Prime Computer Logic Diagrams

UNIT RECORD CONTROLLER (MPC)

PRINTER, C.R., C.P.

LOGIC DIAG. LDS1911

Prime Computer, Incorporated, 145 Pennsylvania Avenue, Framingham, Massachusetts 01701

TABLE OF CONTENTS

UNIT RECORD CONTROLLER (MPC)

PRINTER, C.R., C.P.

LOGIC DIAG. LDS1911

DWG. NO.	DATE	REV.
LDS1911	12-18-75	A

SECTION I

GENERAL DESCRIPTION

LINE PRINTER

The standard version line printer is a compact, high-speed unit capable of printing 300 lines per minute or, optionally, 600 lines per minute. The line printer accepts seven-bit ASCII characters that are stored in a 136-character line buffer. A hardware option that may be attached to the line printer will allow it to accept 9-bit ASCII characters (e.g., Upper and Lower case alphabetics). Paper and inked ribbon pass between a bank of print hammers and a rotating character drum containing alphanumeric characters per column (or 96 alphanumeric character optional drum). The hammer bank contains a hammer for every other print column and oscillates from odd-to-even numbered columns under control of a voice-coil-type actuator mechanism. Odd columns are printed during one revolution of the print drum, and even columns are printed on the next revolution. "Data Printer" printers that function differently from this are also offered to the user with special needs.

The line printer can produce an original and five clear carbon copies. Adjustable pin-feed form tractors can handle forms up to 16-3/4 inches. Table 1-1 summarizes the characteristics of the line printer and its controller.

Table 1-2 lists all related documents that the user may need to program or operate the controller and the devices connected to the controller.

Table 1-1. Line Printer and Controller Characteristics

Unit Type Number	31XX XX = 61 per 300 lpm printer and controller; XX = 65 for 600 lpm printer and controller; XX = 91 for printer, card reader and controller; XX = 95 for printer, card reader, punch, and controller
Speed:	300 lines per minute 600 lines per minute (optional).
Character Set:	64 alphanumeric plus blanks 96 alphanumeric plus blanks (optional).
Characters per line:	136
Print mechanism:	drum
Buffer:	full line (136 characters)
Character formation:	full character on drum
Vertical spacing:	6 lines per inch
Maximum forms width:	16-3/4 inches
Paper feed:	pin feed tractors

Table 1-2. Related Documents

Document Title	Order No.
Data Products 2420/2440/2470 Line Printer Technical Manual	
Volume 1	MAN 1927
Volume 2	MAN 1928
Logic Diagrams for Type 31XX Line Printer (Unit Record) Controller	LDS 9111
Microcode Listing for Line Printer Controller	MIC 1665
General Purpose Interface Manual	MAN 1676
Prime Installation and Maintenance Manual	MAN 1677
Prime Macro Assembler Manual	MAN 1673
Prime System Reference Manual	MAN 1671
Prime Software Library User Guide	MAN 1880
Prime Disk and Virtual Memory Operating System User Guide	MAN 1675
RTOS User Guide	MAN 1856
Card Reader/Punch User Guide	MAN 1941

SECTION 2

OPERATION

CONTROLS AND INDICATORS

Figure 2-1 shows the layout of a typical control panel for the line printer; Table 2-1 lists and describes the controls illustrated in Figure 2-1. The power circuit breaker, the PRINT INHIBIT switch, specific fault conditions that have no indicators, and paper adjustment controls are described in table in the vendor's manual (See Table 1-2).

OPERATING PROCEDURES

Controller Operation

No operator action is required.

Printer Operation

For complete and illustrated details in how to connect the printer to a power source, printer ribbon removal and installation, paper installation, and VFU tape loading; refer to the vendor's manual (Table 1-2).

On-Line Startup: To put the printer in the on-line mode of operation, proceed as follows:

Verify that the PRINT INHIBIT switch is off.

Set Power Circuit Breaker ON. After at least five seconds, check that POWER ON/OFF indicator and READY indicator are lit.

Press and release ON LINE switch/indicator to place printer in operation.

Check printer operation from time to time. Make paper adjustments as necessary (see the vendor's manual, Table 1-2).

Shutdown: The following steps place the printer in an off-line mode of operation:

Press and release the ON LINE sitch/indicator.

Check that indicator goes out after the printing of the current line has been completed.

Set Power circuit breaker OFF; check that the POWER ON and READY indicators go out.

Table 2-1. Controls and Indicators

Control and/or Indicator	Function
POWER ON	Indicates all DC voltages are within their tolerances, and the initial power up delay (approximately 4 seconds) is over.
ALARM Switch/Indicator	1. Indicates a fault condition exists. If the fault condition has an associated indicator, that indicator is lit in conjunction with the ALARM indicator. 2. The ALARM indicator also is lit when the PRINT INHIBIT switch is on, and the READY indicator is lit. Pressing switch/indicator will clear all printer logic.
READY Indicator	Indicates that all interlocks are satisfied, there is no fault condition, and the printer is ready to be put into the on-line mode.
ON/OFF LINE Switch/ Indicator	Indicates that printer is in either the on-line or off-line mode. Indicator is lit when the pritner is in the on-line mode and under control of the user. Pressing the switch/indicator alternately places the printer on-line or off-line. At initial power on, the indicator is out (off-line mode).
TOP OF FORM Switch	Advances paper to top of next form (page). This switch is disabled when printer is in the on-line mode.
PAPER STEP Switch	Advances paper to the next line. This switch is disabled when the printer is in the on-line mode.
Phasing control (located atop of logic board enclosure	An operator adjustment that varies ink density at top and bottom of a character. Adjustment should be made by factory trained personnel only.

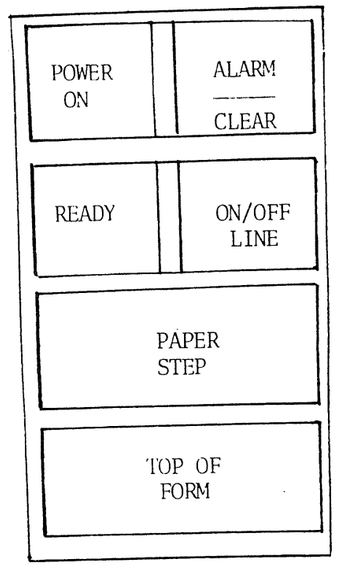


Figure 2-1. Typical Control Panel for Line Printer Connected to Unit Record Controller

SECTION 3

INSTALLATION

UNCRATING AND INSTALLATION

The applicable vendor's manual (Table 1-2) gives detailed instructions for uncrating and installing the printer (card reader, and/or card reader/punch). These instructions discuss installation of all standard and optional aspects of the peripheral devices described and must be followed wherever applicable. It is recommended that these sections of the vendor's manual be read and understood before attempting to uncrate and install a printer, a card reader, and/or a card/punch.

Space Requirements

Space requirements are described in detail in the applicable vendor's manual. However, it is emphasized that the length of cable to the line printer must be within a maximum length of 40 feet.

Printer Unpacking Instructions

Unpacking instructions for the line printer are outlined in the Printer vendor's manual (Table 1-2). After unpacking, be sure to perform the on-receipt inspection described in the Printer manual to ensure that the printer is ready for operation.

INTERCONNECTION AND CABLING

Table 3-1 shows the interface between the controller and the Prime computers for the printer, and shows the cross references for the Prime signals with the printer vendor signals.

Table 3-1. Line Printer Mnemonic Cross Reference Chart

CABLE, LINE PRINT, MPC CONN C

Signal Name			Prime		Line Printer	
	Prime	LBD	SIG	GND	SIG	GND
			1	2		
DATA1	BDC12-	31	3	4	B	D
DATA2	BDC13-	31	5	6	F	J
DATA3	BDC14-	31	7	8	L	N
DATA4	BDC15-	31	9	10	R	T
DATA5	BDC16-	31	11	12	V	X
DATA6	BDC17-	31	13	14	z	b
DATA7	BDC18-	31	15	16	n	k
			17	18		
			19	20		
			21	22		
			23	24		
			25	26		
CTL INTLK	BDC26-	31	27	28	v	
CHAR STROBE	BDC27-	31	29	30	j	m
PAPER INST	BDC28-	31	31	32		
ON LINE	BDC71-	27	33	34	y	AA
DEMAND	BDC72-	27	35	36	E	C
CBL INTLK	BDC72-	27	37	38	x	
			39	40		
			41	42		
			43	44		

SECTION 5

ASSEMBLY LANGUAGE PROGRAMMING

CONTROLLER FUNCTIONS

The principal functions of the unit record controller are:

- Transfer a line of data to the printer.
- Transfer vertical format information to the printer.
- Read a card in ASCII mode.*
- Read a card in binary mode.*
- Punch a card in ASCII mode.*
- Punch a card in binary mode.*

Secondary functions of the unit record controller include the transfer of the following information to the processor's A-Register:

- Printer status
- Printer vector address
- Controller channel number
- Card reader status*
- Card punch status*
- Card reader vector address*
- Card punch vector address*

DATA FORMATS AND CODES

Data transfers between the controller are by direct memory access or direct memory connect (DMA or DMC) only. Formats in memory for the peripheral devices line printer are described in the following paragraphs. Formats for the card reader and card reader punch are described in the applicable User Guide.

* A card reader or reader punch can also be connected to the Unit Record Controller. Card functions are referenced but not described in detail in this document. Refer to the Unit Record Controller for Card Reader & Card/Reader Punch User Guide.

Line Printer Data Formats

Memory Format: The line printer accepts up to 136 printable ASCII characters per line. Figure 5-1 shows the mapping between data in memory and data printed. Unless the drum is equipped with the 96-character option, memory bits 1 and 9 are ignored by the controller.

Printing Codes: Printing codes are shown in Table 5-1 for a line printer equipped with a 64-character drum. Printing codes for a line printer equipped with a 96-character drum are shown in Table 5-2. Note that the standard Prime software (DOS, DOS/VM, Editor, etc.) support the upper/lower case character set (96-character drum) as well as the 64-character drum. In both Tables 5-1 and 5-2, b_7 through b_1 refer to memory bits 2 through 8 for characters in odd print columns and memory bits 10 through 16 for characters in even print columns. The control codes PF, FF, and CR shown in Tables 5-1 and 5-2 cause the following actions:

Code	Action
PF	Print all characters received since the last control code and then advance paper one line.
FF	Print all characters received since the last control code and then advance paper to the top of the next form.
CR	Print all characters received since the last control code. The controller sends a CR code to the printer if the DMX end-of-range will be sent to the printer first.

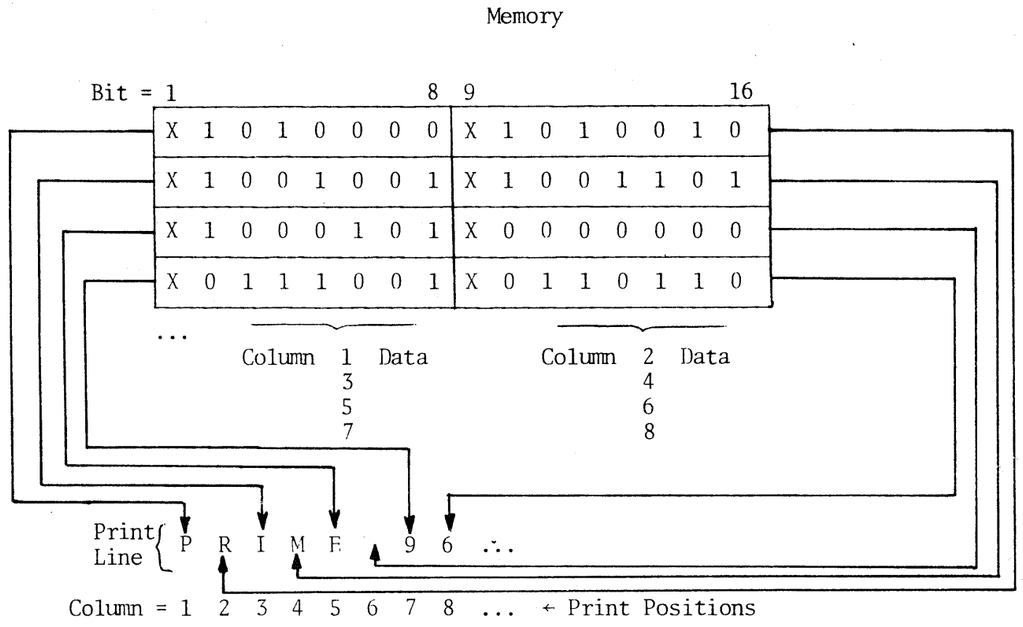
Card Reader/Card Punch Formats

For a description of card formats, refer to the Unit Record Controller for Card Reader and Card Reader Punch User Guide.

CONTROLLER ADDRESS

The controller provides the interface between the Prime I/O Bus and one line printer, one card reader, and one card punch. Any combination of these peripheral devices may be used.

The controller address is $(\emptyset 3)_8$.



Note: Each word in the print buffer contains data for two print positions (columns), the first byte (bits 1 through 8) contains data for the odd numbered print column; the second byte (bits 9 through 16) contains data for the even numbered print column. Bits 1 and 9 are ignored by any controller that has standard features.

Figure 5-1. Memory Buffer Data to Print Line Data Mapping

Table 5-1. Standard 64-Character (ASCII) Set

b_7				0	0	1	1	0	1	0	1
		b_6		0	0	1	1	0	0	1	1
		b_5		0	1	0	1	0	1	0	1
b4	b3	b2	b1								
0	0	0	0		Space	0	@	P			
0	0	0	1		!	1	A	Q			
0	0	1	0		"	2	B	R			
0	0	1	1		#	3	C	S			
0	1	0	0		$	4	D	T			
0	1	0	1		%	5	E	U			
0	1	1	0		&	6	F	V			
0	1	1	1		'	7	G	W			
1	0	0	0		(	8	H	X			
1	0	0	1		)	9	I	Y			
1	0	1	0	PF	*	:	J	Z			
1	0	1	1		+	;	K	[			
1	1	0	0	FF	,	<	L	\			
1	1	0	1	CR	-	=	M	]			
1	1	1	0		.	>	N	↑			
1	1	1	1		/	?	O	←			

All other codes result in the printing of a space

Table 5-2. 96-Character (ASCII) Set

b^7 b^6 b^5 / b^4 b^3 b^2 b^1	0 0 0	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
0 0 0 0		Space	0	@	P	\	p
0 0 0 1		!	1	A	Q	a	q
0 0 1 0		"	2	B	R	b	r
0 0 1 1		#	3	C	S	c	s
0 1 0 0		$	4	D	T	d	t
0 1 0 1		%	5	E	U	e	u
0 1 1 0		&	6	F	V	f	v
0 1 1 1		'	7	G	W	g	w
1 0 0 0		(	8	H	X	h	x
1 0 0 1		)	9	I	Y	i	y
1 0 1 0	PF	*	:	J	Z	j	z
1 0 1 1		+	;	K	[	k	{
1 1 0 0	FF	´	<	L	\	l	|
1 1 0 1	CR	-	=	M	]	m	}
1 1 1 0		.	>	N	↑	n	¯
1 1 1 1		/	?	O	←	o	▭

All other codes result in the printing of a space

PROGRAMMED INPUT/OUTPUT (PIO) COMMANDS

Table 5-3 shows the PIO commands accepted by the unit record controller. Control of each type of peripheral device is by setup words addressed to the particular device. The setup words are described in the following paragraphs.

OTA 01 Setup Printer

There are three operations connected with the printer that are controlled by this setup word. They are:

Print a line

Printer vertical format/line space operation

Setup controller for input to A register (INA)

The A-register bits received by the controller as a consequence of OTA 01 are decoded as follows:

Bits	Interpretation
1	If this bit is set, bits 15 and 16 are decoded to define a particular INA required.
2	If this bit is set and bit 1 is not set, information from memory is transferred to the printer via a DMX channel.
3	If this bit is set and bits 1 and 2 are not set, A register bits 10 to 16 are sent to the printer as VFU or line space information. See the description of Bits 10-16.
15, 16	If bit 1 is set, Bits 15 and 16 are decoded and the following action takes place in preparation for an INA instruction that follows this occurrence of OTA 01.

Value of Bits 15,16	Action
00	Load data register with printer status.
01	Load data register with printer vector address.
10	Load data register with controller channel number.

Bits 10-16	
	If bit 3 is set, Bits 10 to 16 are sent to the printer to line space or control the optional vertical format unit. Three types of paper movement operations are possible:

Table 5-3. PIO Commands

Function Code Bits 7-10 \ Op Code Bits 1-6	14_8 (OCP)	34_8 (SKS) (SKIP IF)	54_8 (INA)	74_8 (OTA)
00		Ready	Input data register	
01		Not Busy		Setup Printer
02				Setup Card Reader*
03				Setup Card Punch*
04		Controller Not Interrupting		
05		Printer Not Interrupting		
06		Card Reader Not Interrupting*		
07		Card Punch Not Interrupting*		
10				
11				
12				
13				
14	Acknowledge Interrupt			DMA/C Channel No.
15	Set Int. Mask			Printer Vector Address
16	Clear Int Mask			Card Reader Vector Address*
17	Initialize			Card Punch Vector Address*

*Refer to the Unit Record Controller for Card Reader & Card Reader/Punch User Guide.

(1) If A register bits 10 and 12 are set, up to 15 lines are spaced in accordance with information given in Table 5-4; (2) If A register bit 10 is set and bit 12 is reset, a particular tape channel is selected in accordance with Table 5-5. The paper spaces until the next hole in that channel is detected; (3) If A register bits 10 - 16 are set to the codes for PF or FF (as defined in Tables 5-1 and 5-2), a PF or a FF operation takes place.

OTA 02 Setup Card Reader

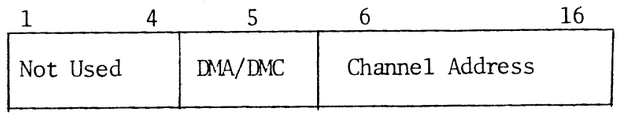
Refer to the Card Reader User Guide (Table 1-2).

OTA 03 Setup Card Punch

Refer to the Card Reader/Punch User Guide (Table 1-2).

OTA 14 DMA/C Channel Number

The channel number is considered to belong to the controller. Hence, the controller has only one register to hold the channel number rather than having one for each of three separate devices. This register is loaded by OTA 14 and the format for the OTA 14 is as follows:

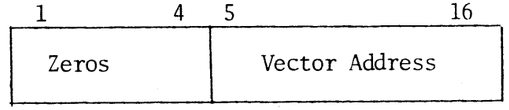


Bit 5 = 1 for DMC transfer
Bit 5 = 0 for DMA transfer

OTA 15, 16, 17 Interrupt Vector Address

OTA 15, OTA 16 and OTA 17 output to the controller the interrupt vector address for, respectively: the printer; the card reader; and the card punch.

The format for these instructions (OTA 15, OTA 16, OTA 17) is as follows:



If the vector address is not specified, the controller assumes the following addresses and interrupts via these locations:

Printer	$(103)_8$
Card reader	$(105)_8$
Card punch	$(106)_8$

Table 5-4. Line Spacing Information

AC BITS							No. of Lines Spaced
10	11	12	13	14	15	16	
1	X	1	0	0	0	0	0
1	X	1	0	0	0	1	1
1	X	1	0	0	1	0	2
1	X	1	0	0	1	1	3
1	X	1	0	1	0	0	4
1	X	1	0	1	0	1	5
1	X	1	0	1	1	0	6
1	X	1	0	1	1	1	7
1	X	1	1	0	0	0	8
1	X	1	1	0	0	1	9
1	X	1	1	0	1	0	10
1	X	1	1	0	1	1	11
1	X	1	1	1	0	0	12
1	X	1	1	1	0	1	13
1	X	1	1	1	1	0	14
1	X	1	1	1	1	1	15

Table 5-5. Tape Channel Selection

AC BITS							Tape Channel
10	11	12	13	14	15	16	
1	X	0	0	0	0	0	0
1	X	0	0	0	0	1	1
1	X	0	0	0	1	0	2
1	X	0	0	0	1	1	3
1	X	0	0	1	0	0	4
1	X	0	0	1	0	1	5
1	X	0	0	1	1	0	6
1	X	0	0	1	1	1	7
1	X	0	1	0	0	0	8
1	X	0	1	0	0	1	9
1	X	0	1	0	1	0	10
1	X	0	1	0	1	1	11

Following an OTA 15 or OTA 16 or OTA 17, the controller maintains the specified interrupt address until initialized. Following initialization, the vector addresses revert to the standard controller address.

INA 00 Input Data Register

The INA 00 instruction transfers the contents of the controller's data register to the processor. The data register normally is loaded by an OTA 01, OTA 02, or OTA 03 instruction and the appropriate setup word (described in the paragraphs that discuss these OTA instructions).

SKS 00 Skip if Ready

The controller sets READY when it has loaded the data register with information that is to be transferred to the processor. The controller expects the processor to issue an INA 00 to perform the information transfer. The INA also resets READY.

SKS 01 Skip if Not Busy

The controller becomes BUSY on receipt of any OTA instruction and remains BUSY until completion of that OTA.

SKS 04 Skip if Controller Not Interrupting

The controller initiates an interrupt under the following circumstances:

The line printer has completed the printing of a line or a paper format operation

The complete data buffer has been transferred to the line printer

Paper movement information (specified by Bit 3 of the OTA setup instruction) has been sent to the printer.

The card reader has transferred 80 columns of data to the controller.*

The complete data buffer (up to 80 columns worth) has been transferred to the card punch.*

SKS 05, 06, 07 Skip if Device Not Interrupting

The SKS 05, SKS 06, and SKS 07 commands are defined as follows:

SKS 05 - Skip if printer is not interrupting

SKS 06 - Skip if card reader is not interrupting *

SKS 07 - Skip if card punch is not interrupting *

* Refer to the Unit Record Controller for Card Reader/Punch User Guide

OCP 14 Acknowledge Interrupt

The OCP 14 instruction must be given following servicing of an interrupt to clear the interrupt request.

If the controller is being serviced without use of interrupts, the controller attempts to interrupt under the circumstances detailed in the description of SKS 04. The interrupt is not recognized by the processor until the controller mask is set and central processor interrupts are enabled. Therefore, it is recommended that this OCP be issued prior to the next time central processor interrupts are enabled, to ensure the controller cannot interrupt falsely.

STATUS WORDS

Status words are concerned with the operation of each peripheral device that is connected to the unit record controller and are transferred to the processor's A-register in response to specific setup information received by the controller during OTA 01, OTA 02, or OTA 03 instruction execution.

Printer Status Word

Definition of the bits of the printer status word are:

Bits	Definition
1-8	Not Used
9	Printer on-line
10	Printer not busy - the printer is able to receive data for a new line or another format command.
11-16	Not used

Card Reader Status Word and Card Reader/Punch Status Words

For a description of these status words, refer to the Unit Record Controller for Card Reader & Card Reader/Punch User Guide.

TIMING DETAILS

Printer Timing

The line printer contains a buffer that accepts a full line of data. Transfer of the data to the buffer is asynchronous and print speed does not slow down, provided the complete line of data is transferred within 2.5 milliseconds of the interrupt that signals that the previous line has been printed.

The maximum rate of transfer of a 136-character line plus a control character is 2.4 milliseconds. This corresponds to one DMX request every 35 microseconds.

Card Reader and Card Reader/Punch

Refer to the Unit Record Controller for Card Reader and Card Reader/Punch User Guide.

Controller Priority

The priority of the unit record controller with respect to others in the system depends on the combination of peripheral devices that the unit record controller is servicing. Table 5-6 shows the time between DMX accesses and the time to honor a DMX request for each of the three peripherals. (Card readers times are included here for convenience of the reader; they refer to the standard card reader model, which reads 285 cards per minute.)

Time to Complete Instructions

The non-peripheral OTA instructions, such as vector address; setup status; etc., require seven microseconds to complete their operation, and the unit record controller will be busy for this time.

An OTA print instruction to a non-busy line printer takes 2-4 milliseconds to complete for a full length line of print, assuming controller operation is not slowed by DMX transfers. The controller will be busy during the execution of the OTA print. Other than these 2-4 milliseconds, the controller can perform other tasks while the line is being printed. Printing of a line takes 200 milliseconds for the 64-character drum.

Card reading and punching OTA timing constraints are discussed in the appropriate user guide (MAN1941).

Controller Concurrency

The unit record controller performs only one OTA at a time. For instance, if a controller is involved in sending data to a line printer, it cannot be awaiting data read from a card. However, if three devices are connected; some degree of concurrency is possible between any two of the three peripheral devices that are connected to the unit record controller. For further information, refer to the following paragraphs.

Table 5-6. Unit Record Controller Priorities

Device	Time between DMX Requests (µs)	Maximum Time to Honor Requests (µs)
Printer	35	None, except print rate will decrease
Card Reader, binary	2000	1500
Card Reader, ASCII	4000	1500
Card Punch, binary	6250	3500
Card Punch, ASCII	12500	3500

Reading Cards and Printing

The printer consumes a line of data and prints without involvement from the controller. On the other hand, the card reader requires constant attention from the controller while a card is passing over the read head, but it requires no attention while the card is being stacked. During this stacking interval, data can be sent to the printer. Table 5-7 shows how nominal device speeds can drop when the printer and a card reader are run concurrently. As shown in Table 5-7, the best performance is obtained when the speeds of the two devices are equal.

Punching Cards and Printing

The controller is devoted to servicing the card punch for a time proportional to the number of columns punched. While the card is being stacked, the line printer may be serviced. Table 5-8 shows how print speed drops as the number of card columns punched increases. (It is assumed that the punch is running at maximum speed.)

Table 5-7. Comparison of Printer Speed when Run Concurrently with Card Reader.

	Stand Alone Speed			Concurrent Operation Speed	
	Printer (lpm)	Card (cpm)	Reader Model	Printer (lpm)	Card Reader cpm
	300	150	D150	155	150
	300	285	M200	300	285
	300	300	M300L	300	300
(a)*	300	600	M600L	300	300
(b)*	300	600	M600L	200	600
	600	150	D150	155	150
	600	285	M200	600	285
	600	300	M300L	400	300
	600	600	M600L	600	600

* The designations (a) and (b) refer to alternate choices of either sacrificing card reading speed or sacrificing print speed.

Table 5-8. Print Speed When Running Concurrently with Punch

Print Speed (lpm)	No. of Columns Punched	Punch Speed (cpm)
300	11	300
160	40	160
101	80	101

Example

The following program, PRNT, is a simple line printer driver that prints a previously specified line of text at the printer and halts.

This sample runs under a stand-alone DOS system. PRNT is bound with an associated program, CALL, that is provided to allow the printer driver to be called under DOS. Usage is (user input is underlined):

```
OK: LOAD
$ LO B←PRNT
$ LO B←CALL
$ LIB
LC
$ SA *PRNT
OK: R *PRNT
```

```
                    (0001) *      SAMPLE STAND ALONE MPC PRINTER DRIVER
                    (0002) *
                    (0003) *       CALL PRNTR(UNIT,BUFFER ADDRESS,WORD COUNT,INSTRUCTION,
                    (0004) *                  STATUS VECTOR)
                    (0005) *
            000000  (0006)         SUBR   PRNTR
                    (0007)         REL
                    (0008)         C64R
            000040  (0009) DMC     EQU    '40             USE DMC CHANNEL
                    (0010) *
                    (0011) *       INSTRUCTION DEFINITIONS
                    (0012) *
            000003  (0013) D       EQU    'D3             DEVICE ADDRESS
                    (0014) *
            000103  (0015) XNBUSY  EQU    'D100+D         SKIP IF NOT BUSY
            000003  (0016) XSTAT   EQU    'D000+D         INA STATUS
            000103  (0017) XPR     EQU    'D100+D         PRINTER SETUP
            001403  (0018) XDMXCH  EQU    'D400+D         DMX CHANNEL
                    (0019) *
000000: 00.000000A  (0020) PRNTR   DAC    **
000001: 10.000000E  (0021)         CALL FSAT
000002:    000005   (0022)         DEC    5
000003:    000000   (0023) UNIT    OCT    0               UNIT NUMBER
000004:    000000   (0024) BA      OCT    0               BUFFER ADDRESS
000005:    000000   (0025) N       OCT    0               WORD COUNT
000006:    000000   (0026) INST    OCT    0               INSTRUCTION WORD
000007:    000000   (0027) STATV   OCT    0               STATUS VECTOR
                    (0028) *
000010: 42.000006   (0029) WAIT    LDA*   INST            GET INSTRUCTION
000011:    140320   (0030)         CSA
000012: 02.000047   (0031)         LDA    ='100000
000013:    170103   (0032)         OTA    XPR
000014: 01.000013   (0033)         JMP    *-1
000015:    130003   (0034)         INA    XSTAT           READ PRINTER STATUS
000016: 01.000015   (0035)         JMP    *-1
000017: 35.000007   (0036)         LDX    STATV
000020: 24.00000lA  (0037)         STA    1,1             STORE STATUS
000021:    100001   (0038)         SRC
```

SAMPLE STAND ALONE MPC PRINTER DRIVER

```
000022: 41.000000   (0039)         JMP*   PRNTR           IF STATUS, JUST RETURN
000023:    100270   (0040)         SAR    9               IS PRINTER ON-LINE
000024:    101271   (0041)         SAS    10              IS PRINTER BUSY
000025: 01.000010   (0042)         JMP    WAIT            YES, WAIT
                    (0043) *
                    (0044) *    SETUP DMC CHANNEL
000026: 42.000005   (0045)         LDA*   N
000027:    101040   (0046)         SNZ                    CHECK WORD COUNT ZERO
000030: 01.000041   (0047)         JMP    INSTR           YES
000031: 46.000004   (0048)         ADD*   BA
000032:    140110   (0049)         S1A
000033: 04.000041A  (0050)         STA    DMC+1
000034: 42.000004   (0051)         LDA*   BA
000035: 04.000040A  (0052)         STA    DMC
000036: 02.000050   (0053)         LDA    =DMC+'4000
000037:    171403   (0054)         OTA    XDMXCH
000040: 01.000037   (0055)         JMP    *-1
                    (0056) *
                    (0057) *    EXECUTE INSTRUCTION
000041: 42.000006   (0058) INSTR   LDA*   INST
000042:    170103   (0059)         OTA    XPR
000043: 01.000042   (0060)         JMP    *-1
                    (0061) *
                    (0062) *    WAIT HERE UNTIL CONTROLLER GOES NON-BUSY
000044:    070103   (0063)         SKS    XNBUSY
000045: 01.000044   (0064)         JMP    *-1
000046: 41.000000   (0065)         JMP*   PRNTR
            000047  (0066)         END

000047: 00.100000A
000050: 00.004040A
```

SAMPLE STAND ALONE MPC PRINTER DRIVER

```
BA          000004  0024  0048  0051
D           000003A 0013  0015  0016  0017  0018
DMC         000040A 0009  0050  0052  0053
INST        000006  0026  0029  0058
INSTR       000041  0047  0058
N           000005  0025  0045
PRNTR       000000  0020  0039  0065
STATV       000007  0027  0036
UNIT        000003  0023
WAIT        000010  0029  0042
XDMXCH      001403A 0018  0054
XNBUSY      000103A 0015  0063
XPR         000103A 0017  0032  0059
XSTAT       000003A 0016  0034
```

0000 ERRORS (PMA-1080-016)

SAMPLE PRINTER DRIVER TEST

```
                    (0001) *      SAMPLE PRINTER DRIVER TEST
                    (0002) *
            001000  (0003)         ORG    '1000
001000:             (0004) STRT    ELM
001001: 10.000000E  (0005)         CALL   PRNTR
001002: 00.001027A  (0006)         DAC    UNIT
001003: 00.001012A  (0007)         DAC    SAMP
001004: 00.001030A  (0008)         DAC    =1†
001005: 00.001031A  (0009)         DAC    ='40000
001006: 00.001025A  (0010)         DAC    STAT
001007:    000000   (0011)         OCT    0
001010:    000000   (0012)         HLT
                    (0013) *
001011: 01.001000A  (0014)         JMP    STRT
001012:    151701   (0015) SAMP    BCI    11,SAMPLE PRINTER DRIVER
001013:    146720
001014:    146305
001015:    120320
001016:    151311
001017:    147324
001020:    142722
001021:    120304
001022:    151311
001023:    153305
001024:    151240
001025:    120240   (0016) STAT    BCI    2
001026:    120240
001027:    000000   (0017) UNIT    OCT    0
001030: 00.000013A  (0018)         FIN
001031: 00.040000A

            001032  (0019)         END    STRT
```

SECTION 6

TEST AND MAINTENANCE

TESTS

The Prime test program, DPCARD, is a verification program that exercises (1) the line-printer; (2) the card reader; and (3) the card read/punch, appropriately as they are configured to a given computer system. Setting sense switches 14, 15 and 16 specify the devices to be tested by DPCARD, and DPCARD branches to the appropriate verification sub-routine, or subroutines.

Minimum hardware configuration to run the test program, DPCARD, is a Prime CPU, a unit record controller, Teletype, and at least 8K of memory.

OPERATION OF DPCARD

The DPCARD program starts at location '1000; the unit record equipment (line printer, card reader, card reader/punch) must be powered on and ready for operation. (Refer to Section 2).

The user must set sense switches (SS) as specified in Table 6-1.

The Test DPCARD is supplied in the UFD named T & M of the master disk Volume 1 or on paper tape (refer to Table 6-2 for further details).

Functional Description of DPCARD

The DPCARD program is divided into five principal sections:

1. Testing of OTA/INA vector addresses and OTA/INA channel number.

2. A routine (named TESTSW) that tests sense switches 14, 15 and 16 sequentially and makes an exit to printer, reader, and punch subroutines as appropriate.

3. A printer subroutine (named PRINT) that prints a line from the appropriate buffer, or that performs a spacing operation if Sense Switches SS4 or SS5 are set.

4. A card reader (named CARDR) subroutine that reads one card, and checks the data of SS7 is set.

5. A card punch subroutine (named CARDP) that punches one card from the appropriate buffer is selected by SS9.

Table 6-1. Sense Switch Settings for DPCARD Test Program

Sense Switch	Position	Meaning
SS1	UP	Select 96-character font for line printer
SS1	DOWN	Select 64-character font for line printer
SS2	UP	Ripple printed data across page
SS2	DOWN	Print across page from Column 1
SS3	UP	Print card reader buffer
SS3	DOWN	Print ASCII buffer
SS4	UP	Do VFU operation, instead of printing
SS4	DOWN	Print instead of VFU operation
SS5	UP	136-column paper
SS5	DOWN	100-column paper
SS6	UP	Read cards in binary mode
SS6	DOWN	Read cards in ASCII mode
SS7	UP	Read non-standard cards (data is not checked)
SS7	DOWN	Read standard ASCII punched cards
SS8	UP	Punch data buffer in binary mode
SS8	DOWN	Punch data buffer in ASCII mode
SS9	UP	Rotate data buffer before punching
SS9	DOWN	No rotate of data buffer before punching
SS11	UP	Halt on error conditions
SS12	UP	Interrupt is in vectored mode
SS12	DOWN	Interrupt is in standard mode
SS13	UP	Enable machine-check mode
SS14	UP	Run line printer
SS15	UP	Run card reader
SS16	UP	Run card punch

Table 6-2. Loading Information for DPCARD

Filename in T & M UFD	Tests	Paper Tape	Low	High	Start
DPCARD	Line Printer Card Reader Card Reader/Punch	SLT0730.002	'66	'3257	'1000

The exit from the subroutines PRINT, CARDR, CARDP, described in items 3, 4, and 5 in the foregoing list of functions, is back to the TESTSW routine described in item 2. The TESTSW routine exits as appropriate to test the next I/O device.

DCARD Error Messages

The following messages may be generated by DCARD:

CARD READER OFF-LINE

CARD READER DMX OVERFLOW

CARD READER HOPPER, MOTION OR READ CHECK ERROR

CARD PUNCH OFF-LINE

CARD PUNCH DMX OVERRUN

CARD PUNCH MOTION OR PUNCH ERROR

PRINTER OFF-LINE

CARD READER READ ERROR

CARD READER ILLEGAL ASCII CODE

PREVENTIVE MAINTENANCE

Testing and maintenance of the line printer that is connected to the unit record controller is outlined in the vendor's manual (Table 1-2). Users are advised to perform preventive maintenance on a regular schedule in the manner recommended by the vendor's manual.

SECTION 1

GENERAL DESCRIPTION

PUNCHED CARD UNITS

The Unit Record Controller supports two types of punched card units: a 300 cpm reader and a reader/punch that reads 400 cpm and punches 285 cpm. Either card unit or a combination of one of the card units and a line printer can be handled by a single controller (See Figure 1-1). The reader and punch mechanisms process cards column-by-column. The controller performs conversion between internal machine language ASCII code and 12-bit Hollerith card code. Cards in binary format are also processed by the controller. Table 1-1 summarizes card reader and card reader/punch characteristics.

The card reader (Figure 1-1) reads standard 12-row, 80 column punched cards. The input hopper holds approximately 550 cards. The cards are stacked in the output hopper in the same order as they were originally placed into the reader.

Table 1-1. Card Reader and Card Reader/Punch Characteristics

	Reader	Reader/Punch
Unit Type Number: (includes unit and controller)	3141 reader and controller; 3191 reader, line printer and controller.	3181 reader/punch and controller; 3195 reader/punch, line printer and controller.
Peak speed:	300 cpm	read: 400 cpm; punch: 285 cpm.
Card format:	80-column ANSII Std	80-column ANSII Std
Input Hopper Capacity:	1000 cards	1000 cards
Output Stacker Capacity:	1000 cards	1000 cards
Picker:	friction	friction
Transport:	roller	roller
Read/Punch orientation:	column-by-column	column-by-column
Mounting	table top	free standing

Table 1-2 lists all related documents that the user may need to program or operate the controller and the devices connected to the controller.

SECTION 2

OPERATION

CARD READER OPERATION

Figure 2-1 shows a typical control panel for the 300 cpm card reader that may be connected to the unit record controller. Figure 2-2 shows a flow chart of the sequence of events that may be encountered in operating the card reader. For detailed operating procedures, refer to the vendor's manual.

In the event of trouble in card reading operation, refer to the trouble-shooting chart in Section 8 of the vendor's manual.

Switches and Indicators

On the front panel (Figure 2-1) are two push button switches, STOP and RESET. Above these switches show reader status; they are RESET (green) and STOP (red).

Five other indicators are located on the front panel. They are: POWER, READ CHECK, PICK CHECK, STACK CHECK and HOPPER CHECK.

At the rear of the reader are two mode switches, a TEST switch and the main AC power circuit breaker (See Figure 2-1).

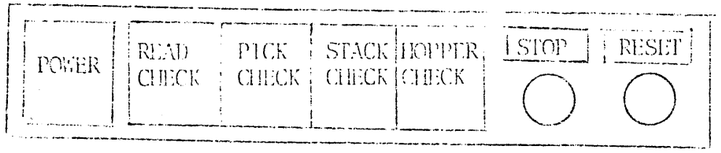
Operational Procedures

These procedures are described in detail in the vendor's manual. Briefly, the procedures to operate the card reader are:

1. Select MANUAL or AUTO mode of operation.
2. Select either LOCAL or REMOTE operation.
3. Place power circuit breaker ON.
4. Press LAMP TEST switch and check that all front panel indicators are lit.
5. Load input hopper and press RESET.
6. PICK CHECK indicator lights if card fails to reach read head.
7. READ CHECK indicator lights and card reader stops if any of the the following conditions occur:
   a. Failure of leading or trailing edge dark check.
   b. Failure of trailing edge light check.
   c. Card stoppage.
   d. Control logic failure.
8. STACK CHECK indicator lights if the card, previously read, has not reached the output stacker.
9. The HOPPER CHECK indicator lights when the input hopper is empty.
10. The STOP push button momentarily terminates card reader operation at the end of a read cycle.

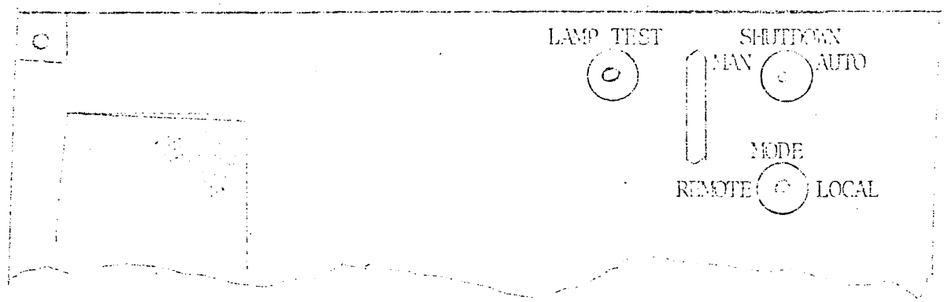
CARD READER/PUNCH OPERATION

The operation of the card reader/punch that may be connected to the unit record controller is described in detail in the vendor's manual.



FRONT CONTROL PANEL

AC POWER



REAR PANEL

Figure 2-1. Card Reader Control Panel, Switch and Indicators

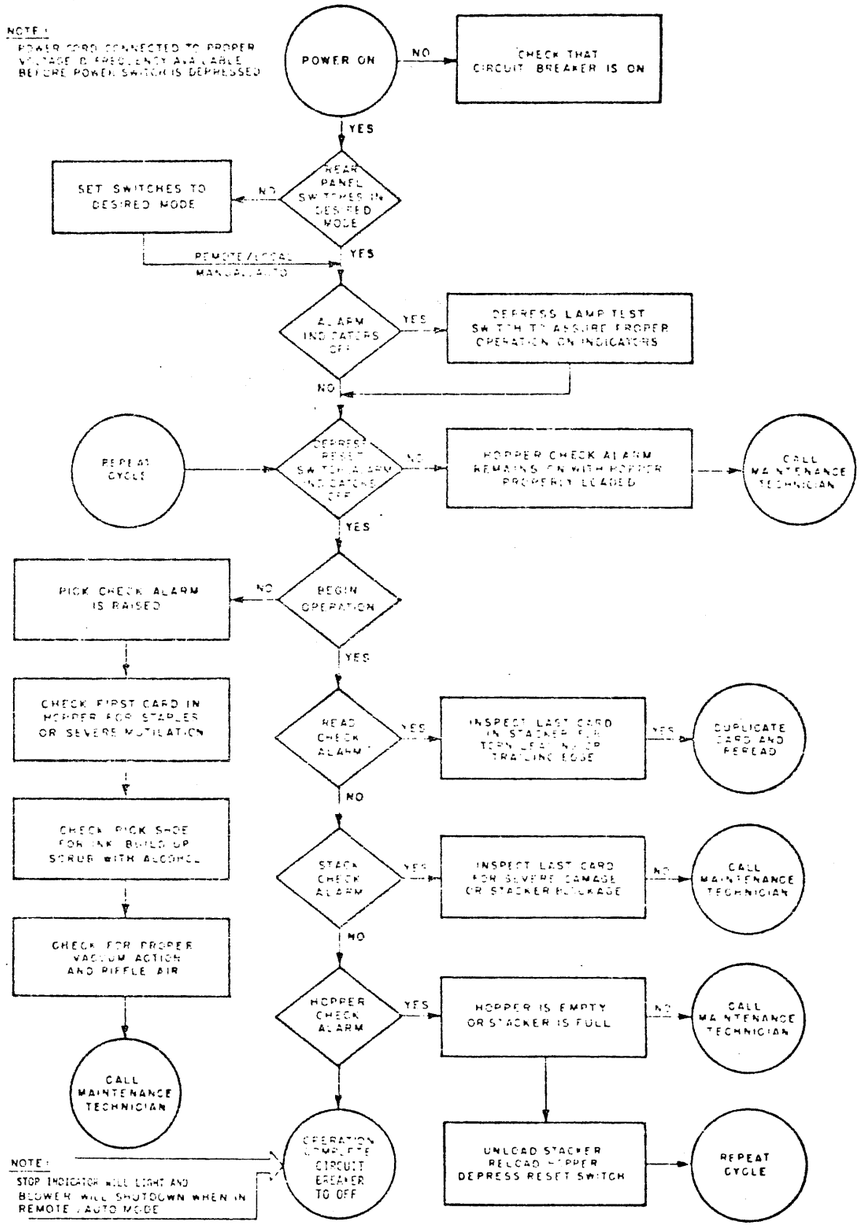


Figure 2-2. Card Reader Operational Flow Chart

SECTION 3

INSTALLATION

UNCRATING AND INSTALLATION

The applicable vendor's manual (see Table 1-2) gives detailed instructions for uncrating and installing the card reader, and/or card reader/punch. These instructions discuss installation of all standard and optional aspects of the peripheral devices described and must be followed wherever applicable. It is recommended that these sections of the vendor's manual be read and understood before attempting to uncrate and install a card reader, and/or card reader/punch.

Space Requirements

Space requirements are described in the applicable vendor's manual.

INTERCONNECTION AND CABLING

Tables 3-1 and 3-2 show the interfaces between the controller and the Prime computers for the various devices (i.e., card reader, card reader/punch and shows the cross references for these Prime signals with the vendor signals.

Table 3-1. Card Reader Mnemonic Cross Reference Chart

CARD, CARD READER MPC CONN D

Signal Name			Prime		Card Reader	
	Prime	LBD	SIG	GND	SIG	GND
PICK COMM	BDC31-	32A	1	2	LL	SS
			3	4		
			5	6		
			7	8		
ROW 12	BDC35-	32	9	10	A	E
ROW 11	BDC36-	32	11	12	B	F
ROW 0	BDC37-	32	13	14	C	H
ROW 1	BDC38-	32	15	16	D	J
ROW 2	BDC41-	32	17	18	K	P
ROW 3	BDC42-	32	19	20	L	R
ROW 4	BDC43-	32	21	22	M	S
ROW 5	BDC44-	32	23	24	N	T
ROW 6	BDC45-	32	25	26	U	W
ROW 7	BDC46-	32	27	28	V	X
ROW 8	BDC47-	32	29	30	Y	cc
ROW 9	BDC48-	32	31	32	Z	dd
HOPPER CHECK	BDC76-	27	33	34	JJ	PP
MOTION CHECK	BDC77-	27	35	36	KK	RR
READ CHK ERROR	BDC78-	27	37	38	HH	NN
READY	BDC81-	27	39	40	BB	FF
INDEX	BDC82-	27	41	42	AA	EE
			43	44		

Table 3-2. Card Punch Mnemonic Cross Reference Chart

CABLE, CARD PUNCH MPC CONN E

Signal Name			Prime		Card Punch	
	Prime	LBD	SIG	GND	SIG	GND
PUNCH COMM	BDC50-	33	1	2	D2	C3
DATA ACK.	BDC52-	33	3	4	D4	C5
			5	6		
			7	8		
ROW 12	BDC55-	33	9		B2	A3
ROW 11	BDC56-	33	11	12	B4	A5
ROW 0	BDC57-	33	13	14	B6	A7
ROW 1	BDC58-	33	15	16	B8	A9
ROW 2	BDC61-	33	17	18	B10	A11
ROW 3	BDC62-	33	19	20	B12	A13
ROW 4	BDC63-	33	21	22	B14	A15
ROW 5	BDC64-	33	23	24	D14	C15
ROW 6	BDC65-	33	25	26	D12	C13
ROW 7	BDC66-	33	27	28	D10	C11
ROW 8	BDC67-	33	[illegible]	[illegible]	D8	C9
ROW 9	BDC68-	33	[illegible]	[illegible]	D6	C7
			33	34		
DATA REQ	BDC84-	27	35	36	F2	E3
READY	BDC85-	27	37	38	U1	T2
TIMING ERROR	BDC86-	27	39	40	S1	H2
ERROR	BDC87-	27	41	42	U9	T10
PUNCH ERROR	BDC88-	27	43	44	S3	H4

SECTION 5

ASSEMBLY LANGUAGE PROGRAMMING

CONTROLLER FUNCTIONS

The principal functions of the unit record controller are:

Read a card in ASCII mode.

Read a card in binary mode.

Punch a card in ASCII mode.

Punch a card in binary mode.

Transfer a line's worth of data to the printer.*

Transfer vertical format information to the printer.*

Secondary functions of the unit record controller include the transfer of the following information to the processor's A-Register:

Card reader status

Card punch status

Controller channel number

Card reader vector address

Card punch vector address

Printer status*

Printer vector address*

DATA FORMATS AND CODES

Data transfers between the controller are by direct memory access or direct memory control (DMA or DMC) only. Formats in memory for the three peripheral devices (line printer, card reader, card reader punch) are described in the following paragraphs.

* Refer to the Unit Record Controller for Line Printer User Guide (MAN1942).

Binary Mode

Binary mode allows data in the computer's main memory to be punched on cards without the definition of any codes. Binary mode also allows cards that have been previously punched with an unknown code to be read into memory. The relationship between the card and data in memory in binary mode is shown in Figure 5-1.

There is a restriction placed on the card reader user by the vendor. When reading or punching cards in binary mode, the density of punching must not exceed 60 percent in order to preserve the structural strength of the card.

This restriction must be taken into account when dumping memory onto binary mode punched cards. One possible solution is to reorganize the card buffer so that data is in the right hand byte only. This byte would be punched in card rows 2 to 9, with forty computer words stored per card.

ASCII Mode

In ASCII mode, the code relationship between each byte in computer memory and each column on the card is rigorously defined. The bytes contain ASCII coded characters, and the card columns contain 12-row coded Hollerith characters.

When reading cards, the 12 rows of data from each card column are converted to their seven-bit ASCII equivalent. The reverse conversion takes place when cards are punched (i.e., seven bit ASCII is converted to a 12 row per column form that depicts a Hollerith code). Table 5-1 defines the relationship between the IBM26 and 29 keypunch codes, the code punched on the card, and the ASCII equivalent in the Prime computer memory. Figure 5-2 shows how data is mapped in memory and the relationship between the card column and data in memory. Illegal codes are read in as spaces (ASCII blanks).

CONTROLLER ADDRESS

The controller provides the interface between the Prime I/O Bus and one line printer, one card reader, and one card punch. Any combination of these peripheral devices may be used.

The controller address is $(\emptyset 3)_8$.

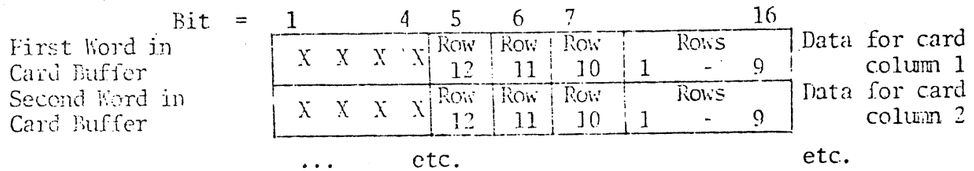


Figure 5-1. Mapping of Card and Data in Memory-Binary Mode

Note: 1. For the card reader, when card data is transferred to memory, memory bits 1 to 4 are zero.

2. For the card punch, memory bits 1 to 4 received by the controller are ignored.

ASCII MODEI	CARD CODE ZONE	NO.	IBM 26 CHAR	IBM 29 CHAR	ASCII CODE	CARD CODE ZONE	NO.	IBM 26 CHAR	IBM 29 CHAR
240	None		Space	Space	255	11	-	-	-
261	-	1	1	1	312	11	1	J	J
262	-	2	2	2	313	11	2	K	K
263	-	3	3	3	314	11	3	L	L
264	-	4	4	4	315	11	4	M	M
265	-	5	5	5	316	11	5	N	N
266	-	6	6	6	317	11	6	O	O
267	-	7	7	7	320	11	7	P	P
270	-	8	8	8	321	11	8	Q	Q
271	-	9	9	9	322	11	9	R	R
272	-	8-2		:	241	11	8-2		!
243	-	8-3	#	#	244	11	8-3	$	$
300	-	8-4	@	@	252	11	8-4	*	*
247	-	8-5		'	251	11	8-5		)
275	-	8-6		=	273	11	8-6		;
242	-	8-7		"	335	11	8-7		¬
260	0	-	0	0	246	12	-	&	&
257	0	1	/	/	301	12	1	A	A
323	0	2	S	S	302	12	2	B	B
324	0	3	T	T	303	12	3	C	C
325	0	4	U	U	304	12	4	D	D
526	0	5	V	V	305	12	5	E	E
327	0	6	W	W	306	12	6	F	F
330	0	7	X	X	307	12	7	G	G
331	0	8	Y	Y	310	12	8	H	H
332	0	9	Z	Z	311	12	9	I	I
333	0	8-2			336	12	8-2		¢
254	0	8-3	´	´	256	12	8-3	.	.
245	0	8-4	%	%	274	12	8-4	<	<
337	0	8-5		_	250	12	8-5	(	(
276	0	8-6		>	253	12	8-6	+	+
277	0	8-7		?	334	12	8-7		{

Table 5-1. Keypunch Codes, Card Codes and Internal ASCII Codes for Prime Computers

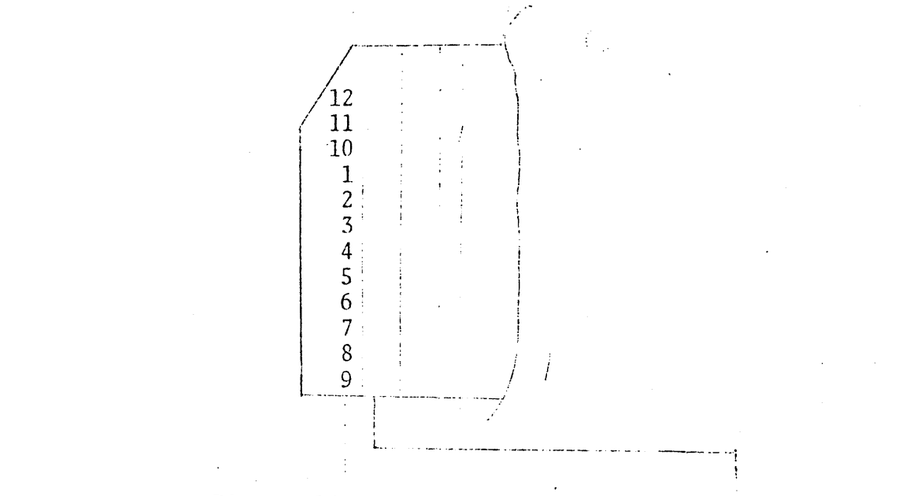


Figure 5-2. Mapping Between ASCII Card Data and Memory Buffer

PROGRAMMED INPUT/OUTPUT (PIO) COMMANDS

Table 5-2 shows the PIO commands accepted by the unit record controller. Control of each type of peripheral device is by setup words addressed to the particular device. The setup words for the line printer are discussed in the Unit Record Controller for Line Printer User Guide; setup words for the card reader and card reader/punch are in the following paragraphs:

OTA 02 Setup Card Reader

The two operations that take place as a consequence of OTA 02 are as follows:

Read a card

Setup controller for input to A Register (INA)

A-Register bits received by the controller during OTA 02, are decoded as follows:

Bits	Interpretation
1	If this bit is set, bits 15 and 16 are decoded to define a particular INA required.
2	If this bit is set (i.e., =1) and bit 1 is reset (i.e., = 0), the card reader reads the next card in the hopper, the controller transfers the information on the card to memory via a DMX channel.
3	If this bit is set, the information from the card is interpreted as binary. If this bit is reset, the card information is interpreted as ASCII.
15-16	If bit 1 is set, these two bits are decoded, and the following action takes place:

Bits 15,16 Value	Action
00	Load data register with card reader status
01	Load data register with card reader vector address
10	Load data register with controller channel number

These actions take place in preparation for an INA instruction that follows the OTA 02 instruction.

