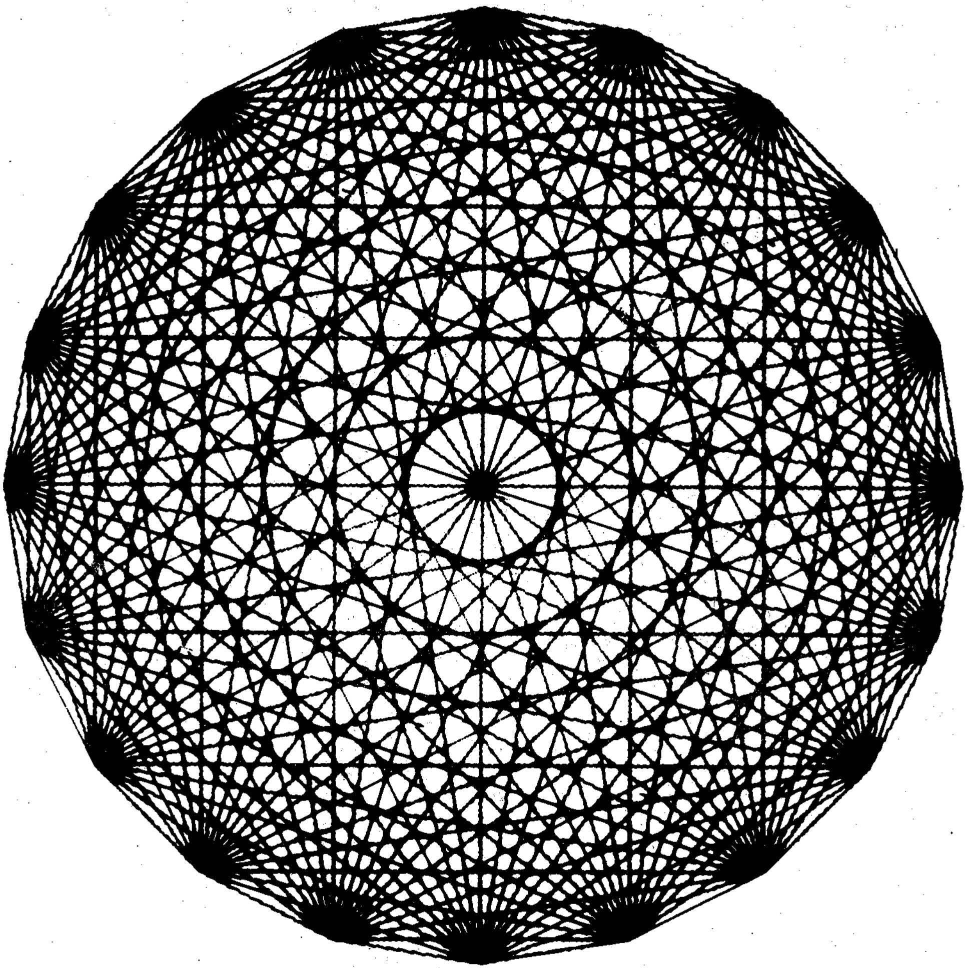
PLATO V

Programmable Plasma Touch Terminal



Copyright 1978
by Carroll Manufacturing
Champaign, IL

TABLE OF CONTENTS

Page

ACKNOWLEDGEMENTS

The development of this report was due in large part to the efforts of several people.

Jack Stifle wrote the X-50 report, which forms the major part of the beginning of this manual. In addition, Jack Stifle, Mike Hightower and the entire staff of Plato have been extremely helpful and cooperative in providing the information necessary to document and manufacture the terminal.

Special thanks are due to Roy Lipschutz, who did all of the drawings and to Kathie Wissmiller for the typing and editing of this report.

INSTALLATION AND OPERATION

PLACEMENT

The Carroll P2T2 is designed to fit on a standard 30" desk top with keyset connected and placed against the front. The terminal operating environment is for room temperature conditions ranging from 20° to 40°C and moderate humidity.

INPUT POWER REQUIREMENTS

Electrical

The terminal requires electrical input power of 115 VAC, 50-60 Hz, 5A maximum. A power cord is included with each terminal.

MECHANICAL

Height:	21½ inches
Width:	18 inches
Depth (with Keyboard):	23 inches
Weight:	90 pounds
Power:	115 V, 60 Hz, 5A 50°F to 100°F 10% to 80% relative humidity no condensation

PLASMA PANEL

Screen Area:	8.3 inches square (512 x 512 lines > 250,000 addressable points).
Luminance:	Avg. point luminance exceeds 50 fl.
Capacity:	2048 1/8 inch character. Up to 4250 inches of vectors (.033 inch grid).
Write/Erase Speed:	3000 characters/second. > 600 inches/second graphics.

Viewing Time: Until erased. (Static panel, will erase by itself after typically 30 minutes.)

Full Screen Bulk Erase: 20 μs.

Panel Assembly Includes Self-contained X-Y Decode and Driver Circuits for both serial and parallel operation.

LOGIC ELECTRONICS

Input/Output interface	(20 bits/word, 60 words/second)	8080 P based
Character generator	180 char/second	8K ROM
256 Character memory	128 ROM and 128 Alterable	8K RAM

KEYSET HOOKUP

The keyset connects to the front of the terminal through a flexible cable approximately 2½ feet long. The connector may be locked in place with the retaining springs mounted on the chassis mating half of the connector. The flexible cord may be folded under the chassis if it is desirable to place the keyset against the front of the terminal.

DATA INPUT (OUTPUT)

Data input (output) from (to) the computer or communications network is at the Serial Data Connector. This connector is located on the rear of the terminal and is identified by the marking "COMM." The connector is a DB-25S type. The mating connector required is a DB-25P. One is supplied with the terminal.

POWER TURN-ON

When all input-output connections and other conditions described above have been satisfied the terminal is ready for turn-on. The power on switch is at the lower center under the front ledge.

Within a second or two after turn-on, the plasma panel borders will light. The borders can be seen at the edge of the display area although they are partially masked by the front panel structure.

CAUTION

If terminal is turned off, do not turn on again for at least 10 seconds as this may damage high voltage circuitry which drive the borders.

A check should be made to verify the fans are running. This may be done by placing a hand at the rear of the terminal to detect air flow. If the fan is not running, it is probable that the logic power supply fuse is blown, and the terminal will not function, even though the panel borders may be lit. Note that the terminal should not be operated if the fan is not operating (or with the back removed) as it will not cool properly.

CONTROLS

The main control interface for the terminal is through the keyset which transmits data to the computer. Two front panel switches and an indicator are also provided. Their functions are described in subsequent paragraphs.

Reset Switch

The white RESET switch is a momentary type switch used for initializing the terminal and clearing the "Abort" mode. The RESET switch will also terminate any user program which may be running in the terminal.

Abort Indicator

When the terminal detects an error on input data, it enters the "Abort" mode and turns on the red ABORT indicator. The indicator remains on as long as the terminal is in the ABORT mode. The terminal remains in the abort mode until it is cleared by an appropriate input word from the computer or by actuation of the RESET switch.

Console Switch

This switch is at the lower left of the terminal. When pressed to the left, the terminal is set for normal operation, as chosen by the user. When in the right position, the console monitor routines have control, and provide an on-screen presentation of register and program status. Details for use of this feature are given subsequently.

THE PLATO® V TERMINAL

J.E. Stifle

Computer-based Education Research Laboratory

University of Illinois, Urbana, Illinois 61801

ABSTRACT

This report describes the architecture and programming of the PLATO V terminal. This terminal contains an 8080 microprocessor and is capable of being operated by programs resident in the terminal or by programs located in a host computer. The terminal contains 8k of memory for storing local programs, a 4k ROM resident program which supervises terminal operation, a 2k ROM character set and 2k of spare ROM memory.

PLATO® is the registered trademark
of the University of Illinois.

Acknowledgments

The development of this terminal was due in large part to the efforts of several people.

Mike Hightower and Ron Klass played a major role in the development of the terminal firmware and the design of the ROM programming system. Leonard Hedges performed the layout of the processor assembly and supervised the construction of the terminal prototypes.

A special mention is due Donald Lee, who wrote an 8080 assembler which greatly simplified and speeded the development of the terminal firmware.

Bruce Sherwood helped define the external IO protocol and Paul Tenczar made a significant improvement in the character plotting program.

A special thanks is due Joyce Lipschutz, who very patiently did all of the typing and editing of this report, and to Roy Lipschutz, who did all of the drawing.

TABLE OF CONTENTS

Chapter 1 - Terminal Architecture

Chapter 2 - Operating Modes

TABLE OF CONTENTS (continued)

1. TERMINAL ARCHITECTURE

1.0 General

The PLATO V terminal is actually a micro computer system containing all of the standard features of large computer systems including a processing unit, memory, and an input-output (IO) structure. Attaching a plasma panel to the IO structure and loading the appropriate program in memory will make this system behave as a PLATO terminal. As with any computer system, the contents of the memory determine the system operating characteristics. This programmable feature enables the terminal to be operated as a standard PLATO IV terminal or as a more powerful PLATO V terminal with enhanced graphics capabilities.

This terminal may be operated by programs located in the central computer, as in PLATO IV, or by local programs residing in the terminal memory. The latter may be loaded into the terminal in much the same manner as character set data is loaded, or they may be loaded from floppy disks or other storage devices attached to the terminal. The terminal is also capable of being operated in a time shared mode between local and centrally resident programs.

It is this local programming feature which dramatically expands the powerful and unique graphics capabilities of the PLATO system. Included in these graphics enhancements are:

- A character plotting rate in excess of 3000 characters per second.
- Character plotting in both horizontal and vertical modes.
- Character plotting in two directions in both modes.
- Character magnification.
- A 6000 character per second selective block erase function.

A block diagram of the terminal is shown in Figure 1.0. Operation of the terminal is under the supervision and control of an 8080 microprocessor constructed on a single LSI chip. This chip is an 8-bit parallel processor operating with a 2μs instruction time containing 7 internal working registers.

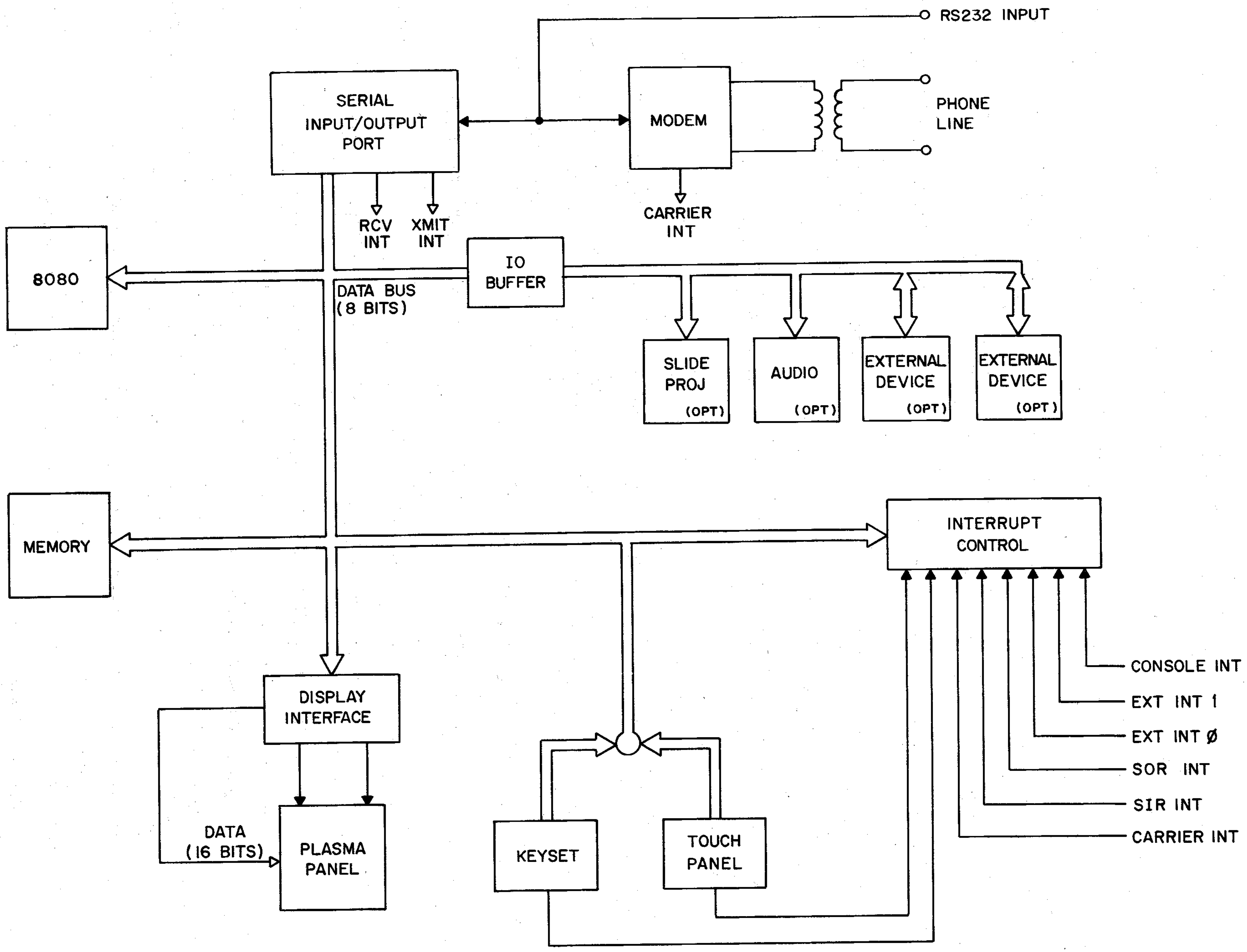


Figure 1.0. Terminal Block Diagram

An 8-bit bidirectional data bus connects the 8080 to the other functional units within the terminal and to IO devices external to the terminal. The data bus acts as a highway over which information flows between internal (and external) portions of the terminal. In addition to performing display generation functions, the 8080 must manage the flow of information on this highway. Also present but not shown is a 16-bit address bus which is used to specify memory and input-output device addresses.

This terminal operates as an interrupt driven device; i.e., all activity occurs as a result of an interrupt requesting service.

All service requests to the 8080 are made via the Interrupt Control Unit (ICU). A device requesting service originates an interrupt and presents it to the ICU. Within the ICU the interrupt requests have a wired-in priority, and the ICU will pass to the 8080 the interrupt request having the highest priority.

When an interrupt request is accepted by the 8080, normal program sequencing is halted; the present contents of the program counter (PC) are pushed into the stack in memory; and an RST (unconditional jump) instruction to location 70 is forced into the instruction register. Following the interrupt, the 8080 reads a word from the ICU which contains the address of the interrupting source. Program control is then transferred to the processing routine for the interrupting source.

Within the 8080 is an interrupt enable flag which must be set before any interrupt requests will be accepted from the ICU. This flag can be set by the EI (enable interrupt) and reset by the DI (disable interrupt) instructions. Each time an interrupt is accepted by the 8080, this flag is automatically reset, thus disabling further interrupts until set by an EI instruction.

The ICU contains an 8-bit Interrupt Mask Register, each bit of which is associated with an interrupt source. An interrupt is enabled if the associated bit in the Mask Register is a "one," disabled if it is a "zero." An interrupt will not be passed through the ICU unless it is enabled.

The 8080 can, therefore, selectively enable or disable interrupts by the data loaded into the Mask Register. The data to be loaded into the Mask Register is maintained in a protected location in memory. An interrupt is said to be 'armed' if the associated bit in this location is a "one," 'disarmed' if it is a "zero." The resident program can selectively set (arm) or clear (disarm) bits in this word before sending it to the Mask Register. Thus, an interrupt may be armed, but temporarily disabled by the resident.

Figure 1.1 and 1.2 contain the flow charts describing the processing for each type of interrupt. In these diagrams the symbol RTN means return to the program in progress when the interrupt occurred; SAVE means save the present contents of the registers and flags in the stack. In performing RTN the saved information will be restored to the registers, and interrupts will be enabled as they are then armed.

The ICU provides for eight interrupt sources with the priority and addresses shown in Table 1.

1.1 Serial IO (SIR and SOR)

PLATO data from the central computer arrives as 21-bit words at a rate of 57.14 words per second (1200 bps). The format of this data is described in section 2.0. After each 8-bit byte of input data is received, the SIR interrupt is generated to indicate the arrival of a new byte.

The 8080 responds to the interrupt by reading the data byte and storing it in memory. After three bytes have been received, the data is assembled into a PLATO job (a job is a PLATO word) and placed in the job stack. The job stack is a section of memory reserved for storing incoming PLATO jobs in the event they arrive while the terminal is busy. The job stack can hold up to 1.48 seconds of PLATO output (85 jobs).

The SOR interrupt indicates that the transmitter section of the serial IO port is available for use. The 8080 responds by loading any data awaiting transmission or, if no data is waiting, by disabling the SOR interrupt.

In addition to data, the serial IO port contains three IO status flags: the Lost Data flag, the XMIT Ready flag, and the RCV Ready flag. The Lost Data flag indicates that the 8080 failed to input one or more previously received words. The XMIT Ready flag indicates the present status of the transmitter. If this bit is "zero," the transmitter is busy transmitting data; while if it is "one," the transmitter is available for use. Before loading data into the transmitter, the 8080 examines this flag to determine availability. The RCV Ready flag indicates the arrival of an 8-bit byte in the port.

Request		Interrupt Address (hex)
SIR	highest priority	00
KST		01
TP		02
SOR		03
EXT0		04
EXT1		05
CONSOLE		06
CARRIER		07

Table 1. Interrupt Priority

1.2 Abort Mode

The 8080 maintains a record (word count) of the number of non-NOP (described in section 2.1) words received. Each time a non-NOP word is transferred into the 8080, the word count is incremented by 1. Upon receipt of a word containing a parity error or an indication of lost data, the 8080 automatically transmits the value of the word count to the computer center, sets the ABORT flag and enters the ABORT mode of operation. The value of the word count transmitted will indicate to the center the address of the word containing the error or the word that was lost.

The ABORT Mode flag indicates the error mode status of the terminal. If this flag is "zero," the terminal is operating normally; while if it is "one," the terminal is in the ABORT mode of operation. A red

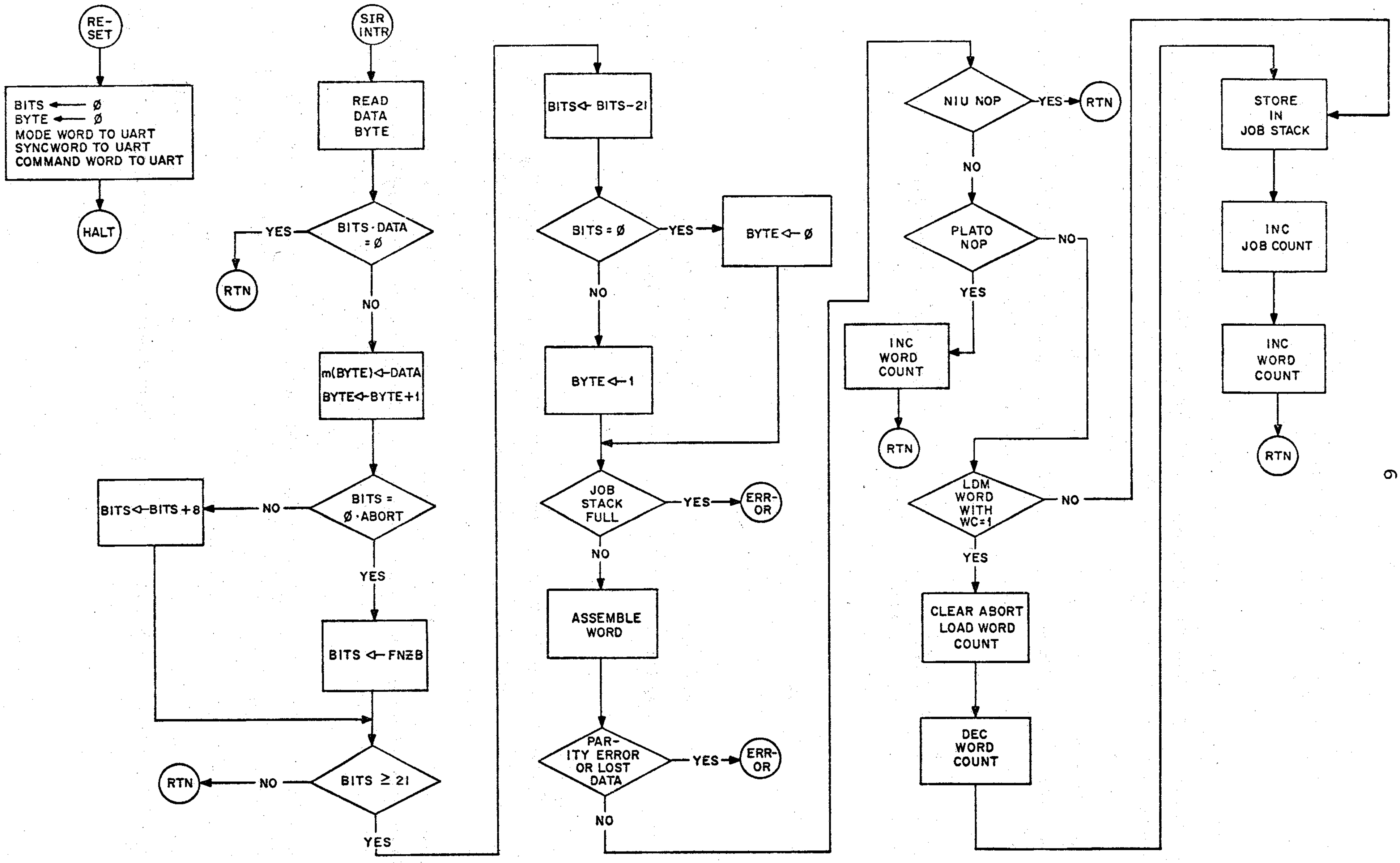


Figure 1.1. SIR Flow Chart

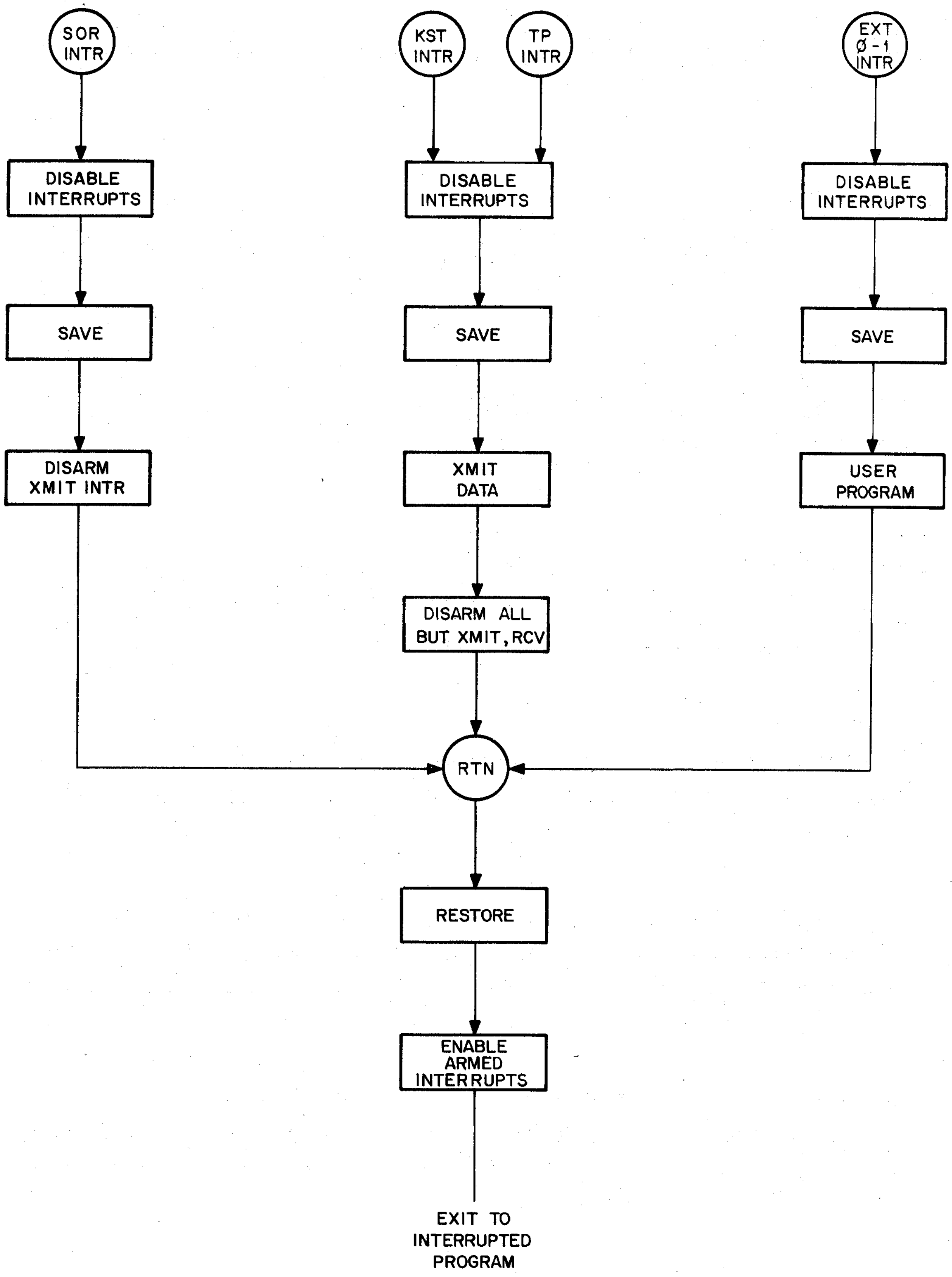


Figure 1.2. SOR, KST, TP, EXT Interrupts

indicator located on the front of the terminal indicates the status of the ABORT flag.

Once in the ABORT mode, the terminal will refuse to accept any further information except for an LDM instruction (described in section 2.1). Receipt of an LDM instruction with bit 14, a "one" will clear the ABORT flag and return the terminal to normal operating mode. This method of error control prevents the terminal from processing data in the wrong mode in the event an erroneous mode change word is received.

Overflowing the job stack will also cause the terminal to enter the ABORT mode. In this case the offending word is treated as though it arrived containing an error.

1.3 Keyset (KST)

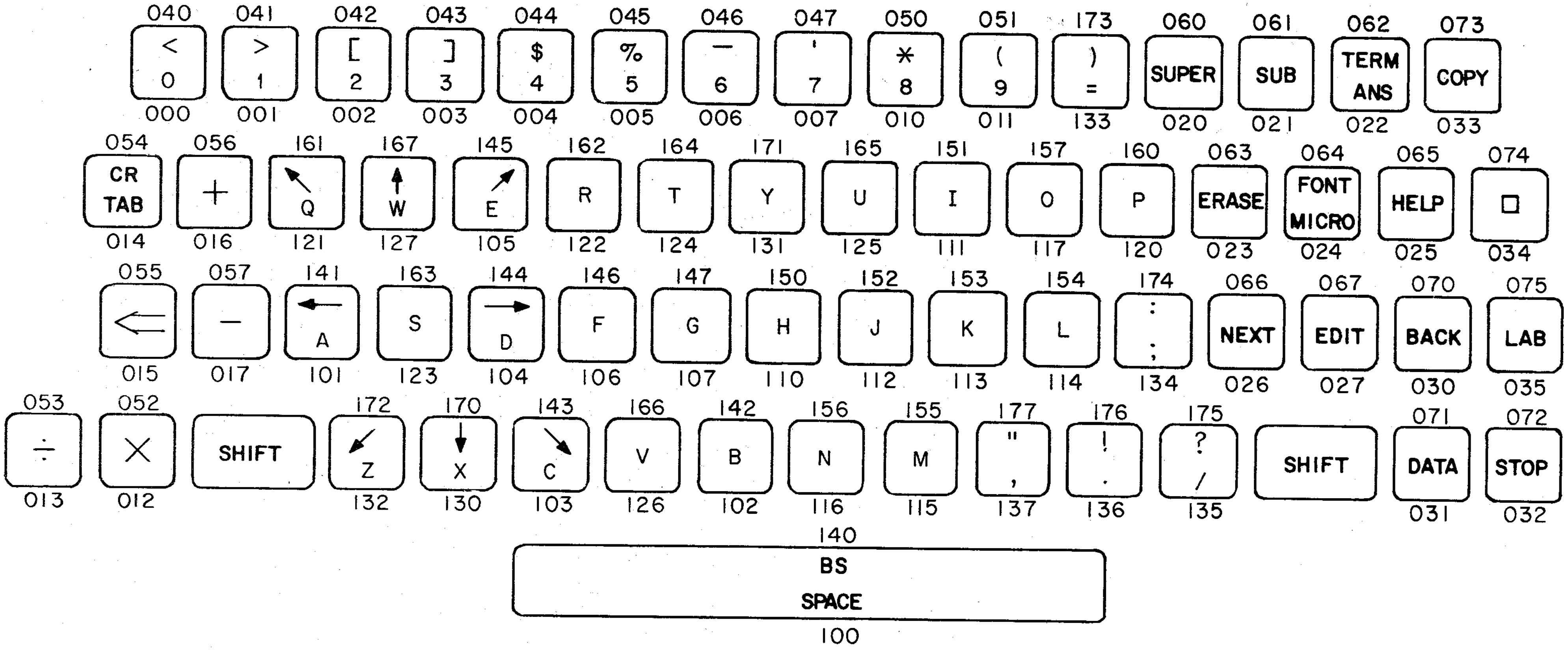
The KST interrupt indicates that a key has been pushed on the keyset. The layout and coding of the keyset are shown in Figure 1.3.

1.4 Touch Panel (TP)

The Touch Panel is an input device which allows the terminal user to touch the display surface and input positional information directly to the microprocessor. The touch sensitive surface is a 16 x 16 array of squares and the TP interrupt is generated whenever any square is touched. A short audio tone is generated each time the TP interrupt is accepted by the microprocessor.

1.5 External Input-Output (EXTØ)

The EXTØ interrupt indicates that a device attached to the external data bus is requesting service. The terminal provides for the attachment to the IO bus of up to 32 input and 32 output devices. Such devices, all optional, include a random-access slide projector for the projection of slide images on the rear of the plasma panel, a random-access audio unit which can play back to the terminal user pre-recorded audio messages, a ROM programmer, and a floppy disk system. Other user-defined output devices may also be attached. Data rates in excess of 25K bytes per second may easily be accomplished on the IO Bus.



Note:

1. Each key has two different inputs. The octal number below the box is the input when a key is pressed singly (normal state), and the number above the box is the input when the "Shift" key is held down as a key is pressed (shift state).

   The Shift key alone does not initiate input data transfer, but merely causes an addition of 040 (octal) to a normal input.

2. There is a total of 124 different inputs.
3. The input codes 036, 037, 076, and 077 are not used.

Figure 1.3. Keyset Coding

The EXT 1 interrupt is used internally by the terminal.

1.6 Console Mode (CONSOLE)

Contained within the resident program is a routine which permits the keyboard and display to be used as program debugging aids. To enter this routine, the RUN-CONSOLE switch must be placed in the CONSOLE position.

In CONSOLE mode the user may display the contents of the 8080 registers, the contents of memory, alter the contents of memory, step through programs one instruction at a step, set a breakpoint, and jump to other programs. This feature is described in detail in section 3.3.

1.7 Carrier Interrupt (CARRIER)

This signal is used to indicate an interruption of communication with the central computer. The resident program will generate a message on the display indicating loss of communication. (Use of this feature requires a modem with carrier detect capability.)

1.8 Display Interface Unit (DIU)

The DIU contains the registers and control circuits required to efficiently attach a plasma panel to the data bus. The DIU, shown in Figure 1.4, contains the 9-bit x and y display address registers, a 16-bit parallel data register (PDL/U), a 3-bit display mode (PDM) register and the write-erase control circuits. Data for the registers and control information is supplied to the DIU from the data bus. The x and y registers are bidirectional counters which can be independently controlled by the 8080. The parallel data register consists of the 8-bit parallel data upper (PDU) and parallel data lower (PDL) registers. The PDU/L registers are used only when operating parallel input display devices.

