Tektronix®
COMMITTED TO EXCELLENCE

Please Check for
CHANGE INFORMATION
at the Rear of this Manual

4024 AND 4025
COMPUTER DISPLAY TERMINALS

(Vol. 1, Theory of Operation)

SERVICE MANUAL

Tektronix, Inc.
P.O. Box 500
Beaverton, Oregon 97077

MANUAL PART NO. 070-2830-00
PRODUCT GROUP 11

First Printing AUG 1979
Revised JAN 1983

WARNING

THE FOLLOWING SERVICING INSTRUCTIONS ARE FOR USE BY QUALIFIED PERSONNEL ONLY. TO AVOID PERSONAL INJURY, DO NOT PERFORM ANY SERVICING OTHER THAN THAT CONTAINED IN OPERATING INSTRUCTIONS UNLESS YOU ARE QUALIFIED TO DO SO.

MANUAL REVISION STATUS

PRODUCT: 4024 and 4025 Computer Display Terminals

This manual supports the following versions of this product: 4024 - B010100 and up; 4025 - B010100 and up

REV.	DATE	DESCRIPTION
@	AUG 1979	Original Issue
A	SEP 1979	Revised: pages 2-2, 5-13, 5-14.
A	FEB 1980	Revised: pages 1-2, 12-7, Tab A, A-14, I-31.
A	MAR 1980	Revised: page 9-39.
-	OCT 1980	Revised: pages 3-1, 3-2 and 3-11.

CONTENTS

ILLUSTRATIONS

TABLES

Figure 1-1. 4024 and 4025 Computer Display Terminals.

Section 1

INTRODUCTION

ABOUT THIS MANUAL

The service documentation for the 4024 and 4025 Computer Display Terminals consists of two volumes. Volume 1 (this volume) contains the circuit descriptions. Volume 2 contains installation procedures, calibration procedures, block diagrams, schematics, and parts lists.

Additional information is available in the following manuals:

- 4024 Computer Display Terminal Operator's Manual.
- 4025 Computer Display Terminal Operator's Manual.
- 4024/4025 Computer Display Terminal Programmer's Reference Manual.
- 4024/4025 Programmer's Reference Guide.
- 4025 Option 41 / 851 Diagnostic Instruction Manual.
- 4020 Series Polling Reference Manual.
- 4020 Series Polling Service Manual.

ABOUT THE TERMINALS

Basic Characteristics

The 4024 and 4025 Computer Display Terminals use a raster scan video display (an internal video monitor). The display uses a 525-line, 60-Hz interlaced raster scan, like that in American television sets. A long-persistence green phosphor minimizes flicker and reduces eyestrain.

Both terminals include an 8080 microprocessor, whose programs are stored in ROM (read only memory). The microprocessor oversees all aspects of the terminal's operation: storing text and graphic information in internal memory, communicating with the host computer and with peripheral devices, interpreting and executing commands from the operator and the computer, etc.

Although the terminal includes a microprocessor, it is not programmable by the user. The terminal will accept a list of commands, and execute them in the order of receipt. However, there are no 4024/4025 commands for "add two numbers" or "go back to step 6 of the program." The only commands are for such functions as "change the baud rate," "send the stored text to the computer," or "draw a vector on the screen."

Differences Between 4024 and 4025

The 4024 is a simplified version of the 4025. The differences are:

- The 4025 has more card slots in its Mother Board than the 4024, and it has many more options available. Options available for the 4025, but not for the 4024, include:
  - Graphics
  - Character Set Expansion
  - GPIB Interface
  - Half Duplex Data Communications
  - Polling Controller
- The 4025's Display Controller can drive a hard copy unit or an external video monitor. Besides its standard character font, it can consult ROMs on a Character Set Expansion Board to display 64 different rulings characters, special math characters, or characters of different alphabets. It may also consult RAMs on a Graphics Memory Board to display graphic information or user-defined character fonts. The 4025 Display Controller can generate enhanced backgrounds, inverted video, and underlines to give special emphasis to portions of the displayed text.

  The 4024's Display Controller, on the other hand, is much more limited. It cannot drive a hard copy unit or external video monitor. It cannot display inverted video or underlines, although it can generate the "enhanced" background. The only character fonts it can display are the standard font and 32 rulings characters.

Options

Table 1-1 lists the options available for the 4024 and 4025.

4024/4025 OPTIONS SUMMARY

4024 Option	4025 Option	Name	Description
—	1	HALF DUPLEX	Permits Half Duplex Normal and Half Duplex with Supervisor data communications. Requires Option 35.
2	2	CURRENT LOOP	Permits the terminal to communicate with the host computer by means of a 20 mA current loop, instead of the standard RS-232 interface.
3	3	RS-232 PERIPHERAL INTERFACE	Provides communications with RS-232 peripheral devices, such as the TEKTRONIX 4642 Printer. In the 4025, requires Options 35 and 36.
—	4	GPIB PERIPHERAL INTERFACE	Provides communications with GPIB peripherals—4924 Tape Drive and 4662 Plotter. Requires Options 35 and 36.
10	10	POLLING INTERFACE	Permits the terminal to be one of up to eight display stations communicating over a 20 mA current loop with the same Option 11 Polling Controller, and communicating through that Controller with the host computer.
—	11	POLLING CONTROLLER	Provides full duplex interface up to 4800 baud. Also controls up to eight 4024/4025 terminals, including the terminal in which the Polling Controller is installed.
20, 21, 22	20, 21, 22	ADDITIONAL DISPLAY MEMORY	Provides additional display memory for a total of 8K, 16K, or 32K eight-bit words.
—	23, 24, 25, 26	GRAPHICS MEMORY	Provides 4K, 8K, 16K, or 32K of graphics memory. Also provides the firmware for drawing graphs in the workspace. Requires Option 35.
—	31	CHARACTER EXPANSION	Provides the Character Set Expansion Board, on which ROMs holding optional character sets can be placed.
32	32	RULINGS CHARACTERS	A ROM holding the rulings characters set. In the 4025, requires Option 31.
—	34	MATH CHARACTERS	A ROM holding the math character set. Requires Option 31.
—	35	ROM EXPANSION	Provides the ROM Expansion Board, in which ROMs holding optional firmware can be inserted.
—	36	PERIPHERALS ROM	Provides firmware to drive RS-232 compatible printers (when used with Option 3), or the 4924 Tape Drive and the 4662 Plotter (when used with Option 4). Requires Option 35.
—	40	HARD COPY AND VIDEO OUT	Provides hard copy capability with TEKTRONIX 4631 Hard Copy Unit. Also provides a BNC connector for attaching an external video monitor.
—	41	SELF TEST	Extended test to provide fault isolation. Requires Option 35.
48	48	220 VOLTS, 50 HZ	

Accessories

The standard and optional accessories for the 4024 and 4025 are:

- Standard Accessories

  4024 Computer Display Terminal Operator's Manual
  4025 Computer Display Terminal Operator's Manual
  Keyboard Overlay (blank)
  Blank Key Cap (1 x 2)
  Blank Key Cap (1 x 1)
  Key Cap Cover (1 x 2)
  Key Cap Cover (1 x 1)
  Power Cord

- Optional Accessories

  4024/4025 Service Manual (Two Volumes)
  4024/4025 Programmer's Reference Manual
  4024/4025 Programmer's Reference Card

Section 2

SYSTEM OVERVIEW

Refer to Figures 2-1 and 2-2, the system block diagrams for the 4024 and the 4025. The basic differences between the terminals are:

- The 4025 has room for nine boards in its Mother Board, while the 4024 has room for only four boards.
- The 4025 has separate Processor and Data Communications boards; in the 4024, these are combined on a single board, the Processor/Comm Board.
- The 4025 Display Controller Board has provisions for obtaining dot patterns for alternate character sets from the optional Graphics Memory and Character Set Expansion Boards. Also, the 4025 Display Controller has provision for attaching connectors (Option 40) to a 4631 Hard Copy Unit and an external video monitor.
- The 4025 has provisions for adding many options: graphics, several alternate character sets, RS-232 peripheral interface, GPIB peripheral interface, etc. Fewer options are available for the 4024.
- The 4025 has a cooling fan; the 4024 does not.

NOTE

The following discussion assumes for convenience that the terminal is a 4025. For instance, it refers always to the "Processor" or "Processor Board," rather than to the "Processor/Comm Board." Significant differences between the two terminals are pointed out.

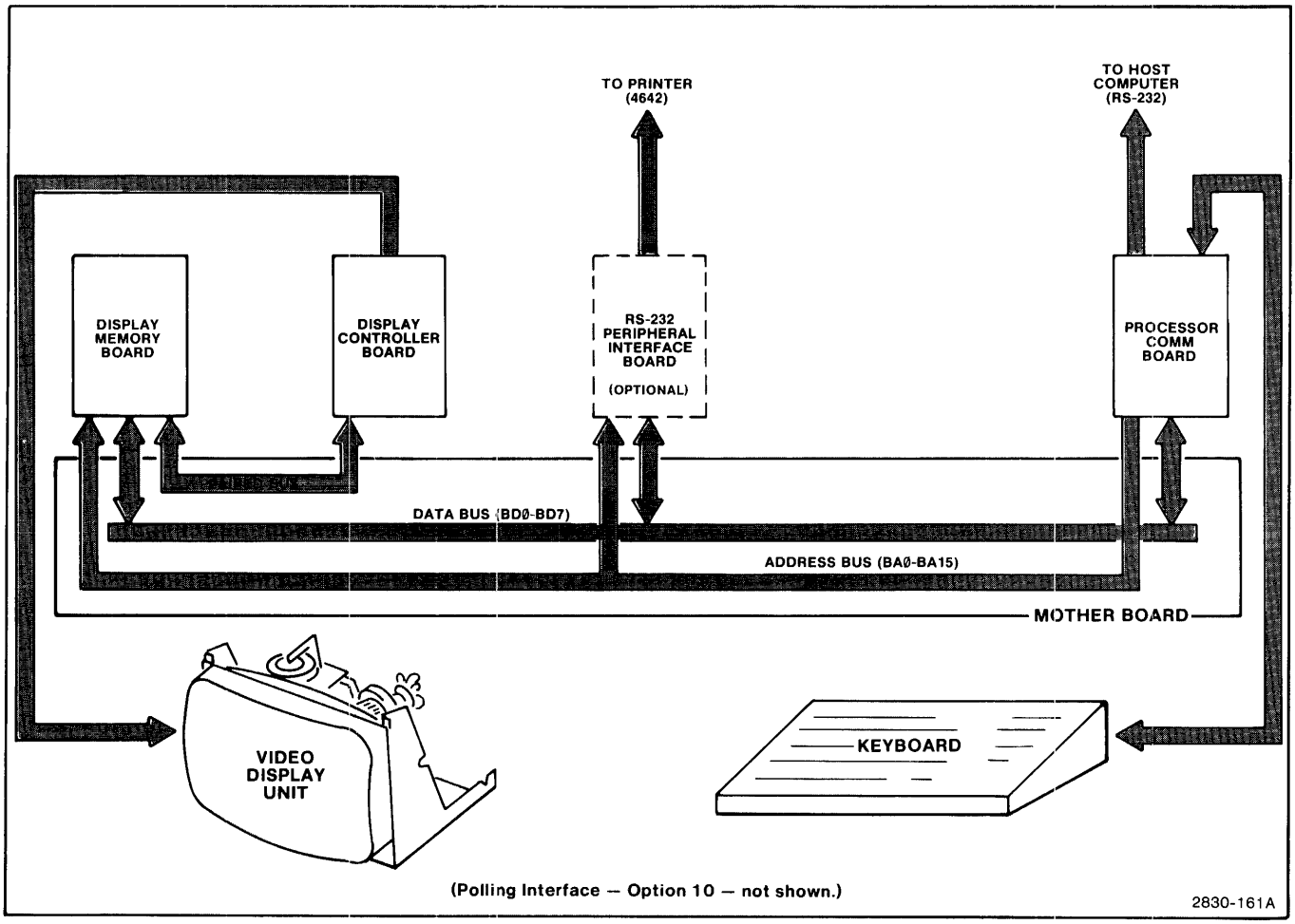


Figure 2-1. 4024 System Block Diagram.

The 4024/4025 is based on an 8080 microprocessor. This device can address up to 65,536 eight-bit words of memory. Part of the memory resides on the Processor Board, but most of it is on other boards, especially the Display Memory Board.

To communicate with memory on other boards, the Processor uses certain signal lines in the Mother Board. The most important of these are the Address Bus (signal lines BA0 to BA15), the Data Bus (BD0-BD7), and the READ and WRITE lines. (See the Mother Board circuit description—in Section 3 of this manual—for information on these and other Mother Board signal lines.)

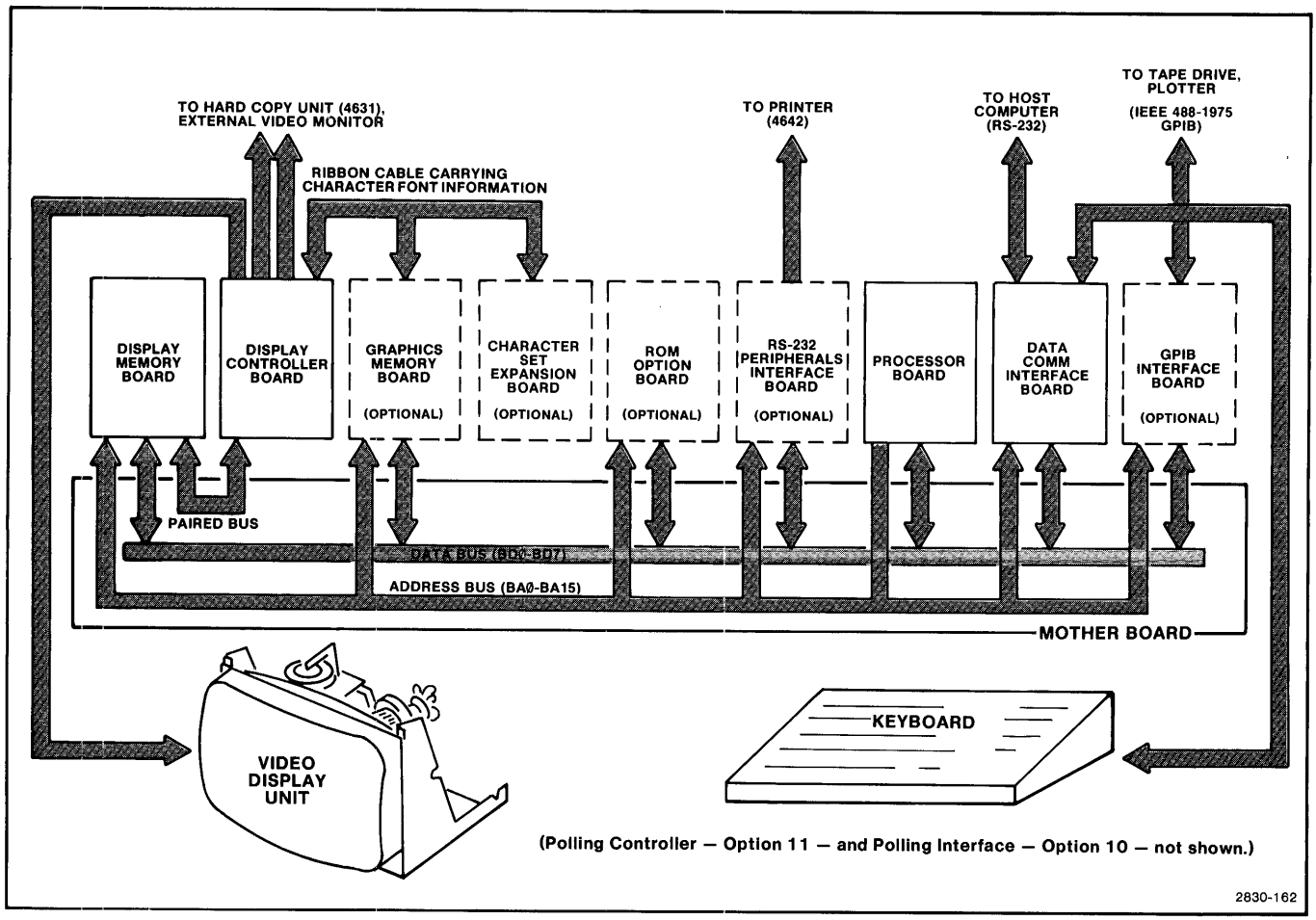


Figure 2-2. 4025 System Block Diagram.

MEMORY ADDRESS SPACE

The Processor *memory address space* is the collection of all 65,536 possible memory addresses. Different parts of memory address space are used for different purposes, and some parts are not used at all. Figures 2-3 and 2-4 show the memory allocations for the 4024 and the 4025.

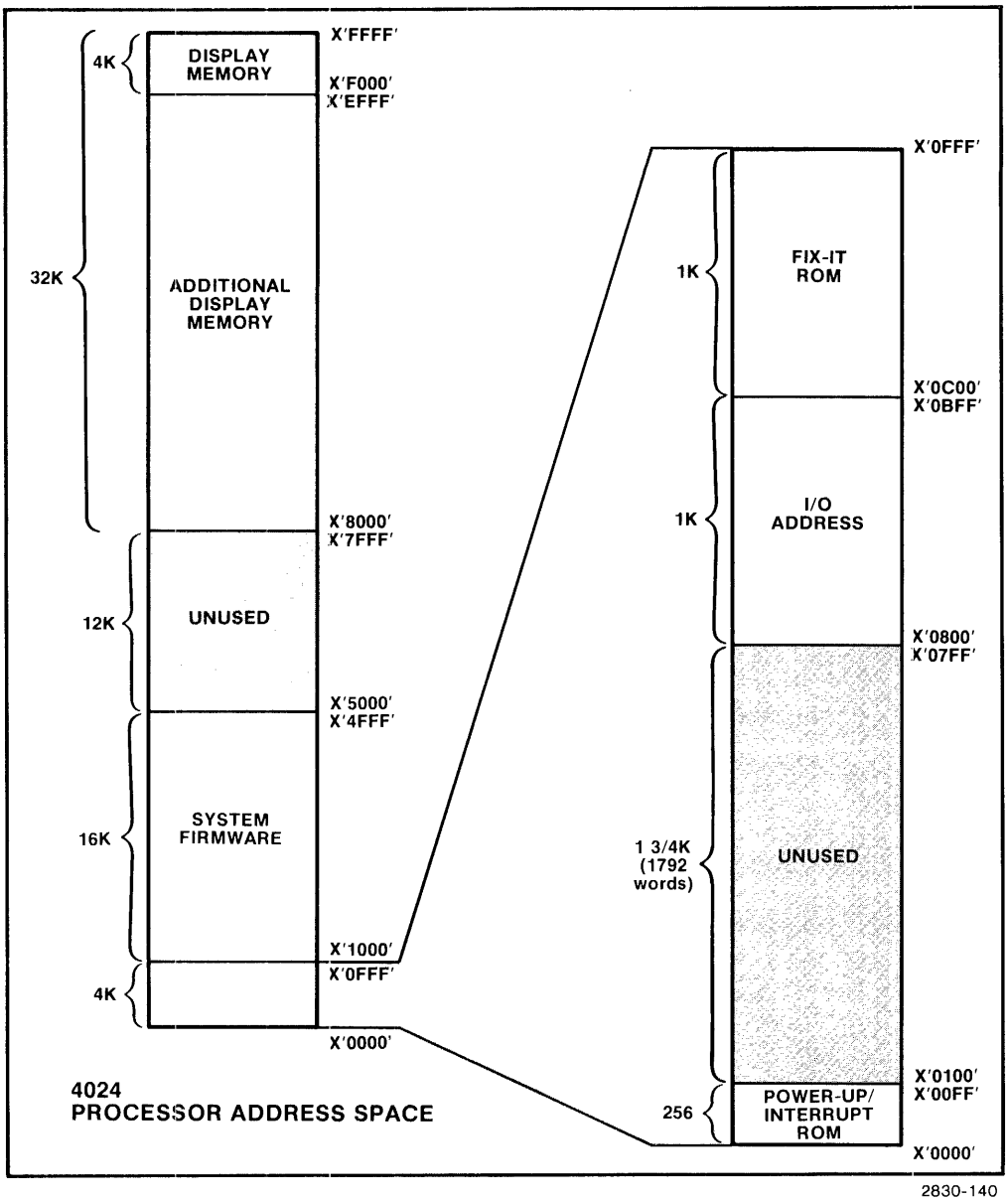
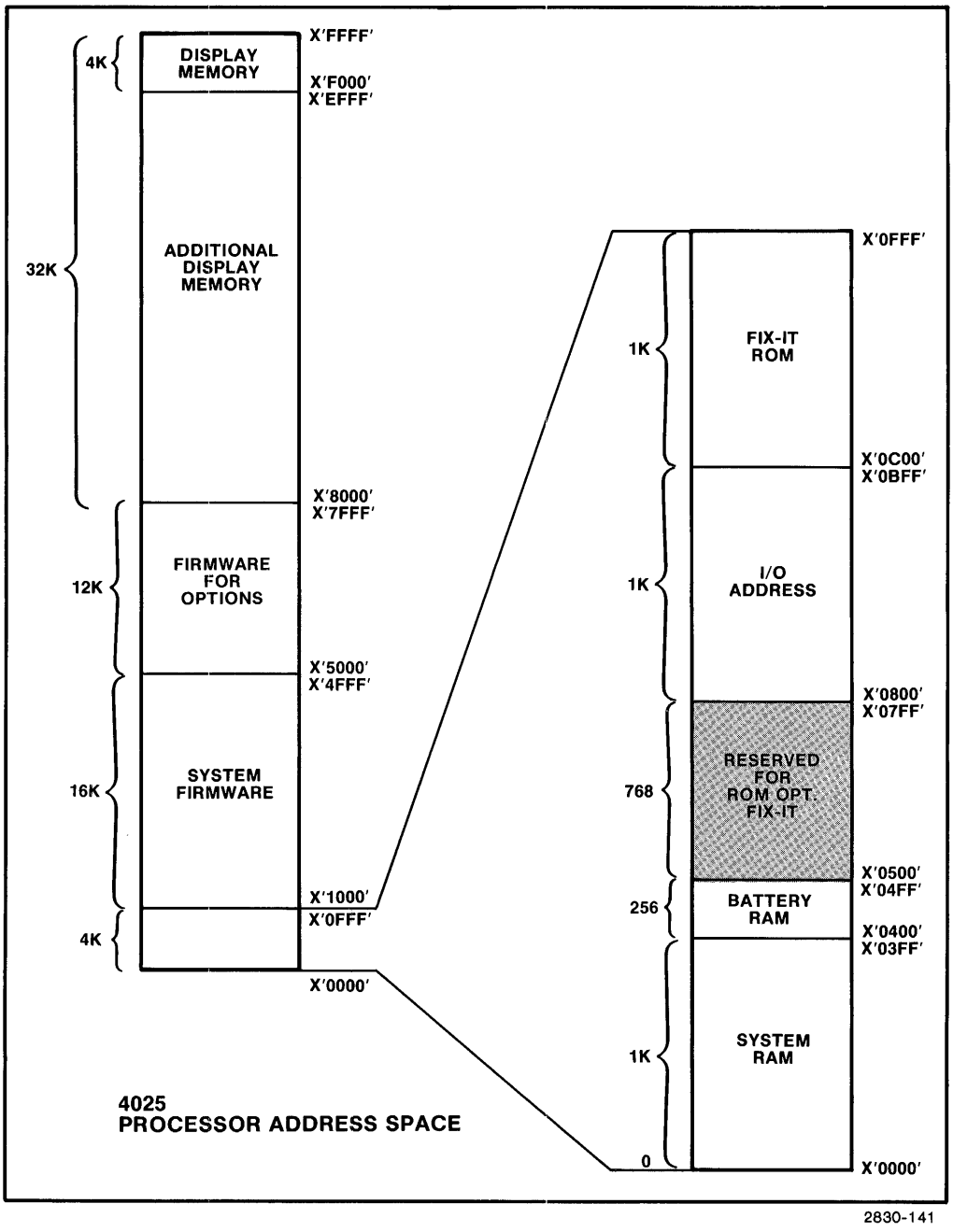


Figure 2-3. 4024 Memory Map.


