CLUSTER ARCHITECTURE

A cluster is a collection of workstations connected by a data communications line. Each workstation includes the memory and processing ability required for the workstation operator. This type of configuration, called distributed processing, allows the processing capabilities of the system to grow uniformly with the number of users.

Cluster workstations are connected by a high-speed (usually 307 kilobaud) RS-422 differential data communications line composed of two twisted pairs of wires: one for data and one for clock. This allows for multidrop, half-duplex operation. A variant of the Advanced Data Communications Control Procedures (ADCCP) protocol is used between workstations for communications.

A cluster is configured to have one master workstation, which polls the other workstations on the cluster. All communications pass between the master workstation and a cluster workstation; communication cannot occur directly between cluster workstations. The cluster workstations share the peripheral devices of the master workstation. The AWS-220 and -230 have their own mass storage peripherals and can either stand alone or share peripherals with the master workstation. Because of its greater mass storage features, the AWS-240 can operate as a master workstation.

Mechanically, the cluster is linked in a daisy-chain configuration. Each cluster workstation has two 9-pin female connectors connected to have parallel electrical lines. Cluster cables consist of two 9-pin male connectors joined by a cable consisting of two twisted pairs of wires and a ground shield.

The daisy chain consists of cluster workstations each having one 9-pin connector with a cable that leads to the master workstation (or to the cluster workstation between it and the master workstation) and one cable that leads away from the master workstation to the next cluster workstation. Cluster workstations at the end of the daisy chain have a special termination connector in place of the cable.

CLUSTER COMMUNICATIONS

Cluster communications are handled by an NEC 7201 (or Intel 8274) MPSC (Multi-Protocol Serial Controller) chip. The MPSC contains two serial channels: A and B. Channel A is used only for cluster communications. Channel B is used for communications between the keyboard and the CPU. While Channel B is similar in operation to Channel A, it is programmed in a completely different manner. For a discussion of Channel B and an example of how it is programmed, see "Keyboard and Keyboard Communications" below.

Under control of Channel 1 of the 8257, Channel A of the 7201 transfers data at speeds of up to 410 kilobaud on the half-duplex, multidrop, RS-422 cluster communications line. The communications protocol used by Convergent software is a variant of Advanced Data Communications Control Procedures (ADCCP), much of which is handled directly by the 7201. Since Channel A is multidrop (that is, the same lines can be driven by any one of several transmitting workstations), the Channel A Request to Send signal is used to ensure that only the transmitting workstation drives the communications lines.

Preparing Channel A for operation involves three steps:

1. Several write registers in the 7201 must be programmed to perform the operation (see Figure 2-5, the "7201 Programming Example" below).

2. The baud rate must be selected by programming the 8253 counter/timer.

3. DMA Channel 1 must be programmed to handle cluster communications. DMA programming is discussed above in "Direct Memory Access."

Control and Status Registers

Channel A is programmed using three input/output ports:

Port	Use
60h	Channel A data (cluster data)
62h	Channel A commands/status
66h	Write Register 2 only

Channel A has a set of three status (read) registers and eight command (write) registers. The CPU transfers information to and from these registers by input/output Port 62h. Port 66h is used by both Channels A and B to store the interrupt vector. The Pointer Register in the 7201 selects which read or write register is accessed when Ports 62h and 66h are used. When either a manual or power-up reset occurs, the Pointer Register is set to 0. Any read or write to Ports 62h or 66h then accesses Read Register 0 or Write Register 0, respectively. The Pointer Register is programmed when Write Register 0 is written to. The Pointer Register is reset to 0 after any command or status access is made to a read or write register other than 0.

For example, to access Read Register 2, a 2 is written to Write Register 0. The next read accesses Read Register 2. Figure 2-4 shows the register hierarchy.