The format of the PDM register is shown in Figure 1.5. Bits 0 and 1 specify the write/erase mode, and bit 2, the panel operating mode. If bit 2 is "zero," the display is operating in the serial mode; and the contents of the x and y registers specify the address of a point to be

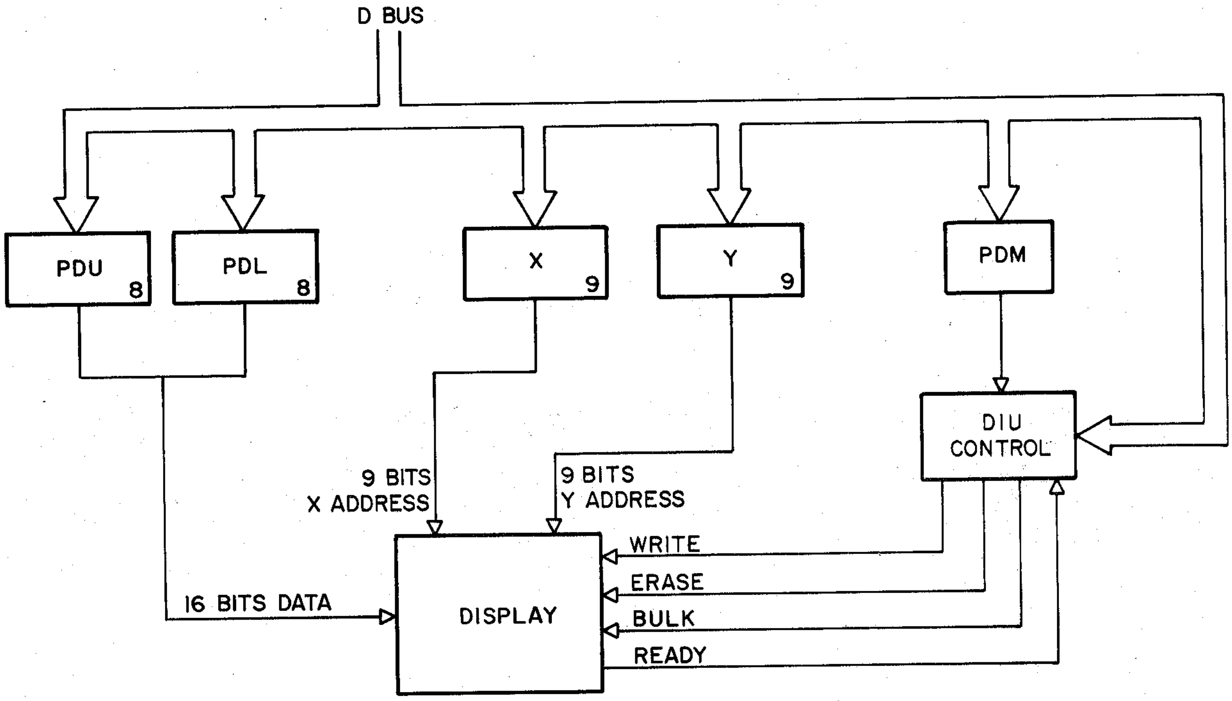


Figure 1.4 Display Interface Unit

written or erased. If bit 2 is "one," the display is operating in the parallel mode; and the contents of PDL/U will be written or erased on the panel at the address specified by the contents of the x and y registers. The data will be displayed in a vertical column with bit 0 of PDL at the bottom and bit 7 of PDU at the top.

Information is written on the display if $WE_0 = 1$ or erased of $WE_0 = 0$. The use of these bits is explained in section 2.5 in the discussion of Mode 3.

2	1	0
PM	WE_1	WE_0

Figure 1.5, PDM Register

1.9 Memory

The terminal memory consists of 16k (k = 1024) words, half of which is ROM and half RAM. The ROM portion of memory contains the resident program and the data for the standard PLATO character set. The resident provides all of the programs necessary to process PLATO data plus programs to service all interrupts. All graphical and IO routines in the resident are callable and may be accessed by user programs located in RAM.

More will be said about the resident in section 3.

2. OPERATING MODES

2.0 PLATO Word Format

The data to be processed by the terminal consists of 20-bit words (with start bit removed) with the format shown in Figure 2.0.

19	18	01	00
C/D	DATA		P

Figure 2.0. Terminal Word Format

Bit 00	Parity bit - even parity
Bits 01 - 18	Data
Bit 19	Control bit - 0 = control word - 1 = data word

Terminal words may be of two types: control words and/or data words. Data words (c = 1) contain the data to be processed by the terminal while control words (c = 0) are instructions used to establish operating conditions within the terminal.

2.1 Control Word Formats

The PLATO control word format is shown in Figure 2.1.

19	18	17	16	15	01	00
0	D			CONTROL INFORMATION		P

Figure 2.1. Control Word Format

Bits 01 - 15	Control Information
Bits 16 - 18	Type of Control Word

D = 000 (NOP)

19	18	17	16	15	01	00
0	0	0	0	0 — — — — — — — — — — — — — —	X	P

This word is a NOP (no-operation instruction). There are two types of NOP words, those generated by the PLATO communications hardware (bit 01 = 0) and those generated by the PLATO software (bit 01 = 1). The hardware NOP is generated automatically when the central computer has no data to be transmitted to the terminal. The software NOP can be used by system software to insert timing delays in the output data stream.

The software NOP will cause the terminal word count to be incremented while the hardware NOP will not affect the terminal status in any way. Neither of the NOPs is stored in the job stack.

D = 001 (LDM) Load Mode (267μs)

19	18	17	16	15	14	13 07	06 01	00
0	0	0	1	I	WC	WORD COUNT	MODE WORD	P

This instruction establishes the operating mode of the terminal. For each mode of terminal operation, there is an associated mode word (bits 01 - 06) which directs the processing of incoming data. Once placed in a given mode, the terminal remains in that mode until receipt of a new LDM instruction.

Eight different processing modes are available, five of which are incorporated in the terminal resident and three of which are reserved for local programs. The processing modes are described later in this section.

If bit 14 (WC) of the LDM word is "one," the word count register will be set to the value specified by bits 07 - 13. It is the receipt of this instruction with bit 14 set which will restore the terminal to normal mode if it is in the ABORT mode. This is the only instruction which the terminal will accept if it is in ABORT mode. Receipt of the LDM instruction while in the ABORT mode will clear the ABORT flag and initialize the word count, but will not alter the terminal processing.

Bit 15 of the LDM word is used to actuate or inhibit external devices attached to the terminal. Receipt of an LDM word with bit 15 a "one" will disable the TP and EXTO interrupt.

The terminal mode word format is shown in Figure 2.2.

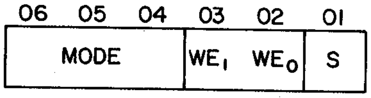


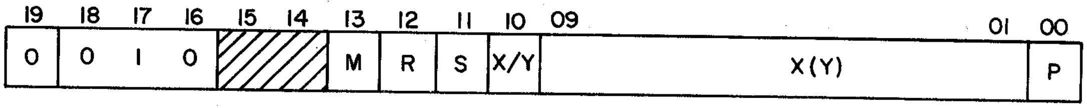
Figure 2.2. Mode Word Format

Bit 01	Screen Command. If this bit is "1," the entire display is erased.
Bits 02 - 03	Select write or erase function in the DIU as follows:

WE_1	WE_0	
0	0	Erase, write character background.
0	1	Write, erase character background.
1	0	Erase, suppress character background write.
1	1	Write, suppress character background erase.

Bits 04 - 06	Specify terminal processing mode.

D = 010 (LDC) Load Co-ordinate (225μs)



This instruction loads the x register (bit 10 = 0) or the y register (bit 10 = 1) with data as follows:

R	S	
0	x	Load register with bits 01 - 09.
1	0	Load register with the arithmetic sum of its present value and bits 01 - 09.
1	1	Load register with the arithmetic difference between its present value and bits 01 - 09.

In addition, if M (bit 13) = 1, the resultant value is also stored in memory location m.margin. All carriage returns performed by the terminal will set the x register (y register if plotting vertically) to the value contained in m.margin.

D = 011 (LDE) Status Request (267μs)

19	18 17 16	15 08	07 01	00
0	0 1 1	▨	STATUS REQUEST	P

This word is used to request information from the terminal or to request certain operations to be performed. Presently used codes are:

Code (hex)	Operation
70	Request terminal type. Terminal responds with code 73.
7B	Generate 1 second audible tone. (Touch panel must be attached for this operation.)
7D	Request terminal status information. The lower seven bits of the memory location specified by the Memory address register (MAR) will be returned to the best computer in a Status Response word (see section 2.8).

D = 100 (LDA) Load Memory Address (229μs)

19	18 17 16	15 01	00
0	1 0 0	INITIAL MEMORY STORE ADDRESS	P

This instruction loads the Memory Address Register (MAR). This data word specifies the first storage address to be used upon entry into a Mode 2 operation.

Note: Memory addresses below 2300 (HEX) are reserved for use by the resident program. Attempting to write below this address may result in erratic terminal behavior.

D = 101 (SSF) Function (223μs)

19	18 17 16	15 11	10	09	08 01	00
0	1 0 1	IO ADDRESS	R/W	I	DATA	P

This instruction is used to read and write data to devices attached to the external bus and to enable interrupts.

Bits 11 - 15 specify the device address.

Bit 10 specifies a read (input) operation if a "1," a write (output) if a "0."

Bit 09 inhibits the actual read or write function, but permits the device address to be saved by the terminal. The inhibit write function can be used to establish a write address for later use by the EXT command. The inhibit read function can be used to establish a read address to be used upon the occurrence of an external interrupt. If location m.extpa contains 0, the resident will perform a read from the selected device and transmit the data to PLATO via an External input word. PLATO may use a local program to process the interrupt by previously loading the program and storing the program address in m.extpa.

Write addresses 0 and 1 are special cases of the SSF instruction. Address 0 is assigned to the slide selector and for this device only the data field is 10 bits long. The SSF word format for this device is shown below.

19	18	17	16	15				11	10	09	08			05	04			01	00
0	1	0	1	0	0	0	0	0	L	S	X				Y				P

Bits 01 - 08 are sent to address 00 and bits 9 - 10 to address 01. Bits 01 - 08 select one of 256 slides for display on the plasma panel. Bit 09 controls the projector shutter. For normal operation this bit is always "0." However, if this bit is a "1," the shutter will be closed and remain closed until receipt of a load slide command with bit 09 = "0." Bit 10 controls the project lamp. The lamp will be turned on if bit 10 is a "1" and off if bit 10 is a "0."

Device address 1 is assigned to the Interrupt Mask register located in the ICU. The SSF word format for this address is shown below. Memory location m.enab will be loaded with a copy of the interrupt mask data.

19	18	17	16	15				11	10	09		07	06	05	04	03		01	00
0	1	0	1	0	0	0	0	1	0	▨	▨	▨	T	▨	E	▨	▨	▨	P

Bit 06 enables the Touch Panel if a "1," disables it if a "0." Bit 04 enables the External bus interrupt if a "1," disables it if a "0." (This data is not actually transmitted to address 01 because the Interrupt Mask Register is internal to the terminal.)

Following execution of this instruction, memory location m.enab will contain a copy of the mask data.

D = 11X (EXT) Load External (275μs)

19	18	17	16	09	08	01	00
0	1	1	BYTE 0		BYTE 1		P

This instruction transfers two 8-bit bytes, byte 0 first, to the external device selected by a previous SSF instruction.

2.2 Processing Modes - Mode 0

In normal operation, the terminal is assigned an operating mode by sending it a LDM instruction followed by all of the data to be processed in that mode.

The terminal resident program contains the programs for processing data in five modes. In addition, up to three additional user-defined-mode programs can be loaded into RAM.

Mode 0 is a point-plotting mode. Each mode 0 data word, Figure 2.3, specifies the address of a point on the panel to be written or erased.

The W/E_0 bit in the mode word determines which operation is performed.

19	18 10	09 01	00
I	X	Y	P

Figure 2.3. Mode 0 Data Word

The processing time for a Mode 0 word is 238μs.

2.3 Mode 1

Mode 1 is a line drawing mode. Each data word, Figure 2.4, specifies the terminal coordinates of a line, the origin of which is contained in the x and y registers.

19	18 10	09 01	00
I	X_i	Y_i	P

Figure 2.4. Mode 1 Data Format

The terminal point of a given line is also interpreted as the origin of the next line. Line origins may be relocated, however, by the use of the LDC command without exiting from Mode 01. A line will be drawn if $WE_{\emptyset}$ is a "1," erased if this bit is a "0."

The processing time for a Mode 1 word ranges from 1ms for a line length of one dot to 11.1ms for the maximum line length of 512 dots.

2.4 Mode 2

Mode 2 is a load memory mode. Each Mode 2 data word, Figure 2.5, contains two eight-bit bytes to be stored in RAM memory. These bytes are stored, lower first, in two successive locations starting with the present contents of the memory address register (MAR). After each byte is stored, the MAR is automatically incremented by 1.

19	18	17	16 09	08 01	00
I	C_1	C_0	BYTE 1	BYTE 0	P

Figure 2.5. Mode 2 Data Word

As each byte is stored, a longitudinal parity check is performed by exclusive "oring" each byte with a check word and left shifting the result. This check should be all zeros at the conclusion of the transmission of a block of data.

Bits 17 and 18 (C_0, C_1) activate the block error check as follows:

C_1	C_0	
0	0	Store data.
0	1	Byte 0 is the last byte of this transmission and contains the block check character to make the longitudinal check word all "zeros."
1	0	Byte 1 is the last byte of this transmission and contains the block check character.
1	1	Not used.

After receipt of the block check byte, if the check word is not zero, the terminal automatically transmits the STATUS REPORT code (05H) to the central computer. If the data being loaded is character data, it will appear when displayed as a vertical column with bit 01 of byte 0 at the bottom and bit 16 of byte 1 at the top.

The processing time for a Mode 2 word is 288μs.

2.5 Mode 3

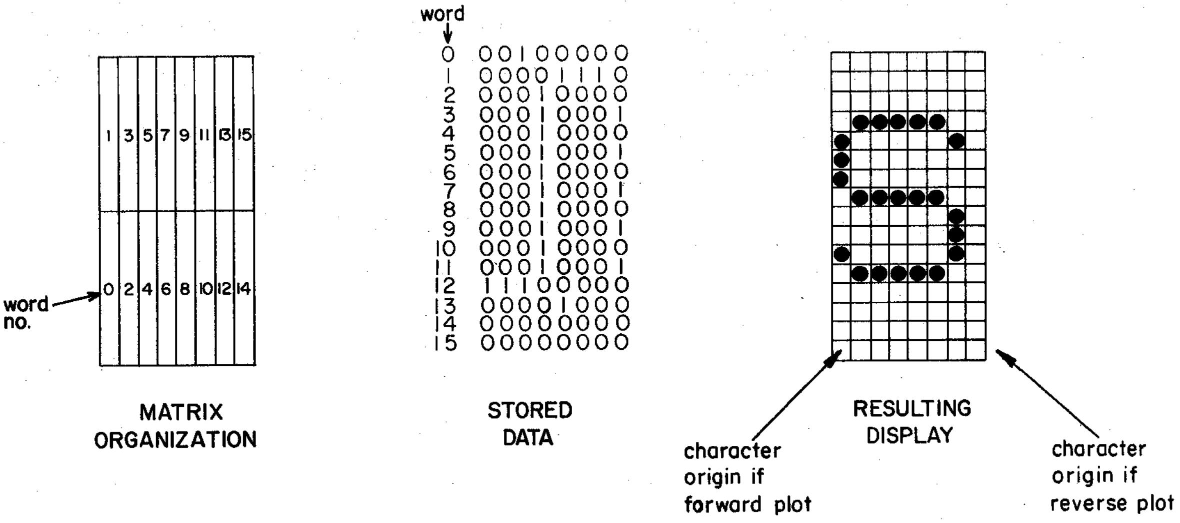
Mode 3 is a character-plotting mode. The data words in this mode contain three 6-bit character codes as shown in Figure 2.6. Each code selects one of 63 characters from one of the character sets contained in the terminal.

19	18 13	12 07	06 01	00
I	CHAR 1	CHAR 2	CHAR 3	P

Figure 2.6. Mode 3 Data Word

The terminal provides for up to eight character sets of 63 characters each. Character sets M0 and M1 are contained within ROM memory and hold the characters shown in Table 2. The other character sets contain user-defined characters and are stored in RAM.

The contents of the character memories are processed in Mode 3 as 63 arrays of sixteen 8-bit words. The contents of 16 consecutive addresses are displayed as one character within a matrix as shown in Figure 2.7. The top three rows and bottom row of all character matrices from M0 and M1 are always unfilled.



```
 0  00100000
 1  00001110
 2  00010000
 3  00010001
 4  00010000
 5  00010001
 6  00010000
 7  00010001
 8  00010000
 9  00010001
10  00010000
11  00010001
12  11100000
13  00001000
14  00000000
15  00000000
```

STORED DATA
RESULTING DISPLAY
character origin if forward plot
character origin if reverse plot

Figure 2.7. Character Matrix

The data for the characters stored in ROM is shown in Figures 2.8.0 - 2.8.3.

The contents of any character memory can be enlarged via selection of character size 2 (size 0 is normal size). Selection of size 2 will result in a 2X magnification of the characters. Figure 2.8.4 illustrates characters drawn in size 2. All character format operations will be automatically adjusted when using size 2 characters. The method of specifying character size is described later in this section.

Character write/erase is specified by the write/erase bits WE0, WE1 in the mode word. (See LDM instruction.) If WE0 = 1, characters are written; if WE0 = 0, characters are erased. The inverse of the operation called for by WE0 will be performed on the background or unfilled portion of the character matrix if WE1 = 0; while if WE1 = 1, the background remains unaltered.

Character plotting speed ranges from a minimum of 355 characters per second using a serial plasma panel and up to 3080 characters per second using a parallel plasma panel.

There are several non-plotting control characters available for formatting the display of data in Mode 3. These control characters may be accessed via the use of the "uncover" code (77). Upon receipt of a 77 code, the terminal interprets the next character code as a control character instead of a character to be plotted. Following execution of the control character, normal plotting mode is resumed. If several uncover codes are sent in sequence, the first non-uncover code will be treated as the control character.

The operations performed by each of the control characters are shown in Table 3. In the case of some characters, the operation performed is a function of the character memory, the plotting mode, horizontal or vertical, and the plotting direction, forward or reverse, which is being used.

Horizontal plotting mode is set by the 30 code; vertical plotting, by the 31 code. Forward plotting, set by the 32 code, is from left to right in horizontal mode and from bottom to top in vertical mode. Reverse plotting direction, set by the 33 code, is from right to left in horizontal mode and from top to bottom in vertical mode.

Boldface characters are selected by the 35 code, normal size by the 34 code.

The terminator code (00) is used to terminate character string plotting from local programs. See the discussion of r.chars in section 3.1.

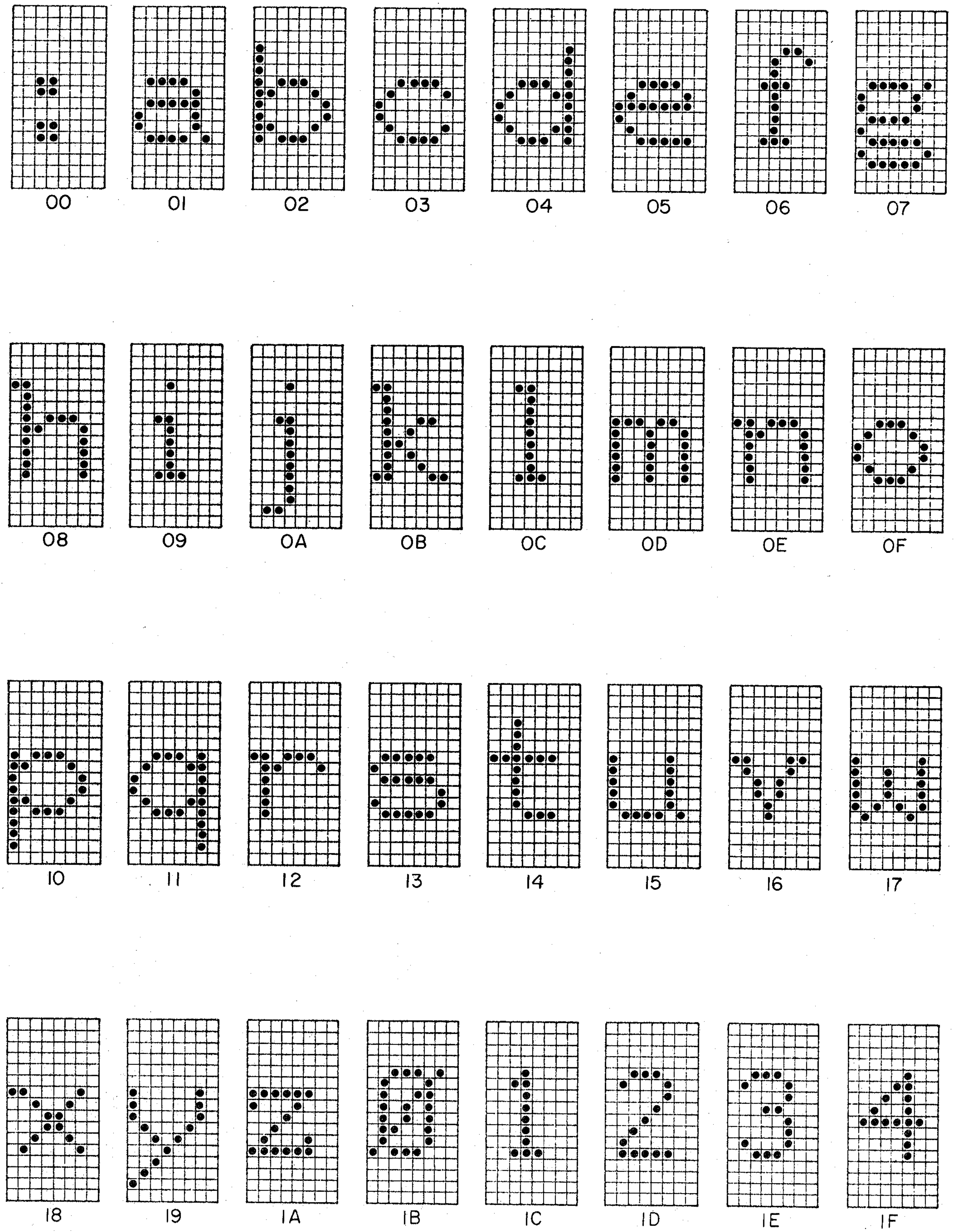


Figure 2.8.0. M0 Characters 00 - 1F

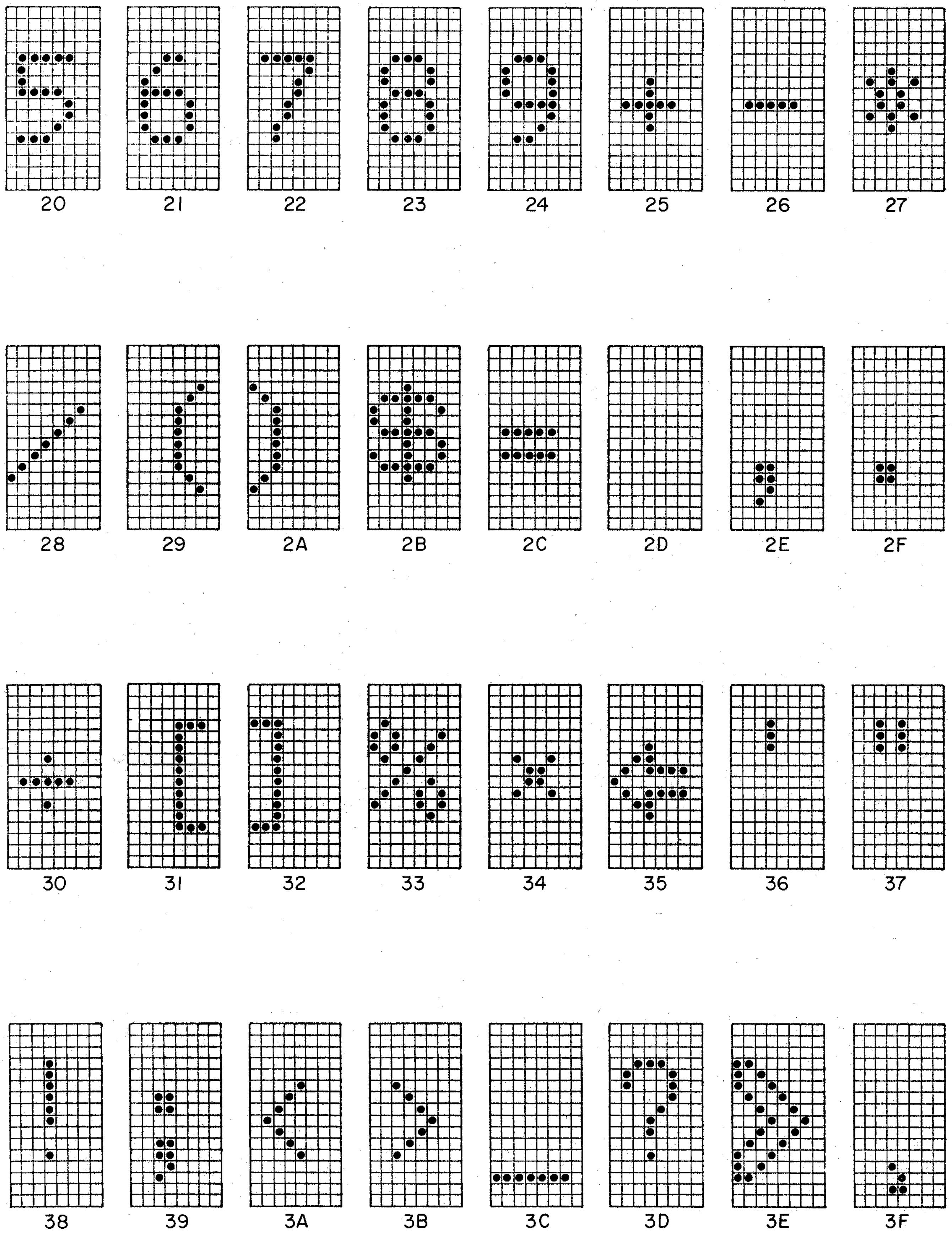


Figure 2.8.1. M0 Characters 20 - 3F

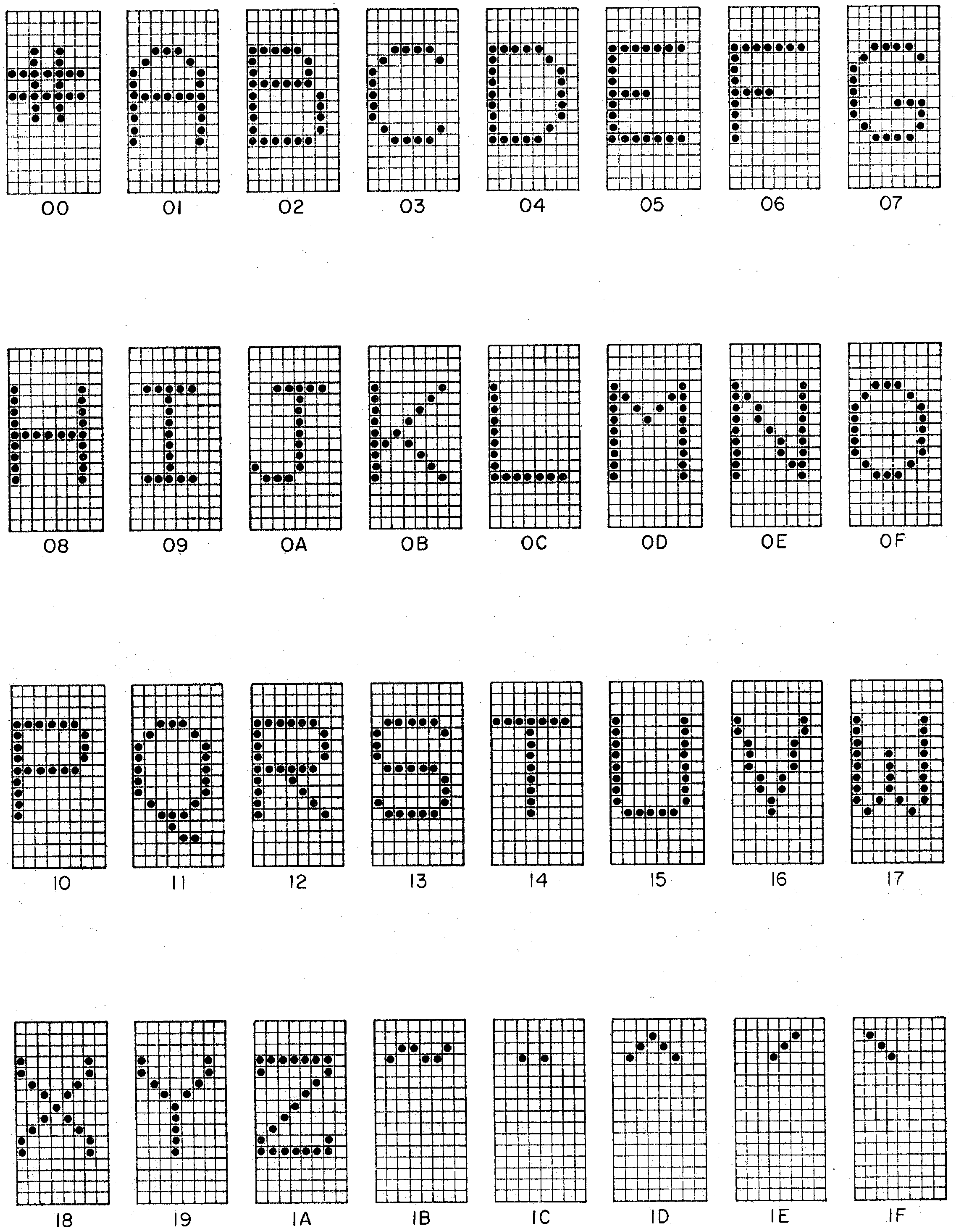


Figure 2.8.2   M1 Characters 00 - 1F

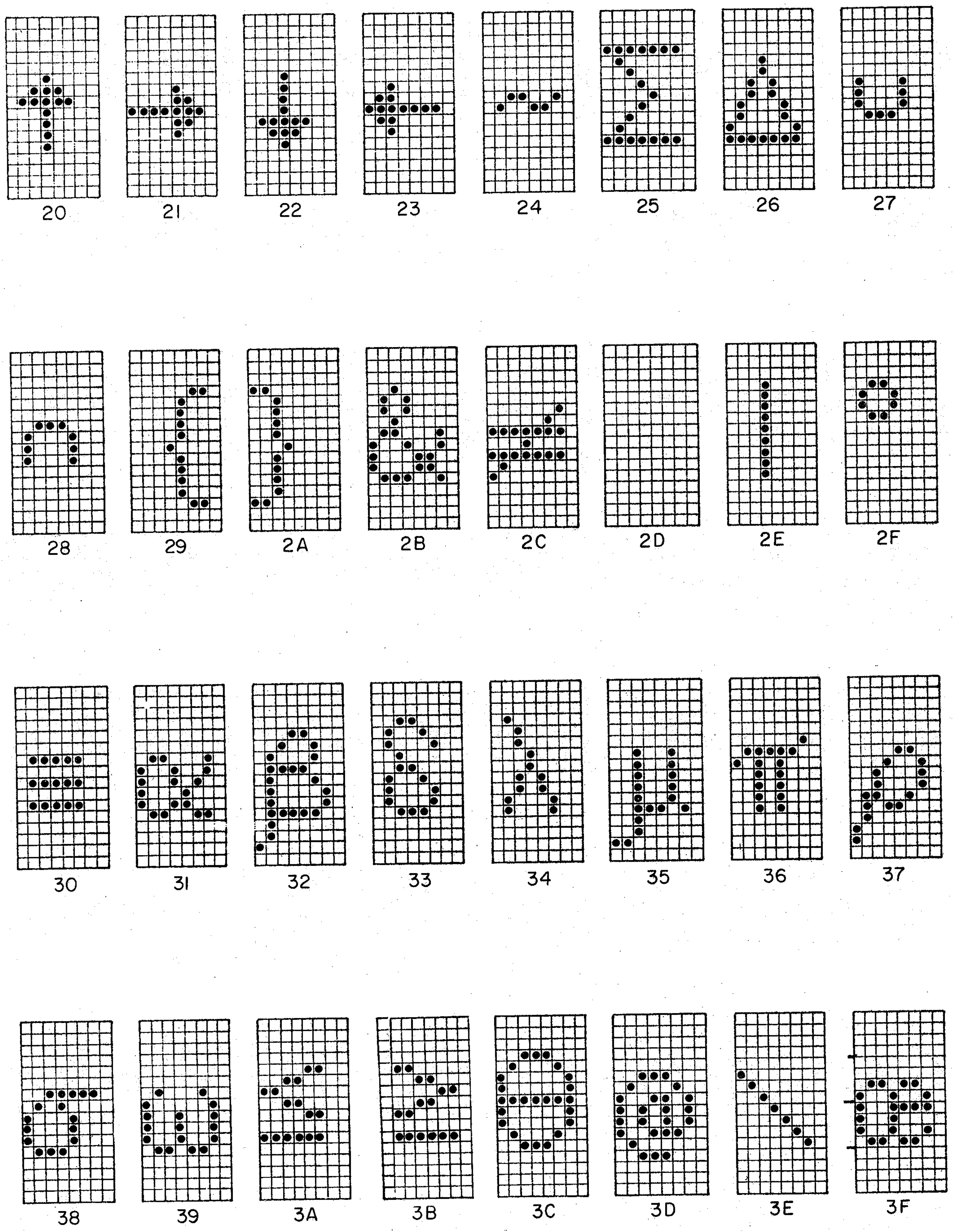
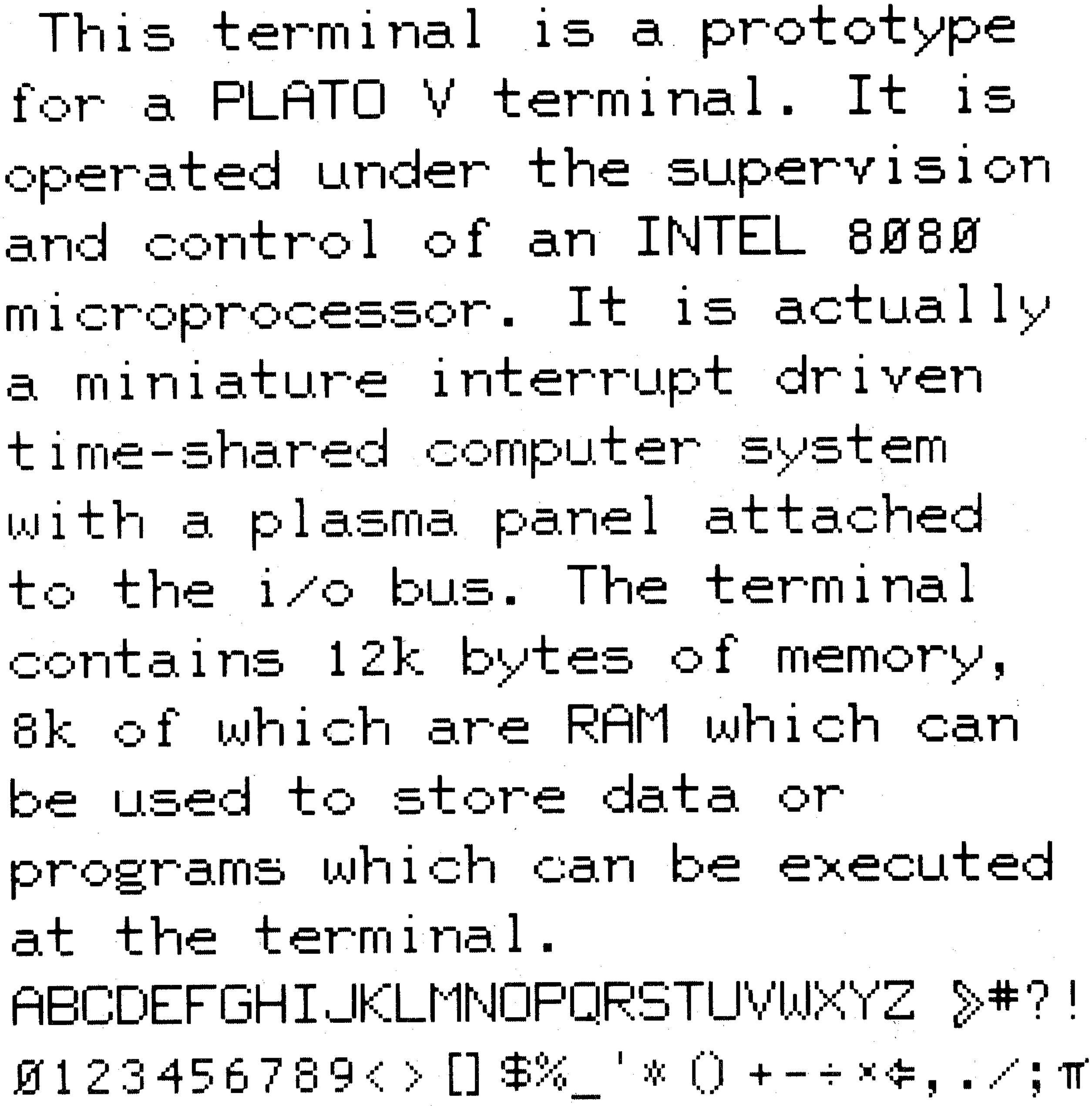