OTA 03 Setup Card Punch

OTA 03 is very similar to OTA 02 except the card punch is selected instead of the card reader.

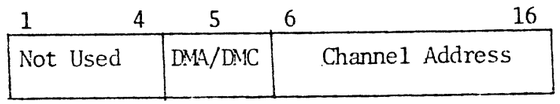
A register bits received by the controller during OTA 03 are decoded as follows:

Bits	Interpreter
1	If this bit is set, bits 15 and 16 are further decoded to define a particular INA required.
2	If this bit is set and bit 1 is not, information memory is transferred to the card punch via a DMX channel.
3	If this bit is set, the card is punched in binary format. If this bit is reset, the card is punched in ASCII format.
15-16	If bit 1 is set, these two bits are decoded and the following action takes place in preparation for an INA that follows this OTA.

Bits 15,16 Value	Action
00	Load data register with card punch status
01	Load data register with card punch vector address
10	Load data register with controller channel number

OTA 14 DMA/C Channel Number

The channel number is considered to belong to the controller. Hence, the controller has only one register to hold the channel number rather than having one for each of the three separate devices. This register is loaded by OTA 14 and the format of the OTA 14 is as follows:

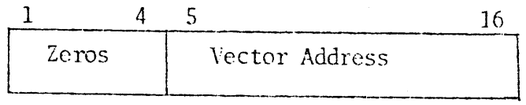


Bit 5 = 1 for DMC transfer
Bit 5 = 0 for DMA transfer

OTA 15, 16, 17 Interrupt Vector Address

OTA 15, OTA 16, and OTA 17 output to the controller the interrupt vector address for, respectively: the printer; the card reader; and the card punch.

The format for these instructions (OTA 15, OTA 16, OTA 17) is as follows:



If the vector address is not specified, the controller assumes the following addresses and interrupts via these locations:

Card reader	$(105)_8$
Card punch	$(106)_8$
Printer*	$(103)_8$

Following an OTA 15 or OTA 16 or OTA 17, the controller maintains the specified interrupt address until initialized. Following initialization, the vector addresses revert to the standard controller address given above.

INA 00 Input Data Register

The INA 00 instruction transfers the contents of the controller's Data Register to the processor. The data register normally is loaded by OTA 01, OTA 02, OTA 03 instruction and the appropriate setup word is described in the paragraphs that discuss these OTA instructions.

SKS 00 Skip if Ready

The controller sets READY when it has loaded the data register with some information that is to be transferred to the processor. The controller expects the processor to issue an INA 00 to perform the information transfer. The INA also resets READY.

SKS 01 Skip if Not Ready

The controller becomes BUSY on receipt of any OTA instruction and remains BUSY until completion of that OTA.

SKS 04 Skip if Controller Not Interrupting

The controller initiates an interrupt under the following circumstances:

The card reader has transferred 80 columns of data to the controller

The complete data buffer (up to 80 columns worth) has been transferred to the card punch

The complete data buffer has been transferred to the line printer *

SKS 06, 07 Skip if Device Not Interrupting

The SKS 06 and SKS 07 commands are defined as follows:

SKS 06 Skip if card reader is not interrupting

SKS 07 Skip if card punch is not interrupting

OCP 14 Acknowledge Interrupt

The OCP 14 instruction must be given after servicing an interrupt to clear the interrupt request.

If the controller is being serviced without use of interrupts, the controller attempts to interrupt under the circumstances detailed in the description of SKS 04. The interrupt is not recognized by the processor until the controller mask is set and Central Processor interrupts are enabled. Therefore, prior to the next time CP interrupts are enabled, it is recommended that this OCP is issued to ensure the controller cannot interrupt falsely.

STATUS WORDS

Status words are concerned with the operation of each peripheral device connected to the unit record controller and are transferred to the processor's A-Register in response to specific set information received by the controller during OTA 01, OTA 02, or OTA 03 instruction execution.

* Refer to Line Printer User Guide (MAN1942).

* Refer to Line Printer User Guide (MAN1942).

Card Reader Status Word

Bits of the card reader status word are defined as follows:

Bits	Definition
1-8	Not Used
9	Card reader on-line
10	Not Used
11	Illegal ASCII code
12	DMX overrun. The controller failed to service data from the card reader within the specified time.
13	Not used
14	Hopper check. No cards left in hopper.
15	Motion check. Card failed to leave the hopper after receipt of a read order. Operator attention is required.
16	Read check error. Each card is subjected to a light and a dark check as it travels down the read path - one of these checks failed. Operator attention is required.

Card Punch Status Word

Bits of the card punch status word are defined as follows:

Bits	Definition
1-12	Not used
13	Card punch on-line
14	DMX overrun. The controller failed to supply the punch with data within the required time period.
15	Error. A combination of a number of punch conditions that require operator attention.
16	Punch error. A check on the data punched on the card failed. The card is sent to the reject hopper.

TIMING

Card Reader

Following the OTA setup word to read a card, data is transferred to the controller at a rate of one card column every 2 milliseconds. This is the rate at which DMx requests are made in binary mode. In ASCII mode, requests are made at half this rate.

It is not necessary for the DMx range to be set to receive 80 columns of data. The controller sends information to memory until a DMx OUT OF RANGE signal is received. At this point, the controller continues to accept data from the remainder of the card currently being read, and then it goes nonbusy.

If a HOPPER CHECK status condition is received by the program, it indicates that the last OTA instruction given to read a card failed because there are no cards left in the input hopper. No data transfer to memory is made. After the reading of a card, there is an eight-millisecond time period in which the card reader is occupied in stacking the card that was most recently read, and the card reader is unable to take any action on a new read-a-card order. The user program may issue another OTA instruction to read a card during this time period, or the program can perform other tasks, without decreasing card reader throughput.

Card Punch

The card punch requires data from the controller at the rate of one card column every 6.25 milliseconds. For binary mode punching, this will be the rate at which the controller makes DMx requests. The rate is 12.5 milliseconds for ASCII mode.

The speed at which cards can be punched depends on the number of columns punched. The controller goes nonbusy after DMx end-of-range is received regardless of the number of card columns (up to 80) that have been punched. The card is ejected from the punching station at high speed into the stacker. During this eject time (88 milliseconds for greater than 72 columns punched or 120 milliseconds for less than 72 columns punched), the card punch is unable to take any action on another OTA instruction to punch-a-card. The controller may issue another OTA to the card punch during this eject time, or it can perform other tasks without decreasing card throughput.

Controller Priority

The priority of the controller with respect to other controllers in the system depends on the mix of peripherals the unit record controller is servicing. The following table shows the time between DMx accesses and the time to honor a DMx request for each of the three peripherals. (The card reader times refer to the standard model which reads at 285 cards/minute.)

	Time between DMx requests (µs)	Max. time to honor requests (µs)
Printer	35	No maximum except print rate will decrease
Card reader, binary	2000	1500
Card reader, ASCII	4000	1500
Card punch, binary	6250	3500
Card punch, ASCII	12500	3500

Time to Complete Instructions

The non-peripheral OTA instructions such as vector address, setup status, etc., take up to 7 microseconds to complete. The controller is busy for this time.

An OTA read-a-card order causes the controller to be busy for 190 milliseconds. This time may extend up to 198 milliseconds.

An OTA punch-a-card order causes the controller to be busy for 6.25 milliseconds per card column punched or 500 milliseconds for a full card. This time may be extended by 120 milliseconds.

Controller Concurrency

The controller performs only one OTA at a time. For instance, if the controller is involved in sending data to the line printer, it cannot be awaiting data read from a card. However, some degree of concurrency is possible between any two of the three peripherals.

Reading Cards and Printing

The printer is characterized by consuming a line's worth of data and then printing it without involvement from the controller. The card reader requires constant attention from the controller while a card is passing over the read head, but no attention while the card is being stacked. During this stacking interval, data can be sent to the printer.

The following table shows how the nominal device speeds can drop when the printer and card reader are run concurrently.

Stand Alone Speed			Concurrent Operation Speed	
Printer (lpm)	Card (cpm)	Reader Model	Printer (lpm)	Card Reader (cpm)
300	150	D150	155	150
300	285	M200	300	285
300	300	M300L	300	300
a) 300	600	M600L	300	300
b) 300	600	M600L	200	600
600	150	D150	155	150
600	285	M200	600	285
600	300	M300L	400	300
600	600	M600L	600	600

The designations a) and b) refer to alternate choices of either sacrificing card reading speed or sacrificing print speed.

The best performance is obtained when the speed of the two devices is equal.

Punching Cards and Printing

The controller is devoted to servicing the card punch for a time proportional to the number of columns punched. While the card is being stacked, the line printer may be serviced.

The table below shows how print speed drops as the number of card columns punched increasts. The punch is running at its maximum speed.

Print Speed (lpm)	No. Columns Punched	Punch Speed (cpm)
300	11	300
160	40	160
101	80	101

Reading Cards and Punching Cards

The card reader and card punch share use of the controller. When one is stacking cards, the other is performing data transfers. The following table shows how punching speed and reading speed are interrelated.

Stand Alone Speed		Concurrent Operation Speed	
Reader (cpm)	Punch Rate/No. Columns	Reader (cpm)	Punch (cpm)
150	206/25	150	150
150	160/40	127	127
150	100/80	100	100

Reader (cpm)	Punch Rate/No. Columns	Reader (cpm)	Punch (cpm)
300	300/10	300	300
300	160/40	160	160
300	100/80	100	100

The exit from the subroutines PRINT, CARDR, CARDP, described in items 3, 4, and 5 in the foregoing list of functions, is back to the TESTSW routine described in item 2. The TESTSW routine exits as appropriate to test the next I/O device.

DCARD Error Messages

The following messages may be generated by DCARD:

CARD READER OFF-LINE

CARD READER DMX OVERFLOW

CARD READER HOPPER, MOTION OR READ CHECK ERROR

CARD PUNCH OFF-LINE

CARD PUNCH DMX OVERRUN

CARD PUNCH MOTION OR PUNCH ERROR

PRINTER OFF-LINE

CARD READER READ ERROR

CARD READER ILLEGAL ASCII CODE

PREVENTIVE MAINTENANCE

Testing and maintenance of the line printer that is connected to the unit record controller is outlined in the vendor's manual (Table 1-2). Users are advised to perform preventive maintenance on a regular schedule in the manner recommended by the vendor's manual.

Cleaning

It is important to keep the card reader clean and cleanliness can prevent problems that appear to be malfunctions. Follow the cleaning procedures and recommendations, described in the vendor's manual, carefully.

Lubrication

The rollers of the rotary solenoid must be checked and one drop of lubricant applied to each roller, every four months of operation. Refer to the vendor's manual for a detailed illustration of lubrication procedures.

LTR	DATE	REVISION	DR.	CK.

MATERIAL	DWN J.ALVES	PRIME COMPUTER INC. NATICK, MASS.			
UNLESS OTHERWISE SPECIFIED –REMOVE ALL BURRS AND SHARP EDGES –DIMENSIONS ARE IN INCHES –TOLERANCES xx ±.02 xxx ±.005 ANGLES ±1/2°	CHK J.C. Chisholm / ENG. David Gardiner / APPRD George F. Dineen	UNIT RECORD CONTROLLER (PRINTER, CARD READER, CARD PUNCH) PRODUCT SPECIFICATION			
	USED ON 3191 / NEXT ASSY 3151,2,3,4,5,	SCALE SHEET 1 OF 20	SIZE	DWG. NO. SPC1824	REV. A

LTR	DATE	REVISION	DR.	CK.

1.0 GENERAL

This controller, built from one PRIME standard sized circuit card, provides the interface between the I/O bus and the following peripherals: (a) line printer, (b) card reader, (c) card punch. The controller will handle one of each of these peripherals on a concurrent basis.

2.0 REFERENCE DOCUMENTS

SPC0588 - Microprogrammed controller product specification (reference A) PE-T-42 - I/O bus specification (reference B)

3.0 PERIPHERAL DEVICES

3.1 Line Printer

The controller/line printer interface is designed to be Data Products compatible. This will allow the complete family of Data Products' printers to be used as well as printers from other manufacturers which are plug-to-plug exchangable with the Data Products' line; (an example of the latter is the Odec line printer).

Basic specifications of the Data Products Model 2230 are as follows:

Speed - 300 lines per minute for 64 character set
200 lines per minute for 96 character set

Characters/Line - up to 136

Line spacing - 6 lines per inch

Paper slew speed - 20 inches per second

Paper used - fanfold, up to 6 parts

Vertical format - (optional) 12 channel with 15 line step count

3.2 Card Reader

The controller interface is designed to operate with Documation card readers. These provide a spectrum of card reading ability from 150 to 1200 cards per minute. All have the same interface.

We intend offering as standard the M200 model. Features of this are as follows:

I-28

	USED ON / NEXT ASSY	SCALE SHEET 2 OF 20	SIZE	DWG. NO. SPC1824	REV. A

PDF-005

Card rate - 285 cards per minute

Card type - standard 80 column ANSI

Hopper capacity - 550 cards

Stacker capacity - 550 cards

Mounting - table top, weighs 60 pounds

3.3 Card Punch

The punch selected is the Documation Model P-100. Characteristics of this are as follows:

Card rate
- 100 cards per minute for all 80 columns
- 300 cards per minute for first 10 columns

Card type - standard 80 column ANSI

Hopper capacity - 1000 cards

Stacker capacity - 1000 cards

Reject hopper capacity - 100 cards

Mounting - floor, weighs 350 pounds

4.0 CONTROLLER CAPABILITY

4.1 Principal functions of the controller are as follows:

Transfer a line's worth of data to the printer

Transfer vertical format information to the printer

Read a card in ASCII mode

Read a card in binary mode

Punch a card in ASCII mode

Punch a card in binary mode

Secondary functions of the controller are as follows:

Transfer to the CP A register

- Printer status
- Card reader status
- Card punch status

- Controller channel number
- Printer vector address
- Card reader vector address
- Card punch vector address

4.2 Data Formats and Codes

Data transfers between the controller and memory are by DMA/C only. Formats of data in memory for the three peripherals are as follows:

4.2.1 Line Printer

The line printer will accept up to 136 printable ASCII characters per line. Figure 1 shows the mapping between data in memory and data printed.

1	2	8	9	10	16	
X	Col 1 Data		X	Col 2 Data		1st word in memory print buffer
X	Col 3 Data		X	Col 4 Data		2nd word
X	Etc		X			etc.
X			X			

FIGURE 1

Memory bits 1 and 9 are ignored by the controller.

Printing codes are shown in Table 1 for a printer equipped with a 64 character drum and in Table 2 for a printer equipped with a 96 character drum. In each table, b_7 through b_1 refer to memory bits 2 through 8 for odd column characters or memory bits 10 through 16 for even column characters.

The control codes PF, FF and CR shown in the tables cause the following to take place:

PF - print all characters received since the last control code and then advance paper one line.

FF - print all characters received since the last control code and then advance paper to the top of the next form.

b_7 b_6 b_5	0 0 0	0 1 0	0 1 1	1 0 0	1 0 1
b_4 b_3 b_2 b_1					
0 0 0 0		Space	0	@	P
0 0 0 1		!	1	A	Q
0 0 1 0		"	2	B	R
0 0 1 1		#	3	C	S
0 1 0 0		$	4	D	T
0 1 0 1		%	5	E	U
0 1 1 0		&	6	F	V
0 1 1 1		'	7	G	W
1 0 0 0		(	8	H	X
1 0 0 1		)	9	I	Y
1 0 1 0	PF	*	:	J	Z
1 0 1 1		+	;	K	[
1 1 0 0	FF	,	<	L	\
1 1 0 1	CR	-	=	M	]
1 1 1 0		.	>	N	↑
1 1 1 1		/	?	O	←

TABLE 1. Coded 64 Character (ASCII) Set

All other codes result in "space"

b^7 b^6 b^5	0 0 0	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
b^4 b^3 b^2 b^1							
0 0 0 0		Space	0	@	P	\	P
0 0 0 1		!	1	A	Q	a	q
0 0 1 0		"	2	B	R	b	r
0 0 1 1		#	3	C	S	c	s
0 1 0 0		$	4	D	T	d	t
0 1 0 1		%	5	E	U	e	u
0 1 1 0		&	6	F	V	f	v
0 1 1 1		'	7	G	W	g	w
1 0 0 0		(	8	H	X	h	x
1 0 0 1		)	9	I	Y	i	y
1 0 1 0	PF	*	:	J	Z	j	z
1 0 1 1		+	;	K	[	k	{
1 1 0 0	FF	,	<	L	\	l	|
1 1 0 1	CR	-	=	M	]	m	}
1 1 1 0		.	>	N	↑	n	~
1 1 1 1		/	?	O	←	o	⌂

TABLE 2. Standard 96 Character Set

All other codes result in "space"

CR - print all characters received since the last control code. The controller will send a CR code to the printer if DMX end-of-range is received. The word accompanying the end-of-range will be sent to the printer first.

4.2.2 Card Reader/Card Punch - Binary Mode

Binary mode enables computer memory data to be punched on cards without reference to any codes. It also allows cards which have been previously punched with an unknown code to be read into memory. The relationship between the card and data in memory is shown in Figure 2

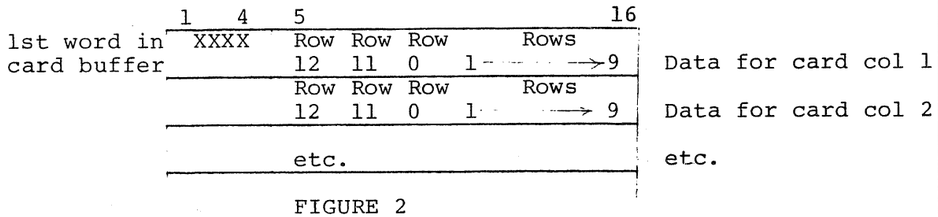


FIGURE 2

For the card reader, when card data is transferred to memory, memory bits 1-4 will be zeros.

For the card punch, memory bits 1-4 received by the controller will be ignored.

A restriction placed on the card reader user by the vendor is that density of punching should not exceed 60% - to preserve the strength of the card. This should be taken into account when dumping memory onto cards. One possible format would be to reorganize the card buffer so data is in the right hand byte only. This byte would be punched in card rows 2-9. Forty computer words would be stored per card.

4.2.3 Card Reader/Card Punch - ASCII Mode

In this mode, the code relationship between each computer byte and each column on the card is rigidly defined; the former are ASCII coded characters and the latter are 12 row coded Hollerith characters.

When reading cards, the twelve rows of data from each card column are converted to their seven bit ASCII equivalent. The reverse takes place when cards are punched. Table 3 defines the relationship between the IBM26 and 29 keypunches, the code punched on the card and the ASCII equivalent in computer memory.

Data in memory is organized as shown in Figure 3.

ASCII CODE	CARD CODE ZONE	CARD CODE NUM	IBM 26 CHAR	IBM 29 CHAR
240	None		Space	Space
261	-	1	1	1
262	-	2	2	2
263	-	3	3	3
264	-	4	4	4
265	-	5	5	5
266	-	6	6	6
267	-	7	7	7
270	-	8	8	8
271	-	9	9	9
272	-	8-2		:
243	-	8-3	#	#
300	-	8-4	@	@
247	-	8-5		'
275	-	8-6		=
242	-	8-7		"
260	0	-	0	0
257	0	1	/	/
323	0	2	S	S
324	0	3	T	T
325	0	4	U	U
326	0	5	V	V
327	0	6	W	W
330	0	7	X	X
331	0	8	Y	Y
332	0	9	Z	Z
333	0	8-2		
254	0	8-3	,	,
245	0	8-4	%	%
337	0	8-5		_
276	0	8-6		>
277	0	8-7		?
255	11	-	-	-
312	11	1	J	J
313	11	2	K	K
314	11	3	L	L
315	11	4	M	M
316	11	5	N	N
317	11	6	O	O
320	11	7	P	P
321	11	8	Q	Q
322	11	9	R	R
241	11	8-2		!
244	11	8-3	$	$
252	11	8-4	*	*
251	11	8-5		)
273	11	8-6		;
335	11	8-7		¬
246	12	-	&	&
301	12	1	A	A
302	12	2	B	B
303	12	3	C	C
304	12	4	D	D
305	12	5	E	E
306	12	6	F	F
307	12	7	G	G
310	12	8	H	H
311	12	9	I	I
336	12	8-2		¢
256	12	8-3	.	.
274	12	8-4	¤	<
250	12	8-5		(
253	12	8-6		+
334	12	8-7		|

TABLE 3

I-31

1	8 9	16
1st word in card buffer	Card Col 2	Card Col 2
	Card Col 3	Card Col 4
	Etc.	

FIGURE 3

5.0 PROGRAMMING DETAILS

5.1 Controller Address

The controller provides the interface between the PRIME I/O bus and one printer, one card reader and one card punch. Any combination of these peripherals may be used.

Controller address is $(03)_8$.

5.2 PIO Commands

Control of each peripheral is by setup words addressed to that particular peripheral. Table 4 shows PIO commands accepted by the controller. They are described below.

5.2.1 OTA 01. Setup Printer

There are three operations connected with the printer that are controlled by this setup word:

a) print a line

b) printer vertical format/line space operation

c) setup controller for input to A register (INA)

A register bits received by the controller as a consequence of this OTA are decoded as follows:

Bit 1 - If this bit is set, bits 15 and 16 are further decoded to define a particular INA required.

Bit 2 - If this bit is set (and bit 1 is not), information from memory will be transferred to the printer via a DMX channel.

Bit 3 - If this bit is set (and bits 1 and 2 are not), A register bits 10 - 16 will be sent to the printer as VFU or line space information - see definition below.

Bits 15, 16 - If bit 1 is set, these two bits are decoded and the following action takes place in preparation for an INA which follows this OTA.

TABLE 4. PIO Commands

Function Code Bits 7-10	Op Code Bits 1-6 14_8 (OCP)	34_8 (SKS) (SKIP IF)	54_8 (INA)	74_8 (OTA)
00		Ready	Input data register	
01		Not Busy		Setup Printer
02				Setup Card Reader
03				Setup Card Punch
04		Controller Not Interrupting		
05		Printer Not Interrupting		
06		Card Reader Not Interrupting		
07		Card Punch Not Interrupting		
10				
11				
12				
13				
14	Acknowledge Interrupt			DMA/C Channel #
15	Set Int Mask			Printer Vector Address
16	Clear Int Mask			Card Reader Vector Address
17	Initialize			Card Punch Vector Address

I-32

00 - Load data register with printer status

01 - Load data register with printer vector address

10 - Load data register with controller channel number

Bits 10 - 16 - If bit 3 is set, these bits are sent to the printer to line space or control the optional vertical format unit. Three types of paper movement operations are possible: (1) If A register bits 10 and 12 are set, up to 15 lines will be spaced in accordance with information given in Table 5(a); (2) If A register bit 10 is set and bit 12 is reset, a particular paper tape channel is selected in accordance with Table 5(b). The paper will space until the next hole in that channel is detected; (3) If A register bits 10 - 16 are set to the codes for PF or FF (as defined in Tables 1 and 2), a PF or a FF operation will take place.

5.2.2 OTA 02. Setup Card Reader

The two operations that take place as a consequence of this OTA are as follows:

a) Read a card

b) Setup controller for input to A register (INA)

A register bits received by the controller during this OTA are decoded as follows:

Bit 1 - If this bit is set, bits 15 and 16 are decoded to define a particular INA required.

Bit 2 - If this bit is set (and bit 1 is not), the card reader will read the next card in the hopper and the controller will transfer the information on the card to memory via a DMX channel.

Bit 3 - If this bit is set, the information from the card will be interpreted as binary. If this bit is reset, the card information will be interpreted as ASCII.

Bit 15, 16 - If bit 1 is set, these two bits are decoded and the following action takes place in preparation for an INA which follows this OTA.

AC BITS							No. Of Lines Spaced
10	11	12	13	14	15	16	
1	X	1	0	0	0	0	0
1	X	1	0	0	0	1	1
1	X	1	0	0	1	0	2
1	X	1	0	0	1	1	3
1	X	1	0	1	0	0	4
1	X	1	0	1	0	1	5
1	X	1	0	1	1	0	6
1	X	1	0	1	1	1	7
1	X	1	1	0	0	0	8
1	X	1	1	0	0	1	9
1	X	1	1	0	1	0	10
1	X	1	1	0	1	1	11
1	X	1	1	1	0	0	12
1	X	1	1	1	0	1	13
1	X	1	1	1	1	0	14
1	X	1	1	1	1	1	15

TABLE 5 (a)

AC BITS							Tape Channel
10	11	12	13	14	15	16	
1	X	0	0	0	0	0	0
1	X	0	0	0	0	1	1
1	X	0	0	0	1	0	2
1	X	0	0	0	1	1	3
1	X	0	0	1	0	0	4
1	X	0	0	1	0	1	5
1	X	0	0	1	1	0	6
1	X	0	0	1	1	1	7
1	X	0	1	0	0	0	8
1	X	0	1	0	0	1	9
1	X	0	1	0	1	0	10
1	X	0	1	0	1	1	11

TABLE 5 (b)

00 - Load data register with card reader status

01 - Load data register with card reader vector address

10 - Load data register with controller channel number

5.2.3 OTA 03. Setup Card Punch

This OTA is very similar to OTA 02 except the card punch is selected instead of the card reader.

A register bits are decoded as follows:

Bit 1 - If this bit is set, bits 15 and 16 are further decoded to define a particular INA required.

Bit 2 - If this bit is set (and bit 1 is not), information from memory will be transferred to the card punch via a DMX channel.

Bit 3 - If this bit is set, the card will be punched in binary format. If this bit is reset, the card will be punched in ASCII format.

Bits 15, 16 - If bit 1 is set, these two bits are decoded and the following action takes place in preparation for an INA which follows this OTA.

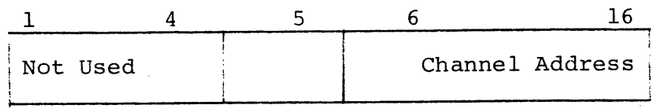
00 - Load data register with card punch status

01 - Load data register with card punch vector address

10 - Load data register with controller channel number

5.2.4 OTA 14. DMA/C Channel Number

The channel number is considered to belong to the controller. Hence the controller has only one register to hold the channel number rather than having one for each of the three separate devices. This register is loaded by OTA 14 and the format of the OTA is as follows:

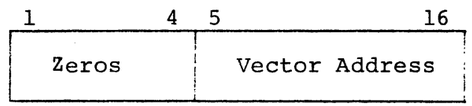


Bit 5 = 1 for DMC transfer
Bit 5 = 0 for DMA transfer

5.2.5 OTA 15, 16, 17

These OTA's output to the controller the interrupt vector address for respectively the printer, the card reader and the card punch.

The format for these instructions is as follows:



If the vector address is not specified the controller will assume the following addresses and will interrupt via these locations:

Printer	$(103)_8$
Card reader	$(105)_8$
Card punch	$(106)_8$

Following an OTA 15/16/17, the controller will maintain the specified interrupt address until initialized. Following initialization, the vector addresses will revert to the standard address given above.

5.2.6 INA 00 Input Data Register

This instruction is used to transfer the contents of the controller's data register to the central processor. The data register will normally have been loaded by OTA 01/02/03 and the appropriate setup word as detailed above.

5.2.7 SKS 00 Skip if Ready

The controller will set Ready when it has loaded the data register with some information which is to be transferred to the central processor. The controller expects the central processor to issue an INA 00 to perform the information transfer. The INA will also reset Ready.

5.2.8 SKS 01 Skip if Not Busy

The controller will become Busy on receipt of any OTA instruction and will remain Busy until completion of that OTA. Paragraph 6.5 below indicates the duration of the Busy state for various orders.

5.2.9 SKS 04 Skip if Controller not Interrupting

The controller will initiate an interrupt under the following circumstances:

a) When the line printer has completed the printing of a line or a paper format operation.

b) When the card reader has transferred 80 columns of data to the controller.

c) When the complete data buffer (up to 80 columns worth) has been transferred to the card punch.

d) When the complete data buffer has been transferred to the line printer.

e) When paper movement information (specified by bit 3 of the OTA setup) has been sent to the printer.

5.2.10 SKS 05/06/07

These SKS commands are defined as follows:

SKS 05 - Skip if printer is not interrupting

SKS 06 - Skip if card reader is not interrupting

SKS 07 - Skip if card punch is not interrupting

Definitions of these SKS commands are as in 5.2.9 (a)/(b)/(c) above respectively.

5.2.11 OCP 14 Acknowledge Interrupt

This OCP should be given following servicing of an interrupt in order to clear the interrupt request.

If the controller is being serviced without use of interrupts, it should be mentioned that the controller will still attempt to interrupt under the circumstances detailed in paragraph 5.2.9 above. The interrupt will not be recognized by the central processor until the controller mask is set and CP interrupts are enabled. Therefore, prior to the next time CP interrupts are enabled, it would be wise to issue this OCP to ensure the controller can not interrupt falsely.

5.2.12 Other OCP's

These are defined in reference A.

5.3 Status Words

Status words concerned with the operation of each peripheral will be transferred to the CP in response to specific setup information received by the controller during OTA 01/02/03 as detailed above.

5.3.1 Printer Status Word

Definition of the bits of this are as follows:

Bits 1-8 - Not used

Bit 9 - Printer on-line

Bit 10 - Printer not busy - the printer is able to receive data for a new line or another format command.

Bits 11-16 - Not used

5.3.2 Card Reader Status Word

Bits of this word are defined as follows:

Bits 1-8 - Not used

Bit 9 - Card reader on-line

Bit 10 - Not used

Bit 11 - Illegal ASCII code

Bit 12 - DMX overrun. The controller failed to service data from the card reader within the specified time (detailed in paragraph 6.4 below).

Bit 13 - Not used

Bit 14 - Hopper check. No cards left in hopper (see paragraph 6.2 below).

Bit 15 - Motion check. Card failed to leave the hopper after receipt of a read order. Operator attention required.

Bit 16 - Read check error. Each card is subjected to a light and a dark check as it travels down the read path - one of these checks failed. Operator attention is required.

E-35

5.3.3 Card Punch Status Word

Bits of this word are defined as follows:

Bit 1-12 - Not used

Bit 13 - Card punch on-line

Bit 14 - DMX overrun. The controller failed to supply the punch with data within the required time period (detailed in paragraph 6.4 below).

Bit 15 - Error. A combination of a number of punch conditions that require operator attention.

Bit 16 - Punch error. A check on the data actually punched on the card failed. The card will be sent to the reject hopper.

6.0 TIMING DETAILS

6.1 Line Printer

This contains a buffer which accepts a full line of data. Transfer of data to the printer is completely asynchronous and print speed will not slow down provided the complete line of data is transferred to the printer within 2.5 milliseconds of the interrupt which signifies the previous line has been printed.

The maximum rate of transfer of a 136 character line plus control character is 2.4 milliseconds. This corresponds to one DMX request every 35 microseconds.

6.2 Card Reader

Following the OTA setup word to read a card, data will be transferred to the controller at the rate of one card column every 2 milliseconds. This is, therefore, the rate at which DMX requests will be made in binary mode. In ASCII mode, requests will be made at half this rate.

It is not necessary for the DMX range to be set to receive 80 columns worth of data. The controller will send information to memory until DMX end-of-range is received. After that the controller will continue to accept data from the rest of the card and then go non-busy.

If a hopper check status condition is received by the program, it means that the last OTA to read a card failed because no cards were left in the hopper. No data transfer to memory will, of course, be made. Following the reading of a card, there is an 8 millisecond time period in which the card reader is

occupied in stacking the card just read and is unable to take any action on a new read-a-card order. The controller may issue another OTA to the card reader during this time period or can more profitably perform other tasks without decreasing card throughput.

6.3 Card Punch

The card punch requires data from the controller at the rate of one card column every 6.25 milliseconds. For binary mode punching, this will be the rate at which the controller will make DMX requests; the rate will be 12.5 milliseconds for ASCII mode.

The speed at which cards can be punched is dependent on the number of columns punched. The controller will go non-busy after DMX end-of-range is received irrespective of how many card columns (up to 80) have been punched - the card will be ejected from the punching station at high speed into the stacker. During this eject time (88 milliseconds for greater than 72 columns punched or 120 milliseconds for less than 72 columns punched) the card punch is unable to take any action on another OTA punch-a-card. The controller may issue another OTA to the card punch during this eject time or can more profitably perform other tasks without decreasing card throughput.

6.4 Controller Priority

The priority of this controller with respect to others in the system depends on the mix of peripherals this controller is servicing. The following table shows the time between DMX accesses and the time to honor a DMX request for each of the three peripherals. (The card reader times refer to the standard model which reads at 285 cards/minute.)

	Time between DMX requests (us)	Max. time to honor requests (us)
Printer	35	No maximum except print rate will decrease
Card reader, binary	2000	1500
Card reader, ASCII	4000	1500
Card punch, binary	6250	3500
Card punch, ASCII	12500	3500

I-36

6.5 Time to Complete Various Instructions

The non-peripheral OTA instructions such as vector address, setup status, etc., take up to 7 microseconds to complete. The controller will be busy for this time.

An OTA print to a non-busy line printer takes 2.4 milliseconds to complete for a full length line assuming the controller is not slowed down by DMX transfers. The controller will be busy for this length of time and then can do other tasks while the line is being printed. The latter takes 200 milliseconds for 64 character drum.

An OTA read-a-card order will cause the controller to be busy for 190 milliseconds. This time may extend up to 198 milliseconds as explained in paragraph 6.2 above.

An OTA punch-a-card order will cause the controller to be busy for 6.25 milliseconds per card column punched or 500 milliseconds for a full card. This time may be extended by 120 milliseconds as explained in paragraph 6.3 above.

6.6 Degree of Controller Concurrency

The controller performs only one OTA at a time. This means, for instance, that if the controller is involved in sending data to the line printer, it cannot be awaiting data read from a card. However, some degree of concurrency is possible between any two of the three peripherals.

6.6.1 Reading Cards and Printing

The printer is characterized by consuming a line's worth of data and then printing it without involvement from the controller. The card reader requires constant attention from the controller while a card is passing over the read head, but no attention while the card is being stacked. It is during this stacking interval that data can be sent to the printer.

The following table shows how the nominal device speeds can drop when the printer and card reader are run concurrently.

	Stand Alone Speed			Concurrent Operation Speed	
	Printer (lpm)	Card (cpm)	Reader Model	Printer (lpm)	Card Reader (cpm)
	300	150	D150	155	150
	300	285	M200	300	285
	300	300	M300L	300	300
a)	300	600	M600L	300	300
b)	300	600	M600L	200	600

Printer (lpm)	Card (cpm)	Reader Model	Printer (lpm)	Card Reader (cpm)
600	150	D150	155	150
600	285	M200	600	285
600	300	M300L	400	300
600	600	M600L	600	600

The designations a) and b) refer to alternate choices of either sacrificing card reading speed or sacrificing print speed.

A conclusion from the table is that best performance is obtained when the speed of the two devices is equal.

6.6.2 Punching Cards and Printing

The controller will be devoted to servicing the card punch for a time proportional to the number of columns punched. While the card is being stacked, the line printer may be serviced.

The table below shows how print speed drops as the number of card columns punched increases. The punch is running at its maximum speed.

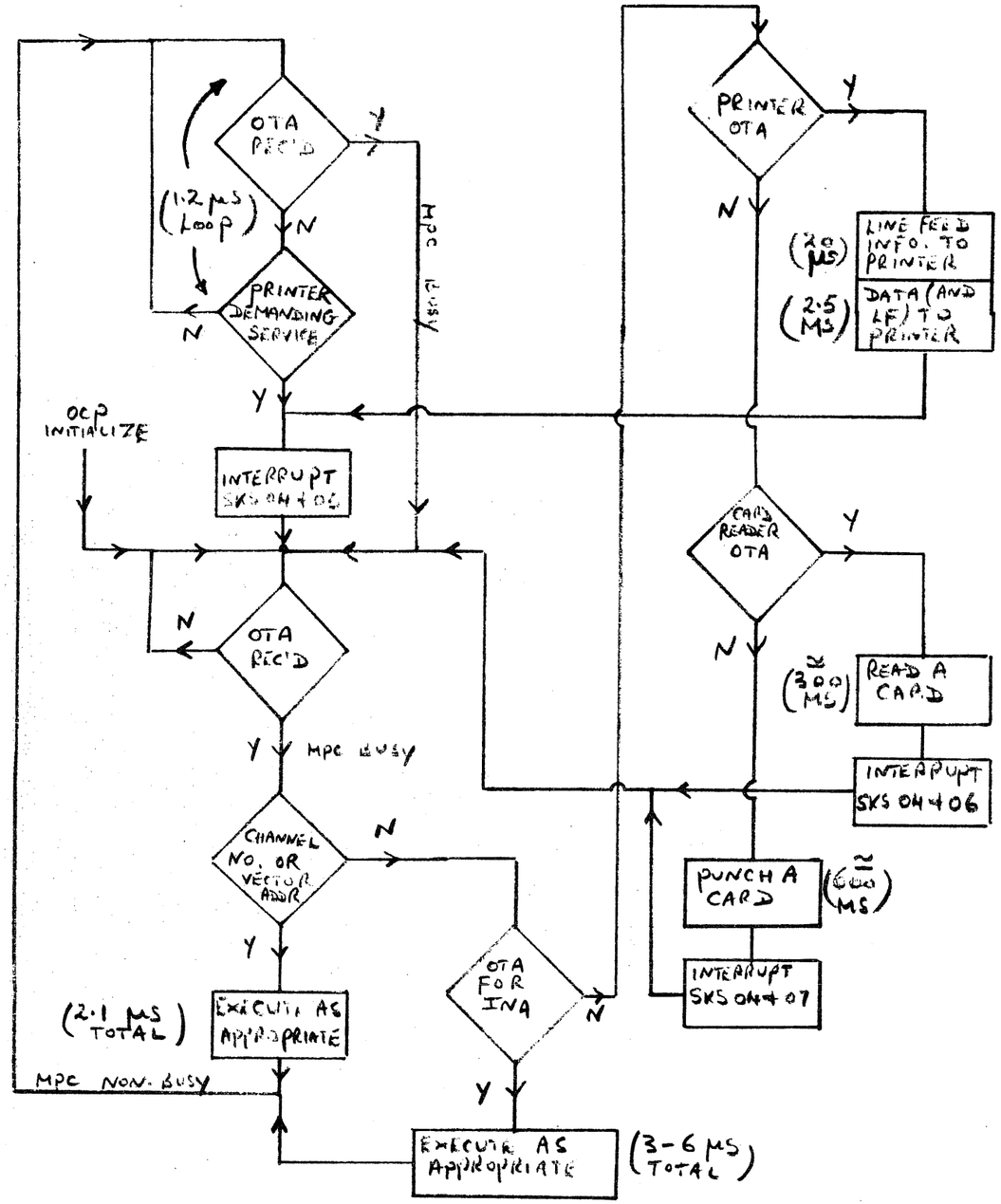
Print Speed (lpm)	No. Columns Punched	Punch Speed (cpm)
300	11	300
160	40	160
101	80	101

6.6.3 Reading Cards and Punching Cards

The card reader and card punch will share use of the controller such that when one is stacking cards, the other will be performing data transfers. The following table shows how punching speed and reading speed are inter-related.

Stand Alone Speed		Concurrent Operation Speed	
Reader (cpm)	Punch Rate/No. Columns	Reader (cpm)	Punch (cpm)
150	206/25	150	150
150	160/40	127	127
150	100/80	100	100
300	300/10	300	300
300	160/40	160	160
300	100/80	100	100

I-37



REGISTER FILE USEAGE

R0	Function Code	
R1	Setup Word, Left Byte	
R2	Setup Work, Right Byte	
R3	SKS Register (left)	
R4	Printer, Data Reg.	
R5	Printer, Usage of DR2	
R6	Printer, CR Code Constand	
R7	Card Reader, data reg. 12 - 1	
R8	Card Reader, data reg. 2 - 9	
R9	Card Reader, Temporary Storage for left byte read (tranlated)	
R10	Card Rdr/Pch, Column Count	
R11	Counter for delay	
R12		
R13	Card Punch, image of DR5	
R14	Card Punch, conversion reg.	
R15	Card Punch, decode reg.	
R16	Card Punch, translation reg.	
R17	Card Reader flag register/temp. status	
R18		
R19		
R20		
R21	Printer Status	
R22	Reader Status	
R23	Punch Status	
R24)	Channel address	Left Byte
R25)		Right Byte
R26)	Printer Vector Address	Left Byte
R27)		Right Byte
R28)	Reader Vector Address	Left Byte
R29)		Righ Byte
R30)	Punch Vector Address	Left Byte
R31)		Right Byte

10/04/1974

FLOWCHART FOR UNIT RECORD CONTROLLER INTERRUPT

LTR	DATE	REVISION	DR.	CK.

MATERIAL	DWN	PRIME COMPUTER INC. NATICK, MASS.		
	CHK HW. Noble 5-15-75			
UNLESS OTHERWISE SPECIFIED –REMOVE ALL BURRS AND SHARP EDGES –DIMENSIONS ARE IN INCHES –TOLERANCES xx ±.02 xxx ±.005 ANGLES ±1/2°	ENG. 5/15/75	MICRO-PROGRAMMED CONTROLLER (M2) PRODUCT SPECIFICATION		
	APPRD 5/9/75			
USED ON	SCALE	SIZE	DWG. NO.	REV.
NEXT ASSY	SHEET 1 OF 40	A	SPC1409	1

LTR	DATE	REVISION	DR.	CK.

1.0 SCOPE

This specification describes the operating characteristics and u-programming characteristics of the Micro-Programmed Controller (MPC). It is intended that this document provide sufficient information for the writer of u-programs.

2.0 APPLICABLE DOCUMENTS

PRIME I/O Bus Specification, PE-T-52 (Reference A)
PRIME 200 CPU Specification (Reference B)
PRIME Writeable Control Store, PE-TN-29

3.0 GENERAL

The Micro-Programmed Controller (MPC) consists of u-program storage (PROM) which controls the operation of device and the controller, a standard PRIME 200 I/O bus interface, with Programmed I/O, DMX, and interrupt capability, general device interfacing logic to the back edge connectors, and an area reserved for additional device specific logic if it is ever required to implement some specific controllers.

This provides an option board which with no variations to the hardware, and with only a different u-program, will meet the needs for the majority of PRIME's future device controllers. The board is capable of running with its own u-program, using an external u-program from a ROM simulator (for controller u-code development), or from an external ROM for field engineering purposes.

The MPC provides the following capabilities:

It interfaces to most TTL interfaced devices by merely inserting one of several different DIP types (with no etch changes). This implies different inventoried models or a final configuration procedure. It is capable of running in all presently defined I/O bus modes and meets all I/O bus timing specifications. It is capable of operational verification via on-board u-programs and it has the capability of displaying ROM addresses and single step control of the u-program.

It is capable of interfacing to multiple devices. No device data concurrency is provided although data interleaving can be implemented under certain circumstances. Thus, a card reader/card punch interface could be provided but an alternate read and punch on a card-by-card basis would be the most concurrency that could reasonably be achieved.

It provides the ability to satisfy some small communications controllers (up to four lines) requirements.

One back-edge connector is designed for a Field Engineering Tester. The other three may be for device connections.

The standard 20 MHZ crystal may be replaced by a crystal of lower frequency to provide particular timing that might be required.

II-01

	USED ON	SCALE	SIZE	DWG. NO.	REV.
	NEXT ASSY	SHEET 2 OF 40	A	SPC1409	I

PDF-005

LTR	DATE	REVISION	DR.	CK.

4.0 MPC HARDWARE SPECIFICATION

4.1 General Description

The MPC consists of (see Figure 1) a TTL Device Interface (of a general nature), PRIME I/O Bus Interface (PIO, DMX, and Interrupts), an Arithmetic Unit, a Register File (scratch pad memory), a Read Only Memory (ROM), a ROM Control Unit, and a Clock and Timing Control Unit. These units communicate via an eight bit Receive Bus and an eight bit Transmit Bus. In addition, there are various hardware assists provided such as a bit test following the Arithmetic Unit and parity checking logic at the Device Interface.

It is necessary that the Peripheral Controller Designer (μ-coder) be intimately familiar with the internal structure of the MPC so that he may effectively utilize the capabilities that it provides.

4.1.1 I/O Bus Interface

The PRIME I/O Bus Interface consists of the Standard PRIME I/O Bus Interface Logic, some registers (Data, Address, SKS, etc.) and some synchronizing logic. This is a sixteen bit interface to the I/O Bus.

4.1.2 Detailed Description

A more detailed diagram of the MPC is shown in Figure 2.

4.1.3 Device Interface

The Device Interface consists of seven registers, line drivers, line receivers, line terminations, and parity generating and checking logic. These are connected to the back-edge connectors (Device Bus) and drive or receive from twisted pair cables to the device(s).

The device interface is wired to the three 44-pin back-edge connectors (connectors C, D and E). Each connector has 22 signal and 22 ground pins providing for 22 twisted pair cables. This gives a total of 66 signal pins that are utilized for the device interface. The fourth back-edge connector is utilized by the Field Engineering Switch Panel (FEP).

The connector pins are preassigned and accommodation for particular device interface characteristics must be made by the controller designer in the cable wiring lists. The pin assignments are shown in Table I. These may be summarized by saying that there are 48 lines driven by the MPC and they are organized (enabled) into six 8-bit bytes. All of these output lines are also received by the MPC. Two additional bytes are received by the MPC (not driven). The data in the six bytes that are driven by the MPC are stored in six 8-bit registers termed Device Registers 1 through 6.

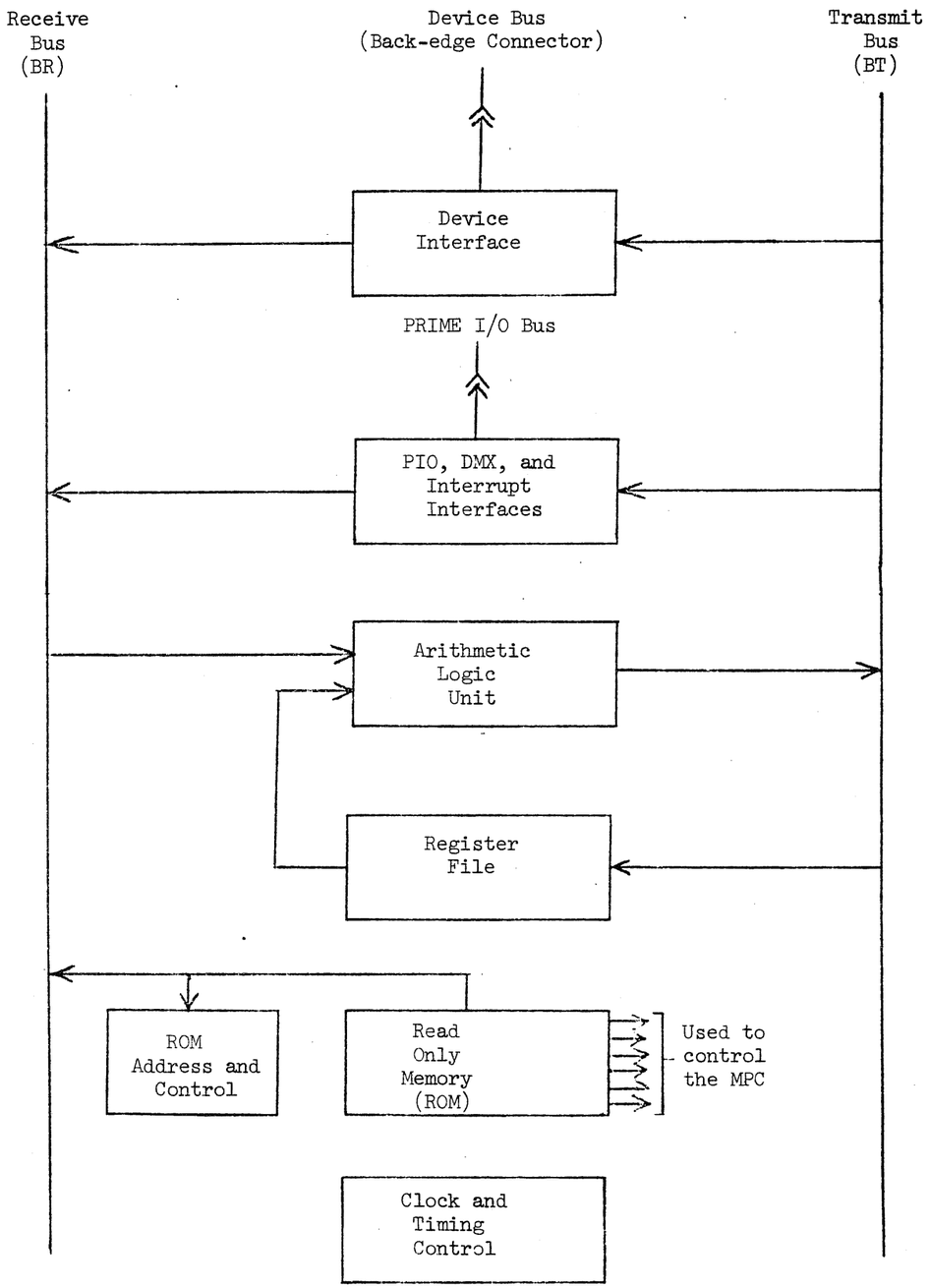
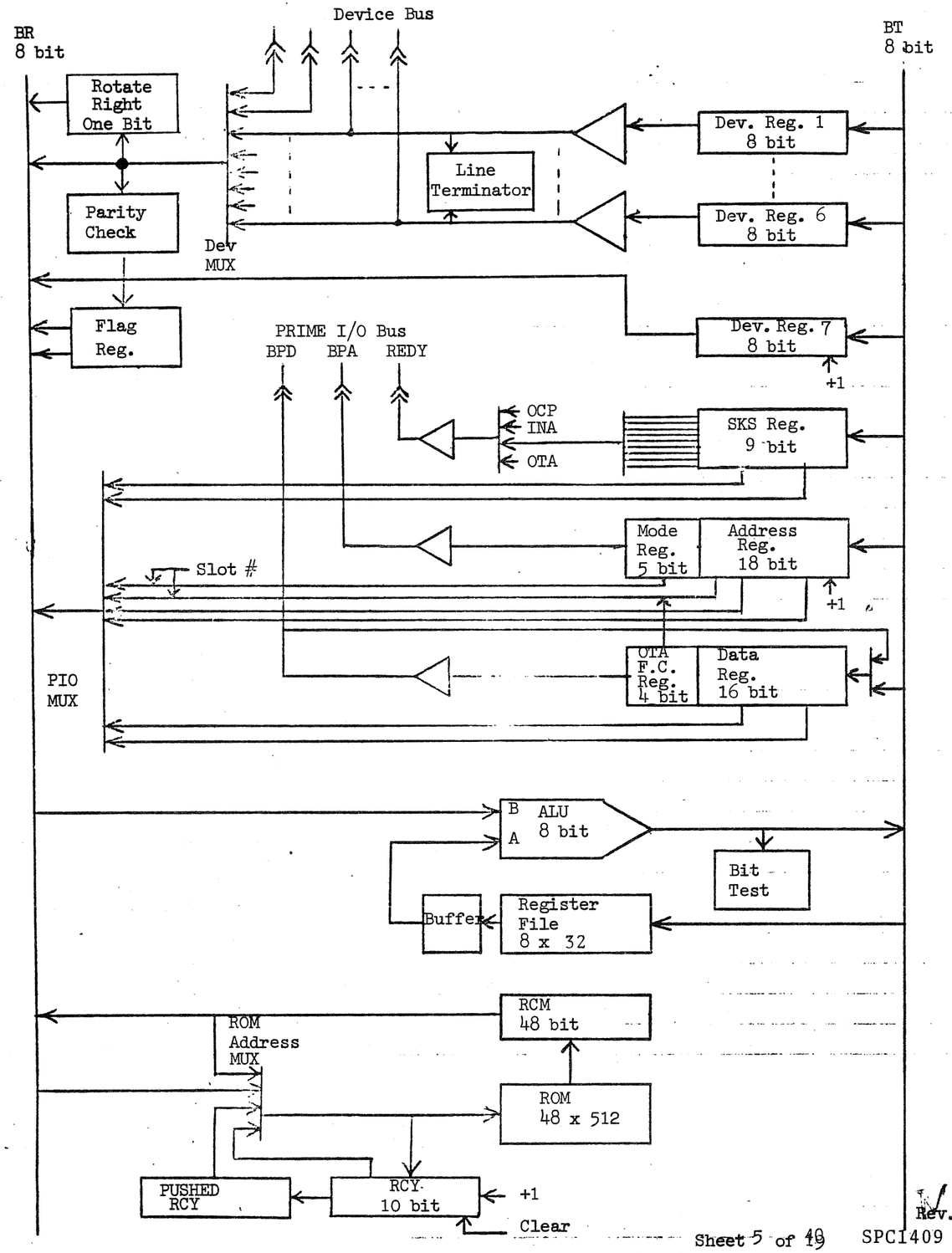


Figure 1. Block Diagram

Figure 2. Data Flow



LTR	DATE	REVISION	DR.	CK.