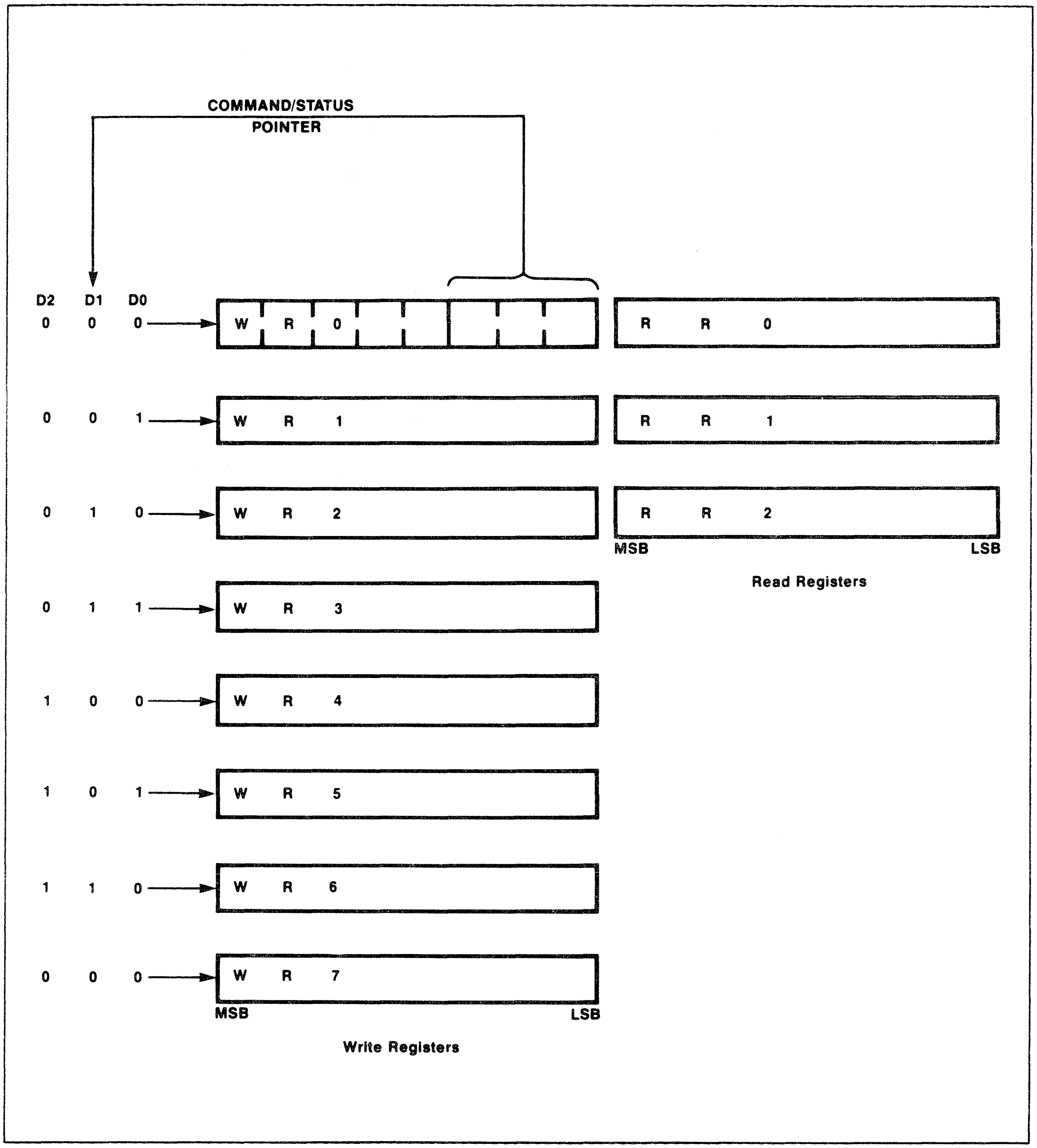

Figure 2-4. 7201 Register Hierarchy.

READ REGISTER 0 (PORT 62h STATUS)

Register AL Bit (CPU)	Read Information
0	Receive character available
1	Interrupt pending
2	Transmit buffer empty
3	Carrier detect
4	Synchronize/Hunt
5	Clear to send
6	Transmit underrun/End of message
7	Break/Terminate

READ REGISTER 1 (PORT 62h STATUS)

Register AL Bit (CPU)	Read Information
0	All sent
1-3	Residue code
4	Parity error
5	Receive overrun
6	CRC/Framing error
7	End of frame

READ REGISTER 2 (PORT 62h STATUS)

Register AL Bit (CPU)	Read Information
0	Present interrupt vector (least significant bit)
1-6	Present interrupt vector
7	Present interrupt vector (most significant bit)

WRITE REGISTER 0 (PORT 62h COMMAND)

Register AL Bit (CPU)	Write Information
0-2	Pointer Register (see below)
3-5	Command code (see below)
6-7	CRC control (see below)

Pointer Register

AL2	AL1	AL0	Register to be Accessed
0	0	0	Read/Write Register 0
0	0	1	Read/Write Register 1
0	1	0	Read/Write Register 2
0	1	1	Write Register 3
1	0	0	Write Register 4
1	0	1	Write Register 5
1	1	0	Write Register 6
1	1	1	Write Register 7

Command Code

AL5	AL4	AL3	Command
0	0	0	Null command
0	0	1	Send terminate
0	1	0	Reset external/status interrupts
0	1	1	Channel reset
1	0	0	Enable interrupt on next receive
1	0	1	Reset transmit interrupt/DMA pending
1	1	0	Error reset
1	1	1	End of interrupt

CRC Control

AL7	AL6	Action
0	0	External interrupt enable
0	1	Reset receive CRC checker
1	0	Reset receive CRC generator
1	1	Reset underrun/EOM latch

WRITE REGISTER 1 (PORT 62h COMMAND)

Register AL Bit (CPU)	Write Information
0	External interrupt enable
1	Transmit interrupt/DMA enable
2	0 = fixed vector (Channel B only)
3-4	Interrupt control (see below)
5	Wait on receive
6	Must be 0
7	Wait enable

Interrupt Control

AL7	AL6	Command
0	0	Receive interrupt/DMA enable
0	1	Receive interrupt on first character or special condition
1	0	Interrupt on all receive characters or special conditions; parity affects vector.
1	1	Interrupt on all receive characters or special conditions; parity does not affect vector.

