CA-12
96 LINE PARALLEL I/O

Preliminary Manual

September 1978

Introduction

CA-12 96 line parallel offers interfacing capability for up to 96 TTL compatible inputs and/or outputs. It also provides 24 interrupt control lines for interrupt, control and handshaking use. Typical applications include interfacing to relays, AC controllers, remote switches and parallel interface peripherals such as paper tape readers, punches, mag tape units and printers.

The CA-12 board is unique in that it actually is three separate PC boards. One small interface board plugs into the computer and occupies one slot. It can be used in any OSI computer since it requires just 5 volts. This board has two 16 pin connectors which are normally connected to two auxiliary boards via 4 ft. ribbon cables. Each of these auxiliary boards contain three PIA chips which collectively provide 48 I/O lines and 12 interrupt control lines (per board).

Each of the PIA boards has provisions for prototyping and Molex connectors. Each board is designed so it can be cut down to two PIA's or one PIA without disrupting the support circuitry for the reduced capacity. This is so that it can be conveniently incorporated in other equipment. This unique approach to parallel interfacing greatly relieves the congestion around parallel interfaces within the computer since it effectively provides 96 remote interface lines via two easy to handle 16 pin ribbon cables.

The system works quite reliably with 4 ft. ribbon cables. Under low noise conditions, the system could have much longer ribbon cables to the PIA ports.

Software

Each PIA line can be programmed as an input or an output. It is TTL compatible and can drive up to two TTL loads. It will be necessary, however, for the user to add his own buffering and protection circuitry to the PIA's. Sample software is provided in BASIC on how to utilize the PIA ports. The board is very simply operated with PEEKS and POKES directly in BASIC at speeds up to a few hundred transitions per second.

For high speed operation such as in controlling a high speed magnetic tape unit, machine code subroutines can be utilized either in conjunction with machine code programs or BASIC programs. Such machine code programs can be modeled directly after the BASIC PEEK and POKE instructions by substituting store accumulator extended instructions for POKEing and load accumulator extended instructions for PEEKing specific memory, hence, PIA register locations.

PIA Programming

There is a total of six internal registers in the PIA, four of which must be set up before the PIA can be used for any specific tasks. This setup is called PIA initialization. In a dedicated application such as an industrial controller, the PIA would be set up on system power-up by code stored in a PROM or ROM. In a general purpose computer, the setup of undedicated ports is the responsibility of the user. Thus, the user must set up the PIA, or initialize it, whenever he wants to use it.

The following program in BASIC is a PIA exerciser containing a general purpose subroutine starting at line 1000 for initializing PIA's for user programs. There are as stated above, six internal registers in the PIA, but only four can be accessed as memory locations at a given time. These registers normally occupy four consecutive memory locations. We consider the address of the PIA to be X, as it is stated in the program. Thus, location X is the address of the peripheral interface register A, or its corresponding data direction register. Address X+1 is the status register in conjunction with the A-port of the PIA. A bit in the status register specifies whether the data direction register of the peripheral interface register or its corresponding data direction register as specified by a bit in its corresponding control, or status register at location X+3. A logical 1 in a bit location in the data direction register specifies that the corresponding pin on its port will be an output. An Ø in that location specifies that that pin will be an input.

The BASIC program further demonstrates the use of the PIA in a simple application. The program goes to line 1000, which inputs the base address of the PIA into the variable X. It then asks if the A-side will be an input or an output, asks the same of the B-side, and stores the responses as strings A$ and B$. It then performs an initialization of the PIA by setting the PIA's control registers to Ø. These control registers are located at X+1 and X+3. By setting these registers completely to Ø, the data direction registers for Ports A and B are accessible at addresses X and X+2, respectively. The user must then specify whether he wants the ports to be inputs or outputs. Line 1040 will set the data direction register lines all low if A$=I, or if the user specifies I for input. Otherwise, it sets the data direction register lines all high, specifying that Port A will be an output (see line 1045). Lines 1050 and 1055 perform the same function for Port B. Line 1060 then POKEs Ø4 into the control register for both Port A and Port B, located at X+1 and X+3. This switches the peripheral interface registers into addresses X and X+2. The peripheral interface registers are the registers which actually input or output data to the pins on the port once the PIA is configured by the subroutine. This subroutine at 1000 to 1070 thus initializes the PIA by setting the control registers to Ø, then specifies the data direction of both Port A and Port B, and finally restores the peripheral interface register to addresses X and X+2, so that the PIA can be used as a simple I/O device.