Table I

Back-edge Connector Pin Assignments

Pin	Connector C	Connector D	Connector E	Connector F
1	Byte 1* Bit 1	Byte 3* Bit 1	Byte 5* Bit 1	TBS
3	Byte 1* Bit 2	Byte 3* Bit 2	Byte 5* Bit 2	
5	Byte 1* Bit 3	Byte 3* Bit 3	Byte 5* Bit 3	
7	Byte 1* Bit 4	Byte 3* Bit 4	Byte 5* Bit 4	
9	Byte 1* Bit 5	Byte 3* Bit 5	Byte 5* Bit 5	
11	Byte 1* Bit 6	Byte 3* Bit 6	Byte 5* Bit 6	
13	Byte 1* Bit 7	Byte 3* Bit 7	Byte 5* Bit 7	
15	Byte 1* Bit 8	Byte 3* Bit 8	Byte 5* Bit 8	
17	Byte 2* Bit 1	Byte 4* Bit 1	Byte 6* Bit 1	
19	Byte 2* Bit 2	Byte 4* Bit 2	Byte 6* Bit 2	
21	Byte 2* Bit 3	Byte 4* Bit 3	Byte 6* Bit 3	
23	Byte 2* Bit 4	Byte 4* Bit 4	Byte 6* Bit 4	
25	Byte 2* Bit 5	Byte 4* Bit 5	Byte 6* Bit 5	
27	Byte 2* Bit 6	Byte 4* Bit 6	Byte 6* Bit 6	
29	Byte 2* Bit 7	Byte 4* Bit 7	Byte 6* Bit 7	
31	Byte 2* Bit 8	Byte 4* Bit 8	Byte 6* Bit 8	
33	Byte 7** Bit 1	Byte 7** Bit 6	Byte 8** Bit 3	
35	Byte 7** Bit 2	Byte 7** Bit 7	Byte 8** Bit 4	
37	Byte 7** Bit 3	Byte 7** Bit 8	Byte 8** Bit 5	
39	Byte 7** Bit 4	Byte 8** Bit 1	Byte 8** Bit 6	
41	Byte 7** Bit 5	Byte 8** Bit 2	Byte 8** Bit 7	
43	Spare	Spare	Byte 8** Bit 8	

Even pins are ground.
* Bi-directional data capability.
** Input only capability.

II-03

USED ON	SCALE	SIZE	DWG. NO.	REV.
NEXT ASSY	SHEET 6 OF 40	A	SPC1409	1

The line drivers are 7416.
drivers

Provision is also made for a component DIP to be connected to each line, to provide termination capability.

The receivers consist simply of an input to a 74151 eight to one multiplexer and the component DIP.

Thus, the two kinds of lines in the device interface are shown in Figure 3.

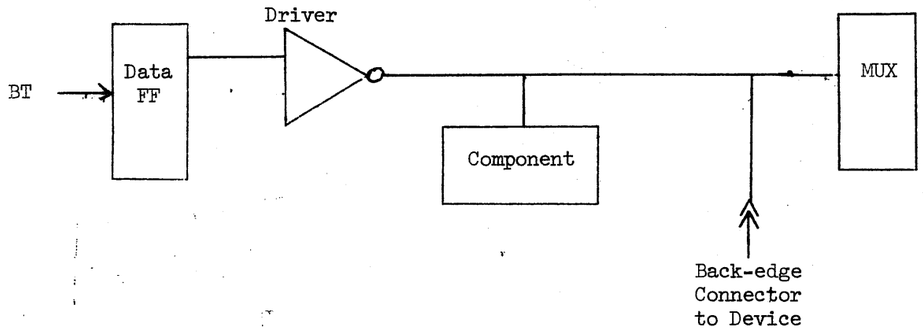


Figure 3. (a) Driven Lines (48 Lines)

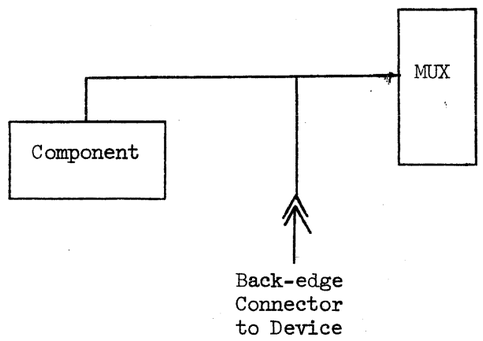


Figure 3. (b) Received Lines (16 Lines)

The use of component DIPs and drivers is optional but those used must be pin compatible with those shown in Figure 4.

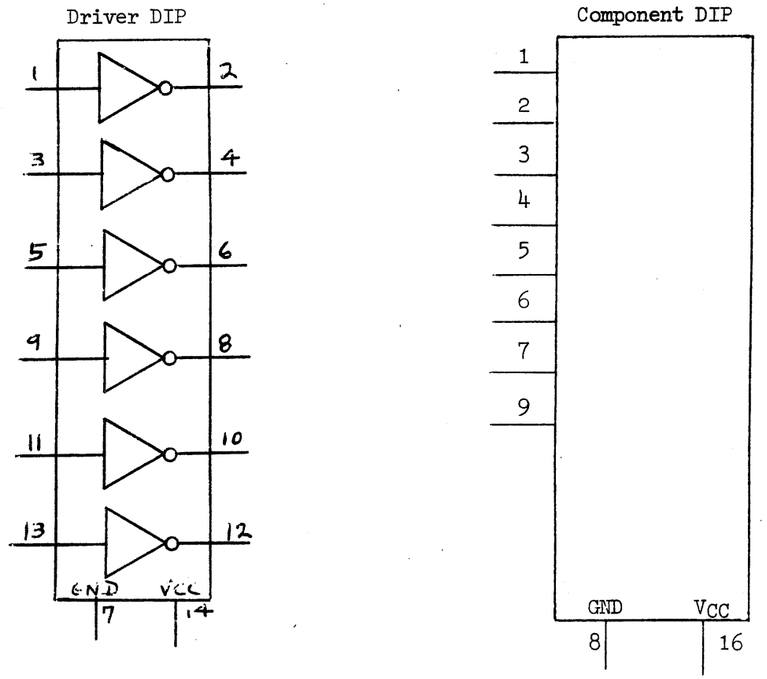


Figure 4

Several of the lines connected to the device interface have had special capabilities added in the hope of assisting the controller designer. These are described below.

(1) The parity (odd) of byte 5 and bit 7 of byte 8 (9 bits) is stored in a flip-flop whenever a leading edge is detected on bit 3 of byte 8. However, this flip-flop is not directly visible to the μ-code. The output of this flip-flop is synchronized and then can be tested at the jump net one and one-third cycles after the leading edge of bit 3 of byte 8.

An IAC will clear this flip-flop, and another IAC will set it. One must avoid clearing the flip-flop when there is a possibility of the flop being set by the interface signals to avoid ambiguous results. This is called Device Input Line Strobed Parity Flag in the μ-code documentation and should not be confused with the Device Parity Flag which comes from the Device Multiplexer.

II-04

(2) Bit 3 of byte 8 (the same signal that clocked the parity flip-flop in (1) also sets (on the leading edge) a flip-flop on every occurrence. This FF is also synchronized and can be tested with a jump condition.

An IAC can also set this FF. Another IAC clears it. Thus this jump condition can be used to either sense a positive edge on the interface line or can be used as a u-program settable and clearable flag. This is called Data Line Flag 3 in the u-code documentation.

(3) Bit 2 of byte 8 has an identical set of properties as those described in (2) above. This is called Data Line Flag 2 in the u-code documentation.

(4) Bit 1 of byte 8 has an identical set of properties as those described in (2) above. This is called Data Line Flag 1 in the u-code documentation.

(5) It should be noted that the three lines that set FF's on their leading edges (byte 8, bits 1, 2 and 3) are the output of Schmitt triggers to provide some noise immunity.

(6) In addition to the parity check described in (1) above, parity is also generated on the eight bit output of the Device Multiplexer. This parity (odd) on the eight bits may be staticized by an IAC in the Flag Register. It is testable by a jump command.

(7) If one wishes to obtain the contents of a device register on BR, one simply selects the proper device register as a source.

Two things must be considered when doing this. The first is that information enabled onto the device bus may cause undesirable action in devices connected to these lines. The second is that reflections and noise pickup on any cables connected to the device drivers may cause erroneous inputs to BR. It is recommended that one maintain the device driver outputs for two ROM cycles and only utilize the information on BR during the second ROM cycle to allow transients to die out.

(8) One may also rotate right one bit any of the Device Multiplexer outputs (a byte from a device or from a device register).

4.1.4 The standard PRIME I/O bus interface logic is discussed in detail in Reference A and will not be discussed here. The rest of the MPC I/O interface logic consists of the following parts.

a. An SKS Register
b. An Address Register
c. A Data Register
d. A Mode Line Register
e. An OTA Function Code Register
f. A Read Encoder
g. Some decoding and synchronizing logic

The use of these parts will be discussed in terms of the MPC I/O bus interactions.

The Programmed I/O (PIO) Commands for the MPC cause well-defined actions in the MPC. The defined PIO command set for the MPC is shown in Table II.

OCP's 07, 10, 15, 16 and 17 are used to control the operating modes of the MPC. They cause direct and defined hardware action and are only indirectly linked to u-code action.

OCP Acknowledge Interrupt clears the Interrupt Request F.F. in the MPC.

OCP Set Interrupt Mask sets the Interrupt Mask F.F. in the MPC. This enables the MPC to request interrupts. This F.F. is testable by the u-code. See the I/O bus specification.

OCP Clear Interrupt Mask clears the MPC's Interrupt Mask F.F. This inhibits the MPC from requesting interrupts.

OCP Initialize sets the MPC's Initialize F.F. which is testable by the u-code. The u-code is responsible for initializing most of the MPC. However, certain actions occur immediately in the MPC (i.e., via hardware) as a result of an OCP Initialize. These are:

a. Set MPC Busy
b. Clear Ready (INA and OTA)
c. Clear SKS F.F.'s 2, 3, 5, 6, 7 and 14
d. Places MPC in the PROM timing mode (300 ns cycle)
e. Causes the u-code to jump to location zero, enter RUN mode, and begin execution.
f. Clear Interrupt Mask and Request F.F.'s
g. Clear DMX Request
h. Device Registers 1, 3, 5 and 6
i. Set Initialize Flag

The result of this command is identical to what happens when Master Clear is activated.

OCP Simulator Initialize. This OCP causes the same action as OCP Initialize except that the MPC clock is put in the Simulator Timing Mode with a 400 ns cycle to allow for longer memory access time. See Figure 6-2.

OCP Stop. This command places the MPC in the Single Step Mode. The u-program may then be executed an instruction at a time by depressing the START switch on the Field Engineering Switch Panel (if connected). The only way to get the MPC back into the run mode after this OCP has been executed is via either Master Clear, OCP Initialize, or OCP Simulator Initialize.

OCP 00 sets a unique bit in the SKS Register (directly with no u-code interaction). The u-code can test the state of this bit, and any function that the controller designer wishes may be attributed to this F.F.'s being set (e.g., OCP Rewind on Mag Tape). Note that both the u-code and an OCP may set this F.F., but only the u-code may clear it. None of the other OCP's may be used.

Table II

MPC PIO Commands

Op Code Bits 1-6 / Function Code 7-10	OCP $(14)_8$	SKS $(34)_8$	INA $(54)_8$	OTA $(74)_8$
00	Set SKS 14 F.F.	Skip if INA Ready Set (H)	Input Data Register (H)	Output Data (H)
01		Skip if Not Busy (H)	--	MBS
02		MBS Skip if Set	--	MBS
03		MBS Skip if Set	--	MBS
04	--	Skip if Not Interrupting (H)	--	MBS
05	--	MBS Skip if Set	--	MBS
06	--	MBS Skip if Set	--	MBS
07	Simulator Init (H)	MBS Skip if Set	--	MBS
10	Stop Clock (H)		--	MBS
11	--		--	MBS
12			--	MBS
13			--	MBS
14	Acknowledge Interrupt (H)	MBS Skip if Set	--	Output DMA/C Channel Number
15	Set Interrupt Mask F.F. (H)		--	MBS
16	Clear Interrupt Mask F.F. (H)		--	Output Interrupt Vector
17	Initialize (H)		--	MBS

MBS = May be specified by the control designer
-- = Not defined (not useable)
H = Implemented in Hardware

Three SKS's are pre-assigned in the MPC. These allow the computer programmer to test the state of the MPC. These are SKS INA Ready, SKS Not Busy, and SKS Not Interrupting. Five other SKS's test the state of F.F.'s in the MPC which may only be set or cleared by the u-code and thus they may be assigned any meaning by the controller designer. These five SKS's skip if set. These SKS's are loaded by specifying the SKS Left Byte as the destination. SKS'02 is loaded by bit 3, SKS'03 is loaded by bit 4, etc. In addition, SKS'14 is set by OCP'00 and loaded from the u-code by bit 7 of the SKS register Right Byte.

Only one INA is implemented in the MPC (INA '00). This is Input the contents of the Data Register (in the MPC) to the CP's A register. The A register will always be cleared prior to being loaded. This INA will only be accepted by the MPC (respond Ready) if the u-code has set the INA Data Ready F.F. by an IAC. (MPC Busy has no effect on Ready.) Otherwise, the INA will not skip. The strobe resulting from an accepted INA 00 will clear the MPC's INA Data Ready F.F.

The controller designer must see that the proper information is placed into the Data Register before he sets INA Ready. This may be coordinated for various kinds of transfers (i.e., status, data, vectors, etc.) by utilizing other PIO commands to tell the MPC u-code which kind of transfer the next INA will expect.

One OTA ('00) operates in a way analogous with INA 00. That is, the MPC will only respond ready to this OTA if the u-code has set OTA Data Ready F.F. via an IAC. Otherwise the OTA will not skip. Busy has no effect on this OTA Ready condition. The strobe resulting from an accepted OTA 00 will clear the MPC's OTA Data Ready F.F.

In all other respects it is identical to the other OTA's.

All other OTA's will only be accepted by the MPC if the MPC Busy Line is false. If an OTA is accepted by the MPC, the function code of the OTA will be stored in the OTA Function Code Register by the strobe. The OTA Flag F.F. will be set. The OTA Flag will set the MPC Busy Line, thus preventing further OTA's from being accepted. The u-code alone may clear the OTA Flag F.F.

The data that accompanies the OTA (from the CP's A register) is stored into the MPC Data Register. It is the responsibility of the controller designer to preserve and interpret this data.

The data that is transferred as a result of an accepted OTA '14 must be interpreted by the controller designer as the DMA/C channel information or not used as defined in the I/O bus spec.

Likewise, the data transferred as a result of an accepted OTA '16 must be interpreted by the controller designer as the Interrupt Vector or not used as defined in the I/O bus spec.

The other 13 OTA's may be used for any purpose whatsoever. When any of these OTA's is accepted by the MPC, the Function Code is stored in the Function Code Register, the Data is stored in the Data Register and the OTA Flag and, consequently, the Busy Line are set.

II-06

The Mode Lines must all be correct during any I/O bus operation (see the I/O bus spec). Consequently the MPC forces the Mode Lines to the proper state (zeroes) during PIO operations. This is done independently of the Mode Line Register. The contents of the Mode Line Register are not altered by the MPC. The Mode Register is only used during DMX operations. For PIO and Interrupts the Mode Lines are always put in the correct state by the MPC.

4.1.5 DMX Operation

The MPC provides full DMX functionality. Either DMA, DMC or DMT transfers can occur. To perform a DMX transfer the u-coder goes through a sequence of operations as follows:

a. Test the DMX Request F.F. (end-of-data phase). If it is set, the previous transfer is still taking place. Wait, or time out.

b. When it is clear, test the End of Range F.F. If it is set, the transfer is complete. Exit this routine and clear the EOR F.F.

c. If it is not set, load the Mode Line Register (if needed); load or increment the Address Register with either the channel number in the case of a DMA or DMC transfer or a memory address in the case of a DMT transfer. Load the Data Register in the case of an input transfer or store the data that was just received in the case of an output transfer.

d. Set the DMX request F.F.

e. Go back to (a).

4.1.6 Interrupt Operation

The MPC provides standard (compatible) mode and vectored interrupt capability. The Interrupt Mask F.F. is controlled directly by OCP's 15 and 16.

The sequence of operations used to request an interrupt is to:

a. Test the Interrupt Request F.F. If it is set, an interrupt is still pending. Typically wait.

b. When the F.F. is clear, load the address register with the vector, and set the Interrupt Request F.F.

c. Test the Interrupt Request F.F. When it is cleared the interrupt has been honored and the Acknowledge Interrupt OCP has been issued.

Note that neither Override Inhibit Interrupts nor Memory Increment capability are provided by the MPC. Also note that the Mode Lines are forced to the correct state by the MPC and the Mode Register should only be used for DMX.

4.2 The Arithmetic Logic Unit (ALU) may perform any one of 32 arithmetic and logical functions on eight bit words. (See the u-code description.) One can also test the Logical Value of any one of the eight bits. The output of the ALU forms the Transmit Bus. The two inputs to the ALU are the Receive Bus (input B) and the output of the Register File (input A). Note that the ALU cannot perform the same operations on either input variable.

4.3 The Register File is a memory used for temporary storage. It is volatile so that its contents are lost when power is turned off. The Register File locations may be written into or read out of. The file is organized as 32 eight bit words. The cell addresses are 0 through 31.

4.4 The outputs of the Read Only Memory control the enabling of the data paths within the MPC on every ROM memory cycle. The ROM contains programs and subroutines which execute very much like machine language programs. The ROM is addressable up to 1024 (256 or 512 are on the MPC board) 48 bit words. The output of the ROM is stored in RCM to provide stable glitch-free outputs.

These u-programs are unique to each type of MPC based controller. They are programmed into the MPC by "blowing" Programmable Read Only Memory (PROM) DIPs, and inserting these onto the MPC board. The ROM Address MUX and RCY acts as a memory interface to the ROM, specifying the address of the next ROM location to be fetched and providing the appropriate timing. The Pushed RCY Register provides a one deep stack for fast subroutining. One may Push onto Pushed RCY or Pop Off of Pushed RCY via independent action codes in the u-code.

4.5 The Flag Register bit assignments are as follows:

	Bit	
(MSB) Right Byte	1	Device Parity F.F. (Even) for Input
	2	Interrupt Mask F.F.
	3	DMA/C End of Range F.F.
	4	Normal Mode F.F.
	5	Strobed Device Parity F.F. (Odd) for Input
	6	Device Input Line F.F. 1 Trailing Edge Trigger
	7	Device Input Line F.F. 2 Trailing Edge Trigger
	8	Device Input Line F.F. 3 Leading Edge Trigger
Left Byte	1	Initialize Flag
	2	OTA Flag
	3	SKS Flag (Set by OCP'00, Loaded by SRR, bit 7)
	4	u-Code Busy Flag
	5	INA (00) Data Ready Flag
	6	DMX Request Flag
	7	Interrupt Request Flag
	8	OTA (00) Data Ready Flag

4.6 Two additional flags (5 and 6) are provided. These may be set or cleared by IAC and their condition tested by a jump.

The Busy condition which determines the acceptance of OTA's is the Logical OR of the u-code Busy Flag, the Initialize Flag, and the OTA Flag.

II-07

5.0 MPC u-CODE SPECIFICATION

The u-code resides in the ROM. The ROM is addressable up to 512 words. A ROM word is 48 bits long and is organized into fields, each of which performs a specific function. It is useful to note that the definitions and actions of the u-code relate directly to the structure of the MPC hardware and Figure 2 will be a visual aid that the u-coder needs in order to write u-code for the MPC.

5.1 u-Code Format

The basic u-code format is shown in Figure 5. The detailed u-code description is given in Table III. A complete description of these fields follows.

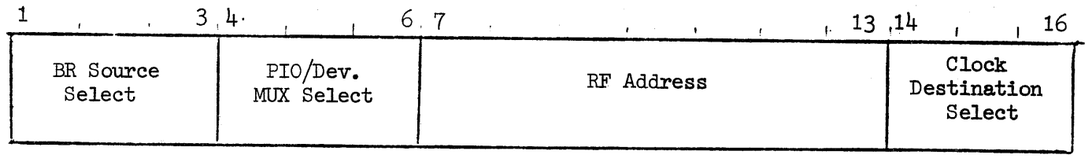
The u-code word is read out of ROM from the addressed location and stored in the RCM register on every ROM cycle. It is the output of the RCM that controls the MPC.

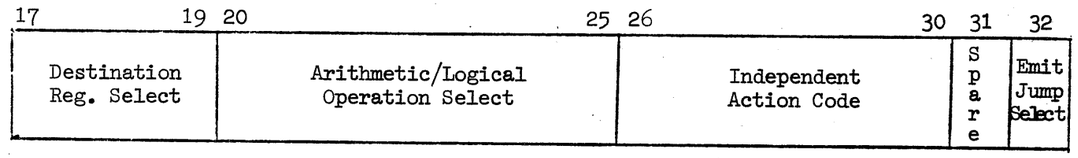
5.2 MPC u-Code Assembler

A description of the MPC u-code Assembly Language Syntax is given in section 7.9.

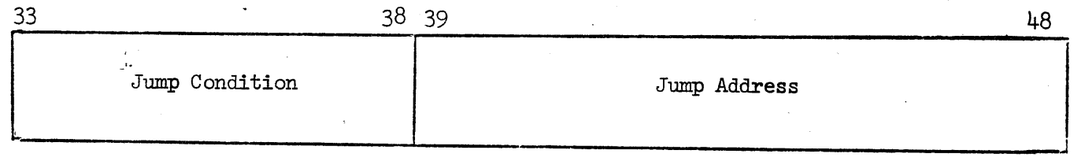
A summary of the mnemonics used is given in Table III.

Figure 5. μ-Code Format

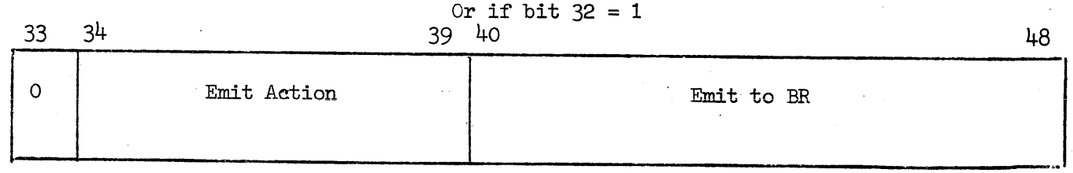




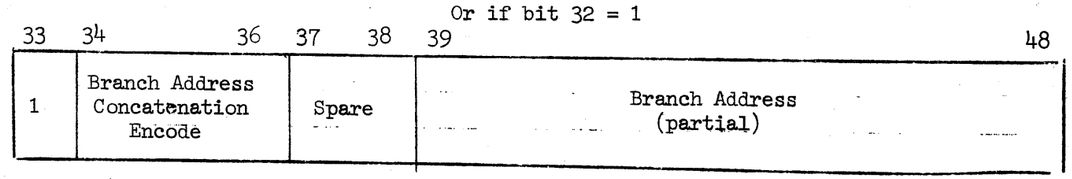
If bit 32 = 0



Or if bit 32 = 1



Or if bit 32 = 1



ROM Bits	Field	Description	Mnemonic(s)
1-3	A	Receive Bus (BR) Source Select	
		0 RCM (bits 41 thru 48)	RCM
		1 Dev MUX	See Field B
		2 PIO MUX	See Field B
		3 Flag Reg-R.B.	FRR
		4 Dev MUX Rotated Right	See Field B
		5 Dev Reg 7	DR7
		6 --	-
		7 Flag Reg-L.B.	FRL

ROM Bits	Field	PIO and Dev MUX Source Select	Dev MUX	PIO MUX	Dev MUX Rotated Right
4-6	B	0 Dev Byte 1 and Data Reg R.B.	DR1	DRR	DR1RR
		1 Dev Byte 2 and Add Reg R.B.	DR2	ARR	DR2RR
		2 Dev Byte 3 and SKS Reg R.B.	DR3	SRR	DR3RR
		3 Dev Byte 4 and Mode Reg*	DR4	MR	DR4RR
		4 Dev Byte 5 and Data Reg L.B.	DR5	DRL	DR5RR
		5 Dev Byte 6 and Add Reg L.B.	DR6	ARL	DR6RR
		6 Dev Byte 7 and SKS Reg L.B.	DB7	SRL	DB7RR
		7 Dev Byte 8 and OTA F.C. Reg**	DB8	OFC	DB8RR

*

1	2	3	4	5	6	7	8
BPCSS01	BPCSS02	BPCSS03	INMOD (4)	0	1	2	3
Slot #				Mode Reg			

**

1	2	3	4	5	6	7	8
BPCEXS1	BPCEXS2	99	00	7	8	9	10
Slot #		High Order Address Bits		OTA Function Code Bits			

Note: R.B. = Right Byte (bits 9-16)
L.B. = Left Byte (bits 1-8)

TABLE III. u-Code Description - Assembly Language Mnemonics

Table III. u-Code Description (continued)

ROM Bits	Field	Description	Mnemonic
7-13	C	Reg File (RF) Address	RF nn
		If bit 7 = 0 the RF Address comes from bits 9-13. Bit 8 is not used.	
		If bit 7 = 1 the RF Address comes from the low order 6 bits of Dev Reg 7. (8-13 are not used.)	
14-16	D	Clock Destination Register Select (always Clock RCM)	
		No Bit set - Clock RCM Only	-
		Bit 14 set - Clocks PIO Registers	See Field E
		Bit 15 set - Clocks Dev Registers	See Field E
		Bit 16 set - Clocks R.F.	RF nn
		Any combination of bits is valid.	

ROM Bits	Field	Destination Register Select	Dev Reg	PIO
17-19	E	0 Data Reg R.B.	-	DRR
		1 Dev Reg 1 & Add Reg R.B.	DR1	ARR
		2 Dev Reg 2 & SKS Reg R.B.	DR2	SRR
		3 Dev Reg 3 & Mode Reg	DR3	MR
		4 Dev Reg 4 & Data Reg L.B.	DR4	DRL
		5 Dev Reg 5 & Add Reg L.B.	DR5	ARL
		6 Dev Reg 6 & SKS Reg L.B.	DR6	SRL
		7 Dev Reg 7 OTA FC & HO Add*	DR7	OFC

*

1	2	3	4	5	6	7	8
		99	00	7	8	9	10
Not Used		High Order Address Bits		OTA Function Code Bits			

ROM Bits	Field	Description
20-25	F	Arithmetic Operation
20-21		0 Arithmetic Mode C = 0
		1 Arithmetic Mode C = 1
		2 Arithmetic Mode C = Carry Out of ALU
		3 Logical Mode

Note: C = Carry into low order ALU bit

II-09

Table III. u-Code Description (continued)

ROM Bits	Field	Description	Mnemonic
20-25	F	Arithmetic and Logic Unit Logical Modes	
		0 A	RF nn
		1 A∧B	AND
		2 A∧B--*	-
		3 0	CON ZERO
		4 AVB	OR
		5 B	BR
		6 A∀B	XOR
		7 A--∧B	-
		8 AVB--	-
		9 --(A∀B)	-
		A --B	BRN
		B --(AVB)	NOR
		C 1 (Logical)	ONE
		D A--VB	-
		E --(A∧B)	-
		F --A	RFN nn
		- - - OR - - -	
		Arithmetic Codes (add 1 if Carry = 1)	
		0 A	INC RF nn (C = 1)
		1 (A∧B--)+A	-
		2 A∧B+A	-
		3 A+A=2A	LS
		4 AVB	-
		5 (AVB--)+(AVB)	-
		6 A+B	PLUS
		7 A+(AVB)	-
		8 AVB--	-
		9 A-B-1	MINUS (C = 1)
		A (A∧B)+(AVB--)	-
		B A+(AVB--)	-
		C -1	CON MINUS 1
		D (A∧--B)-1	-
		E A∧B-1	-
		F A-1	DEC RF nn

Note: -- ≡ NOT
A = Register File
B = Receive Bus

Table III. u-Code Description (continued)

ROM Bits	Field	Description	Mnemonic
26-30	G	Independent Action Codes	
		0 Jump to Zero	JZ
		1 Pop (Jump to Pushed RCY)	POP
		2 Push (Load Pushed RCY with RCY)	PSH
		3 +1 to Add Reg	IAR
		4 +1 to Dev Reg 7	IDR7
		5 Set INA Data Ready F.F. (INA00)	SIRDY
		6 Clear u-Code Busy F.F.	CB
		7 Clear OTA Flag	COF
		8 Clear Initialize Flag	CIF
		9 Set OTA Data Ready F.F. (OTA00)	SORDY
		A	-
		B Load Input Dev Parity into Flag	SIPF
		C Set Flag 5	SF5
		D Set Flag 6	SF6
		E Set u-Code Busy F.F.	SB
		F Store Condition Codes	SCC
		10 NOP	NOP
		11 Set Interrupt Req F.F.	SIRQ
		12 Set DMX Req F.F.	SDRQ
		13 Clear DMX EOR F.F.	CDEOR
		14 Clear Dev Line Input Flag 1	CF1 (CD11)
		15 Clear Dev Line Input Flag 2	CF2 (CD12)
		16 Clear Dev Line INput Flag 3	CF3 (CD13)
		17 Clear Dev Line Parity Flag	CF4 (CDIP)
		18 Set Dev Line Input Flag 1	SF1 (SDF1)
		19 Set Dev Line Input Flag 2	SF2 (SDF2)
		1A Set Dev Line INput Flag 3	SF3 (SDF3)
		1B Set Dev Line Input Parity Flag	SF4 (SDFP)
		1C Clear Flag 5	CF5
		1D Clear Flag 6	CF6
		1E Clear DMX Req F.F.	CDRQ
		1F	-
31		Spare	
32	H	Emit/Jump Select	
		0 = Conditional Jump implies that field J is the Jump condition and field K is the Jump Address	JUMP ON (NOT)
		1 = Emit or Branch. Bits 33-48 are further decoded.	EMIT, BRANCH

II-10

Table III. u-Code Description (continued).

ROM Bits	Field	Description	Mnemonic
33-38	J	Jump Conditions (if bit 32 = 0)	
33		If bit 33 is a 0, the MPC will Jump when the addressed condition is False. If bit 33 is a 1, the MPC will Jump when the addressed condition is True.	
34-38		0 True	T
		1	-
		2 SKS Flag (OCPoo)	SKSF
		3 OTA Flag	OTAF
		4 OCP Initialize Flag	INTF
		5 Carry Out of Add Reg bit 1	ARCO
		6	-
		7 OTA Data Ready Set (OTA00)	ORDY
		8 Bit Test Condition Code SEt	CCBT
		9	-
		A	-
		B	-
		C	-
		D	-
		E	-
		F	-
		10 Carry Out of Dev Reg 7 bit 1	DR7CO
		11 Flag 5	F5
		12 Flag 6	F6
		13 Carry Out of ALU Cond. Code Set	CCACO
		14 ALU High Order bit Cond. Code Set	CCAHO
		15 DMX Req Set (end of Data phase)	DRQDP
		16 Interrupt Req Set (before mask)	IRQ
		17 INA Data Ready Set (INA00)	IRDY
		18 Input Parity Even	PIE
		19 ALU equal zero Cond. Code Set	CCAEZ
		1A End of Range	EOR
		1B ALU less than or equal zero Cond. Code Set	CCALEZ
		1C Device Input Line Flag 1 Set	F1 (DIF1)
		1D Device Input Line Flag 2 Set	F2 (DIF2)
		1E Device Input Line Flag 3 Set	F3 (DIF3)
		1F Device Input Strobed Parity Flag Set	F4 (DIFP)

Table III u-Code Description (continued)

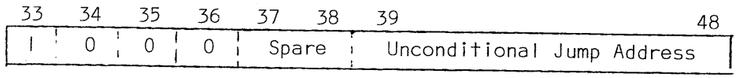
ROM Bits	Field	Description	Mnemonic
39-48	K	Jump Address from ROM (if Bit 32 = 0)	
33-48	J'	If ROM bit 32 = 1, then bits 33-48 are decoded as:	
33		If Bit 33 = 0, Emit or Bit 33 = 1, Unconditional Branch If Emit, ROM Bits 34 thru 48 are interpreted af follows: 33 34 39 40 41 48 0 | Emit Action (see below) | Emit P | Emit to BR Emit P (Parity) is not implemented in current MPC.	
34-39	J'	Emit Action if bit 32 = 1 and bit 33 = 0 bits 34-39 are interpreted as follows:	
34-36		Set the Bit Test Condition Code if bit at the output of the ALU specified by these 34-36) ROM bits is a one, and the Set Condition Code Independent Action Code is given:	
		0 ALU bit 1 = 1	ACT 1
		1 ALU bit 2 = 1	ACT 2
		2 ALU bit 3 = 1	ACT 3
		3 ALU bit 4 = 1	ACT 4
		4 ALU bit 5 = 1	ACT 5
		5 ALU bit 5 = 1	ACT 6
		6 ALU bit 7 = 1	ACT 7
		7 ALU bit 8 = 1	ACT 8

II-11

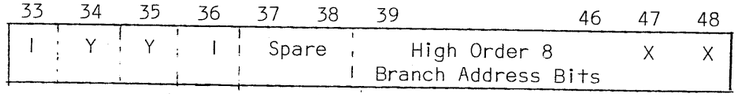
Table III. u-Code Description (continued)

ROM Bits	Field	Description
37-39		TBS

If Bit 33 = 1, RCM Bits 34-48 are interpreted as follows:

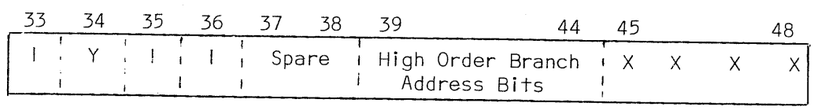


or Four Way Branch



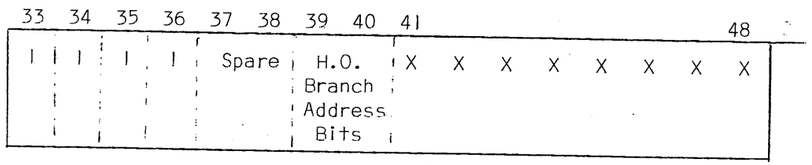
The Low order two address bits come from BR

or Sixteen Way Branch



Low order four address bits come from BR

or 256 Way Branch



Low order 8 address bits come from BR

In other words, bits 34, 35 and 36 specify which Branch address bits come from RCM and which from BR. If the Y bits are not zero, then other concatenations of address occur. These are shown in Table IV.

Table IV

RCM Bits			Branch Address Bit Source			
34	35	36	High Order	--	--	Low Order
0	0	0	RCM 39, 40	RCM 41-44	RCM 45, 46	RCM 47, 48
0	0	1	RCM 39, 40	RCM 41-44	RCM 45, 46	BR 7, 8
0	1	0	RCM 39, 40	RCM 41-44	BR 5, 6	RCM 47, 48
0	1	1	RCM 39, 40	RCM 41-44	BR 5, 6	BR 7, 8
1	0	0	RCM 39, 40	BR 1-4	RCM 45, 46	RCM 47, 48
1	0	1	RCM 39, 40	BR 1-4	RCM 45, 46	BR 7, 8
1	1	0	RCM 39, 40	BR 1-4	BR 5, 6	RCM 47, 48
1	1	1	RCM 39, 40	BR 1-4	BR 5, 6	BR, 7, 8

II-12

6.0 MPC TIMING

Figure 6-1 shows the internal timing of the MPC. TCLKC+, TCLKL, TCLK1, TCLK2, and TCLK3 are the basic timing pulses. All other timing signals are derived from these.

6.1 Detailed Timing Description

The first clock (TCLK1) performs the following:

(a) Load RCM at the beginning of every cycle (300 ns). This staticizes the ROM output and changes the control outputs throughout the MPC.

(b) RCY is loaded on every cycle.

(c) Staticize all inputs to the jump net.

6.2 The second clock (TCLK2) performs the following functions:

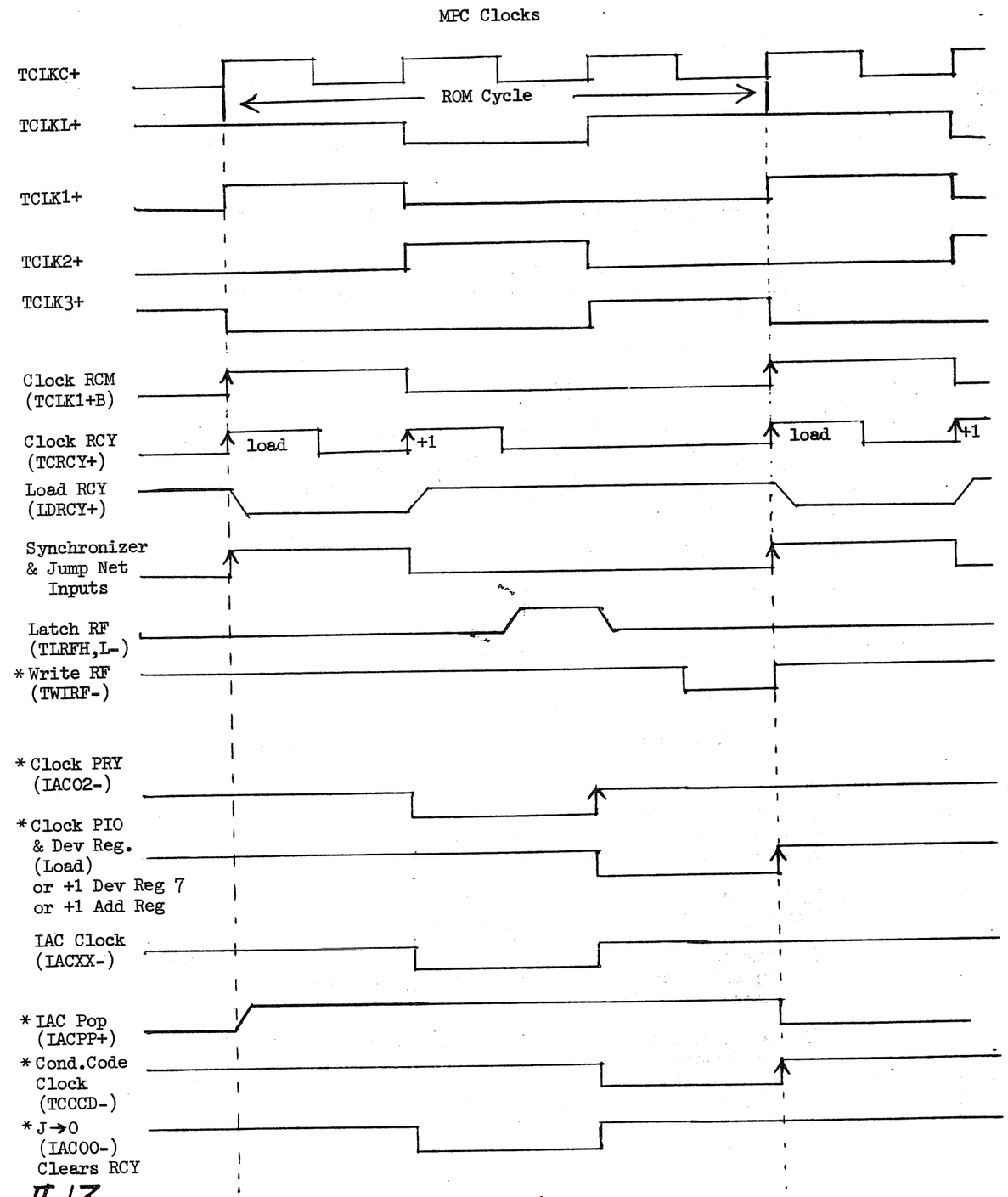
(a) It increments RCY (adds one to RCY). This is so that the next sequential ROM cell will be accessed on the next cycle if no jumps or branches occur.

(b) The independent action code (IAC) (if any) output appears. Some IAC's cause action at this time. Others cause action on the trailing edge of the IAC output. Still others have other timing. These are shown separately in Figure 6-1.

The middle of the second clock causes the Register File output to occur and these are latched at the end of the second clock time.

The middle of the third cycle causes:

(a) The end of the IAC output causing it to go high (if there was one). This would cause the Pushed RCY Register (PRY) to be loaded if that IAC had been coded.



II-13

Figure 6-1

* Does not happen every cycle.

LTR	DATE	REVISION	DR.	CK.

(b) The Register File cell that was addressed will be written into if that was specified as a destination register.

The end of the third cycle (at the same time as the beginning of the next cycle) causes:

(a) The device Register and/or the PIO Register to be loaded if they are specified as destination registers.

(b) Device Register 7 or the Address Register will be incremented if that IAC is specified. This will occur instead of that Register being loaded if both are specified.

Then the process repeats.

6.3 ROM Simulator and External ROM Timing

If the PROM/SIM switch in the clock logic is placed in the SIM position or OCP Simulator Initialize is issued the basic clocks in the MPC change as shown in Figure 6-2. That is, TCLK3 doubles in length. This allows 100 nanoseconds more time for the external memory access (ROM, PROM, ROM Simulator, or Writable Control Store) to take place.

Thus the MPC's cycle time increases 33%.

6.4 Single Cycle Timing

If the RUN/Single Cycle switch in the clock logic is placed in the Single Cycle position, or OCP Stop Clock has been issued, the basic clocks in the MPC suspend every cycle until the Start switch is pushed. The clock suspends at the end of TCLK2. See Figure 6-3.

7.0 MPC μ-PROGRAMMING RULES

7.1 Register Loading

(a) If both the loading of a Register (as a destination) and the incrementing of a Register are specified, the Register will be incremented by one (the previous contents of the register +1) not loaded.

(b) Up to three destination registers may be specified in the same μ-instruction. However, there are certain restrictions.

(1) The same cell in RF whose address is specified will be loaded if RF is specified as a destination register. The output of RF will not change until the middle of the next cycle.

(2) A device register and a PIO register may be loaded if both are specified as destination registers but only in pairs as specified in field E. For example, one could not load both Device Register 3 and the SKS Register with the same μ-instruction.

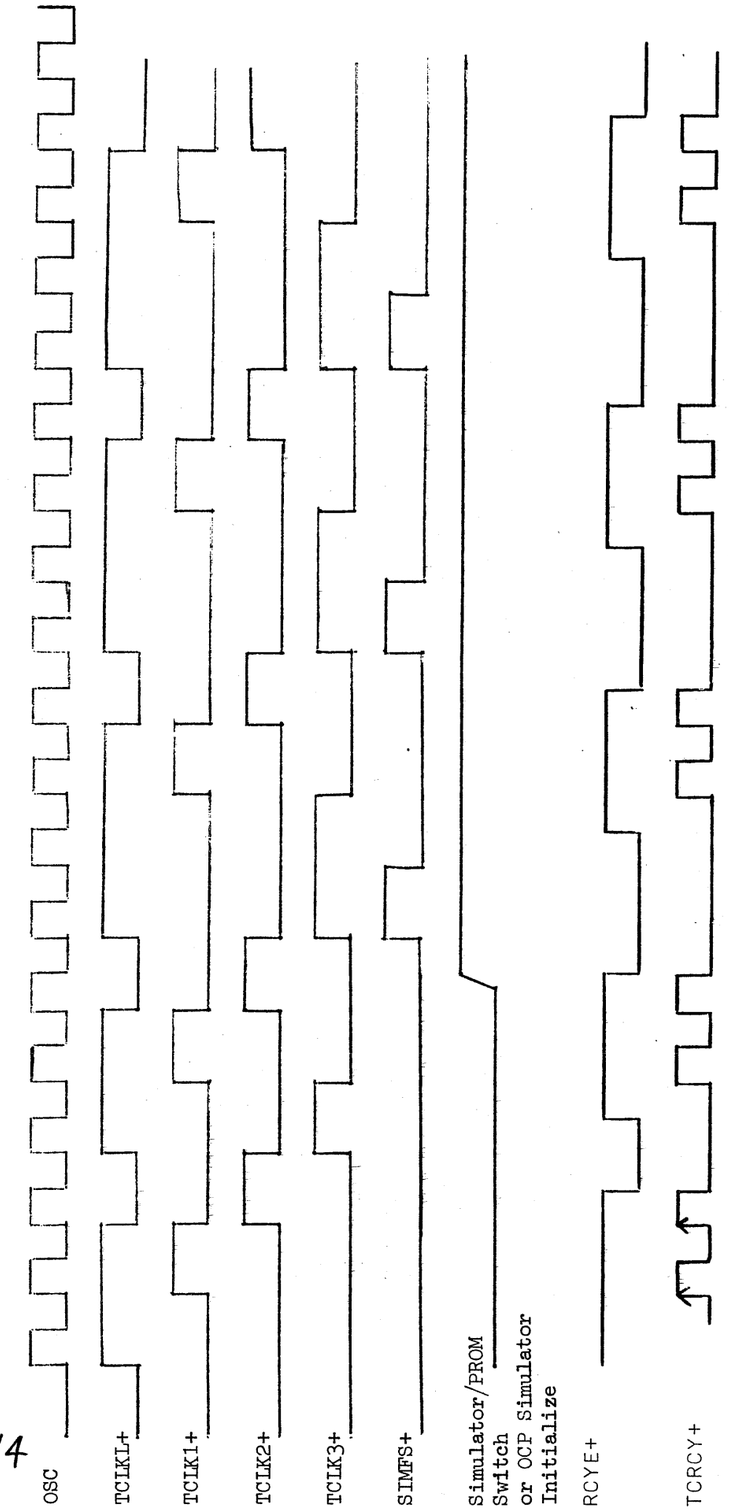


Figure 6-2 MPC Simulator Mode Timing

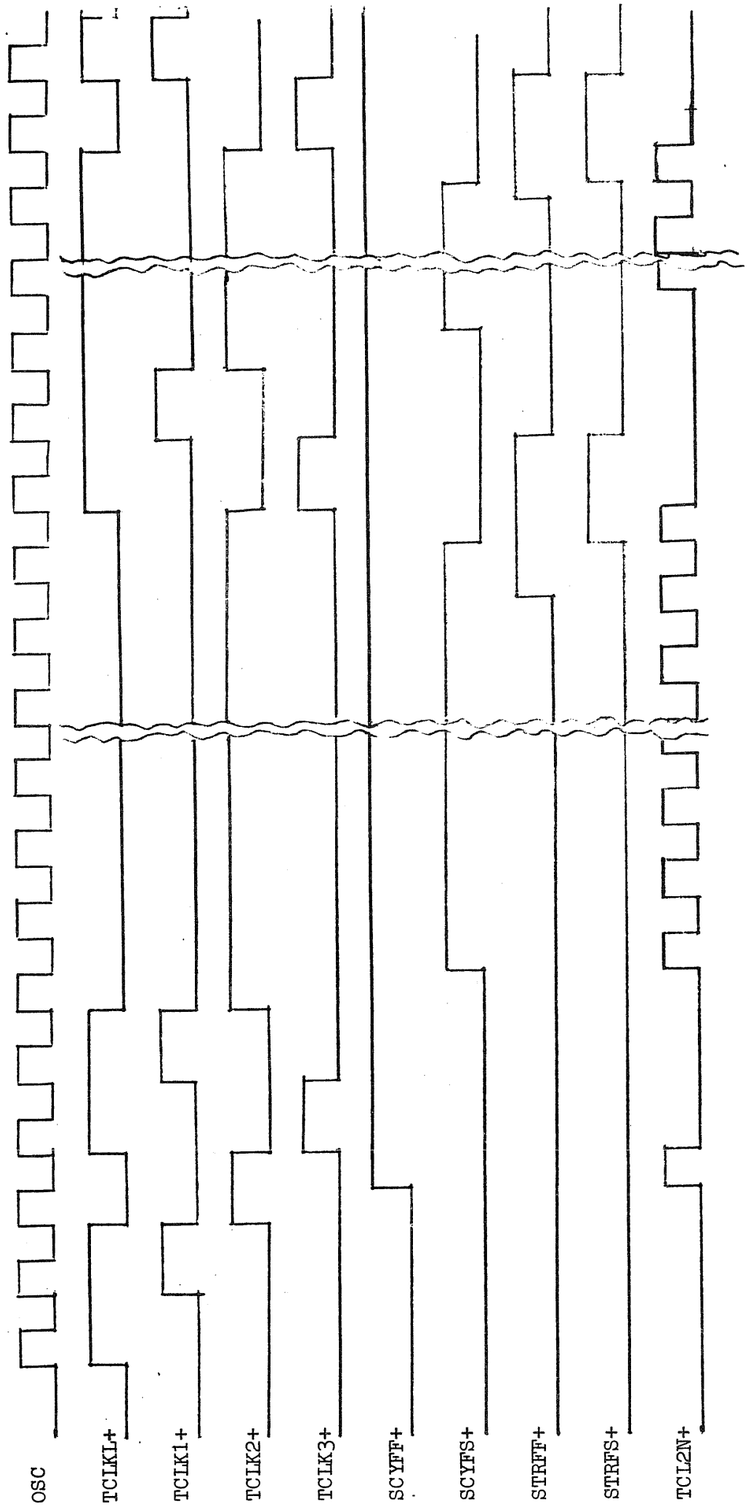


Figure 6-3 MPC Single Step

LTR	DATE	REVISION	DR.	CK.

7.2 Source Select

(a) If either the Device MUX or the PIO MUX is specified as the source of BR, field B must specify what the specific BR source is.

7.3 Arithmetic Operations

(a) The A input of the ALU is always the output of the Register File. The B input of the ALU is always what is enabled to BR.

(b) The carry out of the high order nibble (4 bits) of the ALU, the high order bit of the ALU, the ALU equals zero condition, and the result of the bit test will only be stored in the condition code register at the end of the current cycle if the set condition code IAC is specified in the current instruction. Any of the condition codes may then be tested any number of times in any succeeding cycles, including the next cycle in which the set condition code IAC is specified.

7.4 Independent Action Codes

(a) Only one IAC may be specified in any single u-instruction.

(b) All IAC's will take precedence over any conflicting commands except for the jump to zero IAC (see (c)).

(c) The jump to zero IAC will not take place if a conditional jump is successful or if a branch is specified, i.e., ROM bit 32 = 1 and ROM bit 33 = 0. If a condition jump is not successful, no branch is specified and a jump to zero IAC is specified, the jump to zero will occur.

(d) IAC's Clear OTA Flag and Clear Initialize Flag should only be done when u-code Busy is set.

7.5 Conditional Jumps

The jump net starts to stabilize at the beginning of a ROM cycle. It must result in a stable address to the ROM by the end of the cycle. Thus no jump net inputs are allowed to change after the beginning of a cycle and must remain stable through the end of the cycle.

This implies that either the tested condition is synchronous with the MPC or that a synchronizer is used. Where a two F.F. synchronizer is used, up to one and one-third ROM cycle of delay can result from the time that the condition occurs until it can cause a jump.

The worst case condition is that the condition to be tested just comes up at the first FF clock of the synchronizers (i.e., the end of the second clock). The end of third clock sets the second FF and starts the jump net settling. The settling takes one cycle. If the condition being tested occurs any later than the end of the second clock in the instruction prior to the jump u-instruction, the MPC will not jump. See Figure 6-4.

Location zero of the u-code can not contain a jump on (not) Initialize Flag.

II-15

USED ON	SCALE	SIZE	DWG. NO.	REV.
NEXT ASSY	SHEET 30 OF 40	A	SPC1409	I

PDF-005

LTR	DATE	REVISION	DR.	CK.

The conditions that must by synchronized in this manner are:

```
1
2    SKS Flag (OCP 00)
3    OTA Flag
4    OCP Init Flag
7    OTA Ready Set (00)
15   DMX Req Set (Data Phase)
16   Interrupt Req Set
17   INA Ready Set (00)
18   Input Parity Even
1A   End of Range Set
1C   Device Input Line 1 Flag Set
1D     "      "     " 2   "   "
1E     "      "     " 3   "   "
1F     "      "     Strobe Parity Flag Set
```

IAC's affecting the above conditions will be guaranteed to cause the action so that the synchronizer will work on the current cycle, i.e., before the end of the second clock.

All of the other jump conditions are the result of a previous u-code operation and are guaranteed to be stable in the jump net.

In summary, in order to jump on a condition that must be synchronized by the MPC (listed above) the condition must have occurred prior to clock three of the previous cycle.

All other conditions may be the results of the previous u-instructions. Thus:

```
Add +1 to RF7 - Set Condition Code
Jump on Condition Code Set

IAC Clear OTA Flag
Jump on OTA Flag
Jump* -2
```

are valid u-instruction sequences.

7.6 (a) If an Emit is specified it will only be used if RCM is specified as the source of BR.

