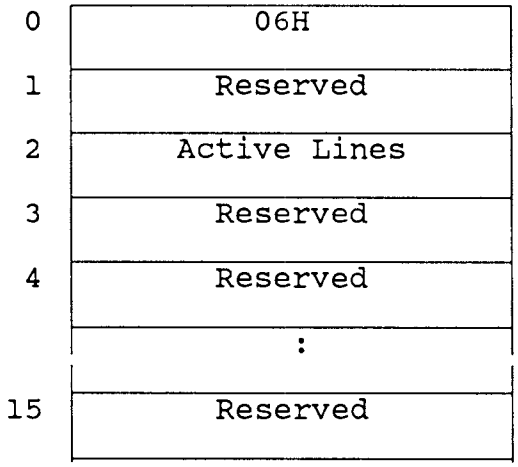
iSBC Controller to Host CPU Messages

B.3.2.2.6 INITIALIZATION RESPONSES. The controllers return this message in response to an Initialize command.

It returns a bit map of the active lines on the board. Only these lines may be used in subsequent line specific commands.

Figure B-32 shows the Initialization Responses message format.

Message Format



NOTE
Reserved space should be set to 00(H).

Active Lines — A bit map representing the lines that may be used in subsequent line specific commands. Bit i is 1 if and only if line i is active for $0 \leq i \leq 7$.

Response — None

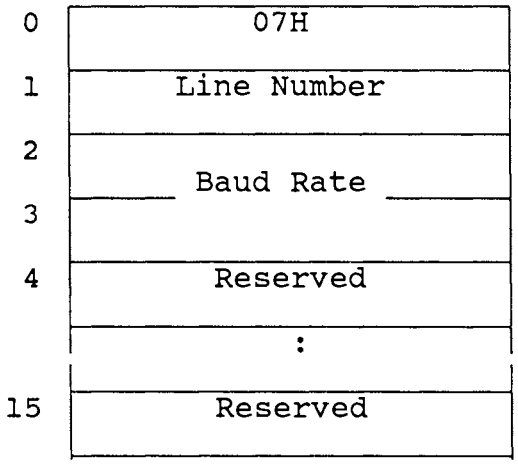
Figure B-32. Initialization Responses Message Format

iSBC Controller to Host CPU Messages

B.3.2.2.7 AUTOBAUD COMPLETE. The Controllers return this message after they have completed a baud rate scan initiated on a line by a previous Configure Line command. The host is allowed to issue line specific commands to the line after it receives this message.

Figure B-33 shows the Autobaud Complete message format.

Message Format



NOTE
Reserved space should be set to 00(H).

Line Number	The serial channel on which the baud rate has been recognized.
Baud Rate	The baud rate of the serial channel.
Response	None.

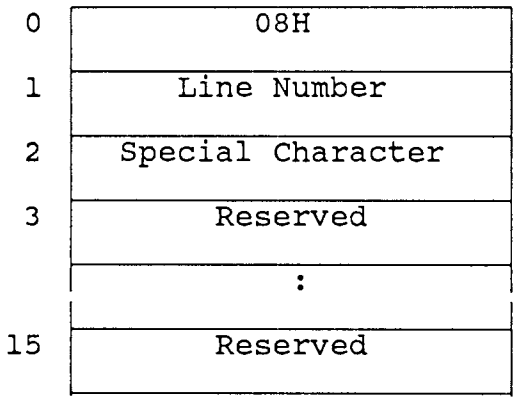
Figure B-33. Autobaud Complete Message Format

iSBC Controller to Host CPU Messages

B.3.2.2.8 SPECIAL CHARACTER RECEIVED. The controllers return this message in Signal Special Character Mode when a special character is received, and the number of characters in the receive buffer of the line exceeds the Signal Special Character high water mark.

Figure 5-34 shows the Special Character Received message.

Message Format



NOTE
Reserved space should be set to 00(H).