The mainline program is then at lines 30 to 260. The program then asks whether you would like to work with Port A or Port B. It checks on the entry at A$ and B$ to find out whether A or B

is an input or an output. If the port you have selected is an input, it then reports the current value on the input pins. If the port you select is an output, it asks you for the bit pattern variable K, and then outputs that bit pattern. When the output occurs, it is latched by the PIA and will not change until the program changes it or the PIA is reset. One important feature of the PIA's is that they must be reset on, or immediately after, power-up by a master reset and then configured by software. The normal configuration for a PIA immediately after power-up shows all lines as inputs, having generally drifted to a logical high state. Thus, if the PIA is read immediately after initialization, with nothing connected to its inputs, it will generally report all highs, and the data direction registers will specify that all lines are inputs. This program is very useful for testing PIA ports in various portions of your system, and can also be used to test I/O devices which you connect to the PIA.

A simple test procedure for the PIA output is to connect a digital voltmeter or VOM between ground and the output pin, and via this program, toggle that pin (or the entire port, for that matter) high and then low, and observe the voltage swing on the voltmeter. One PIA port can be used as a signal source to check inputs of another PIA port. For example, Port A can be configured as an output, and then alligator clips from its connector can be joined to Port B which could then be configured as an input. You could then attempt to write a bit pattern out on Port A and observe the same bit pattern coming back on Port B. These ports, of course, can be used for simple applications such as a switch register and light register for games.

The PIA actually has many more features than are used here. For example, the 6820 PIA has complete interrupt controls for both Ports A and B plus handshaking lines and several other status bits in its control registers. A complete description of the PIA operation is in the Motorola M6800 Microcomputer System Design Data Book.

Installing A CA-12

The 570 board which is the CA-12 driver board can be plugged in directly to any Ohio Scientific computer system that supports the 48 line BUS including C2-4P's, C2-8P's and Challenger III's. The 570 board is so short that it does not require any additional support, however, care must be taken to plug the board in right side up. This can be done by observing that the component side of this board is facing the same direction as the component sides of all other boards of the system. The 572 remote boards can be routed out the back of an 8-slot or 16-slot Challenger case or simply run out the bottom of the C2-4P case.

Check-Out

To check out and program the CA-12 system, the user should refer to the sample program listed in the manual in conjunction with the register address map and the schematic diagrams which indicate which physical PIA position corresponds to which logical PIA board. This test as suggested elsewhere should be performed with a voltmeter.

Interfacing To The CA-12

The CA-12's 572 remote board has several provisions for easy interfacing and mechanical mounting. All important PIA signals are routed out to a row of doughnut pads and are clearly labeled with nomenclature consistent with the Motorola/AMI literature on PIA's. Wires can be soldered directly to the doughnuts or wire wrap pins can be put in place. In addition, these pins are further routed out to another bank of doughnuts and subsequent Molex connectors for connection to external cables. These foil connections can be cut and reordered as necessary. The boards also contain

eight 16 pin prototyping areas which can be utilized for free wire or wire wrap prototyping of additional components and/or ribbon cable sockets as necessary.

Mechanical Considerations

The CA-12 572 boards have several blank doughnuts which can be drilled out and used as mounting holes as necessary. The boards can be cut sheared or nibbled to smaller size as necessary. The smaller size configurations that are possible are specified by arrows on the board. The prototyping and socket area can be cut off the boards without adversely affecting performance. Each board can be cut down to two or one PIA if space is at a premium. If one or two PIA circuits are removed from the board, it will be necessary to relocate the reset capacitor and resistor to the active PIA's. This RC network resets the PIA on power up.