4K
DISPLAY MEMORY
X'FFFF'
X'F000'
X'EFFF'
32K
ADDITIONAL DISPLAY MEMORY
X'8000'
X'7FFF'
12K
FIRMWARE FOR OPTIONS
X'5000'
X'4FFF'
16K
SYSTEM FIRMWARE
X'1000'
X'0FFF'
4K
X'0000'
4025
PROCESSOR ADDRESS SPACE
1K
FIX-IT ROM
X'0FFF'
X'0C00'
X'0BFF'
1K
I/O ADDRESS
X'0800'
X'07FF'
768
RESERVED FOR ROM OPT. FIX-IT
X'0500'
X'04FF'
256
BATTERY RAM
X'0400'
X'03FF'
1K
SYSTEM RAM
0
X'0000'

2830-141

System RAM

In the 4025, the lowest memory addresses (X'0000' to X'03FF') hold RAM (Random Access Memory, or read/write memory). This *System RAM* resides on the Processor Board and is used by the Processor as a scratchpad. Also, certain low-address words of memory hold "pointers" which direct the Processor to the appropriate instructions for responding to interrupt conditions.

The 4024 lacks System RAM. It uses RAM on the Display Memory Board for its scratchpad. To hold the pointers to the interrupt-handling subroutines, the 4024 uses a 256-byte ROM (addresses X'0000' to X'00FF').

Battery RAM

Memory locations X'0400' to X'04FF' are in the battery-powered RAM on the 4025's Processor Board. This RAM holds settings such as baud rate, parity, etc., so that the 4025 "remembers" the settings even when its power is turned off. (The 4024 lacks a battery-powered RAM; in the 4024, these memory locations are unused.)

I/O Address

Memory locations X'0800' to X'0BFF' are reserved for "I/O registers" on boards other than the Processor Board. The Processor can write to, and read from, these registers in exactly the same way as it accesses any other location in memory. Unlike other memory locations, however, the I/O registers do not store data or instructionsfothe Processor. Instead, the Processor uses them to send control signals to the other boards, or to receive signals from those boards.

For instance, by writing into memory location X'0832', the Processor can light the LEDs in the keyboard's lighted function keys; and by reading from that location, it can tell which keys are being pressed. This technique is called "memory-mapped I/O." It gives the Processor access to a wide variety of signal lines, without the need for those lines to be directly wired to the Processor.

Fix-It ROM

Addresses X'0C00' to X'0FFF' are reserved for the *Fix-It ROM*, a read-only memory used to make corrections to firmware. (The Fix-It is discussed later in this section.)

Second Fix-It ROM

Memory locations X'0800' to X'0BFF' are reserved for a Fix-It ROM, used to make changes to the optional firmware. This ROM, if installed, resides on the ROM Expansion Board.

System Firmware

Addresses X'1000' to X'4FFF' are occupied by four ROMs which hold the terminal's system firmware.

Optional Firmware

Addresses X'5000' to X'7FFF' are reserved for optional firmware ROMs. These ROMs are installed in the ROM Expansion Board.

The 4024 does not use this block of memory addresses.

Display Memory

Addresses X'8000' to X'FFFF' are reserved for display memory (on the Display Memory Board). This is the terminal's main memory resource; in it are stored all the text in the workspace and monitor scrolls of memory, as well as input/output buffers, programmed key definitions, etc.

Addresses X'F000' to X'FFFF' are used for the minimum 4K of display memory. As more display memory is added, it extends downward from X'F000'. With the maximum 32K (Option 22) installed, display memory occupies addresses X'8000' to X'FFFF'.

HOW TEXT IS STORED AND DISPLAYED

From the user's point of view, the terminal stores text in two independent scrolls of memory: the monitor and the workspace. Parts of the two scrolls are displayed on the screen. Either the operator or the computer can insert text into either scroll.

Display List

Internally, the two scrolls are just different parts of the same *display list*. That list is a sequence of words in display memory containing all text in the monitor and workspace scrolls, together with instructions for displaying that text. To designate which parts of the display list shall appear on the screen, the Processor inserts instructions in the list.

Other circuitry automatically reads the display list, follows the instructions it contains, and generates video signals for the Video Display Unit. This circuitry consists of the Tracker (on the Display Memory Board), and the Display Controller (on its own board).

Tracker

The Tracker is a specialized processor which reads the display list and composes, line by line, information for the Display Controller. While the Display Controller scans one line of text, the Tracker is sending it the information for the next line. The Tracker communicates with the Display Controller by means of the "paired bus" lines of the Mother Board.

Display Controller

The Display Controller receives from the Tracker information about which characters to display. It consults its own Character Set Memory (not part of Processor address space) to learn how to draw those characters, and then generates signals for the Video Display Unit.

Usually, characters are displayed in character font zero (the standard font). A ROM on the Display Controller Board holds the details of how to draw characters in this font.

Sometimes, though, other fonts are used. For rulings or math characters, the Display Controller consults a ROM on the Character Set Expansion Board. (In the 4024, the rulings ROM is on the Display Controller Board.) For graphics characters, the 4025 Display Controller consults RAMs on the Graphics Memory Board. A ribbon cable carries character font information betweeen the Display Controller, Character Set Expansion and Graphics Memory Boards.

Entering Text From the Keyboard

(See Figure 2-5.)

To see the functions of the different circuit boards, let's consider what happens when the operator types text into the terminal's internal workspace. (This simplified procedure omits many details.)

1. The operator types the letter "w."

2. The keyboard tells the Data Communications Interface that "the W key is being pressed—and the SHIFT key is not pressed."

3. The Communications Interface Board puts a "key pressed" code into its Keyboard Port Data Word. This is a register on the Communications Interface Board which appears to the Processor to be memory location X'0832'.

4. The Processor reads from location X'0832'.

5. The Processor decides, based on the data just read, that "lowercase w" is the character just typed. It inserts the ASCII character w in the display list at the current cursor location.

   (If "lowercase w" were programmed to mean HI MOM, the Processor would have inserted HI MOM in the display list.)

6. While the Processor is busy with other tasks, the Tracker reads the display list. When the Tracker comes to the line containing the newly-typed character, it sends that character, along with the rest of that line, to the Display Controller. To do this, it uses the Mother Board's paired bus lines. That way, it doesn't interfere with the Processor's use of the main terminal address and data busses.

7. When the Display Controller is finished scanning the previous row of text, it starts on the text row just received from the Tracker. For each character to be displayed, it looks up (in its Character Set Memory) that character's dot pattern. Using this information, it generates a VIDEO signal, and sends that signal (together with the synchronizing signals DHDRIVE and DVDRIVE) to the Video Display Unit.

8. In response to the VIDEO, DHDRIVE and DVDRIVE signals, the Video Display Unit displays the text on its screen, including the newly-inserted letter "w."

Figure 2-5 summarizes this process; it shows the flow of information from the keyboard to the screen.

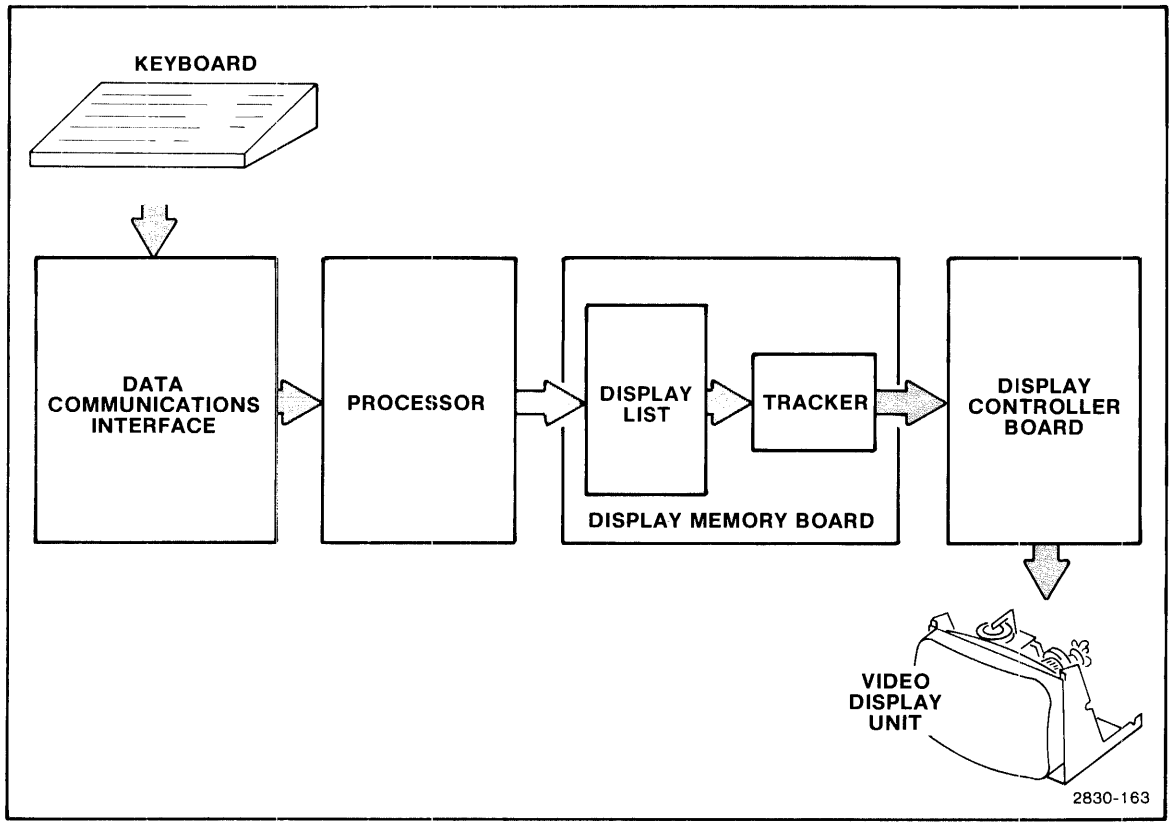


Figure 2-5. Flow of Information from Keyboard to Screen.

VIRTUAL BIT MAP GRAPHICS

Suppose the user wants to draw graphs in an area which occupies all 80 columns of 34 lines of text. Such a graphic region includes 304,640 addressable points. The usual "bit map" method of storing this data would require 304,640 bits: 38,080 eight-bit words of memory.

To save memory, the 4025 uses a "virtual bit map" method of storing graphic information. This method takes advantage of the fact that most pictures and graphs have many blank areas.

For each 8-dot by 14-dot character position ("graphics cell"), there are two bytes in the display list. (This is the same display list that holds all the other text in the workspace and monitor scrolls of memory.) The first byte names an alternate character font, and the second byte specifies one of 128 characters in that font.

When the Tracker reads the display list, it tells the Display Controller, for each graphics cell, which character of which character font to display.

When the graphics region is first created, all its cells are "character number 32 of font number 0"—in other words, ASCII spaces. The Display Controller already knows how to display ASCII spaces, so no additional graphics memory is needed.

To draw a line in the graphics region (execute a VECTOR command), the Processor first determines through which graphics cells the line will pass. It then assigns characters of alternate character fonts to these cells, and tells the Graphics Memory Board how those characters should be drawn. (That is, it specifies which dots should be turned on in each cell's dot matrix.)

Only those character cells through which lines are drawn have their dot patterns stored on the Graphics Memory Board. Thus, a graph with few lines requires little Graphics Memory. A more complex graph, with lines passing through more graphics cells, would require more Graphics Memory.

Section 3

VIDEO DISPLAY UNIT, POWER SUPPLY, AND MOTHER BOARD

This section describes:

- The 4024/4025 Video Display Unit and its power supply.

NOTE

Theory of operation for the Video Display Unit is for all serial numbers. However, for 4024 terminals serial numbered B031219 and up and 4025 terminals serial numbered B054354 and above the partial circuit diagrams shown are only approximations. For a complete circuit diagram for these terminals, see schematic diagram 3-2 in Volume 2 of this service manual.

- The main power supply for the terminal, which is separate from the Video Display Unit's supply.
- The terminal's Mother Boards, which differ only in the number of edge connectors installed.

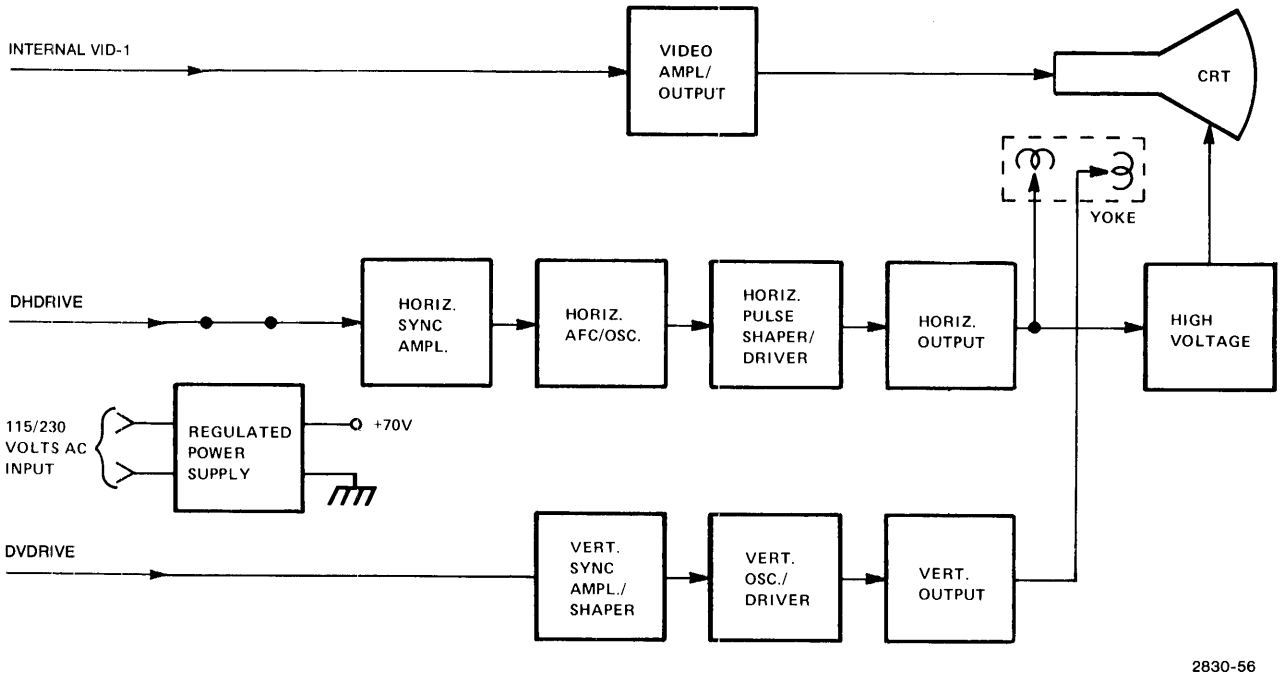


Figure 3-1. Video Display Unit Block Diagram.

VIDEO DISPLAY UNIT

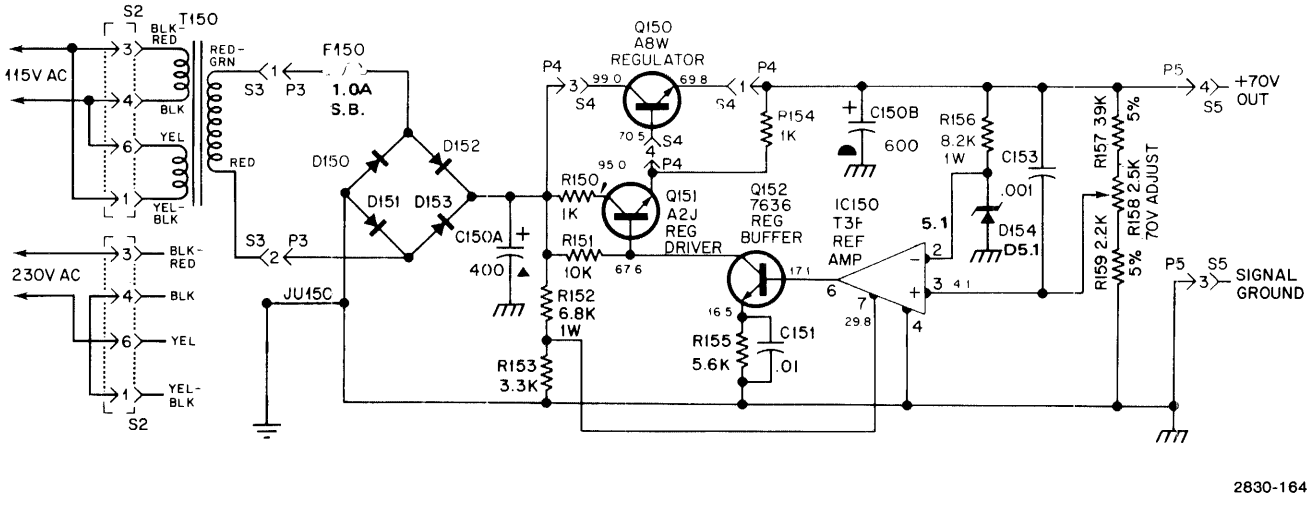
The Video Display Unit's block diagram is shown in Figure 3-1. Detailed schematic diagrams are included with the circuit descriptions; however, for the most complete and accurate schematics and parts lists, see Volume 2 of this Service Manual.

The Display Unit includes the following circuits: Display Power Supply, Video Amplifier, Horizontal Sync Amplifier, Phase Detector, Horizontal Oscillator, Horizontal Pulse Shaper and Driver, Horizontal Output, Vertical Sync Amplifier, Sync Shaper, Vertical Oscillator, Vertical Driver, and Vertical Output. Each is described in turn.

Display Power Supply (See Figure 3-2.)

The Power Supply is a transformer-operated, full wave, regulated series pass circuit that maintains a constant output voltage with line input variations of ±12.5%. Depending on how connector S2 is wired, operation from 115 or 230 volts, 50/60 Hz is possible. Integrated circuit IC150 is the reference amplifier, transistor Q152 is a regulator buffer, transistor Q151 is the regulated output driver, and Q150 is the series pass transistor.

The output voltage, + 70V, appears at the emitter of Q150. This voltage is divided between R157, R158 and R159. The voltage on the arm of potentiometer R158 (70V ADJ control) is the reference input to the non-inverting input of reference amplifier IC150.



2830-164

Figure 3-2. Display Power Supply.

A temperature-compensated zener diode, D154, establishes a fixed reference voltage at the inverting input to IC150. Resistor R156 provides a bias current for D154, which establishes the operating point of D154. Capacitor C153 is a high-frequency filter. Operating voltage for IC150 is derived from a voltage divider consisting of R152 and R153. Components R155 and C151 set the voltage gain of Q152.

An increase in output voltage results in an increase of voltage at the base of Q152 via the non-inverting input of IC150. The change in bias voltage turns Q152 on more, reducing its collector voltage. This reduces the forward bias to Q151, which results in less emitter current for Q150. With Q150 conducting less, the output voltage is lowered.

Dual-section capacitor C150 provides filtering.

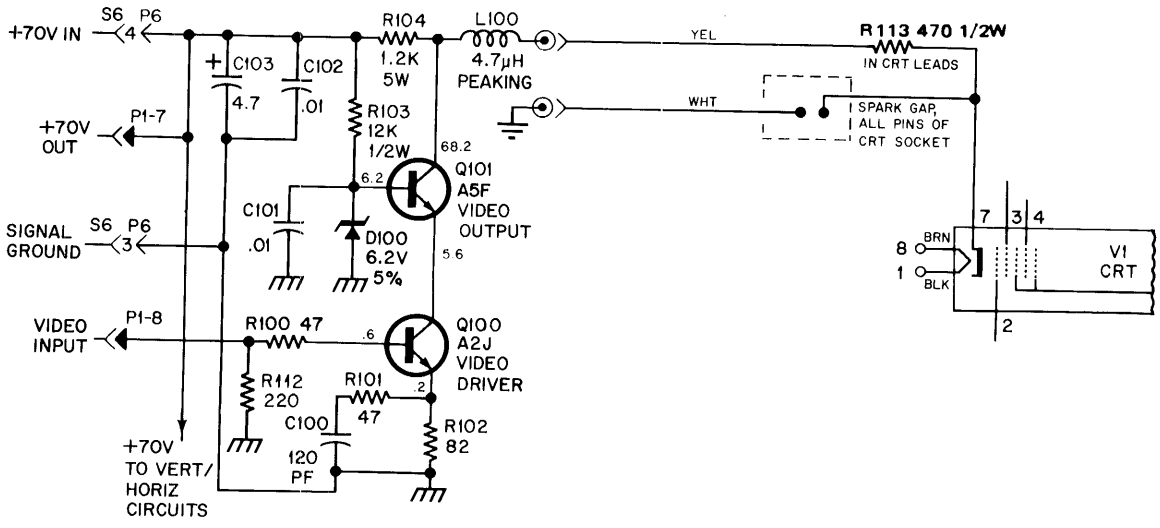
Video Amplifier

(See Figure 3-3.)

The Video Amplifier consists of two stages, Q100 and Q101, connected in a cascode configuration.

A TTL compatible non-composite video signal, approximately 4.0 volts P-P, is dc coupled to the base of Q100 via R100. Resistor R112 provides proper termination for the high-frequency compensation to maintain a flat response when Q100 and Q101 conduct.

During no-signal conditions, Q100 is off. Transistor Q101, however, is forward biased by the 6.2 volts on its base, which is established by zener diode D100. When a video signal is applied to the base of Q100, it conducts, which causes forward biased Q101 to conduct. The resultant output is developed across R104 at the collector of Q101; then is dc-coupled to the cathode of V1 (CRT) via peaking coil L100 and R113. Resistor R113 isolates Q101 from transients that may occur as a result of crt arcing. Capacitor C101 shunts to ground any high frequency video that may appear on the base of Q101. Peaking coil L100 boosts the high frequencies of the video signal. Capacitor C103 provides additional filtering of the + 70V power supply lines, while C102 is a high-frequency AC bypass capacitor.



2830-165

Figure 3-3. Video Amplifier.

Horizontal Sync Amplifier

(See Figure 3-4.)

The Horizontal Sync Amplifier consists of one stage, Q50, which operates as a switch. During a no-signal condition, Q50 is off. When a positive-going horizontal sync signal, approximately 4.0 volts P-P, is applied (dc coupled) to the base of Q50, it goes into saturation. The amplified output is developed across load resistor R51, approximately 35V, which forms a voltage divider with R77. The negative-going horizontal sync pulses are ac coupled to the phase detector circuit via the R-C network consisting of R52 and C68, a high-frequency pass filter.