(b) In order to set the Bit Test Condition Code it is necessary to specify which bit is to be tested in the Emit Action Field. That is, one must specify the ALU operation to be performed, specify a Set Condition Code IAC, and specify an Emit and an Emit Action in the same u-instruction. The jump on the bit test condition code can be done in any subsequent u-instruction.

7.7 The various branches that may be performed will always take place instead of a jump to zero IAC if both are specified in the same u-instruction.

7.8 When writing a C.P. program using the MPC, one must take care not to cause the MPC to alter the Address Register or the Mode Register prior to completing an interrupt or DMX transfer.

USED ON	SCALE	SIZE	DWG. NO.	REV.
NEXT ASSY	SHEET 31 OF 40	A	SPC1409	1

PDF-005

7.9

GENERAL

THIS DOCUMENT DESCRIBES THE LANGUAGE PROCESSED BY THE MPC MICRO-CODE ASSEMBLER. THE ASSEMBLER CONSISTS OF A MACRO PACKAGE THAT ALLOWS THE STANDARD PRIME ASSEMBLER TO ASSEMBLE CODE FOR THE MICROPROGRAMMED CONTROLLER.

APPLICABLE DOCUMENTS

-- MICRO-PROGRAMMED CONTROLLER PRODUCT SPECIFICATION (PE-T-53)
-- PRIME MACRO ASSEMBLER MANUAL (SECTIONS ONE AND TWO)

DESCRIPTION LANGUAGE ELEMENTS

A MODIFIED BNF TYPE LANGUAGE IS USED IN THIS DOCUMENT TO DESCRIBE THE MPC ASSEMBLY LANGUAGE. THE BASIC FORMS OF THIS LANGUAGE ARE:

[...]	ITEMS BETWEEN SQUARE BRACKETS ARE OPTIONAL
↑	SEPARATES ALTERNATIVE CHOICES
<...>	METASYMBOL, ITEM FITTING THE DEFINITION OF THE SYMBOL MUST BE SUBSTITUTED FOR THE SYMBOL
<...> ::	METASYMBOL DEFINITION.

PROGRAM STRUCTURE

PROGRAMS WILL BE STRUCTURED AS FOLLOWS:

-- INTRODUCTORY COMMENTS
-- AN INSERT STATEMENT OF THE FOLLOWING FORM:
   $INSERT MPC
   THIS STATEMENTS DIRECTS THE ASSEMBLER TO READ AND STORE THE MPC MACRO DEFINITIONS
-- VERSION IDENTIFICATION (SEE IDNT MACRO DESCRIPTION)
-- PROGRAM TEXT (SEE MPC AND ORG MACRO'S)
-- END STATEMENT

IDNT MACRO

THE IDNT MACRO IS USED TO SPECIFIY A VERSION IDENTIFICATION. THE FORMAT OF AN IDNT MACRO CALL IS:

```
IDNT   (<STRING>),(<STRING>),...
```

<STRING> := A STRING OF UPTO 30 CHARACTERS. THE STRING MAY NOT INCLUDE SEMICOLONS (;), COLONS (:), OR PARENTHESIS.

UPTO TEN STRINGS MAY BE SPECIFIED, BUT ONLY THE FIRST 60 CHARACTERS WILL BE USED. THE IDENTIFICATION PRODUCED WILL CONSIST OF THE STRINGS (EXCLUDING SURROUNDING PARENTHESIS) SEPERATED BY SPACES. FOR EXAMPLE,

```
IDNT   (MPC TEST),(   JULY 5, 1973)
```

WOULD PRODUCE THE IDENTIFICATION:

ID: MPC TEST JULY 5, 1973

ORG MACRO

THE ORG MACRO IS USED TO CHANGE THE ASSEMBLER PROGRAM COUNTER. THE FORMAT IS:

```
ORG    <EXP>
```

WHERE <EXP> IS EVALUATED AS AN ASSEMBLER EXPRESSION AND USED AS THE NEW PROGRAM COUNTER.

MPC MACRO

THE MPC MACRO IS USED TO SPECIFY ONE MPC INSTRUCTION WORD. THE PERMISABLE FORMS OF MPC MACRO CALLS ARE AS FOLLOWS:

```
<MPC INSTRUCTION> := [<LABEL>] MPC <OPERATION>

<OPERATION> := <BASIC OP> [<IAC> [<ACT SPEC>]]

<BASIC OP> := <TRN OP>↑<ALU OP>↑<INC OP>↑<DEC OP>↑
              <NOT OP>↑<CON OP>↑<NOP OP>

<TRN OP> := TRN <GEN SOURCE> => <GEN DEST>

<ALU OP> := <RF SPEC> <OP> <BR SPEC> [+ <ALUCS>]
            => <GEN DEST>

<INC OP> := INC <RF SPEC> => <GEN DEST>

<DEC OP> := DEC <RF SPEC> => <GEN DEST>

<NOT OP> := NOT <GEN SOURCE> => <GEN DEST>

<CON OP> := CON <CON SPEC> => <GEN DEST>

<CON SPEC> := ZERO↑MINUS1

<NOP OP> := NOP

<GEN SOURCE> := <RF SPEC>↑<BR SPEC>↑ NOP

<RF SPEC> := RF <EXP>↑RFDR7

<BR SPEC> := <DM SPEC>↑<PM SPEC>↑<RCM SPEC>↑
             FRR↑<DMRR SPEC>↑DR7↑FRL↑
             0↑1↑2↑...↑7

<DM SPEC> := DR1↑DR2↑DR3↑DR4↑DR5↑DR6↑
             DB7↑DB8

<DMRR SPEC> := DR1RR↑DR2RR↑DR3RR↑DR4RR↑
               DR5RR↑DR6RR↑DB7RR↑DB8RR

<PM SPEC> := PM <PMS>↑<PMS>

<PMS> := DRR↑ARR↑SRR↑MR↑DRL↑ARL↑SRL↑
         OFC↑0↑1↑2↑...↑7

<RCM SPEC> := RCM↑RCM = <EXP>

<OP> := <LOGOP>↑<AOP1>↑<AOP2>

<LOGOP> := RF↑AND↑ZERO↑OR↑BR↑XOR↑BRN↑
           NOR↑ONE↑RFN
```

```
<AOP1> :: PLUS↑LS↑SUB↑0↑1↑2↑...↑F                (ALUCS may be specified)

<AOP2> :: MINUS                                  (ALUCS may not be specified)

<ALUCS> :: 0↑1↑C↑L

<GEN DEST> :: NOP↑<DEST SPEC>↑
              (<DEST SPEC>[,<DEST SPEC>[,<DEST SPEC>]])

<DEST SPEC> :: <RF SPEC>↑<DM DEST>↑<PM SPEC>

<DM DEST> ::     DR1↑DR2↑DR3↑DR4↑DR5↑DR6↑
              DR7

<IAC> :: JZ↑POP↑PSH↑IAR↑IDR7↑SIRDY↑CB↑ SF5↑SF6↑
         COF↑CIF↑SORDY↑SIPF↑SB↑SCC↑NOP↑CF5↑CF6↑
         SIRQ↑SDRQ↑CDEOR↑CDI1↑CDI2↑CDI3↑CDRQ↑
         CDIP↑SDF1↑SDF2↑SDF3↑SDFP↑0↑1↑
         2↑...↑1E↑1F

<ACT SPEC> :: <JUMP SPEC>↑<EMACT SPEC>↑
              <BRANCH SPEC>↑<GOTO SPEC>↑
              <NOP SPEC>

<BRANCH SPEC> :: BRANCH TO <EXP> <BTYPE>

<BTYPE> :: 4WAYS↑16WAYS↑256WAYS↑0↑1↑2↑...↑7

<JUMP SPEC> :: JUMP ON [NOT] <CONDITION> TO <EXP>

<GOTO SPEC> :: GOTO <EXP> ↑GO TO <EXP>

<CONDITION> :: T↑  SKSF↑OTAF↑INTF↑ARCO↑ORDY↑CCBT↑
               DR7CO↑      CCACO↑CCAHO↑DRQDP↑IRQ↑
               IRDY↑PIE↑CCAEZ↑EOR↑CCALEZ↑F5↑F6↑
               DIF1↑DIF2↑DIF3↑DIFP↑0↑1↑...↑1E↑1F

<EMACT SPEC> :: EMIT <EXP>↑ACT <EXP>↑
                ACT <EXP> EMIT <EXP>↑
                EMIT <EXP> ACT <EXP>

<NOP SPEC> :: NOP

<LABEL> :: ANY VALID SYMBOL ACCEPTABLE TO PMA.  THE
           DOLLAR SIGN ($) CHARACTER SHOULD NOT BE
           USED TO AVOID CONFLICT WITH INTERNAL
           SYMBOLS USED BY THE MACRO PACKAGE.
```



```
<EXP> :: ANY VALID PMA EXPRESSION.  LABELS ASSIGNED IN THE
         PROGRAM MAY BE USED, AS MAY ASTERISK (*), WHICH
         HAS ITS STANDARD VALUE (THE CURRENT PROGRAM
         COUNTER).  NOTE:  IF THE EXPRESSION CONTAINS ANY
         SPACES (AS ARE REQUIRED FOR PERIOD OPERATORS), THE
         ENTIRE EXPRESSION MUST BE SURROUNDED BY PARENTHESIS.
         EXPRESSIONS MUST NOT EXCEED 30 CHARACTERS IN LENGTH.
```

NOTES

1) IMPLICIT ALU MODE/FUNCTION SELECTION

THE FOLLOWING TABLE LISTS THE ALU MODE/FUNCTION SELECTIONS THAT ARE MADE FOR THE TRN, INC, DEC, NOT AND CON OPERATIONS:

OPERATION	ALU MODE/FUNCTION
TRN, BR SOURCE	$35
TRN, RF SOURCE	$30
INC	$10
DEC	$0F
NOT, BR SOURCE	$3A
NOT, RF SOURCE	$3F
CON, ZERO	$33
CON, MINUS1	$0C

2) <OP> ALU MODE/FUNCTION SELECT

<LOG OP> -- ALU MODE 3 (+ L) <ALUCS> MAY NOT BE SPECIFIED

- RF - $30
- AND - $31
- ZERO - $33
- OR - $34
- BR - $35
- XOR - $36
- BRN - $3A
- NOR - $3B
- ONE - $3C
- RFN - $3F

<AOP1> -- ALU MODE 0 (+ 0) <ALUCS> MAY BE SPECIFIED

PLUS - $06
LS - $03
SUB - $09

<AOP2> -- ALU MODE 1 (+ 1) <ALUCS> MAYNOT BE SPECIFIED

MINUS - $19

3) <RCM SPEC>

THE FORM 'RCM = <EXP>' IS USED TO BOTH SELECT THE BR SOURCE TO BE RCM AND TO SPECIFY AN EMIT FIELD CONSTANT (<EXP>). WHEN THIS FORM IS USED, THE <ACT SPEC> IN THE STATEMENT CAN ONLY BE NOP OR ACT <EXP>, AS ALL OTHERS CONFLICT WITH THE EMIT CONSTANT.

4) THE ASSEMBLY OF THE SOURCE FILE IS ACCOMPLISHED BY COPYING THE MPC MACRO INTO THE USERS UFD AND THEN SIMPLY STATING "PMA FILENAME". THE ASSEMBLER WILL CREATE THE BINARY AND LISTING FILES.

8.0 FIELD ENGINEERING SWITCH PANEL

This panel consists (as far as the MPC's functionality with it is concerned) of three hexadecimal LED displays and three switches. The panel has a cable which plugs into the MPC back-edge connector F.

8.1 The three digit display shows the address of the next μ-program location to be accessed. The switches consist of a Single Cycle/Run switch, a Simulator/PROM switch, and Momentary Start switch.

8.2 The Single Cycle/Run switch, when in the Run position, causes the MPC clock to run continuously. In the Single Cycle position the MPC's clock will execute one cycle and then stop just prior to the end of clock 2 (see Figure 6-3). Thus the jump address, if any, will have been formed and be displayed, the register output will be valid and the output of the ALU will be valid. Depressing the Start switch will cause the MPC to execute the next cycle. When the switch is placed back in the Run position, the Start switch must be depressed if the program is to be continued from where it is or Master Clear must be activated or one of the two OCP Initializes must be issued if the program is to be started from location zero.

8.3 The Simulator/PROM switch determines the memory access timing. See Figure 6-2. When in the PROM position, the timing is set to run with PROM on the MPC board. When in the Simulator position, the third clock is doubled in length; thus adding one hundred nanoseconds to the memory access timing.

9.0 MPC OPERATION WITH A ROM SIMULATOR OR WRITABLE CONTROL STORE OR EXTERNAL ROM

The MPC may be connected to either the Signetics ROM Simulator or the PRIME Writable Control Store Option. This is done via four cables that plug into DIP sites on the MPC. When the μ-program being run is stored in either of these devices, the MPC must be placed in the simulator timing mode either via an OCP Simulator Initialize or the Simulator/PROM switch on the Field Engineering Switch Panel. This allows 100 ns longer for the memory access.

9.1 The Writable Control Store Option allows two modes of operation with the MPC. In the first mode, the MPC can execute μ-code out of the WCS PROM from addresses above 511 (currently up to 1023). This is known as the Extended Mode. It is done by simply addressing these locations. See Table 9-1.

In the second mode, the MPC may execute μ-code out of the writable memory on the WCS as though it were coming from any 256 word module; that is, from addresses 0 thru 255, or 256 thru 511, or 512 thru 767, or 768 thru 1023. See Table 9-1. This is done by issuing an OCP Stop to the MPC, loading the WCS RAM from main memory, OTA'ing configuration information to the WCS and then issuing an OCP Simulator Initialize to the MPC. The MPC will then execute the instruction starting in location zero.

Table 9-1

Writable Control Store
Memory Mapping

MPC u-Code Address	WCS Mode						
	Extended Mode	Extended Mode	Extended Mode	Simulate Mode	Simulate Mode	Simulate Mode	Simulate Mode
0 - 255	On Board PROM	On Board PROM	On Board PROM	RAM WCS	On Board PROM	On Board PROM	On Board PROM
256 - 511	On Board PROM	On Board PROM	On Board PROM	On Board PROM	RAM WCS	On Board PROM	On Board PROM
512 - 776	Slow PROM WCS	--	Slow PROM WCS	--	--	RAM WCS	--
777 - 1023	--	Slow PROM WCS	Slow PROM WCS	--	--	--	RAM WCS

9.2 To operate with the Signetics ROM Simulator, the four cables must be plugged into a junction box, which is then cabled to the MPC. Pins 14G-14 and/or 8N-15 must be grounded on the MPC. 14G-14 disables PROM outputs on the MPC of locations 0 thru 255 and 8N-15 disables PROM outputs from locations 256 thru 511. In addition, the MPC must be placed in the Simulator Timing Mode by either placing the switch on the Field Engineering Switch Panel (if connected) in the Simulator position or issuing an OCP Simulator Initialize. If it is desired to run with internal PROM with the ROM Simulator or external PROM still connected, one must ground pin 2L-14 to disable the external PROM.

9.3 To operate the MP with external ROM or PROM, a junction box must be used similar to that used with the ROM simulator to distribute addresses to the ROM/PROM chips and wire OR the output lines. Also the on board PROMs must be disabled by grounding the XTROM- and XTRIM- signals. Up to 1024 words of external PROM may be accessed. The cables to the external PROM must not exceed ten inches. These cables plug into the DIP sites shown in Table 9-2.

9.4 If one desires to run with external PROM or the ROM Simulator with the on board PROMs plugged in, one must ground pins 29K-1 and 33K-1 to disable the on board PROMs.

Table 9-2

MPC/ROM Sim/WCS Interface

DIP Site-Pin	Signal	DIP Site-Pin	Signal
2L-1	MCY09+	29M-1	RCC17+
2	MCY10+	2	RCC18+
3	MCY11+	3	RCC19+
4	MCY12+	4	RCC20+
5	MCY13+	5	RCC21+
6	MCY14+	6	RCC22+
7	MCY15+	7	RCC23+
8	GDO4L+	8	RCC24+
9	MCY16+	9	RCC25+
10	MCY08+	10	RCC26+
11	MCY07+	11	RCC27+
12	XTROM-	12	RCC28+
13	XTR1M-	13	RCC29+
14	PULUP+G	14	RCC30+
15	GDO4L+	15	RCC31+
16	PULUP+G	16	RCC32+
10L-1	RCC01+	45M-1	RCC33+
2	RCC02+	2	RCC34+
3	RCC03+	3	RCC35+
4	RCC04+	4	RCC36+
5	RCC05+	5	RCC37+
6	RCC06+	6	RCC38+
7	RCC07+	7	RCC39+
8	RCC08+	8	RCC40+
9	RCC09+	9	RCC41+
10	RCC10+	10	RCC42+
11	RCC11+	11	RCC43+
12	RCC12+	12	RCC44+
13	RCC13+	13	RCC45+
14	RCC14+	14	RCC46+
15	RCC15+	15	RCC47+
16	RCC16+	16	RCC48+

```
                    (0001) *       FILE PCRD03        06-28-1974      REV 3
                    (0002) *
                    (0003) *
                    (0004) *
                    (0005) *       CONTROLLER FOR PRINTER, CARD READER, CARD PUNCH
                    (0006) *       (UNIT RECORD CONTROLLER)
                    (0007) *
                    (0008) *
                    (0009) *
                    (0010) *              **************
                    (0011) *              *            *
                    (0012) *              * P R 1 M E  *
                    (0013) *              *            *
                    (0014) *              *   M P C    *
                    (0015) *              *            *
                    (0016) *              * ASSEMBLER  *
                    (0017) *              *            *
                    (0018) *              **************
                    (0019) *
                    (0020) *
                    (0021) *
                    (0938) *
                    (0939)         IDNT    (MPC PROM SET DA ),(REV A ),(JUNE 28,1974)
                    (0940) *
                    (0941) *
                    (0942) *       INITIALIZE ROUTINE STARTS HERE
                    (0943) *
                    (0944) *
                    (0945) BEGIN   MPC     TRN RCM = $0F => DR2   SB
000: 0002 5AB9 000F
                    (0946)         MPC     CON ZERO => DR5   CDEOR
001: 0002 B9CC 0000
                    (0947)         MPC     TRN RCM = $FC => DR0   CIF
002: 0002 1AA1 00FC
                    (0948)         MPC     TRN RCM = 31 => (RF 31,DR7,OFC)   CDI1
003: 00FF FAD1 001F
                    (0949) BRFCL   MPC     DEC RF 31 => (RF 31,DR7)   SCC
004: 00FB E7BC 0000
```



```
                    (0950)         MPC     CON ZERO => (RFDR7,DR3)   CDI2 ;
                    (0951)                 JUMP ON NOT CCAEZ TO BRFCL
005: 0203 7904 6404
                    (0952)         MPC     TRN RCM = $43 => RF 27   CDI3
006: 0009 1AD9 0043
                    (0953)         MPC     TRN RCM = $45 => RF 29   COF
007: 00E9 1A9C 0045
                    (0954)         MPC     TRN RCM = $46 => RF 31
008: 00F9 1AC1 0046
                    (0955)         MPC     TRN OFC => SRL   CB   GOTO BWINT
009: 5C04 DA98 8010
                    (0956) BCLB    MPC     TRN RCM = $07 => SRL   CB
00A: 0004 DA99 0007
                    (0957) *
                    (0958) *
                    (0959) *       WAIT HERE FOR AN OTA TESTING PRINTER FOR DEMANDING SERVICE
                    (0960) *
                    (0961) *
                    (0962) BOTAT   MPC     RF 21 BRN DB7 => RF 21   NOP   JUMP ON OTAF TO BWINT
00B: 38A9 1040 8C10
                    (0963)         MPC     RF 21 AND RCM = $E0 => RF 21
00C: 00A9 18C1 00E0
                    (0964)         MPC     RF 21 MINUS RCM = $C0 => NOP   SCC
00D: 00A6 0CBD 00C0
                    (0965)         MPC     NOP   NOP   JUMP ON NOT CCAEZ TO BOTAT
00E: 0000 0040 640B
                    (0966)         MPC     TRN RF 27 => ARR   NOP   GOTO PINT
00F: 00DC 3840 8059
                    (0967) *
                    (0968) *
                    (0969) *       INTERRUPTS AT END OF ROUTINES ENTER HERE
                    (0970) *
                    (0971) *
                    (0972) BWINT   MPC     TRN OFC => RF 0   CDI3   JUMP ON NOT OTAF TO *
010: 5C01 1AD8 0C10
                    (0973) *
                    (0974) *       START DECODING OTA HERE
                    (0975) *
```



```
                    (0975) BCTAR   MPC     RF 0 AND RCM = $0F => RF 0   SCC ACT 5
011: 0001 18BD 400F
                    (0977)         MPC     CON ZERO => MR   SB   JUMP ON NOT CCBT TO BOTSET
012: 0004 7988 2019
                    (0978)         MPC     TRN OFC => NOP   NOP   BRANCH TO BNCH1 4WAYS
013: 5C00 1AC1 9014
                    (0979)         ORG     ((*+3).AND.'77774)
                    (0980) *
                    (0981) *
                    (0982) *       4 WAY BRANCH THAT DECODES OTA'S 14 THROUGH 17 FOLLOWS
                    (0983) *
                    (0984) *
                    (0985) BNCH1   MPC     TRN DRL => RF 24   COF   GOTO BCHAN
014: 50C1 1A9C 803C
                    (0986)         MPC     TRN DRL => RF 26   COF   GOTO BPVEC
015: 5001 1A9C 803D
                    (0987)         MPC     TRN DRL => RF 28   COF   GOTO BCRVEC
016: 50E1 1A9C 803E
                    (0988)         MPC     TRN DRL => RF 30   COF
017: 50F1 1A9C 0000
                    (0989)         MPC     TRN DRR => RF 31   NOP   GOTO BCLB
018: 40F9 1ACD 800A
                    (0990) *
                    (0991) *
                    (0992) *       OTA'S 1 THROUGH 3 ARE FURTHER DECODED HERE
                    (0993) *
                    (0994) *
                    (0995) BOTSET  MPC     TRN DRL => RF 1   SCC   ACT 1
019: 5009 1ABD 0000
                    (0996)         MPC     TRN RF 25 => ARR   COF   JUMP ON CCBT TO BINA
01A: 00CC 3B1C A024
                    (0997)         MPC     RF 24 AND RCM = $07 => ARL
01B: 00C4 B8C1 0007
                    (0998)         MPC     RF 24 AND RCM = $08 => (RF 24,MR)
01C: 00C5 78C1 0008
                    (0999)         MPC     RF 24 OR RCM = $04 => (RF 24,MR)
01D: 00C5 7A91 0004
                    (1000)         MPC     TRN RCM = 80 => RF 10   CDEOR
```



```
01E: 0051 1ACD 0050
                    (1001) *
                    (1002) *
                    (1003) *       DO 4 WAY BRANCH TO DECODE WHETHER OTA 01,02 OR 03
                    (1004) *
                    (1005) *
                    (1006)         MPC     TRN OFC => NOP   NOP   BRANCH TO BNCH2 4WAYS
01F: 5C00 1AC1 9020
                    (1007)         ORG     ((*+3).AND.'77774)
                    (1008) BNCH2   MPC     NOP   NOP   GOTO BCLB
020: 0000 0040 800A
                    (1009)         MPC     CON ZERO => RF 5   CDI3   GOTO PRINT
021: 0029 19D6 803F
                    (1010)         MPC     CON ZERO => (RF 22,DR4)   NOP   GOTO CRD
022: 0083 99C0 805B
                    (1011)         MPC     CON ZERO => RF 17   NOP   GOTO CPBEG
023: 0089 19C0 807E
                    (1012) *
                    (1013) *
                    (1014) *       OTA TO DO INA STARTS HERE
                    (1015) *
                    (1016) *       SET DIF3 FOR VECTOR ADDRESS OR ID INA
                    (1017) *
                    (1018) *
                    (1019) BINA    MPC     TRN DRR => RF 2   SCC   ACT 8
024: 4011 1ABD 7000
                    (1020)         MPC     TRN DB8 => RF 23   NOP   JUMP ON NOT CCBT TO *+2
025: 3C89 1AC0 2027
                    (1021)         MPC     NOP   SDF3
026: 0000 0008 0000
                    (1022)         MPC     TRN DRR => NOP   SCC   ACT 7
027: 4000 1ABD 8000
                    (1023)         MPC     TRN DB7 => RF 17   NOP   JUMP ON CCBT TO BCHINA
028: 3889 1AC0 A03A
                    (1024) *
                    (1025) *
                    (1026) *       STATUS OR VECTOR ADDRESS INA.  DETERMINE WHICH DEVICE
                    (1027) *       TEST FOR OTA 01
```

```
                    (1026) *
                    (1027) *
                    (1028) *
                    (1029) *
                    (1030)          MPC     RF 0 MINUS RCM = 01 => NOP   SCC
0291: 0000 0C80 0001
                    (1031)          MPC     TRN RF 26 => DRL   NOP   JUMP ON CCAEZ TO BPINA
02A1: 0004 9640 E451
                    (1032) *
                    (1033) *
                    (1034) *        TEST FOR OTA 02
                    (1035) *
                    (1036) *
                    (1037)          MPC     RF 0 MINUS RCM = 02 => NOP   SCC
02B1: 0000 0CB0 0002
                    (1038)          MPC     TRN RF 28 => DRL   NOP   JUMP ON CCAEZ TO BCRINA
02C1: 00E4 9640 E435
                    (1039) *
                    (1040) *
                    (1041) *        OTA 03 HAS BEEN DECODED
                    (1042) *
                    (1043) *
                    (1044)          MPC     TRN RF 30 => DRL   NOP   JUMP ON DIF3 TO BCPVEC
02D1: 00F4 9640 F830
                    (1045)          MPC     RF 23 AND RCM = $0F => DRR
02E1: 008C 1861 000F
                    (1046)          MPC     CON ZERO => DRL   SIRDY   GOTO BCLB
02F1: 0004 9994 800A
                    (1047) BCPVEC   MPC     TRN RF 31 => DRR   SIRDY   GOTO BCLB
0301: 00FC 1814 800A
                    (1048) *
                    (1049) *
                    (1050) *        PRINTER INA STATUS AND VECTOR ADDRESS
                    (1051) *
                    (1052) *
                    (1053) BPINA    MPC     TRN RF 27 => DRR   NOP   JUMP ON DIF3 TO BIDFIN
0311: 00DC 1840 F834
                    (1054)          MPC     RF 17 7 RCM = $C0  + L => DRR
0321: 008C 18C1 00C0
                    (1055) BCMEND   MPC     CON ZERO => DRL   SIRDY   GOTO BCLB
```



```
0331: 0004 9994 800A
                    (1056) BIDFIN   MPC     NOP   SIRDY   GOTO BCLB
0341: 0000 0014 800A
                    (1057) *
                    (1058) *
                    (1059) *        CARD READER INA STATUS AND VECTOR ADDRESS
                    (1060) *
                    (1061) *
                    (1062) BCRINA   MPC     RF 17 AND RCM = $07 => DR7
0351: 008A F8C1 0007
                    (1063)          MPC     RF 22 OR DR7 => DR7   NOP   JUMP ON DIF3 TO BCRIVA
0361: A0B2 FA40 F839
                    (1064)          MPC     RF 23 AND RCM = $80 => RF 23
0371: 0059 18C1 0080
                    (1065)          MPC     RF 23 OR DR7 => DRR   NOP   GOTO BCREND
0381: A08C 1A40 8035
                    (1066) BCRIVA   MPC     TRN RF 29 => DRR   SIRDY   GOTO BCLB
0391: 00EC 1814 800A
                    (1067) *
                    (1068) *
                    (1069) *        INA CHANNEL NUMBER JUMPS HERE
                    (1070) *
                    (1071) *
                    (1072) BCHINA   MPC     TRN RF 24 => DRL
03A1: 00C4 9840 0000
                    (1073)          MPC     TRN RF 25 => DRR   SIRDY   GOTO BCLB
03B1: 00CC 1814 800A
                    (1074) *
                    (1075) *
                    (1076) *        OTA CHANNEL NUMBER AND VECTOR ADDRESSES CONTINUE HERE
                    (1077) *
                    (1078) *
                    (1079) BCHAN    MPC     TRN DRR => RF 25   NOP   GOTO BCLB
03C1: 40C9 1AC0 800A
                    (1080) BPVEC    MPC     TRN DRR => RF 27   NOP   GOTO BCLB
03D1: 40D7 1AC0 800A
                    (1081) BCRVEC   MPC     TRN DRR => RF 29   NOP   GOTO BCLB
03E1: 40E9 1AC0 800A
```



```
                    (1082) *
                    (1083) *
                    (1084) *        PRINTER ROUTINE STARTS HERE
                    (1085) *
                    (1086) *
                    (1087) PRINT    MPC     TRN RCM = $0D => RF 6
03F1: 0031 1AC1 000D
                    (1088)          MPC     TRN RF 1 => NOP   SCC   ACT 2
0401: 0008 1830 1000
                    (1089)          MPC     TRN DRR => RF 4   NOP   JUMP ON CCBT TO PDMXR
0411: 4021 1AC0 A045
                    (1090) *
                    (1091) *        NOT A PRINT ORDER
                    (1092) *
                    (1093)          MPC     RF 4 E RCM = $7F  + L => (RF 4,DR1)   SCC   ACT 2
0421: 0023 3F3D 107F
                    (1094)          MPC     NOP   NOP   JUMP ON CCBT TO PTDEM
0431: 0000 0040 A048
                    (1095)          MPC     INC RF 5 => RF 5   NOP   GOTO PTDEM
0441: 0029 0640 8048
                    (1096) *
                    (1097) *        SET DMX REQUEST TO GET DATA WORD
                    (1098) *
                    (1099) PDMXR    MPC     TRN RCM = $92 => DR7   SDRQ
0451: 0002 FAC9 0092
                    (1100)          MPC     TRN DRL => RF 4   CDI3   JUMP ON DRQOP TO *
0461: 5021 1AD8 D446
                    (1101) PSTRQ    MPC     RF 4 E RCM = $7F  + L => (RF 4,DR1)
0471: 0023 3F41 007F
                    (1102) *
                    (1103) *        TEST FOR PRINTER DEMAND LINE TRUE
                    (1104) *
                    (1105) PTDEM    MPC     RF 0 BRN DB7 => NOP   SCC   ACT 2
0481: 3800 1030 1000
                    (1106)          MPC     RF 5 RFN DR2 => DR2   NOP   JUMP ON NOT CCBT TO *-1
0491: 242A 5FC0 2048
                    (1107) *
                    (1108) *        DEMAND LINE IS TRUE, SET CHARACTER STROBE TRUE
```



```
                    (1109) *
                    (1110)          MPC     RF 5 NOR RCM = $02 => DR2
04A1: 002A 5DC1 0002
                    (1111)          MPC     RF 0 BRN DB7 => NOP   SCC   ACT 2
04B1: 3800 1030 1000
                    (1112)          MPC     NOP   NOP   JUMP ON CCBT TO *-1
04C1: 0000 0040 A046
                    (1113) *
                    (1114) *        DEMAND LINE IS FALSE, SET CHARACTER STROBE FALSE
                    (1115) *
                    (1116)          MPC     RF 5 E RCM = $01  + L => DR2
04D1: 002A 5F41 0001
                    (1117) *
                    (1118) *        CHECK WHETHER CODE JUST SENT TO PRINTER IS PF,FF,CR OR PAPER INSTRUCTIO
                    (1119) *
                    (1120)          MPC     RF 4 MINUS RCM = $F5 => NOP   SCC
04E1: 0020 0CB0 00F5
                    (1121)          MPC     TRN RF 27 => ARR   NOP   JUMP ON CCAEZ TO PINT
04F1: 00DC 3640 E459
                    (1122)          MPC     RF 4 MINUS RCM = $F3 => RF 4   SCC
0501: 0021 0CB0 00F3
                    (1123)          MPC     INC RF 4 => NOP   SCC   JUMP ON CCALZ TO PINT
0511: 0020 0830 E459
                    (1124)          MPC     DEC RF 5 => NOP   SCC   JUMP ON CCAEZ TO PINT
0521: 0028 07B0 E459
                    (1125)          MPC     CON MINUS1 => DR2   NOP   JUMP ON CCAEZ TO PINT
0531: 0002 4640 E459
                    (1126) *
                    (1127) *        MORE DATA TO BE SENT TO PRINTER. TEST IF BOTH CHARACTERS IN
                    (1128) *              WORD HAVE BEEN SENT
                    (1129) *
                    (1130)          MPC     TRN RF 25 => ARR   NOP   JUMP ON DIF3 TO PTSTE
0541: 00CC 3840 F856
                    (1131) PRDB     MPC     TRN DRR => RF 4   SDF3   GOTO PSTRQ
0551: 4021 1AE8 8047
                    (1132) *
                    (1133) *        TEST END OF RANGE. FORCE CR CODE IF EOR
                    (1134) *
```

```
                    (1135) PTSTE  MPC     TRN RF 6 => DRR   NOP   JUMP ON EOR TO PRDB
056: 0034 1840 E855
                    (1136) *
                    (1137) *      NOT EOR. DELAY BEFORE MAKING NEXT DMX REQUEST
                    (1138) *
                    (1139)        MPC     NOP   IDR7   JUMP ON NOT DR7CO TO *
057: 0000 0010 4057
                    (1140)        MPC     NOP   NOP   GOTO PDMXR
058: 0000 0040 8045
                    (1141) *
                    (1142) *
                    (1143) *      A PRINTER INTERRUPT JUMPS HERE
                    (1144) *
                    (1145) *
                    (1146) PINT   MPC     TRN RCM = $03 => SRL   SIRQ
059: 0004 DAC5 0003
                    (1147)        MPC     TRN RF 26 => ARL   CB   GOTO BWINT
05A: 0004 8818 8010
                    (1148) *
                    (1149) *      THIS IS END OF PRINTER ROUTINE
                    (1150) *
                    (1151) *
                    (1152) *      CARD READER ROUTINE STARTS HERE
                    (1153) *
                    (1154) *
                    (1155) CRD    MPC     TRN RCM = $80 => (RF 17,DR3)   CDI3
05B: 0088 7AD9 0080
                    (1156)        MPC     RF 24 OR RCM = $10 => MR
05C: 00C4 7A41 0010
                    (1157) *
                    (1158) *      TEST READER READY
                    (1159) *
                    (1160) CRIND  MPC     TRN DB8 => NOP   SCC   ACT 1
05D: 3C00 1ABD 0000
                    (1161)        MPC     CON ZERO => RF 13   NOP   JUMP ON NOT CCBT TO CRINT
05E: 0069 19C0 206C
                    (1162) *
                    (1163) *      YES, READER IS READY, WAIT FOR INDEX AND STORE DATA ROWS 12 TO 0
```



```
                    (1164) *
                    (1165)        MPC     TRN DB8 => NOP   SCC   ACT 2
05F: 3C00 1ABD 1000
                    (1166)        MPC     TRN DR3 => RF 7   NOP   JUMP ON NOT CCBT TO CRIND
0601: 2839 1AC0 205D
                    (1167) *
                    (1168) *
                    (1169) *      STORE CARD ROWS 1 TO 9
                    (1170) *      SET OVER-RUN IF DMX REQUEST STILL TRUE
                    (1171) *
                    (1172) *
                    (1173)        MPC     TRN DR4 => (RF 8,DRR)   NOP   JUMP ON NOT DRQDP TO CRDAT
061: 2C45 1AC0 5463
                    (1174)        MPC     RF 22 OR RCM = $10 => RF 22
062: 0081 1A41 0010
                    (1175) CRDAT  MPC     RF 7 AND RCM = $0F => (RF 7,DRL)
063: 003D 98C1 000F
                    (1176)        MPC     TRN RF 1 => NOP   SCC   ACT 3
064: 0008 183D 2000
                    (1177) *
                    (1178) *      TEST FOR BINARY MODE. ASCII MODE JUMPS TO CARD PUNCH TRANSLATION
                    (1179) *      TABLE
                    (1180) *
                    (1181)        MPC     CON ZERO => (RF 14,DR3)   NOP   JUMP ON NOT CCBT TO CPSBN
065: 0073 79C0 2085
                    (1182) *
                    (1183) *      WAIT FOR END OF INDEX MARK
                    (1184) *
                    (1185)        MPC     TRN DB8 => NOP   SCC   ACT 2
066: 3C00 1ABD 1000
                    (1186)        MPC     NOP   NOP   JUMP ON CCBT TO *-1
067: 0000 0040 A066
                    (1187) *
                    (1188) *      TEST END OF RANGE
                    (1189) *
                    (1190) CREORT MPC     NOP   CDI3   JUMP ON EOR TO CRDCC
068: 0000 0058 E86A
                    (1191)        MPC     NOP   SDRG   GOTO CRDCC
```



```
069: 0000 0040 806A
                    (1192) *
                    (1193) *      TEST FOR 80 CARD COLUMNS RECEIVED
                    (1194) *
                    (1195) CRDCC  MPC     DEC RF 10 => RF 10   SCC
06A: 0051 078C 0000
                    (1196)        MPC     NOP   NOP   JUMP ON NOT CCAEZ TO CRIND
06B: 0000 0040 645D
                    (1197) *
                    (1198) *
                    (1199) *      80 CARD COLUMNS HAVE BEEN RECEIVED. WAIT FOR DMX TO FINISH
                    (1200) *
                    (1201) *
                    (1202) CRINT  MPC     CON ZERO => DR3   NOP   JUMP ON DRQDP TO *
06C: 0002 79C0 D46C
                    (1203)        MPC     TRN RCM = $05 => SRL
06D: 0004 DAC1 0005
                    (1204)        MPC     TRN RF 28 => ARL   SIRQ
06E: 00E4 B844 0000
                    (1205)        MPC     TRN RF 29 => ARR   CB   GOTO BWINT
06F: 00EC 3818 8010
                    (1206) *      ENTER HERE FROM PUNCH TRANSLATION TABLE. RF 13 CONTAINS CARD ROWS
                    (1207) *      12 TO 1 AND RF 15 CONTAINS CARD ROWS 2 TO 9
                    (1208) *
                    (1209) *
                    (1210) CRTRN  MPC     RF 13 AND RCM = $0F => RF 13
070: 0069 18C1 000F
                    (1211)        MPC     TRN RF 7 => DRL
071: 003C 9840 0000
                    (1212)        MPC     RF 13 MINUS DRL => NOP   SCC
072: 5068 0C8C 0000
                    (1213) *
                    (1214) *      COMPARE 12 TO 1 FROM CARD PUNCH TABLE WITH 12 TO 1 READ FROM CARD
                    (1215) *
                    (1216)        MPC     RF 15 MINUS DRR => NOP   SCC   JUMP ON NOT CCAEZ TO CRTRA
073: 4078 0C8C 647B
                    (1217) *
                    (1218) *      THEY WERE EQUAL. NOW COMPARE 2 TO 9 FROM CARD PUNCH TABLE WITH 2 TO 9
```



```
                    (1219) *      READ FROM THE CARD
                    (1220) *
                    (1221)        MPC     TRN DR7 => RF 14   NOP   JUMP ON NOT CCAEZ TO CRTRA
074: A071 1AC0 6478
                    (1222) *
                    (1223) *      THEY WERE EQUAL ALSO. RF 14 (BITS 3-8) HAS A VALUE IN
                    (1224) *      PROPORTION TO THE ASCII CHAR. READ FROM THE CARD. CONVERT
                    (1225) *      THIS PROPORTION TO THE ACTUAL ASCII CHAR.
                    (1226) *
                    (1227)        MPC     RF 14 AND RCM = $BF => RF 14
075: 0071 18C1 00BF
                    (1228)        MPC     RF 14 XOR RCM = $20 => (RF 14,DRR)   SCC   ACT 3
076: 0075 183D 2020
                    (1229)        MPC     TRN RF 9 => DRL   NOP   JUMP ON CCBT TO *+2
077: 004C 9840 A079
                    (1230)        MPC     RF 14 OR RCM = $40 => (RF 14,DRR)
078: 0075 1A41 0040
                    (1231) *
                    (1232) *      RF14 (AND DRR) NOW HOLD THE ASCII EQUIVALENT OF
                    (1233) *      THE 12 BIT HOLLERITH READ FROM THE CARD.TEST BOTH
                    (1234) *      CHARS PER COMPUTER WORD HAVE BEEN READ
                    (1235) *
                    (1236) CRTCOM MPC     TRN RF 14 => DR7   NOP   JUMP ON DIF3 TO CREORT
079: 0072 F840 F868
                    (1237)        MPC     TRN DR7 => RF 9   SDF3   GOTO CRDCC
07A: A049 1AE8 806A
                    (1238) *
                    (1239) *      ENTER HERE IN ASCII MODE WHEN ANOTHER SAMPLING OF PUNCH
                    (1240) *      TRANSLATION TABLE IS REQUIRED
                    (1241) *
                    (1242) CRTRA  MPC     CON ZERO => RF 13   IDR7   JUMP ON NOT DR7CO TO CPTRN
07B: 0069 1990 4086
                    (1243) *
                    (1244) *
                    (1245) *      CHAR. READ IN CAN'T BE TRANSLATED INTO ASCII
                    (1246) *      SET STATUS BIT 11
                    (1247) *
                    (1248) *
```

```
                    (1249)          MPC     RF 22 OR RCM = $20 => RF 22
07C: 0081 1A41 0020
                    (1250)          MPC     NOP   NOP   GOTO CNTCOM
07D: 0000 0040 8079
                    (1251) *
                    (1252) *
                    (1253) *
                    (1254) *        CARD PUNCH ROUTINE STARTS HERE
                    (1255) *
                    (1256) *
                    (1257) CPBEG    MPC     TRN D88 => NOP   SCC   ACT 5
07E: 3C00 1A8D 4000
                    (1258)          MPC     NOP   NOP   JUMP ON NOT CCBT TO CPFIN
07F: 0000 0040 2081
                    (1259) *
                    (1260) *        SET PUNCH COMMAND AND GET DATA
                    (1261) *
                    (1262) CPSDMX   MPC     TRN RCM = $80 => (RF 13,DR5)   SDRW
080: 006B BAC9 0080
                    (1263)          MPC     TRN DRL => RF 14   CDI3   JUMP ON DRUDP TO *
081: 5071 1A08 0481
                    (1264)          MPC     TRN RF 1 => NOP   SCC   ACT 3
082: 0008 183D 2000
                    (1265) *
                    (1266) *        TEST FOR BINARY MODE
                    (1267) *
                    (1268)          MPC     NOP   NOP   JUMP ON CCBT TO CPBIN
083: 0000 0040 A0AC
                    (1269) *
                    (1270) *        NORMALIZE 6 BIT ASCII CHARACTER AND SETUP BRANCH FOR CODE CONVERSION
                    (1271) *
                    (1272) CPASC    MPC     RF 14 XOR RCM = $20 => RF 14
084: 0071 1841 0020
                    (1273) CPSBN    MPC     RF 14 OR RCM = $C0 => DR7
085: 0072 FA41 00C0
                    (1274) CPTRN    MPC     TRN DR7 => NOP   PSH   BRANCH TO BEGIN 256WAYS
086: A000 1A89 F000
                    (1275) *
```



```
                    (1276) *        CONVERTED DATA IS IN RF 14. NEEDS TO BE FURTHER CONVERTED TO
                    (1277) *              TWELVE ROWS OF THE CARD
                    (1278) *
                    (1279)          MPC     RF 14 AND RCM = $0E => DRL
087: 0074 98C1 000E
                    (1280)          MPC     RF 13 OR DRL => RF 13
088: 5069 1A40 0000
                    (1281)          MPC     RF 14 LS DR1 + 0 => RF 14   SCC
089: 2071 018C 0000
                    (1282)          MPC     RF 14 0 DR1 + C => RF 14
08A: 2071 1040 0000
                    (1283) *
                    (1284) *        ROWS 12,11,0 HAVE BEEN TRANSFERRED TO RF13 TEMPORARILY AND RF14
                    (1285) *              HAS BEEN ROTATED LEFT ONE PLACE
                    (1286) *
                    (1287)          MPC     RF 14 AND RCM = $03 => DRL
08B: 0074 98C1 0003
                    (1288)          MPC     RF 14 AND RCM = $E0 => RF 14
08C: 0071 18C1 00E0
                    (1289) *
                    (1290) *        B1,B2 AND B3 REPRESENT CODED DATA FOR CARD ROWS 1 TO 7. THEY
                    (1291) *              ARE NOW DECODED
                    (1292) *
                    (1293) CPTR7    MPC     RF 14 MINUS RCM = $E0 => NOP   SCC
08D: 0070 0C8D 00E0
                    (1294)          MPC     CON ZERO => RF 15   NOP   JUMP ON NOT CCAEZ TO CPTR6
08E: 0079 19C0 6490
                    (1295) *
                    (1296) *        SET BIT FOR HOLE IN ROW 7
                    (1297) *
                    (1298)          MPC     RF 15 OR RCM = $04 => RF 15
08F: 0079 1A41 0004
                    (1299) CPTR6    MPC     RF 14 MINUS RCM = $C0 => NOP   SCC
090: 0070 0C8D 00C0
                    (1300)          MPC     NOP   NOP   JUMP ON NOT CCAEZ TO CPTR5
091: 0000 0040 6493
                    (1301) *
                    (1302) *        SET BIT FOR HOLE IN ROW 6
```



```
                    (1303) *
                    (1304)          MPC     RF 15 OR RCM = $08 => RF 15
092: 0079 1A41 0008
                    (1305) CPTR5    MPC     RF 14 MINUS RCM = $A0 => NOP   SCC
093: 0070 0C8D 00A0
                    (1306)          MPC     NOP   NOP   JUMP ON NOT CCAEZ TO CPTR4
094: 0000 0040 6496
                    (1307) *
                    (1308) *        SET BIT FOR HOLE IN ROW 5
                    (1309) *
                    (1310)          MPC     RF 15 OR RCM = $10 => RF 15
095: 0079 1A41 0010
                    (1311) CPTR4    MPC     RF 14 MINUS RCM = $80 => NOP   SCC
096: 0070 0C8D 0080
                    (1312)          MPC     NOP   NOP   JUMP ON NOT CCAEZ TO CPTR3
097: 0000 0040 6499
                    (1313) *
                    (1314) *        SET BIT FOR HOLE IN ROW 4
                    (1315) *
                    (1316)          MPC     RF 15 OR RCM = $20 => RF 15
098: 0079 1A41 0020
                    (1317) CPTR3    MPC     RF 14 MINUS RCM = $60 => NOP   SCC
099: 0070 0C8D 0060
                    (1318)          MPC     NOP   NOP   JUMP ON NOT CCAEZ TO CPTR2
09A: 0000 0040 649C
                    (1319) *
                    (1320) *        SET BIT FOR HOLE IN ROW 3
                    (1321) *
                    (1322)          MPC     RF 15 OR RCM = $40 => RF 15
09B: 0079 1A41 0040
                    (1323) CPTR2    MPC     RF 14 MINUS RCM = $40 => NOP   SCC
09C: 0070 0C8D 0040
                    (1324)          MPC     TRN RF 13 => DR5   NOP   JUMP ON NOT CCAEZ TO *+2
09D: 006A B840 649F
                    (1325) *
                    (1326) *        SET BIT FOR HOLE IN ROW 2
                    (1327) *
                    (1328)          MPC     RF 15 OR RCM = $80 => RF 15
```



```
09E: 0079 1A41 0080
                    (1329)          MPC     RF 15 OR DRL => (RF 15,DR6)
09F: 5078 DA40 0000
                    (1330) CPTR1    MPC     RF 14 MINUS RCM = $20 => NOP   SCC
0A0: 0070 0C8D 0020
                    (1331)          MPC     TRN RF 17 => NOP   SCC   JUMP ON NOT CCAEZ TO *+2
0A1: 0088 183C 64A3
                    (1332) *
                    (1333) *        SET BIT FOR HOLE IN ROW 1
                    (1334) *
                    (1335)          MPC     RF 13 OR RCM = $01 => (RF 13,DR5)
0A2: 006B BA41 0001
                    (1336) *        TEST FOR CARD READER AND IF SO EXIT
                    (1337) *
                    (1338)          MPC     NOP   NOP   JUMP ON NOT CCAEZ TO CRTRN
0A3: 0000 0040 6470
                    (1339) *
                    (1340) *        TEST FOR DATA REQUEST FROM PUNCH TRUE
                    (1341) *
                    (1342) CPTDR    MPC     TRN D88 => NOP   SCC   ACT 4
0A4: 3C00 1A8D 3000
                    (1343)          MPC     NOP   NOP   JUMP ON NOT CCBT TO *-1
0A5: 0000 0040 20A4
                    (1344) *
                    (1345) *        DATA REQUEST TRUE. SET DATA ACKNOWLEDGE
                    (1346) *
                    (1347)          MPC     RF 13 OR RCM = $40 => (RF 13,DR5)
0A6: 006B BA41 0040
                    (1348) *
                    (1349) *        TEST FOR DATA REQUEST FROM PUNCH FALSE
                    (1350) *
                    (1351)          MPC     TRN D88 => NOP   SCC   ACT 4
0A7: 3C00 1A8D 3000
                    (1352)          MPC     NOP   NOP   JUMP ON CCBT TO *-1
0A8: 0000 0040 A0A7
                    (1353) *
                    (1354) *        RESET DATA ACKNOWLEDGE. TEST FOR COMPLETE WORD PUNCHED
                    (1355) *
```

```
                    (1356)          MPC     RF 13 AND RCM = $8F => (RF 13,DR5)
0A9: 00B8 B8C1 008F
                    (1357)          MPC     DEC RF 10 => RF 10   NOP   JUMP ON DIF3 TO CPCWP
0AA: 0051 07C0 F8AF
                    (1358)          MPC     TRN DRR => RF 14.  SDF3   GOTO CPASC
0AB: 4071 1AE8 8084
                    (1359) *
                    (1360) *        PUNCH BINARY MODE ENTERS HERE. MOVE DATA TO OUTPUT REGISTERS
                    (1361) *
                    (1362) CPBIN    MPC     RF 14 AND RCM = $0F => DRL   SDF3
0AC: 0074 98E9 000F
                    (1363)          MPC     RF 13 OR DRL => (RF 13,DR5)
0AD: 5068 BA40 0000
                    (1364)          MPC     TRN DRR => DR6   NOP   GOTO CPTDR
0AE: 4002 DAC0 80A4
                    (1365) *
                    (1366) *        END CONDITIONS TESTED HERE
                    (1367) *
                    (1368) CPCWP    MPC     TRN RF 10 => NOP   SCC   JUMP ON EQR TO CPFIN
0AF: 0050 1B3C E881
                    (1369)          MPC     NOP   NOP   JUMP ON NOT CCAEZ TO CPBEG
0B0: 0000 0040 647E
                    (1370) *
                    (1371) *        STOP PUNCH AND INTERRUPT
                    (1372) *
                    (1373) CPFIN    MPC     CON ZERO => DR5
0B1: 0002 B9C0 0000
                    (1374)          MPC     TRN RCM = $06 => SRL
0B2: 0004 DAC1 0006
                    (1375)          MPC     TRN RF 30 => ARL   SIRQ
0B3: 00F4 8844 0000
                    (1376)          MPC     TRN RF 31 => ARR   CB   GOTO BWINT
0B4: 00FC 3818 8010
                    (1377) *
                    (1378) *
                    (1379) *        END OF PUNCH FIRMWARE. LOOKUP TABLE,  TO CONVERT ASCII DATA FROM
                    (1380) *                MAIN MEMORY TO A PARTIAL CONVERSION TO HOLLERITH,  FOLLOWS
                    (1381) *
```



```
                    (1382) *
                    (1383) *
                    (1384)          ORG     ((*+63).AND.'77700)
                    (1385)          MPC     TRN RCM => RF 14   POP   EMIT $00
0C0: 0071 1A85 0000
                    (1386)          MPC     TRN RCM => RF 14   POP   EMIT $25
0C1: 0071 1A85 0025
                    (1387)          MPC     TRN RCM => RF 14   POP   EMIT $71
0C2: 0071 1A85 0071
                    (1388)          MPC     TRN RCM => RF 14   POP   EMIT $31
0C3: 0071 1A85 0031
                    (1389)          MPC     TRN RCM => RF 14   POP   EMIT $35
0C4: 0071 1A85 0035
                    (1390)          MPC     TRN RCM => RF 14   POP   EMIT $43
0C5: 0071 1A85 0043
                    (1391)          MPC     TRN RCM => RF 14   POP   EMIT $08
0C6: 0071 1A85 0008
                    (1392)          MPC     TRN RCM => RF 14   POP   EMIT $51
0C7: 0071 1A85 0051
                    (1393)          MPC     TRN RCM => RF 14   POP   EMIT $59
0C8: 0071 1A85 0059
                    (1394)          MPC     TRN RCM => RF 14   POP   EMIT $55
0C9: 0071 1A85 0055
                    (1395)          MPC     TRN RCM => RF 14   POP   EMIT $45
0CA: 0071 1A85 0045
                    (1396)          MPC     TRN RCM => RF 14   POP   EMIT $69
0CB: 0071 1A85 0069
                    (1397)          MPC     TRN RCM => RF 14   POP   EMIT $33
0CC: 0071 1A85 0033
                    (1398)          MPC     TRN RCM => RF 14   POP   EMIT $04
0CD: 0071 1A85 0004
                    (1399)          MPC     TRN RCM => RF 14   POP   EMIT $39
0CE: 0071 1A85 0039
                    (1400)          MPC     TRN RCM => RF 14   POP   EMIT $12
0CF: 0071 1A85 0012
                    (1401)          MPC     TRN RCM => RF 14   POP   EMIT $02
0D0: 0071 1A85 0002
                    (1402)          MPC     TRN RCM => RF 14   POP   EMIT $10
```



```
0D1: 0071 1A85 0010
                    (1403)          MPC     TRN RCM => RF 14   POP   EMIT $20
0D2: 0071 1A85 0020
                    (1404)          MPC     TRN RCM => RF 14   POP   EMIT $30
0D3: 0071 1A85 0030
                    (1405)          MPC     TRN RCM => RF 14   POP   EMIT $40
0D4: 0071 1A85 0040
                    (1406)          MPC     TRN RCM => RF 14   POP   EMIT $50
0D5: 0071 1A85 0050
                    (1407)          MPC     TRN RCM => RF 14   POP   EMIT $60
0D6: 0071 1A85 0060
                    (1408)          MPC     TRN RCM => RF 14   POP   EMIT $70
0D7: 0071 1A85 0070
                    (1409)          MPC     TRN RCM => RF 14   POP   EMIT $01
0D8: 0071 1A85 0001
                    (1410)          MPC     TRN RCM => RF 14   POP   EMIT $80
0D9: 0071 1A85 0080
                    (1411)          MPC     TRN RCM => RF 14   POP   EMIT $21
0DA: 0071 1A85 0021
                    (1412)          MPC     TRN RCM => RF 14   POP   EMIT $65
0DB: 0071 1A85 0065
                    (1413)          MPC     TRN RCM => RF 14   POP   EMIT $49
0DC: 0071 1A85 0049
                    (1414)          MPC     TRN RCM => RF 14   POP   EMIT $61
0DD: 0071 1A85 0061
                    (1415)          MPC     TRN RCM => RF 14   POP   EMIT $63
0DE: 0071 1A85 0063
                    (1416)          MPC     TRN RCM => RF 14   POP   EMIT $73
0DF: 0071 1A85 0073
                    (1417)          MPC     TRN RCM => RF 14   POP   EMIT $41
0E0: 0071 1A85 0041
                    (1418)          MPC     TRN RCM => RF 14   POP   EMIT $18
0E1: 0071 1A85 0018
                    (1419)          MPC     TRN RCM => RF 14   POP   EMIT $28
0E2: 0071 1A85 0028
                    (1420)          MPC     TRN RCM => RF 14   POP   EMIT $38
0E3: 0071 1A85 0038
                    (1421)          MPC     TRN RCM => RF 14   POP   EMIT $48
```



```
0E4: 0071 1A85 0048
                    (1422)          MPC     TRN RCM => RF 14   POP   EMIT $58
0E5: 0071 1A85 0058
                    (1423)          MPC     TRN RCM => RF 14   POP   EMIT $68
0E6: 0071 1A85 0068
                    (1424)          MPC     TRN RCM => RF 14   POP   EMIT $78
0E7: 0071 1A85 0078
                    (1425)          MPC     TRN RCM => RF 14   POP   EMIT $09
0E8: 0071 1A85 0009
                    (1426)          MPC     TRN RCM => RF 14   POP   EMIT $88
0E9: 0071 1A85 0088
                    (1427)          MPC     TRN RCM => RF 14   POP   EMIT $14
0EA: 0071 1A85 0014
                    (1428)          MPC     TRN RCM => RF 14   POP   EMIT $24
0EB: 0071 1A85 0024
                    (1429)          MPC     TRN RCM => RF 14   POP   EMIT $34
0EC: 0071 1A85 0034
                    (1430)          MPC     TRN RCM => RF 14   POP   EMIT $44
0ED: 0071 1A85 0044
                    (1431)          MPC     TRN RCM => RF 14   POP   EMIT $54
0EE: 0071 1A85 0054
                    (1432)          MPC     TRN RCM => RF 14   POP   EMIT $64
0EF: 0071 1A85 0064
                    (1433)          MPC     TRN RCM => RF 14   POP   EMIT $74
0F0: 0071 1A85 0074
                    (1434)          MPC     TRN RCM => RF 14   POP   EMIT $05
0F1: 0071 1A85 0005
                    (1435)          MPC     TRN RCM => RF 14   POP   EMIT $84
0F2: 0071 1A85 0084
                    (1436)          MPC     TRN RCM => RF 14   POP   EMIT $22
0F3: 0071 1A85 0022
                    (1437)          MPC     TRN RCM => RF 14   POP   EMIT $32
0F4: 0071 1A85 0032
                    (1438)          MPC     TRN RCM => RF 14   POP   EMIT $42
0F5: 0071 1A85 0042
                    (1439)          MPC     TRN RCM => RF 14   POP   EMIT $52
0F6: 0071 1A85 0052
                    (1440)          MPC     TRN RCM => RF 14   POP   EMIT $62
```

```
0F7: 0071 1A85 0062
                    (1441)        MPC    TRN RCM => RF 14   POP   EMIT $72
0F8: 0071 1A85 0072
                    (1442)        MPC    TRN RCM => RF 14   POP   EMIT $03
0F9: 0071 1A85 0003
                    (1443)        MPC    TRN RCM => RF 14   POP   EMIT $82
0FA: 0071 1A85 0082
                    (1444)        MPC    TRN RCM => RF 14   POP   EMIT $23
0FB: 0071 1A85 0023
                    (1445)        MPC    TRN RCM => RF 14   POP   EMIT $79
0FC: 0071 1A85 0079
                    (1446)        MPC    TRN RCM => RF 14   POP   EMIT $75
0FD: 0071 1A85 0075
                    (1447)        MPC    TRN RCM => RF 14   POP   EMIT $29
0FE: 0071 1A85 0029
                    (1448)        MPC    TRN RCM => RF 14   POP   EMIT $53
0FF: 0071 1A85 0053
          000400
```



```
BCMAN    003C  A 0985  1079
BCMINA   003A  A 1023  1072
BCLB     000A  A 0956  0989  1008  1046  1047  1055  1056  1066  1073
                     1079  1080  1081
BCPVEC   0030  A 1044  1047
BCREND   0033  A 1055  1065
BCRINA   0035  A 1058  1062
BCRIVA   0039  A 1063  1066
BCRVEC   003E  A 0987  1081
BEGIN    0000  A 0945  1274
BIOFIN   0034  A 1053  1056
BINA     0024  A 0996  1019
BNCH1    0014  A 0978  0985
BNCH2    0020  A 1006  1008
BOTAR    0011  A 0976
BOTAT    0008  A 0962  0985
BOTSET   0019  A 0977  0995
BPINA    0031  A 1031  1053
BPVEC    0030  A 0986  1080
BRFCL    0004  A 0949  0951
BWINT    0010  A 0955  0962  0972  1147  1205  1376
CPASC    0084  A 1272  1358
CPBEG    007E  A 1011  1257  1369
CPBIN    00AC  A 1268  1362
CPCMP    00AF  A 1357  1360
CPFIN    0081  A 1256  1366  1373
CPSBN    0085  A 1181  1273
CPSDMX   0080  A 1262
CPTDR    00A4  A 1342  1364
CPTR1    00A0  A 1330
CPTR2    0090  A 1318  1323
CPTR3    0099  A 1312  1317
CPTR4    0096  A 1306  1311
CPTR5    0093  A 1300  1305
CPTR6    0090  A 1294  1299
CPTR7    0086  A 1293
CPTRN    0086  A 1242  1274
CRD      0058  A 1010  1155
```



```
CRDAT    0063  A 1173  1175
CRDCC    006A  A 1190  1191  1195  1237
CREORT   0068  A 1190  1236
CRIND    0050  A 1160  1166  1196
CRINT    006C  A 1161  1202
CRTCOM   0079  A 1230  1250
CRTRA    007B  A 1216  1221  1242
CRTRN    0070  A 1210  1338
FOMXR    0045  A 1089  1099  1140
PINT     0059  A 0966  1121  1123  1144  1125  1146
PRDB     0055  A 1131  1135
PRINT    003F  A 1009  1087
PSTRD    0047  A 1101  1131
PTDEM    0046  A 1094  1095  1105
PTSTE    0056  A 1130  1135
MPC  [MACRO]
ALU$ [MACRO]
TRN$ [MACRO]
NOT$ [MACRO]
INC$ [MACRO]
DEC$ [MACRO]
CON$ [MACRO]
DEST$  [MACRO]
DST1$  [MACRO]
EMAC$  [MACRO]
PEQU$  [MACRO]
JMPF$  [MACRO]
BRAN$  [MACRO]
GOTO$  [MACRO]
PLUS$  [MACRO]
GEND$  [MACRO]
SET1$  [MACRO]
SAY1$  [MACRO]
ORG  [MACRO]
IDNT [MACRO]
NOP  [MACRO]
CNTV$  [MACRO]
CLST$  [MACRO]
```