Figure 2.8.3 M1 Characters 20 - 3F



This terminal is a prototype for a PLATO V terminal. It is operated under the supervision and control of an INTEL 8Ø8Ø microprocessor. It is actually a miniature interrupt driven time-shared computer system with a plasma panel attached to the i/o bus. The terminal contains 12k bytes of memory, 8k of which are RAM which can be used to store data or programs which can be executed at the terminal.
ABCDEFGHIJKLMNOPQRSTUVWXYZ ⇒#?!
Ø123456789<>[]$%_'*()+-÷×≠,./;π

Figure 2.8.4 "Boldface" Character Set

ADDRESS (OCTAL)	M0 CHAR	M1 CHAR	ADDRESS (OCTAL)	M0 CHAR	M1 CHAR
0	:	#	40	5	↑
1	a	A	41	6	→
2	b	B	42	7	↓
3	c	C	43	8	←
4	d	D	44	9	∿
5	e	E	45	+	Σ
6	f	F	46	-	Δ
7	g	G	47	*	∪
10	h	H	50	/	∩
11	i	I	51	(	{
12	j	J	52	)	}
13	k	K	53	$	&
14	l	L	54	=	≠
15	m	M	55	SP	SP
16	n	N	56	,	|
17	o	O	57	.	°
20	p	P	60	÷	≡
21	q	Q	61	[	α
22	r	R	62	]	β
23	s	S	63	%	δ
24	t	T	64	×	λ
25	u	U	65	⇦	μ
26	v	V	66	'	π
27	w	W	67	"	ρ
30	x	X	70	!	σ
31	y	Y	71	;	ω
32	z	Z	72	<	≤
33	0	~	73	>	≥
34	1	¨	74	_	θ
35	2	^	75	?	@
36	3	´	76	❯	\
37	4	`	77	UNCOVER	UNCOVER

Table 2. ROM Characters

OCTAL CODE	FUNCTION	SIZE Ø				SIZE 2			
		HORIZONTAL		VERTICAL		HORIZONTAL		VERTICAL	
		FWD	RVS	FWD	RVS	FWD	RVS	FWD	RVS
00	Terminate	Terminate Character String Plotting							
10	Backspace	x-8	x+8	y-8	y+8	x-16	x+16	y-16	y+16
11	Tab	x+8	x-8	y+8	y-8	x+16	x-16	y+16	y-16
12	Line Feed	y-16	y-16	x+16	x+16	y-32	y-32	x+32	x+32
13	Vertical Tab	y+16	y+16	x-16	x-16	y+32	y+32	x-32	x-32
14	Form Feed	x←Ø y←496	x←5Ø4 y←496	y←Ø x←15	y←5Ø4 x←15	x←Ø y←480	x←496 y←480	y←Ø x←31	y←496 x←31
15	Carriage Return	x←(MARGIN) y-16	y-16	y←(MARGIN) x+16	x+16	x←(MARGIN) y-32	y-32	y←(MARGIN) x+32	x+32
16	Superscript	y+5	y+5	x-5	x-5	y+10	y+10	x-10	x-10
17	Subscript	y-5	y-5	x+5	x+5	y-10	y-10	x+10	x+10
20	Select MØ	select character memory Ø (ROM)							
21	Select M1	select character memory 1 (ROM)							
22	Select M2	select character memory 2 (RAM)							
23	Select M3	select character memory 3 (RAM)							
24	Select M4	select character memory 4 (RAM)							
25	Select M5	select character memory 5 (RAM)							
26	Select M6	select character memory 6 (RAM)							
27	Select M7	select character memory 7 (RAM)							
30	Horizontal	select horizontal plot mode							
31	Vertical	select vertical plot mode							
32	Forward	select forward plot direction							
33	Reverse	select reverse plot direction							
34	Select Size Ø	select normal-size characters							
35	Select Size 2	select large-size characters							
77	Uncover	next character is control character							

Table 3. Control Characters

Six memory locations are reserved as address registers specifying the origins of the loadable character sets. These locations, shown in Table 4, are normally loaded by PLATO and the character sets are referenced by the codes in Table 3.

Address	Function
2306	Character set 02 origin.
2308	Character set 03 origin.
230a	Character set 04 origin.
230b	Character set 05 origin.
230e	Character set 06 origin.
2310	Character set 07 origin.

Table 4. Character set Address Registers

2.6 Mode 4

Mode 04 is a block erase mode. In this mode each pair of data words specifies the corners of an area to be erased. The area erased is that enclosed by $|\Delta x| = |x_2-x_1|$ and $|\Delta y| = |y_2-y_1|$. If $x_2 = x_1$ and $y_2 = y_1$, a single point is erased; while if $x_2 = x_1$ and $y_2 \neq y_1$, a vertical line is erased; and if $y_2 = y_1$ and $x_2 \neq x_1$, a horizontal line is erased. The previous description assumes $WE_0 = 0$; if $WE_0 = 1$, the area is written.

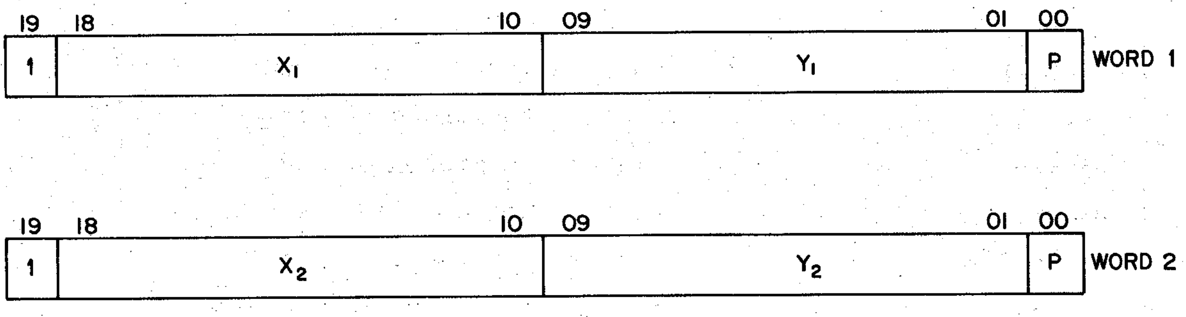


Figure 2.9. Mode 4 Word Format

After processing each pair of words in this mode, the terminal leaves the display address registers set to x_1 and y_1-15. This is the address appropriate to begin writing characters in the erased area.

2.7 Modes 5,6,7

The word format for any of these modes is defined by the user. When operating in any of these modes, the resident places the PLATO data in the C, D, and E registers as shown in Figure 2.10, and transfers control to the local program.

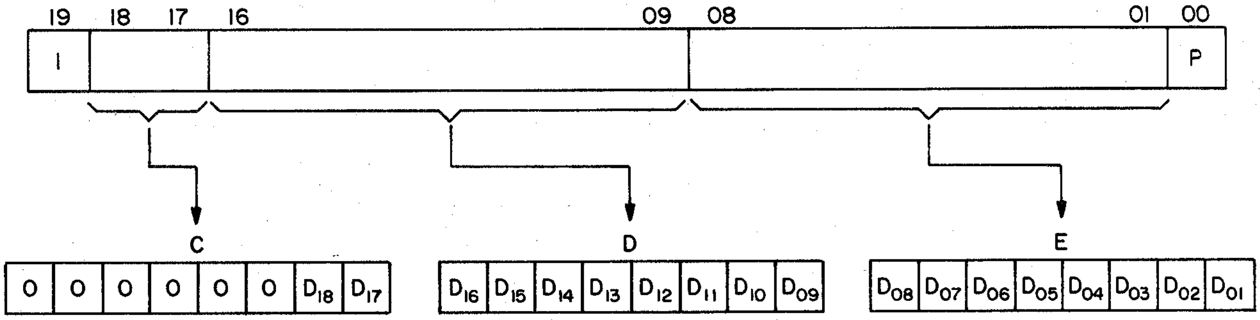

19
18
17
16
09
08
01
00
I
P
C
D
E
0 0 0 0 0 0 D_{18} D_{17}
D_{16} D_{15} D_{14} D_{13} D_{12} D_{11} D_{10} D_{09}
D_{08} D_{07} D_{06} D_{05} D_{04} D_{03} D_{02} D_{01}

Figure 2.10. Modes 5,6,7 Word Format

Three memory locations are reserved for use as address registers specifying the origins of the local programs. These addresses are shown in Table 5.

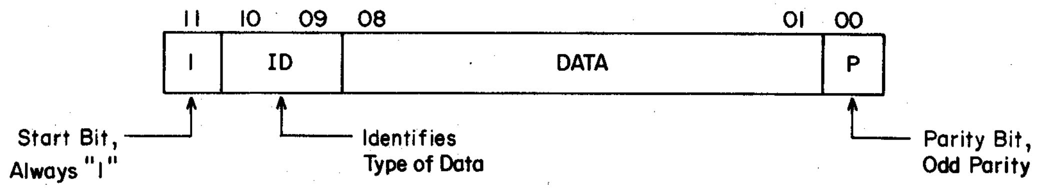
Address	Symbol	Function
2300	m5	Mode 5 program origin.
2302	m6	Mode 6 program origin.
2304	m7	Mode 7 program origin.

Table 5. Local Program Address Registers

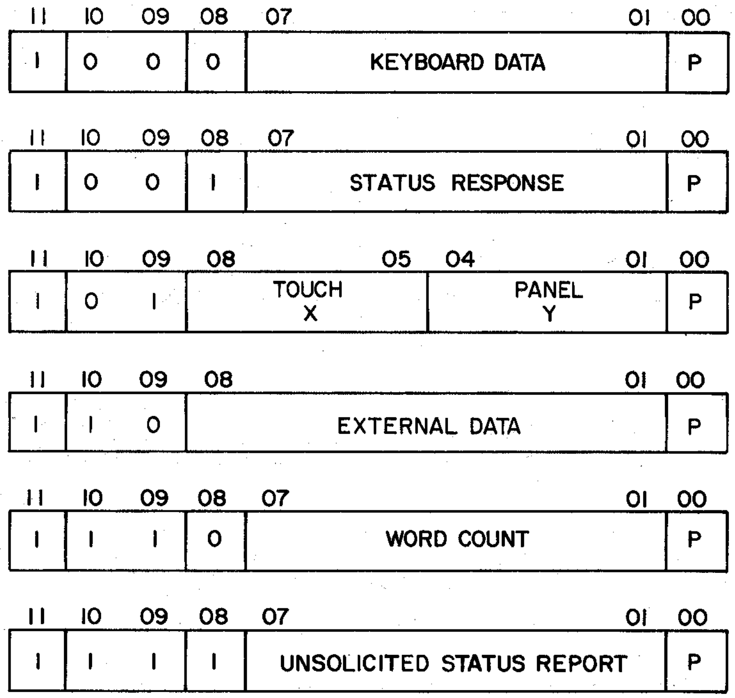
Note: Once the resident has transferred control to a local program, control of the terminal remains with that program until a return instruction is executed (or until the clear switch is depressed). If the interrupts are left enabled the resident will continue to perform all IO functions.

2.8 Output Data Format

Data transmitted from the terminal to the computer center consists of 12-bit words with the format shown below.



The six types of terminal data words are shown below:



Status Request code 70H is used to request terminal type. A response of 73H indicates an 8080-type terminal with 8k of RAM memory.

The Unsolicited Status Report codes are used to inform the central computer of the occurrence of some special event within the terminal. A list of the presently used codes is shown in Table 6.

Status Report (hex)	Event Reported
02	Reset (clear switch has been depressed).
05	Longitudinal parity error occurred in Mode 2.

Table 6. Unsolicited Status Report Codes

3. RESIDENT PROGRAM

3.0 General

The terminal memory allocation is shown in Figure 3.0. All of ROM and that portion of RAM below address 2300 (hex) is reserved for use by the resident. The resident program contains those programs required to process PLATO data plus routines for operating the serial communication port, the parallel IO bus, servicing interrupts and communicating with the Display Interface Unit.

The push down stack is used by the resident to store the status of the terminal during the processing of interrupts. The job stack provides temporary storage for incoming jobs in the event the processor is busy. The resident and PLATO variable sections of memory contain terminal status information which may be used by both the resident and user programs.

3.1 Resident Subroutines

The resident program provides several callable subroutines which may be referenced by user programs. In using these subroutines the following convention should be observed:

1. Single argument subroutines will have the argument passed in the HL register pair.
2. Double argument subroutines will have one argument passed in HL and the other in DE.
3. Results are returned from subroutines in HL.
4. The user must provide for saving and restoring any register or status he wants preserved.
5. Reference to a subroutine should always be by symbol and not memory address.

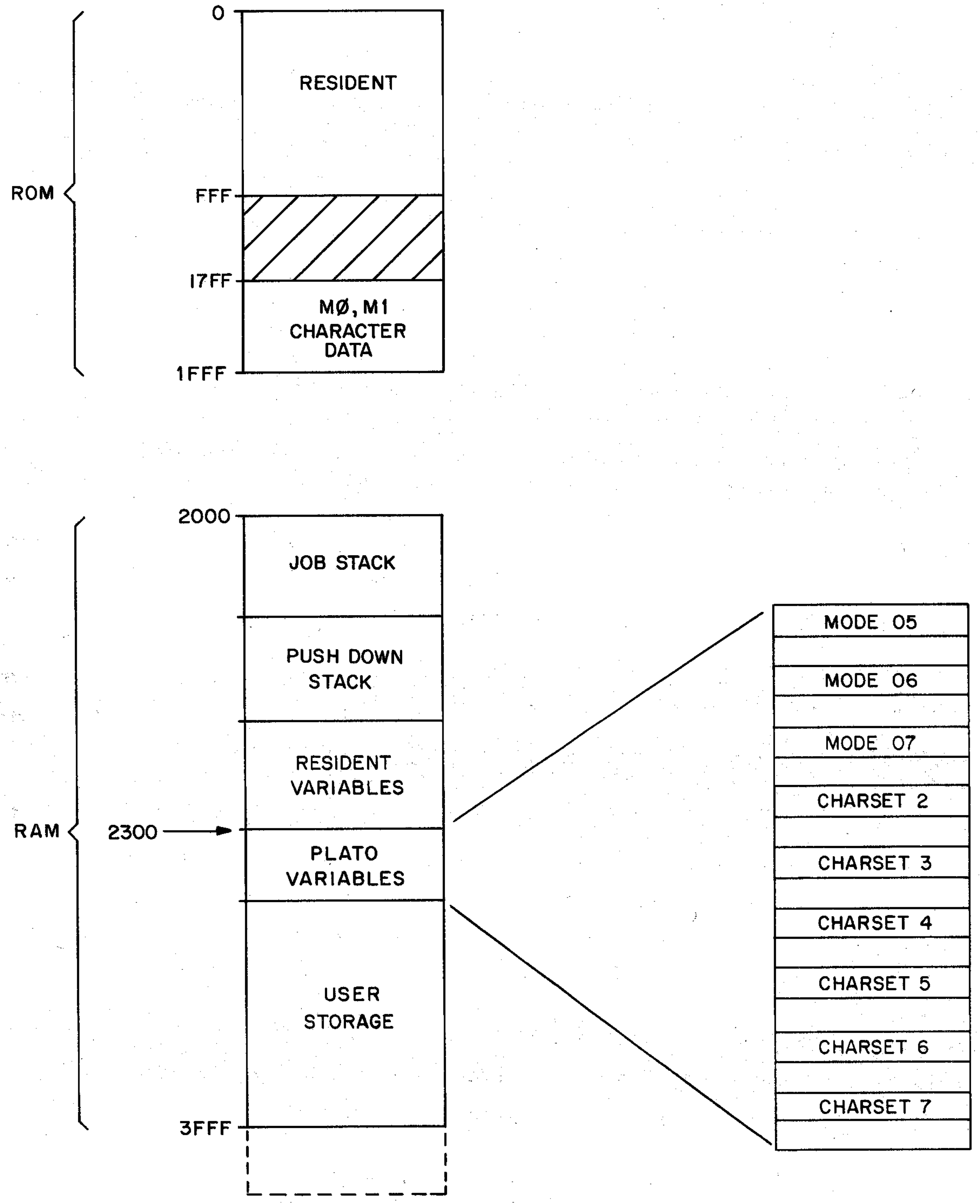


Figure 3.0. Memory Allocation

Following is a list of the subroutines.

r.init (40)

This routine will initialize the terminal operating conditions as follows:

a. The screen will be erased.
b. Memory locations m.margin, m.ksw, and m.extpa will set to zero.
c. M.enab will set to enable the serial input port (SIR, SOR, CARRIER), and the keyset (KST) interrupts.
d. Select character memory m0; select normal character size; select horizontal left to right plotting mode.
e. Initialize other pointers required by the resident.
f. Remove ABORT condition if it exists.

Note: A jump must be made to this routine in which case control will be returned to the resident after execution.

r.dot (43)

This routine will write (or erase) the point on the display screen specified by the contents of the HL and DE registers.

H	
▨	X_{08}

L							
X_{07}	X_{06}	X_{05}	X_{04}	X_{03}	X_{02}	X_{01}	X_{00}

D	
▨	Y_{08}

E							
Y_{07}	Y_{06}	Y_{05}	Y_{04}	Y_{03}	Y_{02}	Y_{01}	Y_{00}

The WE_0 bit in location m.mode specifies a write operation if a "1," an erase operation of a "0." After execution of this routine the x and y registers in the DIU will contain the values entered in HL and DE.

r.line (46)

This routine will write (or erase) a line on the display screen originating at the current address given by the x and y registers and terminating at the address contained in HL and DE.

H	
▨	X_{08}

L							
X_{07}	X_{06}	X_{05}	X_{04}	X_{03}	X_{02}	X_{01}	X_{00}

D	
▨	Y_{08}

E							
Y_{07}	Y_{06}	Y_{05}	Y_{04}	Y_{03}	Y_{02}	Y_{01}	Y_{00}

The WE_0 bit in m.mode specifies a write operation if a "1," an erase operation if a "0." After execution of this routine the x and y registers in the DIU will contain the values entered in HL and DE.

Note: This routine will enable interrupts via execution of an EI instruction.

r.chars (49)

This routine will write (or erase) a string of characters on the display. Register pair HL specifies the string origin address. The string must be terminated with an uncover code followed by the terminator code (7700). Character coding is the same as shown in Tables 2 and 3. Character write/erase is specified by the WE_0 and WE_1 bits in location m.mode as described in the discussion of mode 3.

r.block (4c)

This routine will erase (or write) an area of the display screen specified by a list of coordinates stored in memory. Register pair HL contains the origin address of the list. The coordinates are stored in the following order:

hl	x7------x0	x_1 lower
hl+1	0000000x8	x_1 upper
hl+2	y7------y0	y_1 lower
hl+3	0000000y8	y_1 upper
hl+4	x7------x0	x_2 lower
hl+5	0000000x8	x_2 upper
hl+6	y7------y0	y_2 lower
hl+7	0000000y8	y_2 upper

Coordinates x_1y_1 and x_2y_2 are any two corners of the area involved. The area will be erased if the WE_0 in location m.mode is "0," written if it is a "1."

r.inpx (4f)

This routine will return the current display x address in HL.

H								L							
0	0	0	0	0	0	0	X_{08}	X_{07}	X_{06}	X_{05}	X_{04}	X_{03}	X_{02}	X_{01}	X_{00}

r.inpy (52)

This routine will return the current display y address in HL.

H								L							
0	0	0	0	0	0	0	Y_{08}	Y_{07}	Y_{06}	Y_{05}	Y_{04}	Y_{03}	Y_{02}	Y_{01}	Y_{00}

r.outx (55)

This routine will load the display x address register with the contents (lower 9 bits) of HL.

H								L							
0	0	0	0	0	0	0	X_{08}	X_{07}	X_{06}	X_{05}	X_{04}	X_{03}	X_{02}	X_{01}	X_{00}

r.outy (58)

This routine will load the display y address register with the contents (lower 9 bits) of HL.

H								L							
0	0	0	0	0	0	0	Y_{08}	Y_{07}	Y_{06}	Y_{05}	Y_{04}	Y_{03}	Y_{02}	Y_{01}	Y_{00}

r.xmit (5b)

This routine will transmit to PLATO the contents (lower 10 bits) of HL.

H								L
0	0	0	0	0	0	I_{01}	I_{00}	DATA

i_1i_0 identify the source of data as follows:

0 0	keyset or status response word.
0 1	touch panel word.
1 0	external input word.
1 1	word count or unsolicited status word.

The output word formats are described in more detail in section 2.7.

r.mode (5e)

This routine will load location m.mode with the terminal and display operating mode as specified by the contents of HL.

H	L							
▨	0	0	M_{02}	M_{01}	M_{00}	WE_{01}	WE_{00}	S

s specifies a full screen erase (the screen will be erased when this routine is called if s is "1," the screen is not affected if this bit is "0").

WE_1WE_0 specify the write erase mode as follows:

00	erase, rewrite mode.
01	write, rewrite mode.
10	erase, overstrike mode.
11	write, overstrike mode.

$m_2m_1m_0$ specify terminal processing mode.

Note: The screen bit is discarded before storing the data leaving m.mode with the following format:

0	0	0	M_{02}	M_{01}	M_{00}	WE_{01}	WE_{00}

r.stepx (61)

This routine will increment or decrement the display x address register. The contents of location m.dir specify the operation to be performed as follows:

0	0	0	0	0	0	XD	YD

xd specifies x direction as follows:

0 = forward (increment)
1 = reverse (decrement)

yd specifies y direction.

r.stepy (64)

This routine will increment or decrement the display y address register. The contents of location m.dir specify the operation to be performed as in r.stepx.

r.we (67)

This routine will write (or erase) the current xy address.

r.dir (6a)

This routine will setup the x and y direction flags for later use by the r.stepx and r.stepy routines. Location m.dir will be loaded with contents of HL.

0	0	0	0	0	0	XD	YD

xd specifies x direction as follows:

0 = forward (increment)
1 = reverse (decrement)

yd specifies y direction.

r.input (6d)

This routine will return in HL the last keyset, touch panel or external word received by the terminal. If location m.ksw is 0, the word will also have been sent to PLATO. If m.ksw = 3, the word is returned only to the users program.

The format of the data is shown below.

H								L
D	0	0	0	0	0	I_{01}	I_{00}	DATA

d "0" if data present, "1" of no data.

i_1i_0 specify source of data as follows:

0 0	keyset word.
0 1	touch panel word.
1 0	external word.

r.ssf (70)

This routine is used to read and write data to devices on the external bus and to enable interrupts.

0	A_{04}	A_{03}	A_{02}	A_{01}	A_{00}	R/W	I	DATA

a_4-a_0 specify the device address.

r/w specifies a read (input) if "1," a write (output) if "0."

i if this bit is "1," the device address will be saved by the resident, but the actual read or write operation will be inhibited.

data in an ssf write operation, L contains the data to be sent to the external device. In an ssf read operation, r.ssf will return in L the data read from the external device.

The inhibit write ssf is normally used to establish a write address for later use by the r.extout routine (described later). The inhibit read ssf is used to establish a read address for later use by an external interrupt processing program. In this case, the user must have previously loaded location m.extpa with the address of the interrupt program. If m.extpa contains 0, the resident will perform the read operation when the external interrupt occurs. The data will be stored in memory and may be retrieved via the r.input routine. Note that if the resident interrupt program is used, location m.ksw must contain a 3 to prevent the data from also being sent to PLATO.

Write addresses 0 and 1 are special cases of the ssf routine. Device address 0 is assigned to the slide selector and for this device only, the data bits are sent to address 0 and bits 09 and 10 are sent to address 1.

Device address 1 is assigned to the Interrupt Mask register which is located within the terminal. The data format for this register is described below.

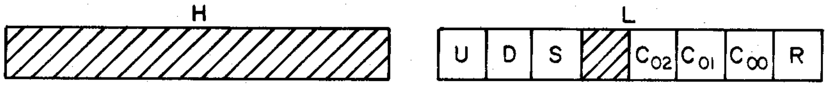
SIR	KST	TP	1	EXT_0	0	CON	CAR

sir = serial input port interrupt
kst = keyset interrupt
tp = touch panel interrupt
ext_0 = external IO interrupt
con = console switch interrupt
car = modem carrier interrupt

An interrupt is enabled if the associated bit is a "1," disabled if "0." Memory location m.enable will be loaded with a copy of the interrupt mask data.

r.ccr (73)

This routine is used to establish character plotting conditions for use by the r.chars routine.



u specifies that the last character plotted was the "uncover" code (77). This bit should normally be set to "0."

d specifies the character plotting direction. A "0" = forward, a "1" = reverse. Forward direction is left to right in horizontal plotting and bottom to top in vertical plotting.

s specifies character size, 0 = normal, 1 = large.

$c_2c_1c_0$ specifies character memory.

r specifies horizontal plot if "0," vertical plot if "1."

Location m.ccr will be loaded with the data.

r.extout (76)

This routine will transmit the data stored in a buffer to the currently selected device on the external IO bus. The device address may be set by the r.ssf routine.

Register HL specifies the origin of the string and DE the length.

r.exec (79)

This routine may be used to execute the jobs waiting in the job stack. The job stack can hold a maximum of 1.48 seconds of PLATO output (excluding NOPs). The number of jobs awaiting execution is contained in location m.jobs.

Careful use of this routine will permit time sharing of the terminal by PLATO and local programs.

r.gjob (7c)

This routine may be used to retrieve the next job awaiting execution in the job stack. The job is returned in registers DE and HL with the format shown below.

E							
D_{19}	▨	D_{18}	D_{17}	D_{16}	D_{15}	D_{14}	D_{13}

H							
0	0	D_{12}	D_{11}	D_{10}	D_{09}	D_{08}	D_{07}

L							
0	0	D_{06}	D_{05}	D_{04}	D_{03}	D_{02}	D_{01}

The job retrieved will not be executed by the resident. Location m.jobs will be decremented for each job retrieved.

Note: Attempting to use this routine when m.jobs is zero (emtpy job stack) will cause erroneous data to be returned in DE and HL.

r.xjob (7f)

This routine may be used to pass to the resident for execution the job contained in registers E, H and L. The format of the job is the same as for r.gjob.

3.2 Resident Variables

This section of memory contains information used by the resident which may also be used by local programs. This data may be read directly by local programs, with the exception of m.dir and m.ksw, should be written using the callable routines described earlier.

Note: Access to this data should always be by symbol and not by address!

Address	Symbol	Function
22EA	m.flagØ	status flags. The lower series bits of this location are used to indicate status information of a local program. The function of each flag (bit) is defined by the user. The resident program will reset (to "0") all flags when control is returned to the resident.
22EB	m.type	specifies terminal type.
22EC	m.clock	16 bit clock having a period of 6.67ms. The SIR interrupt must be enabled to insure accurate timing.
22EE	m.extpa	specifies address of program to process of external interrupts. A value of 0 specifies the resident interrupt processing program.

Address	Symbol	Function
22F0	m.margin	specifies present margin setting to be used for carriage returns.
22F2	m.jobs	specifies number of jobs remaining in job stack. See r.gjob and r.exec for use of this data.
22F4	m.ccr	specifies character mode plotting conditions. See r.chars for definition of data.
22F6	m.mode	specifies terminal operating mode. See r.mode for definition of data.
22F8	m.dir	specifies directional information for display address registers. See r.dir, r.stepx, and r.stepy.
22FA	m.ksw	controls transmission of data to central computer. If this location contains 0, data is transmitted to central computer; if this location contains 3, data is retained at terminal. See r.input for more details.
22FC	m.enab	specifies interrupt selection. See r.ssf for more details.

3.3 Console Program

This program is provided in the resident for use as a program diagnostic tool. It permits the keyset and display to be used as a computer console. To gain access to this program, the RUN-CONSOLE switch must be placed in the CONSOLE position. The contents of the 8080 registers will then be displayed across the top of the display screen and an arrow appears in the lower left corner where directives may be entered from the keyset. These directives and their function are described below.

Keyboard Entry	Function
dxxxx	Display 256 bytes of memory starting at address $xxxx_{16}$.
ixxxx,DD	Insert Data DD_{16} into memory location $xxxx_{16}$.
bxxxx	Set breakpoint register (BP) to $xxxx_{16}$.

Keyboard Entry	Function
jxxxx	Jump (call) to location $xxxx_{16}$. If RUN-CONSOLE switch is in CONSOLE position, one instruction will be executed at xxxx and control will return to the CONSOLE program. If the switch is in RUN position, an exit from CONSOLE mode will occur to address xxxx.
cxxxx	Call program at address $xxxx_{16}$ (called program must contain a RTN instruction to return to CONSOLE routine).
(space Bar)	Step. Execute one instruction at the present value of the program counter (PC).
□ key	Execute program starting at the present value of PC and continue until the breakpoint address is encountered. The contents of the 8080 registers are displayed during program execution.
shift □	Same as above, but do not update register display. The program will be executed faster than above.
f xxxx, nn, cc	Fill memory, beginning at address $xxxx_{16}$ and continue for nn_{16} locations, with data cc_{16}. The full character will be assumed 0 if cc is absent.
rn,xx	Load register n with data xx_{16}. The registers are numbered from left to right as they are displayed across the top of the screen. 0 = A, 1 = flags (P), 2 = B, 3 = C, 4 = D, 5 = E, 6 = H, 7 = L
stop	Stop program execution.

Placing the RUN-CONSOLE switch in the RUN position and pressing SHIFT-STOP or SHIFT-BACK will return control to the resident. Pressing the CLEAR switch will also return control to the resident, but will also initialize the terminal. If the space bar is pressed with the switch in the run position, program execution will resume at the instruction shown in the register display, and the monitor will no longer have control.

The monitor may be entered by execution of a rst 5 instruction in any user program.

3.4 Input-Output Addresses

Input and output operations are performed by the IN and OUT instructions and involve the transmission of 8-bit data words between the accumulator (A) register and devices external to the 8080 processor. In some cases the OUT instruction is used to set control flags in the Display Interface Unit, and no data is actually transferred. In these cases the contents of A are immaterial.

The format of the I/O address is shown in Figure 3.1.

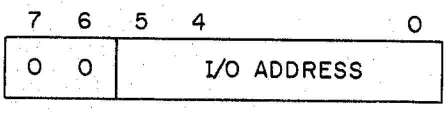


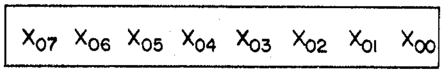
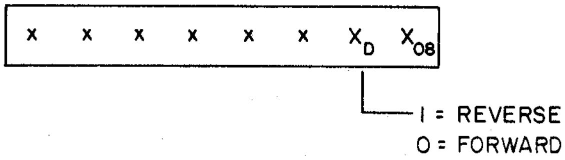
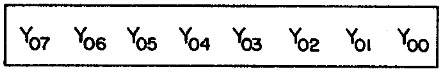
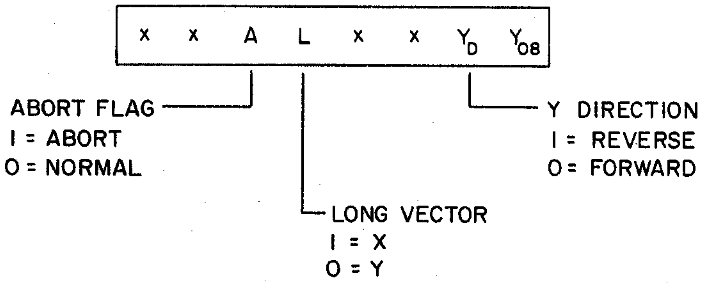
Figure 3.1. I/O Address Format

Addresses 0 through 1F (hex) are used internally by the terminal while 20 through 3F are used externally. Only external addresses may be specified by the SSF instruction.