WRITE REGISTER 2 (PORT 62h COMMAND)

Command affects both channels.

Register AL Bit (CPU)	Write Information
0-1	Configuration code (see below)
2	Priority bit (see below)
3-5	Interrupt code (see below)
6	0
7	0

Configuration Code

AL1	AL0	Action
0	0	DMA not used
0	1	Channel A DMA, Channel B interrupts
1	0	Both channels DMA
1	1	Invalid

Priority Bit

AL2	Interrupt Priorities (in decreasing order)
0	Receive Channel A Transmit Channel A Receive Channel B Transmit Channel B External status Channel A External status Channel B
1	Receive Channel A Receive Channel B Transmit Channel A Transmit Channel B External status Channel A External status Channel B

Interrupt Code

AL5	AL4	AL3	Interrupt Mode
0	X	X	Nonvectored
1	0	0	8085 mode 1
1	0	1	8085 mode 2
1	1	0	8088 mode

WRITE REGISTER 2 (Port 66h COMMAND)

Command affects both channels.

In the status-affects-vector mode, bits 0-2 of the vector are called <u>base vector bits</u> because they depend on the cause of the interrupt.

Register AL Bit (CPU)	Write Information
0	Base vector bit 0 (least significant bit)
1-6	Base vector bit 1-6
2	Base vector bit 7 (most significant bit)

WRITE REGISTER 3 (PORT 62h COMMAND)

Register AL Bit (CPU)	Write Information
0	Receive enable
1	Synchronization character load inhibit
2	Address search mode
3	Receive CRC enable
4	Enter hunt mode
5	Auto-enables
6-7	Length code (see below)

Length Code

AL7	AL6	Receive Character Length (bits)
0	0	5
0	1	7
1	0	6
1	1	8

WRITE REGISTER 4 (PORT 62h COMMAND)

Register AL Bit (CPU)	Write Information
0	Parity enable
1	Even parity
2-3	Synchronous/Asynchronous code (see below)
4-5	Synchronization type (see below)
6-7	Divisor code (see below)

Synchronous/Asynchronous Code

AL3	AL2	Transmission Mode Selected
0	0	Synchronous mode
0	1	1 stop bit
1	0	1 stop bit
1	1	2 stop bits

Synchronization Type

AL5	AL4	Synchronization Character Used
0	0	Character synchronous (1 character) mode
0	1	Character synchronous (2 character) mode
1	0	Bit protocol mode
1	1	External synchronization

Divisor Code

AL7	AL6	8253 Clock Internal Divisor
0	0	1
0	1	16
1	0	32
1	1	64

WRITE REGISTER 5 (PORT 62h COMMAND)

Register AL Bit (CPU)	Write Information
0	Transmit CRC enable
1	Request to send
2	CRC select
3	Transmit enable
4	Send break
5-6	Length code (see below)
7	Data terminal ready

Length Code

AL6	AL5	Transmit Character Length (bits)
0	0	5
0	1	7
1	0	6
1	1	8

WRITE REGISTER 6 (PORT 62h COMMAND)

Address is in bit protocol mode, and synchronization character is in bit synchronous mode. This register must be programmed with either a secondary address (for a bit-synchronous address mark), or a synchronization character for the character synchronous mode.

WRITE REGISTER 7 (COMMAND)

Flag is in bit protocol mode, and second synchronization character is in bit synchronous mode. This register must be programmed with either a flag (75h) character for bit protocol mode, or a second synchronization character for character synchronous (two character) mode.

7201 Programming Example

Figure 2-5 below is a listing of the initialization sequence used for both Channels A and B of the 7201 on the CPU Board. The listing is annotated to show how registers used in Channels A and B are configured for cluster and keyboard communications, respectively. If more extensive information concerning the NEC 7201 (or Intel 8274) is required, see either the 1981 Catalog published by NEC Microcomputers, Inc., or the Component Data Catalog published by the Intel Corporation.

Baud Rate

The baud rate of data received on the cluster communications channel depends entirely on the rate at which the transmitting workstation is sending data. The limits to the clock speed are established by two factors: the upper limit that the 7201 accepts, and the lower limit that the carrier detector recognizes as a constant clock.

The carrier detector input of the 7201 is connected to a one-shot multivibrator triggered by the receive clock. The one-shot's output is asserted any time the clock has a rising edge. For the carrier detector to remain continuously asserted, the minimum clock rate is 100k baud. The maximum clock rate is 410k baud; this is a limitation of the 7201 when it is running with a 2.5-MHz system clock.

When the request-to-send line for Channel A is asserted, the CPU Board transmit clock is driven onto the clock line of the cluster. The frequency of the transmit clock is determined by the divisor that is programmed into Counter 1 of the 8253. The 8253 should be programmed in Mode 3 (square wave) as follows:

1. write 76h to Port 46h,

2. write the low-order byte of the divisor to Port 42h, and

3. write the high-order byte of the divisor to Port 42h.