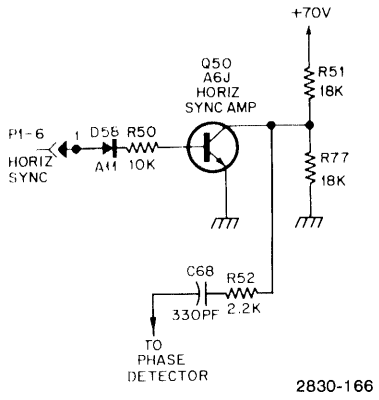


Figure 3-4. Horizontal Sync Amplifier.

Phase Detector

(See Figure 3-5.)

The Phase Detector consists of two diodes (D50 and D51) in a keyed clamp circuit. Two inputs are required to generate the required output, one from the Horizontal Sync Amplifier, Q50, and one from the Horizontal Output circuit, Q54. The required output must be the proper polarity and amplitude to correct phase differences between the input horizontal sync pulses and the horizontal time base.

The Horizontal Output (Q54) collector pulse is integrated into a sawtooth by R56 and C69. During horizontal sync time, diodes D50 and D51 conduct, which shorts C69 to ground. This effectively clamps the sawtooth on C69 to ground at sync time. If the horizontal time base is in phase with the sync (waveform A), the sync pulse occurs when the sawtooth is passing through its AC axis and the net charge on C69 will be zero (waveform B). If the horizontal time base is lagging the sync, the sawtooth on C69 is clamped to ground at a point negative from the AC axis. This will result in a positive DC charge on C69 (waveform C). This is the correct polarity to cause the Horizontal Oscillator to speed up to correct the phase lag. Likewise, if the horizontal time base is leading the sync, the sawtooth on C69 will be clamped at a point positive from its AC axis. This results in a net negative charge on C69, which is the required polarity to slow the Horizontal Oscillator (waveform D).

Components R55, C52, R58 and C53 comprise the Phase Detector filter. The bandpass of this filter is chosen to provide correction of Horizontal Oscillator phase without ringing or hunting. Capacitor C50 times the phase detector for correct centering of the picture on the raster.

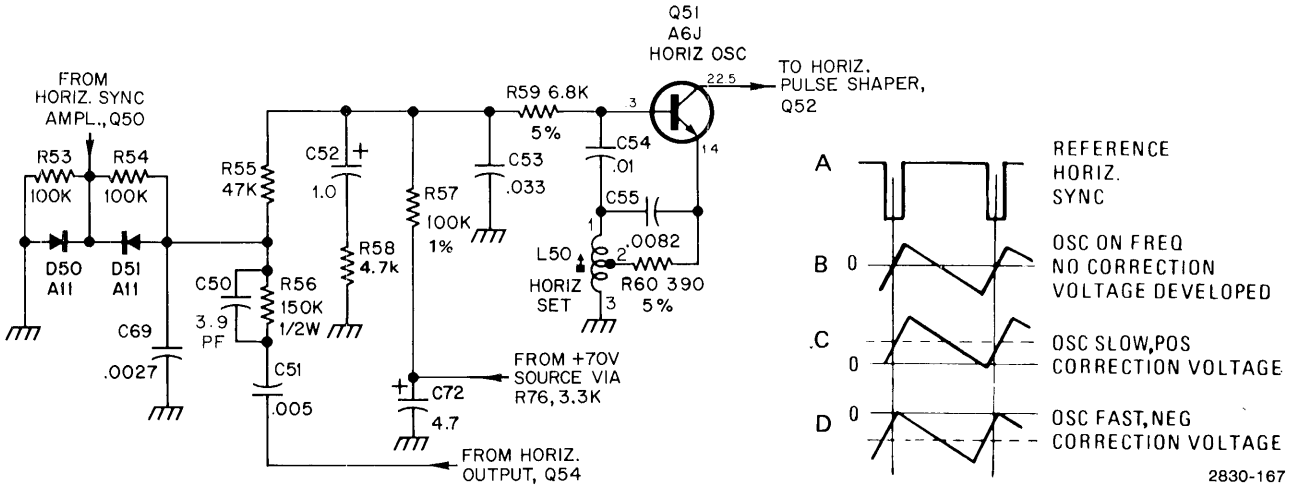


Figure 3-5. Phase Detector and Horizontal Oscillator.

Horizontal Oscillator

(See Figure 3-5.)

The Horizontal Oscillator consists of Q51, which is employed as a modified type of Hartley oscillator. The operating frequency of this oscillator is sensitive to its base input voltage. This permits control by the output of the phase detector. Resistor R57 provides DC bias to turn on Q51 and start the oscillator. The free-running horizontal frequency is adjusted with the HORIZ.SET coil, L50. This coil and capacitor C54 are the frequency determining components. Capacitor C55 and resistor R60 are feedback components for the oscillator circuit.

Horizontal Pulse Shaper and Driver

(See Figure 3-6.)

Transistor Q52 is a buffer stage between the Horizontal Oscillator and Horizontal Driver. It provides isolation for the Horizontal Oscillator as well as a low impedance drive for the Horizontal Driver. Components R62 and C56 form a time constant that shapes the oscillator output to the required duty cycle, approximately 50%, to drive the Horizontal Output circuitry.

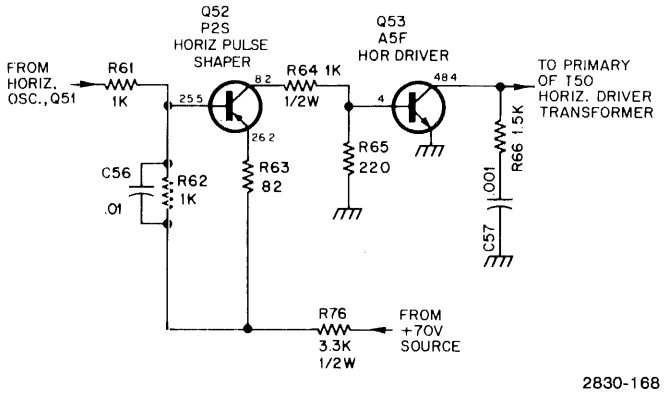


Figure 3-6. Horizontal Pulse Shaper and Driver.

The Hozontal Driver stage, Q53, operates as a switch to drive the Horizontal Output transistor (Q54) through T50. Because of the low impedance drive and fast switching times furnished by Q52, very little power is dissipated in Q53. Components R66 and C57 provide damping to suppress ringing in the primary of T50 when Q53 goes into cutoff. (Reference Figure 3-1; Resistor R68 provides current limiting for Q53 while C58 is an ac bypass capacitor.)

Horizontal Output

(See Figure 3-7.)

The secondary of T50 provides the required low drive impedance for Q54. Components R67 and C59 form a time constant for fast turn-off of the base of Q54. Once during each horizontal period, Q54 operates as a switch that connects the supply voltage across the parallel combination of the horizontal deflection yoke and the primary of the high-voltage transformer. The required sawtooth deflection current (through the horizontal yoke) is formed by the L-R time constant of the yoke and primary winding of the H.V. transformer, T51. The horizontal retrace pulse charges C62 through D54 to provide operating voltage for G2 of the crt. Momentary transients at the collector of Q54, should they occur, are limited to the voltage on C62 since D54 will conduct if the collector voltage exceeds this value.

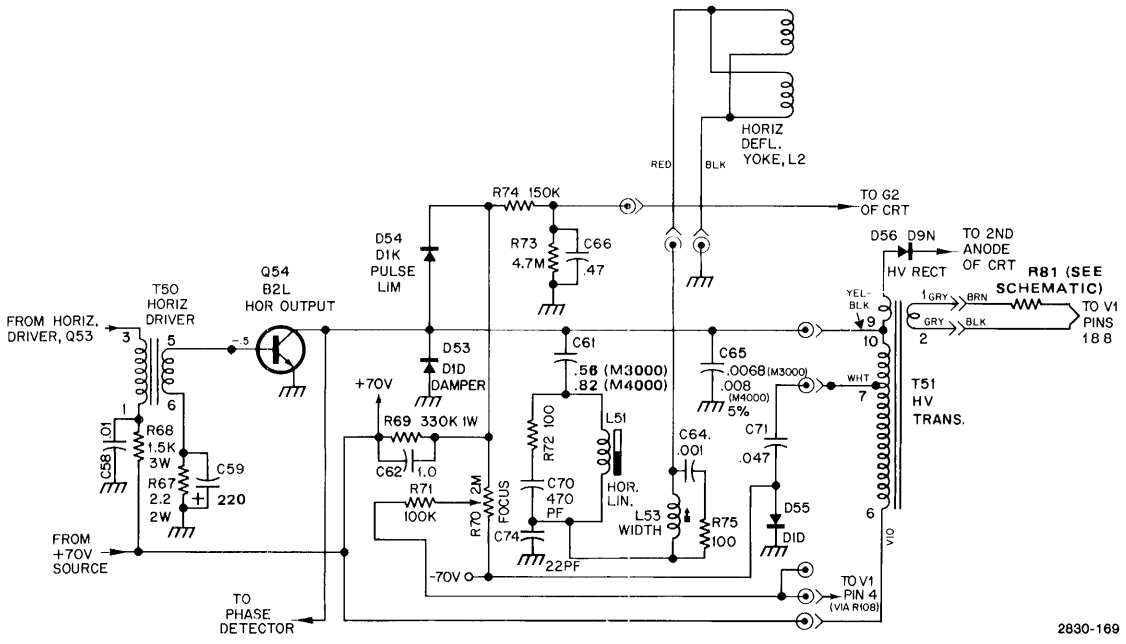


Figure 3-7. Horizontal Output Circuit.

The damper diode, D53, conducts during the period between retrace and turn on of Q54. Capacitor C65 is the retrace tuning capacitor, while C61 blocks dc from the deflection yoke. Coil L51 is a magnetically biased linearity coil that shapes the deflection current for optimum trace linearity. Coil L53 is a series HORIZ.WIDTH control. Components R72 and C70, C64 and R75 are damping network components for the horizontal linearity (L51) and width (L53) controls. Capacitor C71 couples horizontal sync pulses from Pin 7 of T51 to diode clamp D55, which maintains the -70V reference voltage.

Vertical Sync Amplifier

(See Figure 3-8.)

The Vertical Sync Amplifier consists of one stage, Q1, which operates as a switch. During no-signal conditions, Q1 is off. When a positive-going vertical sync signal, approximately 4.0 volts P-P, is applied (direct coupled) to the base, Q1 goes into saturation. The amplified output is developed across load resistor R3 to approximately 11 volts. Jumpers JU3, JU4, and JU5 are inserted, depending on the polarity of the input vertical sync pulse: TTL NEG for negative-going and TTL POS for positive-going.

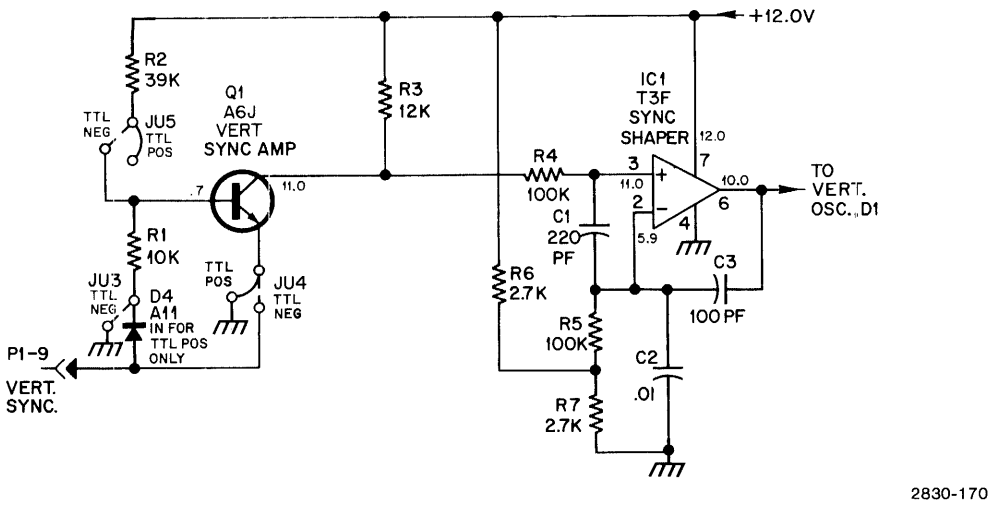


Figure 3-8. Vertical Sync Amplifier and Sync Shaper.

Sync Shaper

(See Figure 3-8.)

The negative-going vertical sync pulses (from Q1) are direct coupled to the non-inverting input of the Sync Shaper stage, IC1. The combined action of an integrating network, consisting of C1, C2, C3, R5, R6, and R7, removes high-frequency noise from the vertical sync pulses. Capacitor C3 performs the actual integrating, while resistors R5-R7 provide biasing for IC1. Capacitors C1 and C2 provide a bypass function.

Vertical Oscillator

(See Figure 3-9.)

The negative-going vertical sync pulses are ac coupled (C4) to the gate of a programmable unijunction transistor device, D1. This device turns on with each negative-going sync pulse applied to its gate. This action permits C6 and C7 to discharge very rapidly and then recharge slowly during the period that a sync pulse is not applied to the gate. The recharge path for C6 and C7 is through R12 and R13. As soon as the next sync pulse is applied to the gate of D1, C6 and C7 discharge very rapidly again. This sequence of events produces a positive-going ramp, or sawtooth waveform, at the anode of D1.

When no vertical sync pulses are connected to the monitor, Vertical Oscillator D1 is kept free-running to maintain a raster on the crt. This is accomplished by biasing the gate of D1 in conjunction with the charge and discharge action of C6 and C7. Resistors R9 and R8 provide the proper bias for D1, which also determines the repetition rate for the charge and discharge action of C6 and C7.

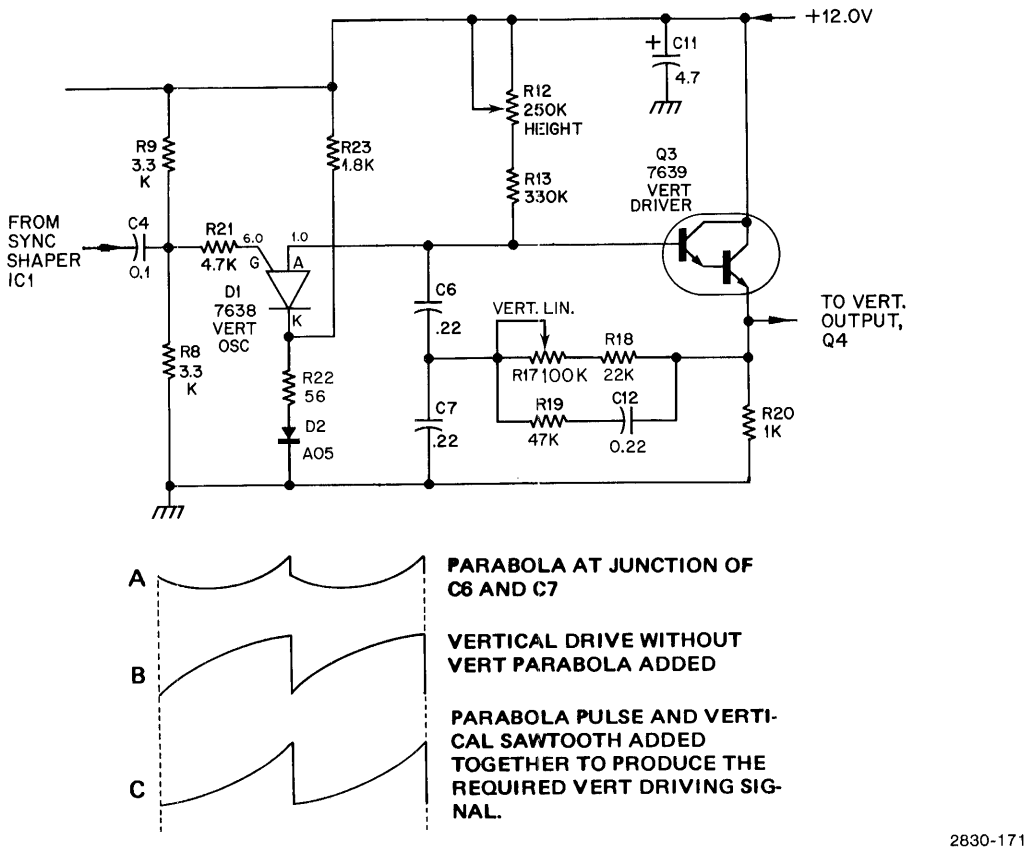


Figure 3-9. Vertical Oscillator/Driver.

In addition, during no-signal conditions, components R22 and D2 (in conjunction with D1) provide a small incremental voltage above ground to compensate for the base-emitter voltage drop of the vertical driver, Q3. This is necessary to keep the Vertical Output stage, Q4, from being driven into cutoff, which could distort the vertical linearity.

Vertical Driver

(See Figure 3-9.)

The positive-going sawtooth waveform from the anode of D1 is direct coupled to the base of Vertical Driver Q3, which operates as an emitter follower. The sharp fall time of the sawtooth is a result of the rapid discharge of C6 and C7 through D1. The amplitude of the sawtooth is varied with the HEIGHT control, R12.

The output sawtooth from the emitter of Q3 is direct coupled to the base of Vertical Output stage Q4. Part of this sawtooth waveform, however, is also coupled back to the junction of C6 and C7 via R18 and VERT.LIN control R17 for proper shaping. Since this path is resistive, the waveform is integrated into a parabola waveform by C7 (waveform A). This results in a predistortion of the drive sawtooth (waveform C). (Waveform B illustrates the drive sawtooth without parabola shaping.) Parabola shaping is necessary to compensate for the non-linear charging of C6 and C7. An additional path for phase compensation is provided through C12 and R19.

Vertical Output

(See Figure 3-10.)

The positive-going sawtooth waveform from the emitter of Q3 is applied to the base of Vertical Output stage Q4, which conducts only during the ramp, or rise time, of the sawtooth waveform. The inverted ramp output (collector current) is the vertical trace period, which is ac coupled (via C14) to the vertical deflection yoke winding. The same collector current output is also applied to L1, which builds up a large electromagnetic field. This field collapses very rapidly when Q4 turns off during the retrace time of the waveform applied to the base of Q4. The back emf is in the form a high-voltage positive pulse, whose duration represents the vertical retrace period. To limit the pulse to a safe value, a varistor is connected across L1, with R26 providing damping.

Except for the vertical output stage, Q4, the vertical circuitry operates from a + 12 volt source, which is derived from the + 70V source. Resistor R27 drops the + 70V source to the required + 12V. Zener diode D3 holds the + 12V constant while C11 provides additional filtering.

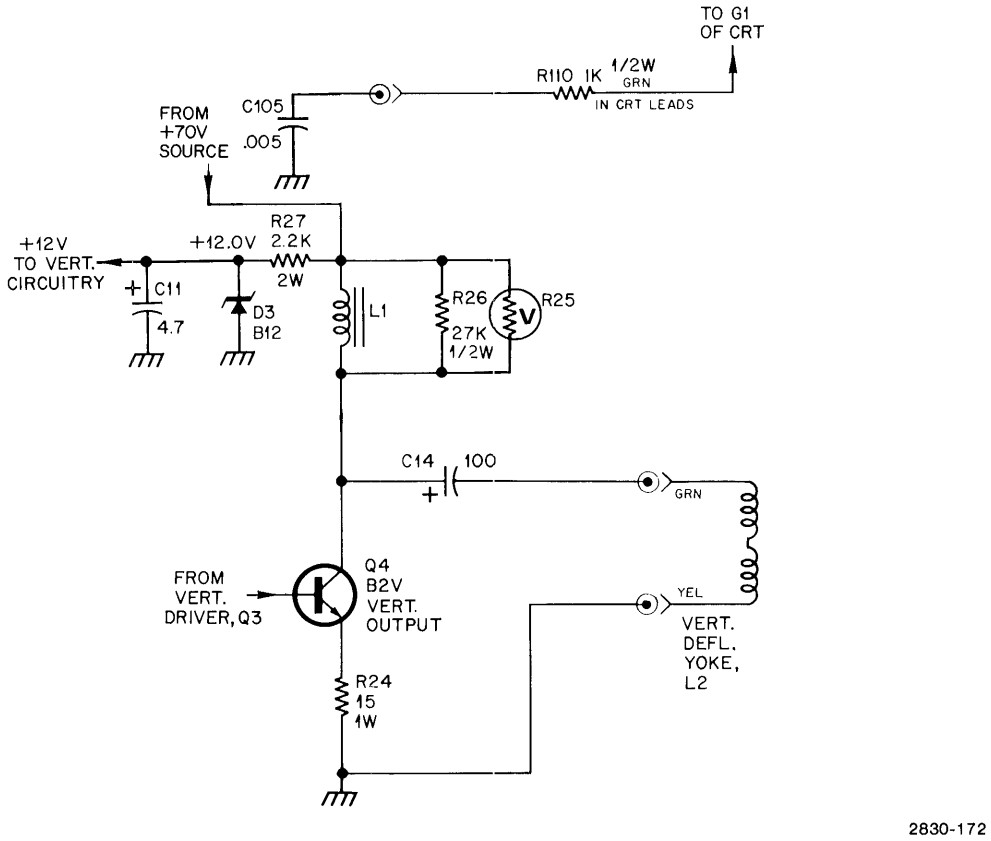


Figure 3-10. Vertical Output Circuit.

Dynamic Focus Adjustment (4024, B031219 and up; 4025, B054354 and up only)

Due to the geometry of the CRT, the electron beam travels a greater distance when deflected to a corner than it does to the center of the CRT screen. As a result, optimum focus is obtained at only one point. For most applications, an adequate adjustment is realized by setting the focus while viewing the screen at some point midway between the center of the screen and a corner. Still better focus can be had by modulating the focus voltage at a horizontal sweep rate. By doing so, optimum focus voltage is made variable along the horizontal axis of the CRT, and this compensates for the beam travel along this axis.

The AC component focus voltage is developed by the series resonant circuit, L52 and C63 (See Figure 3-12). This 80V P-P horizontal rate pulse is coupled from a tap on the horizontal output transformer, T51, via C67. The DC component of this voltage is set by FOCUS control R70. When the DYNAMIC FOCUS coil, L52, is set for best edge focus, a sinusoidal voltage of approximately 200V P-P is developed across C63. This DC voltage with an AC component is coupled through isolating resistor R108 to the CRT focus anode.

TERMINAL POWER SUPPLY

The power supply (Schematic 1-1) provides dc voltages to run the plug-in circuit boards and ac voltages for the Video Display Unit and the fan. (The fan is omitted in the 4024.)

Primary Windings

Main power transformer T1000 is tapped to operate on different line voltages. Its windings are in parallel on the lower voltage ranges and in series on the higher ones. Straps on a barrier strip set the unit for different mains voltages; the strap settings are described in Volume 2 of this service manual, under "Line Voltage Selection."

The fan is connected across trasformer taps 1 and 3, and the Video Display Unit's two primary windings across taps 1 and 6, and 4 and 7. Autotransformer action keeps the voltages across these taps the same, regardless of the voltage range for which the terminal is set. Thus, when changing from one line voltage range to another, it is not necessary to change connections in the Video Display Unit power supply or in the fan.