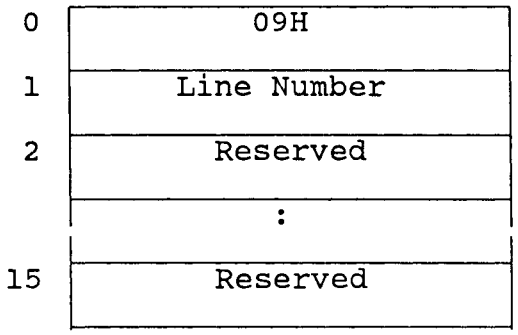
Line Number	The serial channel on which the special character is received.
Special Character	The special character received.
Response	None

Figure B-34. Special Character Received Message Format

iSBC Controller to Host CPU Messages

B.3.2.2.9 DSR DETECTED. This message reports DSR going active on the specified line. The line is in DSR Report Mode.

Message Format



NOTE
Reserved space should be set to 00(H).

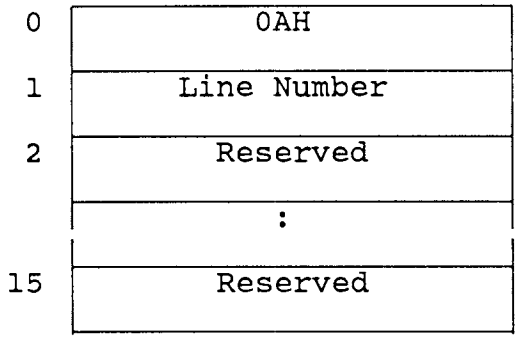
Line Number — The serial channel on which DSR becomes active.

Figure B-35. DSR Detected Message Format

iSBC Controller to Host CPU Messages

B.3.2.2.10 DSR LOST. This message reports DSR going inactive on the specified line. The line is in DSR Report Mode.

Message Format



NOTE
Reserved space should be set to 00(H).

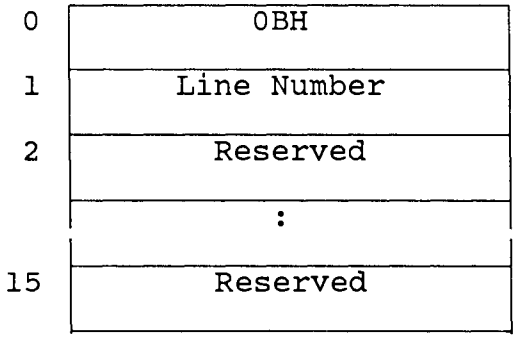
Line Number — The serial channel on which DSR becomes inactive.

Figure B-36. DSR Lost Message Format

iSBC Controller to Host CPU Messages

B.3.2.2.11 RI DETECT. This message reports RI going active on the specified line.

Message Format



NOTE
Reserved space should be set to 00(H).

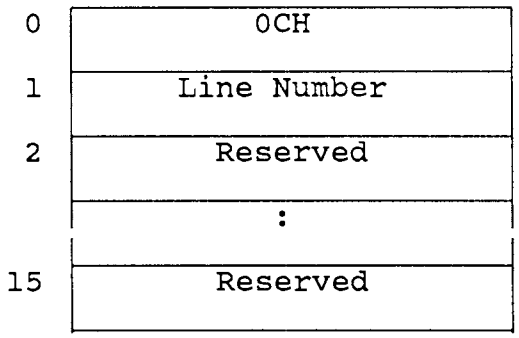
Line Number — The serial channel on which RI becomes active.

Figure B-37. RI Detected Message Format

iSBC Controller to Host CPU Messages

B.3.2.2.12 RI LOST. This message reports RI going inactive on the specified line.

Message Format



NOTE
Reserved space should be set to 00(H).

Line Number — The serial channel on which RI becomes inactive.

Figure B-38. RI Lost Message Format

B.3.2.3 Sample Host CPU to Controller Interaction

This section presents some typical message interchanges between a host CPU and a controller.

Example 1

This example shows a message interchange that typifies normal operation on a single line.