Location	Object	Line	Source		Comments
			PUBLIC	Init7201	
			InitDevCode	SEGMENT PUBLIC 'COED'	
			ASSUME	CS: InitDevCode	
0000					
			Init7201	PROC FAR	
0000	B01B	+1	mov	al, 18h	
0002	E662	+1	out	62h, al	Channel A reset.
0004	90	+1	nop		
0005	90	+1	nop		
0006	90	+1	nop		
0007	90	+1	nop		
0008	B018	+1	mov	al, 18h	
000A	E662	+1	out	62h, al	Channel A reset.
000C	90	+1	nop		
000D	90	+1	nop		
000E	90	+1	nop		
000F	90	+1	nop		
0010	B004	+2	mov	al, 04h	
0012	E662	+1	out	62h, al	Select Port 62h, Write Register 4.
0014	B020	+2	mov	al, 20h	
0016	E662	+1	out	62h, al	Select bit protocol mode. All other bits 0.
0018	B001	+2	mov	al, 01h	
001A	E662	+1	out	62h, al	Select Port 62h, Write Register 1.
001C	B000	+2	mov	al, 00h	
0016	E662	+1	out	62h, al	All bits 0.

Figure 2-5. 7201 Programming Example. (Page 1 of 3)

Location	Object	Line	Source		Comments
0020	B002	43 +2	mov	al, 02h	
0022	E662	44 +1	out	62h, al	Select Port 62h, Write Register 2.
0024	B031	45 +2	mov	al, 31h	
0026	E662	46 +1	out	62h, al	Channel A DMA, Channel B interrupts; 8088 interrupt mode; priority Channel A over Channel B. All other bits 0.
		47 +1			
		48 +1			
0028	B003	49 +2	mov	al, 03h	
002A	E662	50 +1	out	62h, al	Select Port 62h, Write Register 3.
002C	B000	51 +2	mov	al, 00h	
002E	E662	52 +1	out	62h, al	All bits 0.
		53 +1			
		54 +1			
0030	B005	55 +2	mov	al, 05h	
0032	E662	56 +1	out	62h, al	Select Port 62h, Write Register 5.
0034	B000	57 +2	mov	al, 00h	
0036	E662	58 +1	out	62h, al	All bits 0.
		59 +1			
		60 +1			
0038	B018	61 +1	mov	al, 18h	
003A	E666	62 +1	out	66h, al	Channel B reset.
003C	90	63 +1	nop		
003D	90	64 +1	nop		
003E	90	65 +1	nop		
003F	90	66 +1	nop		
0040	B018	67 +1	mov	al, 18h	
0042	E666	68 +1	out	66h, al	Channel B reset.
0044	90	69 +1	nop		
0045	90	70 +1	nop		
0046	90	71 +1	nop		
0047	90	72 +1	nop		
		73 +1			
		74 +1			
0048	B004	75 +2	mov	al, 04h	
004A	E666	76 +1	out	66h, al	Select Port 66h, Write Register 4.

Figure 2-5. 7201 Programming Example. (Page 2 of 3)

Location	Object	Line	Source		Comments
004C	B0C4	77 +2	mov	al, C4h	
004E	E666	78 +1	out	66h, al	Asynchronous, 1 stop bit; 8 bits; divisor code, 64. All other bits 0.
		80 +1			
0050	B003	81 +2	mov	al, 03h	
0052	E666	82 +1	out	66h, al	Select Port 66h, Write Register 3.
0054	B0C1	83 +2	mov	al, C1h	
0056	E666	84 +1	out	66h, al	Length code, 8 bits; receive enable. All other bits 0.
		85 +1			
		86 +1			
0058	B001	87 +2	mov	al, 01h	
005A	E666	88 +1	out	66h, al	Select Port 66h, Write Register 1.
005C	B015	89 +2	mov	al, 15h	
005E	E666	90 +1	out	66h, al	External interrupt enable; no fixed vector; interrupt on all receive characters or special conditions and parity affects vector. All other bits 0.
		91 +1			
		92 +1			
0060	B002	93 +2	mov	al, 02h	
0062	E666	94 +1	out	66h, al	Select Port 66h, Write Register 2.
0064	B008	95 +2	mov	al, 08h	
0066	E666	96 +1	out	66h, al	Base vector, 08h.
		97 +1			
		98 +1			
0068	B005	99 +1	mov	al, 05h	
006A	E666	100 +1	out	66h, al	Select Port 66h, Write Register 5.
006C	B068	101 +2	mov	al, 68h	
006E	E666	102 +1	out	66h, al	Length code, 8 bits; transmit enable. All other bits 0.

Figure 2-5. 7201 Programming Example. (Page 3 of 3)

The 8253 then generates the clock. The clock frequency input to Counter 1 of the 8253 is 1.23 MHz. For example, to set 307 kilobaud (Convergent's standard cluster frequency), use the following sequence:

1. write 76h to Port 46h,

2. write 4h to Port 42h, and

3. write 0 to Port 42h.