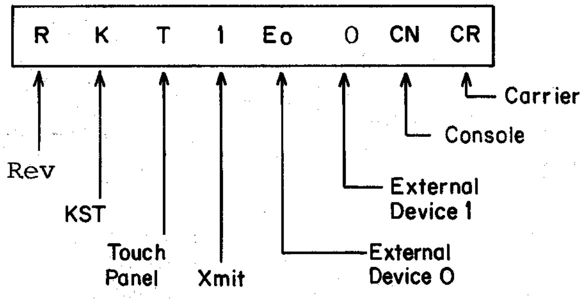
The address assignments are tabulated on the following pages. They appear here for information purposes only; the user should not write programs using these addresses, but instead should make use of the callable subroutines described earlier.

INPUT

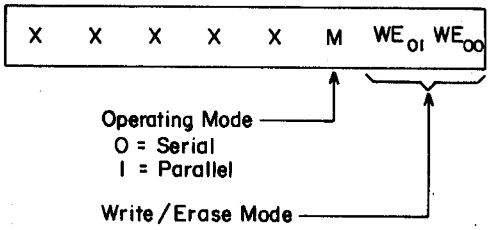
Input Address (hex)	Mnemonic	Function
ØØ	SIO	Input serial data byte D_{08} D_{07} D_{06} D_{05} D_{04} D_{03} D_{02} D_{01}
Ø1	COMSTAT	Input communication port status T_X R_X x x L x x x 1 = lost RCV data Recieve Data Ready 1 = Transmitter Ready Ø = Not Ready
Ø2	INTVECT	Input interrupt vector and display type D X X X X A A A Display Type: Ø = Parallel 1 = Serial Interrupt Device Address as Follows: 000 SIR 001 KST 010 TP 011 SOR 100 EXT φ 101 EXT 1 110 CONSOLE 111 CARRIER
Ø3	Unused	
Ø4	KST	Input keyset word 0 K_{06} K_{05} K_{04} K_{03} K_{02} K_{01} K_{00}
Ø5	TP	Input touch panel word X_{03} X_{02} X_{01} X_{00} Y_{03} Y_{02} Y_{01} Y_{00} Horizontal Position of Touch Vertical Position of Touch
Ø6-ØF	Unused	

Input Address (hex)	Mnemonic	Function
10	XL	Input lower 8 bits of x register  X_{07} X_{06} X_{05} X_{04} X_{03} X_{02} X_{01} X_{00}
11	XU	Input most significant bit (x_8) of x and the x direction flag 
12	YL	Input lower 8 bits of y register  Y_{07} Y_{06} Y_{05} Y_{04} Y_{03} Y_{02} Y_{01} Y_{00}
13	YU	Input most significant bit (y_8) of y and the y direction, long vector, and ABORT flags  x x A L x x Y_D Y_{08} ABORT FLAG 1 = ABORT 0 = NORMAL Y DIRECTION 1 = REVERSE 0 = FORWARD LONG VECTOR 1 = X 0 = Y
14-1F	Unused	
24-3F		Available for user programs

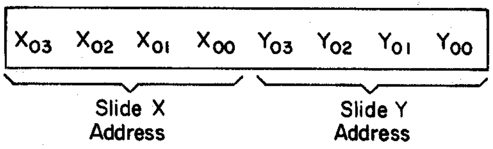
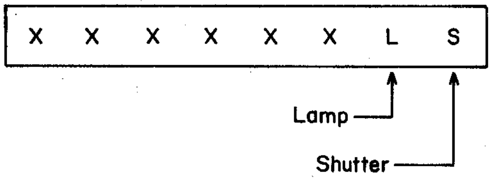
OUTPUT

Output Address (hex)	Mnemonic	Function
ØØ	SIO	Load the transmitter with the contents of A and transmit to the central computer D_{07} D_{06} D_{05} D_{04} D_{03} D_{02} D_{01} D_{00}
Ø1	COMSTAT	Load the serial IO port with the following status word. This word conditions the port to receive PLATO data. 1 1 1 0 1 1 0 0
Ø2	Unused	
Ø3	IMASK	Load the interrupt mask register with the contents of A 
Ø4-Ø7	Unused	
Ø8	XLONG	Set the long vector in the DIU to x. The contents of A are unused. See OUT CLOCKL instruction for use of this flag.
Ø9	YLONG	Set the long vector in the DIU to y. The contents of A are unused. See OUT CLOCKL instruction for use of this flag.
ØA	SETXR	Set the x direction flag in the DIU. The x register will be decremented by all subsequent clock x signals. The contents of A are unused.
ØB	SETXF	Reset (CLEAR) the x direction flag in the DIU. The x register will be incremented by all subsequent clock x signals. The contents of A are unused.

Output Address (hex)	Mnemonic	Function
ØC	SETYR	Set the y direction flag in the DIU. The y register will be decremented by all subsequent clock y signals. The contents of A are unused.
ØD	SETYF	Reset (CLEAR) the y direction flag in the DIU. The y register will be incremented by all subsequent clock y signals. The contents of A are unused.
ØE	SETABT	Set the ABORT flag. This instruction places the terminal in the ABORT mode. The contents of A are unused.
ØF	CLRABT	Reset (CLEAR) the ABORT flag. This instruction places the terminal in the normal operating mode. The contents of A are unused.
1Ø	XL	Load the lower 8 bits of the x register with the contents of A.
11	XU	Load the most significant bit (x_8) of the x register with bit $A_{\emptyset}$ of the accumulator The other bits of A are unused.
12	YL	Load the lower 8 bits of the y register.
13	YU	Load the most significant bit (y_8) of the y register with bith $A_{\emptyset}$ of the accumulator. The other bits of A are unused.
14	PDL	Load the lower 8 bits of the panel parallel data register with the contents of A.
15	PDU	Load the upper 8 bits of the panel parallel data register with the contents of A and write ($WE_{\emptyset}$=1) or erase ($WE_{\emptyset}$=4) the contents of the parallel data register (16 bits) on the display.
16	PDM	Load the DIU mode register with the lower 3 bits of A.



<table>
<tr><th>Output Address (hex)</th><th>Mnemonic</th><th>Function</th></tr>
<tr><td>17</td><td>PDLU</td><td>Load both PDL and PDU with the contents of A and write ($WE_{\emptyset}$=Ø) or erase ($WE_{\emptyset}$=1) the contents of the parallel data register on the display.
NOTE: The write (erase) operation is reversed in this operation.</td></tr>
<tr><td>18</td><td>CLOCKX</td><td>Clock the x register. The x direction flag specifies the direction; forward if reset (Ø), reverse if set (1). The contents of A are unused.</td></tr>
<tr><td>19</td><td>CLOCKY</td><td>Clock the y register. The y direction flag specifies the direction; forward if reset (Ø), reverse if set (1). The contents of A are unused.</td></tr>
<tr><td>1A</td><td>CLOCKXY</td><td>Clock both the x and y registers and write ($WE_{\emptyset}$=1) or erase ($WE_{\emptyset}$=Ø) the resulting address on the panel. The x and y direction flags specify direction of change. The contents of A are unused.</td></tr>
<tr><td>1B</td><td>CLOCKL</td><td>Clock the long vector, x or y, (as specified by the long vector flag) in the DIU if $A_{\emptyset}$=Ø; clock both the x and y registers if $A_{\emptyset}$=1. The resulting address is then written (WE_0=1) or erased ($WE_{\emptyset}$=0) on the display. The direction flags specify direction of xy change.</td></tr>
<tr><td>1C</td><td>HCHAR</td><td>Clock the y register and write (or erase) the resulting address on the panel. The contents of $A_{\emptyset}$ and the WE bits in the Panel Mode register specify the operation performed as follows:
<table>
<tr><th>$A_{\emptyset}$</th><th>WE_1</th><th>$WE_{\emptyset}$</th><th>FUNCTION</th></tr>
<tr><td>Ø</td><td>Ø</td><td>Ø</td><td>write</td></tr>
<tr><td>Ø</td><td>Ø</td><td>1</td><td>erase</td></tr>
<tr><td>Ø</td><td>1</td><td>Ø</td><td>nop</td></tr>
<tr><td>Ø</td><td>1</td><td>1</td><td>nop</td></tr>
<tr><td>1</td><td>Ø</td><td>Ø</td><td>erase</td></tr>
<tr><td>1</td><td>Ø</td><td>1</td><td>write</td></tr>
<tr><td>1</td><td>1</td><td>Ø</td><td>erase</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>write</td></tr>
</table></td></tr>
<tr><td>1D</td><td>VCHAR</td><td>Clock the x register and write (or erase) the resulting address on the panel as shown in the table for OUT HCHAR instruction.</td></tr>
</table>

Output Address (hex)	Mnemonic	Function
1E	WE	Write if $WE_{\emptyset}=1$ or erase if $WE_{\emptyset}=\emptyset$ the address specified by the contents of the x and y registers. The contents of A are unused.
1F	SCREEN	Erase the entire display. The contents of A are unused.
2Ø	SLIDEL	Load the lower 8 bits of the slide projector register with the contents of A.  X_{03} X_{02} X_{01} X_{00} Y_{03} Y_{02} Y_{01} Y_{00} Slide X Address Slide Y Address
21	SLIDEU	Load the upper 2 bits of the slide projector with the lower 2 bits of A.  Note: This address cannot be specified by an SSF instruction. This operation occurs automatically when using address 20.
22	---	Available for user defined equipment.
23	---	Available for user defined equipment.
24	---	PLATO IV Peripherals.
25	---	PLATO IV Peripherals.
26-3F	---	Available for user defined equipment.

4. IO BUS INTERFACE

4.0 IO Signal Definition

This section contains the information needed to design peripheral equipment to be attached to the external IO bus.

External devices communicate with the terminal via the IO bus shown in Figure 4.0.

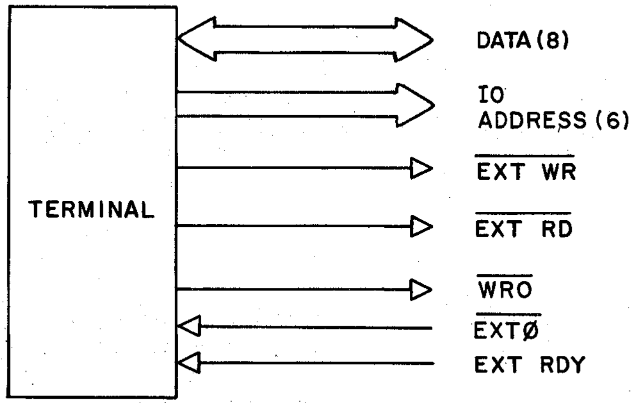


Figure 4.0. IO Bus

The function of these signals is described below. All signals are positive logic TTL levels.

DATA — 8 bits of bidirectional data.

IO ADDRESS — 6 bits of IO address specifying the external equipment to receive or send data.

$\overline{\text{EXT WR}}$ — Low level indicates that IO address is an output (WRITE) address.

$\overline{\text{EXT RD}}$ — Low level indicates that IO address is an input (READ) address.

$\overline{\text{WRO}}$ — Low level indicates data bus contains valid data to be used by external device.

$\overline{\text{EXTO}}$ — Low level indicates that a device attached to the IO bus requests service. This line should be held low until the requested service is performed.

EXT RDY

High level indicates that external equipment specified by address bus can perform requested operation. Addressed device should take this line low if it is unable to perform requested operation. Terminal will then halt until device is ready.

4.1 IO Timing

The timing diagram for the input operation is shown in Figure 4.1. All times are in nanoseconds (NS). No earlier than 90NS after placing the IO address on the address bus, the terminal issues the $\overline{\text{EXT RD}}$ signal. The selected device places the data on the bus within 350NS after receipt of the $\overline{\text{EXT RD}}$ signal. If the device cannot supply data within this time, it must take the EXT RDY line low within 50NS after receipt of the $\overline{\text{EXT RD}}$ signal or within 140NS after decoding the address.

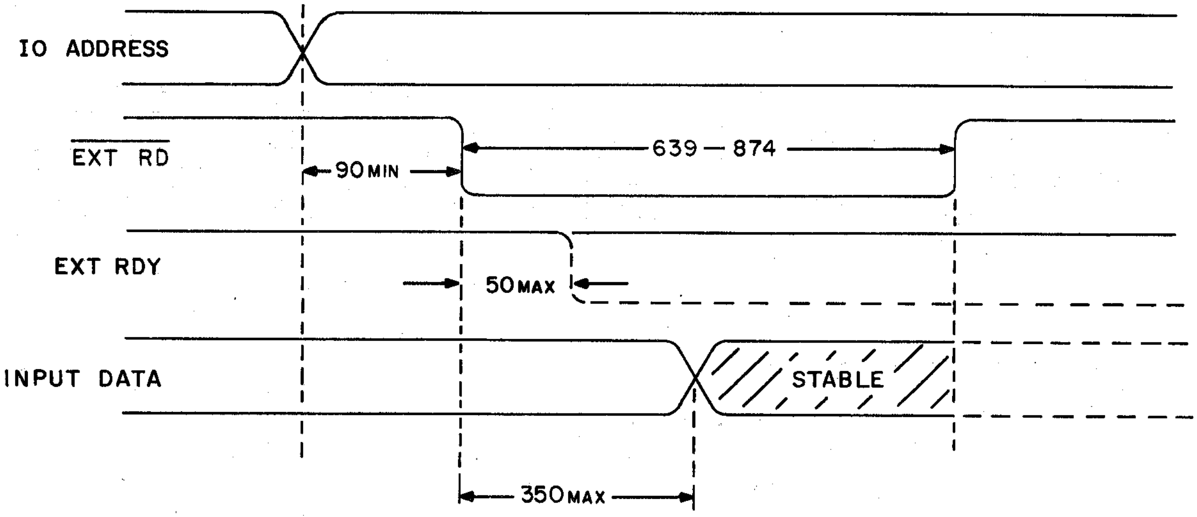


Figure 4.1. Input Timing

The terminal will then wait until the device has data ready. When ready, the selected device places the data on the bus and raises the EXT RDY line. Once issued, data must remain stable until the $\overline{\text{EXT RD}}$ is terminated by the terminal.

The timing diagram for the output operation is shown in Figure 4.2. No earlier than 90NS after placing the IO address on the address bus, the terminal issues the $\overline{\text{EXT WR}}$ signal indicating to the selected device that data is forthcoming. The terminal places the data on the data bus at least 200NS before issuing the $\overline{\text{WRO}}$ signal. The selected device can use this signal to read the data. If the selected device is not ready to receive data, it must take the EXT RDY line low within 50NS after receipt of the $\overline{\text{EXT WR}}$ signal or within 140NS after decoding the address. The terminal will still issue the $\overline{\text{WRO}}$ signal, but will halt with this signal low until the selected device raises the EXT RDY signal. After receipt of the EXT RDY signal, the terminal will hold the $\overline{\text{WRO}}$ signal and the data signals stable for 500NS before terminating the output operation.

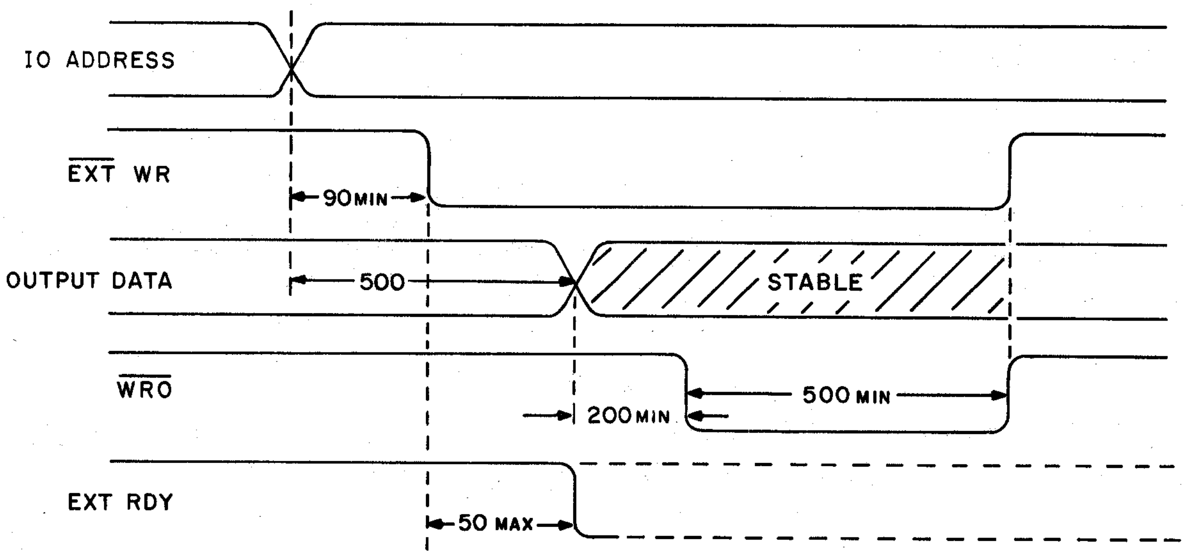


Figure 4.2. Output Timing

NOTE: The $\overline{\text{WRO}}$ signal must be "anded" with the $\overline{\text{EXT WR}}$ signal to prevent erroneous transfer of data.

4.2 External Device Specifications

All equipment designed to operate on the external bus should adhere to the following specifications:

1. All voltage levels must be TTL compatible.
2. All receivers should present no more than one standard TTL load to the bus.

   Note: Address line A_{05} is a special case. This line should contain a 3.3k pull-up resistor to +5vdc. This resistor will help insure that the equipment will operate with other PLATO terminals.
3. All transmitters must be tri-state and be capable of driving 20 TTL loads. Recommended types are National 8097 or 81LS97, Signetics 8T97, and TI74367.
4. All equipment must look at the address lines at all times and must place signals on the bus only when the equipment address appears on the bus.
5. No equipment should interfere with bus operation when the power is removed.
6. Two identical connectors should be provided on each device to permit "daisy chaining" of the bus.
7. All cables should be twisted pair type.
8. The device address should be switch selectable so that it can be easily changed later.
9. The EXT RDY line will stop the 8080 on IO operations if it is held low. Extreme care should, therefore, be exercised in its use. If possible, the equipment should be designed such that it is always ready to send or receive data. The EXT RDY line may then be left disconnected.
10. The pin assignment for the IO connector is shown in Table 7. The mating connector is ITT Cannon DBC-25s or equivalent.

Pin	Function
1	Gnd
2	Address line A_{00}
3	Address line A_{01}
4	Address line A_{02}
5	Address line A_{03}
6	Address line A_{04}

Pin	Function
7	Address line A_{05}
8	$\overline{\text{EXT WR}}$ (external write address)
9	$\overline{\text{WRO}}$ (write output)
10	$\overline{\text{EXT RD}}$ (external read address)
11	EXT RDY (external ready)
12	$\overline{\text{EXT 0}}$ (external interrupt)
13	Gnd
14	Gnd
15	D_{00} data bit 0 (lsb)
16	D_{01} data bit 1
17	D_{02} data bit 2
18	D_{03} data bit 3
19	Gnd
20	Gnd
21	D_{04} data bit 4
22	D_{05} data bit 5
23	D_{06} data bit 6
24	D_{07} data bit 7 (msb)
25	Gnd

Table 7. External IO Connector

DIAGRAMS

FUNCTIONAL BLOCK DIAGRAM

The electronics of the terminal can be partitioned into functional blocks as shown in this illustration. Most of these functions are located on the processor card. The exceptions are the modem, the keyset, and the touch panel. Detailed drawings with explanations are provided below. In the functional diagram, many details, such as bus buffering and bus control are omitted. Generally, the identified blocks correspond to drawing titles, but some functions are included with others (as the parallel interface, on the decoding drawing), and some are distributed beyond the boundaries of their specific drawings.

The central processing unit (CPU) provides the address bus; with the attendant support circuits it provides control over the data bus; and operates under control of programs stored in the read-only memory (ROM) to direct activity in the terminal. The program in the ROM, (the "resident") is fixed and is the basis for the operation of the terminal. A primary use of the random-access memory (RAM) is to provide for storage of flags, jobs, and miscellaneous data which are transitory in nature but required by the resident in performing the terminal operation. The RAM can also be used to store and execute any desired user programs. Programming details are explained elsewhere.

The RAM is dynamic, hence it requires periodic refreshing. Refreshing is accomplished by suspending normal processor operation every 2 ms and performing a read operation from each of 64 "columns" in the memory. This "refreshes" the bits in the 64 "rows" of each column, thus refreshing the entire memory. The refresh sequence is described later in more detail.

The decoding circuits specify the device which has access to the data bus. For the memories, most of the byte access decoding is done on-chip, but the chip enable signals are generated by the decoding circuitry. For the remainder of the circuitry, various strobes, latch triggers, gates, sets and clears are needed, and these are generated as appropriate by the decoding circuitry. Essentially, the decoding takes the 16 bits of the address bus, and selects 1 of 64k locations to access. More detail is provided later.

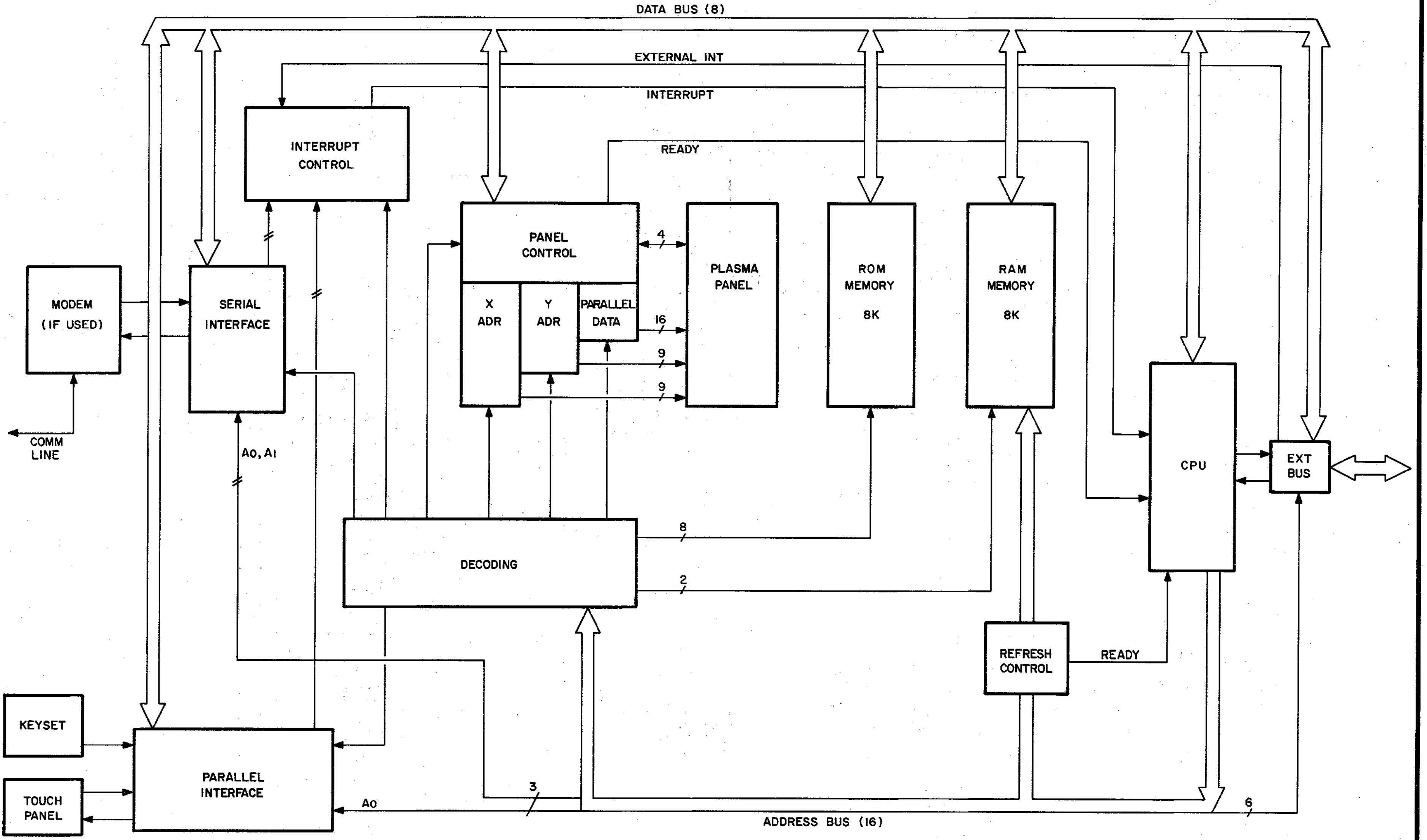
The parallel interface enables the microprocessor to conveniently access both the keyset and the touch panel. It allows each to be addressed as a separate port. In addition, it provides control signals that can be read as status by the microprocessor, or used to generate interrupts of the processor operation.

The serial interface section has been previously described (cf I.A., Terminal Architecture). It interfaces with the serial communications line that connects the terminal to its host. As with the parallel interface there are several status bits which can be read by the processor to determine the state of the interface, or alternately, the status signals can be used as appropriate to generate interrupts to the microprocessor. The central piece of the serial interface in the present terminal is a US/ART (for Universal Synchronous/Asynchronous Receiver-Transmitter), with additional circuitry for level conversion and clock generation. The present system can use any of the various modems previously used in Plato® systems, or can operate directly to either an RS232 line, or to a current-loop line.

The interrupt control logic has been previously described (cf I., General Description). This logic provides the routing information required to process interrupts. The interrupt control includes a "mask," which is used to selectively inhibit the interrupts. Not a part of the interrupt control unit, but very much associated with it, is the interrupt enabled signal from the microprocessor. This signal is not used much outside the processor, but it is used internally to prevent any interrupt request from being recognized.

The panel control circuitry, which includes the x adr, y adr, and parallel data sections, is the most complex and extensive part of the system, occupying approximately 30% of the processor card. The control section includes many features which provide greatly increased speed and simplified operation of the plasma panel. Among these features are the four panel writing modes (defined by WE_0 and WE_1), the fast drawing of lines with a very simple and fast software loop; and the extremely fast plotting of characters (over 3000 per second, if the panel is operated exclusively in the parallel mode).

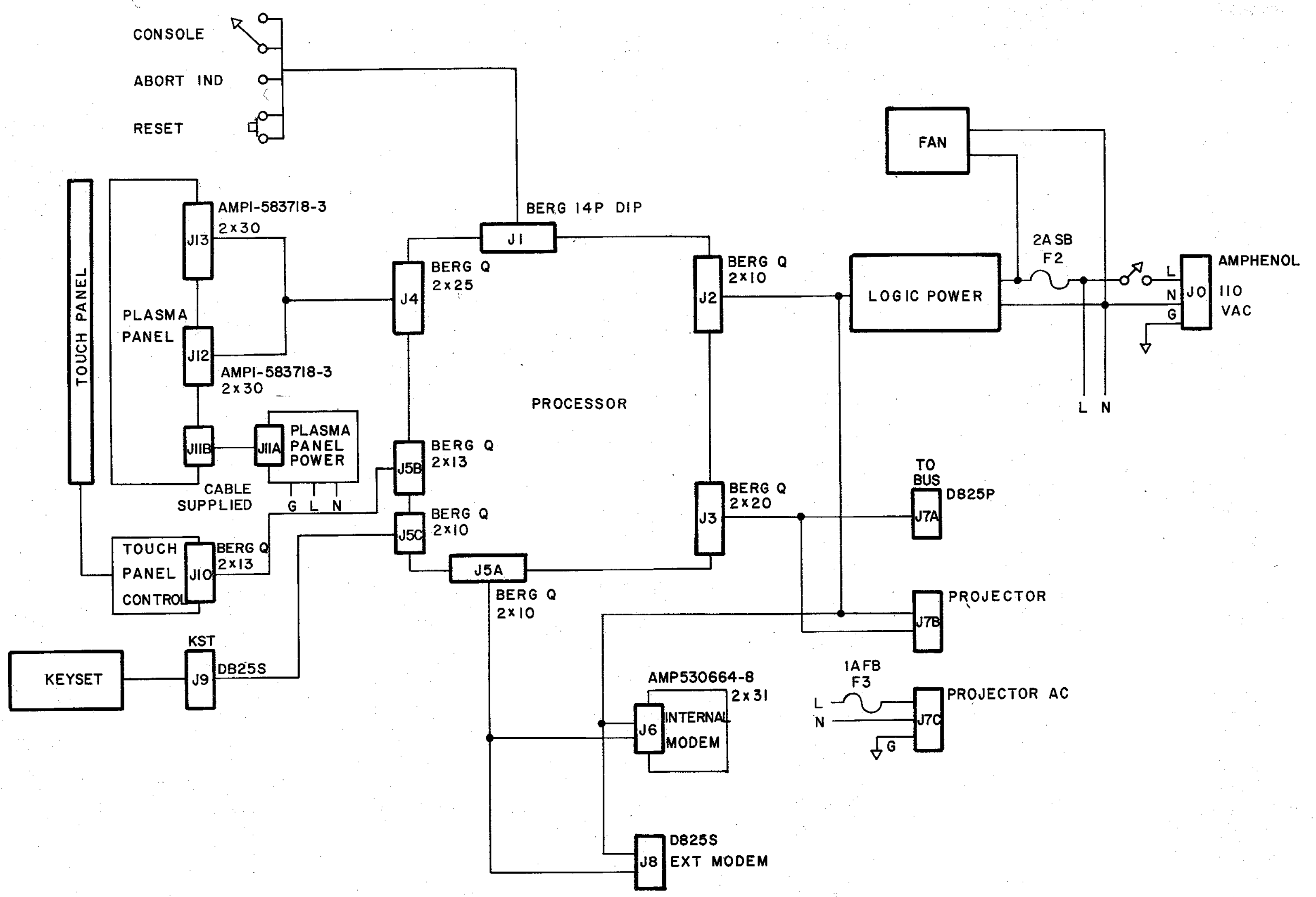
The plasma panel itself is a complex portion of the system, shown here as a simple single block. It provides an outstandingly clear presentation of characters and graphics as generated by the terminal. Its inherent self-sustaining memory characteristics make a bit map memory unnecessary, decreasing the complexity of the circuitry required to generate and provide such displays.