	HOST CPU		CONTROLLERS
		<<<RESET>>>	
1.	INITIALIZE	_____>	
2.		<-_____	INITIALIZATION RESPONSE
3.	CONFIGURE	_____>	
4.	ENABLE LINE	_____>	
5.	ASSERT DTR	_____>	
		<<<Modem establishes carrier>>>	
6.		<-_____	CARRIER DETECT
		<<<Host copies data into transmit buffer>>>	
7.	TRANSMIT BUFFER	_____>	
		<<<Actual Transmission>>>	
8.		<-_____	TRANSMIT COMPLETE
		<<<Data Received over the Serial Link>>>	
9.		<-_____	INPUT AVAILABLE
		<<<CPU reads Received Data>>>	
10.	CLEAR RECV BUFFER	_____>	
		<<<Carrier Drops>>>	
11.		<-_____	CARRIER LOSS
12.	DISABLE	_____>	

Example 2

This example illustrates a typical message exchange for a download and execute application

HOST CPU		CONTROLLERS
	<<<RESET>>>	
1. DOWNLOAD	_____>	
2.	<-_____	DOWNLOAD COMPLETE
3. DOWNLOAD	_____>	
4.	<-_____	DOWNLOAD COMPLETE
5. EXECUTE	_____>	

<<<Down loaded Code begins to execute>>>

B.3.2.4 Line Printer

The iSBC 546 provides a line printer controller interface. The interface, between the line printer and the iSBC 546 is Centronics compatible. To use the line printer controller, the same protocol is used as for an RS232C serial channel except the only input possible from this line is a byte of printer status. the line printer corresponds to iSBC 546 line #5 (Channel 4 when counting from 0 to 7).

To read the status of the line printer a Clear Receive Buffer command (with count = 0) must be sent to line #5. A Receive Buffer command is sent from the iSBC 546 board to the CPU host. The command consists of a single byte containing the status shown below:

Bit 4 Line Printer Busy. Logical 1 indicates line printer is busy, logical 0 indicates it is not busy.

Bit 5 Line Printer Out of Paper. Logical 1 indicates paper is out, logical 0 indicates there is paper.

Bit 6 Select. Logical 1 indicates line printer is selected, logical 0 indicates the printer is not selected.

Bit 7 Line Printer Fault Detected. Logical 1 indicates normal line printer operation. Logical 0 indicates a fault occurred in the line printer.

B.3.2.5 Clock Generator

The iSBC 546 provides a hardware clock and calendar with battery back-up. The clock/calendar is referred to as simply the clock, and the time/data are referred to as simply the time in this discussion. The clock corresponds to iSBC 546 line #6 (channel 5 when counting from 0 to 7).

The clock uses the Transmit and receive Buffers to set and read the time. To set the time a Transmit command is issued with 11 bytes of data in the Transmit Buffer. To read the time a Clear Receive Buffer command is issued, with 0 byte count, and in response 11 bytes of data are received in the Receive Buffer. The data is always communicated in Binary-Encoded Decimal (BCD) format. For both setting and reading, the 11 bytes of data have the following structures:

Byte	Meaning	Values
1	Thousandths of Seconds	x0
2	Hundredths and Tenths of Seconds	00 - 99
3	Seconds	0 - 59
4	Minutes	0 - 59
5	Hours	0 - 23
6	Day of Week	1 - 7
7	Day of Month	1 - 31
8	Month	1 - 12
9	Reserved (Set to Zero)	
10	Year	0 - 99
11	Month (Repeated)	1 - 12

B.3.3 PHYSICAL MESSAGE PASSING

This section describes the physical message passing scheme utilized for communication between the host CPU and the controllers.

B.3.3.1 Data Structures

Message passing is implemented using a pair of circular queues of message-sized (16 bytes) buffers. The circular queues are themselves implemented as 96 element arrays of 16 byte long buffers with head and tail pointers into these arrays.

The arrays are located in dual port memory. The IN Queue area refers to the array used for messages to the controllers and the OUT queue area refers to the corresponding array used for messages from the controller.