To proceed from the current CS:IP, enter:

+P

To go from 1E43:90h, enter:

+1E43:90G

Firmware Functional Description

Cluster Protocol. Cluster architecture is discussed in detail in the System Programmer's Guide. The general protocol of the Operating System dictates that the master workstation poll the cluster workstations, and the master workstation and the cluster workstations exchange messages. The protocol requires that every workstation on the cluster communications line (or all workstations in a minicluster) have a unique workstation identification number.

The master workstation initiates all communications; a cluster workstation recognizes messages intended for it by the workstation identification number. The cluster workstation picks a workstation identification number by (1) monitoring the cluster communications line to find an unused one, (2) taking an unused number, and then (3) monitoring the cluster communications line again to see if it has collided with another workstation that may have picked the same number. If a collision occurs, both workstations wait a random time interval before restarting the search for another identification number.

The workstation identification number can be between 1 and 15, the largest number of cluster workstations allowed on a single cluster communications line. Identification numbers are not the same as user numbers. User numbers are assigned to cluster workstations by the Operating System of the master workstation and are different for every workstation in a cluster. Identification numbers are determined by the cluster workstations and are duplicated on the different cluster communications lines of a cluster, since a cluster can have up to four separate communications lines.

The actual protocol used for cluster communications is a subset of the American

National Standard for Advanced Data Communications Control Procedures (ADCCP), as defined in ANSI X3.66, published by the American National Standards Institute, Inc. The sequence used during the dump and bootstrap routines is illustrated in Figure 2-3 below. The protocol symbols are:

Symbol	Meaning
SNRM	Set Normal Response Mode
RIM	Request Initialization Mode
SIM	Set Initialization Mode
XID	Identification Frame (contains workstation type)
UP	Unnumbered Poll
UI	Unnumbered Data Frame
RD	Request Disconnect
DISC	Disconnect
UA	Unnumbered Acknowledge
UI'	Unnumbered Data Frame (with termination data)

Bootstrap Interface Block. When a program is loaded, and before execution is transferred to it, the bootstrap ROM places a 16-byte structure in memory with a pointer to it at location 1FCh. The structure of the bootstrap interface block is:

```
BootBlock:          RECORD
                    pProgramStart,
                    pCtosBuffer: POINTER;
                    DumpDevice,
                    BootDevice,
                    WsType,
                    DumpWsNumber/DumpErrorCode,
                    BootWsNumber,
                    unused,
                    unused,
                    unused: BYTE;
                    END;
```