CARROLL Mfg Co. Champaign, IL

DES BY	DATE	TITLE PROGRAMABLE PLASMA TOUCH TERMINAL (P2T2)	
DRN BY RTL	DATE 12-28-77		
DESCRIPTION FUNCTIONAL BLOCK DIAGRAM		DWG NO 119 E-000	

REV	DATE



REV	DATE

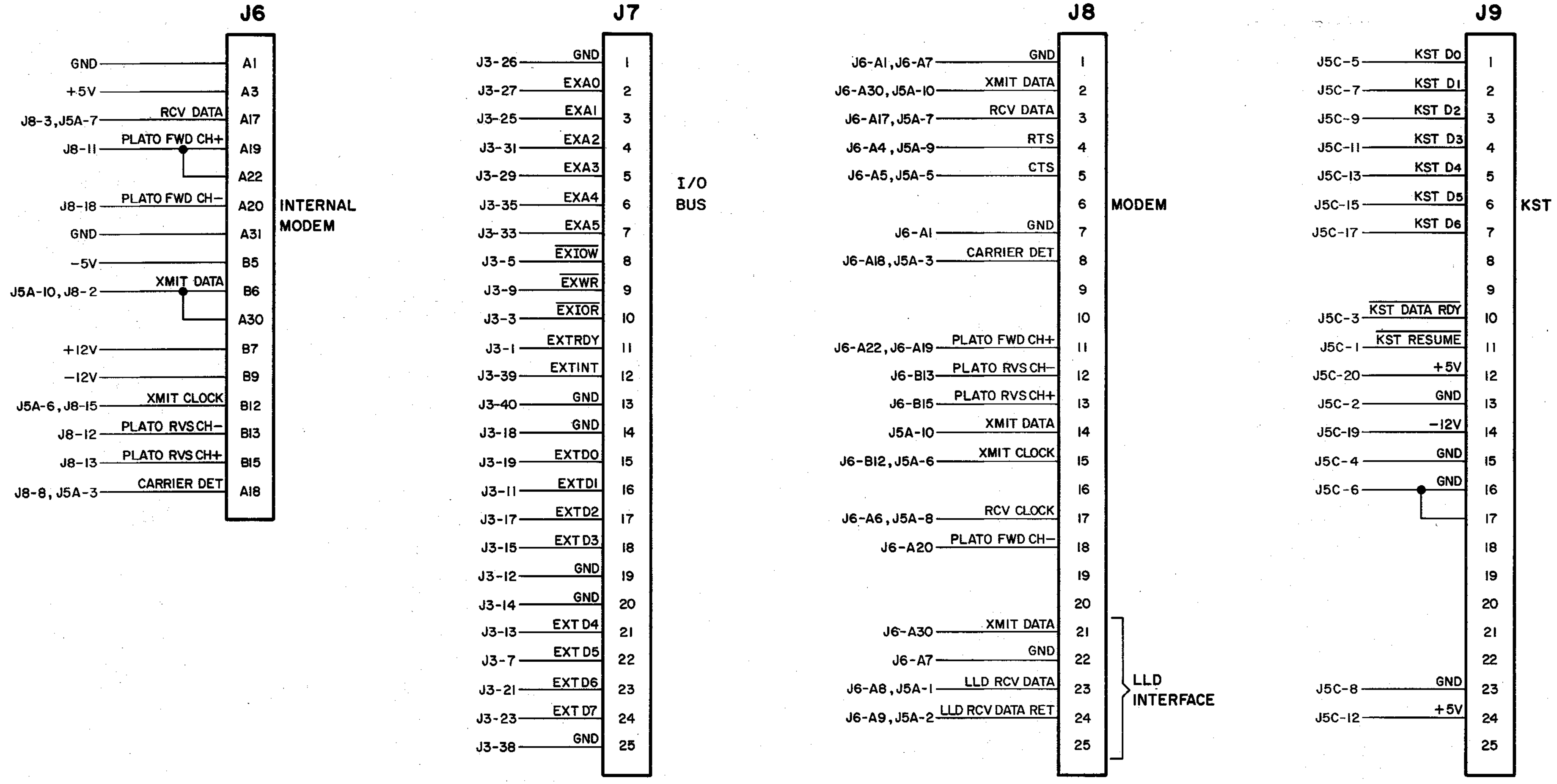
CARROLL Mfg Co. Champaign, IL

DES BY M. Gardner | DATE | TITLE PROGRAMABLE PLASMA TOUCH TERMINAL (P2T2)

DRN BY | DATE

DESCRIPTION INTERNAL WIRING

DWG NO 119 E 001



J6

INTERNAL MODEM

GND		A1
+5V		A3
J8-3, J5A-7	RCV DATA	A17
J8-11	PLATO FWD CH+	A19
		A22
J8-18	PLATO FWD CH−	A20
GND		A31
−5V		B5
J5A-10, J8-2	XMIT DATA	B6
		A30
+12V		B7
−12V		B9
J5A-6, J8-15	XMIT CLOCK	B12
J8-12	PLATO RVS CH−	B13
J8-13	PLATO RVS CH+	B15
J8-8, J5A-3	CARRIER DET	A18

J7

I/O BUS

J3-26	GND	1
J3-27	EXA0	2
J3-25	EXA1	3
J3-31	EXA2	4
J3-29	EXA3	5
J3-35	EXA4	6
J3-33	EXA5	7
J3-5	$\overline{\text{EXIOW}}$	8
J3-9	$\overline{\text{EXWR}}$	9
J3-3	$\overline{\text{EXIOR}}$	10
J3-1	EXTRDY	11
J3-39	EXTINT	12
J3-40	GND	13
J3-18	GND	14
J3-19	EXTD0	15
J3-11	EXTD1	16
J3-17	EXTD2	17
J3-15	EXT D3	18
J3-12	GND	19
J3-14	GND	20
J3-13	EXT D4	21
J3-7	EXT D5	22
J3-21	EXT D6	23
J3-23	EXT D7	24
J3-38	GND	25

J8

MODEM

J6-A1, J6-A7	GND	1
J6-A30, J5A-10	XMIT DATA	2
J6-A17, J5A-7	RCV DATA	3
J6-A4, J5A-9	RTS	4
J6-A5, J5A-5	CTS	5
		6
J6-A1	GND	7
J6-A18, J5A-3	CARRIER DET	8
		9
		10
J6-A22, J6-A19	PLATO FWD CH+	11
J6-B13	PLATO RVS CH−	12
J6-B15	PLATO RVS CH+	13
J5A-10	XMIT DATA	14
J6-B12, J5A-6	XMIT CLOCK	15
		16
J6-A6, J5A-8	RCV CLOCK	17
J6-A20	PLATO FWD CH−	18
		19
		20
J6-A30	XMIT DATA	21
J6-A7	GND	22
J6-A8, J5A-1	LLD RCV DATA	23
J6-A9, J5A-2	LLD RCV DATA RET	24
		25

LLD INTERFACE (21–25)

J9

KST

J5C-5	KST D0	1
J5C-7	KST D1	2
J5C-9	KST D2	3
J5C-11	KST D3	4
J5C-13	KST D4	5
J5C-15	KST D5	6
J5C-17	KST D6	7
		8
		9
J5C-3	$\overline{\text{KST DATA RDY}}$	10
J5C-1	$\overline{\text{KST RESUME}}$	11
J5C-20	+5V	12
J5C-2	GND	13
J5C-19	−12V	14
J5C-4	GND	15
J5C-6	GND	16
		17
		18
		19
		20
		21
		22
J5C-8	GND	23
J5C-12	+5V	24
		25

CARROLL Mfg Co. Champaign, IL

DES BY	DATE	TITLE	PLATO V TERMINAL
DRN BY RTL	DATE 12-2-77		
REV	DATE	DESCRIPTION CONNECTOR ASSIGNMENTS	DWG NO 119 E 002A

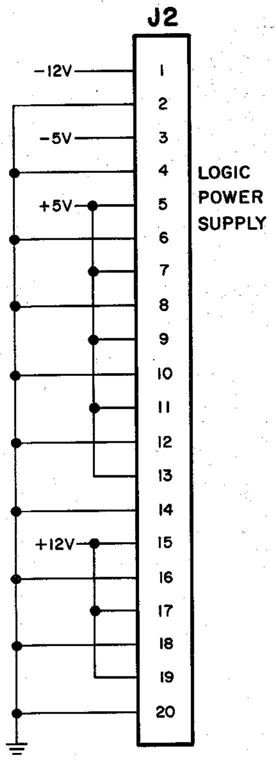
J1

FRONT PANEL

φ2 TTL	1
STSTB	2
RUN	3
	4
$\overline{\text{CONSOLE}}$	5
$\overline{\text{ABORT}}$	6
$\overline{\text{CLEAR}}$	7
GND	8
	9
	10
	11
	12
+5V	13
GND	14

J2

LOGIC POWER SUPPLY



J10

TOUCH PANEL

J5B-1	TP D0	1
J5B-3	TP D1	3
J5B-5	TP D2	5
J5B-7	TP D3	7
J5B-9	TP D4	9
J5B-11	TP D5	11
J5B-13	TP D6	13
J5B-15	TP D7	15
J5B-17	+5V	17
J5B-19	$\overline{\text{TP DATA RDY}}$	19
J5B-21	$\overline{\text{TP RESUME}}$	21
J5B-2		2
J5B-4		4
J5B-6		6
J5B-8	GND	8
J5B-10		10
J5B-12		12
J5B-14		14
J5B-16		16
J5B-20		20
J5B-24		24
* J5B-26		26

* PINS 25 & 26 NOT USED AT TOUCH PANEL END

J12

SERIAL PANEL DATA

		1
		2
		3
J4-11	PANEL RDY	4
J4-3	WRITE	5
J4-13	BULK	6
J4-5	ERASE	7
J4-9	X0	8
J4-19	X3	9
J4-7	X5	10
J4-17	X7	11
J4-1	X6	12
J4-15	X8	13
		14
		15
		16
J4-30	Y1	17
J4-20	X1	18
J4-22	X2	19
J4-31	X4	20
J4-29	Y8	21
J4-26	Y3	22
J4-23	Y4	23
J4-27	Y6	24
J4-28	Y7	25
J4-25	Y2	26
J4-21	Y5	27
J4-24	Y0	28
		29
		30
J4-2/18 EVEN	GND	D/M

J13

PARALLEL PANEL DATA

		1
		2
		3
		4
		5
		6
		7
		8
J4-34	SERIAL	9
J4-38	PD0	10
J4-33	PD1	11
J4-39	PD2	12
J4-42	PD3	13
J4-37	PD4	14
J4-35	PD5	15
J4-40	PD6	16
J4-44	PD7	17
J4-48	PD8	18
J4-50	PD9	19
J4-46	PD10	20
J4-43	PD11	21
J4-45	PD12	22
J4-47	PD13	23
J4-36	PD14	24
J4-41	PD15	25
		26
		27
		28
J4-49	PAR PANEL	K

CARROLL Mfg Co. Champaign, IL

DES BY

DATE

TITLE PLATO V TERMINAL

DRN BY RTL

DATE 12-3-77

DESCRIPTION CONNECTOR ASSIGNMENTS

DWG NO 119 E 002B

REV DATE

J3

I/O BUS

	EXTRDY	1
76-13	$\overline{EXIOR}$	3
76-11	$\overline{EXIOW}$	5
62-3	EXTD5	7
76-7	$\overline{EXWR}$	9
62-13	EXTD1	11
61-10	EXTD4	13
61-13	EXTD3	15
62-6	EXTD2	17
62-10	EXTD0	19
61-6	EXTD6	21
61-3	EXTD7	23
38-13	EXA1	25
38-3	EXA0	27
38-1	EXA3	29
38-5	EXA2	31
38-9	EXA5	33
38-7	EXA4	35
	EXTINT1*	37
	EXTINT	39
BUS	GROUND	2 THRU 40

* NOT BROUGHT TO REAR CONNECTOR (J7)

PLASMA PANEL CONNECTOR

J4 ODD

126-5	X6	1
119-12	WRITE REQ	3
120-12	ERASE REQ	5
104-2	X5	7
104-12	X0	9
118-7	PANEL RDY	11
118-12	BULK	13
126-15	X8	15
126-2	X7	17
104-7	X3	19
109-12	Y5	21
109-15	Y4	23
109-10	Y2	25
126-10	Y6	27
126-12	Y8	29
104-15	X4	31
112-15	PD1	33
112-2	PD5	35
112-5	PD4	37
112-10	PD2	39
113-10	PD15	41
129-2	PD11	43
129-5	PD12	45
129-7	PD13	47
39-10	$\overline{PARALLEL}$	49

J4 EVEN

	GROUND BUS	2
		4
		6
		8
		10
		12
		14
		16
		18
104-5	X1	20
104-10	X2	22
109-5	Y0	24
109-7	Y3	26
126-7	Y7	28
109-2	Y1	30
86-9	1.6 MHZ	32
100-8	SERIAL	34
113-15	PD14	36
112-12	PD0	38
113-12	PD6	40
112-7	PD3	42
113-7	PD7	44
129-15	PD10	46
129-12	PD8	48
129-10	PD9	50

TOUCH PANEL CONNECTOR

J5B ODD

131-18	PB0	1
131-19	PB1	3
131-20	PB2	5
131-21	PB3	7
131-22	PB4	9
131-23	PB5	11
131-24	PB6	13
131-25	PB7	15
BUS	+5V	17
131-16	$\overline{TPDRDY}$	19
86-3	$\overline{TPRSME}$	21
	NC	23
	NC	25

J5B EVEN

GROUND BUS	2
	4
	6
	8
	10
	12
	14
	16
KEY (+5V)	18
GROUND	20
	22
	24
	26

KEYSET CONNECTOR

J5C ODD

86-6	$\overline{KSTRSME}$	1
131-13	$\overline{KSTDRDY}$	3
131-4	PA0	5
131-3	PA1	7
131-2	PA2	9
131-1	PA3	11
131-40	PA4	13
131-39	PA5	15
131-37	PA6	17
	-12V	19

J5C EVEN

GROUND BUS	2
	4
	6
	8
KEY	10
+5V	12
	14
	16
	18
	20

MODEM CONNECTOR

J5A ODD

114-1	LLD	1
76-10	CARDET	3
75-7	CTS	5
75-6	RCV DTA	7
93-8	RTS	9

J5A EVEN

114-2	LLD RETURN	2
73-16	SYNC	4
75-5	XMIT CLK	6
75-4	RCV CLK	8
93-3	XMIT DTA	10

CARROLL Mfg Co. Champaign, IL

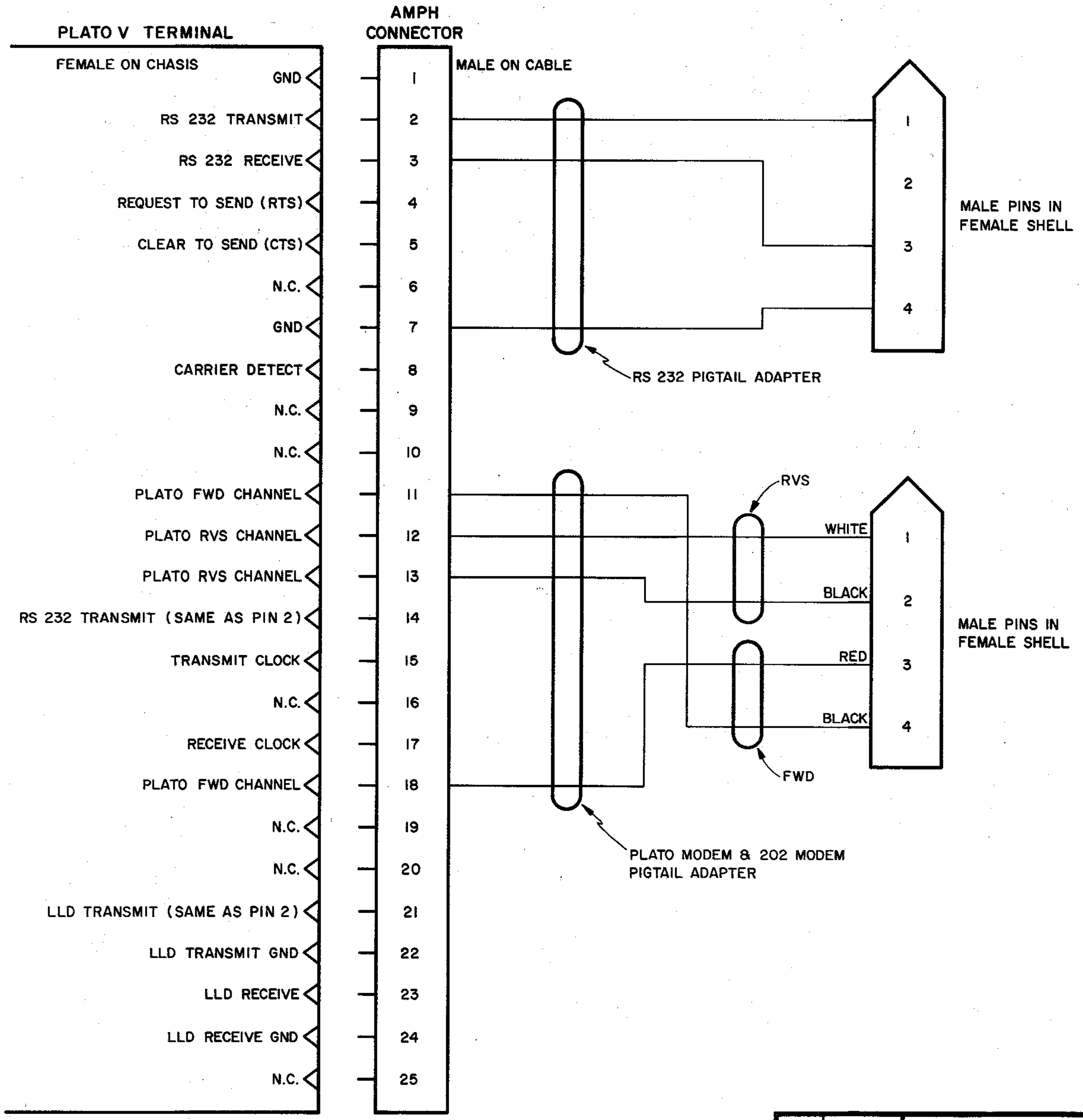
DES BY | DATE | TITLE PLATO V TERMINAL

DRN BY RTL | DATE 12-30-77

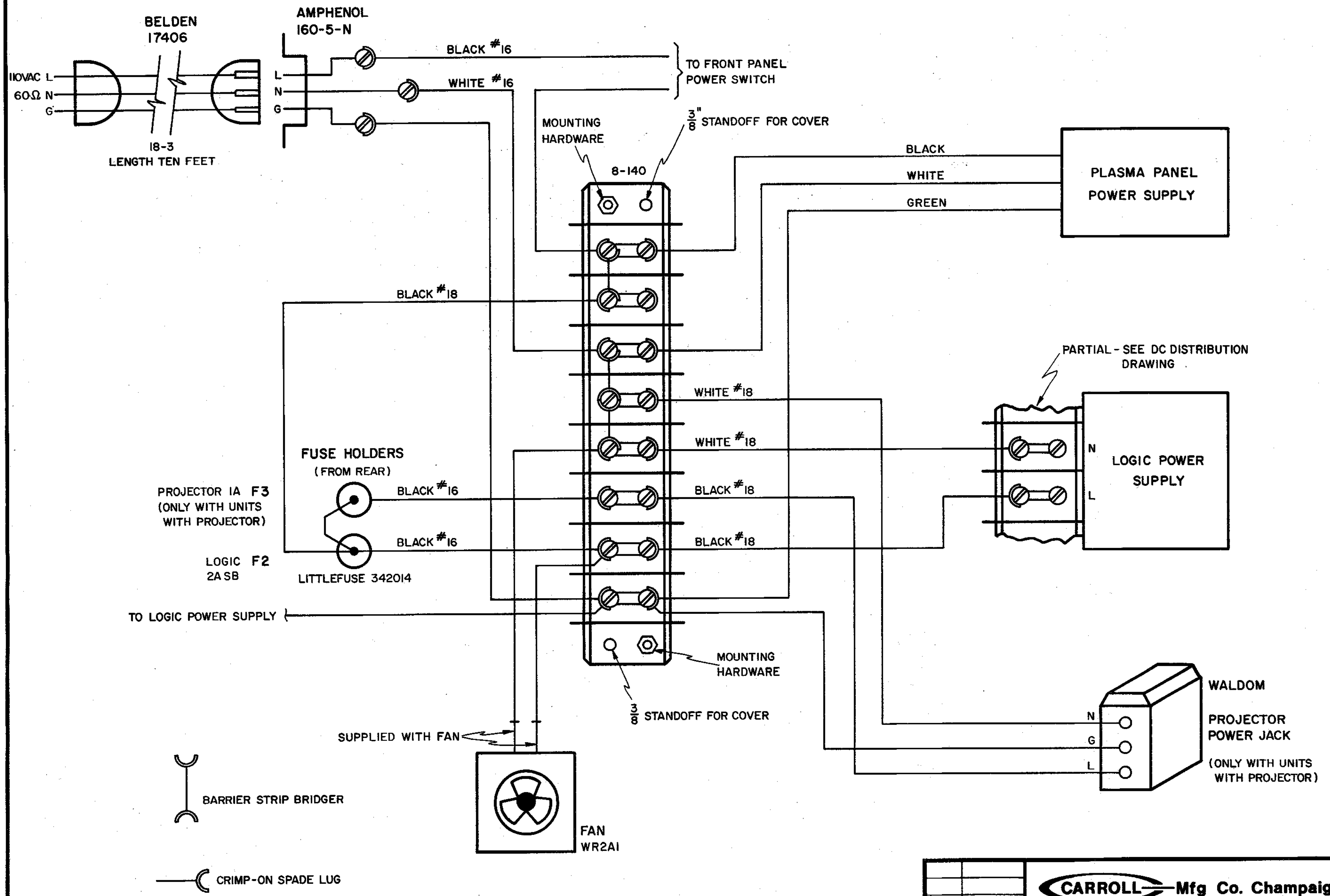
DESCRIPTION CONNECTOR ASSIGNMENTS

DWG NO 119E002C

REV | DATE

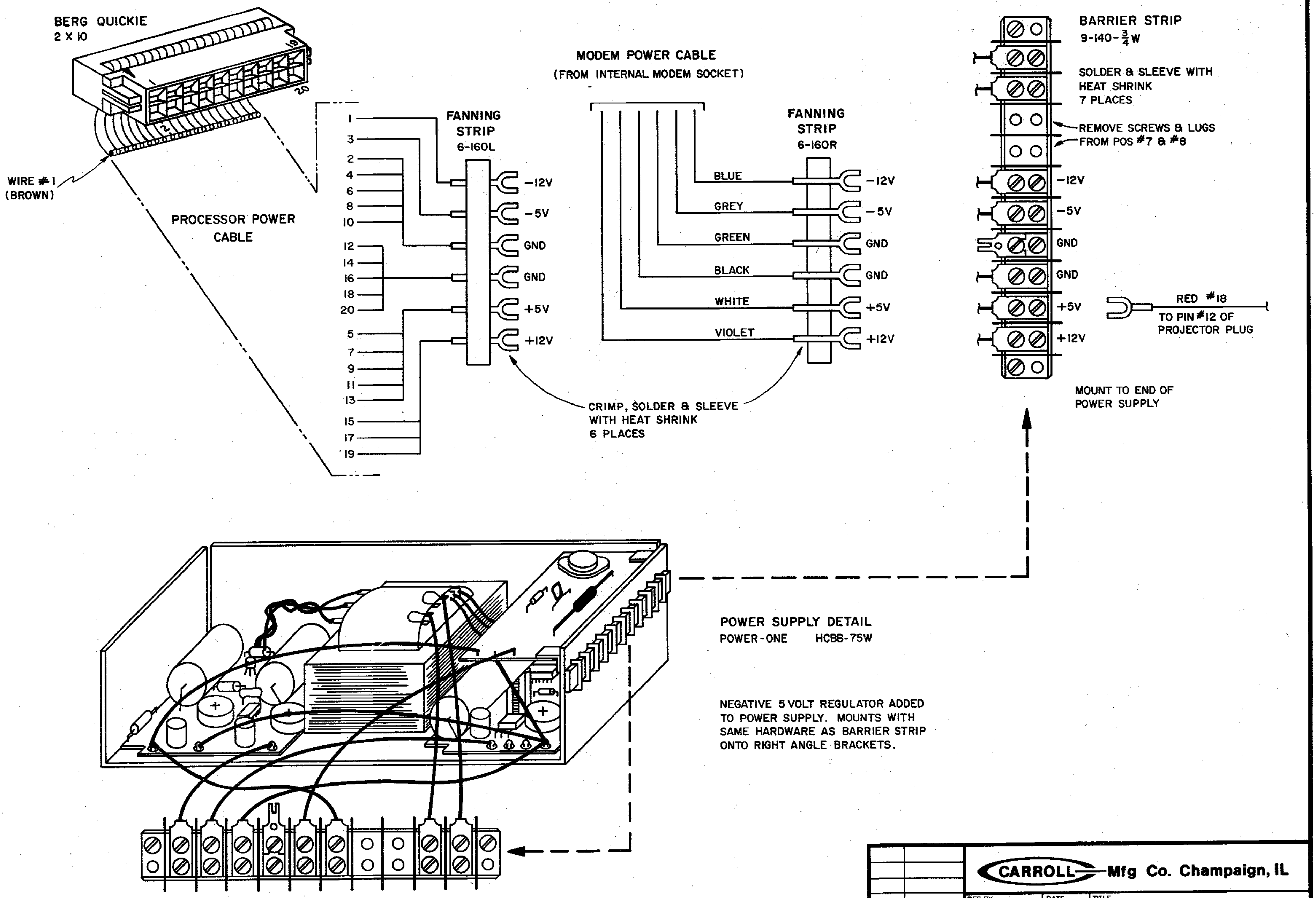


CARROLL Mfg Co. Champaign, IL				
DES BY	DATE	TITLE PLATO V TERMINAL		
DRN BY RTL	DATE 12-1-77			
DESCRIPTION LINE ADAPTERS			DWG NO 119E002C	
REV	DATE			



CARROLL Mfg Co. Champaign, IL

		DES BY	DATE	TITLE
		DRN BY RTL	DATE 12-7-77	PLATO V TERMINAL
		DESCRIPTION AC WIRING		DWG NO 119 E 003
REV	DATE			

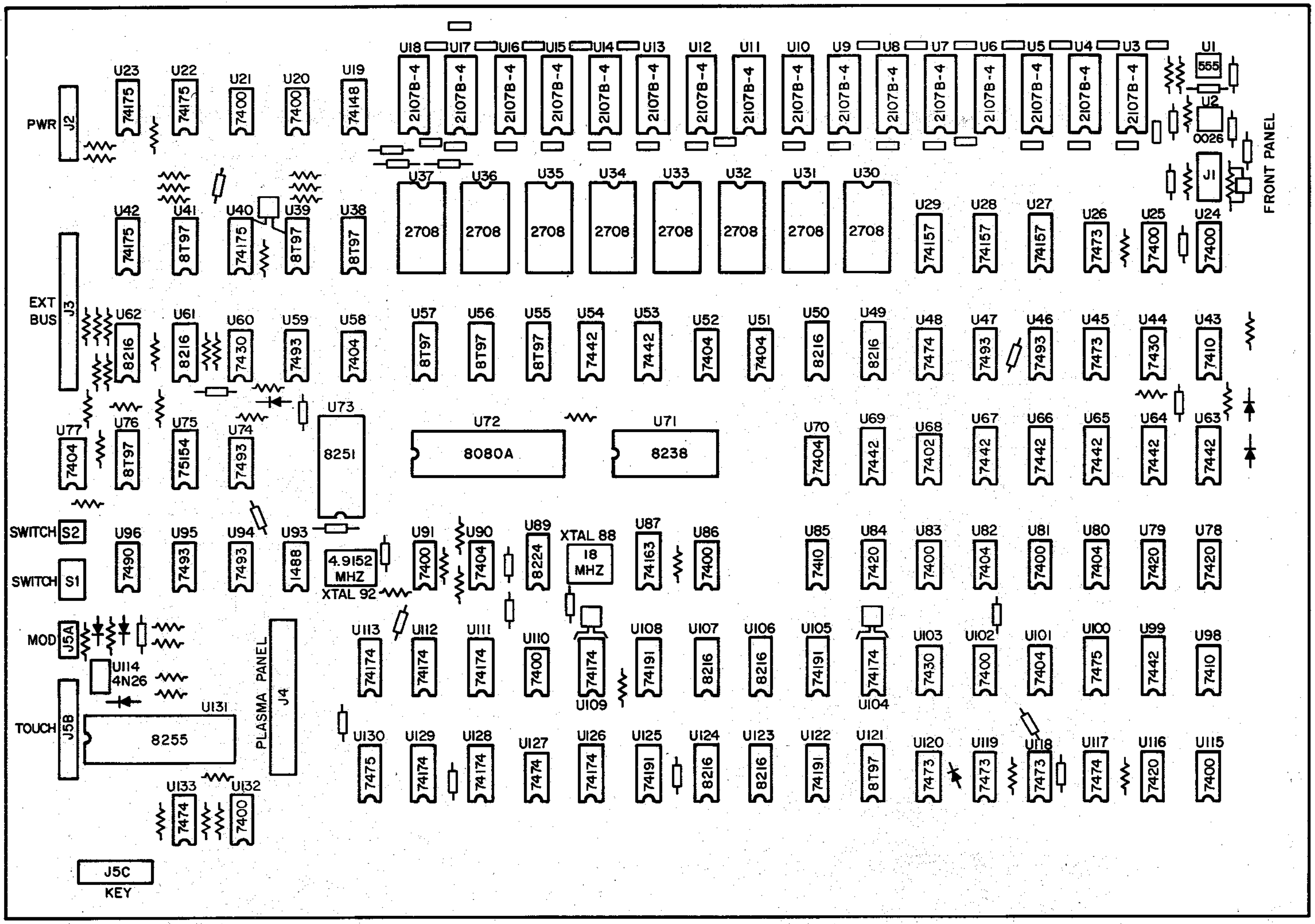


POWER SUPPLY DETAIL
POWER-ONE HCBB-75W

NEGATIVE 5 VOLT REGULATOR ADDED TO POWER SUPPLY. MOUNTS WITH SAME HARDWARE AS BARRIER STRIP ONTO RIGHT ANGLE BRACKETS.

CARROLL Mfg Co. Champaign, IL

DES BY	DATE	TITLE		
DRN BY RTL	DATE 12-13-77	PLATO V TERMINAL		
DESCRIPTION POWER DISTRUBITION			DWG NO 119E004	
REV	DATE			



CARROLL Mfg Co. Champaign, IL

DES BY | DATE | TITLE

PLATO V TERMINAL

DRN BY RTL

DATE 11-15-77

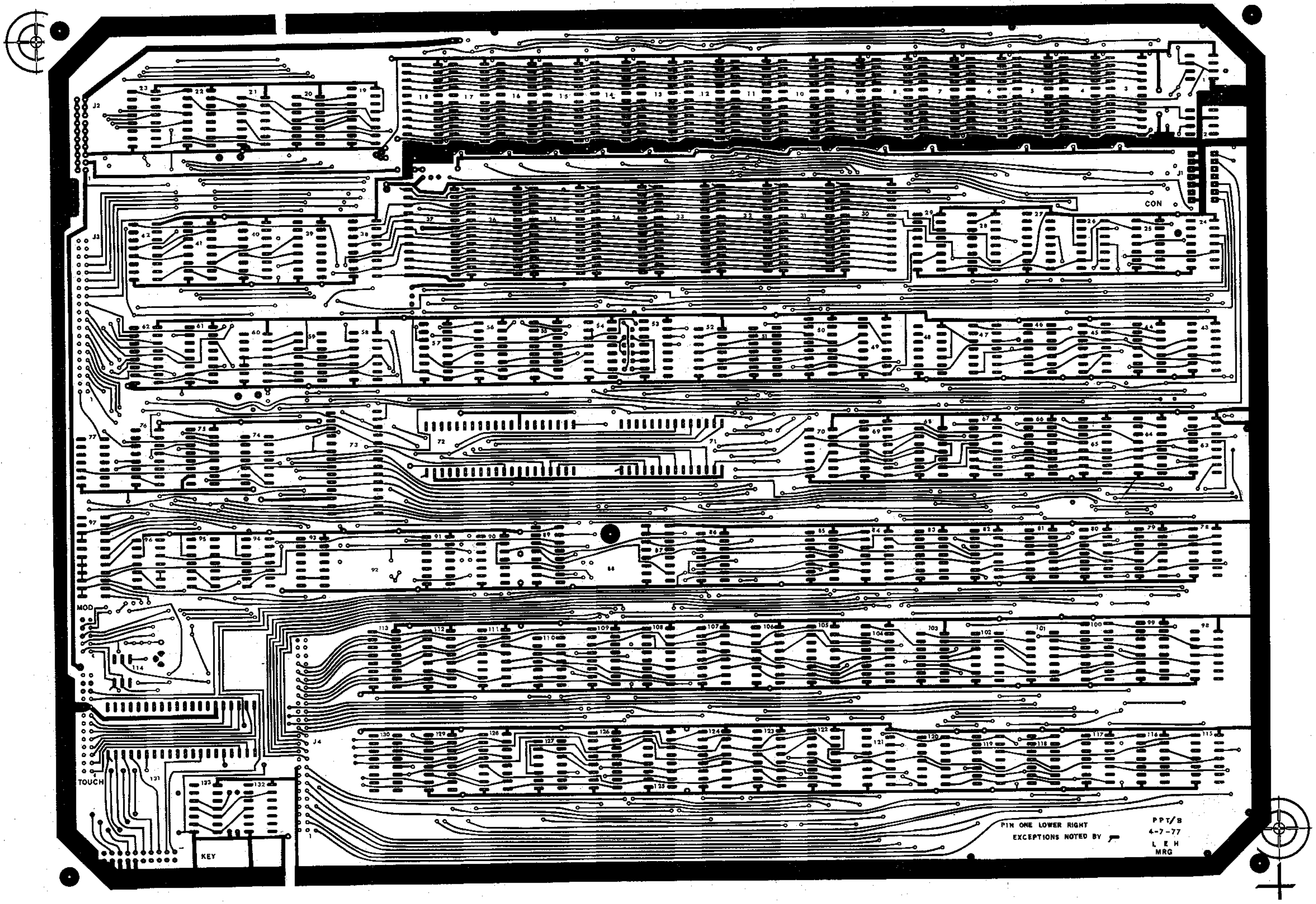
DESCRIPTION

PROCESSOR CARD LAYOUT

DWG NO

119 E 005

REV | DATE



CARROLL Mfg Co. Champaign, IL

DES BY | DATE | TITLE

DRN BY | DATE

PLATO V TERMINAL

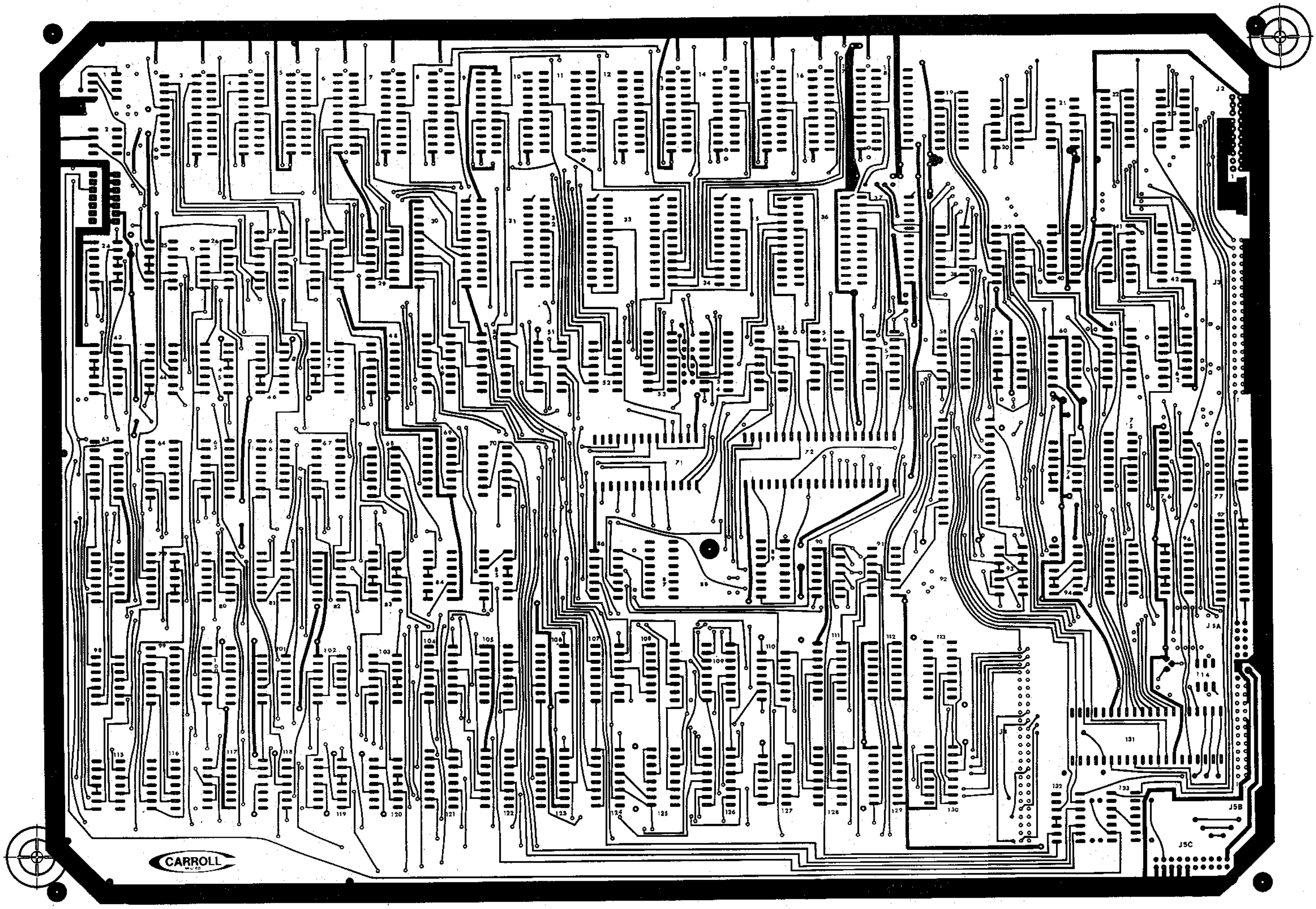
DESCRIPTION

PROCESSOR PC - FRONT

DWG NO

119 E 005A

REV | DATE



		CARROLL —Mfg Co. Champaign, IL			
		DES BY	DATE	TITLE PLATO V TERMINAL	
		DRN BY	DATE		
REV	DATE	DESCRIPTION PROCESSOR PC - REAR			DWG NO 119 E 005B

CPU

The CPU (central processing unit) contains an 8080A microprocessor, an 8238 system controller chip, an 8224 clock generator and other support circuits.