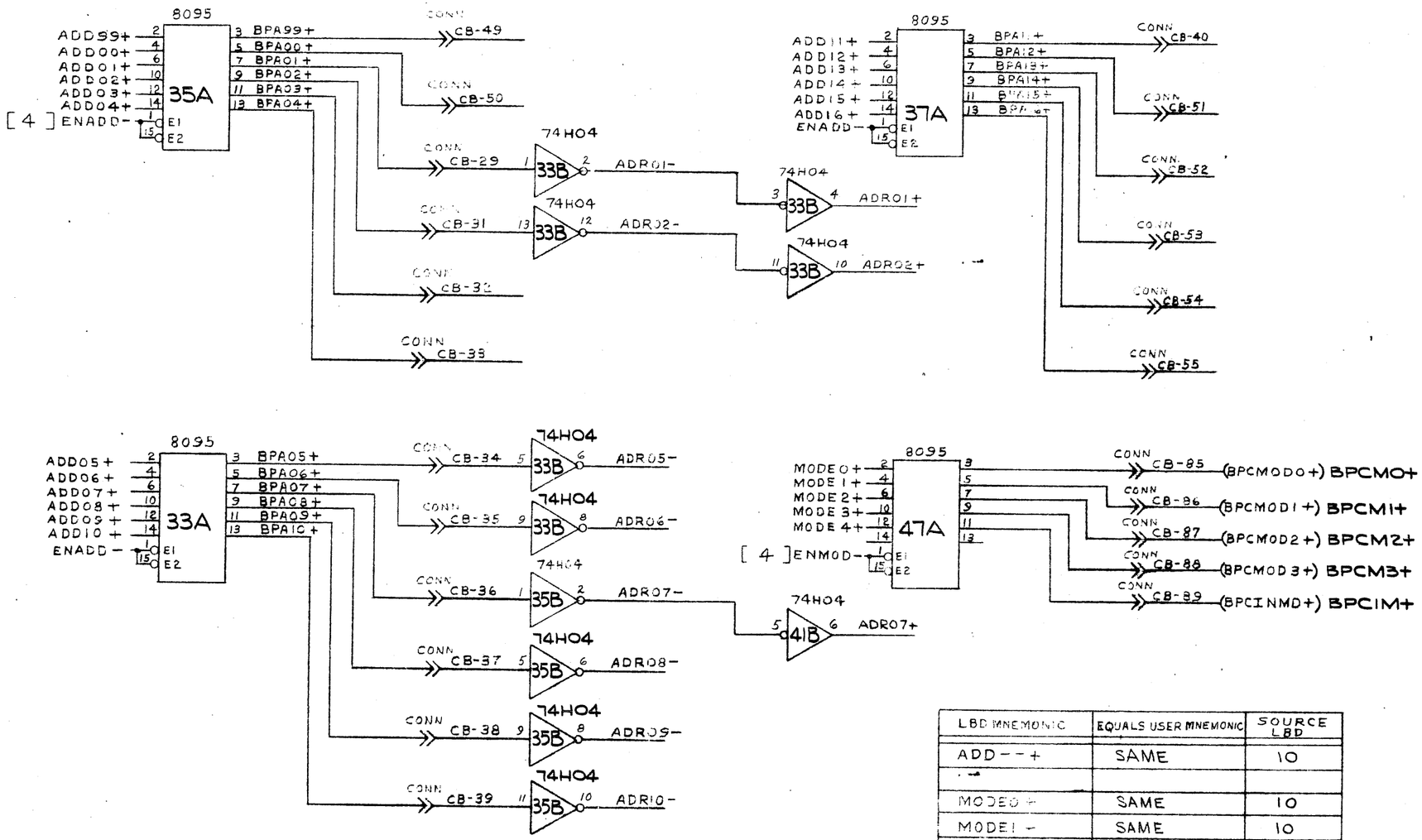

0000 ERRORS (PMA-1080.015)

LTR	DATE	REVISION	DR.	CK.
A	10/10/74	RELEASED	SW 12	JBL
B	12/11/74	REVISED PER ECN 1502	SW 16	[illegible]
B1	4/14/75	SHT 36E ADDED (CPI) OPTION	MK	MC
C	5/22.75	REVISED PER ECN 1530 (DISKETTE)	SW 12	W.F.

SHEET	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	27A	27E	27C	28	29	30	31	31A	31C	31D	32	32A	32B	32C	32D	33	33A	33B	33C	33D	34	35	35B	36	36A	36B	36C	36E	DATE
REVISION LEVEL	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	10/10/74
	B							B					B																	B																							B			B			12/11/74
	B1																																																										4/14/75
	C																													B																							B			B			5/22.75

MATERIAL	DWN H. Bogan 8/7/74	PRIME COMPUTER, INC. NATICK, MASS.	
UNLESS OTHERWISE SPECIFIED —REMOVE ALL BURRS AND SHARP EDGES: —DIMENSIONS ARE IN INCHES —TOLERANCES .XY ±.02 .XXX ±.005 ANGLES ±1/2°	CHK J. P. Gagnon 10/9/74	REVISION STATUS SHEET MPC-2 E.V.	
	ENG. D. Gardiner 10/10/74		
	APPRD [illegible] 10-9-74		
	USED ON	SCALE	SIZE C DWG. NO. LBD1829 REV. C
	NEXT ASSY 3155-002	SHEET 1 OF 36	

III-01

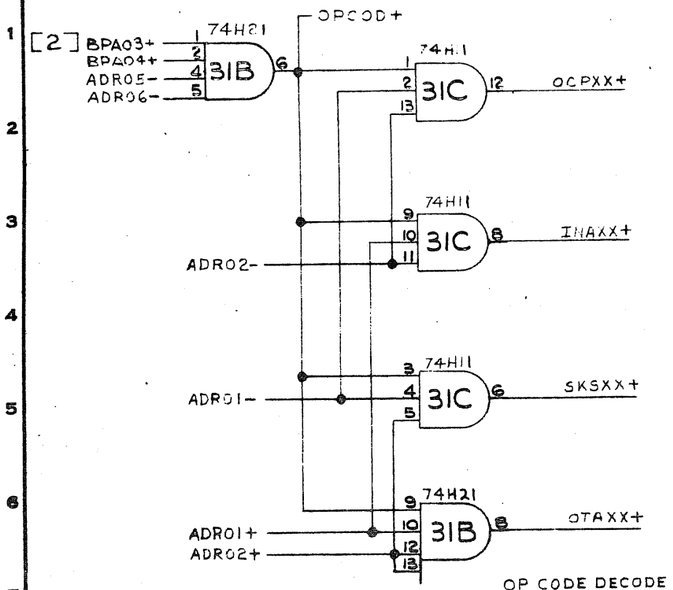


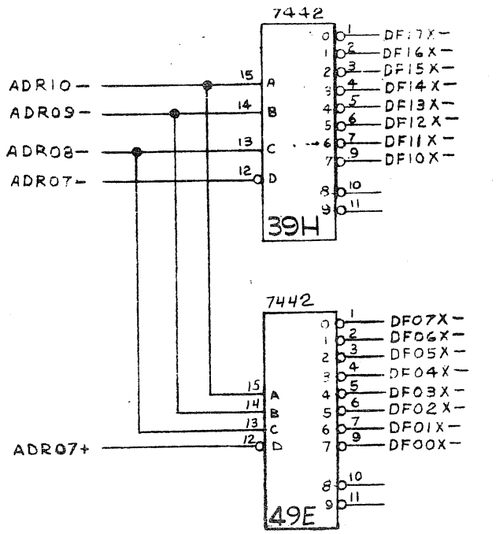
I/O BUS ADDRESS
DRIVERS & RECEIVERS

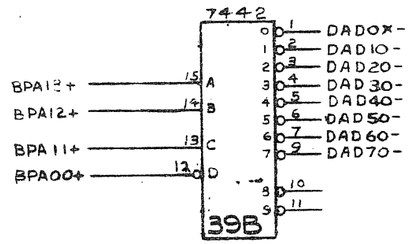
LBD MNEMONIC	EQUALS USER MNEMONIC	SOURCE LBD
ADD--+	SAME	10
. .		
MODE0+	SAME	10
MODE1+	SAME	10
MODE2+	SAME	10
MODE3+	SAME	10
MODE4+	SAME	10

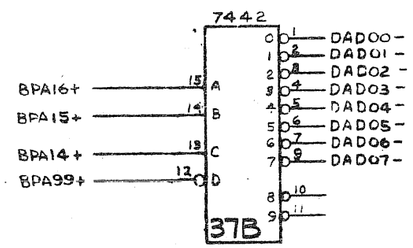
MATERIAL	DWN 3/13/73	PRIME COMPUTER, INC. NATICK, MASS.
UNLESS OTHERWISE SPECIFIED —REMOVE ALL BURRS AND SHARP EDGES; —DIMENSIONS ARE IN INCHES —TOLERANCES .XX ±.02 .XXX ±.005 ANGLES ±1/2°	CHK ENG. APPRD USED ON NEXT ASSY	I/O BUS INTERFACE LOGIC ADDRESS AND MODE LINES MPC2 (EV)
	SCALE SHEET 2 OF	SIZE C DWG. NO. LBD1829 REV. H

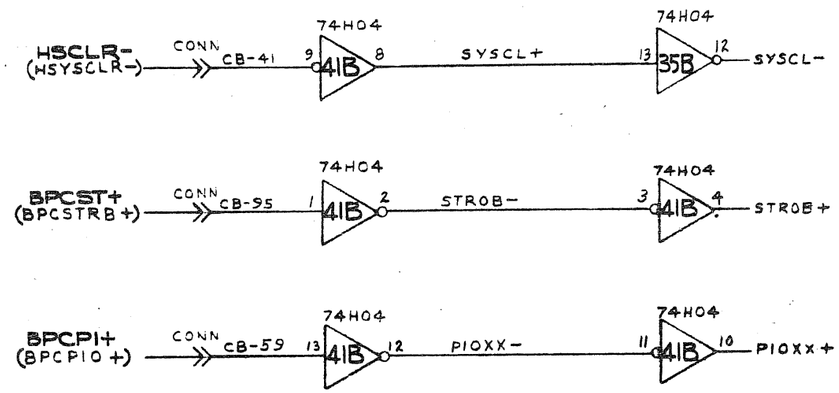
PDF-003





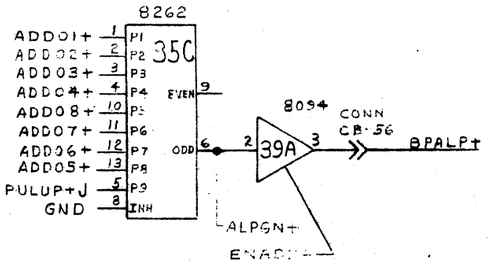


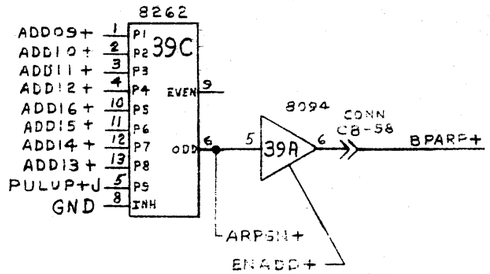


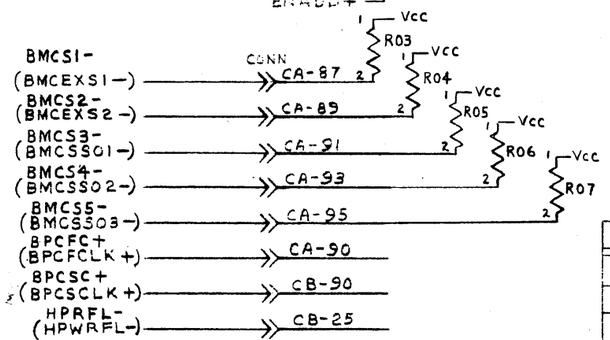


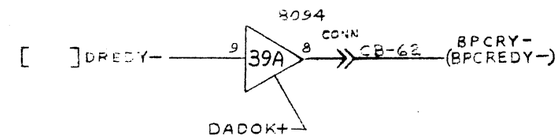
MATERIAL	DWN	PRIME COMPUTER, INC. NATICK, MASS.	
	CHK	I/O BUS INTERFACE LOGIC ADDRESS DECODING MPC2 (EV)	
UNLESS OTHERWISE SPECIFIED —REMOVE ALL BURRS AND SHARP EDGES —DIMENSIONS ARE IN INCHES —TOLERANCES .XX ±.02 .XXX ±.005 ANGLES ±1/2°	ENG.		
	APPRD		
	USED ON / NEXT ASSY	SCALE / SHEET 3 OF	SIZE C / DWG. NO. LBD1829 / REV. 11

PDF-003

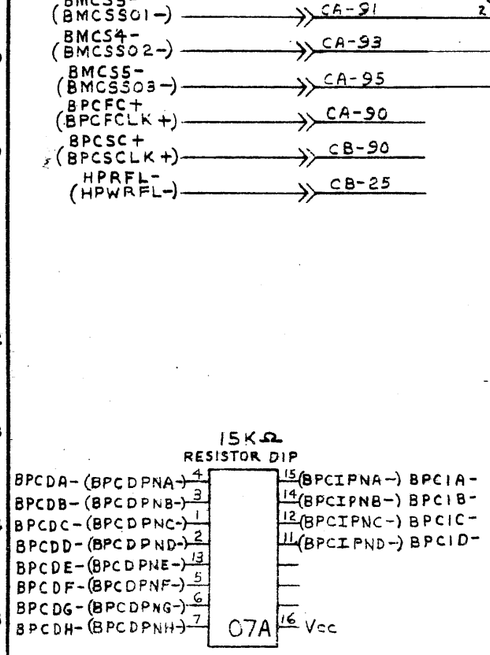


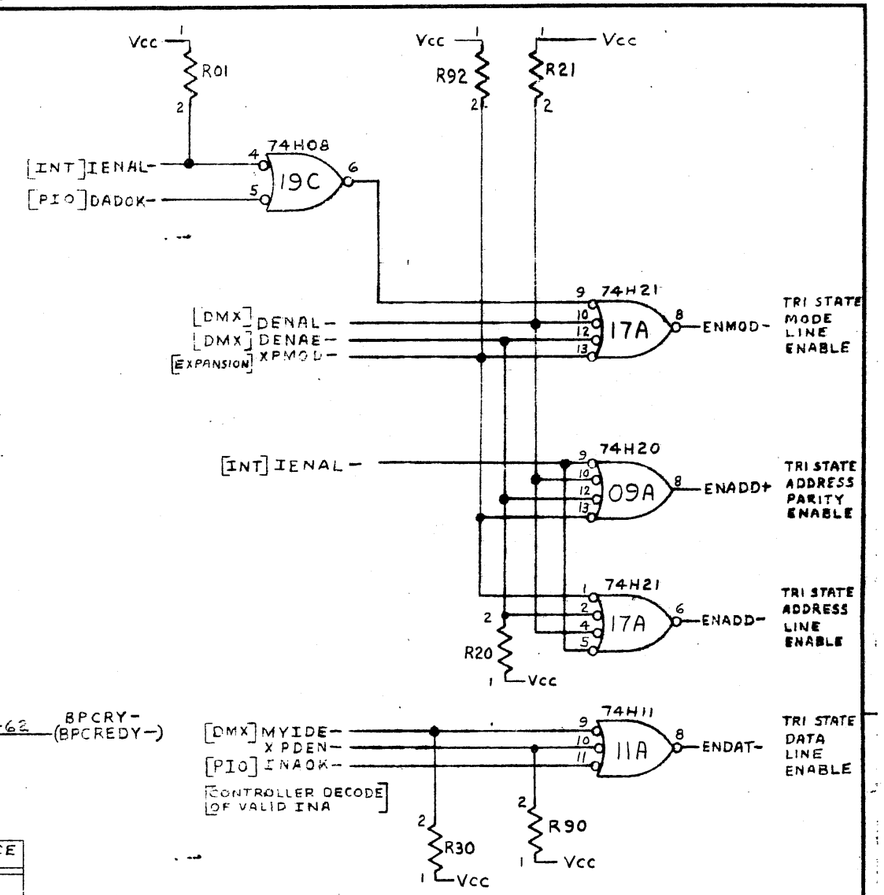






LBD MNEMONIC	EQUALS USER MNEMONIC	SOURCE LBD
COCMD +	MOD14-	10
ADD--+	SAME	10
DADOK-	SAME	9
XPMOD-	NOT USED	—
XPDEN-	NOT USED	—
INAOK-	SAME	9
OTAOK+	SAME	9
XPDEE-	NOT USED	—
DREDY-	SAME	8
DADOK+	SAME	9

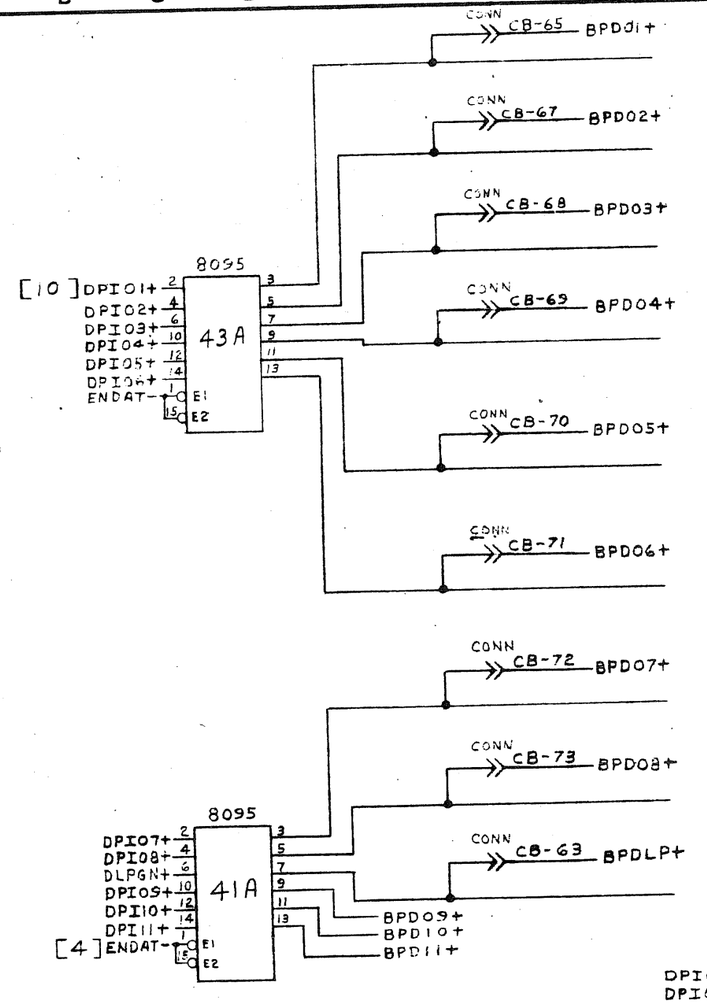


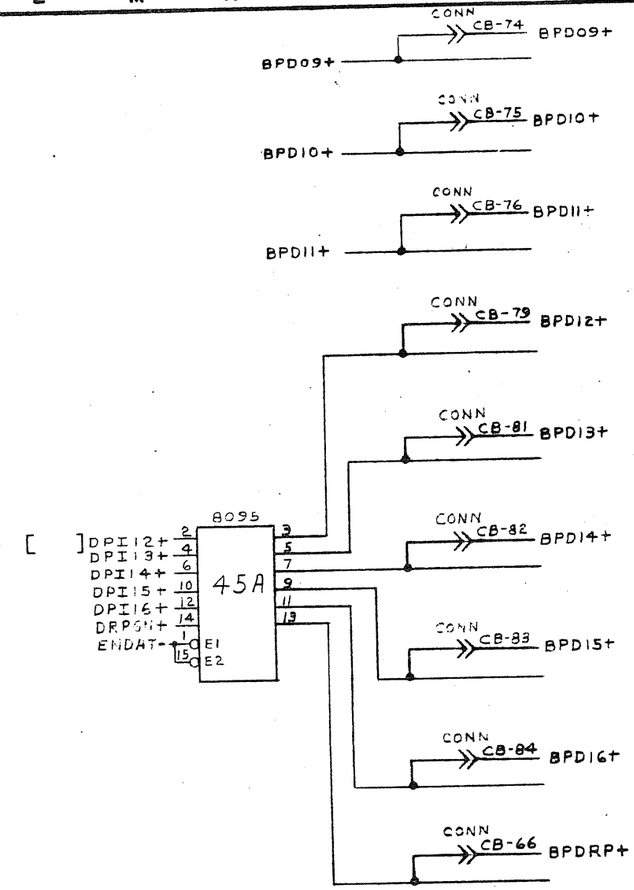


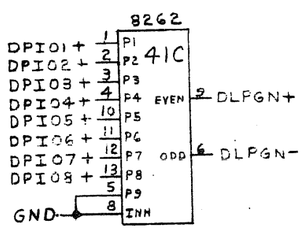
MATERIAL	DWN 3/15/73	PRIME COMPUTER, INC. NATICK, MASS.
	CHK	I/O BUS INTERFACE LOGIC ADDRESS PARITY MPC2 EV
UNLESS OTHERWISE SPECIFIED — REMOVE ALL BURRS AND SHARP EDGES: — DIMENSIONS ARE IN INCHES — TOLERANCES .XX ±.02 .XXX ±.005 ANGLES ±1/2°	ENG.	
	APPRD	
	USED ON	SCALE SHEET 4 OF
	NEXT ASSY	SIZE C DWG. NO. LBD1829 REV. A

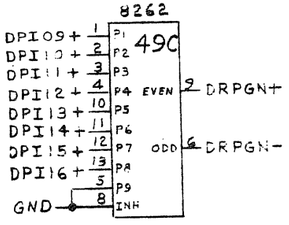
PDF-003

PRIME COMPUTER, INC.









LBD MNEMONIC	EQUALS USER MNEMONIC	SOURCE PIN
DPI +	SAME	10

III-05

MATERIAL	DWN. 3/15/73	PRIME COMPUTER, INC. NATICK, MASS.	
	CHK		
UNLESS OTHERWISE SPECIFIED —REMOVE ALL BURRS AND SHARP EDGES: —DIMENSIONS ARE IN INCHES —TOLERANCES .XX ±.02 .XXX ±.005 ANGLES ±1/2°	ENG.	I/O BUS INTERFACE LOGIC DATA BUS LINES MPC2	
	APPRD		
	USED ON / NEXT ASSY	SCALE / SHEET 5 OF	SIZE C / DWG. NO. LBD1829 / REV. A

PDF-003

PRIME COMPUTER, INC.

07A
Vcc
11 12 14 15

BPCIA – (BPCIPNA–) CONN CA-18
BPCIB – (BPCIPNB–) CA-19
BPCIC – (BPCIPNC–) CA-20
BPCID – (BPCIPND–) CA-21
74H20 9A
IPNOR–
BPCIPND– ENDS IN THIS CONTROLLER

74H04 7B
IPNOR+

74H53 5B
ISHIC – HPAIF +
GND
IPNRQ + ISHIC –
ICPNP –
IXPIP– [EXPANSION]
IXPIP+
CONN CA-17 BPCIO– (BPCIPNO–)

BPCOV– (BPCIOVI–) CONN CA-38
74H04 1A
ISHIP+
IENAP+ 74H00 13B
ISHIC–

CONTROLLER INTERRUPT REQUEST
IENAP– 74H00 11D
GCNIR–
74H00 13B
IPNRQ–
IPNOR+
HPAIF–
74H10 15B
GIRQ–

BPCIN+ (BPCIEN+) CONN CA-16
74H04 7B
IENAP–
74H04 7B
IENAP+
PULUP+G
CNIRQ+
74H74 9C
IPNRQ+
IPNRQ–

BPCPN+ (BPCICPN+) CONN CA-23
74H04 7B
ICPNP–
CINIT-A
74H08 15C
KIPNR–

74H00 13B
CTIRQ+
74H01 21B
OPEN COLLECTOR
CONN CA-92 BPCIR– (BPCIRQ–)
XGIRQ–
2
2KΩ
Vcc
R89

PULUP+G
LHPAI+
IENAP+
GND
KIPNR–
74H106 13C
IENAL+
IENAL–

IPNOR +
ISHIC –
IPNRQ +
74H11 11A

PULUP+G
LHPAI+
GND
KHPAI–
74H106 13C
HPAIF+
HPAIF–

PULUP+G
74H74 9C
KHPAI+
KHPAI–
74H08 15C
CINIT-A
CONTROLLER INITIALIZE

BPCHI+ (BPCCHI+) CONN CA-25
74H04 7B
KLHII–
HPAIF+

LBD MNEMONIC	EQUALS USER MNEMONIC	SOURCE LBD
CINIT-A	MINIT-A	9
CNIRQ+	SAME	9
XGIRQ–	NOT USED	–

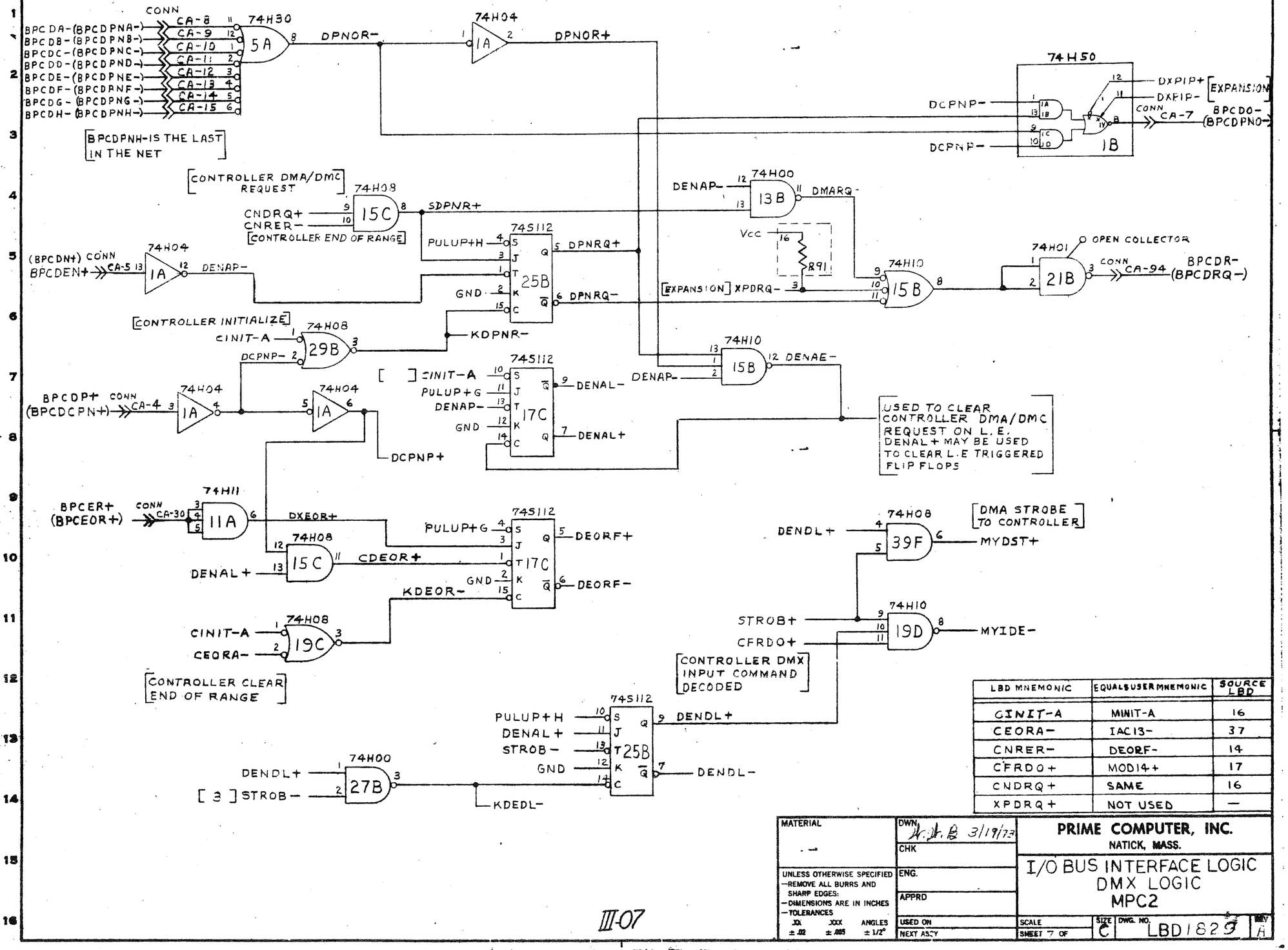
PRIME INTERRUPT LOGIC

III-06

MATERIAL	DWN 3/16/73	PRIME COMPUTER, INC. NATICK, MASS.
UNLESS OTHERWISE SPECIFIED –REMOVE ALL BURRS AND SHARP EDGES: – DIMENSIONS ARE IN INCHES – TOLERANCES .XX ± .02 .XXX ± .005 ANGLES ± 1/2°	CHK	I/O BUS INTERFACE LOGIC INTERRUPT LOGIC MPC2
	ENG.	
	APPRD	
USED ON / NEXT ASSY	SCALE / SHEET 6 OF	SIZE C / DWG. NO. LBD1829 / REV. A

PDF-003

PRIME COMPUTER, INC.

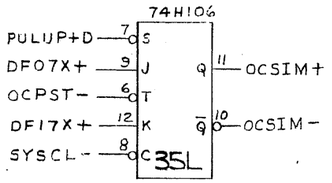


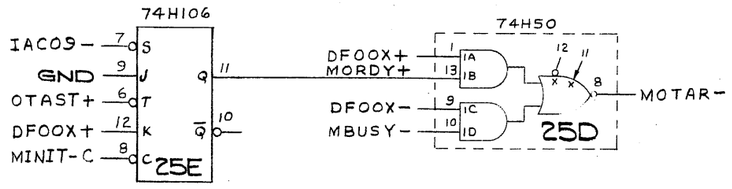
LBD MNEMONIC	EQUALS USER MNEMONIC	SOURCE LBD
CINIT-A	MINIT-A	16
CEORA-	IAC13-	37
CNRER-	DEORF-	14
CFRDO+	MOD14+	17
CNDRQ+	SAME	16
XPDRQ+	NOT USED	—

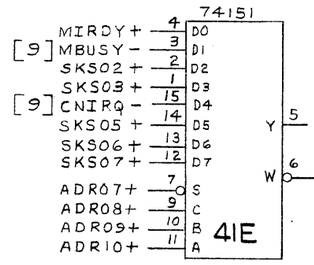
MATERIAL —	DWN H.H.B 3/19/73	PRIME COMPUTER, INC. NATICK, MASS.
UNLESS OTHERWISE SPECIFIED —REMOVE ALL BURRS AND SHARP EDGES: —DIMENSIONS ARE IN INCHES —TOLERANCES .XX ± .02 .XXX ± .005 ANGLES ± 1/2°	CHK	I/O BUS INTERFACE LOGIC DMX LOGIC MPC2
	ENG.	
	APPRD	
	USED ON / NEXT ASCY	SCALE / SHEET 7 OF
		SIZE C · DWG. NO. LBD1829 · REV A

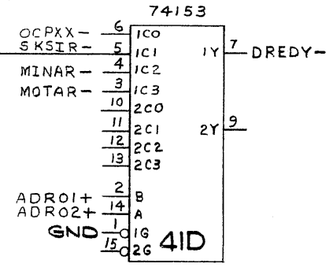
III-07

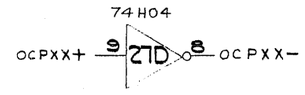
PDF-003



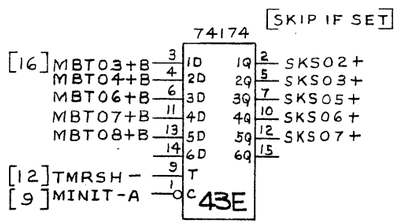


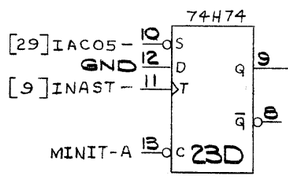


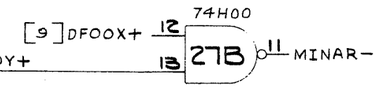


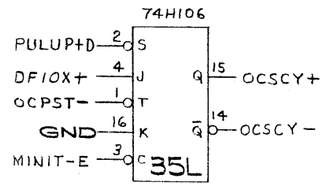


[SKIP IF SET]









MATERIAL	DWN 12/27/73	PRIME COMPUTER, INC. NATICK, MASS.
	CHK	OPTION MPC-2 (EV) MICROPROGRAMMED CONTROLLER SKS REG & READY
UNLESS OTHERWISE SPECIFIED —REMOVE ALL BURRS AND SHARP EDGES: —DIMENSIONS ARE IN INCHES —TOLERANCES .XX ± .02 .XXX ± .005 ANGLES ± 1/2°	ENG.	
	APPRD	
USED ON	SCALE	SIZE C — DWG. NO. LBD1829 — REV. B
NEXT ASSY	SHEET 8 OF	

PDF-003

PRIME COMPUTER, INC.

A B C D E F G H J K L M N P R S T V W X Y

ADDRESS DECODING

GROUND OR PULUP ALL UNUSED INPUTS

DEVICE ADDRESSES	
7/9 TAPE	14
FLOPPY DISK	12'
LP,CP,CR	03'

[3] DADXA- 74S04 37C DADXA+

DADXB- 74S04 37C DADXB+

DADXA+, DADXB+ → 74H08 29B

[2] FIOXX+, DADDC+10 → 74H08 29B → DADOK+ → 74H04 27D → DADOK-

OCPST+ → 74S00 43J → OCS00- → 74S74 43H (MBT05+B, TMRSL-, MINIT-C) → SKS14+, SKS14-

[3] OCPXX+ → 74H08 19C → OCPOK+

MINIT-C, IACIE- → 74H08 13M → KCNDR-

DF00X- → 74H04 27H → DF00X+

DF00X+ → 74H11 43G (INAXX+) → INAOK+

74H04 9D → INAOK-

74H04 11 9D 10 → CNIRQ+

DF15X- → 74H04 27H → DF15X+

PULUP+F → 74H106 (OCPST+, MINIT-A) → IMASK+, IMASK-

DF16X- → 74H04 27H → DF16X+

74H08 29B (OTAXX+) → OTAOK+

74H00 11D (IMASK+, CNIRQ+) → CNIRQ-

DF10X- → 74H04 43L → DF10X+

DF07X- → 74H04 43L → DF07X+

[29] IAC11- → 74H74 (GND, INTFF+) 13D

DF14X- → 74H04 27H → DF14X+ → 74H00 11D (OCPST+, MINIT-C, OCS14-)

DF17X- → 74H04 33L → DF17X+

DF17X-, DF07X- → 74H00 45L → DFX7X+

OCPOK+, STROB+ → 74H08 19C → OCPST+

OCPST+, OCS17- → 74H00 45L

SYSCL- → 74H00 45L → MINIT+

INAOK+, STROB+ → 74H00 27C → INAST-

74H04 33L → OCPST-

OTAOK+, STROB+ → 74H00 27C → OTAST-

OTAST- → 74H04 27D → OTAST+

MINIT+ →
- 74H04 43L → MINIT-B
- 74H04 37K → MINIT-D
- 74H04 33K → MINIT-A
- 74H04 43L → MINIT-E
- 74H04 37K → MINIT-C

III-09

DMX REQUEST

[29] IAC12-, GND, [7] DENAL+, KCNDR- → 74H74 23C → CNDRQ+

IAC12-, GND, MYDST+, PULUP+H, KCNDR- → 74H106 25E → MNDRQ+

LBD MNEMONIC	USER MNEMONIC
DADXA	
DADXB	
DADXC	
DADXD	

IAC12-, GND, DENAL-, KCNDR- → 74H74 23C → MADRQ+

PULUP+H, OTAST- → 74H74 23D → OTAFL+, OTAFL-

[29] IAC07- → 74H74 13D (MINIT-A, GND, [29] IACO8-, PULUP+G) → INLFL+, INLFL-

74H10 19D → MBUSY+

74H04 27D → MBUSY-

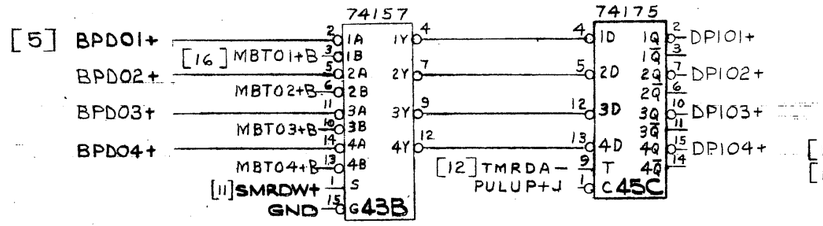
IACOE-, GND → 74H74 37J (IACO6-) → UBUSY+, UBUSY-

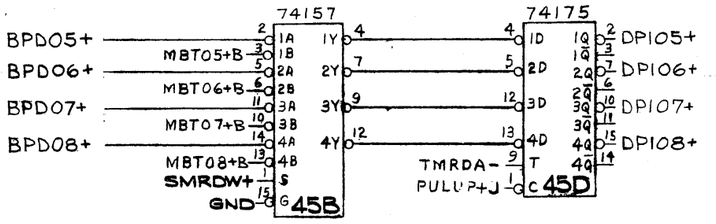
MATERIAL	DWN M.H.E. 4/2/73	PRIME COMPUTER, INC. NATICK, MASS.
	CHK	OPTION MPC-2 (EV)
UNLESS OTHERWISE SPECIFIED —REMOVE ALL BURRS AND SHARP EDGES: —DIMENSIONS ARE IN INCHES —TOLERANCES .XX ±.02 .XXX ±.005 ANGLES ±1/2°	ENG.	U-PROGRAMMED CONTROLLER PIO DECODE, BUSY & ATTENTION
	APPRD	
USED ON	SCALE	SIZE C DWG. NO. LBD1829 REV.
NEXT ASSY	SHEET 5 OF	

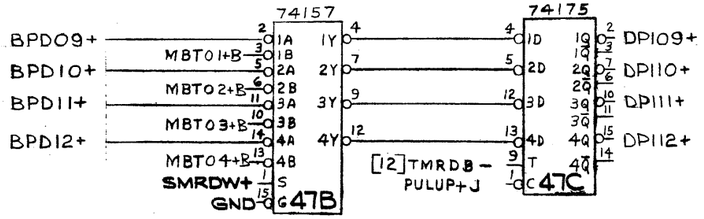
PDF-003

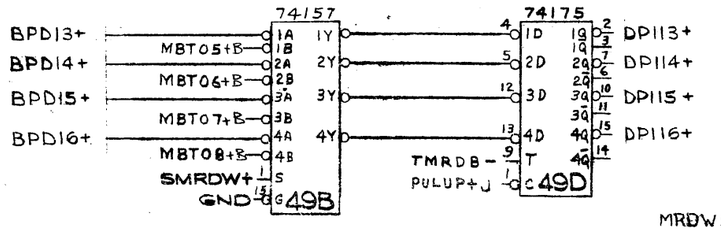
PRIME COMPUTER, INC.