To be definitive consider the following array declarations made for the two queue areas and the control variables.

If In_ queue is non-empty then
In_queue/In_ queue_head/ is the next queue element to be processed by the controller.

If In_queue is non-full then
In _queue/In_queue_tail/ is the next free slot in In_queue

If Out_queue is non-empty then
Out-queue/Out-queue-head/ is the next queue element to be processed by the host CPU

If Out_queue is non-full then
Out_queue/Out_queue_tail/ is the next free slot in Out_queue

In_queue is empty if and only if
in_queue_head = In_queue_tail

In_queue is full if and only if
In_queue tail + 1 = In_queue_head
(modulo 96)

```
Out_queue is empty if and only if
                  Out_queue_head = Out_queue_tail

Out_queue is full if and only if
                  Out_queue_tail + 1 = Out_queue_head
                  (modulo 96)
```

B.3.3.2 Operations

To ensure correct operation the host CPU must use the following procedures to add elements to the In_queue (i.e. to send messages to the controllers) and to remove elements from the Out_queue (i.e. to receive messages from the controller).

In the following procedures the host CPU is allowed to modify In_queue_tail and Out_queue_head but is allowed only to read In_queue_head and Out_queue_tail.

Send Message

```
Wait until In_queue is non-full;
Copy message into In_queue/In_queue_tail/;
Increment In_queue_tail by 1 (modulo 96)
Signal the controller (write 02H to the flag byte I/O port)
```

Receive Message

```
Clear the interrupt request (write 04H to the flag byte I/O
port);

while Out_queue is non-empty
        Copy Out_queue/Out_queue_head/ into local message
        buffer;
        Increment Out_queue_head by 1 (modulo 96);
        Process message;
endwhile;
```

B.3.4 POWER-UP CONFIDENCE CHECKS

Upon power-up the controller firmware performs some confidence tests to verify the operation of the major on-board chips. When all tests are successfully completed the value FFH is stored in the Confidence Test Result Byte of the dual port RAM (offset = 19). If any test fails, the 80186 processor is halted and the Result byte contains an indication of the failed test.

Note that during the power-up tests, the processor runs with interrupts disabled.

Table B-2 shows the Confidence Test Result Codes and the corresponding tests performed. A description of each appears in the following sections.

Table B-2. Confidence Test Result Codes

Result	Test	Part	Board Location
00	EPROM Checksum Test	-	-
10	DRAM March Test	-	-
11	DRAM Ripple Test	-	-
30	PIT Countdown Test	80186	U30
50	SCC Register Test	82530	U24
51	SCC Register Test	82530	U15
52	SCC Register Test*	82530	U10
53	SCC Register Test*	82530	U1
60	Clock/calendar**	58167	U1

* iSBC 547 and 548 only
** iSBC 546 only

B.3.4.1 EPROM Checksum Test

The Controller firmware resides in EPROMs. The contents of the EPROMs are static, so the unsigned byte-wise sum of all EPROM addresses is known. This sum can be anywhere between 0 and 255 times the address length of the EPROM. The sum is stored at the lowest address of the EPROM and is called a checksum.

The checksum is stored as a DWORD under the PL/M Language data storage conventions. This means that the most significant bits are

in higher addresses than the less significant bits. The highest byte of the checksum is defined as always masked to zeros. This is because the EPROM size is never more than 64 KBytes.

Figure B-29 shows the format of the Checksum Test. Note that the checksum is defined not to include the sum of the EPROM addresses in which the checksum itself is stored.