Power Fail Detector

The Power Fail Detector monitors the voltage at the secondary windings of T1000. When power fails, this voltage drops, turning off Q35 and turning on Q123, which shorts the PWDN line to ground. The low on PWDN provides an early warning that the power source has failed. This enables the Processor to disable the CMOS Battery RAM before the RAM's power source is transferred to the battery.

+12 Volt Supply

The + 12 V supply uses voltage regulator IC U123 (a μA723 or equivalent). This IC has two independent parts: an error amplifier with series pass transistor, and a 7.15 V voltage reference. The difference between the voltage on the + 12 V sense line (fed to Pin 4 of the IC) and the reference voltage (Pin 5) is amplified by the IC and Q241, and used to control series pass element Q1105.

The voltage across R208 is a measure of the current being drawn from the supply. This voltage is sensed by the voltage regulator IC, which makes the series pass transistor limit the current at about 1.8 A.

SCR Q125 provides "crowbar" protection; excessive voltage on the + 12 V line fires the SCR, which shorts the + 12 V output to ground, blowing fuse F209.

+5.1 Volt Supply

Pin 6 of U123 provides a reference voltage of 7.1 V, which is tapped down by a voltage divider. R258 in that divider is adjusted so that the voltage at its wiper is 5.1 V, the reference voltage for the + 5.1 V supply.

The error voltage (the difference between the reference voltage and the voltage on the + 5.1 V sense line) is amplified by U205, Q109, and Q1110, and used to control series pass transistor Q1112. Note that this transistor is in series with the *negative* lead of bridge rectifier CR1006.

The voltage across R1112 is a measure of the current being drawn from the supply. This voltage is amplified by Q38 and Q29 and used to bias Q110. When too much current is drawn from the supply, Q110 turns on, causing Q101 to think that the error voltage has gone positive. Consequently, Q101 and Q1110 shut down series pass transistor Q1112. This provides current limiting. R47 biases Q29, setting the current at which the supply self-limits; this should be set at 10.5 A.

Q1014 provides crowbar protection; should excessive voltage appear on the + 5.1 V output line, Q1014 shorts that output to ground, blowing fuse F1007.

−12 Volt Supply

The − 12 V supply uses the same reference voltage as the + 5.1 V supply. Currents derived from this reference voltage and from the − 12 volts sense line are added at U55 non-inverting input. R45 and R46 are chosen so that when the − 12 V sense line is indeed at − 12 V, the junction of R45 and R46 is at 0.00 volts. Any error in the sense line voltage is amplified by U55 and Q357 and used to control series pass transistor Q1118.

The voltage across R58 and R356 is a measure of the current being drawn from the supply. If more than about 800 mA is drawn, Q55 turns on, turning off Q357 and Q1118.

−5 Volt Supply

The − 5 V sense line and the + 5.1 V reference voltage are compared in U52, which controls series pass element Q53. This supply obtains its power from the − 12 V supply. It is protected from excessive current by that supply's current limiting and by R56. If the − 5 V supply's current exceeds 200 mA, the voltage across R56 causes the − 5 V output to decrease in voltage.

MOTHER BOARD

The Mother Board (Schematic 2-1) holds the 4024/4025 bus and the sockets into which other boards are inserted.

Signal Lines

Figure 3-11 shows the connector pin identification, and Table 3-1 lists the signal lines on the Mother Board.

Flip Side			Component Side
GND	1-	- 2	GND
-5V	3-	- 4	-5V
SPARE	5	6	SPARE
-12V	7-	- 8	-12V
SPARE	9	10	SPARE
12V	11-	- 12	12V
SPARE	13	14	SPARE
+5V	15-	- 16	+5V
RDIS	17@	18	SPARE
BA0	19#	#20	BA1
BA2	21#	#22	BA3
BA4	23#	#24	BA5
BA6	25#	#26	BA7
BA8	27#	#28	BA9
BA10	29#	#30	BA11
BA12	31#	#32	BA13
BA14	33#	#34	BA15
WACK	35#	$36	RESET
PWDN	37-	%38	IOADR
READ	39#	#40	WRITE
WAIT	41@	@42	BRQ
BGIN	43*	*44	BGOUT
IA0	45%	%46	IA1
IA2	47%	@48	IRQ
LCLK	49%	%50	HCLK
PB0	51*	*52	PB1
PB2	53*	*54	PB3
PB4	55*	*56	PB5
PB6	57*	*58	PB7
PB8	59*	*60	PB9
PB10	61*	*62	PB11
BD0	63#	#64	BD1
BD2	65#	#66	BD3
BD4	67#	#68	BD5
BD6	69#	#70	BD7
GND	71-	-72	GND

Designation

- \- Power Supply
- % Processor board driven, no pull up
- $ Processor board driven, pull up through 2.7K to +4.3V.
- # Tri-state, pull up through 2.7K to +4.3V.
- @ Open collector, pull up through 1K to +4.3V.
- * Non-bused signals (signal is not necessarily the same at every connector).

2830-173

Figure 3-11. Bus Connector Pin Identification.

Table 3-1

MOTHER BOARD SIGNAL LINES

Name	Pin No.	Description
P/S Lines	1-16	Pins 1 and 2 carry grounds. The power supply lines have spare lines interspersed between them to prevent shorts if the boards are accidentally removed or replaced with the power on.
RDIS	17	The RDIS (ROM Disable) line is normally high. Putting this line low disables all ROMs (Read Only Memories) on the bus.
BA15-BA0	19-34	The 16 lines BA0 (Bus Address line 0) to BA15 carry the 16 address bits used by the 8080 microprocessor to specify any word in its memory. BA0 is the least significant bit, BA15 the most significant.
WACK	35	The Processor or DMA (Direct Memory Access) device sends a WACK (Wait Acknowledge) signal to the device sending the WAIT signal. The WACK signal indicates that the WAIT has been received.
RESET	36	The RESET line is driven low by the Processor Board during power-up or when the MASTER RESET button is pressed. This signal resets all boards on the bus to a known initial state.
PWDN	37	The PWDN (Power Down) signal is a notification from the Power Supply that primary power has been lost.
IOADR	38	The IOADR (Input/Output Address) line, when in the high state, indicates that the address on the bus is in the range 0800 to 0BFF. This is the section of memory reserved for device registers.
READ	39	The READ signal indicates the start of a read operation.
WRITE	40	Similar to the READ signal, but used for write operations.
WAIT	41	The WAIT line, when low, tells the Processor that the addressed device cannot complete the required operation at the moment. As soon as the device can complete the operation, it should release this line.
BRQ	42	Not used. (Reserved for DMA operations.)
BGIN, BGOUT	43-44	The BGIN (Bus Grant In) and BGOUT (Bus Grant Out) lines are not used. (Like BRQ, these lines are reserved for future DMA devices.)
IA0-IA2	45-47	The IA0 (Interrupt Address 0), IA1, and IA2 lines are part of the interrupt servicing hardware. They indicate, by a three-bit binary number, the level of interrupt that is being polled. The 4025 Processor Board uses these lines, but the 4024 does not; consequently, on the Display Memory Board, which is used by both the 4024 and the 4025, there is a strap option which can be set to ignore these signals.
IRQ	48	A circuit board pulls the IRQ (Interrupt Request) line low when it needs to generate an interrupt and when the IA2-IA0 lines are at its selected interrupt level.
LCLK	49	The LCLK (Low Frequency Clock) line carries the inverse of the phase 2 clock of the 8080 microprocessor. It is at 2.048 MHz with a 45% duty cycle.

Table 3-1 (cont)

MOTHER BOARD SIGNAL LINES

Name	Pin No.	Description
HCLK	50	HCLK (High Frequency Clock) is an 18.432 MHz clock.
PB11-PB0	51-62	The twelve "paired bus" lines do not run the entire length of the bus; instead, they connect pairs (in one case, a triplet) of adjacent boards. These lines may be used to connect pairs of boards inserted in adjacent slots. Note: So far, only the Display Memory Board and the Display Controller Board communicate with each other via the paired bus.
BD7-BD0	63-70	The lines BD0 (Bus Data line number 0) to BD7 make up the eight-bit data bus. BD0 is the least significant bit; BD7 is the most significant bit. These lines are driven by whichever device is currently placing data on the bus. They are driven by the Processor Board only during Processor write operatons.
GND	71-72	Common ground connections for all circuit boards.

Section 4

KEYBOARD

The keyboard (Schematic 4-1) is connected to the Data Communications Interface's Keyboard Port by a thin, flexible, 8-conductor cable. Its circuitry has two major functional blocks: the Key Scanner and the Lights and Bell Circuitry.

KEY SCANNER

The Key Scanner (Figure 4-1) tells the communications interface which keys are being pressed. A CLOCK signal from the communications interface drives a 7-bit binary ripple counter. The four low-order bits are used by a 74L154 to scan the 16 columns of the keyswitch matrix. Likewise, the three high-order bits are used by a 74LS151 to scan the eight rows of the matrix. Thus, each keyswitch is queried once during each 128-count cycle of the counter.

For instance, the W switch is queried on the count of 35 (binary 0100011), when the fourth column (74LS154 output D3) and third row (74LS151 input D2) are both selected. If the switch is closed, the low on the column line drives the D3 row input of the 74LS151; the 74LS151 Pin 6 then goes high, driving the KDATA (Key Data) line low. The fact that it is the 35th count when KDATA goes low tells the communications interface that it is the W switch which is being pressed.

Once each 128 CLOCK pulses, the communications interface clears the counter with a SYNC pulse. This ensures that the counter is always in step with a similar counter in the communications interface.

LIGHTS AND BELL

The Lights and Bell circuitry (Figure 4-2) circuitry receives a multiplexed LDATA (Light Data) signal from the communications interface, together with SCLCK (Shift Clock). LDATA is demultiplexed and used to drive the lights in the four lighted function keys, or to sound the bell.

SCLK clocks the LDATA signal into U5, a 74LS164 shift register. Figure 4-3 shows SCLK's waveform; it consists of eight clock pulses, followed by a period (equal in duration to those eight pulses) during which SCLK is turned off. During the eight clock pulses, the open-collector output of inverter U96G is repeatedly driven low, and there is not enough time between pulses for capacitor C4 to charge. After the string of pulses, SCLK remains off for a longer period; C4 then charges and turns on U10E, clocking the type D latches in U6.

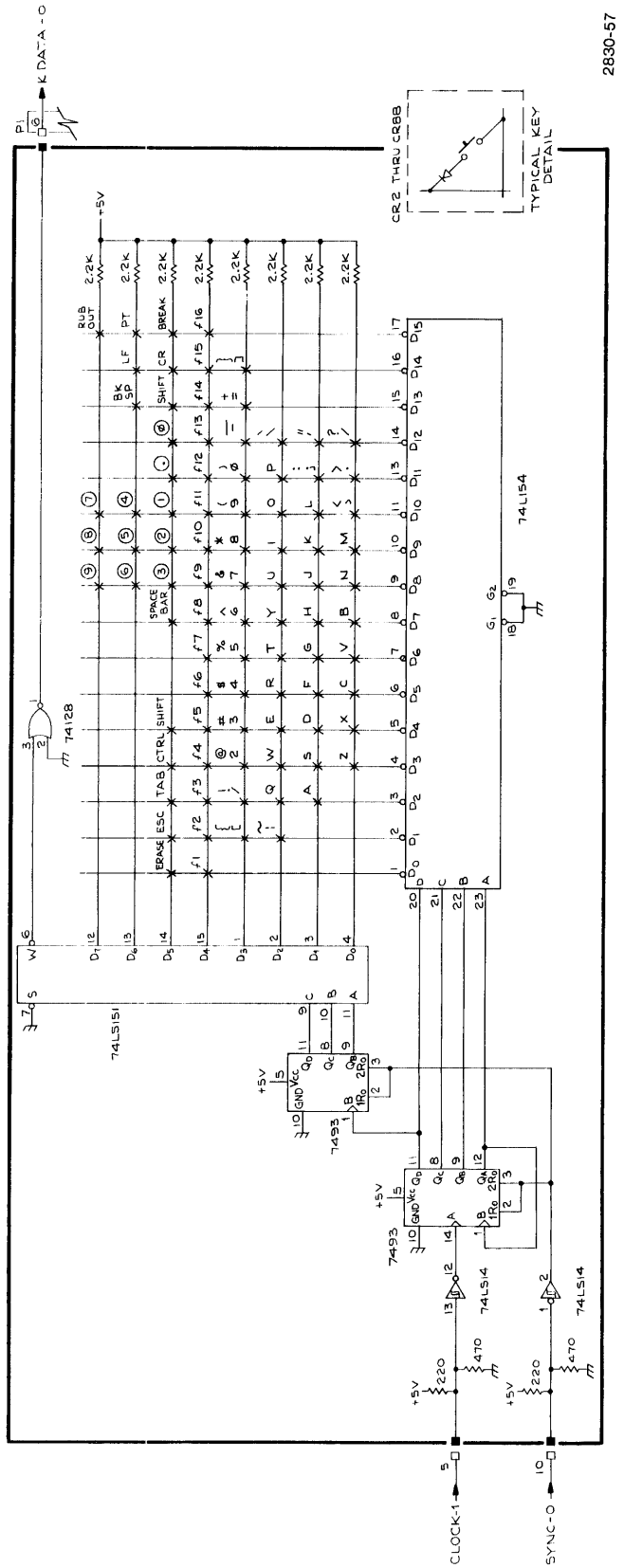


Figure 4-1. Key Scanner.

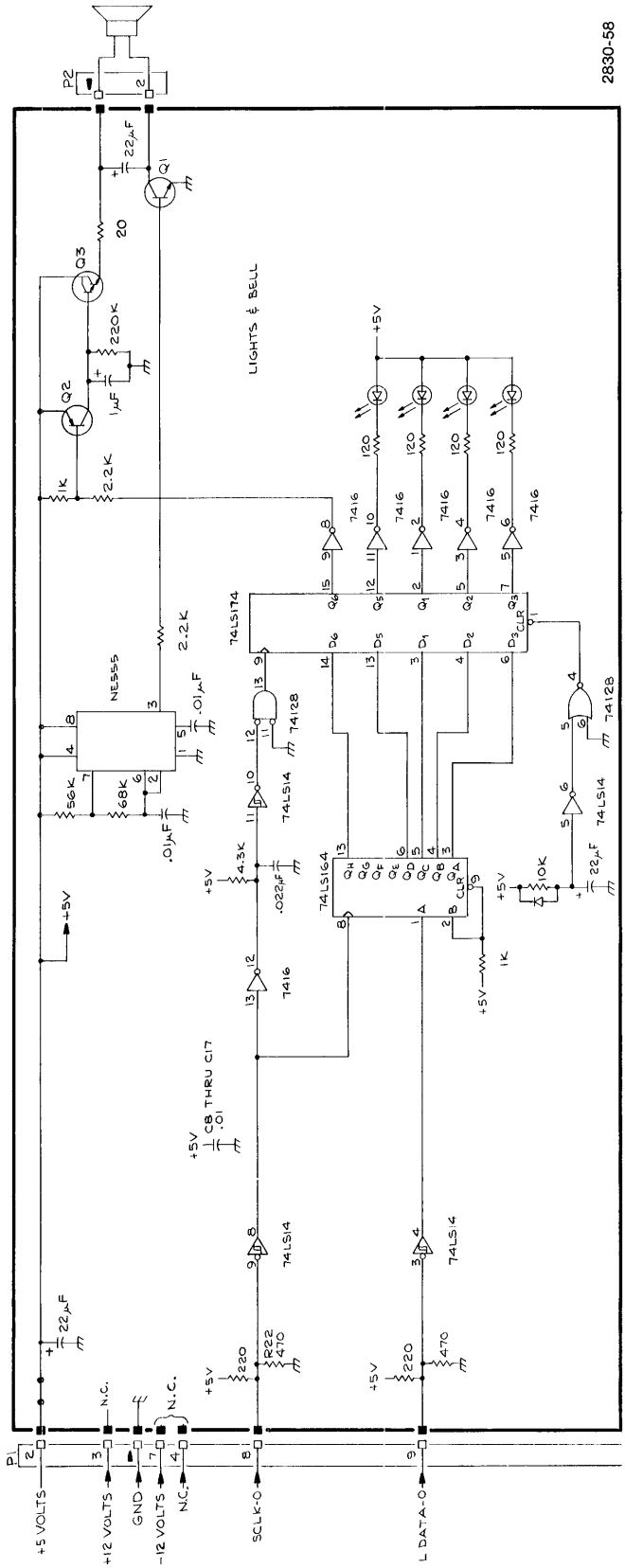


Figure 4-2. Lights and Bell Circuitry.

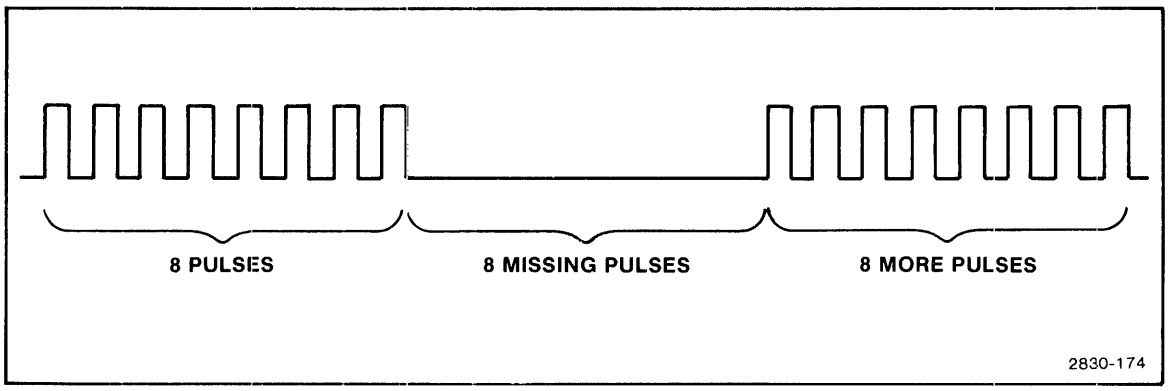


Figure 4-3. SCLK Waveform.

Each group of eight SCLK pulses, then, clocks an eight-bit serial LDATA word into the shift register. Gaps between the pulse groups clock the parallel outputs of the shift register into the type D latches in U6. The latch outputs drive the four LEDs in the lighted function keys and control the bell.

The bell is an oscillator which runs all the time. The D6 output of U6 keys the oscillator output, turning the speaker on and off. When the speaker is turned on, C5 charges. When it is turned off, C5 discharges gradually, rather than abruptly. This gives a pleasant bell-like quality to the tone.

Section 5

4025 PROCESSOR BOARD

INTRODUCTION

The 4025 Processor Board holds the "brain" of the 4025: the 8080 microprocessor and the system firmware for that microprocessor.

In the 4025, the system firmware occupies four ROMs on the Processor Board. The microprocessor instructions in these ROMs enable the terminal to perform its standard functions. However, if options are installed, the microprocessor may also need instructions for handling those options. For most options, the corresponding optional firmware occupies ROMs on the ROM Expansion Board. For instance, if Graphics Memory (Options 23 to 26) is installed, a Graphics ROM is installed on the ROM Expansion Board. Likewise, if the GPIB Interface (Option 4) or the RS-232 Peripheral Interface (Option 3) is installed in the 4025, then the Peripherals ROMs (Option 36) must be installed on the ROM Expansion Board.

Comparison of 4024 and 4025 Processors

The 4025 Processor Board is described in this section of the manual, and the 4024 Processor/Comm Board in Section 7. The processors have many similarities. For instance, both boards use the 8080A microprocessor, and both include a Fix-It. (The Fix-It is a device for making updates to to system firmware.)

However, the two boards also differ in several ways. The 4025 Processor Board includes *System RAM* (read/write memory for use as a "scratchpad" by the microprocessor). The 4024 Processor/Communications Board lacks this RAM; for its scratchpad, it uses RAM on the Display Memory Board.

Also, the 4024 and the 4025 differ in how they handle *interrupts* (requests by peripheral devices for the Processor's attention). In the 4025, a *vectored interrupt* scheme is used. Each peripheral device (such as RS-232 Peripheral Interface) is assigned an *interrupt address*. It may only request an interrupt when the Processor Board's Interrupt Control circuitry is presenting that address on the Mother Board interrupt address lines (IA0-IA2). (The interrupt address is used in transferring control to the appropriate interrupt handling subroutine.) In the 4024, however, devices are not assigned interrupt addresses, and the hardware for vectored interrupts is omitted.

4025 Processor Board Contents

Figure 5-1 is a block diagram for the 4025 Processor Board. This board holds the 8080 microprocessor and its associated circuits (Clock and Reset Circuitry, Address Buffers, Data and Control Buffers, Interrupt Control, and DMA Control).
This board also holds part of the terminal's memory. Most of that memory resides on other boards, especially the Display Memory Board. The memory on the Processor Board includes:

- Four ROMs holding the system firmware.
- A *Fix-It* including an FPLA and an FPROM to patch errors in system firmware.
- *System RAM* used as a scratchpad.
- A CMOS battery-powered backup RAM to remember settings such as baud rate and parity when the 4025 is turned off.

The following summarizes the contents of the Processor Board:

I. Processor and Associated Circuitry
   A. 8080 Microprocessor
      1. Data Bus
      2. Address Bus
      3. Control Bus
   B. Reset Address Modifier
   C. Bus Address Buffers
   D. Clock and Reset Circuitry
   E. Data and Control Buffer
   F. Interrupt Control
      1. Interrupt Address Counter
      2. IRQ Sampler
      3. Interrupt Priority Register
      4. RESTART Instruction Register
   G. DMA Control

II. Memory Circuitry
   A. Memory Address Buffers
   B. Fix-it
   C. System ROM
   D. System RAM
   E. CMOS Battery RAM
   F. Memory Data Buffers
   G. Power Down Circuitry
   H. Memory Address Decoder
   I. Memory Control Logic

Each of these circuits is discussed in turn.