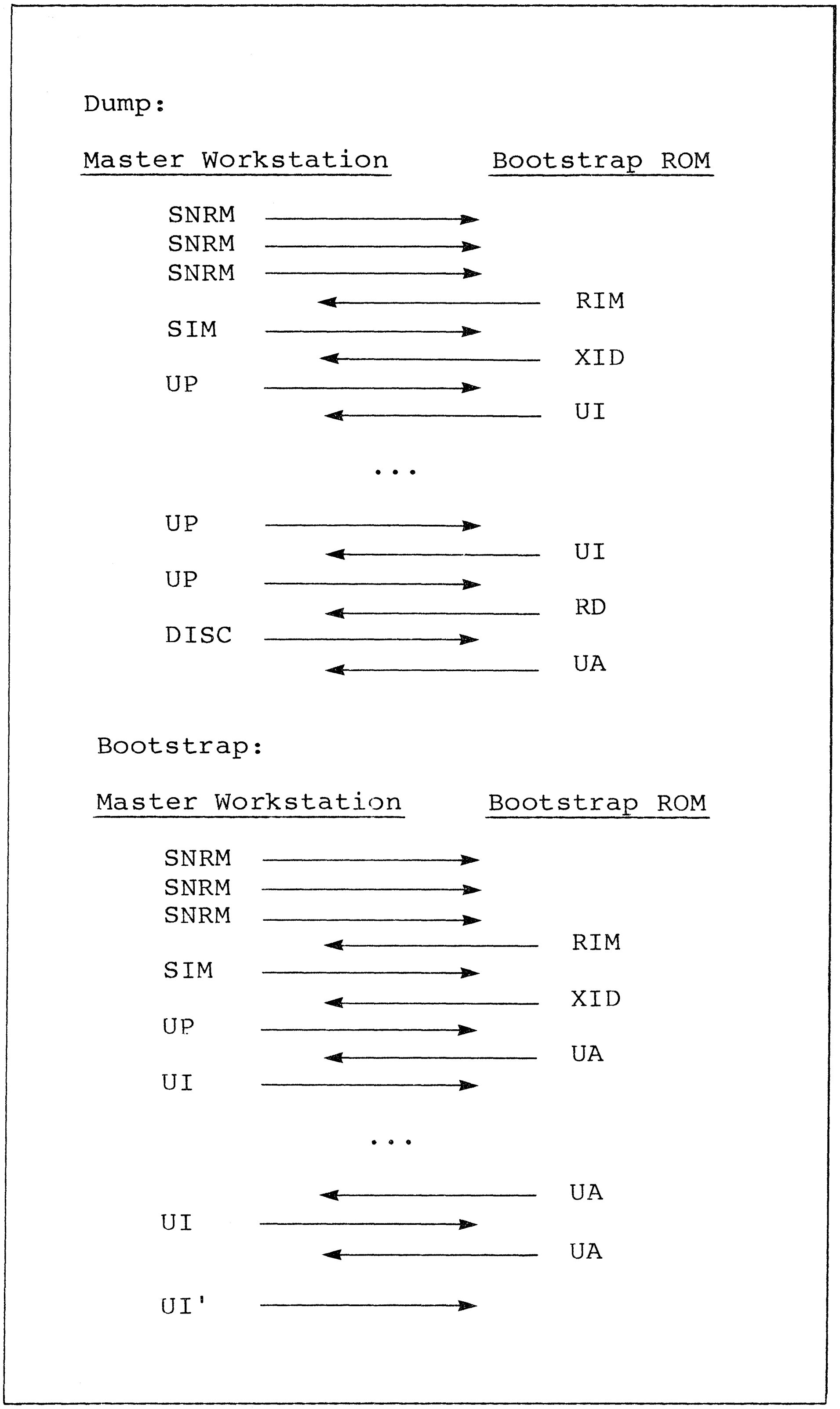

Figure 2-3. Communications Dump and Bootstrap Protocol.

The dump and bootstrap device numbers are either 0 (failed), 1 (floppy disk drive 0), 2 (hard disk drive, on the AWS-240 only) or 3 (communications line). The workstation type (WsType) is either 253, 254, or some other type selected using the T option on the menu (see Menu Mode above). Dump and bootstrap workstation identification numbers (WsNumbers) are those that the bootstrap ROM picked during the dump and/or the bootstrap. If the dump failed, the failing error code is saved in the DumpWsNumber field.

The CTOS Operating System also examines the ROM type number, which is a word (80h for the bootstrap ROM) found at location FFFF:0006h. Any application that wants to access this ROM type number must do so with parity disabled or a parity error results.

CTOS Operating System Buffer. The bootstrap ROM saves a 64-byte buffer for the CTOS Operating System. The CTOS Operating System has a pointer to the buffer at location 0000:0240h. The buffer must lie beyond the first 6 kilobytes of RAM, which are the bootstrap ROM's work area. The bootstrap ROM copies the buffer into its work area and sets the pointer (pCtosBuffer) to this copy in the Bootstrap Interface Block. The CTOS Operating System uses this pointer to recover the buffer after it has been bootstrapped.

Bootstrap Errors

When the workstation is bootstrapped, it goes through diagnostic and bootstrapping routines, which are resident in the ROM of the CPU. When an error is detected by the bootstrap ROM, the error code appears on the video display. For E0 and E1 error codes only, the audible alarm is cycled on and off five times and the error code appears on the keyboard LEDs.

Errors During Bootstrap or Dump. The communications bootstrap or dump routines do not stop to report an error if there is no activity on the RS-422 cluster communications line. This can occur when the cable to the master workstation is disconnected, when the master workstation crashes, or when the master workstation is disabled by the Disable Cluster utility. When the connection with the master workstation is reestablished, the bootstrap or dump routine automatically starts (indicated on the video display by a "." for every sector transferred).

Interpreting Keyboard Error Codes. The E0h and E1h error codes are displayed on the keyboard LEDs. They are interpreted as follows.

LED	Error E0h	Error E1h
OVERTYPE	on	on
LOCK	on	on
F1	on	on
F2	off	off
F3	off	off
F8	off	off
F9	off	off
F10	off	on

Error Codes. For most of the error codes listed in hexadecimal format below, there is also a list of possible causes for the error (listed with the most likely cause first).

AWS-220 and -230 Floppy Disk Drive Errors

Error Code	Message/Meaning/Possible Causes
00-02	Unused
03	Timeout waiting for an interrupt after a seek command. The floppy disk controller did not interrupt the CPU after being issued a seek command. Check: 1. that the operator did not open the door of the floppy disk drive, or 2. the seating of the FDC and CPU Boards on the Motherboard.