DATA REG

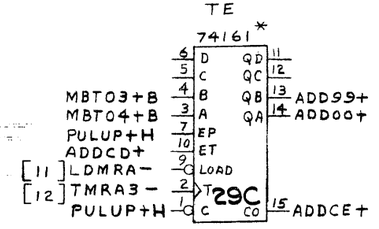


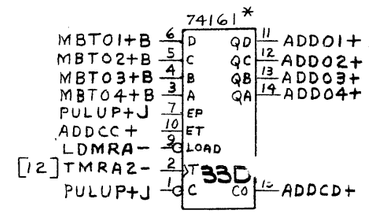


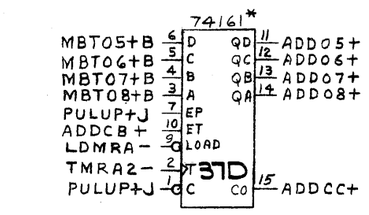


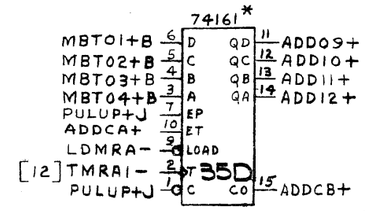


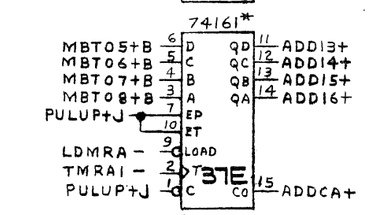
TE



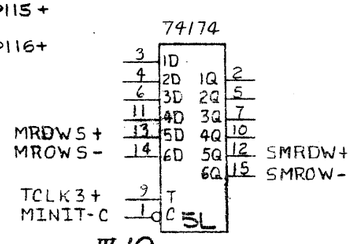




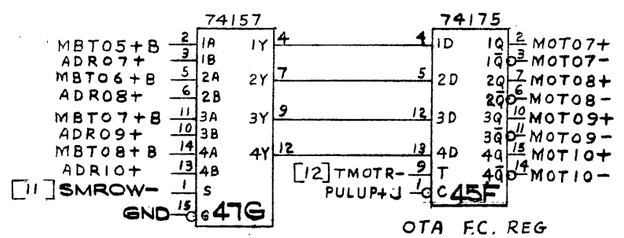




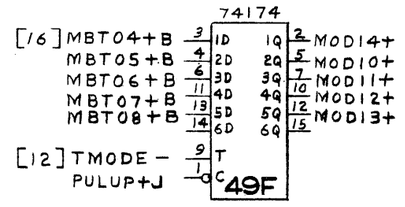
ADDRESS REGISTER



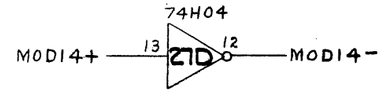
III-10

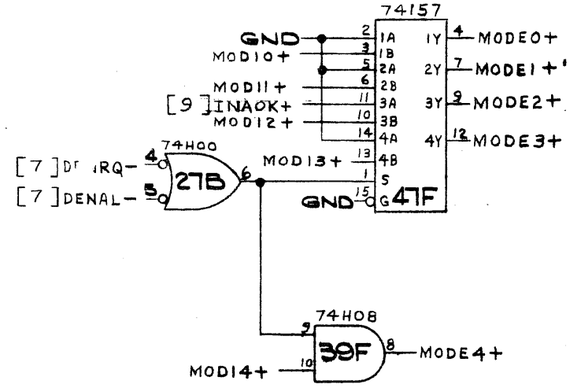


OTA F.C. REG



MODE LINE REGISTER

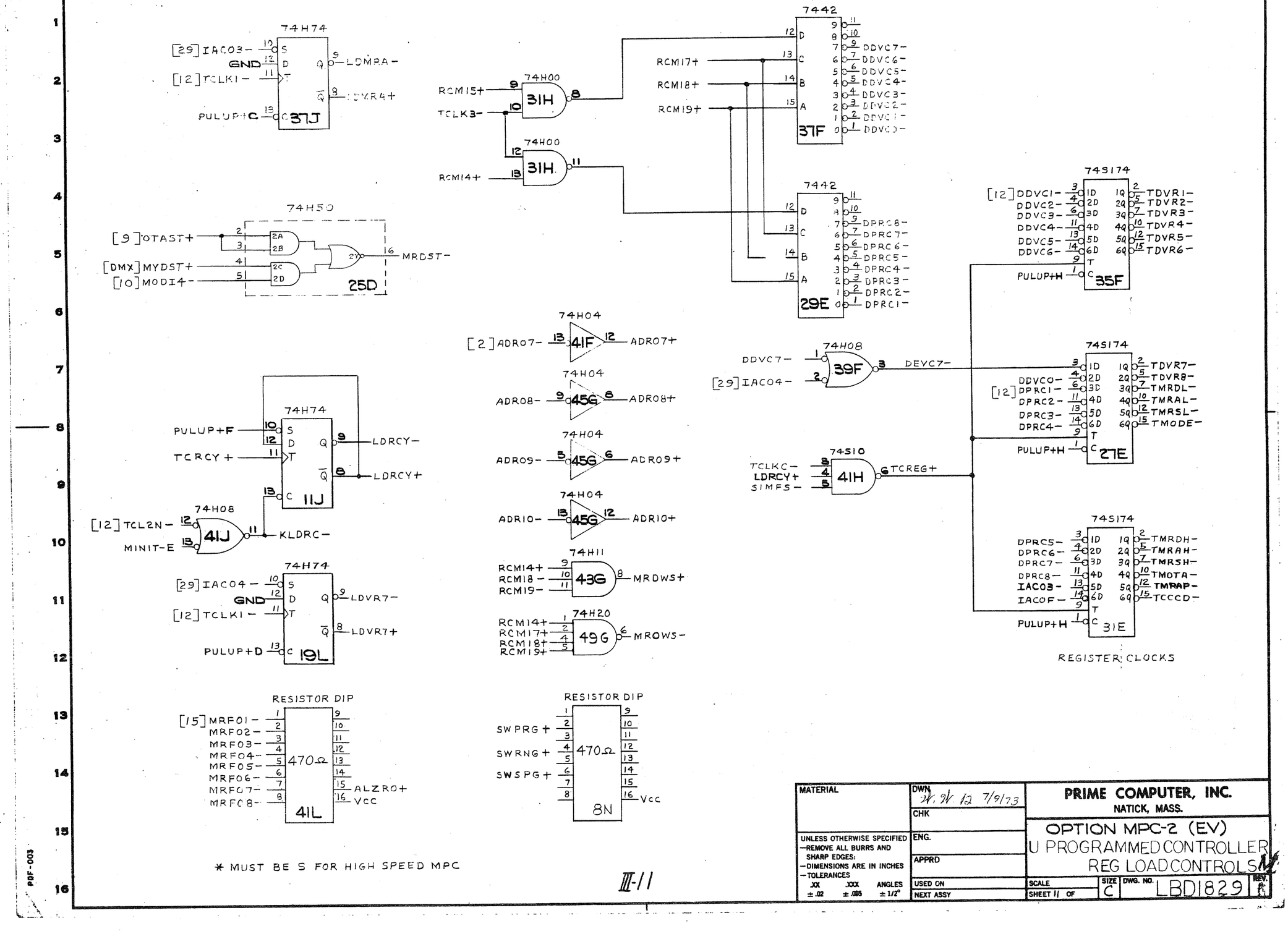




* NEED S IN HIGH SPEED MPC

MATERIAL	DWN JH JH. 13 3/20/73	PRIME COMPUTER, INC. NATICK, MASS.	
	CHK		
UNLESS OTHERWISE SPECIFIED -REMOVE ALL BURRS AND SHARP EDGES: -DIMENSIONS ARE IN INCHES -TOLERANCES .XX ± .02 .XXX ± 005 ANGLES ± 1/2°	ENG.	OPTION MPC-2 (EV) PROGRAMMED CONTROLLER ADDRESS & DATA REGISTERS	
	APPRD		
	USED ON / NEXT ASSY	SCALE / SHEET 10 OF	SIZE C / DWG. NO. LBD1829 / REV. A

PDF-003



* MUST BE S FOR HIGH SPEED MPC

III-11

MATERIAL	DWN W. W. 12 7/9/73	PRIME COMPUTER, INC. NATICK, MASS.
	CHK	OPTION MPC-2 (EV) U PROGRAMMED CONTROLLER REG LOAD CONTROLS
UNLESS OTHERWISE SPECIFIED —REMOVE ALL BURRS AND SHARP EDGES: —DIMENSIONS ARE IN INCHES —TOLERANCES .XX ±.02 .XXX ±.005 ANGLES ±1/2°	ENG.	
	APPRD	
	USED ON	SCALE — SIZE C — DWG. NO. LBD1829 — REV. A
	NEXT ASSY	SHEET 11 OF

PDF-003

OSCILLATOR
Vcc 14
GND 7
31J
1 OSCOF-
74S113
PULUP+E 4
3
2
15
29J
5 OSCXX+
6

74H08
TCLK3+ 12
SIMFS- 13
39F 11
74S112
PULUP+E 10
TCBHS+ 11
TCLKC+ 13
TCLKC+ 12
14
29J
9 RCYCE+
7 RCYCE-

SIMULATOR
GND
PROM
EXTERNAL SWITCH
[8] OCSIM- 3
SWSWG+ 4
5
74H10
39L 6
2
R87
1 Vcc
74H00
2
41M 3 SWSIM-
7414
11
29L
10
SWSMG- 1
TCLK3+ 9
74H08
SWSIM+ 10
41J 8

74S11
RCYCE+ 5
4
3
47H 6 TCRCY+
TCLKC+

74S74
PULUP+C 10
SIMDY+ 12
D
TCLKC+ 11
T
TCLK1- 15
C 43H
9 SIMFS+
8

74S112
PULUP+C 4
SWSCY+ 3
TCLK1+A 1
SWSCY-A 2
15
47L
5
6
PULUP+C 10
SCYFF+ 11
TCL2N- 13
12
14
47L
9 SCYFS+
7 SCYFF-

74S00
OSCXX+ 12
13
43J 11 TCLKC-
74S04
1
49J 2 TCLKC+

74H04
SWSCY-A 13
43L 12 SWSCY-A

SINGLE CYCLE
GND
RUN
EXTERNAL SWITCH
[8] OCSCY- 1
SWSCG+ 2
13
74H10
39L 12
2
R86
1 Vcc
74H00
5
41M 6 SWSCY-
7414
9
29L
8 SWSCG- 4
SWSCY-A 12
74H00
STRFS+ 13
45L 11 KSCYF-

MINIT-E
START MOMENTARY
GND
EXTERNAL SWITCH
2
R88
1 Vcc
SWSTR+
74S112
10
11
PULUP+C 13
GND 12
14
49L
9 STRFF+
7
74S112
4
3
1
OSCXX+ 2
STRFF-
15
49L
5
6
STRFS-
TCLK3-

TCLK2+ 4
STRFS+
TCLKC- 5
74S00
43J 6 TCL2N-

[11] MRDST-
TMRDH-
74S11
31D 12 TMRDA-
3
4
5
31D 6 TMRDB-
TMRDL-

[11] OTADT-
TMOTA-
74S11
5
4
3
33E 6 TMOTR-

74S10
RCYCE+ 1
2
TCLKC- 13
41H 12 TWIRF-

74S194
PULUP+C 2 SR
3 A
4 B
5 C
6 D
9 SO
TCLK1+A 10 S1
TCLKC+ 11 CLK
GND 7 SL
MINIT-C 1 CL
45J
15 TCLKL+
14 TCLK1+
QB
13 TCLK2+
QC
12 TCLK3+
QD

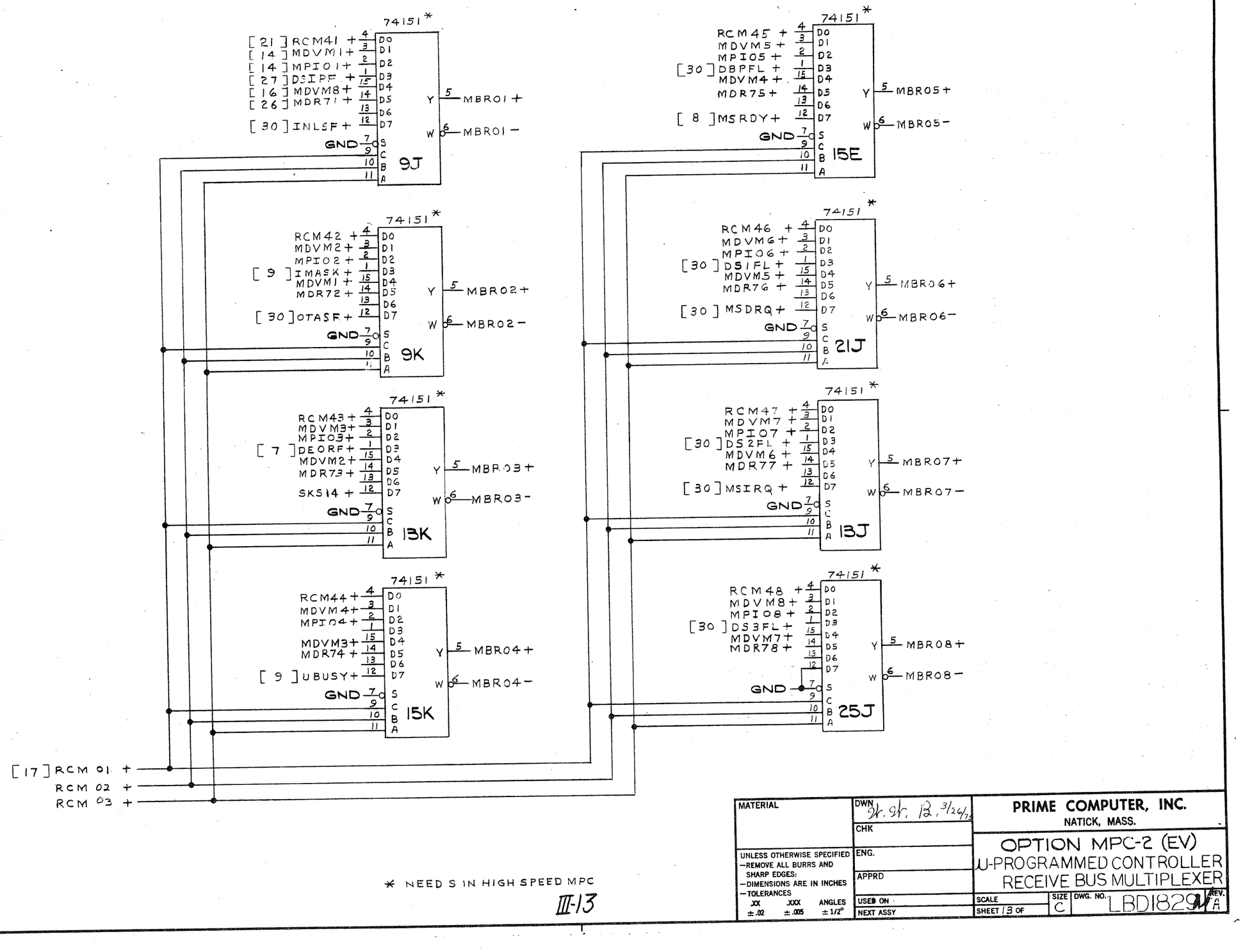
[11] TMRAL-
74S11
33E 12 TMRA1-
TMRAH- 10
11
9
33E 8 TMRA2-
74S11
TMOTA- 9
10
[29] TMRAP- 11
31D 8 TMRA3-
74H04
3
43L 4 TCLIA- TO FEP
74S04
3
49J 4 TCLKI-
74S11
1
2
13
47H 12 TCLKI+A
74S11
11
10
9
47H 8 TCLKI+B
74S04
5
49J 6 TCLK3-

ENGINEERING PANEL

III-12

MATERIAL	DWN 2K.H.B 7/11/73	PRIME COMPUTER, INC. NATICK, MASS.
	CHK	
UNLESS OTHERWISE SPECIFIED —REMOVE ALL BURRS AND SHARP EDGES: —DIMENSIONS ARE IN INCHES —TOLERANCES .XX ±.02 .XXX ±.005 ANGLES ±1/2°	ENG.	OPTION MPC-2 (EV) U-PROGRAMMED CONTROLLER CLOCK
	APPRD	
USED ON	SCALE	SIZE C DWG. NO. LBD1829 REV. B
NEXT ASSY	SHEET 12 OF	

PDF-003



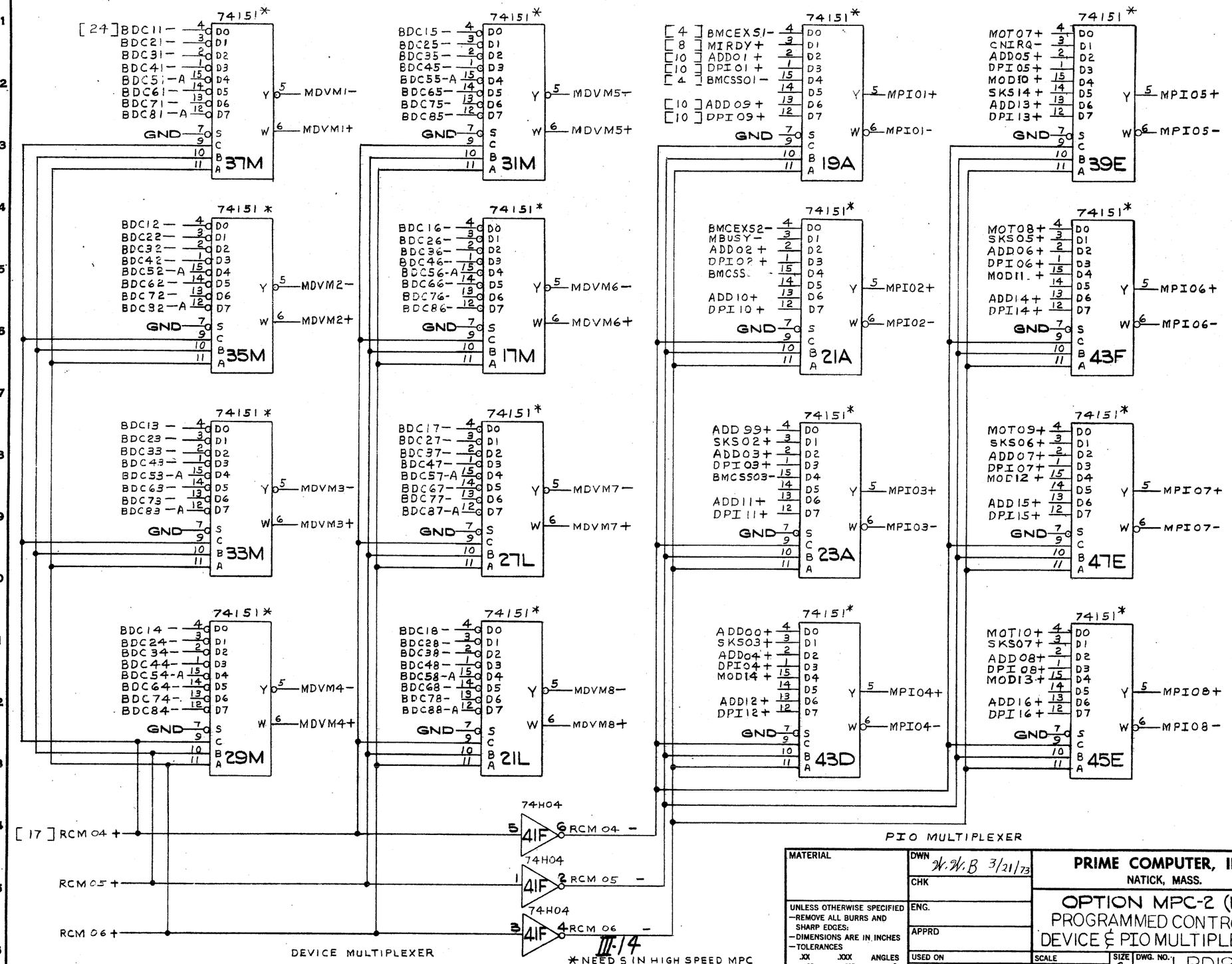
* NEEDS IN HIGH SPEED MPC

III-13

MATERIAL	DWN W. H. B. 3/26/75	PRIME COMPUTER, INC. NATICK, MASS.
	CHK	
UNLESS OTHERWISE SPECIFIED —REMOVE ALL BURRS AND SHARP EDGES: —DIMENSIONS ARE IN INCHES —TOLERANCES XX ±.02 XXX ±.005 ANGLES ±1/2°	ENG.	OPTION MPC-2 (EV) U-PROGRAMMED CONTROLLER RECEIVE BUS MULTIPLEXER
	APPRD	
	USED ON	SCALE — SIZE C DWG. NO. LBD1829 REV. A
	NEXT ASSY	SHEET 13 OF

PDF-003

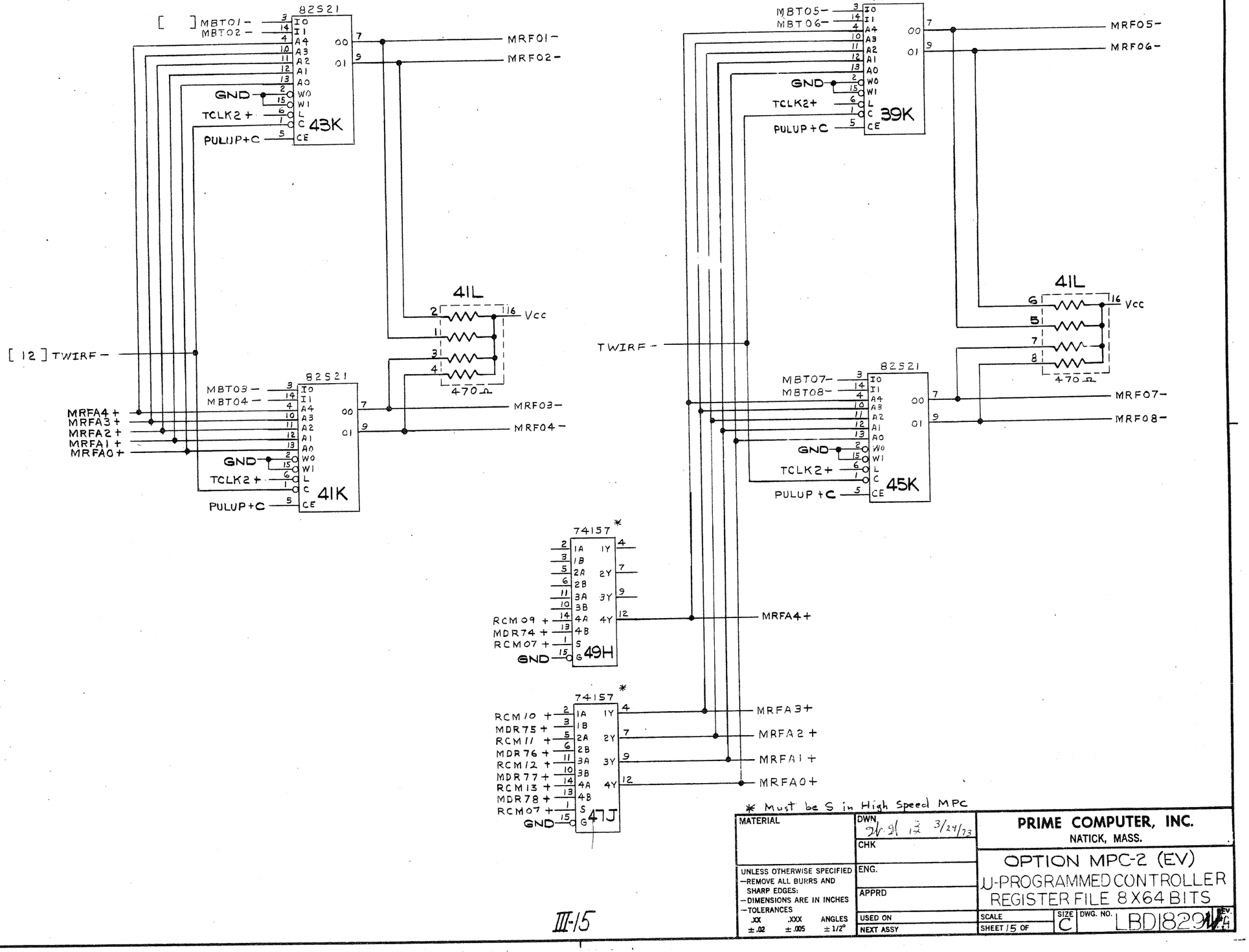
PRIME COMPUTER, INC.



III-14

MATERIAL	DWN W.W.B 3/21/73	PRIME COMPUTER, INC. NATICK, MASS.	
	CHK		
UNLESS OTHERWISE SPECIFIED —REMOVE ALL BURRS AND SHARP EDGES; —DIMENSIONS ARE IN INCHES —TOLERANCES .XX ±.02 .XXX ±.005 ANGLES ±1/2°	ENG.	OPTION MPC-2 (EV) PROGRAMMED CONTROLLER DEVICE & PIO MULTIPLEXERS	
	APPRD		
	USED ON	SCALE	SIZE C DWG. NO. LBD1829 REV. A
	NEXT ASSY	SHEET 14 OF	

PDF-003



* Must be S in High Speed MPC

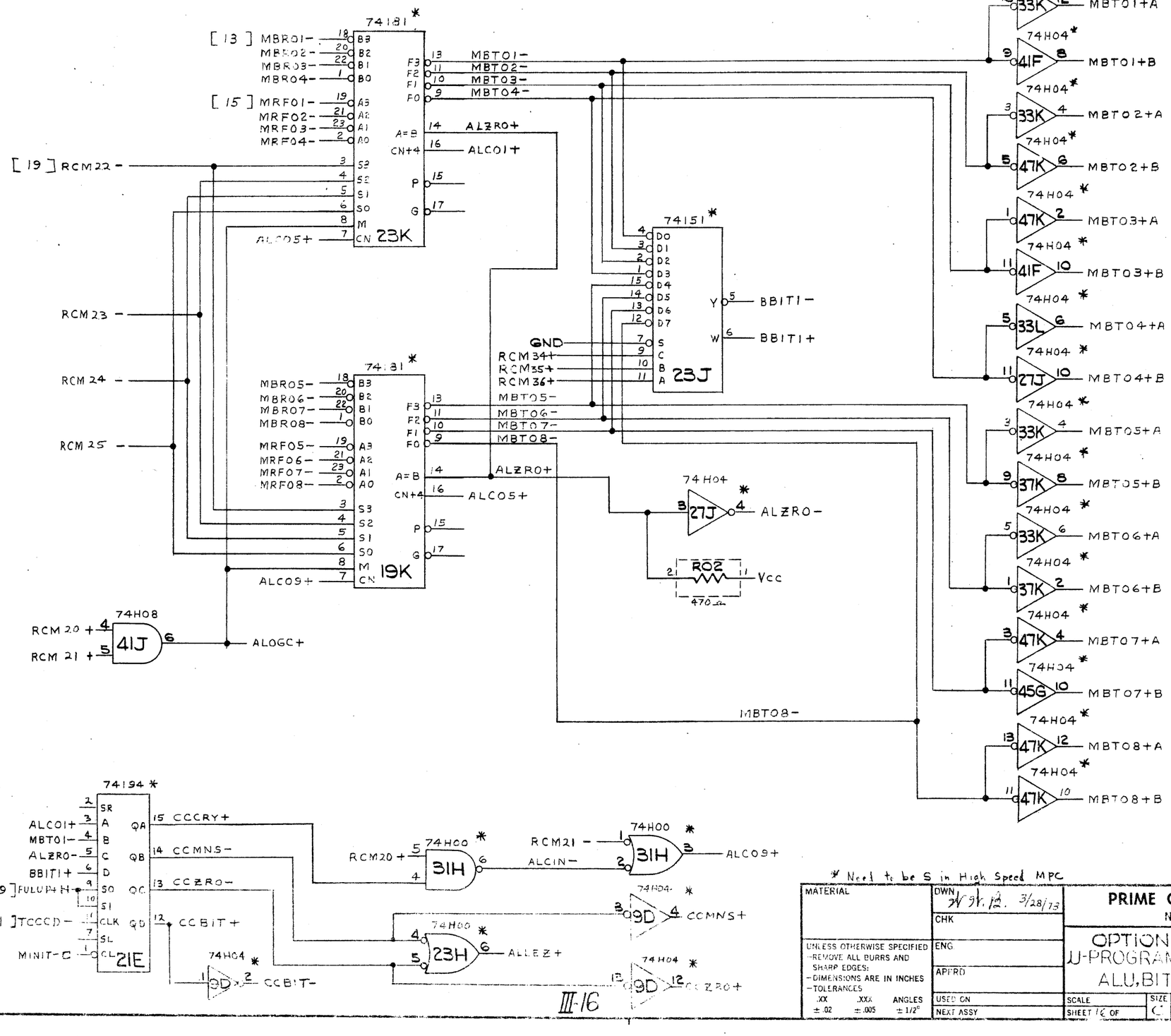
MATERIAL	DWN. 3/24/73	PRIME COMPUTER, INC. NATICK, MASS.
UNLESS OTHERWISE SPECIFIED –REMOVE ALL BURRS AND SHARP EDGES: –DIMENSIONS ARE IN INCHES –TOLERANCES .XX ±.02 .XXX ±.005 ANGLES ±1/2°	CHK	OPTION MPC-2 (EV)
	ENG.	U-PROGRAMMED CONTROLLER REGISTER FILE 8X64 BITS
	APPRD	
	USED ON / NEXT ASSY	SCALE / SHEET 15 OF
		SIZE C DWG. NO. LBD18229 REV. H

PRIME COMPUTER, INC.

Arithmetic Unit

Bit Test

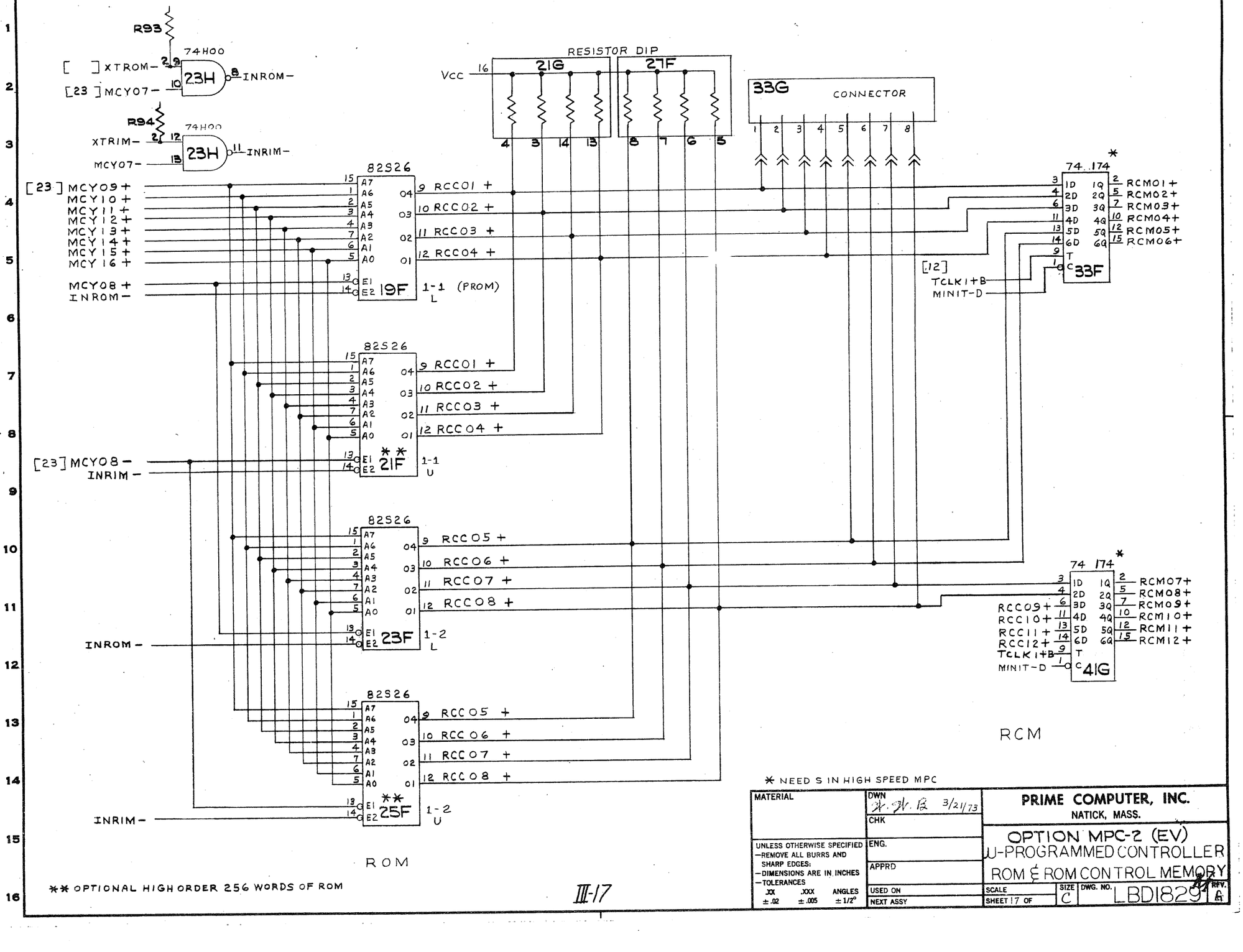
Transmit Bus



* Need to be S in High Speed MPC

MATERIAL	DWN 3/28/73	PRIME COMPUTER, INC. NATICK, MASS.
	CHK	
UNLESS OTHERWISE SPECIFIED –REMOVE ALL BURRS AND SHARP EDGES: –DIMENSIONS ARE IN INCHES –TOLERANCES .XX ±.02 .XXX ±.005 ANGLES ±1/2°	ENG	OPTION MPC-2 (EV) U-PROGRAMMED CONTROLLER ALU, BIT TEST AND BT
	APPRD	
	USED ON	SCALE SIZE C DWG NO LBD1S201 REV. A
	NEXT ASSY	SHEET 16 OF

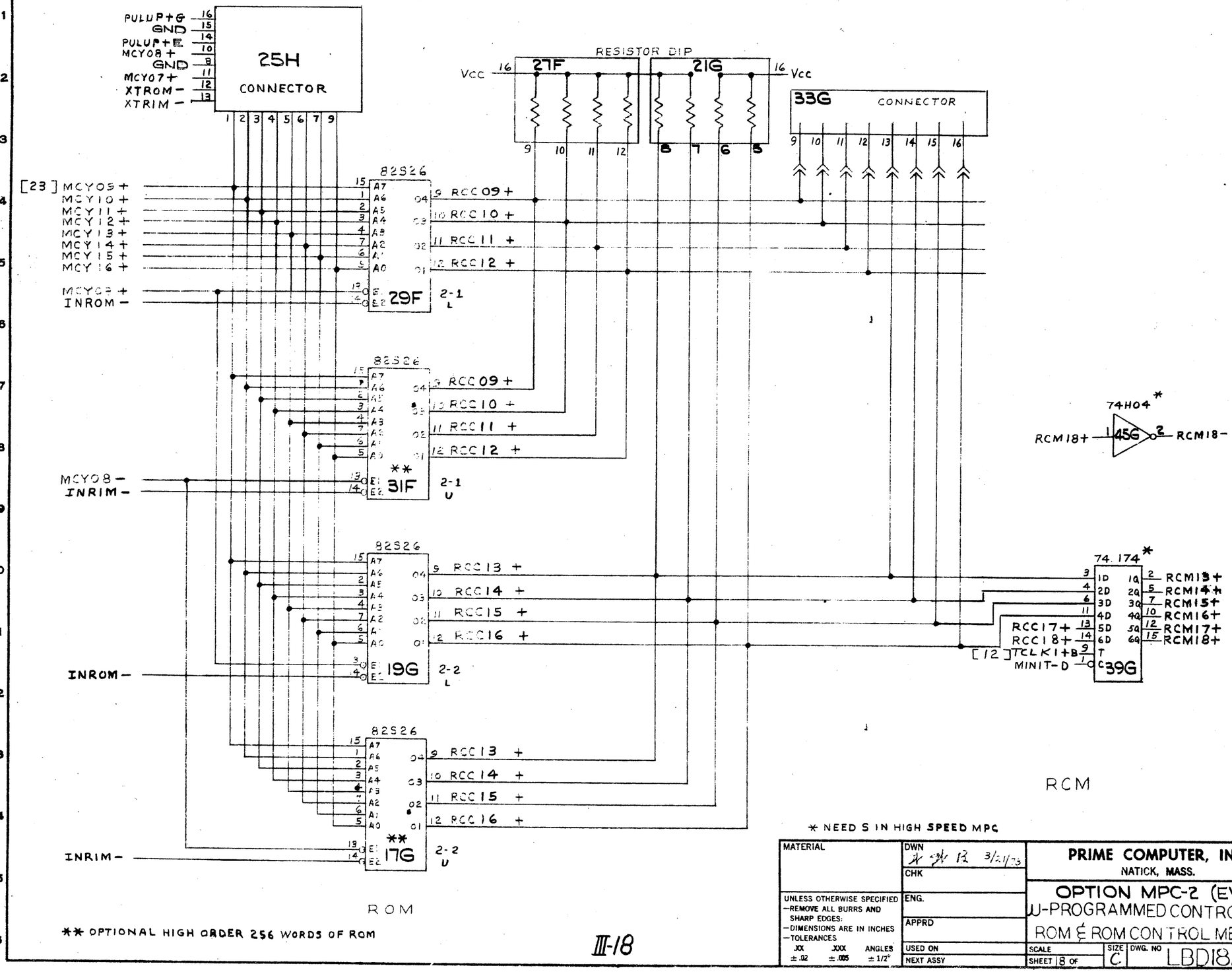
PDF-003



* NEEDS IN HIGH SPEED MPC

** OPTIONAL HIGH ORDER 256 WORDS OF ROM

MATERIAL	DWN W. W. 12 3/21/73	PRIME COMPUTER, INC. NATICK, MASS.
	CHK	
UNLESS OTHERWISE SPECIFIED –REMOVE ALL BURRS AND SHARP EDGES: –DIMENSIONS ARE IN INCHES –TOLERANCES .XX ±.02 .XXX ±.005 ANGLES ±1/2°	ENG.	OPTION MPC-2 (EV) U-PROGRAMMED CONTROLLER ROM & ROM CONTROL MEMORY
	APPRD	
	USED ON / NEXT ASSY	SCALE / SHEET 17 OF
		SIZE C
		DWG. NO. LBD1829
		REV. A



** OPTIONAL HIGH ORDER 256 WORDS OF ROM

* NEED S IN HIGH SPEED MPC

MATERIAL	DWN	PRIME COMPUTER, INC. NATICK, MASS.	
	CHK		
UNLESS OTHERWISE SPECIFIED —REMOVE ALL BURRS AND SHARP EDGES; —DIMENSIONS ARE IN INCHES —TOLERANCES .XX ±.02 .XXX ±.005 ANGLES ±1/2°	ENG.	OPTION MPC-2 (EV) U-PROGRAMMED CONTROLLER ROM & ROM CONTROL MEMORY	
	APPRD		
	USED ON	SCALE	SIZE C
	NEXT ASSY	SHEET 8 OF	DWG. NO LBD1829 REV. A

PRIME COMPUTER, INC.

A B C D E F G H J K L M N P R S T V W X Y

RESISTOR DIP

Vcc 16 — 27F (3 15 14 2) — 13F (2 15 14 1)

35G CONNECTOR 1 2 3 4 5 6 7 8

82S26

[23] MCY09 +
MCY10 +
MCY11 +
MCY12 +
MCY13 +
MCY14 +
MCY15 +
MCY16 +
MCY08 +
INROM –

A7 A6 A5 A4 A3 A2 A1 A0 → O4 O3 O2 O1
E1 E2 29G 3-1 L

9 RCC17 +
10 RCC18 +
11 RCC19 +
12 RCC20 +

82S26

MCY08 –
INRIM –

A7 A6 A5 A4 A3 A2 A1 A0 → O4 O3 O2 O1
E1 E2 27G 3-1 U **

9 RCC17 +
10 RCC18 +
11 RCC19 +
12 RCC20 +

82S26

INROM –

A7 A6 A5 A4 A3 A2 A1 A0 → O4 O3 O2 O1
E1 E2 15G 3-2 L

9 RCC21 +
10 RCC22 +
11 RCC23 +
12 RCC24 +

82S26

INRIM –

A7 A6 A5 A4 A3 A2 A1 A0 → O4 O3 O2 O1
E1 E2 13G 3-2 U **

9 RCC21 +
10 RCC22 +
11 RCC23 +
12 RCC24 +

ROM

** OPTIONAL HIGH ORDER 256 WORDS OF ROM

74H04 *
RCM19+ — 3 45G 4 — RCM19–

74H04 *
RCM21+ — 9 27J 8 — RCM21–

74H04 *
RCM22+ — 3 37K 4 — RCM22–

74H04 *
RCM23+ — 5 37K 6 — RCM23–

74H04 *
RCM24+ — 13 27J 12 — RCM24–

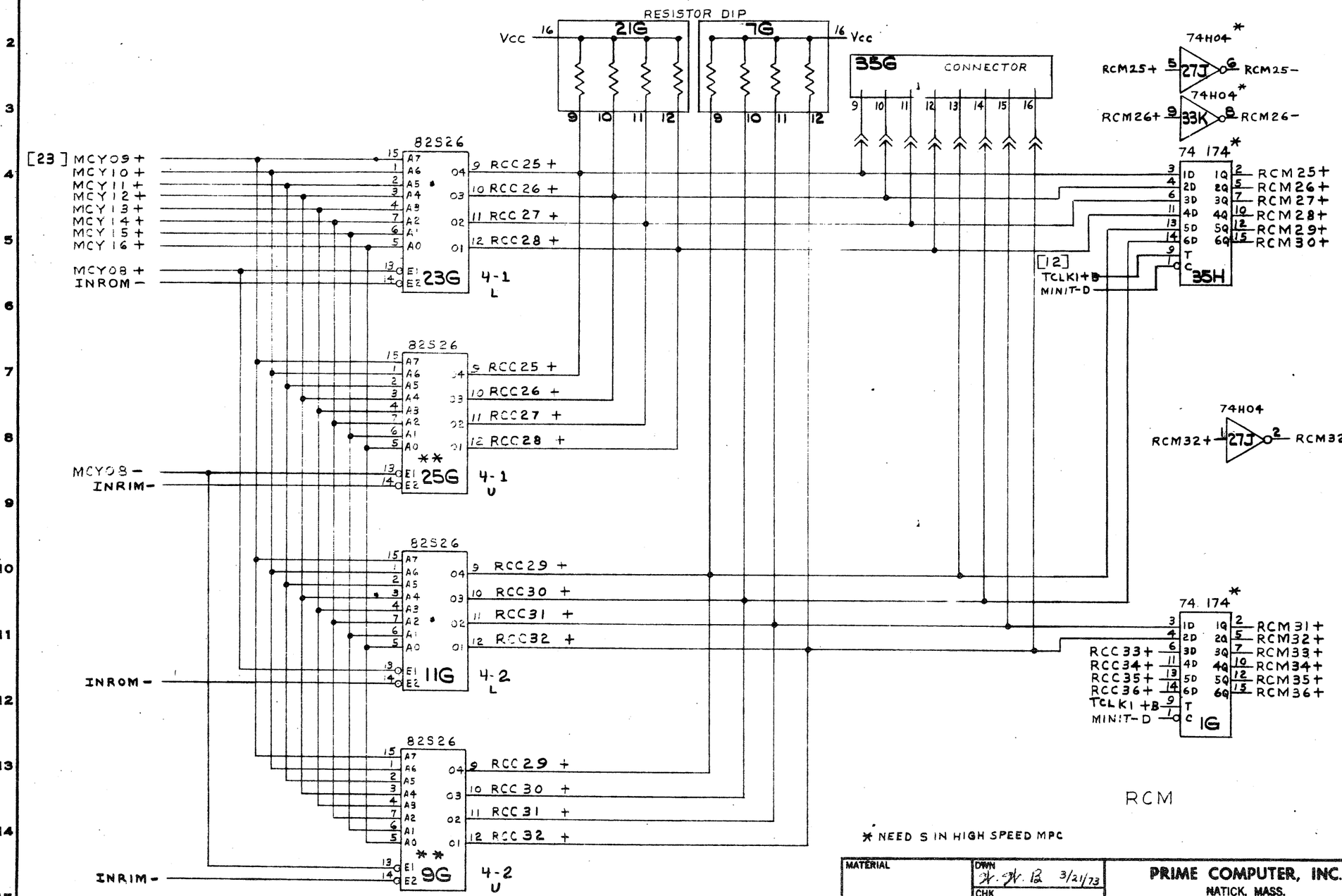
74 174 *
3D 4D 5D 6D → 1Q 2Q 3Q 4Q 5Q 6Q
2 RCM19+
5 RCM20+
7 RCM21+
10 RCM22+
12 RCM23+
15 RCM24+
T C
[12] TCLK1+B
37H

RCM

* NEED 5 IN HIGH SPEED MPC

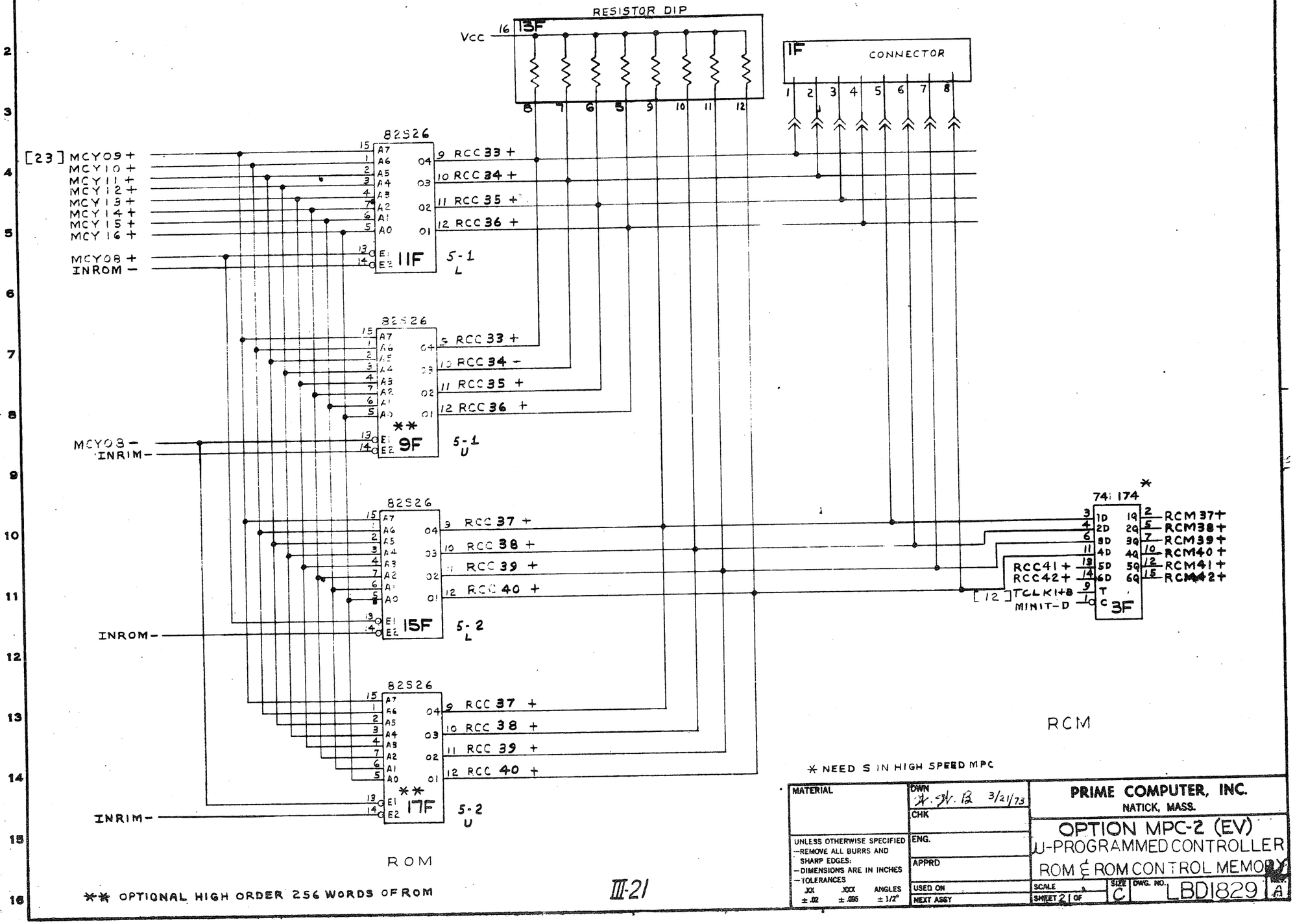
MATERIAL	DRWN J.M.B. 3/21/80	PRIME COMPUTER, INC. NATICK, MASS.
UNLESS OTHERWISE SPECIFIED — REMOVE ALL BURRS AND SHARP EDGES — DIMENSIONS ARE IN INCHES — TOLERANCES .XX ±.02 .XXX ±.005 ANGLES ±.5° ±.1°	CHK	OPTION MPC-2 (EV) U-PROGRAMMED CONTROLLER ROM & ROM CONTROL MEMORY
	ENGR	
	APPRD	
	USED ON	SCALE
	NEXT ASSY	SHEET 19 OF — SIZE C — DWG. NO. LBD1829 — REV. A

III-19



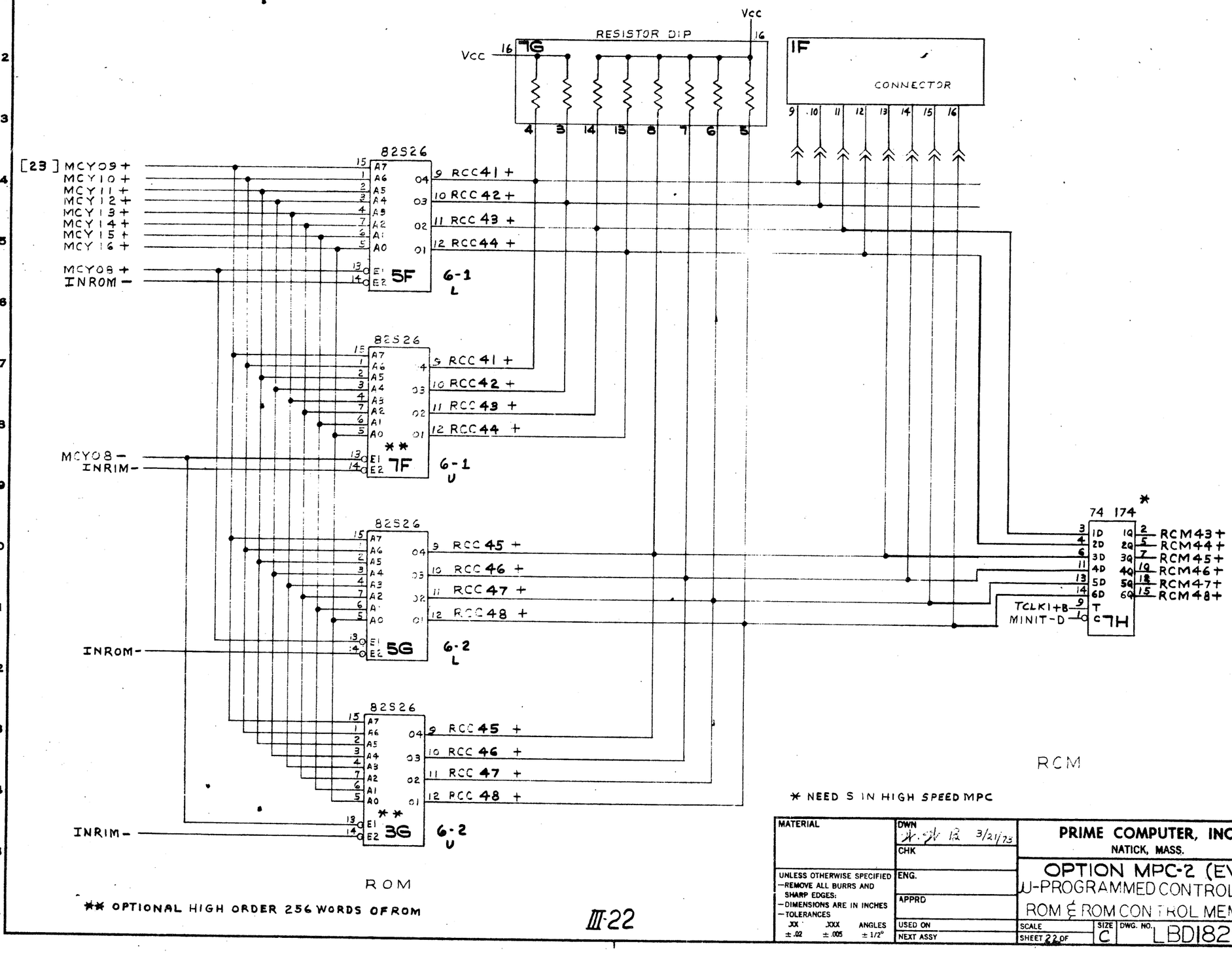
MATERIAL	DWN M.W.B 3/21/73	PRIME COMPUTER, INC. NATICK, MASS.
	CHK	
UNLESS OTHERWISE SPECIFIED —REMOVE ALL BURRS AND SHARP EDGES: —DIMENSIONS ARE IN INCHES —TOLERANCES .XX ±.02 .XXX ±.005 ANGLES ±1/2°	ENG.	OPTION MPC-2 (EV) U-PROGRAMMED CONTROLLER ROM & ROM CONTROL MEMORY
	APPRD	
	USED ON	SCALE
	NEXT ASSY	SHEET 20 OF
		SIZE C DWG. NO. LBD1829 REV. A

PDF-003



MATERIAL	DWN A.J.V. P2 3/21/73	PRIME COMPUTER, INC. NATICK, MASS.	
UNLESS OTHERWISE SPECIFIED —REMOVE ALL BURRS AND SHARP EDGES: —DIMENSIONS ARE IN INCHES —TOLERANCES XX ±.02 XXX ±.005 ANGLES ±1/2°	CHK	OPTION MPC-2 (EV) U-PROGRAMMED CONTROLLER ROM & ROM CONTROL MEMORY	
	ENG.		
	APPRD		
	USED ON	SCALE	SIZE C DWG. NO. LBD1829 REV. A
	NEXT ASSY	SHEET 2 OF	

PRIME COMPUTER, INC.