Technical Information

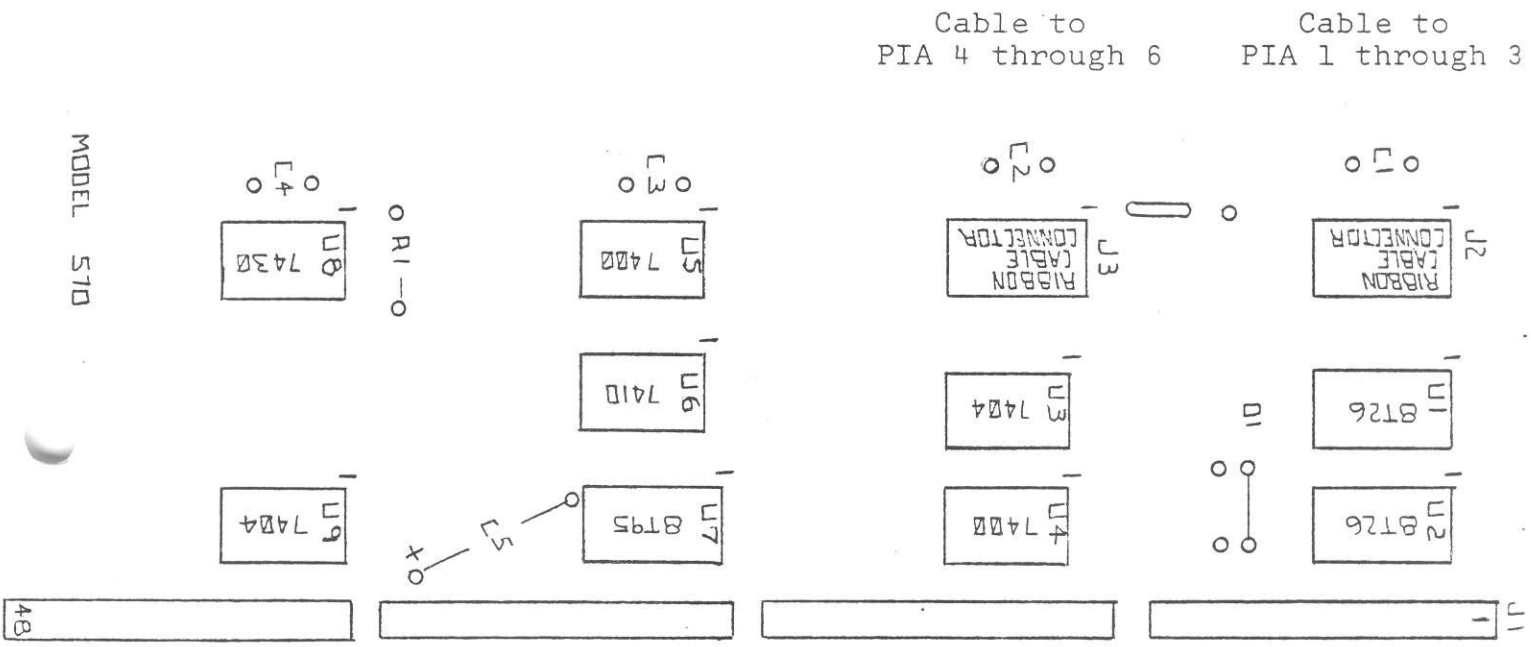
The CA-12 is delivered with Motorola 6821 PIA's. These are physically identical to 6820's and 6520's except that they have the capability to drive two TTL loads instead of one TTL load. Ohio Scientific recommends the use of Motorola 6821 PIA's in this circuit application because of its higher drive capability and improved short circuit protection over the earlier 6820 and 6520 PIA's. However, these units can be used interchangeably on the board if absolutely necessary. The 6821 PIA programs identically to the 6820 PIA's so that the accompanying spec sheet is appropriate. For further investigation and interfacing specifications for the PIA circuits, please refer to the detailed PIA specification sheets included in this manual.