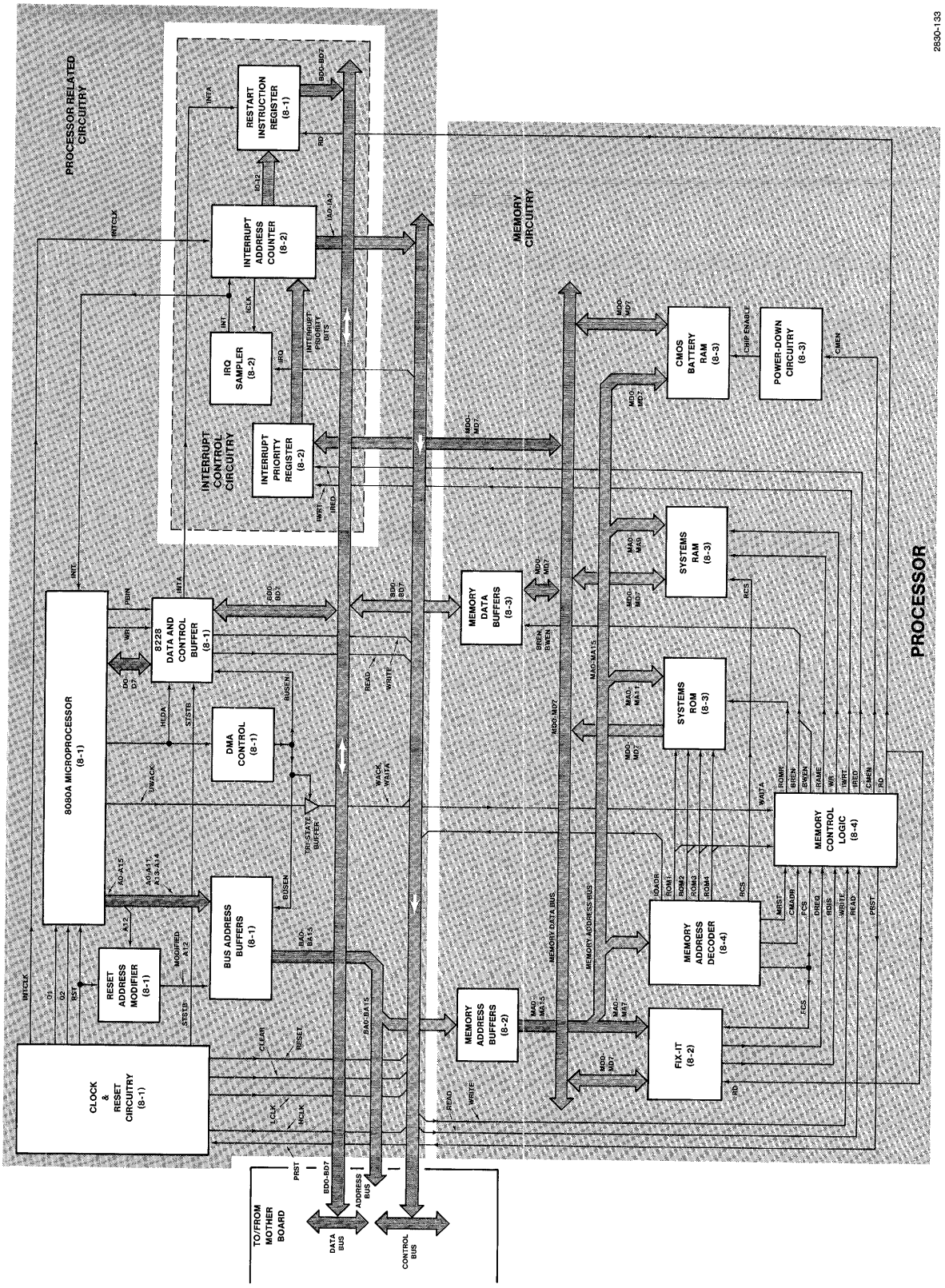


Figure 5-1. 4025 Processor Board Block Diagram.

PROCESSOR AND ASSOCIATED CIRCUITRY

8080A Microprocessor

The 4025's microprocessor (Intel 8080A or equivalent) is the "brain" of the terminal. It communicates with its memory by means of three busses, or groups of signal lines: the data, address, and control busses.

Data Bus

Pins D0-D7 are the data bus. They are buffered by the Data and Control Buffer to become the BD0-BD7 (Bus Data) lines on the Mother Board. Within the Memory Circuitry on the Processor Board, the BD0-BD7 lines are buffered (by the Memory Data Buffer) to become the MD0-MD7 (Memory Data) lines.

Address Bus

Pins A0-A15 drive the address lines. These lines are buffered (by the Bus Address Buffers) to become the BA0-BA15 (Bus Address) lines on the Mother Board. Within the Processor Board's Memory Circuitry, the BA0-BA15 lines are buffered (by the Memory Address Buffer) to become the MA0-MA15 (Memory Address) lines.

Control Bus

The control bus includes the following lines: SYNC, DBIN, READY, WAIT, WR, HOLD, HLDA, INTE, INT, RESET, and INTA. All of these except INTA connect directly to the microprocessor. (The INTA line is driven by the Data and Control Buffer, and is discussed later.)

SYNC. The SYNC signal from the 8080 is used to synchronize the 8224 Clock Generator. This signal is required so that the Clock Generator can generate the STSTB (Status Strobe) signal for the 8228 Data and Control Buffer.

DBIN. The 8080's DBIN pin provides a signal to steer the 8228 Data and Control Buffer. This pin has a TTL "high" when the 8080's data bus is in its input mode.

READY. The READY signal indicates to the 8080 that valid memory data is available on its data bus. If, after sending an address out during a memory read operation, the 8080 does not receive a READY signal, it will enter a WAIT state and remain in that state for as long as the READY line is low.

WAIT. The WAIT signal acknowledges that the 8080 is in a WAIT state. Its output is buffered by an inverter to provide the UWACK (Microprocessor Wait Acknowledged) signal. Provided the BUSEN (Bus Enabled) signal from the DMA Control is true, UWACK is passed on to the Mother Board as the WACK (Wait Acknowledged) signal.

WR. The WR (Write) signal is used when the microprocessor writes to one of its memory locations. It indicates to the Data and Control Buffer that the data on D0-D7 has stabilized.

HOLD. The HOLD signal is an inverted and buffered version of the BRQ (Bus Request) signal on the Mother Board. It tells the 8080 that an external device is seeking DMA (Direct Memory Access).

HLDA. The HLDA (Hold Acknowledge) signal is sent by the 8080 in response to the HLD signal. It indicates that the microprocessor is about to release the address and data busses (put its A0-A15 and D0-D15 outputs in the high-impedance state). The HLDA signal is sent to the DMA Control circuitry and the Data and Control Buffer.

INTE. The INTE (Interrupt Enable) output indicates the state of the 8080's internal interrupt enable flip-flop. This flip-flop is set and cleared under program control. When cleared, it inhibits the 8080 from responding to interrupt requests. It is automatically cleared, disabling further interrupts, at the start of an instruction fetch cycle when the 8080 responds to an interrupt. It is also cleared on power-up by the RST signal.

INT. The INT (Interrupt Request) signal comes to the 8080 from the Interrupt Request circuitry. The 8080 responds to the interrupt after completing its current instruction. That is, the 8080 signals "interrupt acknowledged" to the Data and Control Buffer, which in turn sends the INTA (Interrupt Acknowledged) signal to the Interrupt Control Circuitry.

RST. The RST (Reset) signal clears the 8080's internal program counter, causing it to set its address bus lines to X'0000' when fetching its next instruction. (Normally, this would cause the 8080 to begin its program at address X'0000'. In the 4025, however, a Reset Address Modifier changes one bit on the address bus, causing the microprocessor to start its program at X'1000' instead. This is discussed later, under "Restart Address Modifier.")

Phase One and Phase Two Clocks. Pins 22 and 15 on the 8080 are its two clock inputs. The clock signals, which are not TTL compatible, are provided by an 8224 Clock Generator device in the Clock and Reset Circuitry.

Reset Address Modifier

(See Schematic 8-1.)

Figure 5-2 shows part of the 4025's memory address space. Note that the System Firmware starts at address X'1000' and that System RAM occupies addresses from X'0000' to X'03FF'. The System RAM includes eight *interrupt vectors* (pointers to interrupt handling subroutines) at addresses X'0000', X'0008', X'0010', etc.

On power-up, or when MASTER RESET is pressed, the 8080 microprocessor always branches to X'0000'. However, we want it to branch to X'1000', where the system firmware begins. The Reset Address Modifier overcomes this difficulty; on power-up or MASTER RESET, it alters bit A12 on the microprocessor address bus. Altering that bit changes the X'0000' (which the 8080 places on the bus) to X'1000' (the firmware starting address).

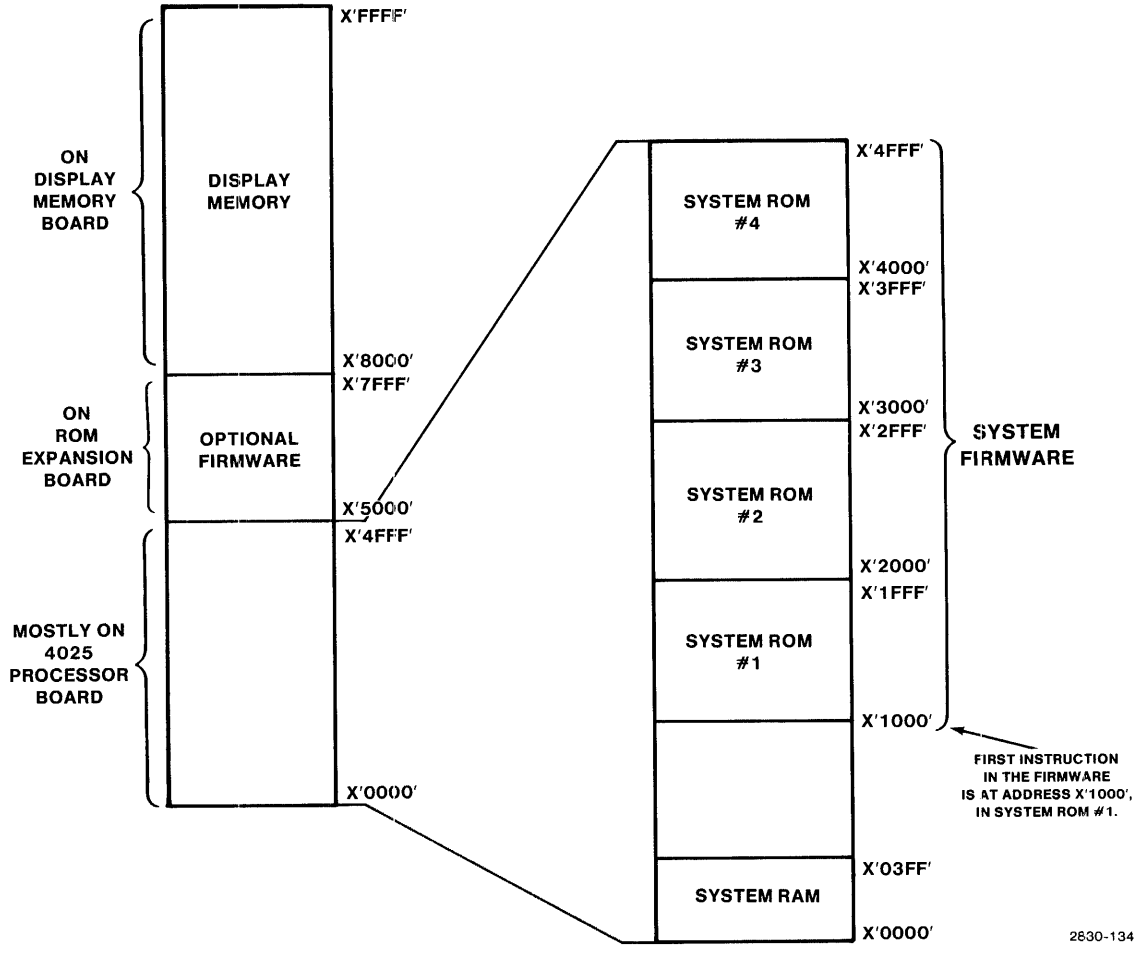


Figure 5-2. Location of System Firmware in 4025 Memory Address Space.

Schematic 8-1 shows the Reset Address Modifier. On power-up or MASTER RESET, the RST pulse is inverted and clears the Restart Address Modifier flip-flop. The OR gate drives address line A12 high whenever (a) the flip-flop is cleared, or (b) the Processor really is driving its A12 address pin high. The result is that, on power-up or MASTER RESET, the Processor accesses address X'1000' instead of X'0000'.

Bus Address Buffers

(See Schematic 8-1.)

The Bus Address Buffers take the A0-A15 signals from the microprocessor and buffer them to drive the main terminal address bus, lines BA0-BA15. These buffers use 74LS367 ICs, each containing six tri-state buffers (Figure 5-3). Four of these are enabled by the 74LS367 G1 input, and two by the G2 input. Of the 18 buffers in three ICs, 16 are used as Bus Address Buffers. (The other two are used in the Restart Instruction Register.)

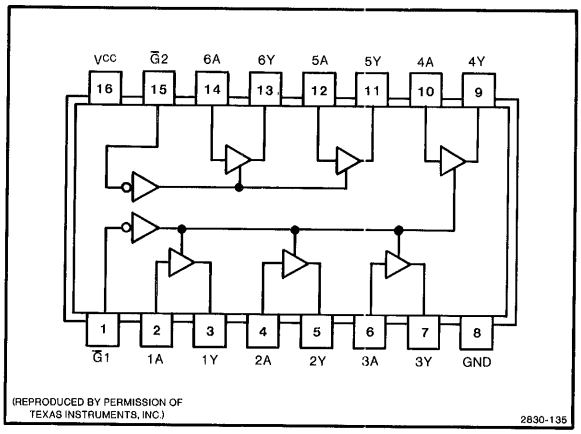


Figure 5-3. 74LS367 Tri-State Buffers.

Clock and Reset Circuitry

(See Schematic 8-1.)

The Clock and Reset Circuitry consists of an Intel 8224 Clock Generator and several logic gates. It provides these signals:

- **Phase One and Phase Two clocks** for the microprocessor.
- **HCLK (High Frequency Clock).** An 18.432 MHz clock.
- **LCLK (Low Frequency Clock).** A 2.048 MHz clock.
- **INTCLK (Internal Clock).** INTCLK is logically identical to LCLK, but is driven by a different gate. INTCLK is used only on the Processor Board.

  (Normally, LCLK and INTCLK are derived from the Phase Two Clock. However, an external clock may be used for test purposes. To use the external clock, ground Pin 2 of connector J3 and supply the external clocks on Pin 2 of that connector.)

- **RST (Reset).** On power-up or MASTER RESET, a Schmidt trigger and type D flip-flop within the 8224 Clock Generator provide a RST pulse to reset the 8080.
- **RESET and CLEAR.** The RESET and CLEAR signals (which reset circuitry thoughout the terminal) are logically identical. CLEAR is used on the Processor Board, and RESET is supplied to the Mother Board for use by the rest of the terminal. These signals are sent when a RST pulse occurs (on MASTER RESET or power-up), or when a processor-initiated reset occurs (PRST signal).
- **STSTB (Status Strobe).** STSTB is a special-purpose signal from the 8224 Clock Generator. It is used to strobe *status words* from the 8080 into the 8228 Data and Control Buffer. These status words are sent during the first clock cycle of a microprocessor instruction, and indicate to the 8228 when certain conditions occur. For instance, one such status word indicates when the 8080 is acknowledging an interrupt.

Data and Control Buffer

(See Schematic 8-1.)

The Data and Control Buffer is an 8228 "system controller" IC especially designed for use with the 8080 microprocessor. It buffers the 8080 data lines D0-D7 to drive the terminal bus data lines BD0-BD7. It also generates several control signals for other circuits on the Processor Board (and elsewhere in the terminal). These signals are:

- **WRITE.** Indicates when the microprocessor is writing to memory.
- **READ.** Indicates when the microprocessor is reading from memory.
- **INTA (Interrupt Acknowledged).** Indicates when the microprocessor has acknowledged an interrupt. The Data and Control Buffer sends this signal when it receives an "interrupt acknowledged" status word (binary 00100011 on lines D0-D7) from the 8080 at the same time as it receives a STSTB (Status Strobe) signal from the 8224 Clock Generator. The INTA signal is used by the Interrupt Control Circuitry, which is described next.

Interrupt Control

To understand the Interrupt Control cirucitry, you should know how the 8080 is designed to respond to interrupts. You should also know about the 4025's interrupt handling procedures.

8080 Interrupt Handling

The 8080 microprocessor handles interrupts as follows:

1. The device requesting the interrupt sends an IRQ (interrupt request) signal. (In the 4025, this signal, after a short delay, appears as the INT signal at Pin 14 of the 8080A.)
2. After completing its current instruction, the 8080 causes the 8228 Data and Control Buffer to send an INTA (Interrupt Acknowledged) signal. (It does this by sending the "interrupt acknowledged" status word at the same time as the Clock Generator sends the STSTB status strobe signal. This was described earlier in this section, under "Data and Control Buffer.")

3. In most 8080 systems, the device requesting the interrupt places a single eight-bit byte on the data bus. This byte is interpreted by the 8080 as an instruction; usually it is the RESTART instruction, binary 11AAA111, where AAA represents the binary interrupt address. In the 4025, a special Restart Instruction Register places this byte on the data bus, thereby relieving the interrupting device from this task.

4. The RESTART instruction, binary 11AAA111, causes the 8080 to branch to the subroutine whose starting address is binary 00000000 00AAA000. For instance, if AAA is 000, the processor branches to address X'0000'; if AAA is 001, it branches to X'0008'.

5. The eight addresses accessed by the RESTART instruction (X'0000', X'0008', X'0010', X'0018', etc.) contain *interrupt vectors*: These are JUMP instructions which direct the processor to the interrupt handling subroutines. The processor executes the appropriate interrupt routine, then returns to the program it was executing when the interrupt occurred.

4025 Interrupt Protocol

Several parts of the 4025 can request Processor interrupts; the Keyboard and Host Ports on the Data Communications Interface Board are examples. Each device capable of requesting interrupts is assigned an *interrupt level*, usually by setting straps on its circuit board.

The eight interrupt levels are numbered from 0 to 7. These interrupt level numbers have two functions:

- They tell the microprocessor which interrupt handling subroutine to use in responding to an interrupt.

- They can be used, in conjunction with the Interrupt Priority Register, to decide which interrupts have high priority and which have lower priority. Devices assigned interrupt level 0 have the highest priority; those assigned interrupt level 7 have the lowest priority.

NOTE

The Processor Board includes an Interrupt Priority Register for allowing the microprocessor to respond to some interrupts, but not to others. However, the 4025's firmware does not actually use this feature; it either enables all levels of interrupts or disables them all.

An Interrupt Address Counter issues an interrupt level in binary form on the interrupt address lines IA0-IA2. Any device which is assigned the current interrupt address level may request an interrupt by pulling the interrupt request line, IRQ, low.

The Interrupt Control circuitry examines the IRQ line about 380 ns after the interrupt address level is issued. If there is no interrupt request, the IA0-IA2 lines are incremented to the next level. If there is an interrupt request, this request is presented to the 8080 Processor on the INT line. This resets the Processor's internal interrupt enable flip-flop, inhibiting further incrementing of the interrupt address lines and sampling of the interrupt request line.

The interrupt address lines increment up to and including the number stored in the Interrupt Priority Register. This register occupies address X'0800', with the priority level represented in binary form in the least three significant bits of data. The register is cleared, indicating a level 0 priority, only when the terminal is powered up, the MASTER RESET switch is operated, or a Processor-controlled reset is issued.

Any time the Processor writes into the interrupt priority register, the interrupt address lines are returned to level 0. The Processor interrupts must be disabled before writing into the interrupt priority register to ensure correct operation.

Interrupt Control Circuit Descriptions

The Interrupt Control Circuitry is shown in Figure 5-4. It includes these blocks: Interrupt Address Counter, IRQ Sampler, Interrupt Priority Register, and Restart Instruction Register. Each of these is discussed in turn.

Interrupt Address Counter. The Interrupt Address Counter appears in Figure 5-4 and Schematic 8-2. It includes: (a) a J-K flip-flop, (b) an AND gate controlling the flip-flop's J and K inputs, (c) a 74LS161 counter, and (d) a 74LS85 magnitude comparator.

The gate enables or disables the counter by controlling the J and K flip-flop inputs. When the inputs are high, the flip-flop toggles, providing a clock for the counter; but when the J and K inputs are low, the flip-flop does not change state, and the counter stops. The gate enables the counter when (a) interrupts are enabled (INTE signal from the microprocessor is true), and, moreover, (b) the IRQ Sampler has not already detected an interrupt request.

The flip-flop is cleared, disabling the counter, when an interrupt is acknowledged (INTA signal received), when a RESET occurs (CLEAR signal), or when the Processor writes into the Interrupt Priority Register (IWRT signal). The latter two conditions also clear the 74LS161 counter.

The flip-flop's output is the ICLK (interrupt clock) signal. Both flip-flop outputs are used, providing noninverted (ICLK-1) and inverted (ICLK-0) versions of the clock. ICLK-0 clocks the 74LS161 on each of its positive edges. ICLK-1 latches the counter outputs into the Restart Instruction Register.

The counter and the magnitude comparator are the heart of the Interrupt Address Counter. The counter outputs drive the interrupt address lines I0-I2 (for the Restart Instruction Register) and IA0-IA2 (for the rest of the terminal). The magnitude comparator compares the current count (on its "B" inputs) with the contents of the Interrupt Priority Register (on the "A" inputs). When the counter reaches the number in the Interrupt Priority Register, the comparator's "A>B" output goes low, zeroing the counter on the next clock pulse. Thus, the current interrupt address can never exceed the maximum interrupt level stored in the Interrupt Priority Register.

NOTE

In practice, the 4025's firmware sets the maximum interrupt level to 7 by loading the Interrupt Priority Register with a number whose least significant bits are binary 111. This maximum interrupt level is never changed by the firmware, so the "interrupt priority" feature is unused.

IRQ Sampler. The IRQ Sampler is a type D flip-flop (Figure 5-4). When the flip-flop is set, its INT output is low, signifying that no interrupt request has been received. When an interrupt request is received, the IRQ line goes low; the next ICLK pulse clears the IRQ Sampler, sending the INT (Interrupt Request) signal to the Processor. (If interrupts are disabled, there will be no ICLK signal to clear the IRQ Sampler.)

The flip-flop is set, enabling it for the next interrupt request, whenever the interrupt is acknowledged (INTA signal), the Processor writes into the Interrupt Priority Register (IWRT signal), or a RESET occurs (CLEAR signal).

Interrupt Priority Register. The Interrupt Priority Register is shown in Figure 5-4 and Schematic 8-2. This register functions as a word in the microprocessor's memory. (It occupies address X'0800'.) When the Processor writes into this word, the IWRT line goes low; when that line comes high again, the three least significant data bits (MD0-MD2) are clocked into the register's latch. The latch outputs drive the "A" inputs of the magnitude comparator in the Interrupt Address Counter.