MATERIAL	DWN 12 3/21/75	PRIME COMPUTER, INC. NATICK, MASS.
	CHK	
UNLESS OTHERWISE SPECIFIED —REMOVE ALL BURRS AND SHARP EDGES: —DIMENSIONS ARE IN INCHES —TOLERANCES .XX ±.02 .XXX ±.005 ANGLES ±1/2°	ENG.	OPTION MPC-2 (EV) U-PROGRAMMED CONTROLLER ROM & ROM CONTROL MEMORY
	APPRD	
	USED ON	SCALE SIZE C DWG. NO. LBD1829
	NEXT ASSY	SHEET 22 OF

PSF-005

74S64 5J
RCM32+
RCM33+
[29] IACPP -
RCM32-
MJUMP+
GND
8

74153 * 11H
RCM39+
RCY07+
PRY07+
RCM40+
RCY08+
PRY08+
GD02J+
MCY07+
MCY08+

74153 * 9H
RCM41+
MBR01+
RCY09+
PRY09+
RCM42+
MBR02+
RCY10+
PRY10+
GD06J+
MCY09+
MCY10+

74153 * 13H
RCM43+
MBR03+
RCY11+
PRY11+
RCM44+
MBR04+
RCY12+
PRY12+
GD10J+
MCY11+
MCY12+

74153 * 19H
RCM45+
MBR05+
RCY13+
PRY13+
RCM46+
MBR06+
RCY14+
PRY14+
GD12J+
MCY13+
MCY14+

74153 * 21H
RCM47+
MBR07+
RCY15+
PRY15+
RCM48+
MBR08+
RCY16+
PRY16+
GD02F+
MCY15+
MCY16+

74157 * 7J
[29] IAC PP +
GND
RCM34+
RCM35+
RCM36+
IAC PP -
1A 1B 2A 2B 3A 3B 4A 4B S G
1Y 2Y 3Y 4Y

74H08 41J
IAC PP -

74S00 43J
RCM32+
RCM33+
RM332-

74161 15H
PULUP+E
RCYCB+
[11] LDRCY-
[12] TCRCY+
[29] IACZR-
D C B A EP ET LOAD T C
QD QC QB QA Co
RCY07+
RCY08+
RCYCC+

74161 17H
PULUP+E
RCYCA+
[11] LDRCY-
[12] TCRCY+
IACZR-
RCY09+
RCY10+
RCY11+
RCY12+
RCYCB+

74161 17J
PULUP+E
[11] LDRCY-
[12] TCRCY+
IACZR-
RCY13+
RCY14+
RCY15+
RCY16+
RCYCA+
RCY

74174 * 5H
ID 2D 3D 4D 5D 6D T C
1Q 2Q 3Q 4Q 5Q 6Q
IACO2-
PULUP+F
PRY07+
PRY08+
PRY09+
PRY10+

74174 * 19J
[29] IACO2-
PULUP+E
PRY11+
PRY12+
PRY13+
PRY14+
PRY15+
PRY16+
PUSHED RCY

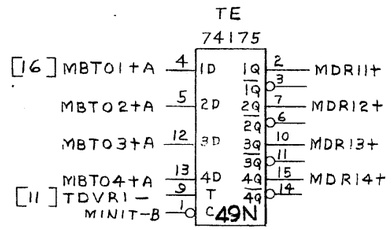
74H04 27H
MCY07+ → MCY07-
MCY08+ → MCY08-

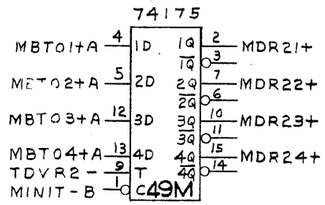
ROM ADDRESS MULTIPLEXER

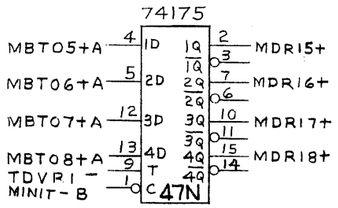
* NEED 6 IN HIGH SPEED MPC

MATERIAL	DWN JK.9F.12 4/3/73	PRIME COMPUTER, INC. NATICK, MASS.
	CHK	OPTION MPC-2 (EV)
UNLESS OTHERWISE SPECIFIED —REMOVE ALL BURRS AND SHARP EDGES: —DIMENSIONS ARE IN INCHES —TOLERANCES .XX ± .02 .XXX ± .005 ANGLES ± 1/2°	ENG.	U-PROGRAMMED CONTROLLER ROM ADDRESS, RCY, PUSHED RCY
	APPRD	
USED ON	SCALE	SIZE C DWG. NO. LBD1829 REV. A
NEXT ASSY	SHEET 23 OF	

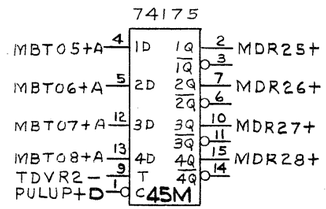
PDF-003







DEVICE REGISTER 1



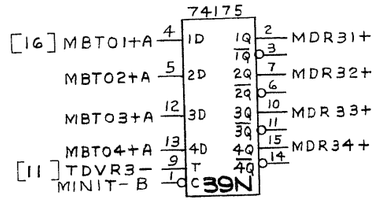
DEVICE REGISTER 2

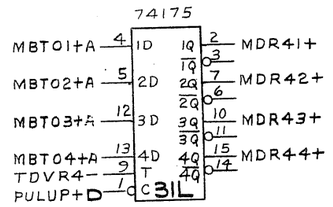
* NEED S IN HIGH SPEED MPC

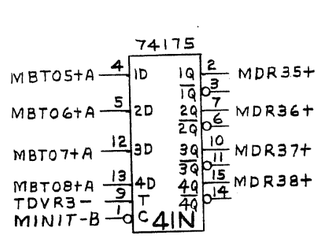
III-24

MATERIAL	DWN M.P. 7/11/73	PRIME COMPUTER, INC. NATICK, MASS.
	CHK	
UNLESS OTHERWISE SPECIFIED –REMOVE ALL BURRS AND SHARP EDGES: –DIMENSIONS ARE IN INCHES –TOLERANCES .XX ±.02 .XXX ±.005 ANGLES ±1/2°	ENG.	OPTION MPC-2 (EV) U-PROGRAMMED CONTROLLER DEVICE REGISTERS 1 AND 2
	APPRD	
USED ON	SCALE	SIZE C DWG. NO. LBD1829 REV. A
NEXT ASSY	SHEET 24 OF	

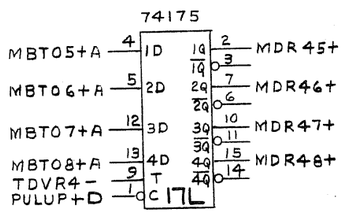
PDF-003



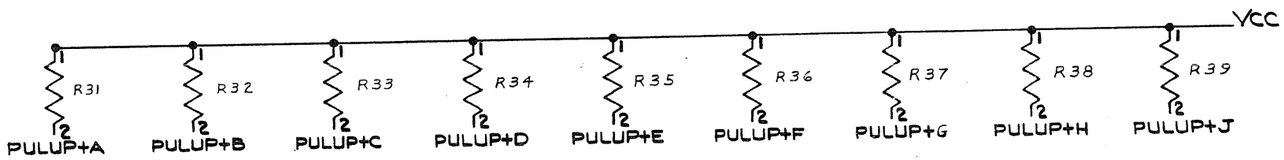




DEVICE REGISTER 3



DEVIC_ REGISTER 4

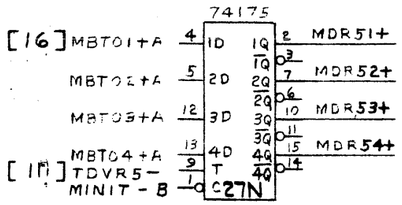


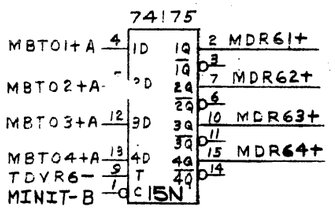
*NEED S IN HIGHSPEED MPC

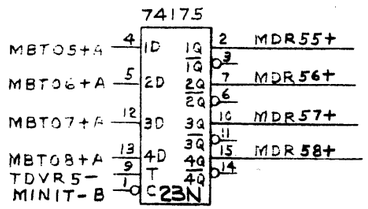
III-25

MATERIAL	DWN 7/11/73	PRIME COMPUTER, INC. NATICK, MASS.
	CHK	OPTION MPC-2 (EV)
UNLESS OTHERWISE SPECIFIED —REMOVE ALL BURRS AND SHARP EDGES: —DIMENSIONS ARE IN INCHES —TOLERANCES .XX ±.02 .XXX ±.005 ANGLES ±1/2°	ENG.	U-PROGRAMMED CONTROLLER DEVICE REGISTERS 3 AND 4
	APPRD	
	USED ON	SCALE — SIZE C DWG. NO. LBD1829 REV. A
	NEXT ASSY	SHEET 25 OF

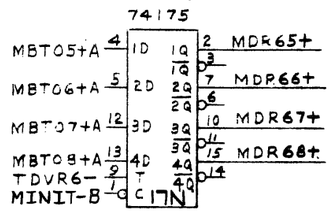
PDF-003



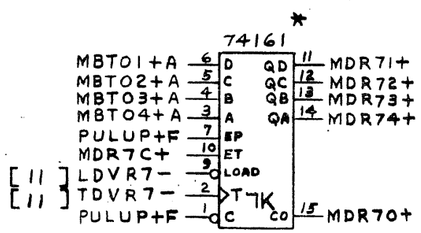


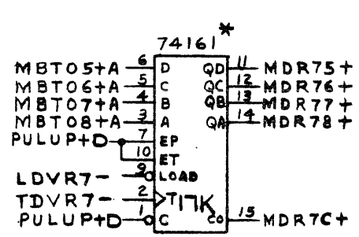


DEVICE REGISTER 5



DEVICE REGISTER 6



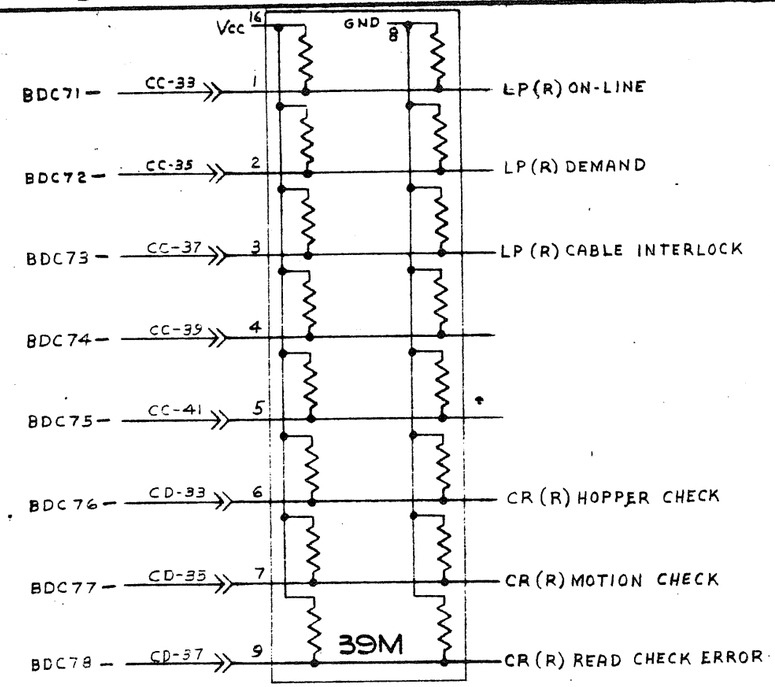


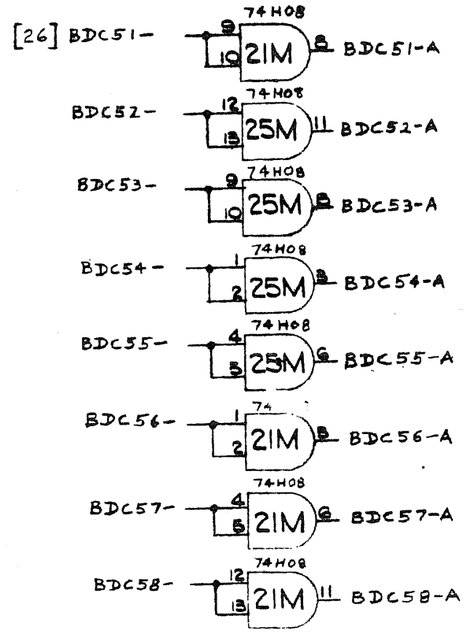
DEVICE REGISTER 7

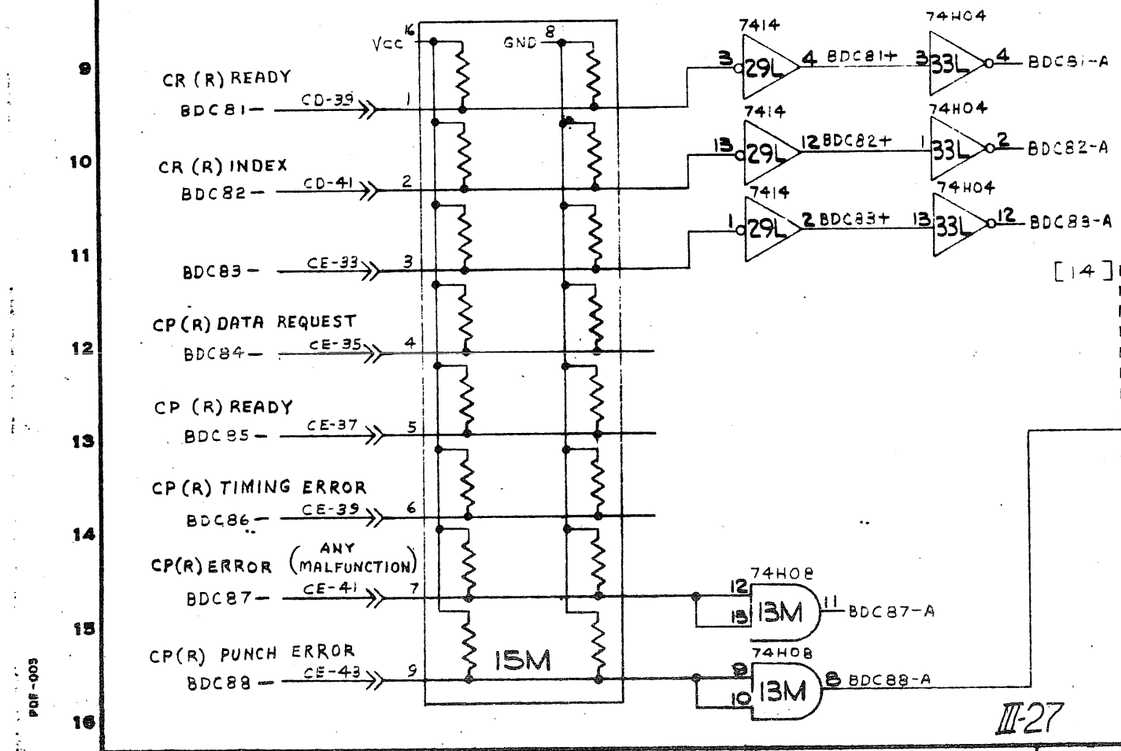
* NEED S IN HIGH SPEED MPC

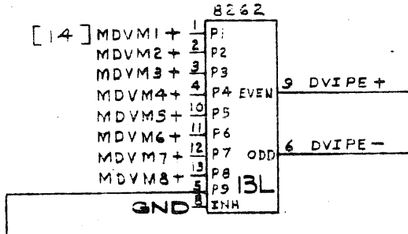
MATERIAL	DWN M.P.B. 3/23/73	PRIME COMPUTER, INC. NATICK, MASS.
	CHK	
UNLESS OTHERWISE SPECIFIED —REMOVE ALL BURRS AND SHARP EDGES: —DIMENSIONS ARE IN INCHES —TOLERANCES XX ±.02 XXX ±.005 ANGLES ±1/2°	ENG.	OPTION MPC-2 (EV) U-PROGRAMMED CONTROLLER DEVICE REGISTERS 5 THRU 7
	APPRD	
USED ON	SCALE	SIZE C DWG. NO. LBD1829 REV. A
NEXT ASSY	SHEET 26 OF	

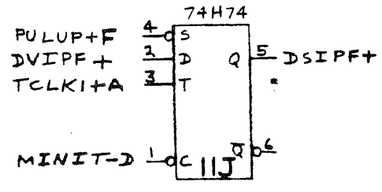
PDF-003

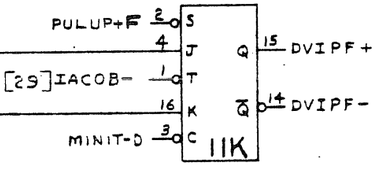




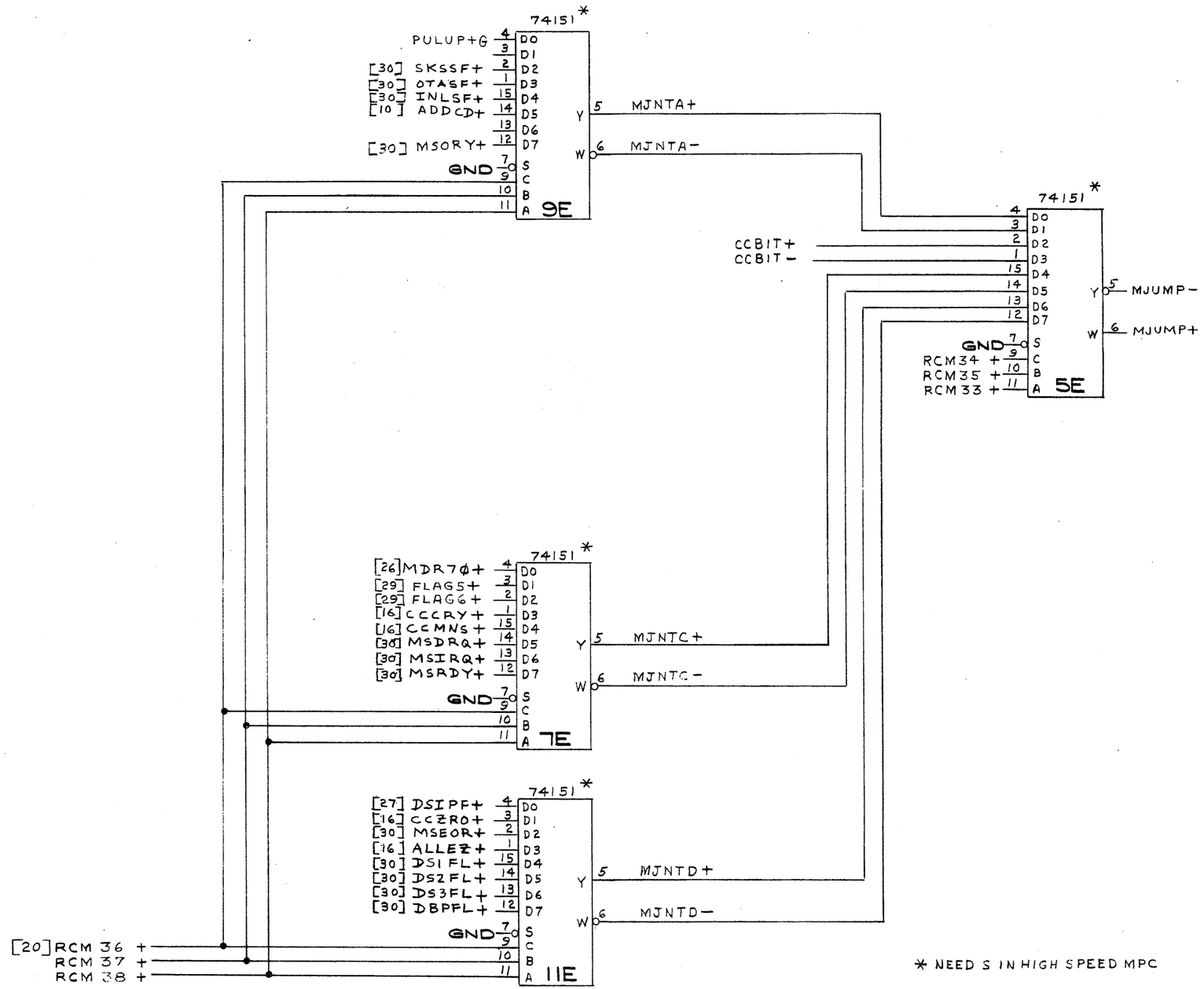








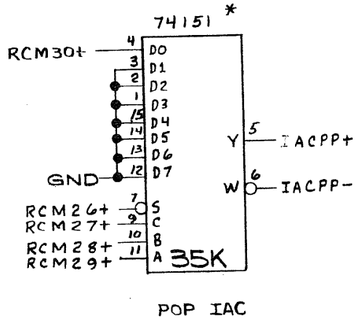
MATERIAL	DWN H. H. B 3/22/73	PRIME COMPUTER, INC. NATICK, MASS.	
	CHK	URC MICRO-PROGRAMMED CONTROLLER DEVICE BUS INPUT & PARITY	
UNLESS OTHERWISE SPECIFIED –REMOVE ALL BURRS AND SHARP EDGES; –DIMENSIONS ARE IN INCHES –TOLERANCES .XX .XXX ANGLES ± .03 ± .005 ± 1/2°	ENG.		
	APPRD 10-9-74		
	USED ON	SCALE	SIZE C DWG. NO. LBD1829 REV. A
	NEXT ASSY	SHEET 27 OF	



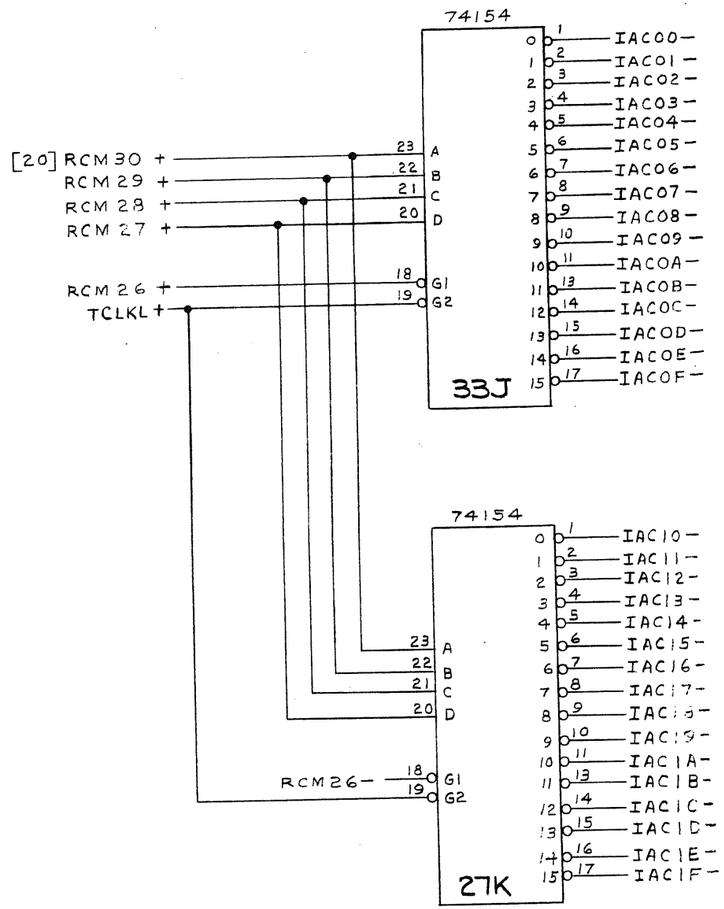
* NEED S IN HIGH SPEED MPC

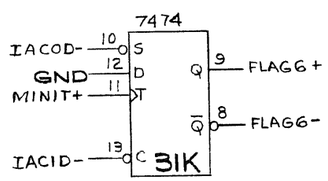
MATERIAL	DWN M. M. L. 4/2/73	PRIME COMPUTER, INC. NATICK, MASS.	
	CHK		
UNLESS OTHERWISE SPECIFIED —REMOVE ALL BURRS AND SHARP EDGES; —DIMENSIONS ARE IN INCHES —TOLERANCES .XX ±.02 .XXX ±.005 ANGLES ±1/2°	ENG.	OPTION MPC-2 (EV) U-PROGRAMMED CONTROLLER JUMP NET	
	APPRD		
	USED ON	SCALE	SIZE C
	NEXT ASSY	SHEET 28 OF	DWG. NO. LBD1829 REV. A

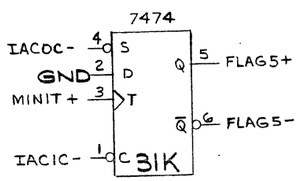
PDF-003

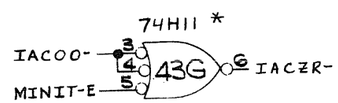


POP IAC







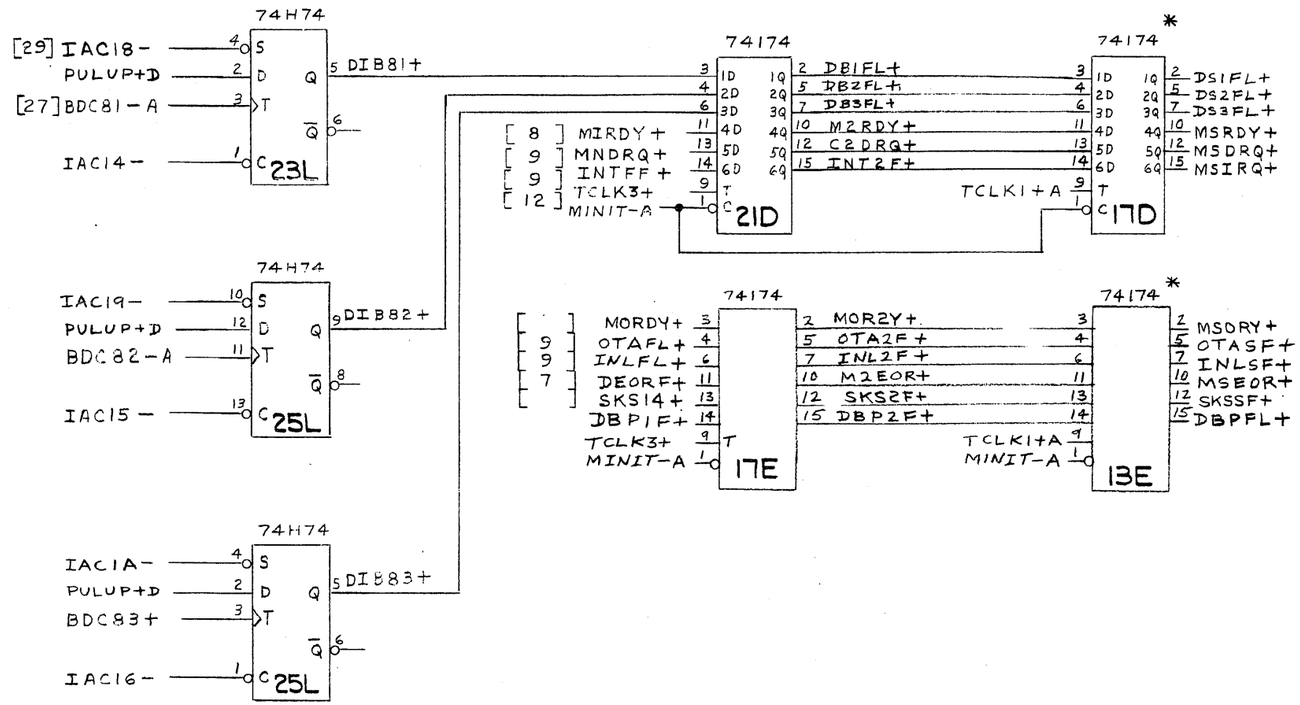


* NEED S IN HIGH SPEED MPC

MATERIAL	DWN M.W. B 4/2/73	PRIME COMPUTER, INC. NATICK, MASS.
	CHK	
UNLESS OTHERWISE SPECIFIED —REMOVE ALL BURRS AND SHARP EDGES: —DIMENSIONS ARE IN INCHES —TOLERANCES .XX ±.02 .XXX ±.005 ANGLES ±1/2°	ENG.	OPTION MPC-2 (EV) U-PROGRAMMED CONTROLLER INDEP ACTION CODES
	APPRD	
	USED ON	SCALE: SIZE C DWG. NO. LBD1829 REV. M
	NEXT ASSY	SHEET 29 OF

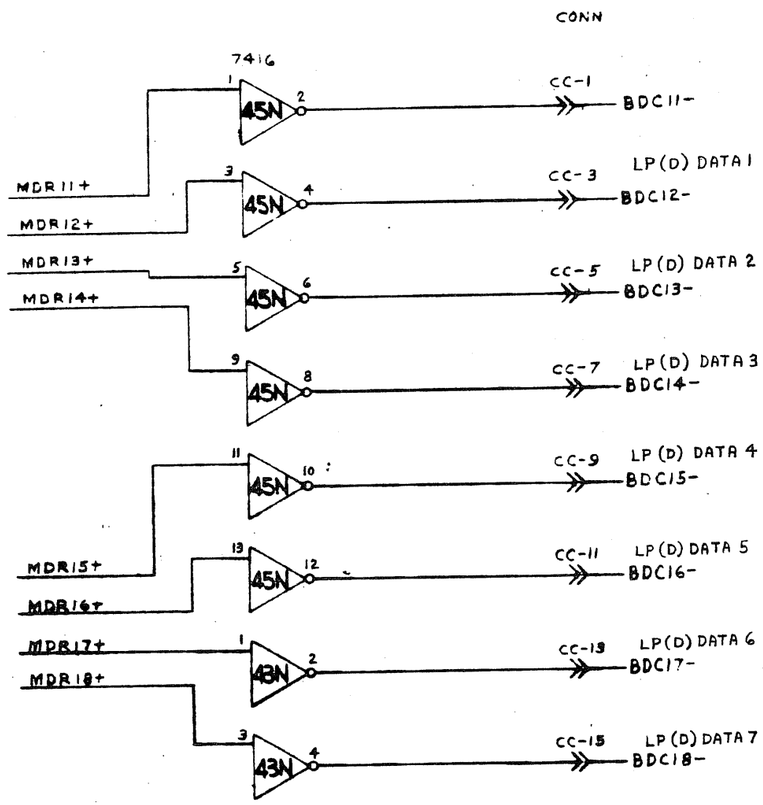
PDF-003

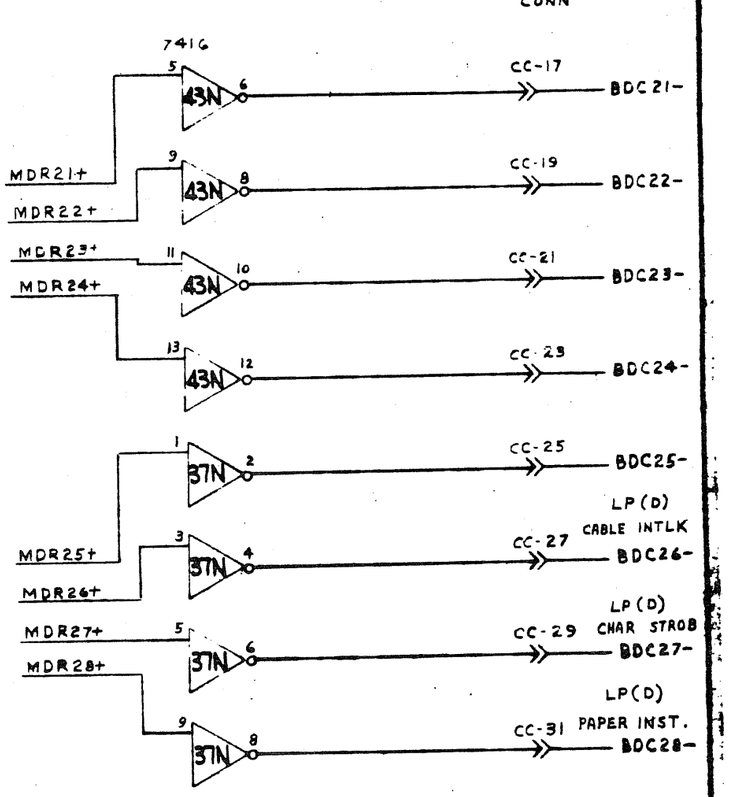
Synchronizer



MATERIAL	DWN Dr. Dr. B 4/5/73	PRIME COMPUTER, INC. NATICK, MASS.
	CHK	OPTION MPC-2 (EV)
UNLESS OTHERWISE SPECIFIED —REMOVE ALL BURRS AND SHARP EDGES; —DIMENSIONS ARE IN INCHES —TOLERANCES XX ±.02 .XXX ±.005 ANGLES ±1/2°	ENG.	MICRO-PROGRAMMED CONTROLLER DEVICE RELATED LOGIC
	APPRD	
	USED ON	SCALE / SIZE C / DWG. NO. LBD1829 / REV. M
	NEXT ASSY	SHEET 30 OF

PDF-003





MATERIAL

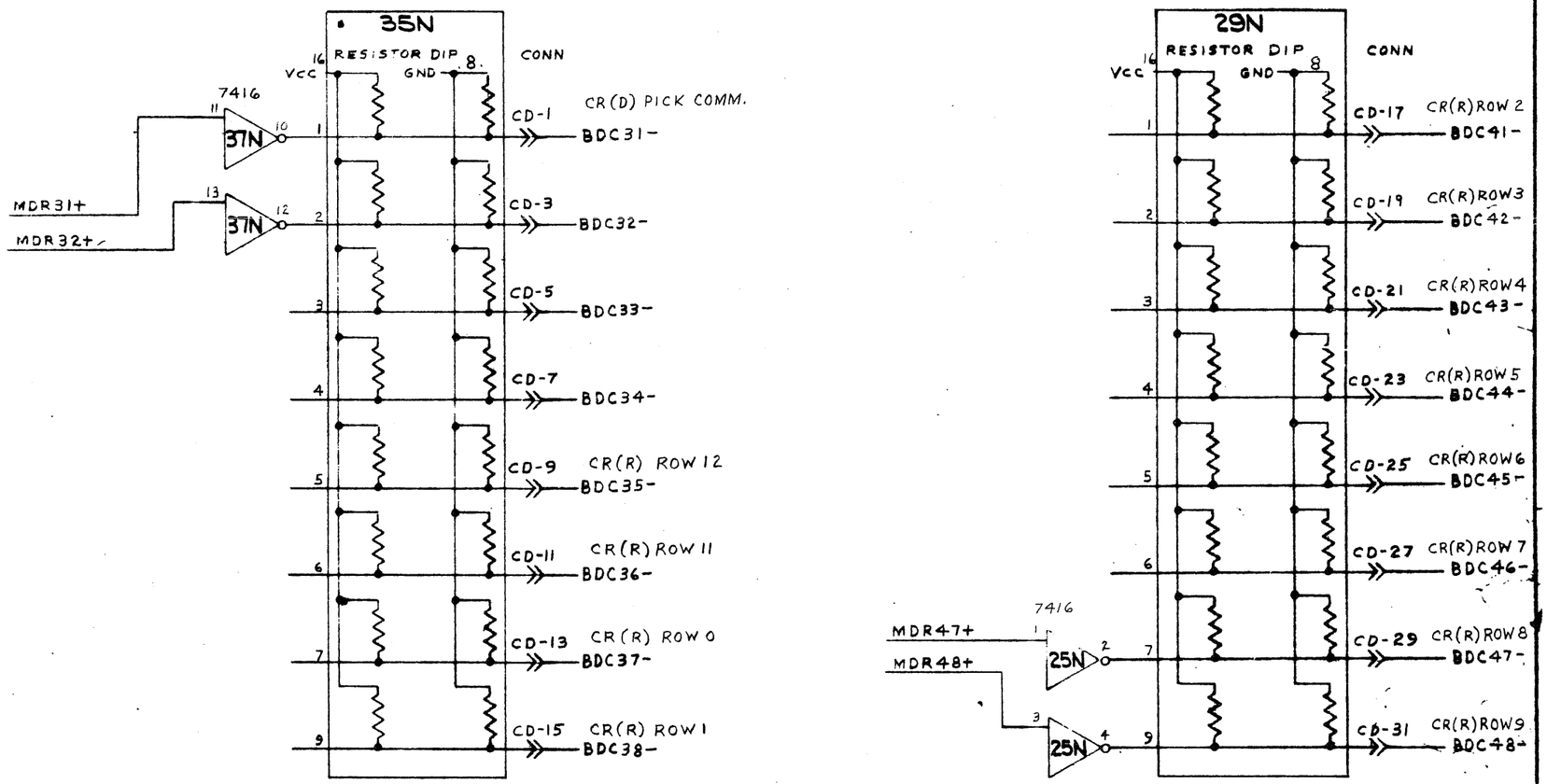
UNLESS OTHERWISE SPECIFIED
— REMOVE ALL BURRS AND SHARP EDGES.
— DIMENSIONS ARE IN INCHES
— TOLERANCES

DRWN 7/11/73
CHK
ENG.
APPVD 10-9-74

PRIME COMPUTER, INC.
NATICK, MASS.

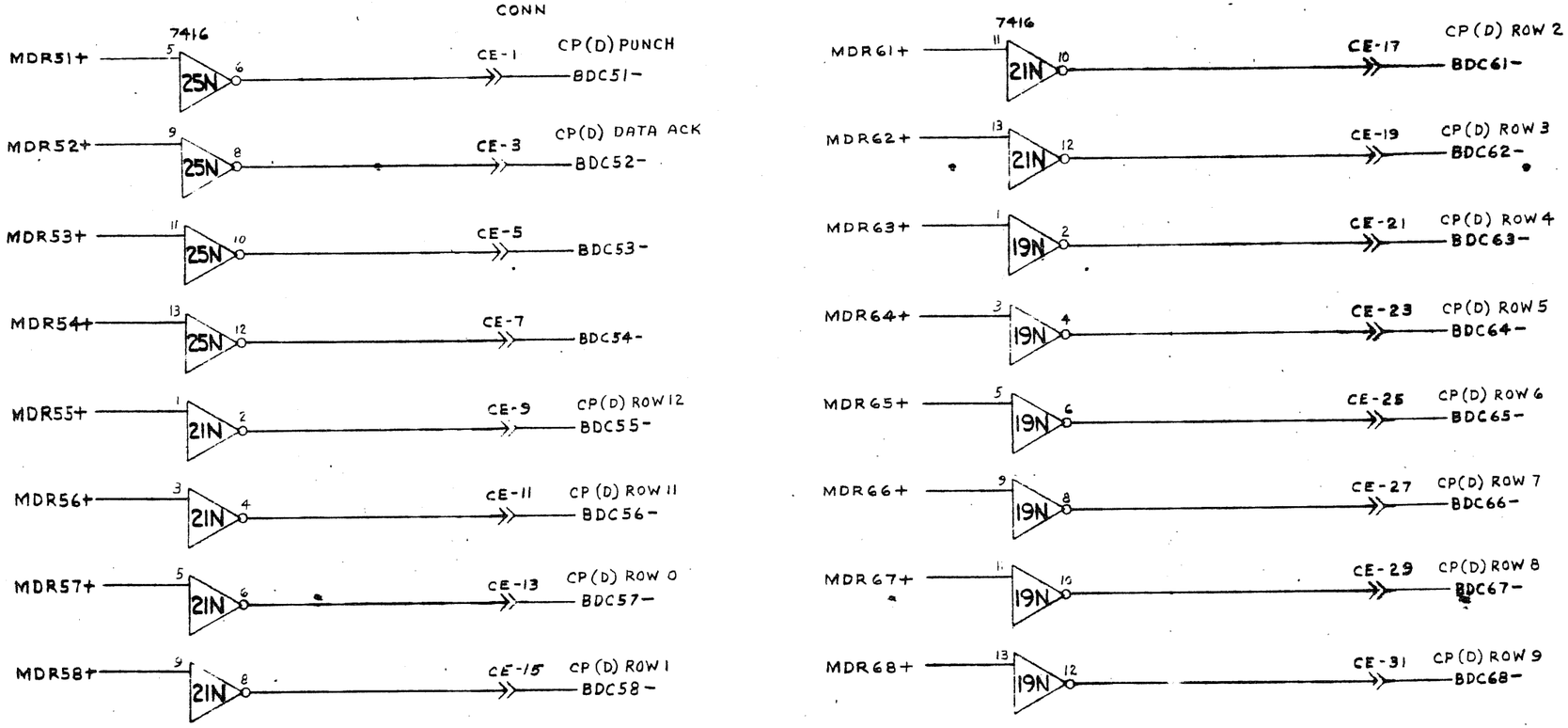
URC
MICROPROGRAMMED CONTROLLER
DEV REG OUTPUT DRIVERS

SIZE D



MATERIAL	DWN 3/23/73	PRIME COMPUTER, INC. NATICK, MASS.	
UNLESS OTHERWISE SPECIFIED —REMOVE ALL BURRS AND SHARP EDGES: —DIMENSIONS ARE IN INCHES —TOLERANCES .XX ± .02 .XXX ± .005 ANGLES ± 1/2°	CHK	URC U-PROGRAMMED CONTROLLER DEV REG OUTPUT DRIVERS	
	ENG.		
	APPRD 10-9-74		
	USED ON	SCALE	SIZE C DWG. NO. LBD1829 REV.
	NEXT ASSY	SHEET 32 OF	

PSF-003



MATERIAL	DWN J.Y.B 7/12/73	PRIME COMPUTER, INC. NATICK, MASS.
	CHK	
UNLESS OTHERWISE SPECIFIED —REMOVE ALL BURRS AND SHARP EDGES. —DIMENSIONS ARE IN INCHES —TOLERANCES .XX ±.02 .XXX ±.005 ANGLES ±.5°	ENG.	URC MU-PROGRAMMED CONTROLLER DEV REG OUTPUT DRIVER
	APPVD 10-9-74	
	USED ON	SCALE SHEET 33 OF
	NEXT ASSY	SIZE C DWG. NO. LBD1829 REV A

MPC-2 DIP ALLOCATION

SHT 34 | DWG NO. LBD 1029 | REV

74H04 MAG TAPE CPI

LTR	DATE	REVISION	DR.	CK.

B2 A2 (SOLDER SIDE)

NAME		CONN PIN
VCC3		CA-1
VCC3		CA-2
SHIELD(GND)		CA-3
BPCPCPH+		CA-4
BPCPDEN+		CA-5
GND		CA-6
BPCPDND-		CA-7
BPCPDNA-		CA-8
BPCPDNB-		CA-9
BPCPDNC-		CA-10
BPCPDND-		CA-11
BPCPDNE-		CA-12
BPCPDNF-		CA-13
BPCPDNG-		CA-14
BPCPDNH-		CA-15
BPCRUN+		CA-16
BPCIRQN-		CA-17
BPCIPNA-		CA-18
BPCIPNB-		CA-19
BPCIPNC-		CA-20
BPCIPND-		CA-21
SHIELD(GND)		CA-22
BPCICPN+		CA-23
GND		CA-24
BPCRNE+		CA-25
SHIELD(GND)		CA-26
		CA-27
		CA-28
		CA-29
BPCIOK+		CA-30
		CA-31
		CA-32
		CA-33
		CA-34
		CA-35
		CA-36
		CA-37
BPCIOV3-		CA-38
		CA-39
		CA-40
		CA-41
GND		CA-42
BPCVBB1+		CA-43
		CA-44
		CA-45
		CA-46
BPCVBB1+		CA-47
SHIELD(GND)		CA-48
VCC2		CA-49
VCC2		CA-50

NAME		CONN PIN
BMCADL+		CA-51
		CA-52
		CA-53
		CA-54
		CA-55
BMAPEL+		CA-56
BMAPER+		CA-57
BMDPEL-		CA-58
BMDPER-		CA-59
GND		CA-60
BMDO1+		CA-61
BMDO2+		CA-62
BMDO3+		CA-63
BMDO4+		CA-64
BMDO5+		CA-65
BMDO6+		CA-66
BMDO7+		CA-67
BMDO8+		CA-68
BMDO9+		CA-69
BMDI0+		CA-70
BMDI1+		CA-71
BMDI2+		CA-72
BMDI3+		CA-73
BMDI4+		CA-74
BMDI5+		CA-75
BMDI6+		CA-76
BMDLP-		CA-77
BMDRP-		CA-78
SHIELD		CA-79
GND		CA-80
BMCSELB-		CA-81
BMCSELV-		CA-82
BMCDIR0+		CA-83
BMCWRB-		CA-84
BMCWLB-		CA-85
BMCRFSH-		CA-86
BMCEYS1-		CA-87
BMCWSTRB-		CA-88
BMCEXS2-		CA-89
BMCFCLK+		CA-90
BMCSS01-		CA-91
BPCIRQ-		CA-92
BMCSS02-		CA-93
BPCDRQ-		CA-94
BMCSS03-		CA-95
BMCPACH-		CA-96
BMCCNBL-		CA-97
GND		CA-98
VCC2		CA-99
VCC2		CA-100

NAME		CONN PIN
VCC1		CB-1
VCC1		CB-2
GND		CB-3
ISOSPARE		CB-4
BMA99-		CB-5
BMA00-		CB-6
BMA01-		CB-7
BMA02-		CB-8
BMA03-		CB-9
BMA04-		CB-10
BMA05-		CB-11
BMA06-		CB-12
BMA07-		CB-13
BMA08-		CB-14
BMA09-		CB-15
BMA10-		CB-16
BMA11-		CB-17
BMA12-		CB-18
BMA13-		CB-19
BMA14-		CB-20
BMA15-		CB-21
BMA16-		CB-22
BMALP-		C 23
BMARP-		CB-24
HPWRFL-		CB-25
GND		CB-26
VCORE1		CB-27
VCORE1		CB-28
BPAD1+		CB-29
GND		CB-30
BPAD2+		CB-31
BPAD3+		CB-32
BPAO4+		CB-33
BPAO5+		CB-34
BPAO6+		CB-35
BPAO7+		CB-36
BPAO8+		CB-37
BPAO9+		CB-38
BPAI0+		CB-39
BPAI1+		CB-40
HSYSCLR-		CB-41
GND		CB-42
VCORE2		CB-43
VCORE2		CB-44
BPDPEL-		CB-45
GND		CB-46
BPDPER-		CB-47
BPAPEL-		CB-48
BPA99+		CB-49
BPA00+		CB-50

NAME		CONN PIN
BPAI2+		CB-51
BPAI3+		CB-52
BPAI4+		CB-53
BPAI5+		CB-54
BPAI6+		CB-55
BPALP+		CB-56
GND		CB-57
BPARP+		CB-58
BPCPIO+		CB-59
BPAPER-		CB-60
BPCGOCY+		CB-61
BPCREDY-		CB-62
BPDLP+		CB-63
		CB-64
BPDO1+		CB-65
BPDRP+		CB-66
BPDO2+		CB-67
BPDO3+		CB-68
BPDO4+		CB-69
BPDO5+		CB-70
BPDO6+		CB-71
BPDO7+		CB-72
BPDO8+		CB-73
BPDO9+		CB-74
BPDI0+		CB-75
BPDI1+		CB-76
V12-		CB-77
V12-		CB-78
BPDI2+		CB-79
GND		CB-80
BPDI3+		CB-81
BPDI4+		CB-82
BPDI5+		CB-83
BPDI6+		CB-84
BPCMOD0+		CB-85
BPCMOD1+		CB-86
BPCMOD2+		CB-87
BPCMOD3+		CB-88
BPCINMD+		CB-89
BPCSCLK+		CB-90
V12+		CB-91
V12+		CB-92
HRUN-		CB-93
HRUN-		CB-94
BPCSTRB+		CB-95
GND		CB-96
VSS		CB-97
VSS		CB-98
VBB		CB-99
VBB		CB-100

F2 E2 D2 C2 (SOLDER SIDE)

NAME		CONN PIN
BDC11-		CC-1
GND		CC-2
BDC12-		CC-3
GND		CC-4
BDC13-		CC-5
GND		CC-6
BDC14-		CC-7
GND		CC-8
BDC15-		CC-9
GND		CC-10
BDC16-		CC-11
GND		CC-12
BDC17-		CC-
GND		CC-14
BDC18-		CC-15
GND		CC-16
BDC21-		CC-17
GND		CC-18
BDC22-		CC-19
GND		CC-20
BDC23-		CC-21
GND		CC-22
BDC24-		CC-23
GND		CC-24
BDC25-		CC-25
GND		CC-26
BDC26-		CC-27
GND		CC-28
BDC27-		CC-29
GND		CC-30
BDC28-		CC-31
GND		CC-32
BDC71-		CC-33
GND		CC-34
BDC72-		CC-35
GND		CC-36
BDC73-		CC-37
GND		CC-38
BDC74-		CC-39
GND		CC-40
BDC75-		CC-41
GND		CC-42
		CC-43
		CC-44

NAME		CONN PIN
BDC31-		CD-1
GND		CD-2
BDC32-		CD-3
GND		CD-4
BDC33-		CD-5
GND		CD-6
BDC34-		CD-7
GND		CD-8
BDC35-		CD-9
GND		CD-10
BDC36-		CD-11
GND		CD-12
BDC37-		CD-13
GND		CD-14
BDC38-		CD-15
GND		CD-16
BDC41-		CD-17
GND		CD-18
BDC42-		CD-19
GND		CD-20
BDC43-		CD-21
GND		CD-22
BDC44-		CD-23
GND		CD-24
BDC45-		CD-25
GND		CD-26
BDC46-		CD-27
GND		CD-28
BDC47-		CD-29
GND		CD-30
BDC48-		CD-31
GND		CD-32
BDC76-		CD-33
GND		CD-34
BDC77-		CD-35
GND		CD-36
BDC78-		CD-37
GND		CD-38
BDC81-		CD-39
GND		CD-40
BDC82-		CD-41
GND		CD-42
		CD-43
		CD-44

NAME		CONN PIN
BDC51-		CE-1
GND		CE-2
BDC52-		CE-3
GND		CE-4
BDC53-		CE-5
GND		CE-6
BDC54-		CE-7
GND		CE-8
BDC55-		CE-9
GND		CE-10
BDC56-		CE-11
GND		CE-12
BDC57-		CE-13
GND		CE-14
BDC58-		CE-15
GND		CE-16
BDC61-		CE-17
GND		CE-18
BDC62-		CE-19
GND		CE-20
BDC63-		CE-21
GND		CE-22
BDC64-		CE-23
GND		CE-24
BDC65-		CE-25
GND		CE-26
BDC66-		CE-27
GND		CE-28
BDC67-		CE-29
GND		CE-30
BDC68-		CE-31
GND		CE-32
BDC83-		CE-33
GND		CE-34
BDC84-		CE-35
GND		CE-36
BDC85-		CE-37
GND		CE-38
BDC86-		CE-39
GND		CE-40
BDC87-		CE-41
GND		CE-42
BDC88-		CE-43
GND		CE-44

NAME		CONN PIN
5V+		CF-1
GND		CF-2
GND		CF-3
GND		CF-4
—		CF-5
—		CF-6
—		CF-7
GND		CF-8
—		CF-9
—		CF-10
—		CF-11
—		CF-12
—		CF-13
—		CF-14
—		CF-15
—		CF-16
—		CF-17
SWSTR+		CF-18
—		CF-19
—		CF-20
—		CF-21
—		CF-22
—		CF-23
—		CF-24
—		CF-25
SWSCG+		CF-26
TCLIA-		CF-27
GND		CF-28
SWSNG+		CF-29
MINIT-E		CF-30
MCY16+		CF-31
MCY05+		CF-32
MCY15+		CF-33
MCY06+		CF-34
MCY14+		CF-35
MCY07+		CF-36
MCY13+		CF-37
MCY08+		CF-38
MCY12+		CF-39
MCY09+		CF-40
MCY11+		CF-41
MCY10+		CF-42
GND		CF-43
5V+		CF-44

135

MATERIAL	DWN M. Bojan 8/8/74	PRIME COMPUTER, INC. NATICK, MASS.	
UNLESS OTHERWISE SPECIFIED —REMOVE ALL BURRS AND SHARP EDGES. —DIMENSIONS ARE IN INCHES —TOLERANCES .XX ±.02 .XXX ±.005 ANGLES ±1/2°	CHK	CONNECTOR SIGNAL(E.V.) NAME LIST, MPC2	
	ENG.		
	APPRD		
	UBED ON	SCALE	SIZE C DWG. NO. LBD1829 REV. A
	NEXT ASSY	SHEET 35 OF	

CABLE, CARD PUNCH MPC CONN E

SIGNAL NAME			PRIME		CARD PUNCH	
	PRIME	LBD	SIG	GND	SIG	GND
PUNCH COMM	BDC51-	33	1	2	D2	C3
DATA ACK.	BDC52-	33	3	4	D4	C5
			5	6		
			7	8		
ROW 12	BDC55-	33	9	10	E3	F3
ROW 11	BDC56-	33	11	12	B4	A5
ROW 0	BDC57-	33	13	14	B6	A7
ROW 1	BDC58-	33	15	16	B8	A9
ROW 2	BDC61-	33	17	18	B10	A11
ROW 3	BDC62-	33	19	20	B12	A13
ROW 4	BDC63-	33	21	22	B14	A15
ROW 5	BDC64-	33	23	24	D14	C15
ROW 6	BDC65-	33	25	26	D12	C13
ROW 7	BDC66-	33	27	28	D10	C11
ROW 8	BDC67-	33	29	30	D8	C9
ROW 9	BDC68-	33	31	32	D6	C7
			33	34		
DATA REQ.	BDC84-	27	35	36	F2	E3
READY	BDC85-	27	37	38	U1	T2
TIMING ERROR	BDC86-	27	39	40	S1	H2
ERROR	BDC87-	27	41	42	U9	T10
PUNCH ERROR	BDC88-	27	43	44	S3	H4

CABLE, LINE PRINTER MPC CONN C

SIGNAL NAME			PRIME		LINE PRINTER	
LINE PRINTER	PRIME	LBD	SIG	GND	SIG	GND
			1	2		
DATA1	BDC12-	31	3	4	B	D
DATA2	BDC13-	31	5	6	F	H
DATA3	BDC14-	31	7	8	L	N
DATA4	BDC15-	31	9	10	R	T
DATA5	BDC16-	31	11	12	V	X
DATA6	BDC17-	31	13	14	Z	b
DATA7	BDC18-	31	15	16	f	K
			17	18		
			19	20		
			21	22		
			23	24		
			25	26		
CBL INTLK	BDC26-	31	27	28	Y	
CHAR STROBE	BDC27-	31	29	30	j	m
PAPER INST.	BDC28-	31	31	32		
ON LINE	BDC71-	27	33	34	Y	RA
DEMAND	BDC72-	27	35	36	E	C
CBL INTLK	BDC73-	27	37	38	X	
			39	40		
			41	42		
			43	44		

CABLE, CARD READER MPC CONN D

SIGNAL NAME			PRIME		CARD READER	
	PRIME	LBD	SIG	GND	SIG	GND
PICK COMM	BDC31-	32A	1	2	LL	SS
			3	4		
			5	6		
			7	8		
ROW 12	BDC35-	32	9	10	A	E
ROW 11	BDC36-	32	11	12	B	F
ROW 0	BDC37-	32	13	14	C	H
ROW 1	BDC38-	32	15	16	D	J
ROW 2	BDC41-	32	17	18	K	P
ROW 3	BDC42-	32	19	20	L	R
ROW 4	BDC43-	32	21	22	M	S
ROW 5	BDC44-	32	23	24	N	T
ROW 6	BDC45-	32	25	26	U	W
ROW 7	BDC46-	32	27	28	V	X
ROW 9	BDC47-	32	29	30	Y	CC
ROW 9	BDC48-	32	31	32	Z	DD
HOPPER CHECK	BDC76-	27	33	34	JJ	PP
MOTION CHECK	BDC77-	27	35	36	KK	RR
READ CHK ERROR	BDC78-	27	37	38	HH	NN
READY	BDC81-	27	39	40	BB	FF
INDEX	BDC82-	27	41	42	AA	EE
			43	44		

W. Bryan 8/30/74
Hom 9-3-74
Hom 10-9-74

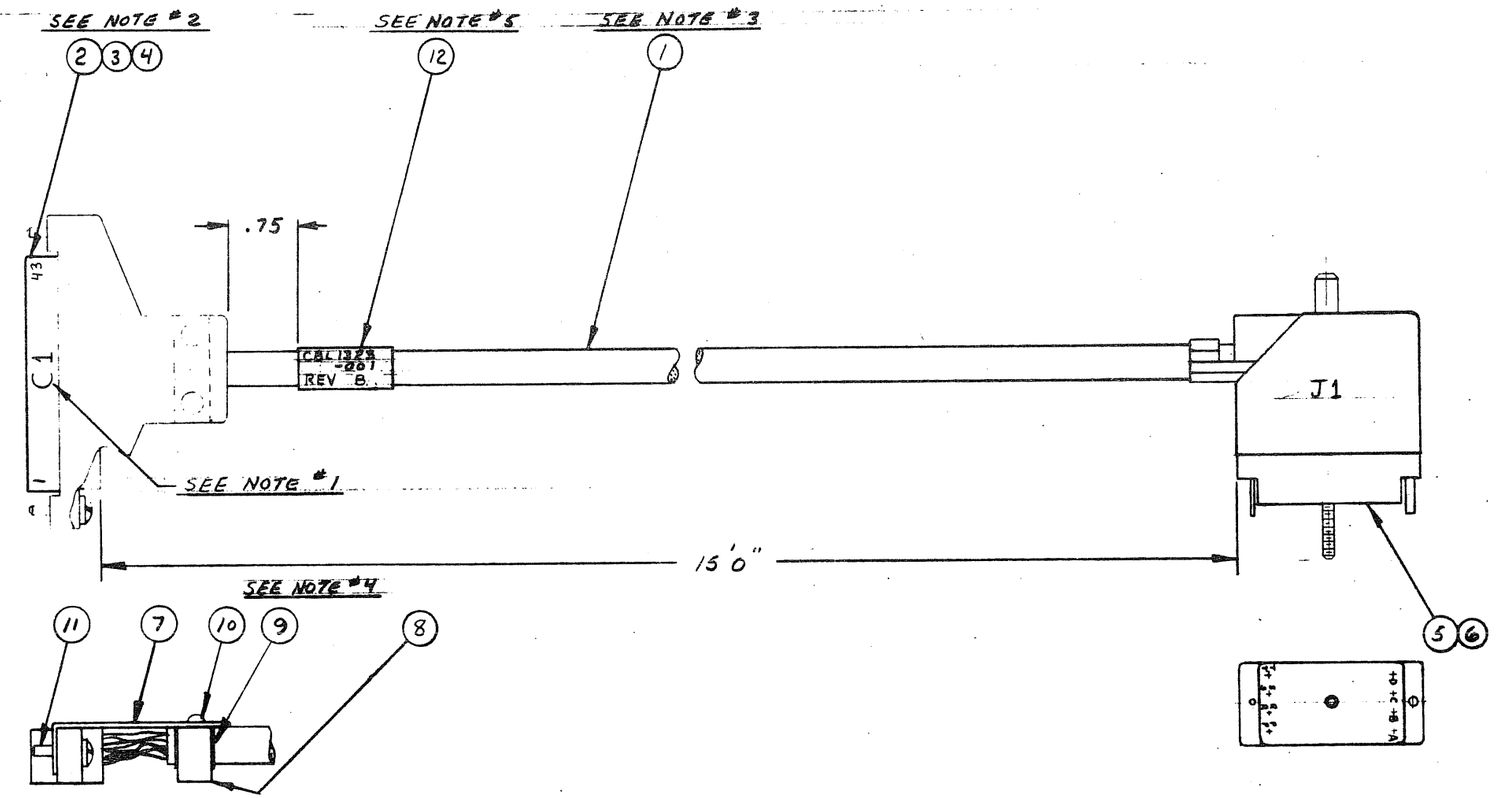
PRIME COMPUTER, INC.
NATICK, MASS.