The 8080 address outputs are buffered through U55, U56, and U57 to the decoding circuits, where they are used to generate signals enabling the various ROMs, RAMs, and other functions of the processor card.

The data bus is buffered by the 8238 system controller U71, which also latches (under control of the 8080) the control signals required for operation of the memory and input/output (I/O) functions. These control signals are IOW, IOR, MEMW, and MEMR. The interrupt acknowledge input, pin 23, on the 8238, is connected to +12 v through a 1k resistor which causes the 8238 to force an RST 7 instruction on the data lines to the 8080 during an interrupt servicing sequence. This action causes the 8080 to perform a call to location 38h which is the location of the interrupt servicing program. Additional information concerning interrupt processing can be found in the description of the resident program elsewhere in this manual. Appropriate data, address, and control signals are buffered through the U61, U38, and provided at J3 as an external bus for the terminal.

Clocking signals for the microprocessor are provided by the 8224 clock generator, U89. The two clocks, Ø1 and Ø2, are high level clocks (to +12 volts) required for the proper operation of the 8080 microprocessor. The 8224 also accepts a sync signal from the 8080 and combines it with Ø1 to generate the STSTB signal. This signal occurs at the beginning of each machine cycle to initiate one of the four functions: IOW, IOR, MEMW, or MEMR. STSTB is used elsewhere on the board for control functions and is also taken off the board at J1 to permit the use of single stepping test hardware. The output from pin 6, Ø2TTL, is taken through U76 and provided to the serial interface circuitry and the panel interface circuitry. The 8224 also generates the rest and ready signals to the 8080.

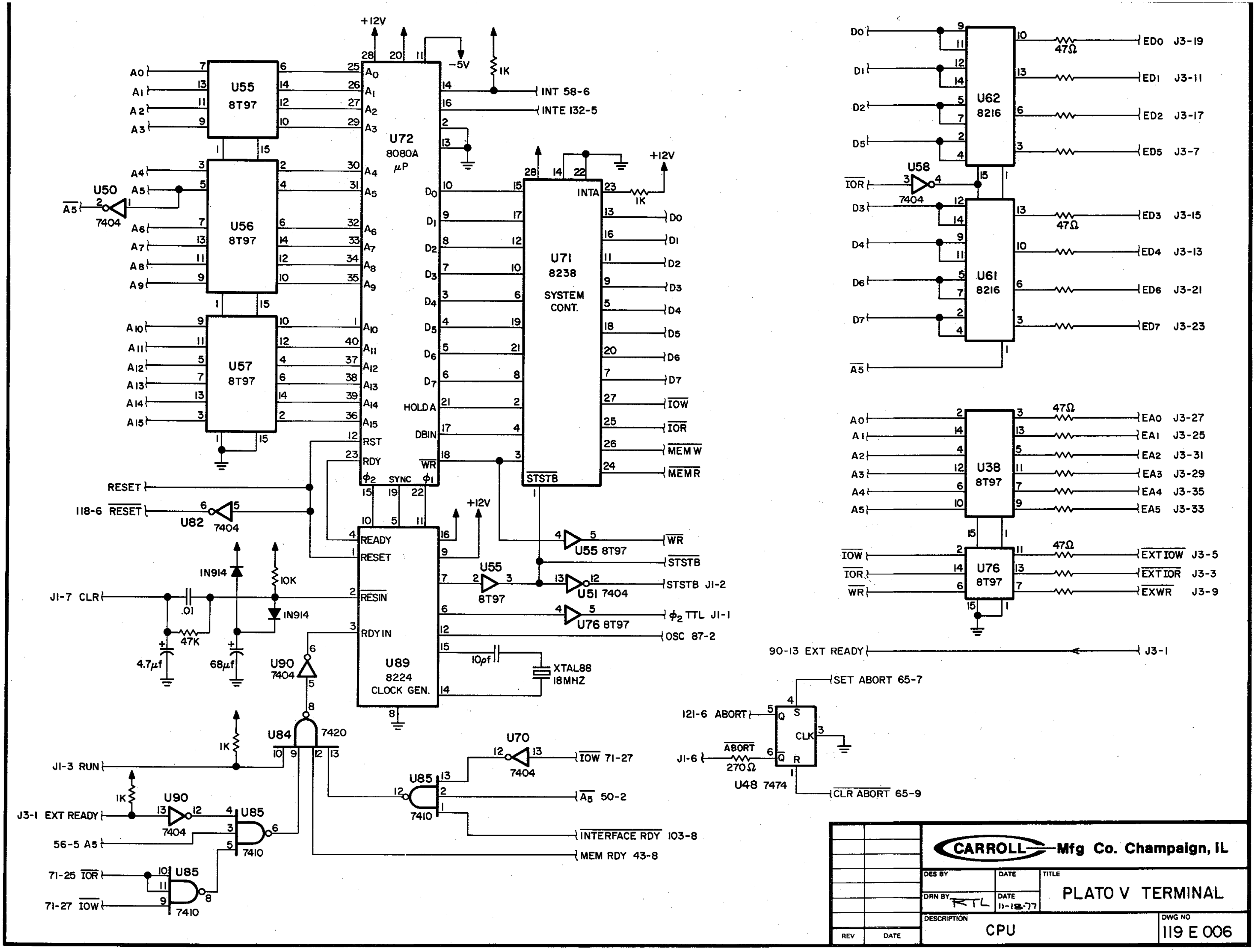
An input from the discrete circuitry at the left connected to the reset switch on the front panel, provides for a power-on reset via the 68 UFD capacitor, and for a switched reset through the filter network comprised of the .1 UFD capacitor and 47K resistor.

The ready line is used to halt the 8080 when other portions of the terminal have operations pending which must be concluded before continuing processor operation. The interface ready signal as input to U85, pin 1 is false whenever the panel interface already has an operation pending and hence is unable to accept an additional operation. If the processor attempts to write a new operation to the panel, as decoded by IOW and A5, then the processor is halted with the output from U85 operating through U84 and U90. The processor remains halted until the panel interface has completed its pending operations on the panel, at which time the processor resumes activity, completes the load of the new operation to the panel interface, and continues execution of the program.

The memory ready signal originates in the memory refresh circuitry and halts the microprocessor any time an attempt is made to read or write RAM while a memory refresh cycle is in progress. When the refresh ends, memory ready again comes true and the processor resumes activity completing the reference to RAM.

The processor can also be halted by references to the external bus when the external ready signal is false, thereby permitting the operation of devices which respond slowly. The processor can also be halted with the RUN line, an input from J1, provided to permit the use of external test equipment with the card.

The abort flip-flop provides external storage of the abort flag. In the abort mode the terminal will accept only LDM commands from the host. The $\overline{Q}$ output from U48 is used to drive a front panel LED which indicates the abort state.



CARROLL Mfg Co. Champaign, IL

DES BY	DATE
DRN BY RTL	DATE 11-18-77

TITLE PLATO V TERMINAL

DESCRIPTION CPU

DWG NO 119 E 006

REV DATE

DECODING

The decoding circuitry (shown principally on the facing page but also on others) generates control signals derived from the address present on the address bus. These control signals can specify "locations" and/or "ports" to be written into or read from.

During memory accesses, the MEMCYCLE signal at U69 pin 12 is active, locking out any decoding of I/O ports. In this case, only U53 and U54 are used, providing enable signals to the ROM and RAM. The upper address bits (A10 through A15) are decoded by these two circuits to provide for 8K of ROM starting at location 0000hex and ending at ifffhex. ROM enable signals are generated by U53, using A10, A11, and A12, together with a signal from U54 that enables U53 only when bits A13, A14, and A15 are all zero.

At location 2000hex, A13 becomes high, and U53 is disabled, and U54 becomes active. U54 has six RAM enable outputs, only two of which are used in the present terminal. These RAM enable outputs are RAM2 through RAM7, and are specified by decoding A13, A14, and A15.

Additional signals required for the operation of the memory are generated by circuits shown on the refresh control drawing. These signals include CE (chip enable) and WE (write enable). More details are found in the description of the referenced drawing.

When an I/O port is to be accessed, U69 divides the I/O space into four parts based on the value of A5, A4, and A3. When these three address lines are low, the signal OX is active, enabling U67 for further decoding based on the value of A2, A1, and A0. The 8255 Parallel Interface is also enabled by OX. For other values of A3, A4, and A5, circuits U65, U66, and U68 become active, providing control over additional ports. In the case of U68 and U69, modified forms of 2X and 3X are delivered for further decoding in the plasma panel control circuitry.

Circuit U67 decodes ports 0 through 7. Ports 0 and 1 address respectively the data and control functions of the serial interface. Ports 2 and 3 are used to load and read the interrupt vector. Port 4 reads the keyset and Port 5 reads the touch panel. (These last two are actually decoded by the parallel interface. The decoding of these ports at U67 is for the generation of resume signals, which automatically clear the keyset or the touch panel

when they are read, so as to eliminate the need for separate software action to accomplish this.) Ports 6 and 7 are additional ports of the parallel interface which should not be used, as they can change the mode of operation. See the Intel Data Manual for details.

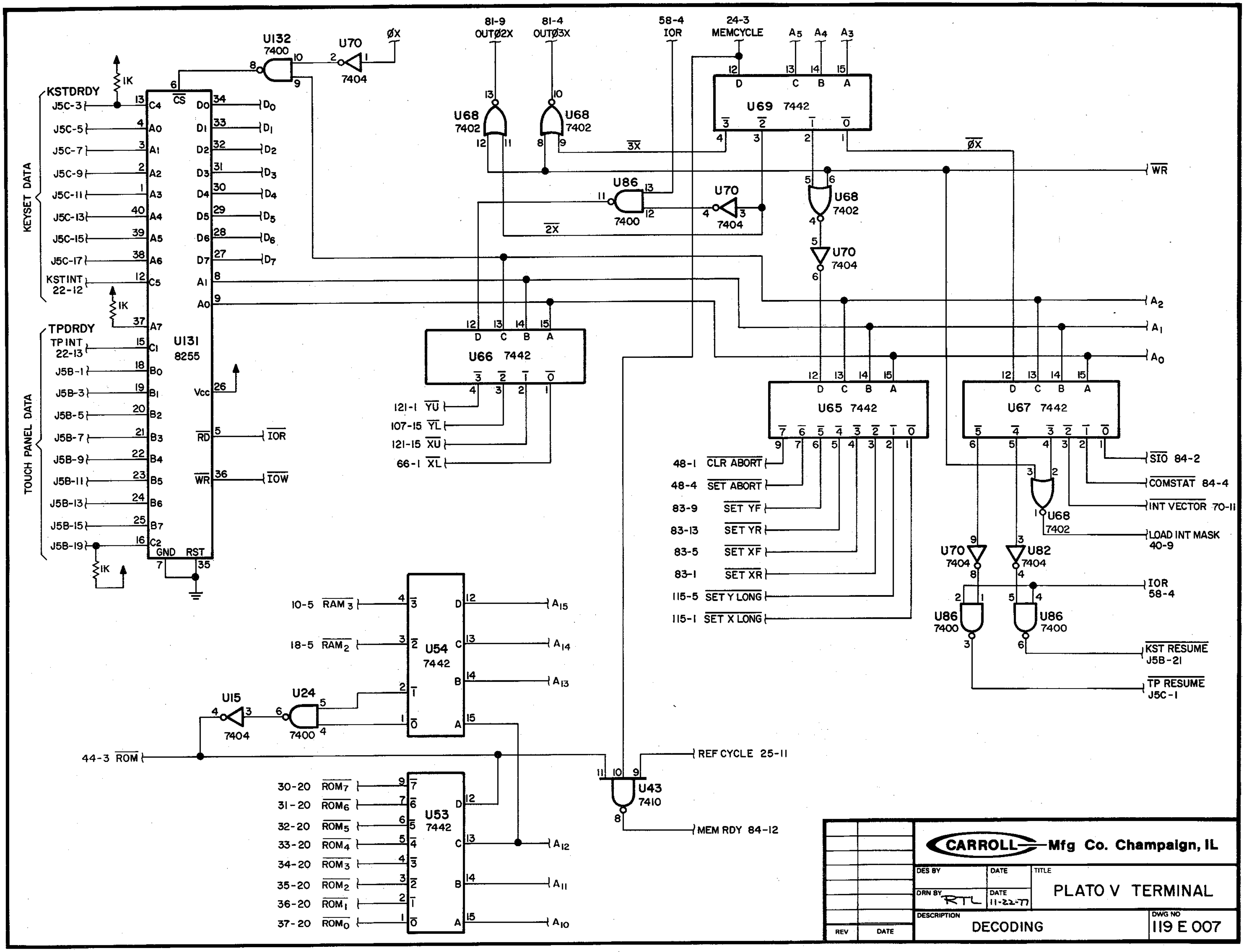
The parallel interface decodes ports 4 through 7 by being enabled by $\overline{A2}$ (note 132 pins 10-9-8), with A0 and A1 as inputs. With software, the 8255 is configured as two input ports with appropriate control signals for operating the keyset and touch panel. Port 4 is the keyset, port 5 is the touch panel, port 7 is the control port, used only by the resident. Port 6 (C) is used primarily for generating the control signals for the two devices. Though designed for interrupt control, port 6 can be read and used to indicate the data ready status of the two attached devices. (Refer to the Intel Data Manual for further details on the operation of the 8255.)

The signals corresponding to selection of ports 0 and 1 are ORed together at U84 to provide a single chip select for the US/ART. The US/ART itself then uses A0 to determine which of the two ports is being accessed.

Circuit U65 decodes ports 0 through 15. Note that 1X is ANDed with WR before being delivered to the enable input of U65, making all of these ports write-only and the outputs from U65 actually used to operate four flip-flops. The abort flag is used to indicate error conditions and drives the abort indicator on the front panel. The other three flags are used in the panel interface.

Circuit U66 decodes ports 16 through 19. Note that the enable from U69 is ANDed with IOR, making these four ports read-only. These ports are used to read the current screen address from the panel interface. Output operation of these same ports is provided by decoding in the panel interface.

Signals 2X and 3X are NANDed with WR by U68, and provided to other circuitry for the generation of write-only ports. These are discussed in more detail as they are encountered in the drawings.



CARROLL Mfg Co. Champaign, IL

DES BY	DATE	TITLE
DRN BY RTL	DATE 11-22-77	PLATO V TERMINAL

DESCRIPTION DECODING

DWG NO 119 E 007

REV DATE

INTERRUPT CONTROL

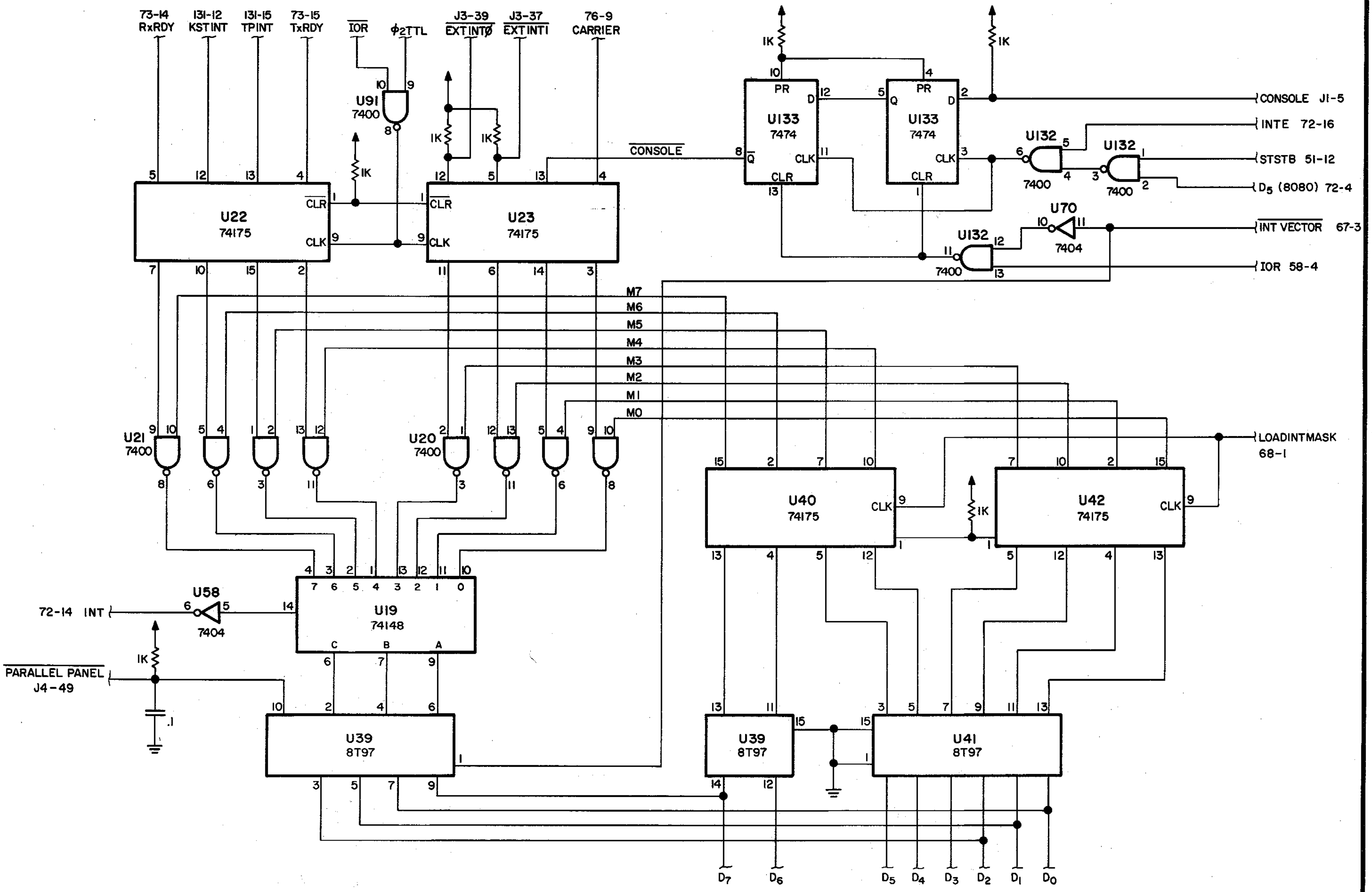
This section provides for the retention and priority encoding of the interrupt signals. A mask register is provided which permits inhibiting of undesired interrupts. Additional circuitry provides for data bus buffering of the interrupt data and mask, and also conditions the console interrupt so that its flag is set only after the instruction first following an enable-interrupt instruction.

The holdoff of the console interrupt is a special interface with the console software, allowing the single stepping modes of operation required in the console procedures. The INTE signal (interrupt enabled) inhibits all clocks to the D flip-flops making up the console-interrupt delay. The program sequence that uses this hardware is 1) enable interrupts, 2) return, 3) execute the instruction that is returned to 4) be interrupted again by the console flag. This software-hardware sequence is a result of the two step delay in U133. The enable interrupt instruction enables clocks to U133. The clock occurs when a new instruction is being fetched (indicated by D5 being high during STSTB). The first clock occurs with the return instruction, and sets pin 5 of U133. The second clock occurs with the instruction that is given control by the return, and sets pin 8 (low) of U133. This signal is latched through U23 and generates an interrupt in the normal manner.

The four interrupt signals at U23 are active low, while the four at U22 are active high. Note that the $\overline{Q}$ outputs from U23 are used, so that the latched interrupt signals at the inputs to the mask gates are all active high. A given interrupt signal is applied to a NAND gate with the associated mask bit from the mask register. If the mask bit is low, the interrupt is inhibited (blocked). If the mask bit is high, the interrupt is operated on by the priority encoder and passed to the processor as the interrupt signal, INT.

U19 is a priority encoder that generates binary coded information detailing the state of its highest priority input. If any input is low, then pin 14 is also low. This becomes the interrupt signal for the microprocessor. At the same time, the outputs C, B, and A take on the inverse of the value indicating which input is active. If pin 1, for example, is low (a TxRDY interrupt), then pins 6, 7, and 9 will show a binary 011. This is the inverse of 100, or bcd 4, which is the weight of pin 1. Similarly, an interrupt from RxRDY will cause the output of 000, from KSTINT an output of 001, etc., to the interrupt from CARRIER which causes an output of 111. If two inputs are simultaneously active, the one with the greatest weight will predominate.

The binary coded value is buffered to the data bus by U39 (together with the signal from the parallel panel, if available), and read under control of INTVECTOR. The interrupt mask is loaded directly from the data bus under control of the LOADINTMASK signal, generated by the decoding circuitry.



		CARROLL Mfg Co. Champaign, IL		
		DES BY	DATE	TITLE PLATO V TERMINAL
		DRN BY RTL	DATE 11-26-77	
REV	DATE	DESCRIPTION INTERRUPT CONTROL		DWG NO 119 E 008

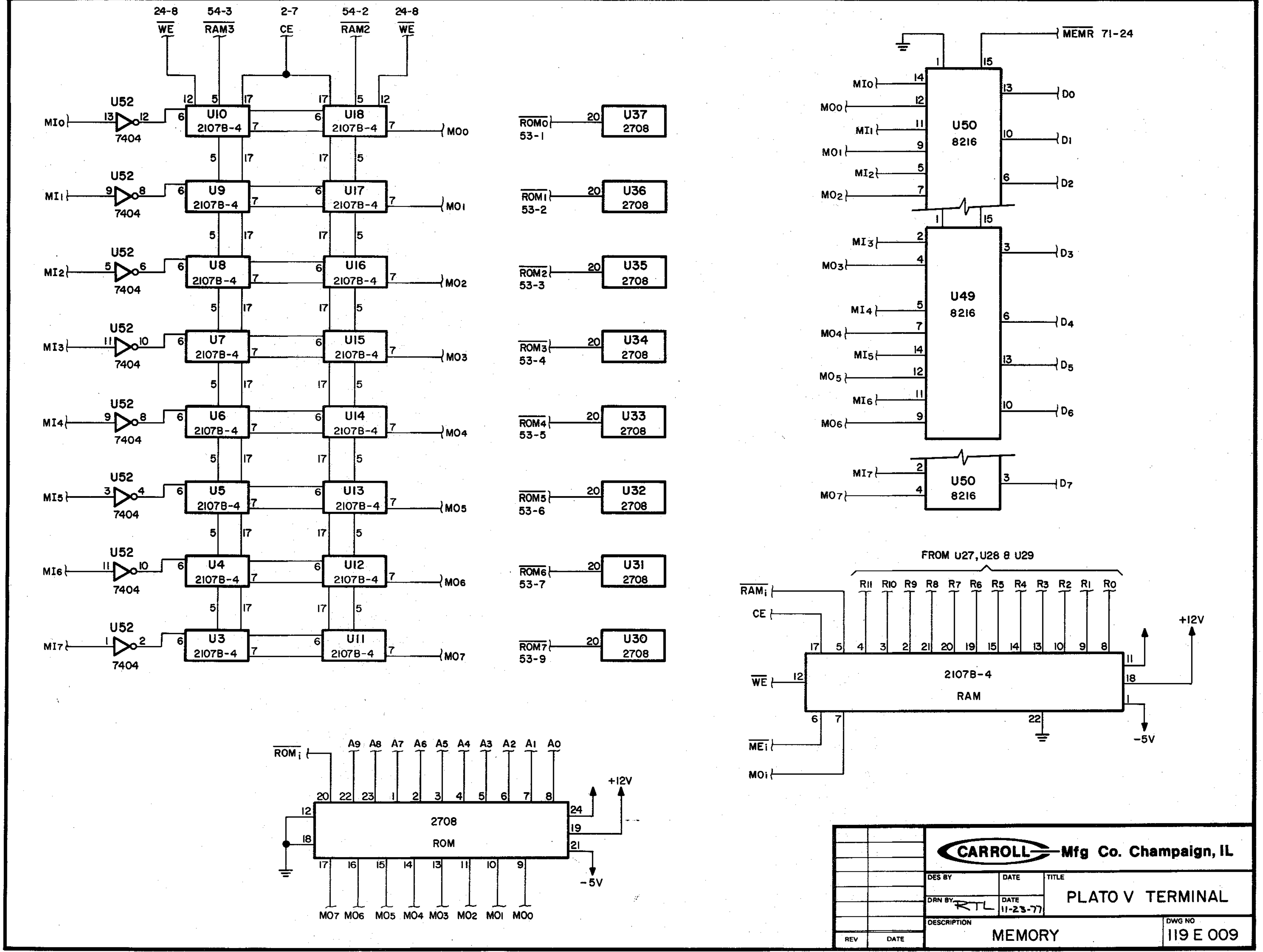
MEMORY

On this drawing, the address signals are omitted. These are bussed to all the memories and are indicated on the pinout drawings of the 2708 and the 2107B-4 shown at the bottom of the diagram. For the 2708 ROM's, the data signals are also bussed, and these are omitted also. The 2107B-4 RAM's, however, support only one data bit per chip, and these are indicated.

Address bits A0-A9 are supplied directly to the ROM's from the buffered address bus. The enable signal ROMi is provided by the decoding circuitry to select one of the eight ROMs, within which the address lines specify a single 8-bit word. The output is applied to the memory data bus as MO0-MO7. This memory data bus is buffered to the processor data bus by U49 and U50.

RAM addressing is accomplished with the address signals R0-R11, bussed to all RAMs and buffered from the processor address bus by multiplexors U27, U28, and U29 on the refresh control drawing. These multiplexors also supply the refresh address during refresh periods. Their operation will be discribed in more detail in the discussion of the refresh control circuitry.

If the memory reference is between 2000hex and 3fffhex, RAM2 or RAM3 will be active, selecting eight of the RAM chips. The chip enable signal (CE) will become active, allowing the chips to actively drive the memory data bus MO0-MO7 with the selected data. If the reference is a write, the data presented as MI0-MI7 will be written into the RAMs when $\overline{\text{WE}}$ is brought low. Note that these memories require that data be presented at their inputs in complemented form, while output data is normal polarity.



CARROLL Mfg Co. Champaign, IL

DES BY | DATE | TITLE
DRN BY RTL | DATE 11-23-77 | PLATO V TERMINAL

DESCRIPTION MEMORY

DWG NO 119 E 009

REV | DATE

REFRESH CONTROL

The refresh control circuitry performs the refresh of the dynamic RAM memories, as previously mentioned, and also prevents interference of the refresh operation with the CPU. Refresh is accomplished by supplying a "refresh address" to the memories in place of the CPU bus address.

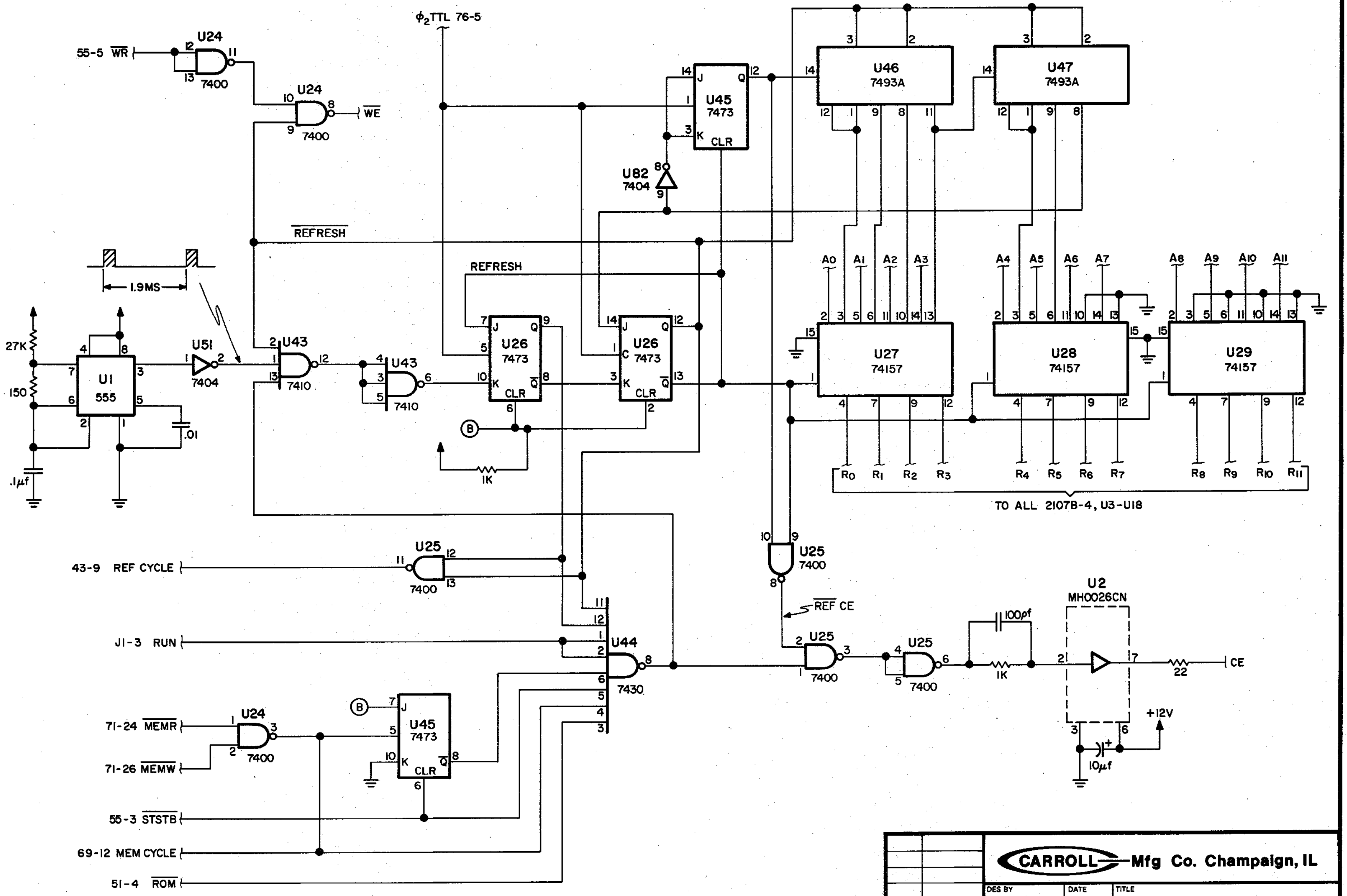
The interval between refresh cycles is determined by oscillator U1, which generates a short pulse every 1.9ms. This pulse is gated with $\overline{\text{REFRESH}}$ and a "busy" signal by U43. The gating with the refresh signal is to provide that U26 pin 9 be cleared once U26 pin 12 is set. The "busy" signal is generated by the multiple-input NAND gate U44, and prevents initiation of a refresh cycle during pending accesses to the RAM.

Once a request has been successfully requested by the assertion of K at pin 10 of U26, the two flip-flops synchronize the refresh cycle with the operation of the processor. Ø2TTL is used as clock to provide the synchronization. REFRESH is set active by the clock, and cleared by the same clock after the J input has been asserted by the counter output from pin 8 of U47. This counter output becomes active after 64 counts have accumulated in counters U46 and U47. Note that this same counter output is fed back to disable the counter trigger at U45. The refresh period thus contains 64 counts, the binary equivalents of which are applied as addresses to the RAM's by the multiplexors U27, U20, and U29.

The multiplexors are switched by the REFRESH signal, so that the processor address signals (A0-A11) are inhibited from reaching the memories, and the refresh address (from counters U46 and U47) is applied instead. The outputs from the multiplexers, R0-R11, are the address signals applied to the memory chips. The blocks of memory are chosen using A12-A15 as described under "decoding."

During half of each refresh address count time, a chip-enable ($\overline{\text{REFCE}}$) is generated by NAND gate U25 pins 9, 10, 8 and passed to the chips via the CE circuitry by U25 pins 1, 2, 3. This extra chip enable is generated only during refresh periods.

The U24 NAND gates in the upper left hand corner of the diagram inhibit the write-enable signal during the refresh period. This inhibiting occurs if the processor attempts a write to memory after a refresh cycle has started and results in the halting of the processor until the refresh cycle is concluded.