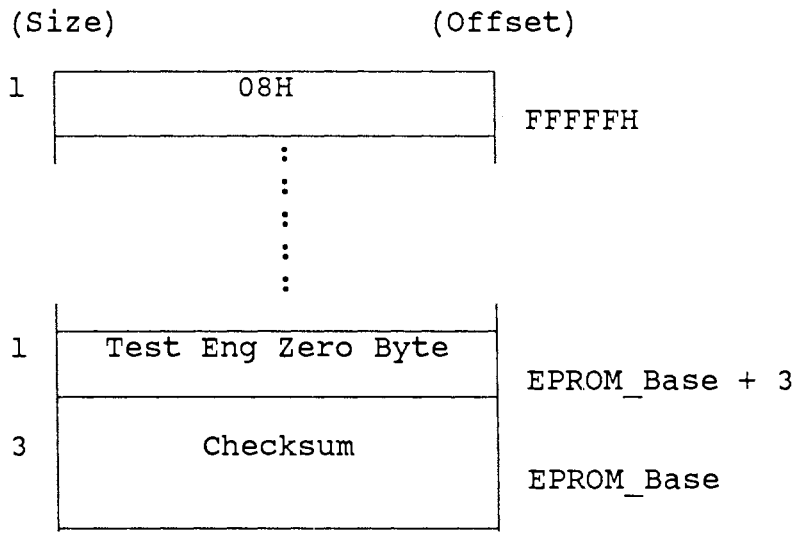


Figure B-39. EPROM Checksum Test

The Checksum Test is performed using the following procedure:

```
Checksum = 0;
for EPROM_address = EPROM_Base + 4 to FFFFFH do
        Checksum = Checksum + unsigned_contents_of
                   (EPROM_address);

end for;
 Checksum = Checksum and FFFFFFH;
 if Checksum is not equal to Checksum_at_EPROM_base then
        fail_test
```

B.3.4.2 DRAM TESTS

The dynamic RAM is tested for integrity by writing a pattern to an address and then reading the pattern back for comparison. Patterns are chosen to test both ON and Off states. The procedure is repeated for all of the on-board non-dual ported RAM.

```
for RAM_address = 0 to 17FFEH do
        for each Pattern do
                write Pattern to RAM_address
                read Check Pattern from RAM_address;
                if Check Pattern is not equal to Written
                Pattern then
                        fail_test;
                  end if;
            end for;
   end for;
    pass_test;
```

B.3.4.2.1 MARCH TEST. The March Test uses two patterns per address tested. The first is 0101010101010101B and the second is its opposite 1010101010101010B.

B.3.4.2.2 RIPPLE TEST. The Ripple Test uses sixteen patterns per address tested. The first is 0000000000000001B and subsequent patterns are generated by shifting this pattern left by one bit at a time until 1000000000000000B is reached.

B.3.4.3 PIT Countdown Test

The PIT Countdown Test verifies the operation of the Programmable Interval timers by counting the counters down and generating an interrupt.

The following procedure is used for each of the three timers in the 80186.

```
program 80186 PIC;
set IMR to mask all but level 0; program PIT
program PIT modes as:
        Priority 0;
        Software triggered strobe (CONT =0, ALT = 0)
load count time (Max Count Register A) as 0064H;
        loop up to 100 times
        read PIC POLL Register;
        if most_significant_bit is ON and highest_level = 0 then
              pass test;
       end if;
end loop;
fail test:
```

B.3.4.4 SCC Register Test

The SCC Register Test verifies the ability of the host to access the Time Constant Registers of the Serial Communications Controllers. The following procedure is used:

```
               loop twice using March Patterns
               write March Pattern to Time Constant Registers
               (WR12 and WR13);
               read Check Pattern back (RR12 and RR13);
               if Check Pattern is not equal to Written Pattern
               then
                          fail test;
                    end if;
             end loop;
              pass_test;
```

B.3.4.5 Clock/Calendar Test

The clock/calendar test verifies the ability of the 58167 component to be set and to keep time. The following procedure is used:

```
Set timer to interrupt every 1.5 msec
On the first interrupt read and store time;
On the second interrupt set time to 12:31:7:23:59:99:90;
On the third interrupt read the time, it should be
1:1:1:0:0:0:0:xx
       else test fails;
On the fourth, do nothing;
On the fifth interrupt set the time to the value stored plus
6 msec;
```