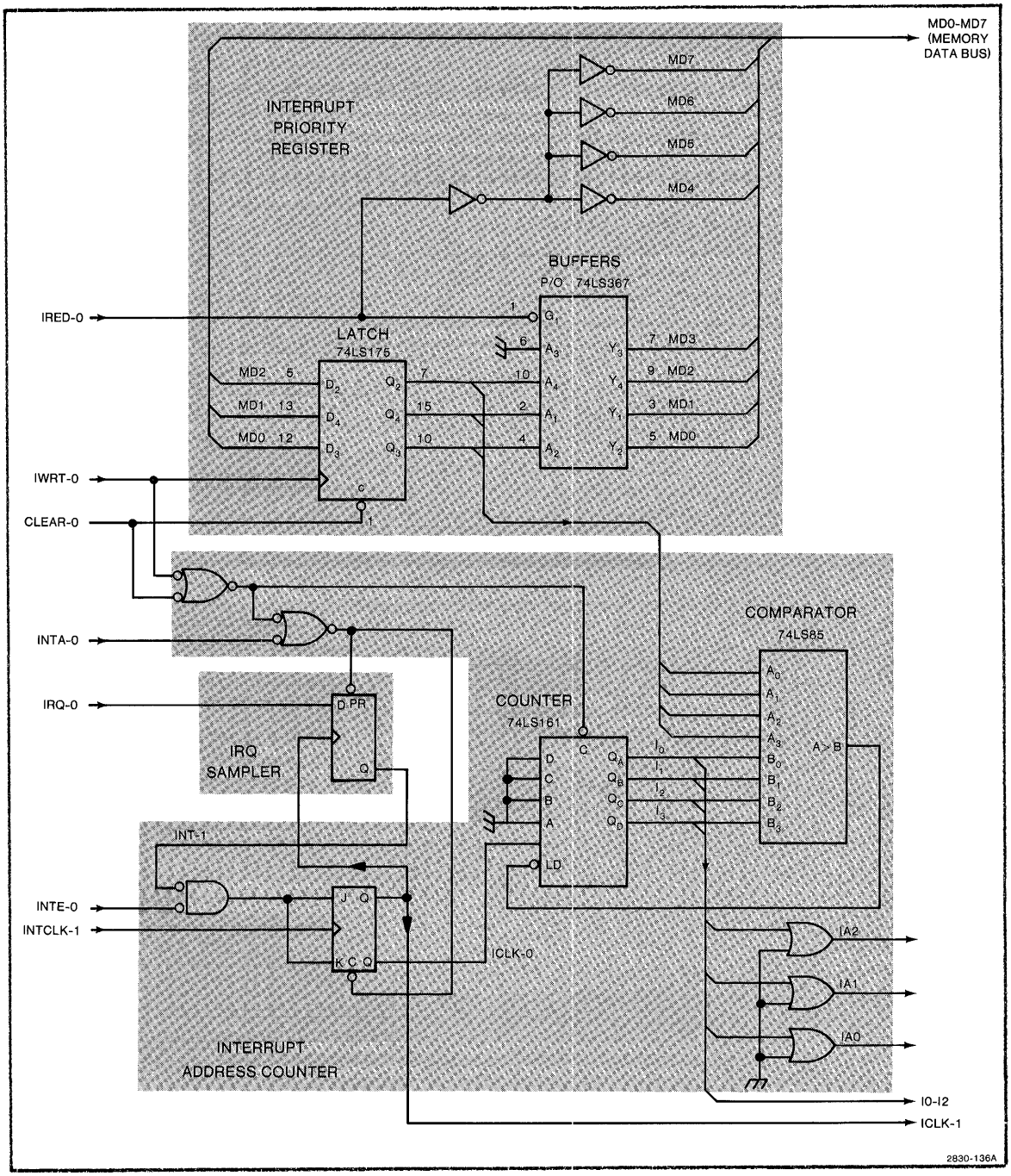


Figure 5-4. Interrupt Control Circuitry.

When the Processor reads from the Interrupt Priority Register, the IRED line goes low. This enables four tri-state buffers (part of a 74LS367); the buffers place the Interrupt Priority Register data bits on data lines MD0-MD3. (MD0-MD2 hold the current interrupt level, while MD3 is always zero.) The IRED signal also drives several open-collector inverters, which send data lines MD4-MD7 low. The result is that the Processor reads a word whose three low-order bits (MD0-MD2) show the number stored in the latch, and whose five high-order bits (MD4-MD7) are all zeroes.

Restart Instruction Register. Recall that after an interrupt has been acknowledged, the 8080 requires an instruction (usually a Restart instruction, binary 11AAA111) to be placed on the data bus. The RESTART Instruction Register does this.

See Schematic 8-1. The Restart Instruction Register consists of a latch and some tri-state buffers. The latch holds the current interrupt level, which is clocked into it by the INTCLK signal each time that interrupt level changes.

When an interrupt is acknowledged, the INTA signal enables the tri-state buffers, placing the binary word 11AAA111 on the data bus. (Here, AAA is the interrupt level stored in the latch.) This word is interpreted by the processor as a RESTART instruction, causing it to branch to the subroutine which starts at binary address 00000000 00AAA000.

DMA Control

The DMA Control circuitry supervises DMA (Direct Memory Access) operations. When no DMA operation is proceeding, this circuitry sends the BUSEN (Bus Enable) signal, enabling the the Data and Control Buffer and the Bus Address Buffers. This gives the microprocessor access to the terminal's address and data busses (BA0-BA15 and BD0-BD7). When a DMA operation is in progress, BUSEN is false, effectively disconnecting the microprocessor from the terminal's memory.

NOTE

Although DMA circuitry is included in the 4025, it is not used. (There are no 4025 boards capable of Direct Memory Access.)

Since the 4025's DMA capability is not used, you may wish to skip over the following description of DMA operation. The description is included only as an aid in understanding the circuit, for trouble-shooting purposes if any of its components should fail.

The DMA Control circuitry is shown in Schematic 8-1. Should a DMA operation ever occur, it would proceed as follows:

1. The circuit board requesting direct memory access pulls the BRQ line low. The inversion of this signal (a TTL high) is applied to the microprocessor's HOLD input and to the two DMA Control flip-flops, enabling those flip-flops. (The absence of the BRQ signal normally holds these flip-flops in their reset, or cleared, condition.)

2. When the microprocessor finishes its current instruction, it responds with the HLDA (Hold Acknowledged) signal. This signal clocks the first of the two DMA Control flip-flops, setting it.

3. On the next LCLK pulse, the second DMA Control flip-flop is clocked, causing it to set. When set, it turns off the BUSEN signal, effectively disconnecting the microprocessor from the terminal's memory.

   Turning off the BUSEN signal also disables a buffer (shown in Schematic 8-2) which passes the UWACK (Microprocessor Wait Acknowledged) Signal on to the Mother Board as the WACK (Wait Acknowledged) signal. Thus, the Processor Board cannot drive the WACK line during a DMA operation. (The board requesting DMA is responsible for this line.)

   Also, as the second DMA Control flip-flop goes set, it sends the BGOUT (Bus Grant Out) signal. This informs the DMA device that it now has access to the terminal's memory.

4. When the DMA device is finished, it ceases to send the BRQ signal. As BRQ goes false, the two DMA Control flip-flops are cleared. The second flip-flop sends the BUSEN signal once again, reconnecting the microprocessor to the terminal's address and data busses (and to the WACK line). The second flip-flop also turns off the BGOUT signal.

4025 PROCESSOR BOARD MEMORY CIRCUITRY

Besides the Processor and its related circuitry, the 4025 Processor Board contains part of the 8080's memory. In particular, it contains the system firmware (in the four *System ROMs*), a *Fix-It* circuit to patch any errors in the System ROMs, *System RAM* (used by the Processor as a scratchpad and to hold *interrupt vectors*), and a battery-powered backup RAM in which operating parameters are stored when the terminal is turned off.

The Memory Circuitry on the Processor Board may be divided into the following blocks: Memory Address Buffers, Memory Data Buffers, Fix-It, System ROM, System RAM, CMOS Battery RAM, Power-Down Circuitry, and the Address Decoding and Control Logic. Each of these is discussed in turn.

Memory Address Buffers

(See Schematic 8-2.)

The Memory Address Buffers consists of sixteen gates which buffer the main terminal address bus lines BA0-BA15, to drive the "memory address" lines MA0-MA15 on the Processor Board.

A pack of eight 10K resistors connects the lines MA0-MA7 to the + 5V supply. When the terminal is turned off, the resistors pull the MA0-MA7 lines down to ground potential. When the terminal is turned on, they function as pull-up resistors, guaranteeing a TTL "high" on the MA0-MA7 lines. In this way, the address inputs are prevented from "floating" between the high and low states, even when power is turned off.

Fix-It

(See Schematic 8-2.)

The Fix-It is a special circuit for making minor changes to the firmware without the need for new system ROMs. It works like this:

- The parts of firmware to be changed are flagged by programming their addresses into an FPLA (Field Programmable Logic Array).
- When a flagged memory location is addressed, the FPLA provides seven output signals. One signal disables the system firmware ROMs and enables an FPROM (Field Programmable Read-Only Memory). The other six outputs, together with address lines MA0 and MA1, select a word from the FPROM.

- The data in the FPROM word is placed on the memory data bus. In effect, the word in the FPROM replaces the flagged word in the system firmware.

See Schematic 8-2. The FPLA has only 14 input lines, so it does not fully decode the 16-bit memory addresses. Instead, it ignores address lines MA0 and MA1; the remaining 14 address lines specify four-word segments in memory. To change a particular four-word segment, the FPLA is programmed to recognize that segment's address and, when it does, drive its F0 output low.

Provided the Fix-It is not disabled (by grounding J3 Pin 1), the low on the FPLA F0 output sends the following signals:

- **RDIS (ROM Disable).** This disables the ROMs which hold the system firmware.

- **FCE (FPROM Chip Enable).** This drives one of the FPROM's chip select inputs. It also informs the Address Decode and Control Circuitry that the FPROM, rather than the faster masked ROM, is being accessed. (Since the FPROM is a slower device, it is necessary to generate a WAIT signal to give the FPROM time to respond before the microprocessor tries to read the data from it.)

- **DREQ (Data Request).** This steers two 74LS157 data selectors, causing the FPROM address inputs to come from the FPLA rather than from the memory address bus. (This signal is also supplied as an input to the Memory Control circuitry.)

The Fix-It FPROM holds 1024 eight-bit words. Of these, 256 words are accessed by the FPLA; these are reserved for changes to tables or jump instructions. The other 728 words are addressed from the memory address bus, just like other parts of Processor's memory. These words are used for patch programs. (Jump instructions in the lower 256 FPROM words are used to pass control to these patch programs.)

When the FPLA detects a flagged location in memory, it addresses a word in the FPROM. The address on the FPROM address pins A0-A9 is as follows:

Pins	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
Address Bits	0	0	F7	F6	F5	F4	F3	F2	MA1	MA0

Here, A0-A9 are the FPROM address bits, F2-F7 are the outputs of the FPLA, and MA0-MA1 are the least significant two address bits on the memory address bus.

When the Fix-It FPROM is addressed, but not by the FPLA, memory address signals MA0-MA9 drive the A0-A9 FPROM address pins.

The Fix-It's two data selectors switch the FPROM address inputs between the FPLA and the memory address bus. When the FPLA has detected a flagged address, the DREQ signal switches the FPROM inputs to the FPLA; at other times, the FPROM address inputs are driven by lines on the memory address bus. The FPROM is enabled by either (a) the FPLA detecting a flagged address and driving its F0 output low, or (b) the Memory Address Decoder detecting an address in the range X'0C00' to X'0FFF' and sending the FCS (FPROM Chip Select) signal. When either of these two events occur, the FCE (Fixit Chip Enable) signal enables the FPROM and informs the Memory Control circuitry of the fact that it is doing so.

System ROM

(See Schematic 8-3.)

The system firmware resides in four ROMs, which occupy addresses between X'1000' and X'4FFF'. System ROM number 1 occupies addresses X'1000' to X'1FFF', ROM number 2 occupies X'2000' to X'2FFF', etc. These ROMs appear in Schematic 8-3. They are enabled by the ROM1, ROM2, ROM3, ROM4 chip select signals from the Memory Address Decoder, take their address inputs from MA0-MA9 on the memory address bus, and place their outputs MD0-MD7 on the memory data bus.

System RAM

(See Schematic 8-3.)

The System RAM is the microprocessor's "scratchpad." It stores numbers during calculations; it holds the "stack" of return addresses for subroutines, etc.; and it holds the "interrupt vectors" which point to the interrupt handling subroutines.

This RAM consists of two AM9130A integrated circuits, shown in Schematic 8-3. One of these holds the least significant four bits of each word in System RAM, and the other the most significant four bits. These RAMs occupy addresses X'0000' to X'03FF'. They are enabled by three signals: RCS (RAM Chip Select), RCE (RAM Chip Enable), and RAME (RAM Enable).

CMOS Battery RAM

(See Schematic 8-3.)

The CMOS battery-powered RAM holds settings such as baud rate, parity, etc., when the 4025 is turned off. It takes its power from the $+5V_1$ line, which carries power to and from battery BT1001. To conserve the battery, when the terminal is turned off this ROM is disabled and its address inputs are forced low.

Two 256 x 4 RAMs are used. One RAM holds the least significant four bits of each 8-bit word; the other holds the most significant four bits.

Power Down Circuitry

(See Schematic 8-3.)

The Power Down circuitry protects the data in the CMOS Battery RAM during power-off, power-on sequences. It does this by disabling the CMOS RAM and sending a PWT (Power Wait) signal to pull the WAIT line low.

The RAM is disabled, and the PWT signal sent, when the Power Supply sends a PWDN signal to warn of impending loss of power on the + 5 V line. On power-up, the RAM is not enabled, and the PWT signal is not turned off, until the terminal is completely powered-up. This prevents transients on the address or data lines from garbling the data in the CMOS Battery RAM.

Memory Data Buffers

(See Schematic 8-3.)

The Memory Data Buffers are shown in Schematic 8-3. The 74LS241 integrated circuits each hold eight tri-state buffers, divided into two groups of four buffers each. One group is enabled by the BREN (Buffer Read Enable) signal from the Memory Control circuitry. During microprocessor read operations, this group takes data from the board's "memory data" lines (MD0-MD7) and places it on the main terminal data bus (lines BD0-BD7). The other group of buffers in each 74LS241 is enabled during write operations; it places data from the main terminal data bus (BD0-BD7) onto the board's internal memory data bus (MD0-MD7).

Memory Address Decoder

(See Schematic 8-4.)

The Memory Address Decoder consists of gates and data demultiplexer ICs. These devices decode the memory address lines and provide chip enable signals for Processor Board memory. Those signals are:

- **ROM1,ROM2,ROM3,ROM4.** Enable signals for the four System ROMs.

  ROM1: Address in the range X'1000' to X'1FFF'
  ROM2: X'2000' to X'2FFF'
  ROM3: X'3000' to X'3FFF'
  ROM4: X'4000' to X'4FFF'

  Jumpers ("zero-ohm resistors") in series with these signal lines can be removed to disable the four System ROMs.

- **IOADR.** This signal indicates that the address on the address bus is in the range X'0800' to X'0BFF', the "I/O address range." This address range holds the *I/O registers* on the various circuit boards. (The I/O registers function as special words of memory. When the microprocessor writes into these words, it causes certain signals to be sent by the I/O register to the board on which the I/O register is located. By reading from an I/O register, the Processor can learn the state of signal lines on that register's circuit board.)

  The IOADR signal relieves other circuit boards of part of the task of decoding their I/O register addresses; it may be considered as a supplement to the terminal's address bus.

- **FCS (FPROM Chip Select).** Indicates that a location in the Fix-It's FPROM (addresses X'0C00' to X'0FFF') is being accessed.

- **CMADR (CMOS Address).** Indicates that a location in the CMOS Battery RAM (addresses X'0400' to X'04FF') is being accessed.

- **RCS (RAM Chip Select).** Indicates that the System RAM (addresses X'0000' to X'03FF') is being accessed.

- **INTADR.** Indicates that the Interrupt Priority Register (X'0800') is being accessed.

- **MRST (Master Reset).** Indicates that the Processor is accessing memory location X'0801'. (Reading at this address sends the MRST signal, causing a "memory wait." Writing at this address also sends the the PRST (Processor-initiated Reset) signal.)

Figure 5-5 summarizes the Processor Board memory address assignments.

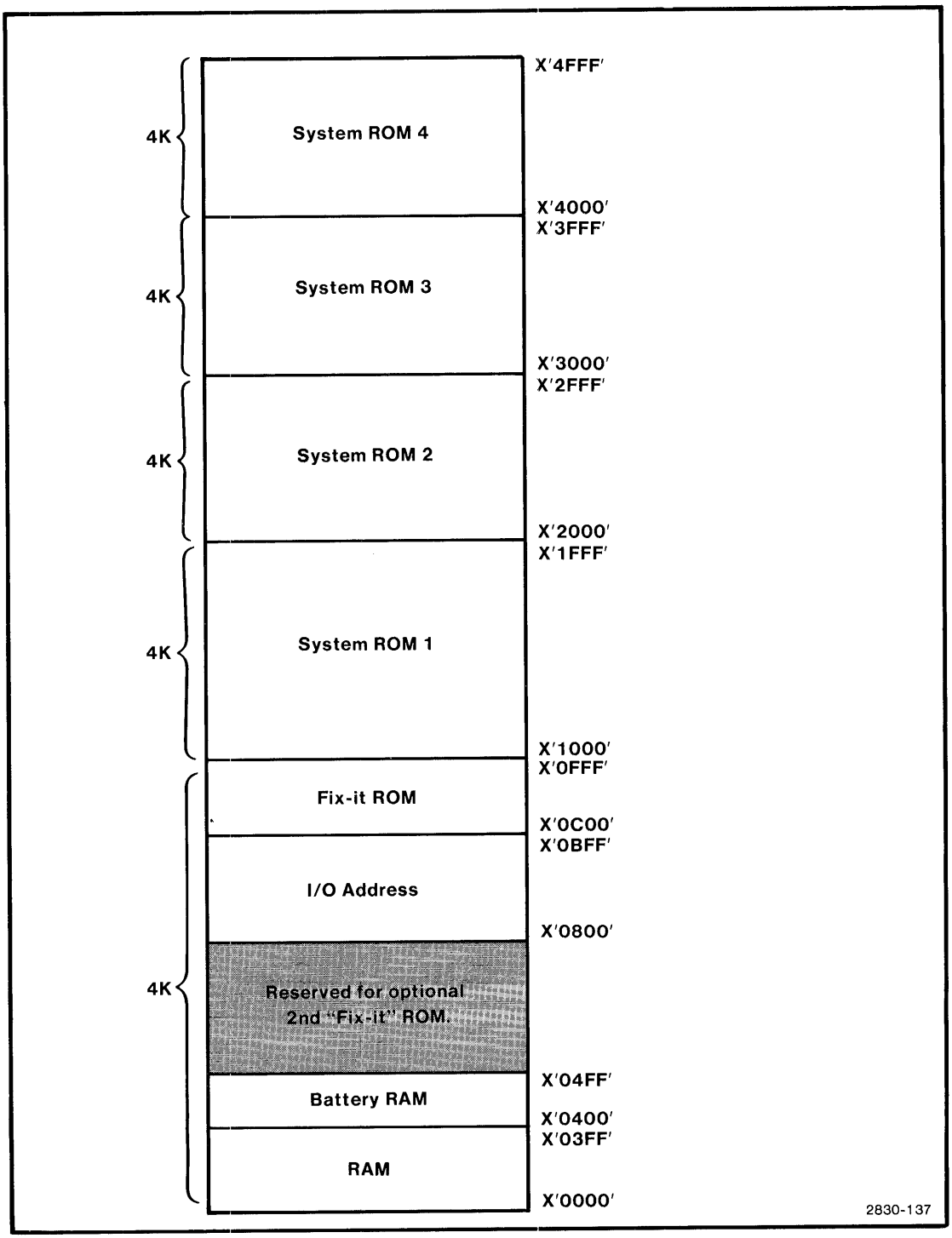


Figure 5-5. 4025 Processor Board Memory Map.

Memory Control Logic

(See Schematic 8-4.)

The Memory Control Logic generates signals which control the transfer of data between the memory devices on the Processor Board.

Inputs to the Memory Control Logic are:

- The signals from the Memory Address Decoder which specify which part of memory is to be accessed. These signals are: ROM1, ROM2, ROM3, ROM4, FCS, RCS, INTADR, and MRST.

- The DREQ and RDIS signals. DREQ indicates that the Processor Board's Fix-It has detected a flagged address. RDIS indicates when the terminal's ROMs are to be disabled because one of the two Fix-Its (on the Processor Board or on the ROM Option Board) has detected a flagged address. (If RDIS is true, but DREQ is false, then the ROM Option Board's Fix-It is requesting the Memory Control Logic to disable the ROMs on the Processor Board—including the Fix-It FPROM.)

- The WAITA (Wait Acknowledged) signal from the microprocessor.

- The TRIST (Tri-State) signal, used to disable the Processor Board's memory.

- The READ and WRITE signals, which signal memory read and write operations.

The outputs of the Memory Control Logic are:

- **BREN (Buffer Read Enable).** Enables the Memory Data Buffers for a read operation. This signal occurs during a READ from any of the System ROMs or the Processor Board's Fix-It FRPOM (ROM1, ROM2, ROM3, ROM4, or FCE signal), provided the ROM Option Board's Fix-It is not disabling the terminal's ROMs (DREQ false and RDIS true). The BREN signal is also sent when reading (READ signal) from System RAM (RCS signal), the Interrupt Priority Register (INTADR signal), or the CMOS Battery RAM (CMADR signal).

  The gates which generate the BREN signal implement the following boolean logic equation, which summarizes the above paragraph:

  $$\text{BREN} = (\text{ROM1} + \text{ROM2} + \text{ROM3} + \text{ROM4} + \text{FCE}) \cdot (\text{DREQ} + \text{not RDIS}) \cdot \text{READ} + \text{READ} \cdot (\text{RCS} + \text{INTADR} + \text{CMADR})$$

- **BWEN (Buffer Write Enable).** Enables the Memory Data Buffers for a write operation. This signal occurs during a write operation to System RAM, the Interrupt Priority Register, or the CMOS Battery RAM.

  Writes occur when the WRITE signal is true, while accesses to System RAM, the Interrupt Priority Register, and the CMOS Battery RAM are indicated by the RCS, INTADR, and CMADR signals, respectively. Thus BWEN is true when WRITE is true and, moreover, RCS or INTADR or CMADR is true.

  The gates which generate the BWEN signal implement the following logic equation, which summarizes the last paragraph:

$$BWEN = WRITE \cdot (RCS + INTADR + CMADR)$$

- **MWT (Memory Wait).** The MWT signal pulls down on the terminal's WAIT line, telling the microprocessor that it must wait for the memory to respond before trying to read from, or write to, memory. (This signal is required when accessing the CMOS Battery RAM or the System RAM, which are slower memory devices than the System ROM.)

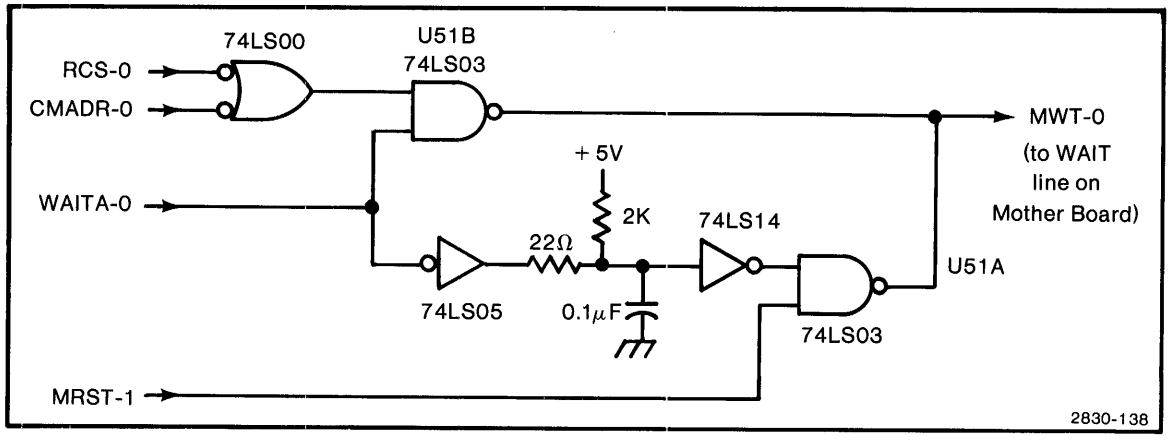


Figure 5-6. Logic for Generating the MWT Signal.

Refer to Figure 5-6, which shows the gates which generate the MWT signal. This signal can be generated in two ways:

- When the Processor tries to access either the System RAM or the CMOS Battery RAM (when RCS or CMADR is true), MWT signal pulls down on the Mother Board's WAIT line. The MWT (or WAIT) signal continues until the microprocessor acknowledges the wait; when this occurs, WAITA (Wait Acknowledge) goes true low, turning off the MWT signal.