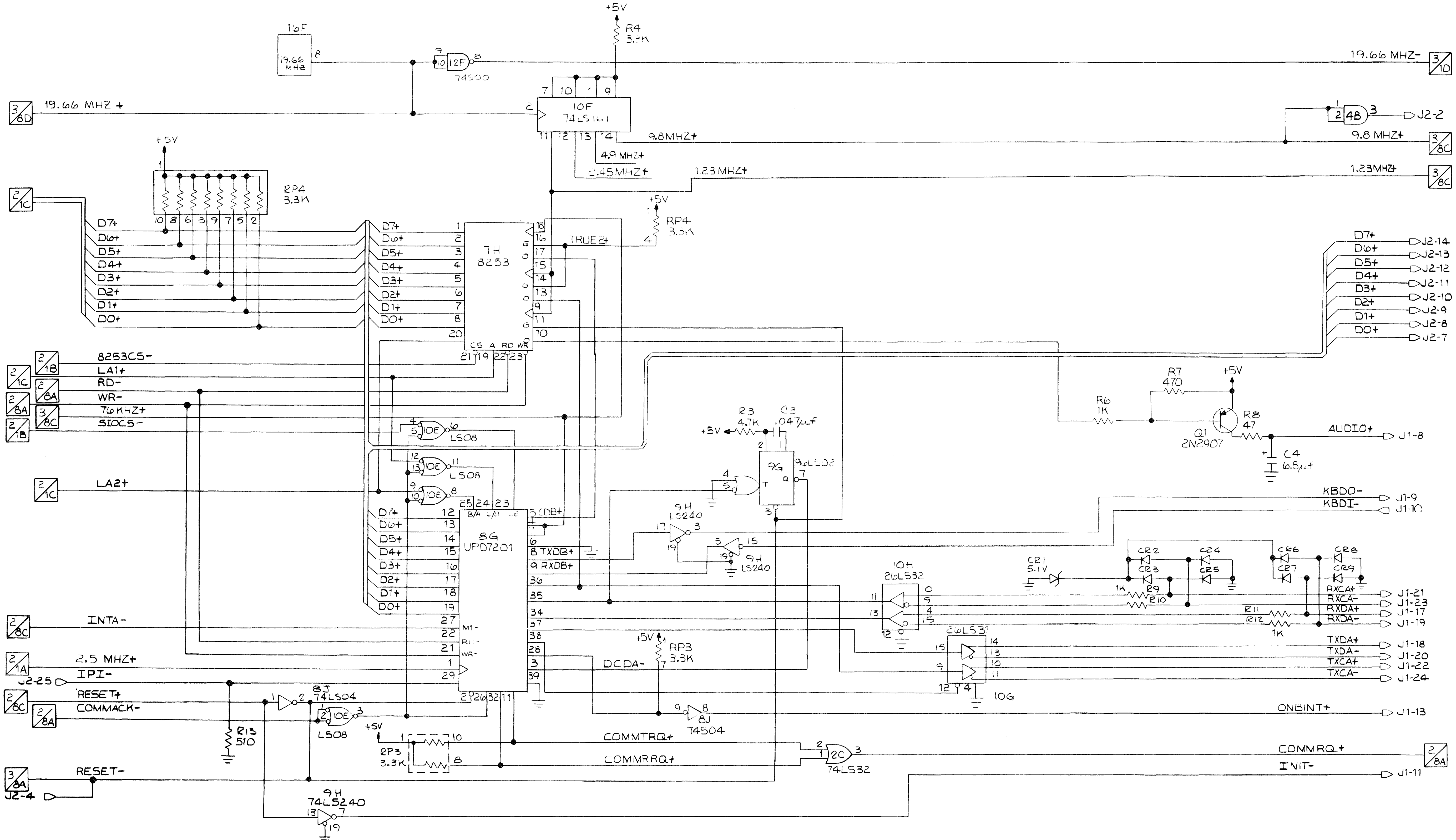

Figure 3-3. 8088 CPU Board Schematic. (Page 4 of 6)

Table D-8. I/O Extender Board Connectors.

Number	Board	Pins
P1	Main Motherboard	120
P2	Multibus Motherboard	120
P3	Main Motherboard	80
P4	Cluster Communications	9
P5	Cluster Communications	9
J1	Disk Controller Interface	50
J2	Printer	25
J3	Channel A	25
J4	Channel B	25