CARROLL Mfg Co. Champaign, IL

DES BY	DATE	TITLE PLATO V TERMINAL
DRN BY RTL	DATE 11-21-77	
DESCRIPTION REFRESH CONTROL		DWG NO 119 E 010

REV	DATE

SERIAL INTERFACE

The serial interface provides a serial communications link to a host processor. Provisions are included for the conditioning of both the received and transmitted signals and for the generation of appropriate synchronized clocks. Two sets of miniature toggle switches specify the mode of operation of the serial interface.

The heart of the interface is an 8251 US/ART (Universal Synchronous/ Asynchronous Receiver-Transmitter). This device is connected directly to the data bus. It has loadable command words which configure it for the reception and transmission of serial data under direct control of the microprocessor. Command words which specify the operating mode of the US/ART are loaded via the data bus and the four control signals A0, IOW, SIO, and COMSTAT. For Plato® operation the device is configured as a synchronous transmitter and receiver, and for ASCII operation it is configured as an asynchronous transmitter and receiver.

The US/ART has the capability of interrupting the processor to request service by issuing either the RxRDY (receiver ready) or TxRDY (transmitter ready) signal. These signals are supplied to the interrupt control unit where they are processed according to the priority established by that circuit. The status of the US/ART can also be read directly as a data word from the data bus under the control of A0 and IOR.

The two signals SIO and COMSTAT are active when the ports 1 and 0 respectively are addressed by the microprocessor. A0 is used by the US/ART to distinguish between control and data words.

When A0 is low (port 0 is being addressed) the SIO port is specified causing the US/ART to interpret the word on the bus as data.

If A0 is high (port 1 is being addressed) the COMSTAT port is specified causing the US/ART to interpret the word on the bus as control of status information.

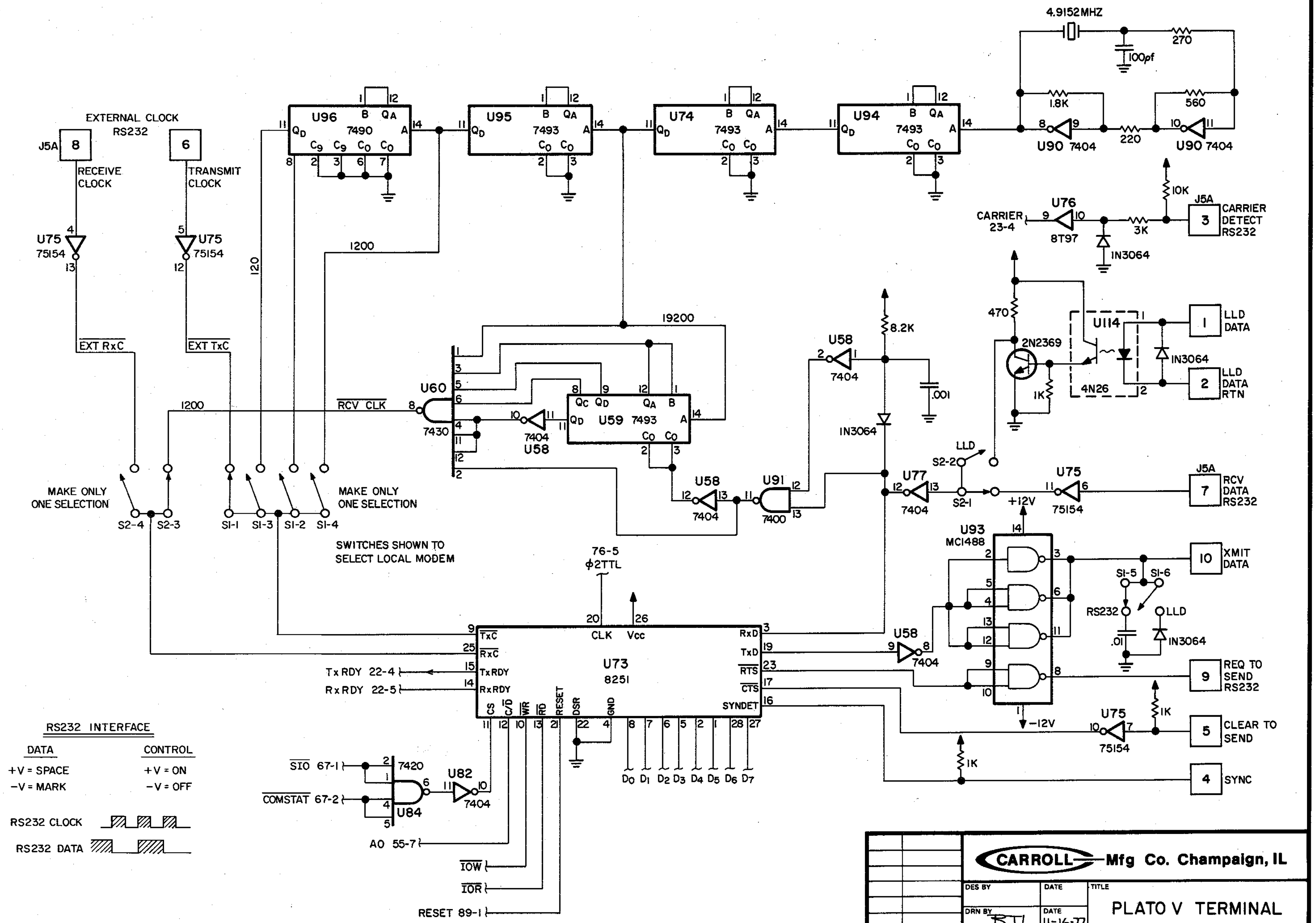
The 4.9152 MHz oscillator in the upper right corner of the diagram is the master clock used to generate the baud rate signals. The output from this oscillator is divided by U94, U95, and U96 to the various required baud rate frequencies. The 19.2 KHz signal is taken from the output of U74 and applied to the logic consisting of counter U59 and NAND gate U60. Another input to this logic is a one-shot made of U91

required baud rate frequencies. The 19.2 KHz signal is taken from the output of U74 and applied to the logic consisting of counter U59 and NAND gate U60. Another input to this logic is a one-shot made of U91

and inverter U58 together with the 8.2k resistor, .001 mfd capacitor, and IN3064. This one-shot delivers a reset pulse at every positive-going transition in the data stream. With this data-generated synchronizing pulse, the counter/NAND structure will generate a clock pulse at the output of U60 which will occur near the center of every received data bit. This is a self-synchronizing clock used for the reception of synchronous data, typically from a Plato® host. External receive and transmit clocks may be provided at RS232 signal levels on the inputs to U75 and routed through the switches to the US/ART.

The received data arrives either as an RS232 signal at the input of U75 or as a current loop through optical isolator U114. The received serial data is inverted by U77 and applied to the receive-data input, pin 3, of the US/ART and to the synchronizing one-shot at U91. Transmitted data leaves the US/ART at the transmit-data output, pin 19, is inverted by U558, and applied to line driver chip U93. Several of the drivers in this chip are tied together to provide a limited 30 ma current loop output. In the RS232 mode, this chip will drive the load to the required limits of ± 12 volts.

Additional signals which are provided which are typically used in interfacing to dial-up modems of various types. There are two outputs, request-to-send and sync, and two inputs, clear-to-send and carrier-detect. The carrier detect signal is unique among these in that it generates an interrupt which halts the normal processing of the terminal when no carrier is detected, if that signal is made available from the modem.



REV	DATE

CARROLL Mfg Co. Champaign, IL

DES BY | DATE | TITLE

DRN BY RTL | DATE 11-16-77 | PLATO V TERMINAL

DESCRIPTION SERIAL INTERFACE

DWG NO 119 E 011

PLASMA PANEL CONTROL

This drawing shows most of the panel control circuitry, including decoding of principal control signals. The exceptions are the x and y direction control flip-flops, which are shown on the respective xadr and yadr drawings, and several control signals which are generated on the subsequent three drawings.

All the remaining decoding of the panel control signals is done by U62 and U63 in the upper left corner of the diagram. The enables for these are gated through U81 (with INTERFACE READY, to prevent a change in control status while a job is pending). The enables are 02x and 03x, the decoded enable signals from the decoding circuitry. Recall that these have been formed by gating with WR, so that the outputs from U62 and U63 can be active only during an IOW operation. Once one of the panel control decoders is enabled, which of its outputs is active (low) is determined by the low order address bits A0, A1, and A2.

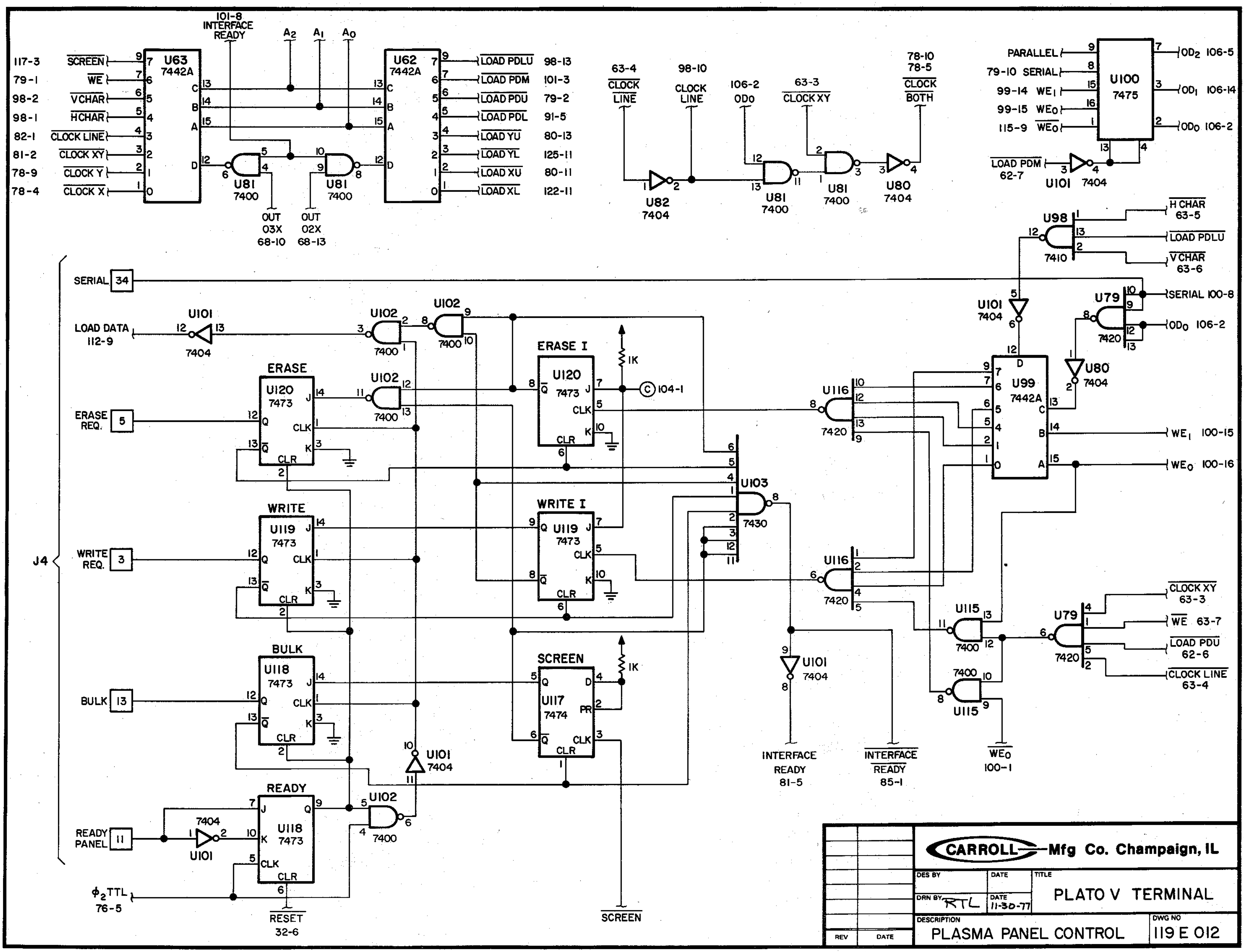
Each of the signals from these two decoders has a load or count function. The x and y direction, and the long vector, are set with control signals from U65. SCREEN is an easy one of these to describe first, and will serve as an introduction to the panel command register, shown below the decoding. The LDPDM, or load panel data mode, is also easy to describe, and will be subsequently presented before descriptions of the remaining signals and circuitry.

The SCREEN signal from pin 9 of U63 is supplied to the screen flip-flop U117 at pin 3, setting pin 5 high when a bulk erase has been requested by the program. The output of this flip-flop is applied to the J input of the bulk flip-flop, and the inverted output is applied to NAND gate U102 at pin 13, causing a high signal to be applied to the erase flip-flop at its J input, U120 pin 14. When the panel is ready, U118 pin 9 will go high, allowing 02TTL to pass NAND gate U102 pin 6, clocking the other three panel control flip-flops. The erase and bulk flip-flops will be set, since their J inputs are held high by the screen flip-flop. As soon as they are set, their $\overline{Q}$ outputs will go low, feeding back to clear the screen flip-flop (and also the erase flip-flop, if it was set). The output from the erase and bulk flip-flops is presented via J4 to the plasma panel, which will presently acknowledge the request and set PANEL READY low, internally latching the request. This will cause the ready flip-flop to change state (synchronously with Ø2TTL) inhibiting further clocking outputs from NAND gate U102 pin 6, and also clearing the request flip-flops.

At this moment the panel interface is empty, and a new request can be written into the command input registers (erasei, writei, and screen) even tough the panel is currently busy with the just-accepted task. If a new request is output, the input flip-flops will be set waiting the panel ready condition, at which time the request will be, as described above, transferred to the request flip-flops until the panel accepts the request and clears them. During any time that a panel request is pending, some $\overline{Q}$ output will be low at an input to the multiple-input NAND gate U103, thereby making the INTERFACE READY signal go false. If during this time, the processor attempts another write to the panel interface, the processor will be halted by the ready circuitry previously described in the CPU section.

The LDPDM (Load Panel Data Mode) signal causes the loading of the panel mode register shown in the upper right hand corner of the drawing. The inputs OD0, OD1, and OD2, are buffered directly from the data bus. The outputs establish the mode of operation of the panel interface, and their action on other control signals will be described below. WE_0 and WE_1 specify panel write or erase operation, and SERIAL and PARALLEL (inverses) specify serial or parallel operation. In the case of parallel operation the parallel data register contains the data to be written (or erased) on the panel.

Circuits U80, U81, and U82, are used principally while drawing or erasing lines on the panel. The software calculates and sets x or y as the long vector. Then, to draw the line, it merely issues outputs on CLOCKLINE. This causes whichever counter is chosen as the long vector to be incremented (or decremented, depending on the setting of the direction flip-flops), and simultaneously requests panel operation for a write or an erase as appropriate. Whenever it is time to step the short vector as well, the least significant data bit, OD0 at pin 12 of U81, is set high during the IOW, thereby generating a CLOCKBOTH pulse,



CARROLL Mfg Co. Champaign, IL

REV	DATE

DES BY | DATE

DRN BY RTL | DATE 11-30-77

TITLE PLATO V TERMINAL

DESCRIPTION PLASMA PANEL CONTROL

DWG NO 119 E 012

which clocks both the short and long vectors simultaneously. Most of the time, the long vector is the only one being clocked, and ODO is low during the output time. When a request is made to clock both, namely the signal CLOCKXY, then this passes through NAND gate U81 and generates a CLOCKBOTH signal just as did CLOCKLINE with ODO set.

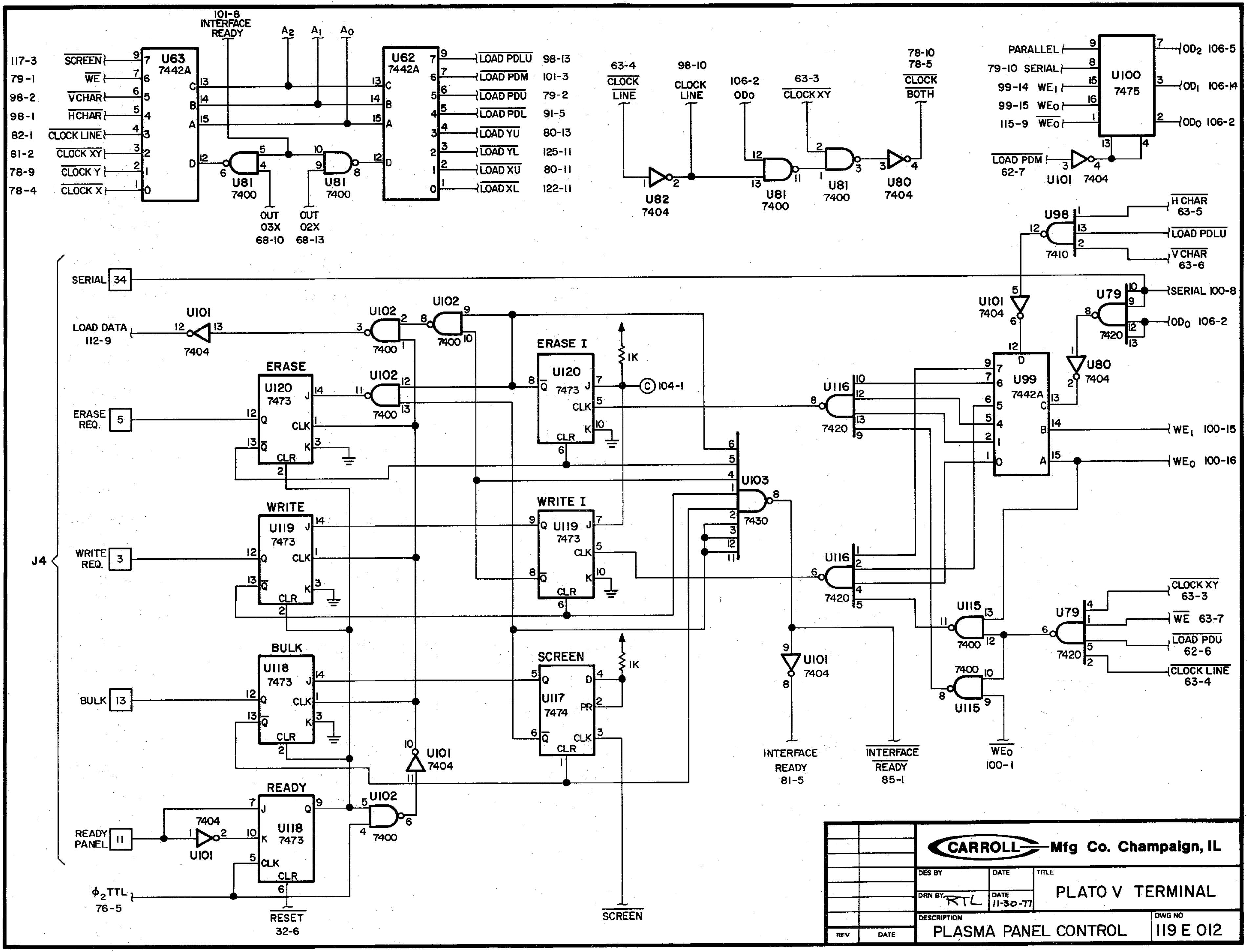
The remainder of the control circuitry, on the right, routes the control signals to cause either a write or an erase depending on the settings of WE_0 and WE_1. To get started, consider the four-input NAND gate U79 at the lower left of the drawing. Its output is active (high) during an IOW to any of its four inputs. This output is then routed to one of the two four-input NAND gates depending on whether WE_0 is set or clear. If WE_0 is set, then the IOW pulse is routed to set the write flip-flop, and if WE_1 is clear, then it is routed to the erase flip-flop. In this way, outputs to CLOCKXY, WE, LOADPDU, and CLOCKLINE can automatically and simultaneously cause a write or erase depending on the setting of WE_0.

Inputs to U99 are similarly routed by both WE_0 and WE_1 to automatically cause a write or erase as required. This is the structure used when the resident is plotting characters on the plasma panel. The activating input to U99 is at its D input, pin 12. This is provided by any of the three inputs to U98 at pins 1, 2, or 13. The other three inputs to U99 serve to route the input to the appropriate output depending on D0, WE_0 and WE_1. D0 is provided from the character information in memory, and WE_0 and WE_1 define the mode, or, how the character is to be plotted.

When WE_0 and WE_1 are both low, output 0 (pin 1) will be active if D0 is low, causing a write; output 4 (pin 5) will be low if D0 is high, causing an erase. In this way, an image will be formed with dots on where the character is off, and vice versa, in other words the terminal is in mode reverse. With WE_1 still off, but WE_0 high, the two active outputs are 1 (pin 2) and 5 (pin 6), which are wired opposite of the two just discussed, thereby causing an image corresponding to the stored character pattern, and the terminal is in the normal rewrite mode (the background is erased). With WE_0 clear but WE_1 set, the active outputs are 2 (pin 3) and 6 (pin 7). Note that pin 3 is not connected anywhere

and pin 7 is connected to coupled to the erase flip-flop. Hence, the terminal is in the overstrike erase mode, where dots corresponding to the character pattern are erased, but the background is untouched. With both WE_0 and WE_1 high, the active outputs are 3 (pin 4) and 7 (pin 9). Note that pin 4 is not connected anywhere, and pin 9 is connected to the write flip-flop. Hence the overstrike write mode, where dots corresponding to the character pattern are written, but the background is untouched.

The LOADPDLU input to U98 is normally inactive except when SERIAL is low, and WE_1 has been set low. Then the loading of data to both halves of the parallel data register at once will cause a write or erase depending on the setting of WE_0. The sense of control here for WE_0 is reversed from its use as control for the writing of parallel characters, which use the LOADPDU signal at U79 pin 5. This is so block erases followed by parallel writes can take place without having to change the panel mode.



CARROLL Mfg Co. Champaign, IL

DES BY	DATE
DRN BY RTL	DATE 11-30-77

TITLE
PLATO V TERMINAL

DESCRIPTION
PLASMA PANEL CONTROL

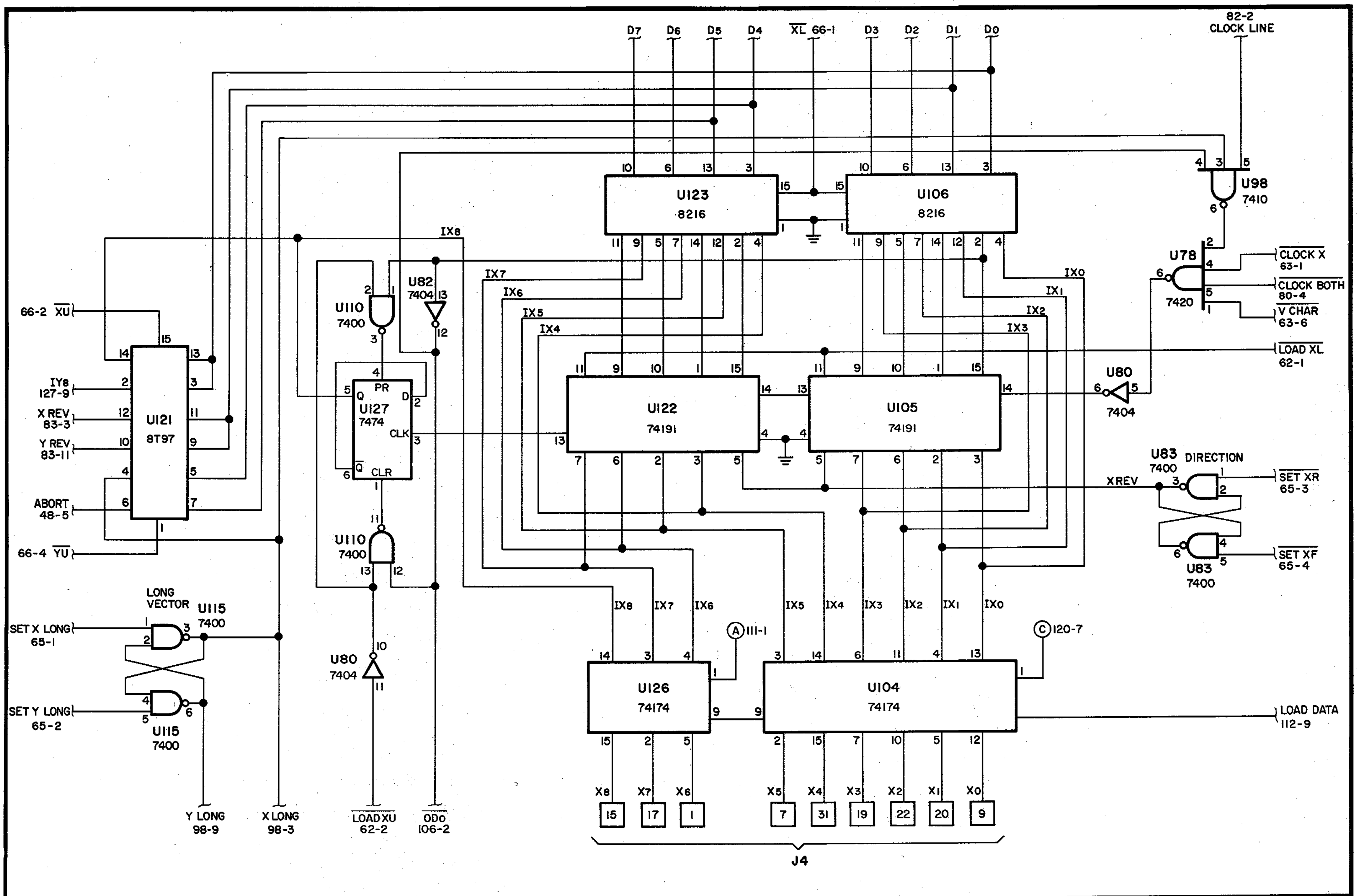
DWG NO
119 E 012

REV DATE

PANEL INTERFACE (X ADR)

This drawing shows the buffers, counters, and latches that provide the x panel address. The data bus interface is made up of U106 and U123, which route the data bus to the inputs of the counters U105 and U122. These two counters contain the lower 8 bits of the 9 bit panel address, and are loaded from the data bus by LOADXL. Once loaded, the counters may be incremented or decremented by the clocking pulses provided from NAND gate U78, from any of CLOCKX, CLOCKBOTH, or VCHAR. The direction of the count is determined from the setting of the XREV flip-flop at the right of the page. The outputs of the counters (the present panel address) are latched for application to the panel by U104 and U126. These same outputs are fed back as IXn to the data bus buffers U106 and U102. When reading this port, XL provides a low signal to pin 15 of the two buffers, driving the data bus with the output of the counters.

The loading of the most significant bit of the x address is done with the logic aroudn U127, the flip-flop that serves as the most significant bit of the counter. It is configured as a toggling flip-flop, with a clock provided by the borrow-carry output from the counter U122. It is ready by the microprocessor by its connection to U121, which also serves as the input port for a number of other data as well. On U121, if XU is active, then IX8 is read on D0, and XREV is read on D1. If YU is active, then IYB (the most significant bit of the y address) is read on D0, YREV is read on D1, XLONG is read on D4, and ABORT is read on D5.



CARROLL Mfg Co. Champaign, IL

DES BY	DATE
DRN BY RTL	DATE 11-24-77

TITLE PLATO V TERMINAL

DESCRIPTION PANEL INTERFACE (X ADR)

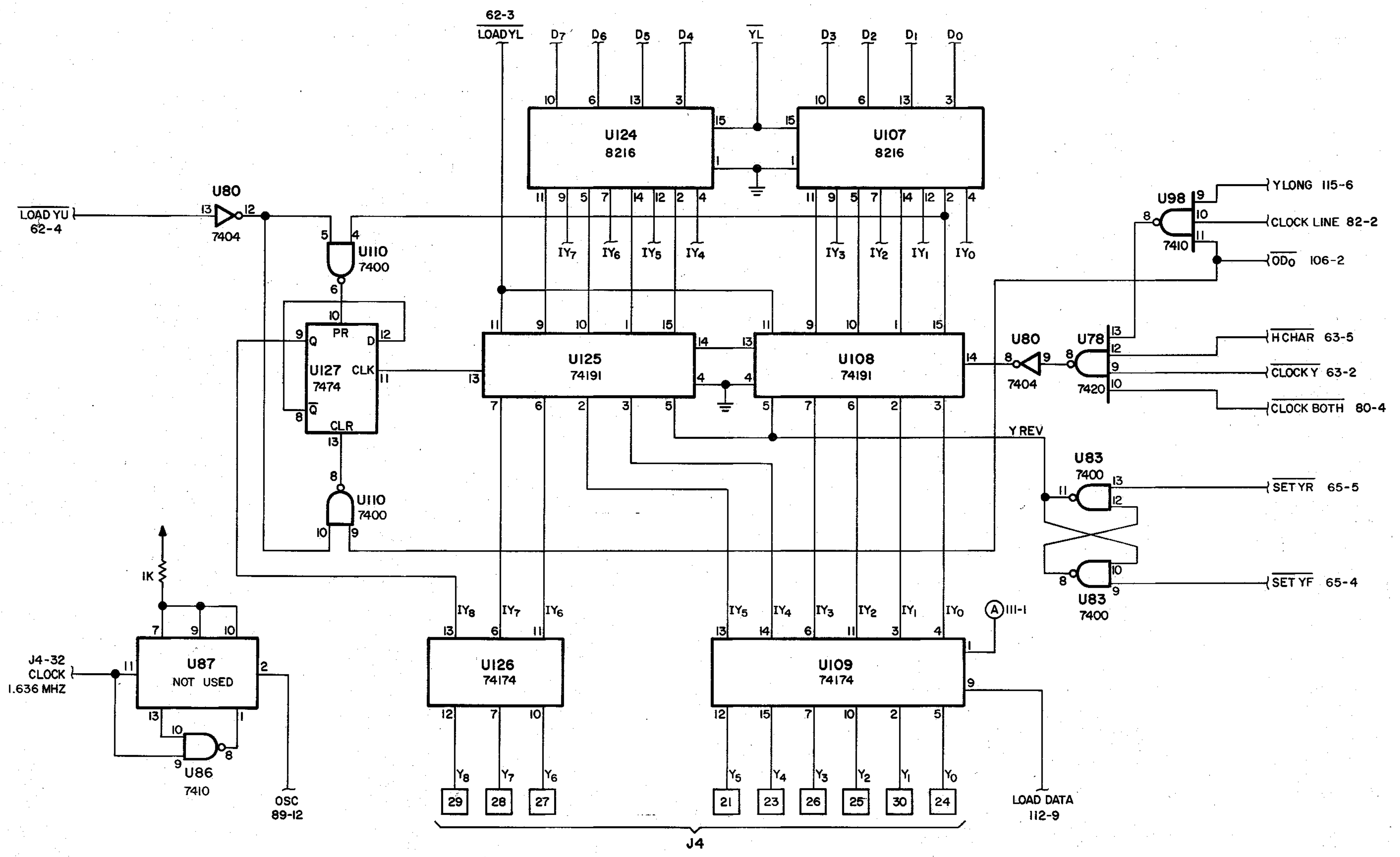
DWG NO 119 E 013

REV DATE

PANEL INTERFACE (Y ADR)

The operation of the y address circuitry is essentially identical to that of the x address circuitry. Signals named similarly have similar functions on this drawing. The x address drawing contained the bus driver used for reading the YU information, so that is not shown here. Also, the long vector flip-flop was shown on that diagram, and only its output, YLONG, is shown here as the input to pin 9 of U98.

The circuitry at the lower left generates a 1.6 MHz clock for use by a non-standard plasma panel, but is not used in the present terminal.



CARROLL Mfg Co. Champaign, IL

DES BY	DATE	TITLE	
DRN BY RTL	DATE 11-28-77	PLATO V TERMINAL	
REV	DATE	DESCRIPTION PANEL INTERFACE (Y ADR)	DWG NO 119 E 014

PANEL INTERFACE (DATA)

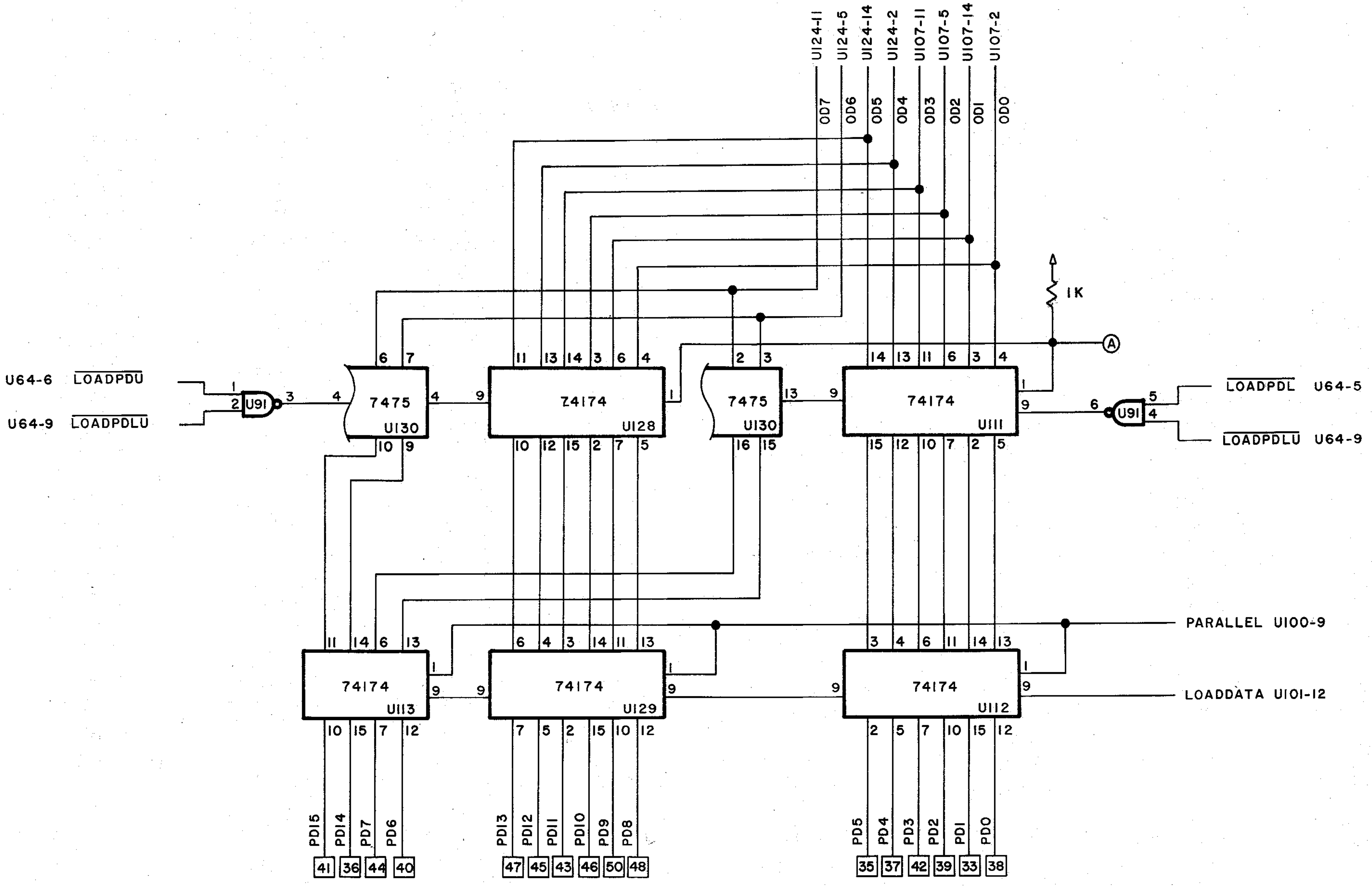
When in parallel mode, the panel will perform simultaneous write or erase to all the bit positions specified in a vertical column of 16 dots. The register shown on this page holds this data when output from the microprocessor and provides it to the parallel data inputs of the plasma panel.