- To reset memory, the firmware causes the Processor to access location X'0801'; this sends the MRST (Memory Reset) signal; MRST enables the MWT signal. As before, MWT stays enabled until the processor acknowledges the wait (sends WAITA true). However, for "memory resets" the MWT (and WAIT) signals must last for at least 40 microseconds. This is accomplished by the two inverters and the RC network between them, which delay the WAITA signal by at least 40 μs. (The delay is typically 100 μs or more.)

- **CMEN (CMOS Memory Enable).** An enable signal for the CMOS Battery RAM. This signal is sent when a read or write is occurring (READ or WRITE true) to an address in the CMOS Battery RAM (CMADR true), and memory is not disabled by the TRIST signal or by the Fix-It (TRIST and DREQ both false).

  The gates which generate the CMEN signal implement this logic formula:

  $$CMEN = CMADR \cdot (READ + WRITE) \cdot (\text{not } TRIST \cdot \text{not } DREQ)$$

- **ROMR (ROM Read).** A signal used to enable the system ROMs during read operations. It is sent whenever a READ is occurring and neither the TRIST signal is disabling memory nor the Fix-It FPROM is being addressed (DREQ signal). The gates which generate the ROMR signal implement this logic equation:

  $$ROMR = READ \cdot (\text{not } TRIST \cdot \text{not } DREQ)$$

- **IRED (Interrupt Priority Register Read).** This signals a READ from the Interrupt Priority Register. IRED is sent whenever neither the TRIST signal nor the Fix-It is disabling memory (TRIST and DREQ both false), a read operation is in progress (READ true), and the Interrupt Priority Register is being addressed (INTADR true).

- **IWRT (Interrupt Priority Register Write).** Signals a write to the Interrupt Priority Register.

- **PRST (Processor-Initiated Reset).** This signal indicates that the microprocessor is initiating a terminal RESET. It occurs when the microprocessor writes to address X'0801' (MRST and WRITE both true).

- **RAME (RAM Enable).** Drives the System RAMs' OE (Output Enable) pins during READ operations, provided neither the TRIST signal nor the DREQ signal from the Fix-It is disabling memory.

- **RD (Read) and WR (Write).** Buffered versions of the Mother Board's READ and WRITE signals.

CONNECTORS

The Processor Board communicates with the rest of the terminal mainly through its edge connector, which plugs into a socket in the Mother Board. See the section on the Mother Board for a description of this connector's signal lines.

There are four other connectors on the Processor Board. Two of these, J1 and J4, are used in normal terminal operation. The other two, J2 and J3, are provided as a convenience in testing or troubleshooting the board. J2 brings out the signals on the microprocessor data pins, together with the INTE and STSTB signals. J3 allows the technician to disable the Fix-It, disable all RAMs and ROMs by forcing them into tri-state condition, or disable the terminal's low frequency clock and accept an external signal to drive the LCLK line.

Table 5-1 lists the signals on each pin of these connectors.

Table 5-1

PROCESSOR BOARD CONNECTORS

Connector	Pin	Signal	Description
J1	1	RST	Connects to MASTER RESET Switch.
	2	GND	Connects to MASTER RESET Switch.
J2	1	INTE	Interrupt enable signal.
	2	STSTB	Status strobe signal.
	3	D0	D0 of the microprocessor data bus.
	4	D1	D1 of the microprocessor data bus.
	5	D2	D2 of the microprocessor data bus.
	6	D3	D3 of the microprocessor data bus.
	7	D4	D4 of the microprocessor data bus.
	8	D5	D5 of the microprocessor data bus.
	9	D6	D6 of the microprocessor data bus.
	10	D7	D7 of the microprocessor data bus.
J3	1	FDIS	Ground to disable the Fix-It.
	2	CLKDIS	Ground to disable internal clock.
	3	ϕ 2IN	External clock when Pin 2 is grounded.
	4	TRIST	Ground to force all ROM, RAM into tri-state condition.
J4	1	GND	NiCad battery ground.
	2	BAT+	Connects to battery positive terminal.
	3	GND	NiCad battery ground.

Section 6

4025 COMMUNICATIONS INTERFACE BOARD

INTRODUCTION

The 4025 Communications Interface Board (labeled "Deluxe Communications Bd") includes the following features which are not included in the 4024 Processor/Comm Board:

- Interrupt address circuitry.
- Half duplex communications circuitry.
- Circuitry for changing baud rates under firmware control.

Host and Keyboard Ports

The Communications Interface Board has two major parts: the Host Port and the Keyboard Port. These ports are interfaces through which the microprocessor communicates with the host computer and the keyboard. Each port appears to the microprocessor to be a series of words in its memory.

I/O Registers

These special words of memory are called *I/O registers*. The microprocessor can control signal lines on the Communications Interface Board by writing into these I/O registers. Conversely, it ascertains the states of signal lines by reading from the I/O registers.

Each I/O register consists of two devices: the "read" and "write" halves of the register. The read half of an I/O register usually consists of tri-state buffers which, during read operations, place the data on the terminal's data bus. The write half of an I/O register is usually a latch which, during write operations, saves the data on the terminal's data bus. The latch outputs drive signal lines on the Communications Interface Board.

Tables 6-1 and 6-2 list the functions of the Host and Keyboard Port I/O registers. Figures 6-1, 6-2, and 6-3 show the locations of these registers in memory address space.

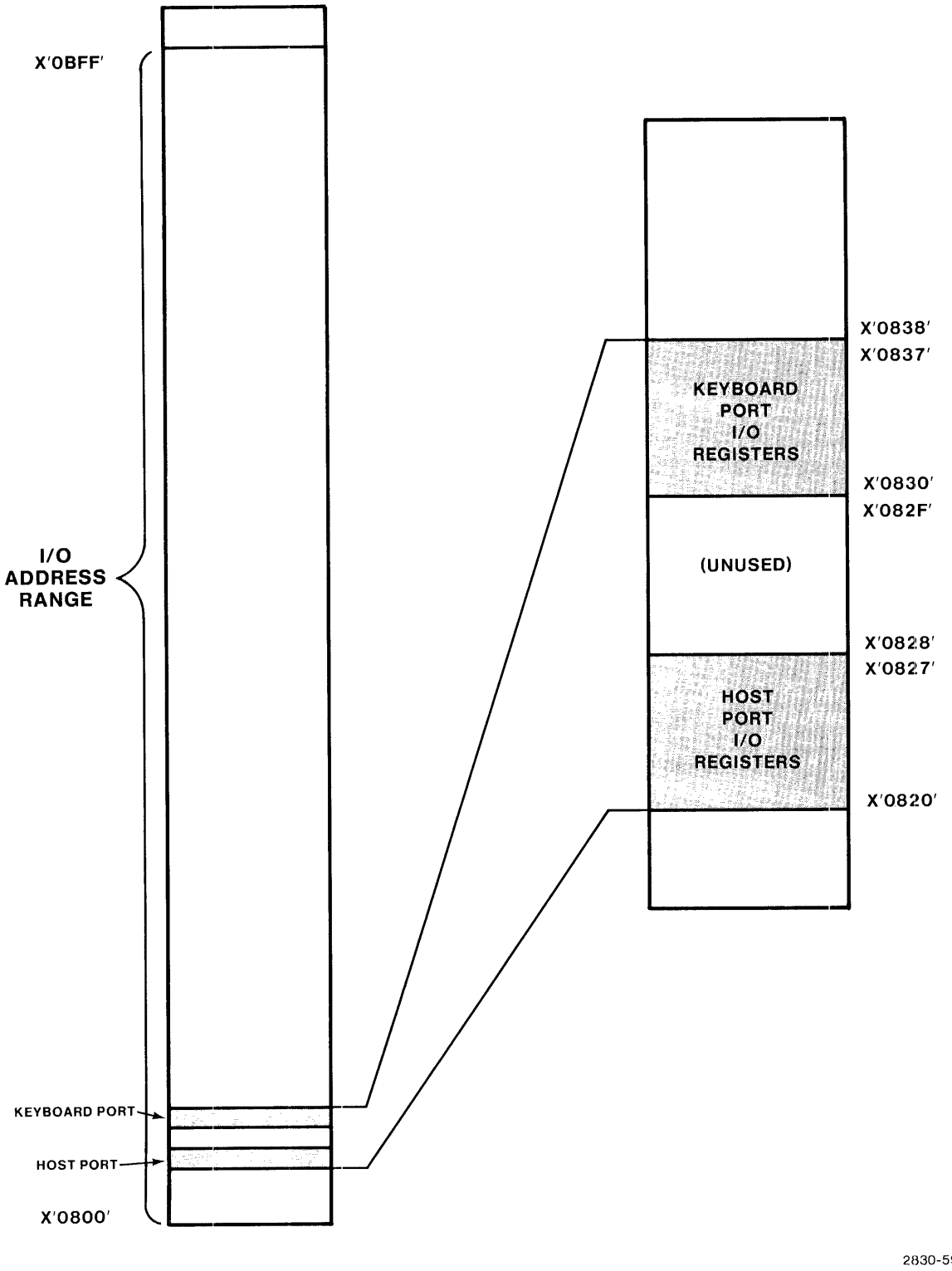


Figure 6-1. Location of Host and Keyboard Ports in I/O Address Space.

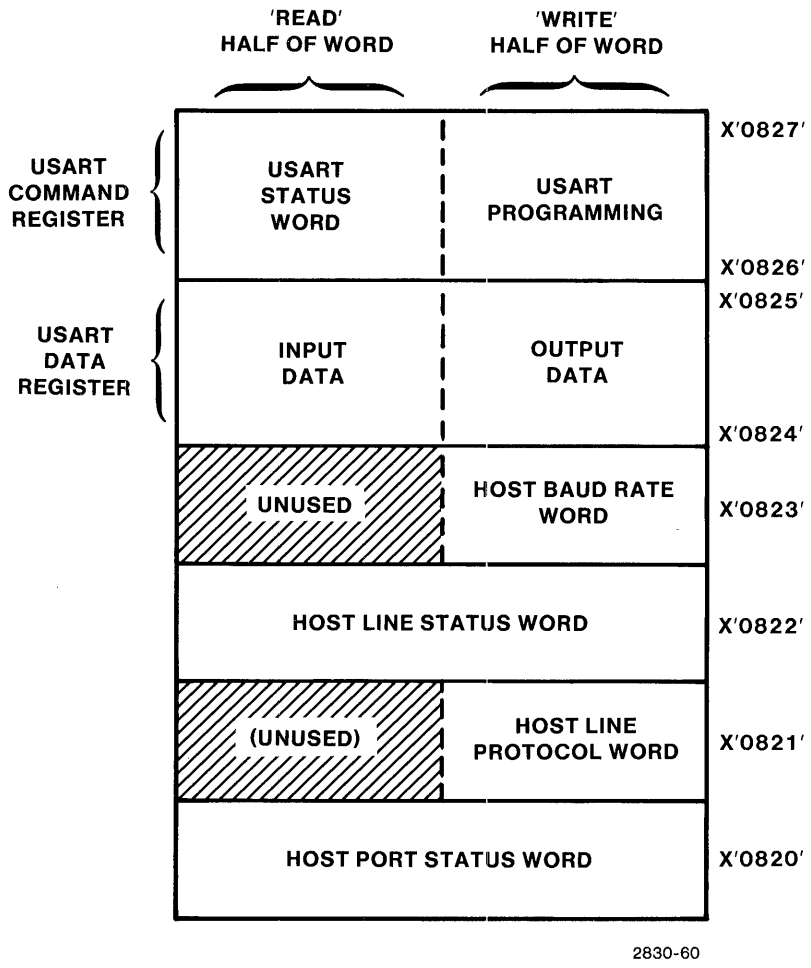


Figure 6-2. Host Port Memory Map.

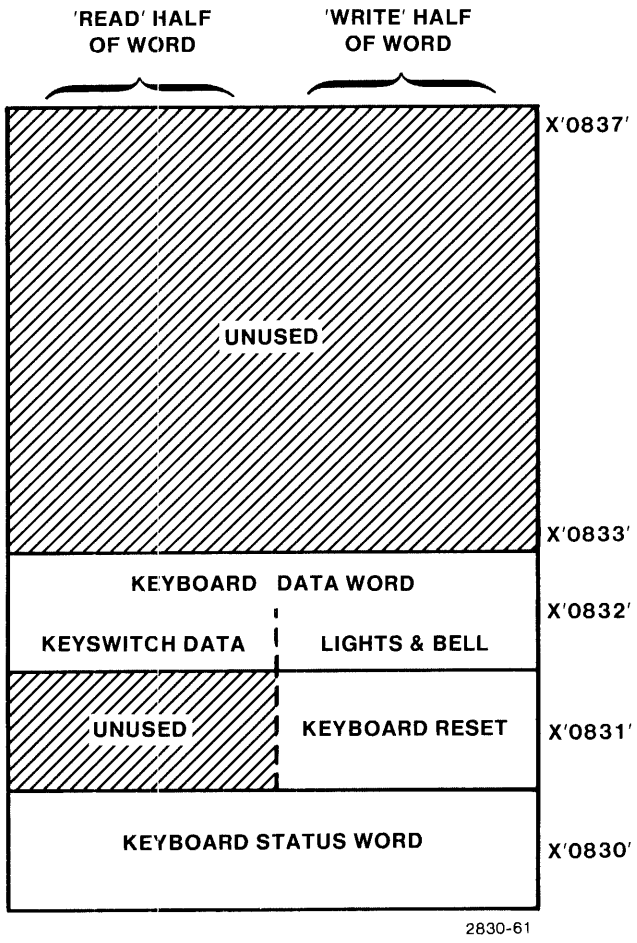


Figure 6-3. Keyboard Port Memory Map.

Table 6-1

4025 HOST PORT I/O REGISTERS

Register Name	Address	Function	
Host Port Status Word	X'0820'	Read	Bit 0: Unused Bit 1: OUTRDY (Output Ready) Bit 2: Unused Bit 3: INRDY (Input Ready) Bit 4: Unused Bit 5: LSC (Line Status Change) Bit 6: TXE (Transmitter Empty) Bit 7: Interrupt Flag
		Write	Bit 0: OUTRDY Interrupt Enable Bit 1: Unused Bit 2: INRDY Interrupt Enable Bit 3: Unused Bit 4: LSC Interrupt Enable Bits 5-7: Unused
Line Protocol Word	X'0821'	Read	Bits 0-7: Unused
		Write	Bit 0: LP0 Bit 1: LP1 Bits 2-7: Unused
Line Status Word	X'0822'	Read	Bit 0: DSR (Data Set Ready) Bit 1: SDCD (Sec. Data Carrier Detect) Bit 2: DCD (Data Carrier Detect) Bit 3: CTS (Clear to Send) Bits 4-7: Unused
		Write	Bits 0-3: Unused Bit 4: DTR (Data Terminal Ready) Bit 5: SRTS (Sec. Req. to Send) (Pin 11) Bit 6: SRTS (Pin 19) Bit 7: RTS (Request to Send)
Baud Rate Word	X'0823'	Read	Bits 0-7: Unused
		Write	Bits 0-3: Receive Baud Rate Bits 4-7: Transmit Baud Rate
USART Data Word	X'0824', X'0825'	Read	Bits 0-7: Read Data
		Write	Bits 0-7: Write Data
USART Programming and Status Words	X'0826', X'0827'	Read	(USART Status Word) Bit 0: TXRDY (Transmitter Ready) Bit 1: RXRDY (Receiver Ready) Bit 2: TXE (Transmit Enable) Bit 3: PE (Parity Error) Bit 4: OE (Overflow Error) Bit 5: FE (Framing Error) Bit 6: Sync Detect Bit 7: Ring Indicator
		Write	(USART Programming and Command Words) Bits 0-7: (See USART Circuit Description.)

Table 6-2

4025 KEYBOARD PORT I/O REGISTERS

Register Name	Address	Functions	
Keyboard Port Status Word	X'0830'	Read	Bits 0-2: Unused. Bit 3: KEYRDY (Keyboard Data Ready). Bits 4-6: Unused. Bit 7: IFLAG (Interrupt Flag).
		Write	Bits 0-1: Unused. Bit 2: KEYRDYIE (KEYRDY Interrupt Enable). Bits 3-7: Unused.
Keyboard Reset	X'0821'	Read	Bits 0-7: Unused.
		Write	Bit 0-7: Unused. A write to this word, regardless of the data written, resets the keyboard port.
Keyboard Data Word	X'0822'	Read	Bits 0-7: Code showing which keys are currently being pressed.
		Write	Bit 0: COMMAND LOCKOUT light. Bit 1: NUMERIC LOCK light. Bit 2: TTY LOCK light. Bit 3: INSERT MODE light. Bits 4-6: Unused. Bit 7: Bell.

BLOCK DIAGRAM

Refer to Figure 6-4. The circuitry on the Communications Interface Board may be grouped into three categories:

- **Processor Interface.** Included in this category are circuits which interface the Host and Keyboard Ports with the terminal's microprocessor. These circuits are: Data Buffers, Address Decoders, and Interrupt Address Decoders.
- The **Keyboard Port,** which interfaces the microprocessor with the the keyboard.
- The **Host Port,** which interfaces the microprocessor with the host computer.

Each of these is described in turn.

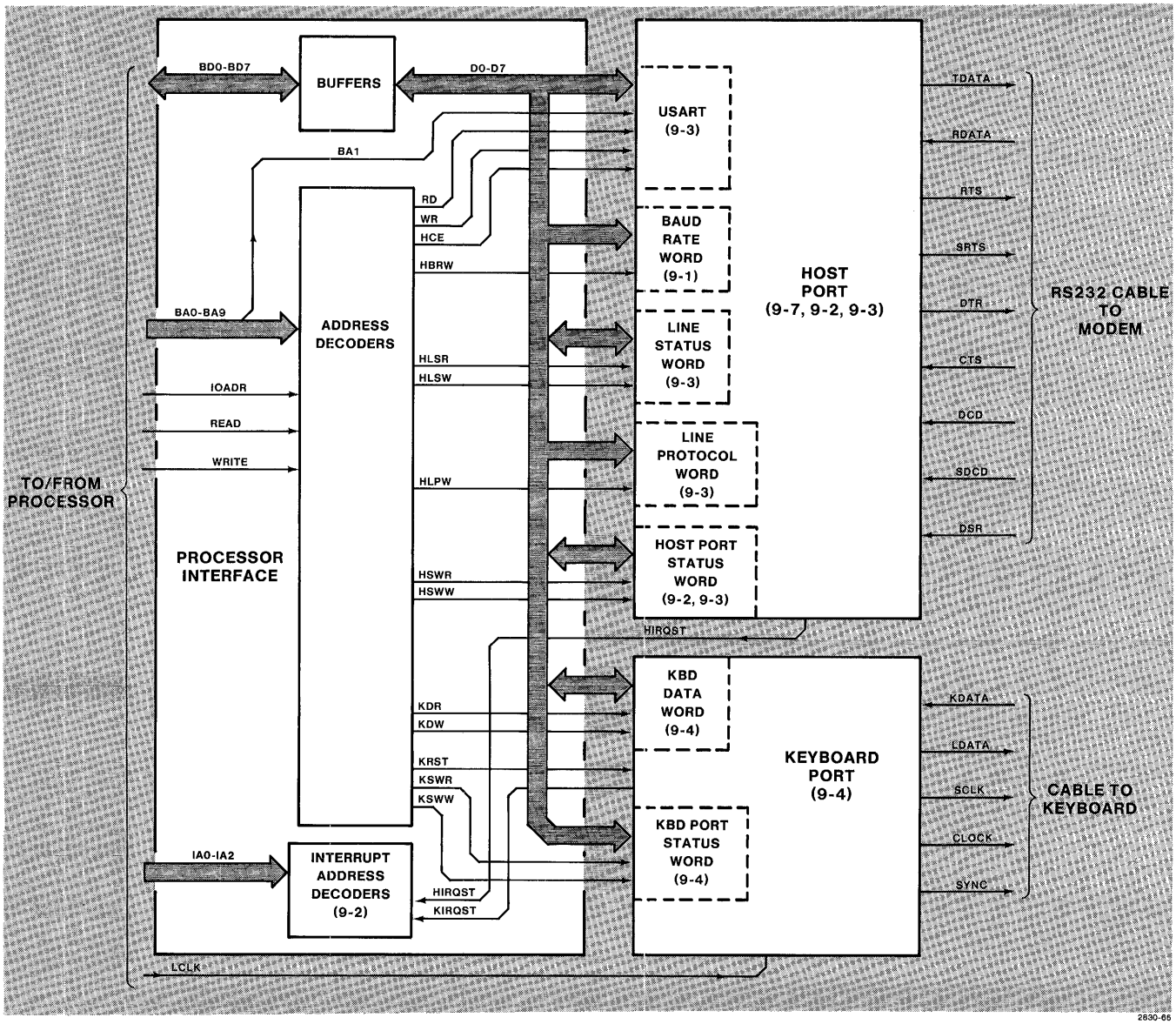


Figure 6-4. Communications Interface Board Block Diagram.

PROCESSOR INTERFACE

Data Buffers

(See Schematic 9-1.)

The Data Buffers buffer the data on the terminal's data bus BD0-BD7, placing it on a local data bus (lines D0-D7) within the Communications Interface Board. During reads from I/O registers on the board, the data buffers buffer the lines D0-D7, placing their signals on the terminal data bus.

The 74LS243 bidirectional Data Buffers are steered by the MYRDADR (My Read Address) signal from the Address Decoders. When MYRDADR is false, the terminal data bus BD0-BD7 drives the local data lines D0-D7. When MYRDADR is true, the microprocessor is reading from the an I/O register on the Communications Interface Board; in that case, the local data lines D0-D7 drive the terminal data bus lines BD0-BD7.

Address Decoders

(See Schematic 9-1.)

The Address Decoders monitor the terminal's address bus (BA0-BA9) and the IOADR, READ, and WRITE lines. The decoder outputs enable the I/O registers in the Keyboard and Host Ports.

The Address Decoder outputs are:

- **MYRDADR (My Read Address).** Signals a read from an address on the Communications Interface Board (an address in the ranges X'0820'—X'0827' and X'0830'—X'0837').
- **KDR (Keyboard Data Read).** True during a read from X'0832'.
- **KDW (Keyboard Data Write).** True during a write to X'0832'.
- **KRST (Keyboard Reset).** True during a write to X'0831'.
- **KSWR (Keyboard Status Word Read).** Signals a read from X'0830'.
- **KSWW (Keyboard Status Word Write).** Signals a write to X'0830'.
- **HCE (Host Port Chip Enable).** Enables the USART during reads or writes at addresses X'0824'-X'0827'.

- **HBRW (Host Baud Rate Word Write).** Signals a write to X'0823'.
- **HLSR (Host Line Status Word Read).** Signals a read from X'0822'.
- **HLSW (Host Line Status Word Write).** Signals a write to X'0822'.
- **HLPW (Host Line Protocol Word Write).** Signals a write to X'0821'.
- **HSWR (Host Port Status Word Read).** Signals a read from X'0820'.
- **HSWW (Host Port Status Word Write).** Signals a write to X'0820'.

Interrupt Address Decoders

(See Schematic 9-2.)