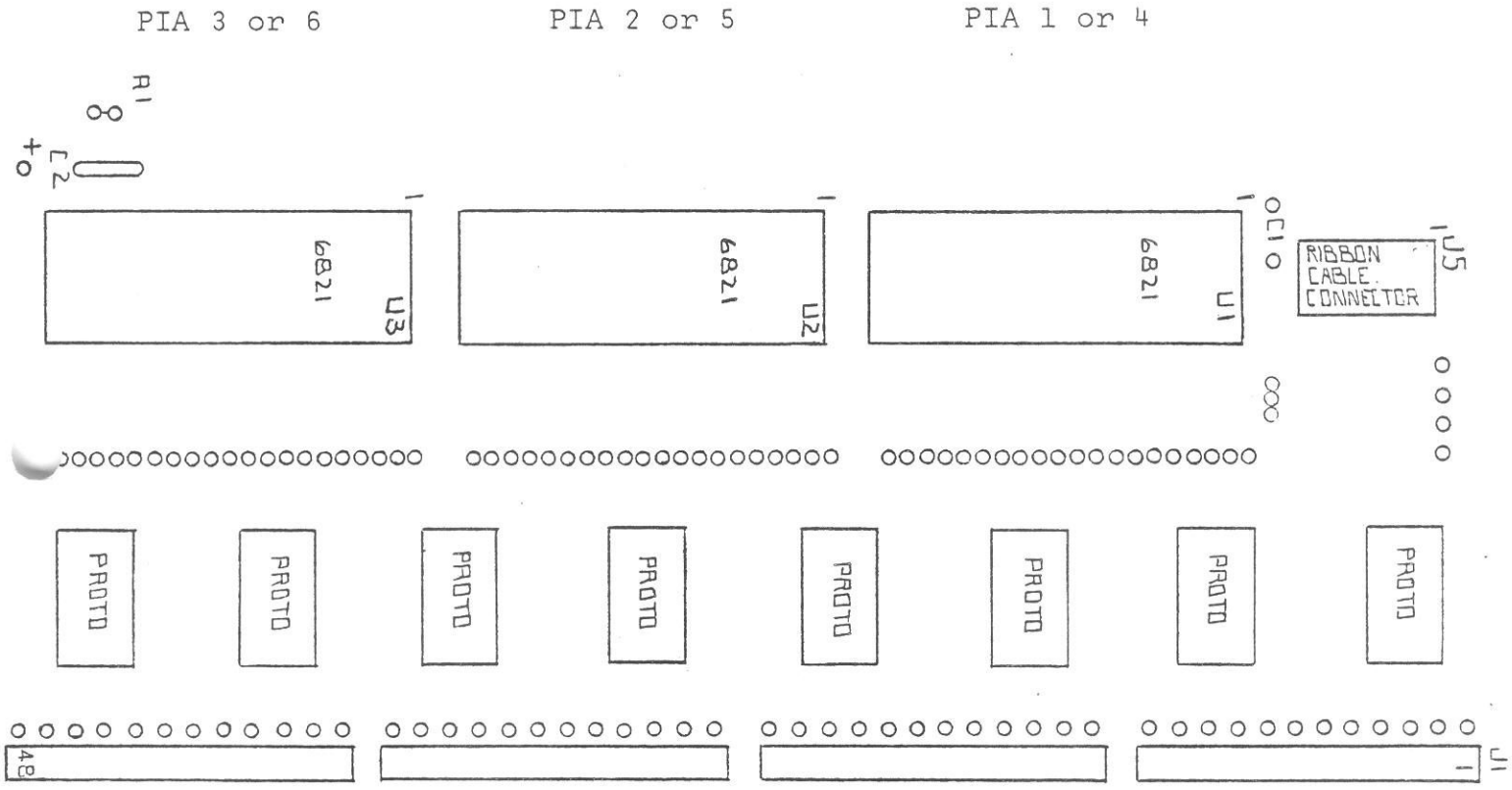
CA-12 PIA Address Map

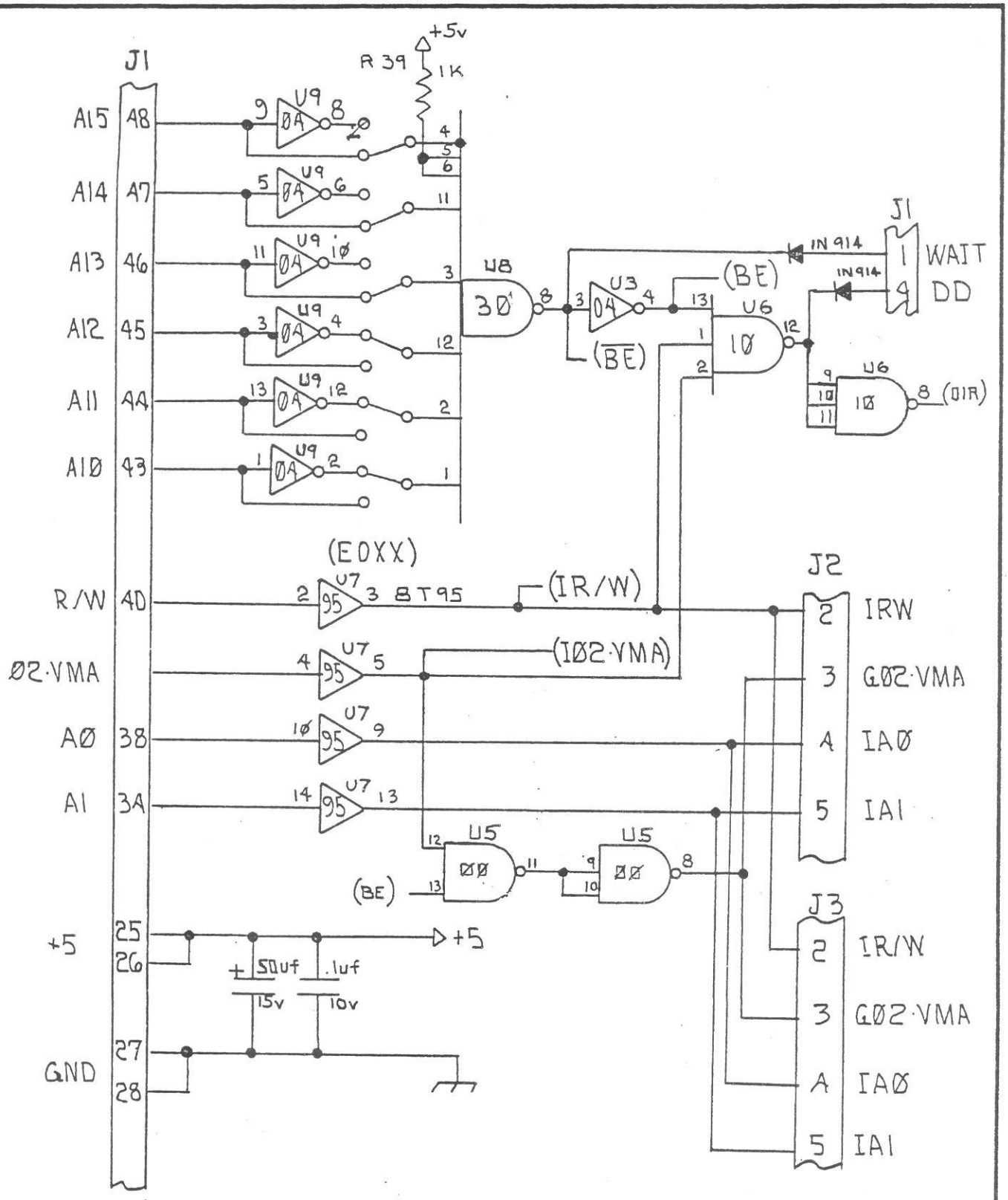
	Dec.	Hex	Relative
PIA 1	57348	E004	X
PIA 2	57352	E008	X+4
PIA 3	57356	E00C	X+8
PIA 4	57364	E014	X+16
PIA 5	57368	E018	X+20
PIA 6	57372	E01C	X+24

The CA-14 has no assigned address. It is delivered strapped at E000 Hex up but should be rejumpered for the desired location in your particular system.(E000 is used for the hard disk buffer and 6800 and 8080 compatible configurations.)