LINE PRINTER, CARD READER
CARD PUNCH
CROSS REF CHART

CTL-D1829

LTR	DATE	REVISION	DR.	CK.
A	3/29/74	RELEASED TO PRODUCTION	JCM	
B	1/8/74	PER ECO 1521	JCM	JCM

WIRE LIST			
FROM	TO	COLOR	REMARKS
C1-1	J1-LL	BLK	TWISTED PAIR
C1-2	J1-SS	WHT	
C1-9	J1-A	BLK GRN	
C1-10	J1-E	BLK ORN	
C1-11	J1-B	BLK YEL	
C1-12	J1-F	BLK BLU	
C1-13	J1-C	WHT BLK	
C1-14	J1-H	WHT RED	
C1-15	J1-D	WHT GRN	
C1-16	J1-J	WHT ORN	
C1-17	J1-K	WHT BLU	
C1-18	J1-P	WHT YEL	
C1-19	J1-L	WHT BRN	
C1-20	J1-R	WHT GRY	
C1-21	J1-M	YEL BLK	
C1-22	J1-S	YEL RED	
C1-23	J1-N	YEL GRN	
C1-24	J1-T	YEL BLU	
C1-25	J1-U	YEL BRN	
C1-26	J1-W	YEL GRY	
C1-27	J1-V	ORN BLK	
C1-28	J1-X	ORN RED	
C1-29	J1-Y	ORN GRN	
C1-30	J1-CC	ORN BLU	
C1-31	J1-Z	ORN BRN	
C1-32	J1-DD	ORN GRY	
C1-33	J1-JJ	GRN BLK	
C1-34	J1-PP	GRN RED	
C1-35	J1-KK	GRN WHT	
C1-36	J1-RR	GRN BLU	
C1-37	J1-HH	GRN BRN	
C1-38	J1-NN	GRN YEL	
C1-39	J1-BB	GRN GRY	
C1-40	J1-FF	GRY BLK	
C1-41	J1-AA	GRY RED	TWISTED PAIR
C1-42	J1-EE	GRY WHT	



NOTES:

1. STAMP MARKINGS C1 & J1 .19 HIGH IN WHITE INK. POSITION APPROX AS SHOWN.
2. INSERT ITEM #4 (KEY) BETWEEN SLOTS 85/86 & 27/28 OF ITEM #2.
3. UNUSED WIRES ARE TO BE TERMINATED INSIDE CABLE. (CUT OFF AT BOTH ENDS)
4. WRAP ITEM #3 (TAPE) AROUND ITEM #1 AS REQUIRED & LOCATED AS SHOWN.
5. TYPE PART NO. & REVISION ON ITEM #12 IN BLACK INK AS SHOWN.
6. FOR SIGNAL NAMES SEE 2880585 (SHEET 350
7. SEE DWG 1NS1210 FOR CABLE CODING LOCATION.

III-37

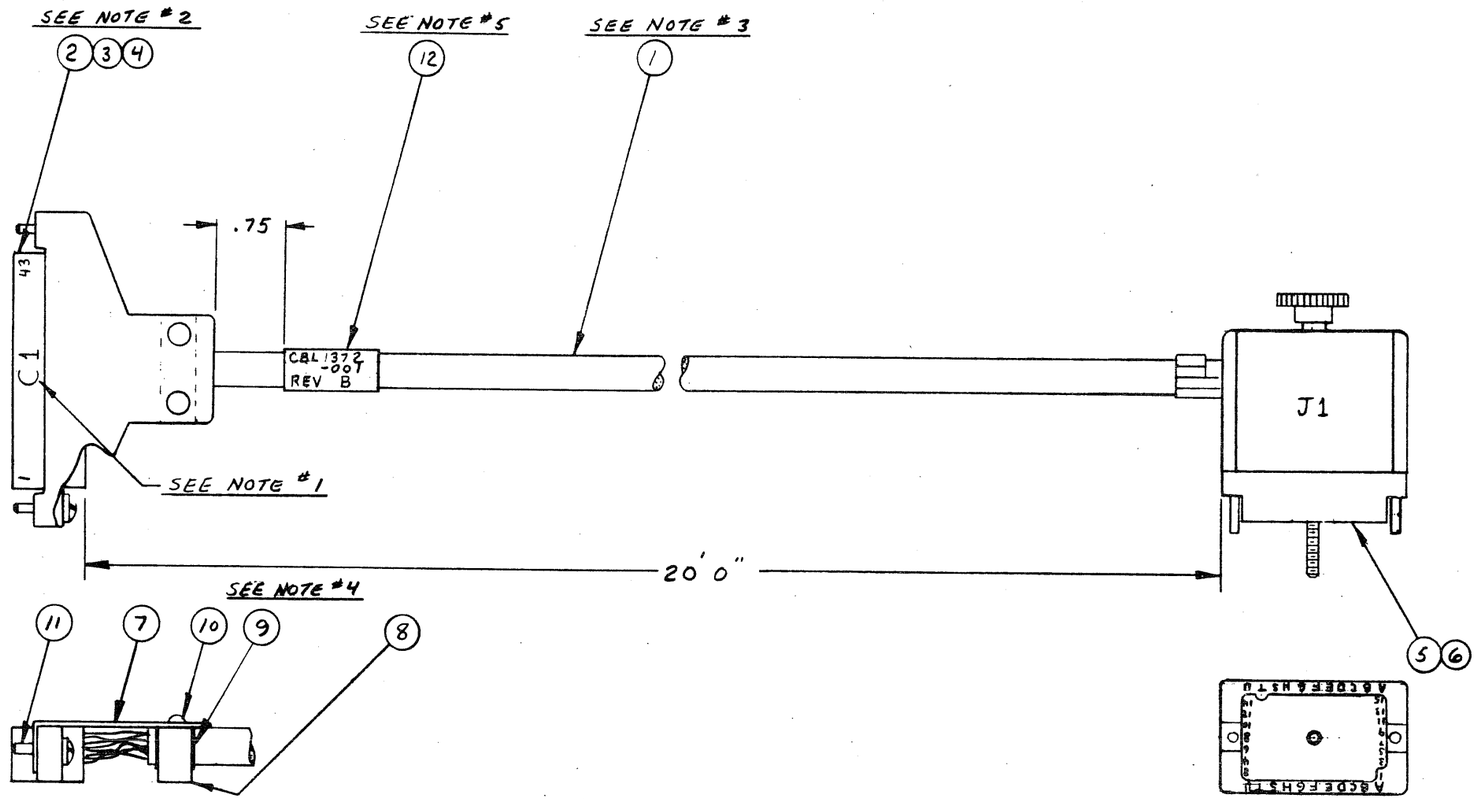
MATERIAL: SEE BOM	DWN: J.E.E. 10/23/73	PRIME COMPUTER, INC. NATICK, MASS.
UNLESS OTHERWISE SPECIFIED —REMOVE ALL BURRS AND SHARP EDGES; —DIMENSIONS ARE IN INCHES —TOLERANCES .XX ±.02 .XXX ±.005 ANGLES ±1/2°	CHK: BH 1/9/75	CABLE, PUNCH CARD READER TO URC (DOCUMATION M200) CONN. D
	ENG.	
	APPRD: 1-14-75	
	USED ON: CBL1463	SCALE: 1:1
	NEXT ASSY: 3141-001	SHEET 1 OF 1
		SIZE C DWG. NO. CBL1323-001 B

CBL1323-001 B

LTR	DATE	REVISION	DR.	CK.
A	3/24/74	RELEASED TO PRODUCTION	[illegible]	
B	1/8/75	PER ECO 1521	[illegible]	JGM

WIRE LIST

FROM	TO	COLOR	REMARKS
C1-1	J1-D2	BLK	TWISTED PAIR
C1-2	J1-C3	WHT	
C1-3	J1-D4	YEL	
C1-4	J1-C5	ORN	
C1-9	J1-B2	BLK GRN	
C1-10	J1-A3	BLK ORN	
C1-11	J1-B4	BLK YEL	
C1-12	J1-A5	BLK BLU	
C1-13	J1-B6	WHT BLK	
C1-14	J1-A7	WHT RED	
C1-15	J1-B8	WHT GRN	
C1-16	J1-A9	WHT ORN	
C1-17	J1-B10	WHT BLU	
C1-18	J1-A11	WHT YEL	
C1-19	J1-B12	WHT BRN	
C1-20	J1-A13	WHT GRY	
C1-21	J1-B14	YEL BLK	
C1-22	J1-A15	YEL RED	
C1-23	J1-D14	YEL GRN	
C1-24	J1-C15	YEL BLU	
C1-25	J1-D12	YEL BRN	
C1-26	J1-C13	YEL GRY	
C1-27	J1-D10	ORN BLK	
C1-28	J1-C11	ORN RED	
C1-29	J1-D8	ORN GRN	
C1-30	J1-C9	ORN BLU	
C1-31	J1-D6	ORN BRN	
C1-32	J1-C7	ORN GRY	
C1-35	J1-F2	GRN WHT	
C1-36	J1-E3	GRN BLU	
C1-37	J1-U1	GRN BRN	
C1-38	J1-T2	GRN YEL	
C1-39	J1-S1	GRN GRY	
C1-40	J1-H2	GRY BLK	
C1-41	J1-U9	GRY RED	TWISTED PAIR
C1-42	J1-T10	GRY WHT	
C1-43	J1-S3	GRY YEL	TWISTED PAIR
C1-44	J1-H4	GRY ORN	



NOTES:

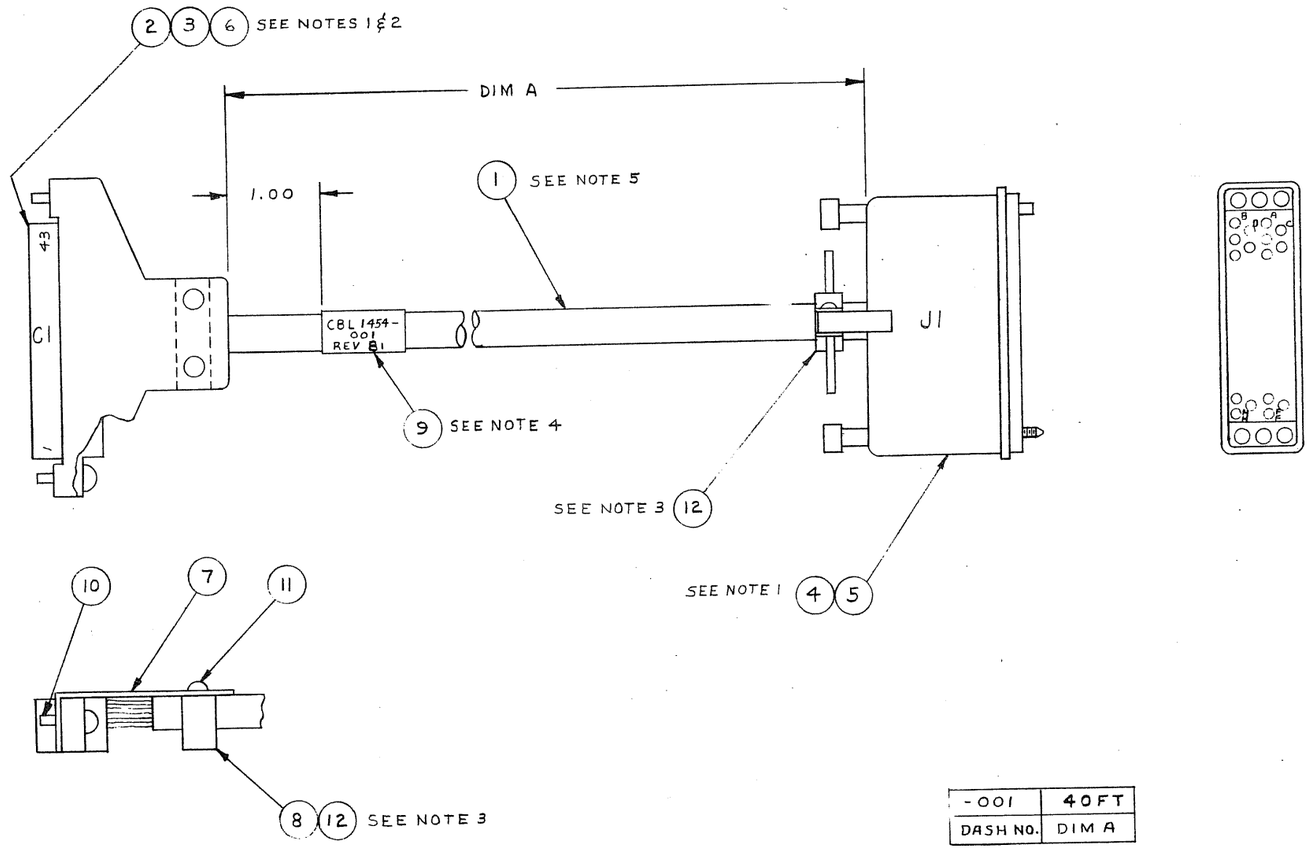
1. STAMP MARKINGS C1 & J1 .19 HIGH IN WHITE INK. POSITION APPROX AS SHOWN.
2. INSERT ITEM #4 (KEY) BETWEEN SLOTS #25/26 & #27/28 OF ITEM #2.
3. UNUSED WIRES ARE TO BE TERMINATED INSIDE CABLE. (CUT OFF AT BOTH ENDS)
4. WRAP ITEM #3 (TAPE) AROUND ITEM #1 AS REQUIRED & LOCATED AS SHOWN.
5. TYPE PART NO. & REVISION ON ITEM #12 IN BLACK INK AS SHOWN.
6. FOR SIGNAL NAMES SEE LBD0583 (SHEET 35)
7. SEE DWG INS1210 FOR CABLE CODING LOCATION.

MATERIAL: SEE BOM	DWN: J.C.L. 10/23/73	PRIME COMPUTER, INC. NATICK, MASS.
UNLESS OTHERWISE SPECIFIED —REMOVE ALL BURRS AND SHARP EDGES: —DIMENSIONS ARE IN INCHES —TOLERANCES .XX ±.02 .XXX ±.005 ANGLES ±1/2°	CHK: JBN 1/9/75	CABLE, CARD PUNCH TO URC (DOC. P-100)
	ENG.	
	APPRD: [signature] 11-14-75	CONN. D
	USED ON CBL 1462 NEXT ASSY 3181-XXX	SCALE NONE SHEET 1 OF 1 SIZE C DWG. NO. CBL1372-001 REV B

CBL 1372-001 B

LTR	DATE	REVISION	DR.	CK.
A	9/12/74	RELEASED	WB	JPM
B	1/9/75	PER ECN 1522	WB	DJF
B1	12/3/75	PER ECN 1682	WB	JPM

WIRE LIST				COMMENT
FROM	TO	BASE COLOR	STRIPE COLOR	
C1-03	J1-n	YEL	—	TP
C1-04	J1-k	ORN	—	
C1-05	J1-Z	GRN	—	
C1-06	J1-b	GRY	—	
C1-07	J1-V	BLK	WHT	
C1-08	J1-X	BLK	RED	
C1-09	J1-R	BLK	GRN	
C1-10	J1-T	BLK	ORN	
C1-11	J1-L	BLK	YEL	
C1-12	J1-N	BLK	BLU	
C1-13	J1-F	WHT	BLK	
C1-14	J1-J	WHT	RED	
C1-15	J1-B	WHT	GRN	
C1-16	J1-D	WHT	ORN	
C1-29	J1-J	ORN	GRN	
C1-30	J1-m.	ORN	BLU	
C1-31	J1-P LOWER CASE	ORN	BRN	
C1-32	J1-S LOWER CASE	ORN	GRY	
C1-33	J1-y	GRN	BLK	
C1-34	J1-AA	GRN	RED	
C1-35	J1-E	GRN	WHT	
C1-36	J1-C	GRN	BLU	
C1-27	J1-x	ORN	BLK	TP
C1-37	J1-v	GRN	BRN	



-001	40FT
DASH NO.	DIM A

NOTES:-

1. STAMP MARKINGS C1 & J1 .19 HIGH IN WHITE INK. LOCATE APPROX AS SHOWN.
2. INSERT KEY ITEM 6, BETWEEN SLOTS 25/26 & 27/28 OF ITEM 2.
3. INSTALL ITEM 12 IN CUTOUT OF ITEM 8 AND IN STRAIN RELIEF OF ITEM 4 TO INSURE TIGHT FIT OF CABLE.
4. TYPE PART NO. & REVISION IN BLACK ON ITEM 9.
5. ALL UNUSED WIRES ARE TO BE TERMINATED INSIDE CABLE. CUT OFF AT BOTH ENDS.

3170 (MPC-2)	LPR1814	2470
3152 (MPC-2)	LPR1294	2410
3152 (MPC-2)	LPR1293	2230
CONTROLLER	LINE PRINTER	DATA PRODUCTS MODEL NO.
PRIME PART NO.		

MATERIAL: SEE BOM

DWN St. Boyan 4/9/74
CHK 4/12/74
ENG. Dover Gardner 6-12-74
APPRD
USED ON
NEXT ASSY 3161-001

UNLESS OTHERWISE SPECIFIED
—REMOVE ALL BURRS AND SHARP EDGES:
—DIMENSIONS ARE IN INCHES
—TOLERANCES
XX ±.02 XXX ±.005 ANGLES ±1/2°

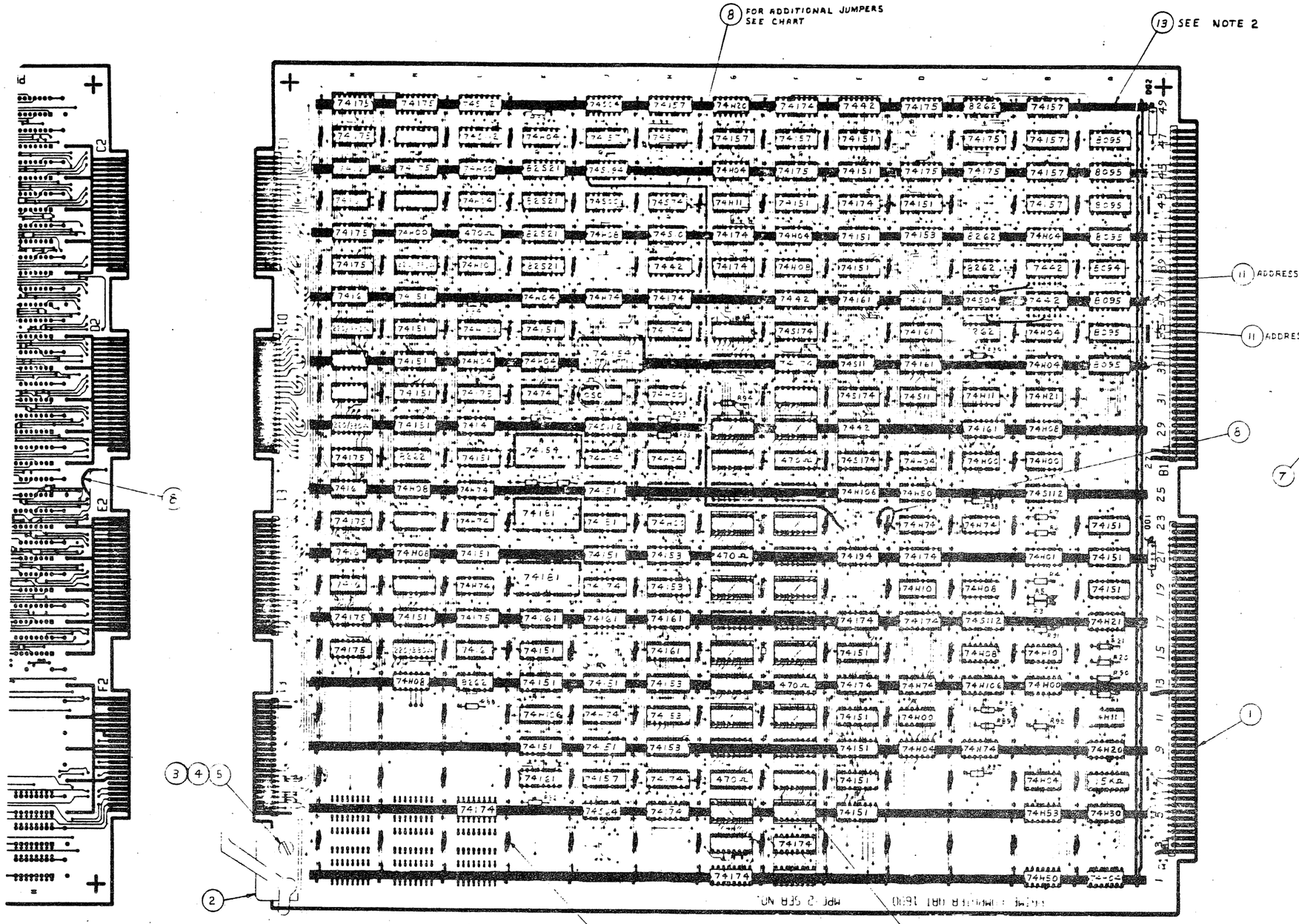
PRIME COMPUTER, INC.
NATICK, MASS.

CABLE, LINE PRINTER (DATA PRODUCTS) TO CONTROLLER

SCALE NONE | SHEET 1 OF 1 | SIZE C | DWG. NO. CBL1454-XXX | REV. B1

DWG. NO. CBL1454-XXX REV. B1

LTR	DATE	REVISION	DR.	CK.
A	9/11/74	RELEASED	WE	JCH
B	11/17/75	ADDED ETCH CUT PER ECN 1533	PB	JCH



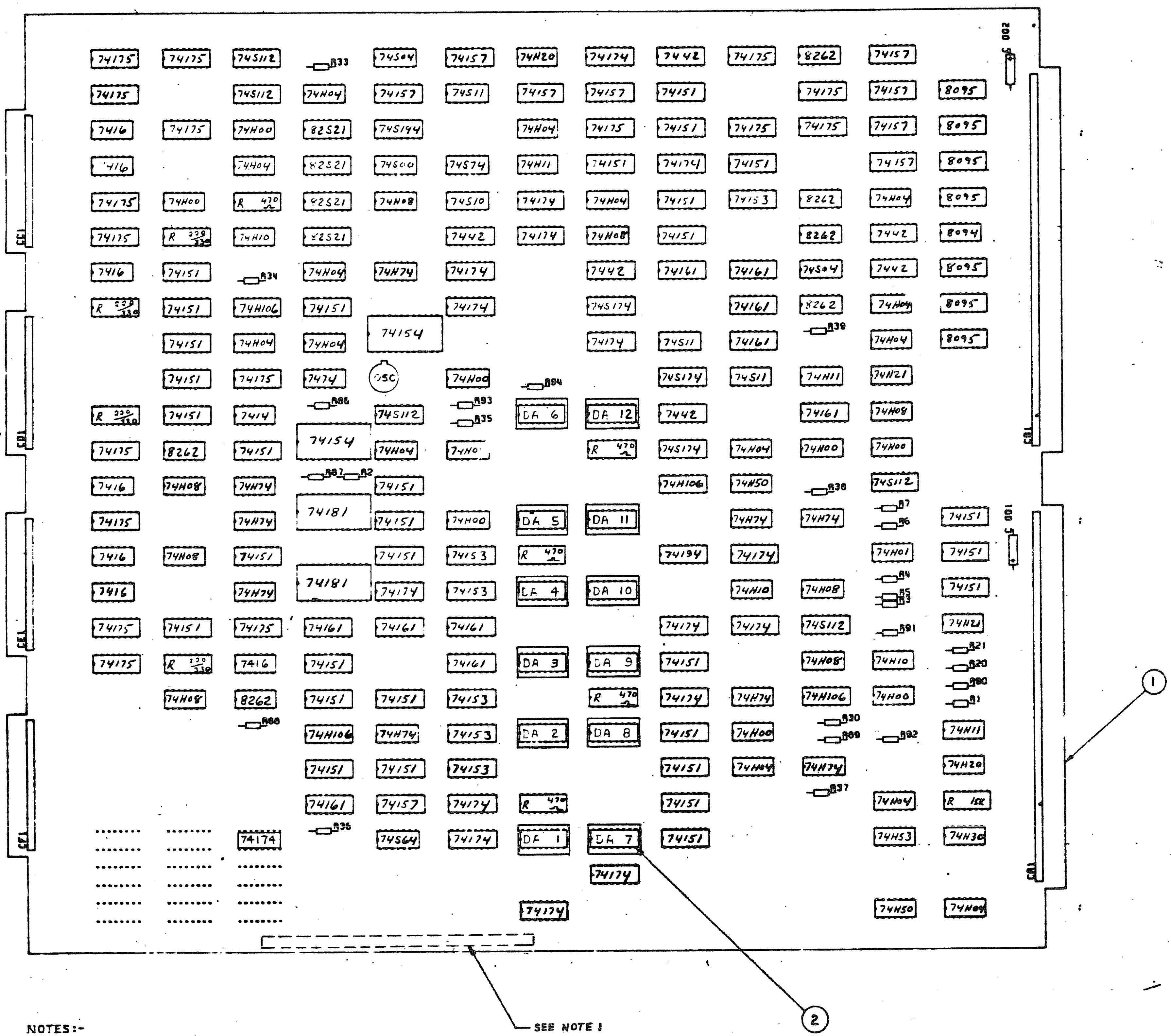
JUMPER CHART				
FROM	TO	SIGNAL NAME	AWG	COLOR
25E-03	23C-13	KCNDR-	30	YEL
23C-01	13M-06	KCNDR-	30	
21E-01	25E-06	MINIT-C	30	
13M-05	37K-10	MINIT-C	30	
13M-04	27K-16	IACIE-	30	
23C-07	23D-12	GND	30	
21E-09	23D-04	PLUP+H	30	
21E-10	23D-02	PLUP+H	30	
23D-03	27D-01	STAST-	30	
23D-01	33J-08	IACC7-	30	
05L-01	13M-05	MINIT-C	30	
05L-09	17E-09	TCLK3+	30	
05L-12	43B-01	SMRDW+	30	
05L-13	43G-08	MRDWS+	30	
05L-14	49G-06	IKROWS-	30	
05L-15	47G-01	SMROW-	30	
43L-01	43L-11	MINIT+	30	
43L-10	49M-01	MINIT-B	30	
39N-01	27K-01	MINIT-B	30	YEL
27E-01	29C-07	PULUP+H	30	YEL

FROM	TO	TO	SIGNAL NAME	AWG	COLOR
19K-12	23K-12	27K-12	GND	24	BLK

NOTES:-
1. X DENOTES ITEM 6 TO BE INSTALLED AT THESE DIP SITES.
2. CUT TABS ON ITEM 13 AT THE FOLLOWING LOCATIONS, 21K, 25K, 29K, 33J (8 TABS) CUT CLOSE TO INSULATION.

DRN H Bryan 9/11/74	PRIME COMPUTER INC. NATICK, MASS.	
CHK Ian Hume 10/9/74	URC CONTROLLER SUB ASSY	
ENGR D. Gardner 10/10/74		
APPD W.M. 10-9-74		
USED ON NEXT ASSY 3155-002	SCALE NONE SHEET 1 OF 1	DWG NO. D MEC2042-XXX REV. B

LTR.	DATE	REVISION	DR.	CK.
A	10/15/74	RELEASED	KG	[illegible]
B	1/17/75	ETCH CUT PER ECN 1533	[illegible]	[illegible]



NOTES:-
1. MARK SERIAL NO. & MODEL NO. WITH APPROPRIATE REV IN WHITE INK. LOCATE APPROX AS SHOWN.

DRAWN: H. Bogan 9/12/74	PRIME COMPUTER INC. NATICK, MASS.
CHK: [illegible] 10/10/74	CONTROLLER ASSY
ENG: D. Gauchenier 10/10/74	CARD READER (300 CPM) EV
APPRD: [illegible] 10-9-74	
NEXT ASSY 3141-001	SCALE 1/1 SHEET 1 OF 1 SIZE D DWG. NO. 3151-002 REV. B

PRIME COMPUTER INC.
NATICK MASS.

DWN. [illegible] 9/7/74
CHK. J.C.L. 9/9/74
ENG. [illegible] 9/11/74
APPRD. [illegible] 9-16-74
STANDARD COST _____ DATE _____

TITLE: CONTROLLER AND CARD READER (300 CPM) 117V 60HZ

BOM 3141 -XXX REV. B
NHA: SHT. 1 OF 1

REV.	ECN	CK	REV.	ECN	CK
A	REL	JPL			
B	1477	JPL			

ITEM	SIZE	PART NUMBER	QUANTITY -001	-002	-003	-004	-005	-006	-007	-008	DESCRIPTION	STANDARD COST
1	D	3151-001	1	—							CONTROLLER ASSY CARD READER (WW)	
1	D	3151-002	—	1							CONTROLLER ASSY CARD READER (EV)	
2		PCR1313-001	1	1							CARD READER, 117V 60 HZ	
3	C	CBL1323-001	1	1							CABLE, CARD READER, DOCUMATION	

PDF-004A

PRIME COMPUTER INC.
NATICK MASS.

DWN. [illegible] 9/12/74
CHK. J.C.L. 10/9/74
ENG. D. Gardner 10/10/74
APPRD. [illegible] 10-9-74
STANDARD COST _____ DATE _____

TITLE: CONTROLLER ASSY CARD READER (300CPM) EV

BOM 3151 -XXX REV. B
NHA: 3141-002 SHT. 1 OF 1

REV.	ECN	CK	REV.	ECN	CK
A	REL	—			
B	1533	JPL			

ITEM	SIZE	PART NUMBER	QUANTITY -001	-002	-003	-004	-005	-006	-007	-008	DESCRIPTION	STANDARD COST
1	D	MEC2042-001	—	1							URC CONTROLLER SUB ASSY	
2	A	MEC1901-017	—	1							PROM SET, URC (PCR D03) DA	
	A	SPC 1409		REF							SPEC PRODUCT MPC 2	
	A	SPC 1824		REF							SPEC PRODUCT	
	C	LBD 1829		REF							LOGIC BLOCK DIAGRAM MPC-2 EV	

PDF-004A

PRIME COMPUTER INC.
NATICK MASS.

DWN. [illegible] 9/12/74
CHK. J.C.L. 10/9/74
ENG. [illegible] 10/10/74
APPRD. [illegible] 10-9-74
STANDARD COST _____ DATE _____

TITLE: CONTROLLER ASSY LINE PRINTER (300LPM) EV

BOM 3141-A -XXX REV. A
NHA: SHT. 1 OF 1

REV.	ECN	CK	REV.	ECN	CK
A	REL	—			

ITEM	SIZE	PART NUMBER	QUANTITY -001	-002	-003	-004	-005	-006	-007	-008	DESCRIPTION	STANDARD COST
1	D	3151-001	1	—							CONTROLLER ASSY CARD READER (WW)	
1	D	3151-002	—	1							CONTROLLER ASSY CARD READER (EV)	
2		PCR1313-002	1	1							CARD READER, 230V, 50 HZ	
3	C	CBL1323-001	1	1							CABLE, CARD READER, DOCUMATION	

PDF-004A

PRIME COMPUTER INC.
NATICK MASS.

DWN. [illegible] 9/12/74
CHK. J.C.L. 10/9/74
ENG. [illegible] 10/10/74
APPRD. [illegible] 10-9-74
STANDARD COST _____ DATE _____

TITLE: CONTROLLER ASSY LINE PRINTER (300LPM) EV

BOM 3152 -XXX REV. B
NHA: 3141-002 SHT. 1 OF 1

REV.	ECN	CK	REV.	ECN	CK
A	REL	—			
B	1533	JPL			

ITEM	SIZE	PART NUMBER	QUANTITY -001	-002	-003	-004	-005	-006	-007	-008	DESCRIPTION	STANDARD COST
1	D	MEC2042-001	—	1							URC CONTROLLER SUB ASSY	
2	A	MEC1901-017	—	1							PROM SET, URC (PCR D03) DA	
	A	SPC 1409		REF							SPEC, PRODUCT MPC 2	
	A	SPC 1824		REF							SPEC, PRODUCT	
	C	LBD 1829		REF							LOGIC BLOCK DIAGRAM MPC-2 EV	

PDF-004A

PRIME COMPUTER INC.
NATICK MASS.

DWN. WE 9/13/74
CHK. J.C.T. 19/9/74
ENG. D. Gardner 10/10/74
APPRD. Mom 10-9-74

STANDARD COST _____ DATE _____

TITLE: CONTROLLER ASSY CARD READER/PUNCH (400 CPM, 100 CPS) EV

BOM 3153 -XXX REV. B
NHA: 3181-002 SHT. 1 OF 1

REV.	ECN	CK	REV.	ECN	CK
A	REL	—			
B	1533	JCT			

ITEM	SIZE	PART NUMBER	QUANTITY -001	-002	-003	-004	-005	-006	-007	-008	DESCRIPTION	STANDARD COST
1	D	MEC 2042-001	–	1							URC CONTROLLER SUB ASSY	
2	A	MEC 1901-017	–	1							PROM SET, URC (PCRD03) DA	
	A	SPC 1409		REF							SPEC, PRODUCT MPC2	
	A	SPC 1824		REF							SPEC, PRODUCT	
	C	LBD 1829		REF							LOGIC BLOCK DIAGRAM MPC2 EV	

PDF-004A

PRIME COMPUTER INC.
NATICK MASS.

DWN. WE 9/13/74
CHK. J.C.T. 19/9/74
ENG. D. Gardner 10/10/74
APPRD. Mom 10-9-74

STANDARD COST _____ DATE _____

TITLE: CONTROLLER ASSY CARD READER/ PUNCH/ LINE PRINTER EV

BOM 3155 -XXX REV. B
NHA: 3135-002 SHT. 1 OF 1

REV.	ECN	CK	REV.	ECN	CK
A	REL	—			
B	1533	JCT			

ITEM	SIZE	PART NUMBER	QUANTITY -001	-002	-003	-004	-005	-006	-007	-008	DESCRIPTION	STANDARD COST
1	D	MEC 2042-001	–	1							URC CONTROLLER SUB ASSY	
2	A	MEC 1901-017	–	1							PROM SET, URC (PCRD03) DA	
	A	SPC 1409	REF								SPEC, PRODUCT MPC2	
	A	SPC 1824	REF								SPEC, PRODUCT	
	C	LBD 1829	REF								LOGIC BLOCK DIAGRAM MPC2 EV	

PDF-004A

PRIME COMPUTER INC.
NATICK MASS.

DWN. WE 9/13/74
CHK. J.C.T. 9/13/74
ENG. D. Gardner 10/10/74
APPRD. Mom 10-9-74

STANDARD COST _____ DATE _____

TITLE: CONTROLLER AND LINE PRINTER (300LPM) 117V 60HZ EV

BOM 3154 -XXX REV. B
NHA: 3191-002 SHT. 1 OF 1

REV.	ECN	CK	REV.	ECN	CK
A	REL	—			
B	1533	JCT			

ITEM	SIZE	PART NUMBER	QUANTITY -001	-002	-003	-004	-005	-006	-007	-008	DESCRIPTION	STANDARD COST
1	D	MEC 2042-001	–	1							URC CONTROLLER SUB ASSY	
2	A	MEC 1901-017	–	1							PROM SET, URC (PCRD03) DA	
	A	SPC 1409	REF								SPEC, PRODUCT MPC2	
	A	SPC 1824	REF								SPEC, PRODUCT	
	C	LBD 1829	REF								LOGIC BLOCK DIAGRAM MPC2 EV	

PDF-004A

PRIME COMPUTER INC.
NATICK MASS.

DWN. 7-19-74 KOPPENAL
CHK. J.C.T. 9/13/74
ENG. [illegible]
APPRD. [illegible] 9-14-74

STANDARD COST _____ DATE _____

TITLE: CONTROLLER AND LINE PRINTER (300LPM) 117V 60HZ

BOM 3161 -XXX REV. B
NHA: SHT. 1 OF 1

REV.	ECN	CK	REV.	ECN	CK
B	1488	JCT			

ITEM	SIZE	PART NUMBER	QUANTITY -001	-002	-003	-004	-005	-006	-007	-008	DESCRIPTION	STANDARD COST
1	D	3152-001	1	–							CONTROLLER ASSY LINE PRINTER (WW)	
1	D	3152-002	–	1							CONTROLLER ASSY LINE PRINTER (EV)	
2		LPR1273-001	1	1							LINE PRINTER, 117V 60HZ	
3	C	CBL1454-001	1	1							CABLE, LINE PRINTER DATA PRODUCTS	

N-02A

PRIME COMPUTER INC.
NATICK MASS.

DWN. [illegible] 9/13/74
CHK. [illegible] 10/4/74
ENG. [illegible] 10/8/74
APPRD. [illegible] 10-9-74
STANDARD COST _____ DATE _____

TITLE: CONTROLLER AND LINE PRINTER (300 LPM (230V 50HZ

BOM 3161-A -XXX **REV.** A
NHA: SHT. 1 OF 1

REV.	ECN	CK	REV.	ECN	CK
A	REL	–			

ITEM	SIZE	PART NUMBER	QUANTITY -001	-002	-003	-004	-005	-006	-007	-008	DESCRIPTION	STANDARD COST
1	D	3152-001	1	–							CONTROLLER, ASSY LINE PRINTER (WW)	
1	D	3152-002	–	1							CONTROLLER ASSY LINE PRINTER (EV)	
2		LPR1293-002	1	1							LINE PRINTER, 230V 50HZ	
3	C	CBL1454-001	1	1							CABLE, LINE PRINTER DATA PRODUCTS	

PDF-004A

PRIME COMPUTER INC.
NATICK MASS.

DWN. [illegible] 9/13/74
CHK. [illegible] 10/4/74
ENG. D. Gaviner 10/10/74
APPRD. [illegible] 10-9-74
STANDARD COST _____ DATE _____

TITLE: CONTROLLER ASSY LINE PRINTER (1200 LPM) EV

BOM 3170 -XXX **REV.** B
NHA: 3171-002 SHT. 1 OF 1

REV.	ECN	CK	REV.	ECN	CK
A	REL	–			
B	1533	[illegible]			

ITEM	SIZE	PART NUMBER	QUANTITY -001	-002	-003	-004	-005	-006	-007	-008	DESCRIPTION	STANDARD COST
1	D	MEC2042-001	–	1							URC CONTROLLER SUB ASSY	
2	A	MEC1901-017	–	1							PROM SET, URC (PCRDØ3) DA	
	A	SPC1409	REF								SPEC, PRODUCT MPC2	A
	A	SPC1824	REF								SPEC, PRODUCT	
	C	LBD1629	REF								LOGIC BLOCK DIAGRAM MPC2 EV	

PDF-004A

PRIME COMPUTER INC.
NATICK MASS.

DWN. 7-19-74 KOPPENAL
CHK. [illegible] 7/24/74
ENG. [illegible] 7/25/74
APPRD. [illegible]
STANDARD COST _____ DATE _____

TITLE: CONTROLLER AND LINE PRINTER (1200LPM) 117V 60HZ

BOM 3171-XXX **REV.** B
NHA: SHT. 1 OF 1

REV.	ECN	CK	REV.	ECN	CK
A	REL	[illegible]			
B	1533	JB			

ITEM	SIZE	PART NUMBER	QUANTITY -001	-002	-003	-004	-005	-006	-007	-008	DESCRIPTION	STANDARD COST
1	D	3170-001	1	–							CONTROLLER ASSY LINE PRINTER (WW)	
1	D	3170-002	–	1							CONTROLLER ASSY LINE PRINTER (EV)	
2		LPR1814-001	1	1							LINE PRINTER, 117V 60HZ	
3	C	CBL1454-001	1	1							CABLE, LINE PRINTER DATA PRODUCTS	

PDF-004A

PRIME COMPUTER INC.
NATICK MASS.

DWN. [illegible] 9/16/74
CHK. [illegible] 10/4/74
ENG. D. Gaviner 10/10/74
APPRD. [illegible] 10-9-74
STANDARD COST _____ DATE _____

TITLE: CONTROLLER AND LINE PRINTER (1200 LPM) 230V 50HZ

BOM 3171-A -XXX **REV.** A
NHA: SHT. 1 OF 1

REV.	ECN	CK	REV.	ECN	CK
A	REL	–			

ITEM	SIZE	PART NUMBER	QUANTITY -001	-002	-003	-004	-005	-006	-007	-008	DESCRIPTION	STANDARD COST
1	D	3170-001	1	–							CONTROLLER ASSY LINE PRINTER (WW)	
1	D	3170-002	–	1							CONTROLLER ASSY LINE PRINTER (EV)	
2		LPR1814-002	1	1							LINE PRINTER 230V 50HZ	
3	C	CBL1454-001	1	1							CABLE, LINE PRINTER DATA PRODUCTS	

PRIME COMPUTER INC.
NATICK MASS.

DWN. JKB 9/16/74
CHK. J.C.T. 10/9/74
ENG. D. Gavlick 10/10/74
APPRD. Hom 10-9-74

STANDARD COST _____ DATE _____

TITLE: CONTROLLER AND CARD READER/PUNCH (400 CPM/100-285 CPM) 117V 60HZ

BOM 3181 **-XXX** REV. A
NHA: SHT. 1 OF 1

REV.	ECN	CK	REV.	ECN	CK
A	REL	–			

ITEM	SIZE	PART NUMBER	QUANTITY -001	-002	-003	-004	-005	-006	-007	-008	DESCRIPTION	STANDARD COST
1	D	3153-001	1	–							CONTROLLER ASSY CR/PUNCH (WW)	
1	D	3153-002	–	1							CONTROLLER ASSY CR/PUNCH (EV)	
2		PCR 1797-001	1	1							CARD READER/PUNCH 117V 60HZ	
3	C	CBL 1372-001	1	1							CABLE, CARD PUNCH	
4	C	CBL 1323-001	1	1							CABLE, PUNCH CARD READER	

PDF-004A

PRIME COMPUTER INC.
NATICK MASS.

DWN. JKB 9/16/74
CHK. J.C.T. 10/9/74
ENG. D. Gavlick 10/10/74
APPRD. Hom 10-9-74

STANDARD COST _____ DATE _____

TITLE: CONTROLLER AND CARD READER LINE PRINTER 117V 60HZ

BOM 3191 **-XXX** REV. A
NHA: SHT. 1 OF 1

REV.	ECN	CK	REV.	ECN	CK
A	REL	–			

ITEM	SIZE	PART NUMBER	QUANTITY -001	-002	-003	-004	-005	-006	-007	-008	DESCRIPTION	STANDARD COST
1		3154-001	1	–							CONTROLLER ASSY LINE PRINTER (WW)	
1	D	3154-002	–	1							CONTROLLER ASSY LINE PRINTER (EV)	
2		LPR 1293-001	1	1							LINE PRINTER 117V 60HZ	
3	C	CBL 1454-001	1	1							CABLE, LINE PRINTER	
4	C	CBL 1323-001	1	1							CABLE, PUNCH CARD READER	
5		PCR 1313-001	1	1							CARD READER 117V 60HZ	

PDF-004A

PRIME COMPUTER INC.
NATICK MASS.

DWN. JKB 9/16/74
CHK. J.C.T. 10/9/74
ENG. D. Gavlick 10/10/74
APPRD. Hom 10-9-74

STANDARD COST _____ DATE _____

TITLE: CONTROLLER AND CARD READER/PUNCH (400 CPM/100-285 CPM) 230V 50 HZ

BOM 3181-A **-XXX** REV. A
NHA: SHT. 1 OF 1

REV.	ECN	CK	REV.	ECN	CK
A	REL	–			

ITEM	SIZE	PART NUMBER	QUANTITY -001	-002	-003	-004	-005	-006	-007	-008	DESCRIPTION	STANDARD COST
1	D	3153-001	1	–							CONTROLLER ASSY CR/PUNCH (WW)	
1	D	3153-002	–	1							CONTROLLER ASSY CR/PUNCH (EV)	
2		PCR 1797-002	1	1							CARD READER/PUNCH 230V 50HZ	
3	C	CBL 1372-001	1	1							CABLE, CARD PUNCH	
4	C	CBL 1323-001	1	1							CABLE, PUNCH CARD READER	

PDF-004A

PRIME COMPUTER INC.
NATICK MASS.

DWN. JKB 9/16/74
CHK. J.C.T. 10/9/74
ENG. D. Gavlick 10/10/74
APPRD. Hom 10-9-74

STANDARD COST _____ DATE _____

TITLE: CONTROLLER AND CARD READER LINE PRINTER 230V 50HZ

BOM 3191-A **-XXX** REV. A
NHA: SHT. 1 OF 1

REV.	ECN	CK	REV.	ECN	CK
A	REL	–			

ITEM	SIZE	PART NUMBER	QUANTITY -001	-002	-003	-004	-005	-006	-007	-008	DESCRIPTION	STANDARD COST
1		3154-001	1	–							CONTROLLER ASSY LINE PRINTER (WW)	
1	D	3154-002	–	1							CONTROLLER ASSY LINE PRINTER (EV)	
2		LPR 1293-002	1	1							LINE PRINTER 230V 50HZ	
3	C	CBL 1454-001	1	1							CABLE, LINE PRINTER	
4	C	CBL 1323-001	1	1							CABLE, PUNCH CARD READER	
5		PCR 1313-002	1	1							CARD READER 230V 50HZ	

PRIME COMPUTER INC.
NATICK MASS.

DWN. JKE 9/16/74
CHK. J.C.T. 10/1/74
ENG. [illegible] 10/10/74
APPRD. [illegible] 10-9-74
STANDARD COST _____ DATE _____

TITLE: CONTROLLER AND CARD READER/PUNCH LINE PRINTER 230V 50HZ

BOM 3195-A -XXX REV. A
NHA: SHT. 1 OF 1

REV.	ECN	CK	REV.	ECN	CK
A	P.E.L.	—			

ITEM	SIZE	PART NUMBER	QUANTITY -001	-002	-003	-004	-005	-006	-007	-008	DESCRIPTION	STANDARD COST
1		3155-001	1	—							CONTROLLER ASSY CR/PUNCH, LP (WW)	
1	D	3155-002	—	1							CONTROLLER ASSY CR/PUNCH, LP (EV)	
2		LPR1293-002	1	1							LINE PRINTER 230V 50HZ	
3		PCR1797-002	1	1							CARD READER PUNCH 230V 50HZ	
4	C	CBL1454-001	1	1							CABLE, LINE PRINTER	
5	C	CBL1372-001	1	1							CABLE, CARD PUNCH	
6	C	CBL1323-001	1	1							CABLE, PUNCH CARD READER	

PDF-004A

PRIME COMPUTER INC.
NATICK MASS.

DWN. JKE 9/16/74
CHK. J.C.T. 10/1/74
ENG. [illegible] 10/10/74
APPRD. [illegible] 10-9-74
STANDARD COST _____ DATE _____

TITLE: CONTROLLER AND CARD READER/PUNCH LINE PRINTER 117V 60HZ

BOM 3195 -XXX REV. A
NHA: SHT. 1 OF 1

REV.	ECN	CK	REV.	ECN	CK
A	P.E.L.	—			

ITEM	SIZE	PART NUMBER	QUANTITY -001	-002	-003	-004	-005	-006	-007	-008	DESCRIPTION	STANDARD COST
1		3155-001	1	—							CONTROLLER ASSY CR/PUNCH, LP (WW)	
1	D	3155-002	—	1							CONTROLLER ASSY CR/PUNCH, LP (EV)	
2		LPR 1293-001	1	1							LINE PRINTER 117V 60HZ	
3		PCR 1797-001	1	1							CARD READER PUNCH 117V 60HZ	
4	C	CBL 1454-001	1	1							CABLE, LINE PRINTER	
5	C	CBL 1372-001	1	1							CABLE, CARD PUNCH	
6	C	CBL 1323-001	1	1							CABLE, PUNCH CARD READER	

PDF-004A

PRIME COMPUTER INC.
NATICK MASS.

DWN. JKB 9/9/74
CHK. [illegible] 10/9/74
ENG. D. Ga. 10/9/74
APPRD. HSM 10-9-74
STANDARD COST _____ DATE _____

TITLE: URC CONTROLLER SUB ASSY EV

BOM MEC 2042-XXX REV. B1
NHA: 3155-002 SHT. 1 OF 3

REV.	ECN	CK	REV.	ECN	CK
A	SEE	–			
B	1533	JKB			

ITEM	SIZE	PART NUMBER	QUANTITY -001	-002	-003	-004	-005	-006	-007	-008	DESCRIPTION	STANDARD COST
1	D	MEC1849-001	REF								P.C. BOARD ETCH CUTS MPC2	A
2	C	MEC 0587	1								STIFFENER ASSY, P.C. BOARD	
3		MEC 0303-005	5								SCREW, BD HD, CRES #4-40X1/4LG	
4		MEC 0356	5								WASHER, FLAT FIBER No. 4	
5		MEC 0388-002	5								NUT, SELF LOCKING #4-40	
6		CON0650-001	2								SOCKET, 16 PIN (LOW PROFILE)	
7		WIR1221-003	A/R								WIRE, BUS 22 AWG	
8		WIR1365-004	A/R								WIRE, 30 AWG (YEL)	
9		WIR1365-009	A/R								WIRE, 30 AWG (WHT)	
10		WIR1365-002	A/R								WIRE, 30 AWG (RED)	
11		WIR 0633-000	A/R								WIRE, 24 AWG SINGLE CON. (BLK)	
12		CAP 0129	152								CAP, CERAMIC DISC .01MF 25V	
13	B	MEC 1406-001	13								INSULATED BUS BAR	
14												
15		RES 0250-153	1								RESISTOR DIP 15KΩ	
16		RES 0250-471	5								RESISTOR DIP 470Ω	
17		RES0667-003	4								RESISTOR DIP 220/330Ω	
18		RES 0221-550	5								RESISTOR 15K, 1/4W R3, 4, 5, 6, 7	
19		RES 0221-400	9								RESISTOR 2K, 1/4W R1, 20, 21, 30 R84-92	
20		RES 0221-302	13								RESISTOR 470Ω, 1/4W R2, R33-39 R86, 87, 88, 93, 94	
21		CAP 0552-315	3								CAPACITOR, TANT 3.3MF 15V C1, 2	
22		PCB1601-001	1								P.C. BD. MPC-2 E.V.	A

PDF-004A

PRIME COMPUTER INC.
NATICK MASS.

DWN. JKB 9/9/74
CHK.
ENG.
APPRD.
STANDARD COST _____ DATE _____

TITLE: URC CONTROLLER SUB ASSY

BOM MEC 2042-XXX REV. B2
NHA: SHT. 2 OF 3

REV.	ECN	CK	REV.	ECN	CK

ITEM	SIZE	PART NUMBER	QUANTITY -001	-002	-003	-004	-005	-006	-007	-008	DESCRIPTION	STANDARD COST
23												
24		XTL0654	1								OSCILLATOR	
25		ICD 0025	3								74H00	
26		ICD 0026	1								74H01	
27		ICD 0028	16								74H04	
28		ICD 0029	3								74H08	
29		ICD 0030	3								74H10	
30		ICD 0031	3								74H11	
31		ICD 0033	2								74H20	
32		ICD 0034	2								74H21	
33		ICD 0035	1								74H30	
34		ICD 0038	2								74H50	
35		ICD 0040	1								74H53	
36		ICD 0043	3								74H74	
37		ICD 0046	4								74H106	
38		ICD 0048	11								74151	
39		ICD 0049	6								74153	
40		ICD 0051	3								74157	
41		ICD 0052	0								74161	
42		ICD 0053	7								74174	
43		ICD 0054	7								74175	
44		ICD 0055	2								74131	

PDF-004A

PRIME COMPUTER INC.
NATICK MASS.

DWN. JKB 9/9/74
CHK.
ENG.
APPRD.
STANDARD COST _____ DATE _____

TITLE: URC CONTROLLER SUB ASSY

BOM MEC 2042-XXX REV. B2
NHA: SHT. 3 OF 3

REV.	ECN	CK	REV.	ECN	CK

ITEM	SIZE	PART NUMBER	QUANTITY -001	-002	-003	-004	-005	-006	-007	-008	DESCRIPTION	STANDARD COST
45		ICD 0057	1								74194	
46		ICD 0058	6								7442	
47		ICD 0059	1								8094	
48		ICD 0060	3								8262	
49		ICD 0070	1								74S64	
50		ICD 0071	1								74S74	
51		ICD 0072	5								74S112	
52		ICD 0076	3								74S174	
53		ICD 0078	1								74S194	
54		ICD 0085	1								74S00	
55		ICD 0086	2								74S04	
56		ICD 0088	1								74S10	
57		ICD 0089	2								74S11	
58		ICD 0106	7								7416	
59		ICD 0112	2								8095	
60		ICD 0186	2								74154	
61		ICD 0201	1								7474	
62		ICD 0664	1								7414	
63		ICD 0665	4								82S21	

PDF-004A