The microprocessor can only output an eight bit piece of data at a time, so this register must generally be loaded in two parts, PDL (parallel data lower) and PDU (parallel data upper). The load signals for these portions are called LOADPDL and LOADPDU, and are provided by the decoding that was shown on the plasma panel control drawing. With these two load signals, generated by -out- instructions in the microprocessor, any sixteen bit data word can be written in the register.

There are two registers, cascaded so that one operation for the panel can be pending while a previously loaded one is actually being processed. The buffered data bus, OD0 through OD7, is brought from buffers U107 and U124 to the inputs of registers U111, U130, and U128. It is loaded into these registers under control of the latching signals as described above. The data resides in these registers until such time as a panel operation is initiated, at which time the load data pulse from U101 transfers the data into the output registers U112, U113, and U129. The parallel data are delivered to the plasma panel through a portion of J4.

For block write/erase, the LOADPDLU signal is provided. With a single -out-, the microprocessor can load both halves of the parallel data register at the same time. This is used by the resident to speed up the operation of r.block, which will fill or erase the screen at about a 5000 char/sec rate.

If the panel is operating in the serial mode, the line labeled PARALLEL coming to pins 1 of U112, U113, and U129 is low, thereby holding the parallel data output at zero.



PARALLEL DATA TO PLASMA PANEL

(PART OF J4)

CARROLL Mfg Co. Champaign, IL

DES BY	DATE	TITLE	PLATO V TERMINAL
DRN BY	DATE		
DESCRIPTION	PANEL INTERFACE (DATA)	DWG NO	119 E 015
REV	DATE		

MODEM CARD

The modem shown is called a "local modem" and is the one normally supplied with the terminal. It is designed for operation on full duplex (4 wire) telephone circuits of up to a few miles in length. Other modems designed for operation in other conditions are available. If suitable facilities for RS232 or current-loop data are available, the terminal will operate without a modem.

The modulator section generates the transmit data. U9 oscillates at 4800 Hz, which is divided by U10 to provide the 2400 Hz and 1200 Hz signals used by the modulator logic. The 1200 Hz signal is provided as an output to be used by the US/ART on the processor card as the transmit clock signal.

The NAND gates U6 with the inverter U3 form a multiplexer, selecting one cycle of the 1200 Hz signal if the data is a one, or two cycles of the 2400 Hz signal if the data is a zero. The composite signal is passed through a low pass filter (R36, C8, L1, C9, R12, and C6). This filter removes the high frequency components and transforms the signal into an approximation of a sine wave.

The signal is then amplified by Q5 and delivered to the telephone line. The gain adjustment (R17) should be set to provide a 2.1 volt peak to peak signal across the phone line.

The demodulator (receiver) circuitry is designed to recover signals coded similarly to those just described. The incoming signal from the phone line is passed through a limiting amplifier U8 and comparator U11 (pins 12, 11, 9) which convert the sine wave back into a digital waveform. The signal is then supplied simultaneously to a two-stage counter and to an integrator and comparator (Q6 and U11 pins 1, 2, and 4). The recovered data wave is coupled to the counter and integrator via the one-shots in U7, which provide clocking pulses on both the leading and trailing edges of the signal. The pulses clock the counter and reset the integrator.

The counter will provide an output after two short half-cycles, but only if the integrator/comparator has not detected a long half-cycle, since the comparator latch (U4 pins 3, 2, 6) holds the counter at zero unless set. As soon as the comparator detects a "one," the comparator latch is clocked high, which then, via its $\overline{Q}$ output, sets the output data latch to a one. It also then allows the counter to count, which increments until pin 6 goes low, resetting the comparator latch and the data latch. In this way, the comparator becomes a detector of "ones" and the counter a detector of "zeros."

There are three things that have to be adjusted on the modem -- frequency, transmit amplitude, and receive threshold.

Frequency

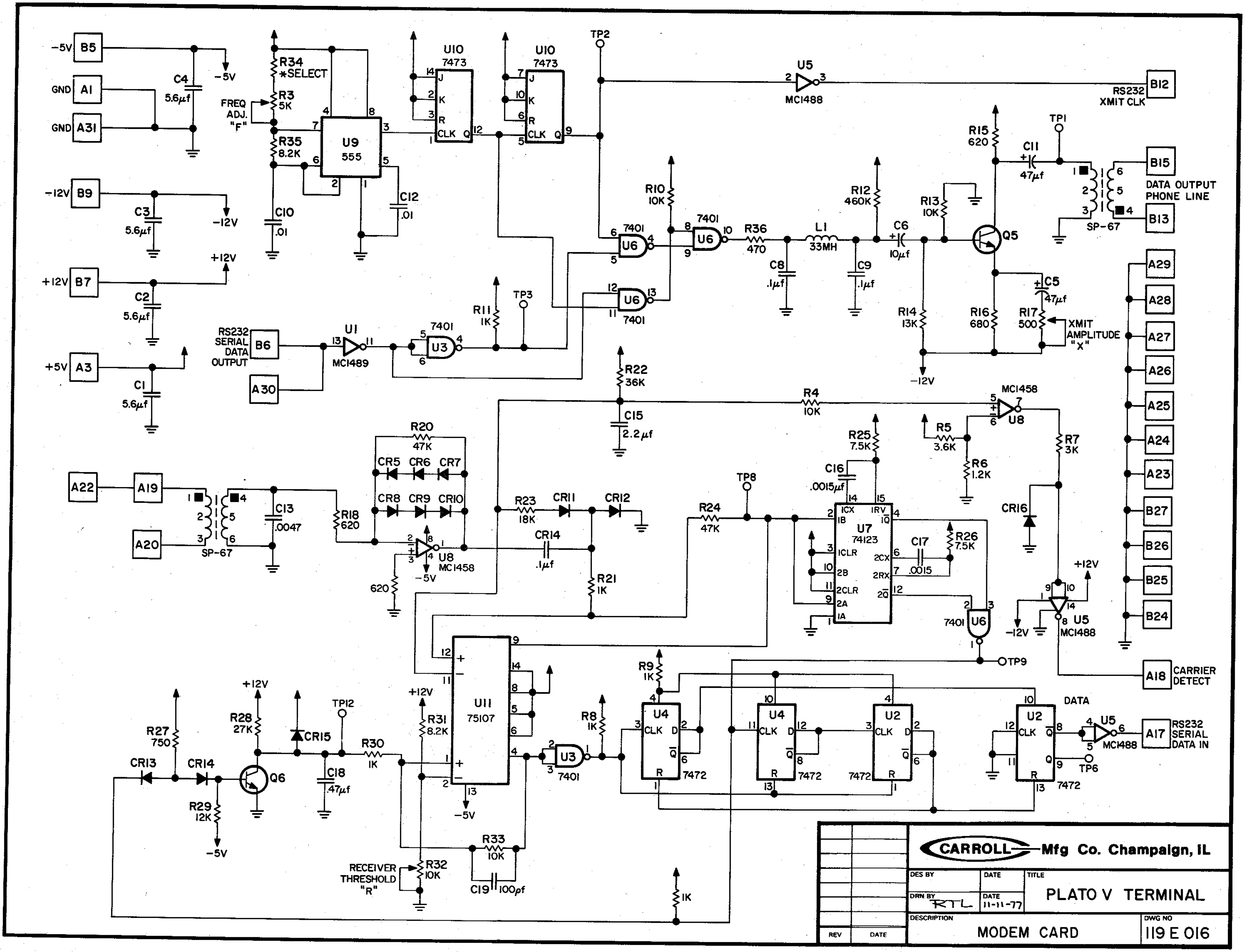
The frequency at test point 2 should be set to 1200 Hz. The most accurate reference for this is the crystal controlled 1200 Hz signal at pin 14 of U96 on the processor card.

Amplitude

The setting of the amplitude has already been discussed.

Threshold

The threshold adjustment can be done with various techniques, one of them being to watch the output data on test point 6 while the terminal is connected to a Datacom line. Adjusting R32 to the upper third of the range in which the start bit is plain and stable is reasonable. If a more scientific approach is desired, simultaneously monitor test point twelve and pin 1 of U3. In this way an exact knowledge of where the integrated signal crosses the threshold will be obtained. It should be set so that ones are comfortably detected, but zeros have little chance of being mistaken for ones. Any single crossing of the threshold will generate a one, so do not set the threshold too close to the maximum excursion of the integrated short cycles. A half-turn past the point where the signal is wide and stable is generally found to be a satisfactory operating point.

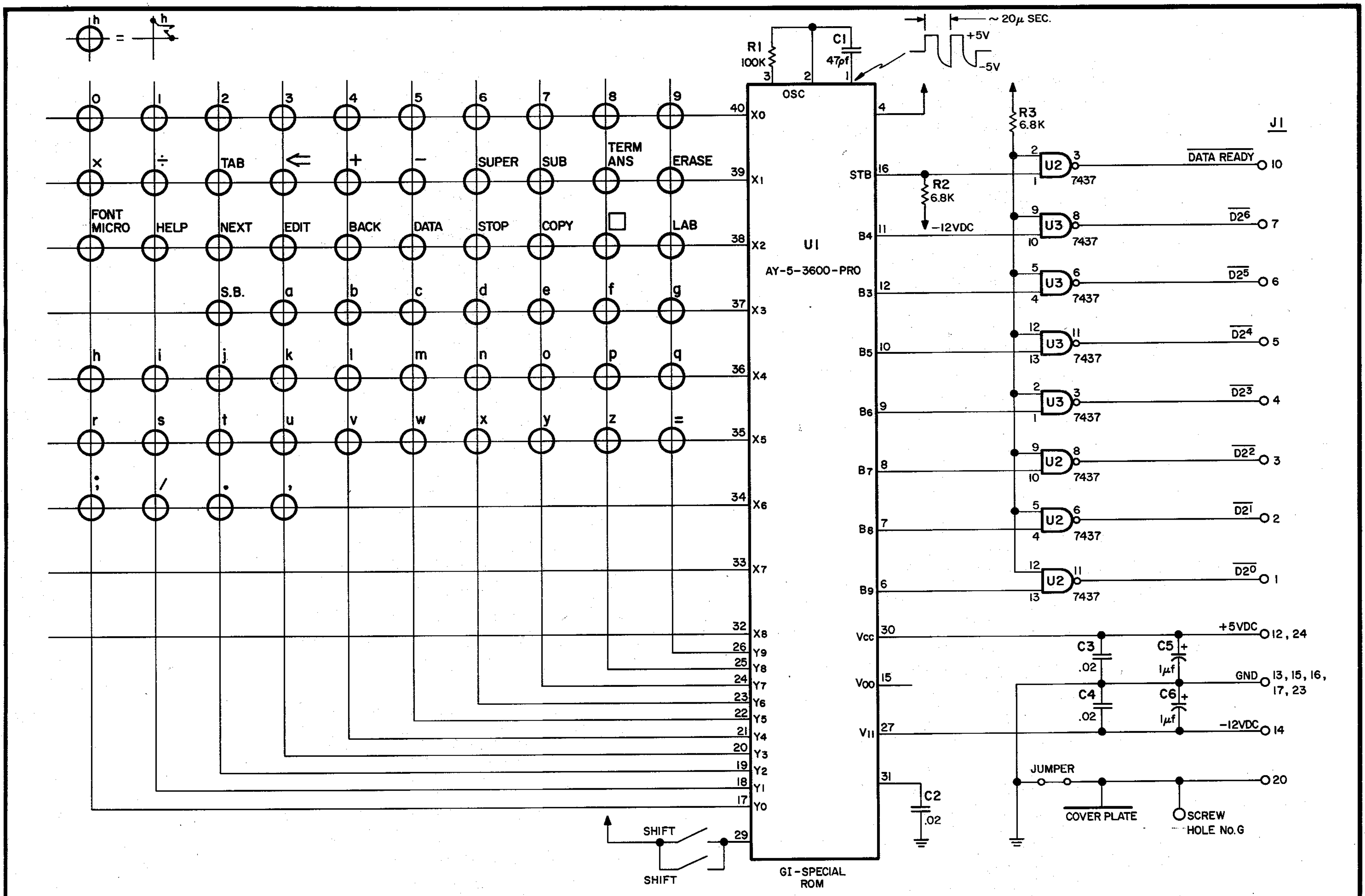


CARROLL Mfg Co. Champaign, IL

DES BY	DATE	TITLE
DRN BY RTL	DATE 11-11-77	PLATO V TERMINAL
DESCRIPTION MODEM CARD		DWG NO 119 E 016
REV	DATE	

KEYSET

The keyset is a simple crosspoint structure using a General Instrument lookup ROM. The keyset is scanned and provides one-key rollover. The codes that are generated are given in the X-50 report. The line drivers U2 and U3 buffer the data to the cable. The DATAREADY signal is a short pulse that sets the interrupt flag in the 8255 parallel interface chip in the CPU section.



LNU
U3
R3
J1

7437	U2,U3	14	7	—
AY-5-3600-PRO	U1	30	15	27
PART	REF	+5V	GND	-12V

CARROLL Mfg Co. Champaign, IL

DES BY	DATE	TITLE PLATO V TERMINAL	
DRN BY RTL	DATE 12-4-77		
DESCRIPTION KEYSET			DWG NO 119 E 017

REV DATE

TOUCH PANEL OPERATION

General

The touch panel is a standard input device which allows the operator to touch the display panel and input X-Y coordinate pairs directly to the computer. The touch panel consists primarily of a 16 x 16 array of light beams projected slightly above the display surface. Whenever both an X and Y light beam have been interrupted by a finger or other opaque pointer, the X/Y coordinates are transmitted to the computer.

The electronics consists of the scanning system, light sources and detectors, and control logic. Sequential scanning of the panel eliminates optical collimation problems. Scanning is controlled by an Oscillator, U27; a divide-by-four counter, U28; a strobe pulse circuit, U39; and a decoder, U30. These circuits, in conjunction with a four-bit counter, U9, and drivers U14 and U15 ensure that only one pair of X-Y light emitting diodes is pulsed at any time. The LED's utilized are TIL-31, narrow beam, infrared devices.

The phototransistors utilized are TIL-81's. These devices are located directly opposite the LED's. Every fourth phototransistor shares a common amplifier, U17 & U25 (with a gain of 8) and a common voltage comparator, U22 and U20. Due to the narrow beam-width of the LED's and the natural collimation of the light beams, only four amplifiers per axis are required. U21 is used to select one X and one Y comparator's output. This can be done, since the light beams from activated LED's do not spread to the other phototransistors connected to that comparator.

The voltage comparator is provided with a dynamically changing reference voltage, produced by U35-4, diodes D1 and D2, and resistors R24 and R25. The lower voltage threshold is used to detect a broken beam, while the higher voltage is used to detect unbroken beams while the panel is being pointed at by some type of pointer. The outputs of U21 (X-Signal, Y-Signal) are sensed by the control logic. When an X beam is broken, its address is stored in U7, while Y addresses are stored in U8. When both an X and Y beam are broken, these addresses are sent to the Plato® terminal, along with a Data Ready strobe signal. Scanning continues, and with each scan any new detected positions are compared by circuits U4, U5, and U6 against the addressed held in U7 and U8. Scanning continues until these addresses disagree, whereupon the control logic resets. Touch inputs for the Plato® terminal are limited to approximately 10/second by the control logic.

Circuit Operation

Oscillator U27 and circuits U28, U29, U30, and U31 generate the timing signals required to operate the touch panel. These signals include LED GATE, RESET GATE, LATCH RESET, φ3, φ1 and the clock for counter U9.

The outputs of counter U9 are decoded by U11, U14, and U15 and strobed by LED GATE. These drivers provide a current sink of -100ma for -50usec for the respective LED's. This produces a burst of light detectable by the phototransistors.

The signal from the phototransistors is coupled through a 0.1μfd capacitor to an amplifier (U17 or U25) where it is amplified 8 times and applied to the voltage comparator (U22 or U20).

The comparator outputs are applied to multiplexor (U21) which uses the two low order bits of counter (U9) to select the appropriate input line. The multiplexor outputs (X-Signal, Y-Signal) are pulse trains representing light pulses seen by the phototransistors. If pulses are detected, they are stored in the X and Y DETECT flip-flops, U31. These flip-flops are sampled by 1 and reset by 3. If a flip-flop is "0" when sampled by 1 (no light pulse detected), the corresponding LATCH flip-flop (U33) is set. This causes the contents of counter U9 to be stored in the appropriate register (U7 or U8). If only one broken beam is detected during a scan, both X and Y LATCH flip-flops are reset by U35.

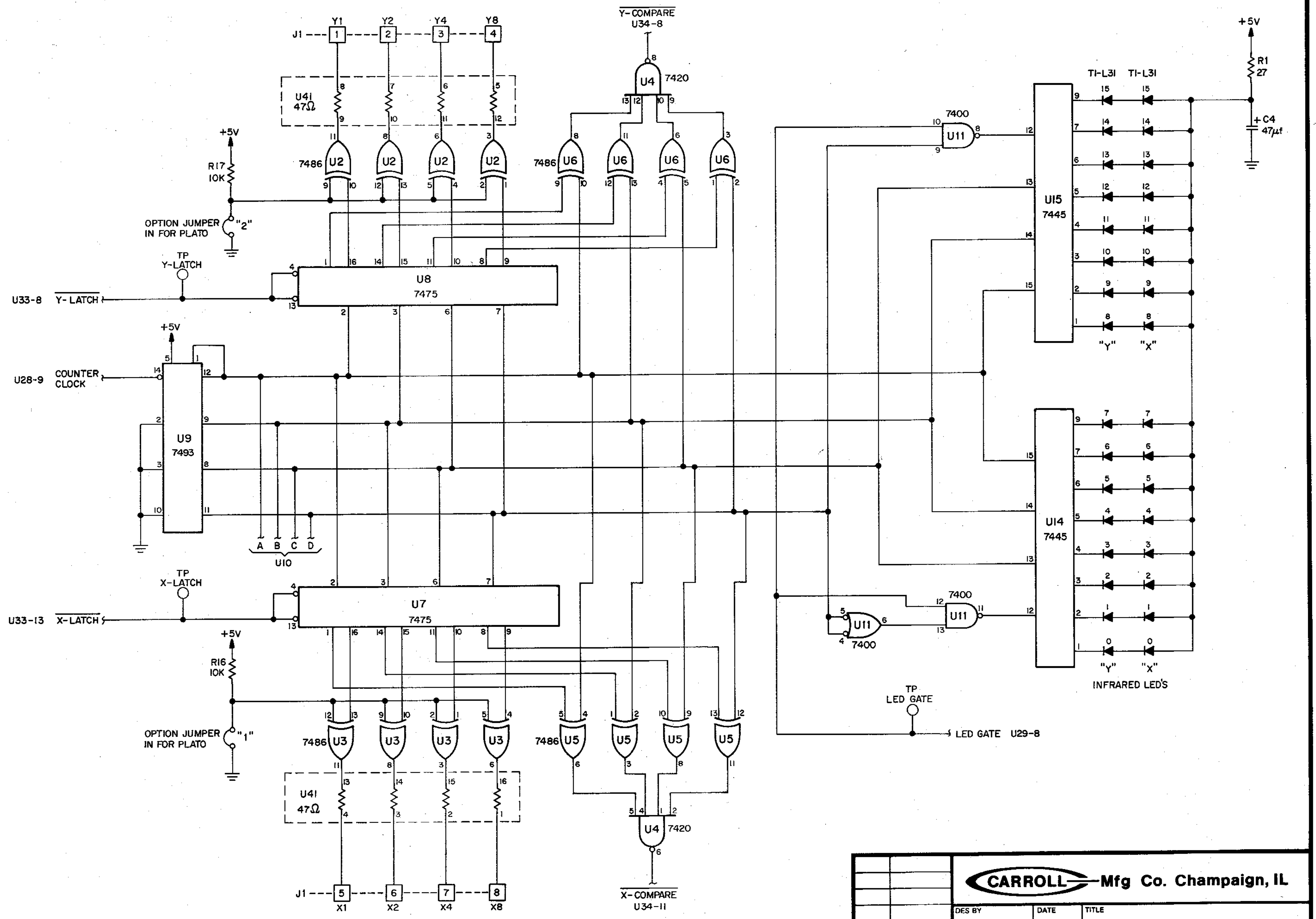
When both X and Y LATCH have been set, the GO signal is generated which produces the DATA READY signal (U37) to the terminal. Upon receipt of the DATA RESUME signal from the terminal, the DATA READY flip-flops reset and timing circuits U12 and U13 are activated. This timing circuit produces a 50ms tone at -500Hz indicating to the user that the X-Y coordinates have been accepted by the terminal. Additional control logic prevents multiple input of the same coordinate pairs by comparing the new detected position with the previous one.

Maintenance

Numerous test points are provided to aid in the checkout of the panel. Additionally, an oscilloscope SYNC point is provided which is asserted at the end of each complete panel scan. The presence of this pulse also indicates that power is present, the oscillator is running and the clock is functioning.

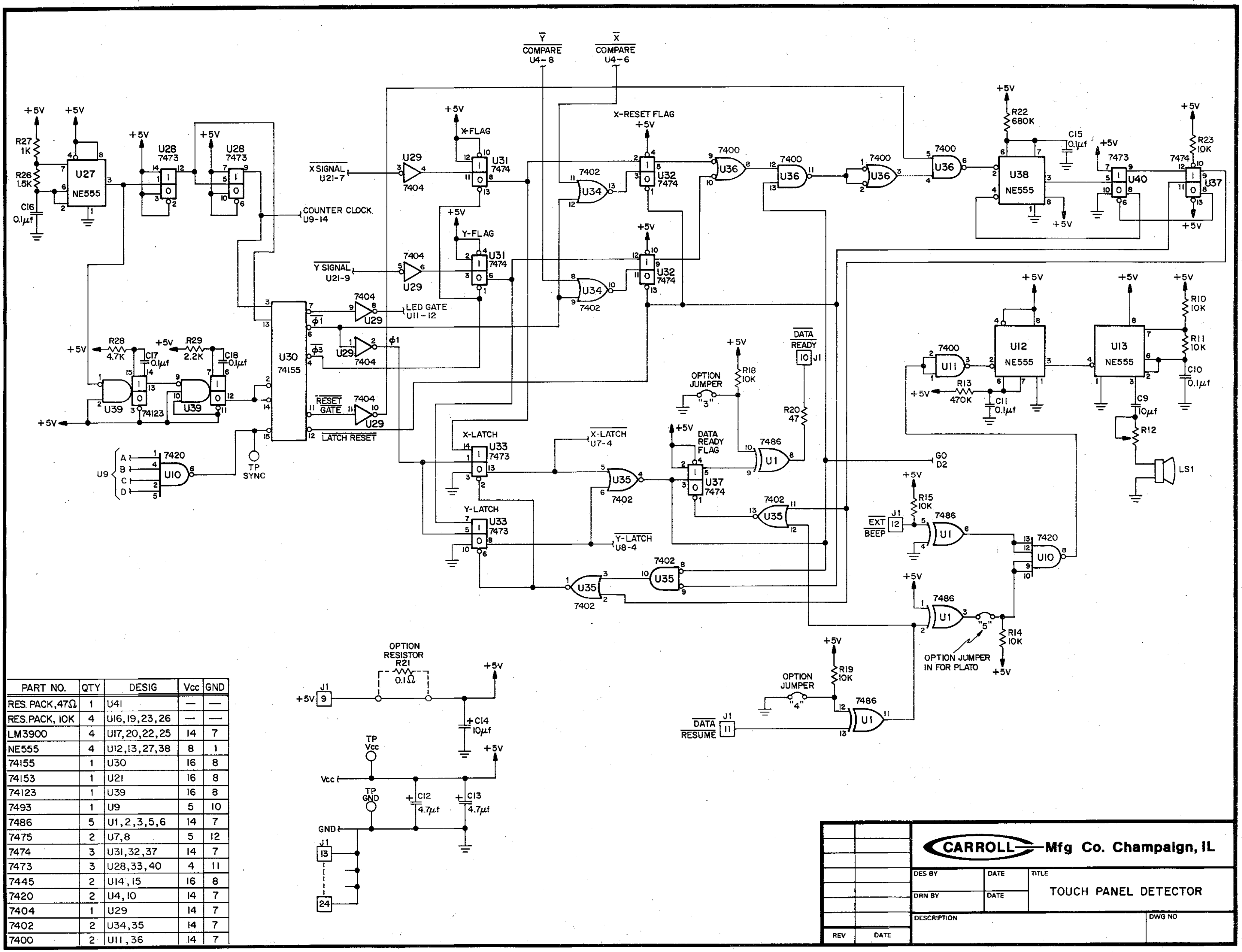
X and Y SIGNAL test points are most useful in troubleshooting. Each of these should appear as a continuous train of pulses (16 between SYNC pulses).

No maintenance is required except to keep the LED and Phototransistor appertures free from dirt and debris.



CARROLL Mfg Co. Champaign, IL

DES BY	DATE	TITLE		
DRN BY	DATE	TOUCH PANEL SCANNER		
DESCRIPTION			DWG NO	
REV	DATE			



PART NO.	QTY	DESIG	Vcc	GND
RES. PACK, 47Ω	1	U41	—	—
RES. PACK, 10K	4	U16,19,23,26	—	—
LM3900	4	U17,20,22,25	14	7
NE555	4	U12,13,27,38	8	1
74155	1	U30	16	8
74153	1	U21	16	8
74123	1	U39	16	8
7493	1	U9	5	10
7486	5	U1,2,3,5,6	14	7
7475	2	U7,8	5	12
7474	3	U31,32,37	14	7
7473	3	U28,33,40	4	11
7445	2	U14,15	16	8
7420	2	U4,10	14	7
7404	1	U29	14	7
7402	2	U34,35	14	7
7400	2	U11,36	14	7

CARROLL Mfg Co. Champaign, IL

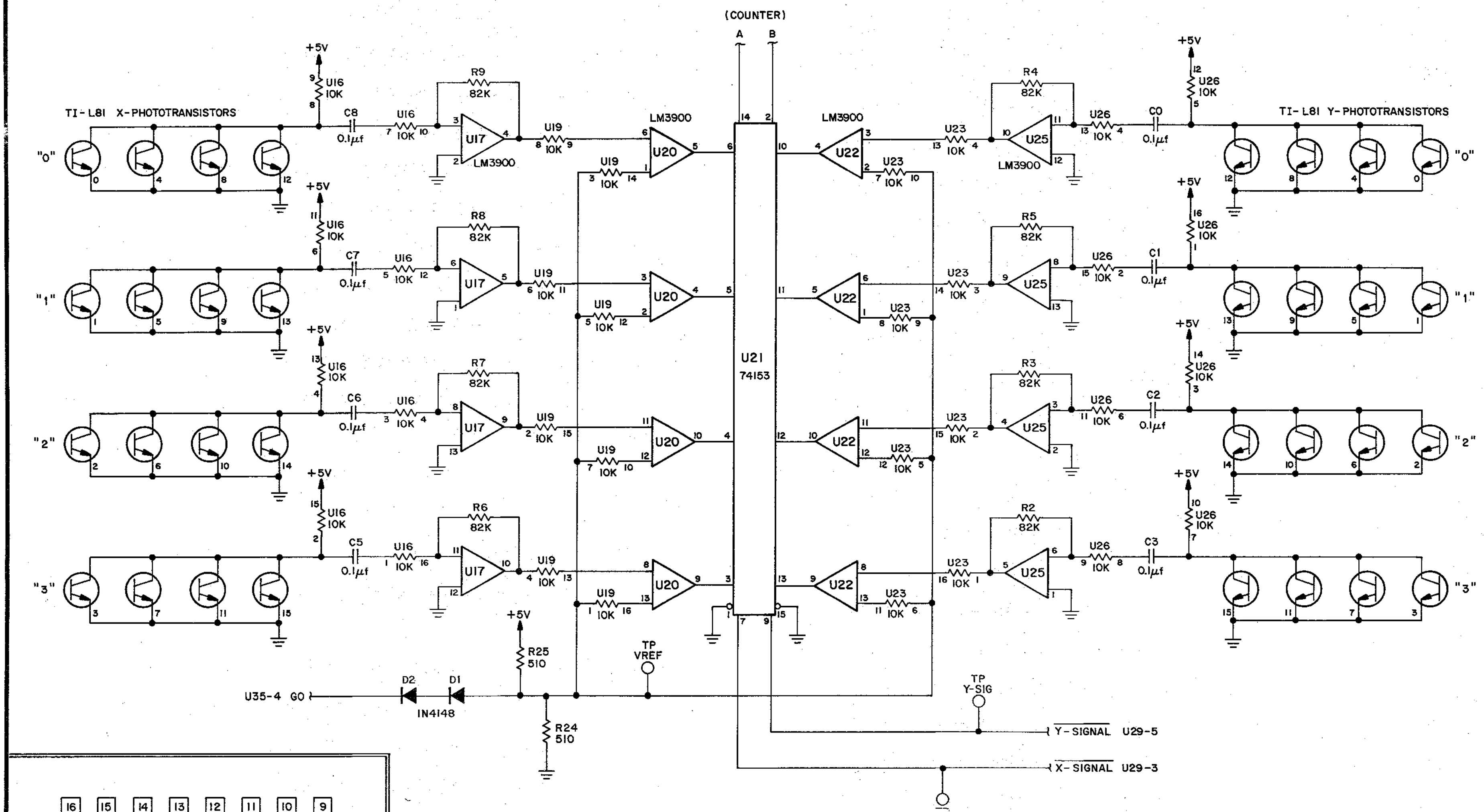
DES BY | DATE | TITLE

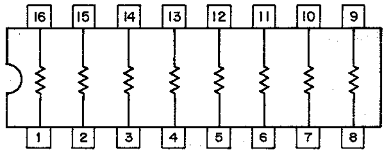
TOUCH PANEL DETECTOR

DRN BY | DATE

DESCRIPTION | DWG NO

REV | DATE





RESISTOR PACK U16, U19, U23, U26

CARROLL Mfg Co. Champaign, IL

DES BY	DATE
DRN BY	DATE
TITLE	TOUCH PANEL CONTROL
DESCRIPTION	DWG NO
REV	DATE

PANEL REMOVAL

To remove the plasma panel from the terminal:

1. remove the front cover by loosening the four flathead allen screws near the four corners of the screen. They are captive to the cover, and so turn CCW until you can hear or feel the last thread "skip."
2. remove the top cover plate by loosening the four button head cap screws that are at the four top corners of the terminal.
3. remove the touch panel
4. remove the lower front cover plate by removing the two button head cap screws that fasten it to the frame. These are located under and at each end of the plate, and are most easily loosened by sliding the terminal so that the front of it hangs over the edge of a table. There are only two other screws just behind these, be careful to loosen only the two forward screws.
5. remove the rear cover of the terminal by loosening four button head cap screws, two on top of the terminal and two at the rear bottom. Slide out gently so as to avoid any scratching of the wood slides, remove the power plug at the fan, and lay the panel aside.
6. remove the connectors from the rear of the plasma panel, by loosening the six button head cap screws, one at each end of each of the three connectors. Be careful to avoid dropping any, since they are directly above the plasma power supply and difficult to find if dropped. It is recommended that the power supply be covered with paper or cardboard to prevent this.
7. remove the panel by loosening four button head cap screws, two on each side of the plasma panel. Then slide the panel forward, being careful to keep it square with the sides of the terminal, to prevent binding and scratches. It slides easily, since it is set on plastic "runners" for the purpose. After it is about three inches forward, it can be gripped firmly by its edges, or by the rear cutout and top. A firm grip is strongly recommended since the panel value represents about half the cost of the terminal itself.
8. If a new panel is to be installed that does not have the shield board and backets mounted, these can be removed from the old panel and installed on the new one. The procedure is nearly self-evident, the only exception being to note that the two side rails are NOT interchangeable.