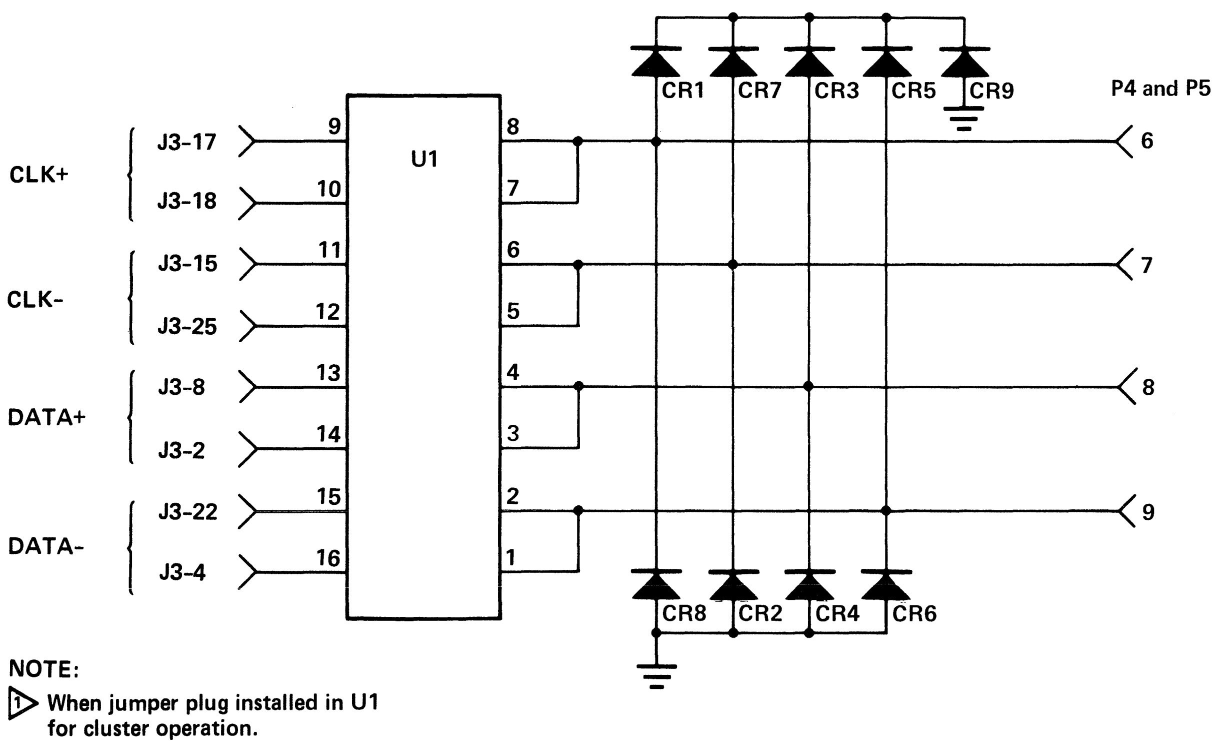

Figure D-14. I/O Extender Board U1 Mode Selector Switch Circuitry.

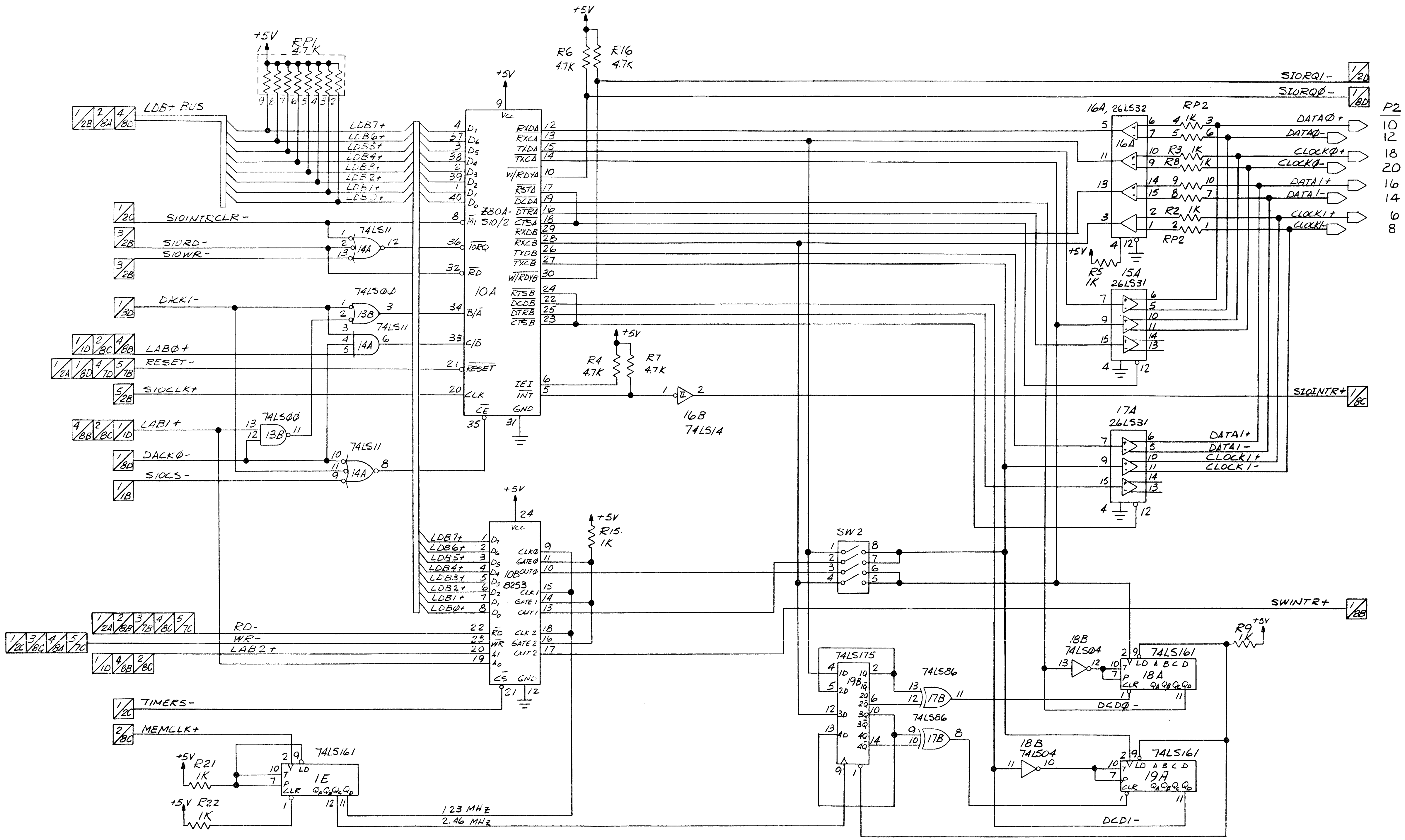

Figure 13-12 Communications I/O Processor Schematic
(Page 6 of 6)