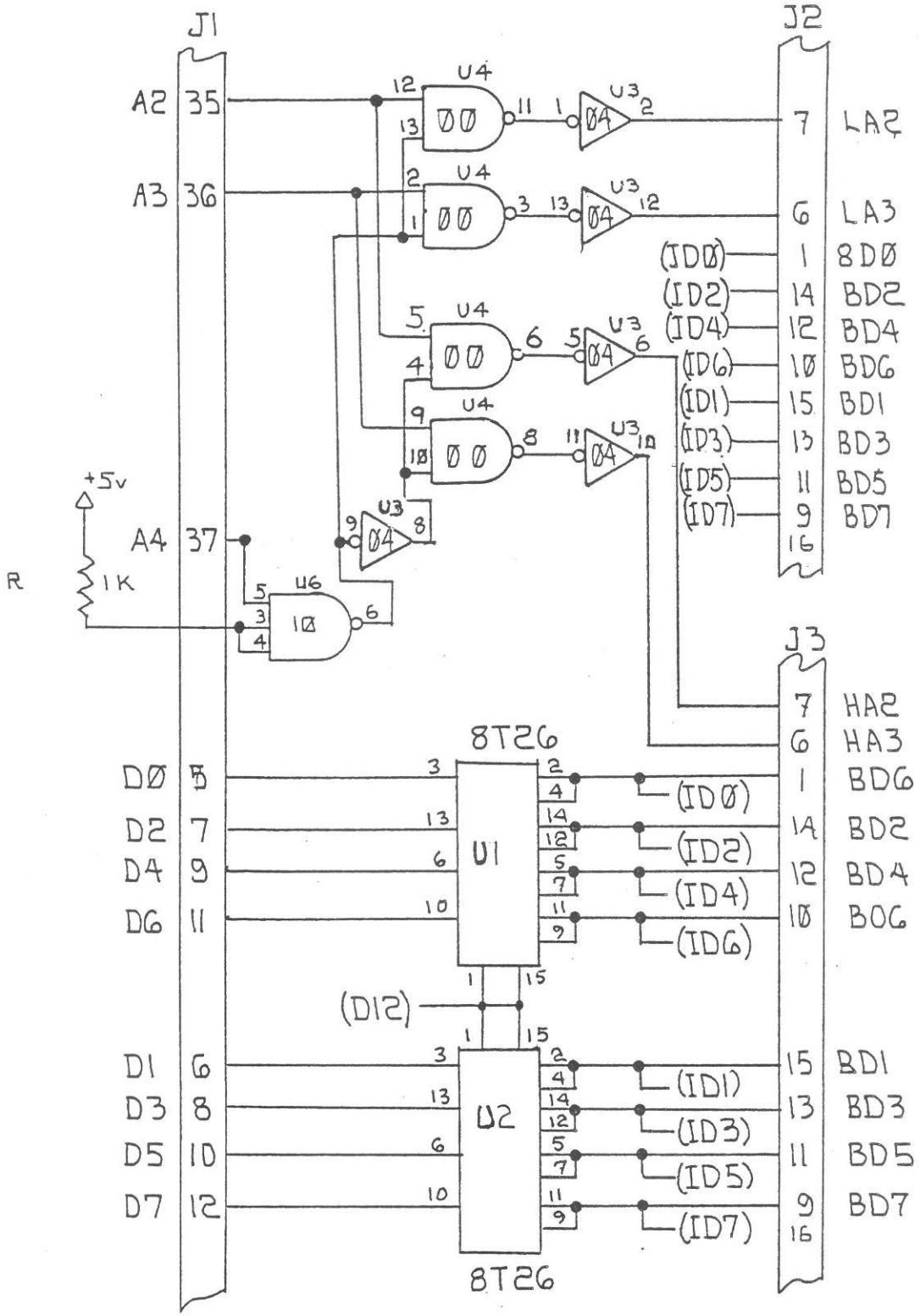
Address strapping is shown on Page 1 of the accompanying schematics.


Cable to
PIA 4 through 6
Cable to
PIA 1 through 3
MODEL 510
J1
J2
J3
RIBBON
CABLE
CONNECTOR
U1 8T26
U2 8T26
U3 74Ø4
U4 74ØØ
U5 74ØØ
U6 741Ø
U7 8T95
U8 743Ø
U9 74Ø4
R1
C1
C2
C3
C4
C5
48


PIA 3 or 6
PIA 2 or 5
PIA 1 or 4
R1
8
C2
+o
6821
U3
6821
U2
6821
U1
RIBBON
CABLE
CONNECTOR
J5
PROTO
PROTO
PROTO
PROTO
PROTO
PROTO
PROTO
PROTO
48
J1



OHIO SCIENTIFIC			product name/number CA-12 PARALLEL INTERFACE CONTROLLER	
date	revision	page	status	sheet 1 of 4



OHIO SCIENTIFIC			product name/number CA-12 PARALLEL INTERFACE CONTROLLER	
date	revision	page	status	sheet 2 of 4

BOARD	CONN	FUNCTION	# LINES / TYPE
CONTROLLER BD.	J1	BACKPLANE	48-MOLEX
	J2	CONN TO INTERFACE 1	16-RIBBON
	J3	CONN TO INTERFACE 2	16-RIBBON
	J4	POWER	4-MOLEX
INTERFACE BDS.	J5	CONN. FROM J2/J3	16-RIBBON
	J6	PARALLEL INTERFACE I/O	48-MOLEX
	J8	POWER	4-MOLEX

OHIO SCIENTIFIC			product name/number CA-12 PARALLEL INTERFACE CONN.	
date	revision	page	status	sheet 4 of 4