The Interrupt Address Decoders monitor the teminal's IA0-IA2 interrupt address lines and the HIRQST (Host Port Interrupt Request) and KIRQST (Keyboard Port Interrupt Request) signals. When the IA0-IA2 lines are at the Host Port's interrupt address and the HIRWST signal is true, the Interrupt Address Decoder sends the IRQ (Interrupt Request) signal to the Processor. Likewise, IRQ is sent when KIRQST is true and the Keyboard Port's interrupt address is on the IA0-IA2 lines.

HOST PORT

Introduction

The description of the Host Port covers three major topics:

- First, there is an overview of data communications. This includes: asynchronous and synchronous modes, the RS-232 interface, full and half duplex modes, remote and local echo.
- Next, the communications firmware is discussed: how the Host Port is initialized at power-up, how characters are transmitted, how characters are received.
- Finally, the actual circuitry in the Host Port is discussed, circuit block by circuit block.

Data Communications Overview

Asynchronous and Synchronous Modes

There are two main methods of serial data communications: *synchronous* and *asynchronous* modes. The 4025 uses asynchronous mode.

In asynchronous data communications, each character is framed with *start* and *stop* bits (Figure 6-5). When no characters are being sent, the terminal holds the TDATA (Transmit Data) RS-232 line low, in the "mark" or binary 1 condition. Each character begins with a *start bit*, always binary 0 (RS-232 high or "space" condition). Following the start bit are several *selector bits* which determine which character is being sent. (If the ASCII code is used, there are usually eight selector bits: seven bits for the ASCII character—least significant bit first—followed by a parity bit.) Each character ends with a *stop bit*, always binary 1 (RS-232 low or "mark" condition).

The start bit signals the receiving terminal that a character is beginning. The stop bit (or bits) gives the terminal time to recover before the next character comes along. Using the start bit, the receiver re-synchronizes itself at the start of each character. Thus, the the transmitting and receiving terminals can have slightly different data rates and still be able to communicate.

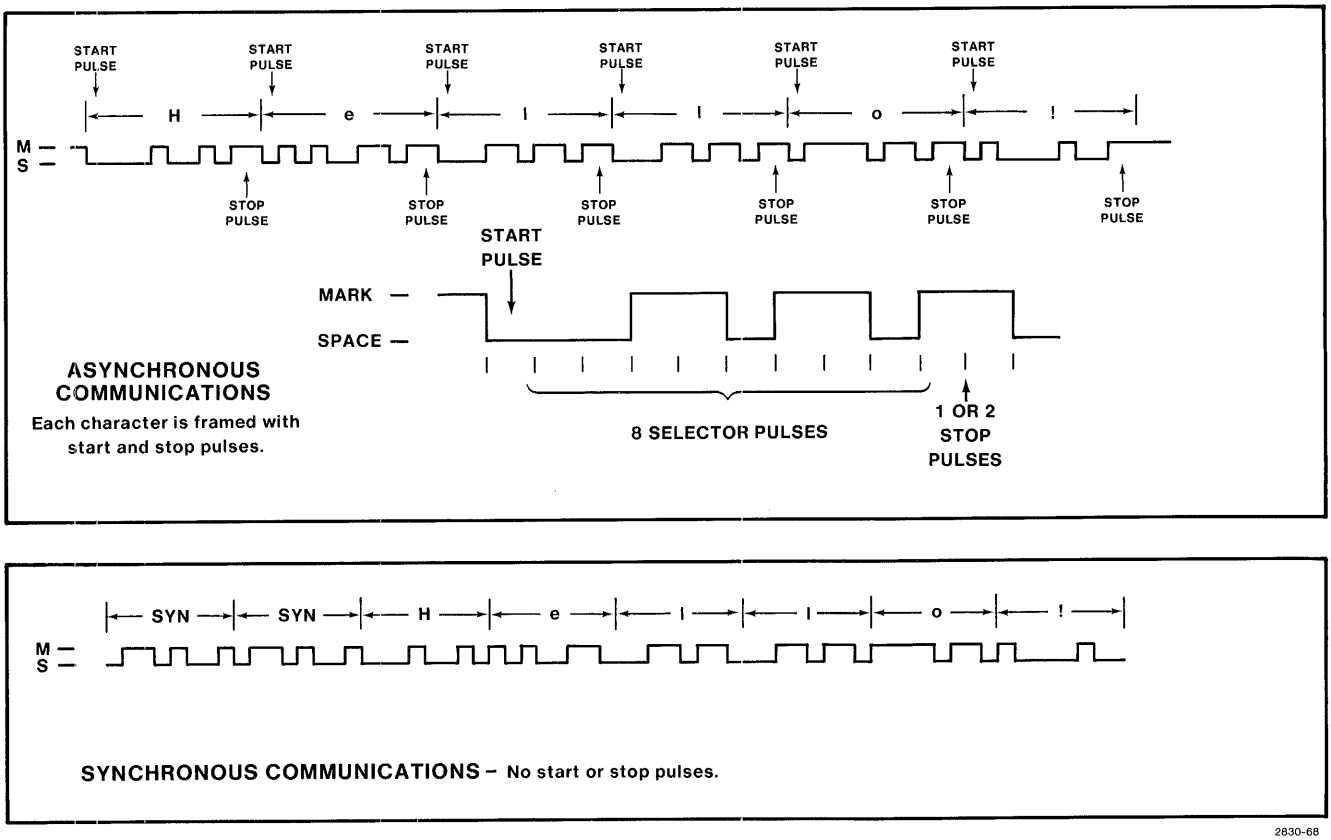


Figure 6-5. Asynchronous and Synchronous Serial Data Formats.

In synchronous data communications, the start and stop bits are omitted. The first bit of one character immediately follows the last bit of the preceding character (Figure 6-5).

NOTE

The 4025 system firmware does not support synchronous data communications. However, there is circuitry on the Data Communications Board for use with synchronous mode, should firmware ever be provided for it.

RS-232 Interface

The signals between a terminal and its modem (or other data communications equipment) are defined in Electronic Industries Association Standard RS-232-C. The voltages at the terminal-modem connector may be between + 3 and + 25 volts (an "RS-232 high"), or between -3 and -25 volts (an "RS-232 low"). Zero volts is commonly interpreted as an RS-232 low. The TDATA and RDATA (Transmitted and Received Data) signals have a low defined as "mark" or binary one, and a high defined as "space" or binary zero. For the other RS-232 signals, a "true" or binary one is a high, and a "false" or zero is low.

The more important RS-232 signals are: TDATA (Transmitted Data), RDATA (Received Data), DSR (Data Set Ready), DTR (Data Terminal Ready), RTS (Request To Send), CTS (Clear To Send), DCD (Data Carrier Detect), SRTS (Secondary Request To Send), SDCD (Secondary Data Carrier Detect), XMT CLOCK, and RCV CLOCK. Usually not all of these signals are used with any particular modem.

- In ordinary asysnchronous full duplex communications, the TDATA and RDATA signals are always used. The following may also be used: DSR, DTR, RTS, DCD.

- In asynchronous half duplex communications (with "supervisory secondary carrier"), the following additional signals are used: SRTS, SDCD.

Full Duplex Mode

Every 4024 and 4025 is equipped with the firmware for *full duplex* data communications. In this communications mode, terminal and modem can send and receive simultaneously. This mode is often used in computer time-sharing systems, especially at data rates of 300 baud or less. Commonly *full duplex remote echo* is used, in which the computer echoes each character sent by the terminal and the echo, rather than the transmitted character, is displayed on the terminal's screen.

However, full duplex communications requires that both "transmit" and "receive" data channels be available. At higher data rates, the simpler types of modems cannot fit both data channels into the same telephone line. One way to overcome this is to use separate telephone lines; another is to use a *half duplex* communication mode.

Half Duplex Modes

When equipped with Option 1, the 4025 offers two half duplex communication modes: *half duplex normal* and *half duplex with supervisor.*

Half Duplex Normal. In half duplex normal mode, the computer and the terminal take turns using the data channel. Normally, the terminal is operating in buffered mode. As the 4025 transmits to the computer, it ends each line with an *end-of-line string* (previously defined by an EOL command). The end-of-line string serves as a *line turnaround character* (or characters): it tells the computer that it may transmit to the 4025. The computer responds by sending some text, which must end with a *prompt string* (previously defined with the PROMPT command). The prompt string serves to turn the line around again: it tells the 4025 that it may send the next line of text.

Half Duplex With Supervisor. To avoid the need for line turnaround characters, often *half duplex with supervisor mode* is used. In this mode, the computer controls the direction of data transfer. To do this, it uses a secondary "supervisory" carrier tone sent over the telephone line. (Sending the SRTS signal to a modem causes the modem to place this secondary carrier on the telephone line.) The process is as follows:

1. Except when sending data to the terminal, the computer listens on the telephone line and sends a secondary carrier tone to tell the terminal that it is listening. (Actually, the computer asserts SRTS and its modem sends the secondary carrier).

   When the 4025's modem hears the secondary carrier, it asserts SDCD (Secondary Data Carrier Detect). So long as SDCD is true, the 4025 stays in the transmit state and asserts RTS (Request to Send), causing its modem to send the primary carrier tone. Even if it has nothing to say, the 4025 must stay in transmit state and send RTS until the computer commands it to enter receive state.

2. When the computer has something to say to the terminal, it turns off the secondary carrier.

   As the secondary carrier vanishes, the 4025's modem drops SDCD (Secondary Data Carrier Detect). The 4025 finishes sending the characters in its USART and then exits transmit mode. As it it does so, it sends SRTS (Secondary Request To Send); this causes its modem to send a secondary carrier tone back to the computer. (The secondary carrier's presence means, "O.K., I'm listening.")

   Within a short time, the computer asserts RTS (Request to Send), and its modem sends the carrier tone to the teminal's modem. The terminal's modem sends DCD (Data Carrier Detect) to the terminal, and the terminal enters "receive" mode. At this point, the computer may start sending data to the terminal.

3. The 4025 remains in receive state until the computer is done sending and turns off the data carrier. When the data carrier is no longer received, the 4025's modem turns off DCD. The 4025 responds by exiting receive mode, turning off SRTS, and turning on RTS to send a carrier tone back to the computer.

4. After a delay (to give the computer time to bring up its secondary carrier) the 4025 enters transmit mode once again and is free to send data to the computer.

Note that the computer controls the direction of data transfer. When the computer places the secondary carrier on the line, the 4025 is required to enter transmit state. Likewise, when the computer turns off the secondary carrier, the 4025 must enter receive state.

If the computer provides *no* carrier tone (neither the primary carrier nor the secondary carrier), the 4025 alternates (or "hunts") between transmit and receive states. While in transmit state, the absence of the secondary carrier forces the terminal into receive state. While in receive state, the absence of the primary carrier forces the terminal back into transmit state.

The operator may request that the computer stop sending by pressing BREAK twice in quick succession. This sends a "break" signal to the computer. (The 4025 drops SRTS, causing its modem to stop sending the secondary carrier. The computer interprets the lack of secondary carrier as a "break.") However, the computer need not honor the break request; it may just keep sending.

Local and Remote Echo

Most time-sharing systems use *full duplex, remote echo* mode, especially at data rates of 300 baud or less. "Full duplex" means that the terminal simultaneously transmits and receives. "Remote echo" means that as the terminal transmits, the computer echoes each transmitted character back to the terminal. It is the echoed characters, not the originally transmitted ones, which are displayed on the screen.

When full duplex communications cannot be used (as at high baud rates over ordinary telephone lines), the remote echo technique must be abandoned. In these cases, the terminal provides its own *local echo* of each transmitted character. Circuitry is provided in the Host Port to generate this local echo.

Host Port Firmware

The following description of the firmware used to drive the Host Port is general in nature; the system firmware programs are not described exactly. However, the general descriptions of the firmware should help you understand how the Host Port circuitry is used.

Initializing the Host Port

On power up, or when MASTER RESET is pressed, the firmware initializes the Host Port. The initialization procedure varies slightly according to the mode of data communications to be used. If full duplex, remote echo mode is to be used, the procedure is as follows:

1. The Processor writes into the Line Protocol Word (address X'0821'), to establish which line protocol will be used. For full duplex, remote echo mode, X'00' is written at this location.

2. The Processor sets the RTS and DTR bits in the Line Status Word by writing at address X'0822'.

3. The Processor writes the baud rates into the Baud Rate Word, address X'0823'. For instance, if the terminal is to transmit and receive at 300 baud, it writes X'DD' into this word.

4. The Processor writes X'00' three times into the USART Command Word, address X'0827'. It then waits 32 microseconds, and writes X'40' into that word. Regardless of the state of the USART on power-up, this procedure is guaranteed to reset the USART, preparing it to receive a command instruction word. The Processor then waits 16 microseconds to give the USART time to reset itself.

5. The Processor writes a *mode instruction word* at address X'0827'. This word tells the USART how many bits per character to expect, what the parity will be, etc. After writing this mode instruction word, the Processor waits 32 microseconds to give the USART time to set itself accordingly.

6. The Processor then writes a *command instruction word* at address X'0827'. This word tells the USART whether to start receiving or transmitting characters. (Subsequent words written at X'0827' will be treated by the USART as command instructions. With these command instructions the Processor controls the USART, telling it to start or stop sending or receiving, or to reset itself to prepare for another mode instruction word. The operation of the USART is described in more detail later in this section.)

7. The Processor reads from the USART Data Word (address X'0825') to clear any garbage bits that may be in the USART input buffer.

8. The Processor writes X'15' into the Host Port Status Word (address X'0820'). This enables the Host Port interrupts.

Receiving Characters

When the terminal receives characters from the host computer, the process is as follows:

1. The input interrupts are enabled at initialization, and are always enabled.

2. When the USART receives a character from the host, the Host Port requests an interrupt by pulling the IRQ line (on the Mother Board) low.

3. The Processor interrupts its current task and reads the character just received. (It reads this character from the USART Data word, address X'0824'.) It places the character in an "input queue" in another part of memory.

4. The Processor reads from the USART Status Word (address X'0826') to see if any of the USART error bits in that word have been set. If so, it places a "flag" in the input queue, indicating that the character just received was in error.

5. As more characters are received, Steps 2 to 4 are repeated again and again.

Later the Processor will process the input queue, doing any of the following:

- It may place the character in the display list, causing it to appear on the screen.

- If the character is part of a command to the terminal, the firmware will interpret the command and execute it.

Transmitting Characters

The following describes how characters are sent when the terminal is operating in unbuffered mode.

1. With the terminal in unbuffered mode, the *output interrupts* (the "transmitter ready" and "transmitter empty" interrupts) are always enabled. To do this, the Processor writes into the Host Port Status Word (address X'0820'), setting the appropriate bits of that word to enable these interrupts. This occurs either at initialization or when the terminal enters unbuffered mode in response to a BUFFERED NO command.

2. The Processor places the characters to be transmitted into an *output queue* or *transmit buffer.*

3. When the USART is ready to send a character, the Host Port interrupts the Processor by pulling the IRQ line (on the Mother Board) low.

4. The Processor interrupts its current task and writes the next character from the transmit buffer into the USART Data Word, address X'0825'.

   The Processor then returns from the interrupt handling routine and resumes its previous task.

5. The USART sends the character out the TDATA line. When it is ready to accept the next character, it generates another "transmitter ready" interrupt.

6. As long as there are characters waiting in the transmit buffer, Steps 4 and 5 will repeat again and again.

7. When the transmit buffer is finally emptied, the Processor turns off the output interrupts (again by writing into the Host Port Status Word, address X'0820').

8. The next time there are characters to be transmitted, the Processor enables the interrupts again. Steps 3 through 7 are then repeated.

Host Port Circuitry

Figure 6-6 is a block diagram for the Host Port. This port includes the following blocks: USART, Baud Rate Word, Baud Rate Generator, RS-232 Receivers and Transmitters, RDATA Logic, Line Status Word (Read), Line Status Word (Write), Line Protocol Word, Host Port Status Word (Read), Host Port Status Word (Write), Debouncer, LSC Detector, Host Port Interrupt Requestor, Line Protocol Logic, and Line Turnaround Logic.

USART

The USART (Universal Synchronous/Asynchronous Receiver/Transmitter) appears in Schematic 9-3 and Figure 6-7. This integrated circuit (Intel 8251A or equivalent) is especially designed for use with the 8080 microprocesor.

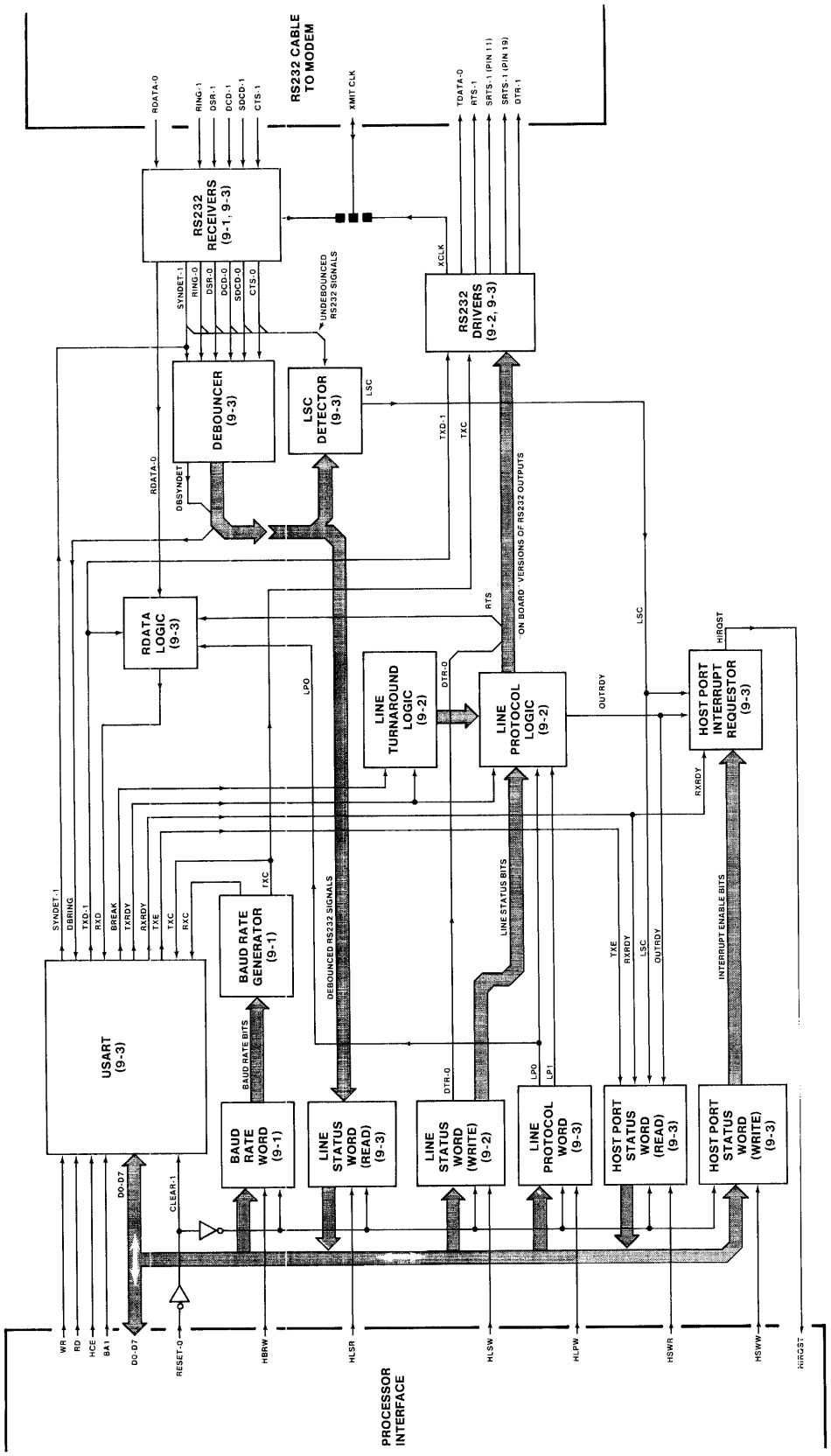


Figure 6-6. Host Port Block Diagram.

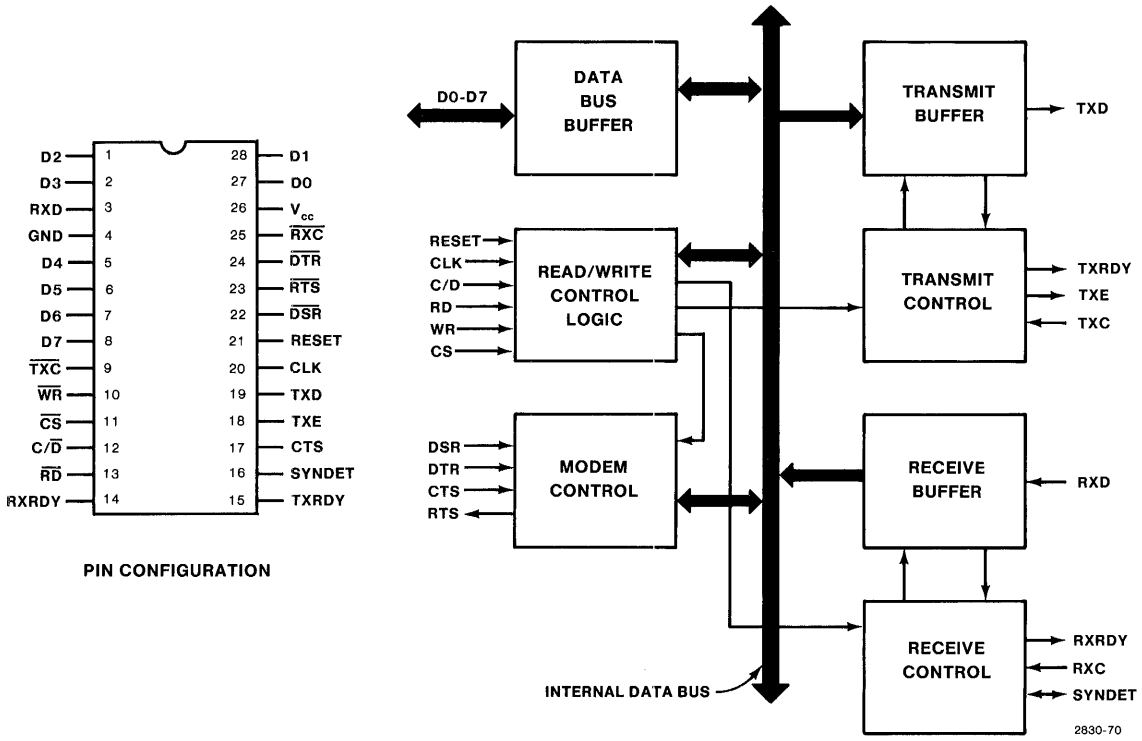


Figure 6-7. 8251A USART.

Location in Memory. The USART is designed to occupy two different words in the 8080's memory. In the 4025, address bit BA1 (rather than BA0) is applied to the USART's register select pin (C/D pin), and bits BA2-BA15 are decoded to drive the chip select pin. The least significant address bit (BA0) is ignored. The result is that each of the USART's two main registers (Command Register, Data Register) occupies two memory addresses, and the USART as a whole occupies four memory addresses. (This is shown in Figure 6-2; the Command Register occupies addresses X'0826' and X'0827', while the Data Register occupies addresses X'0824' and X'0825'.)

Consequently, a read or write at address X'0824' has the same effect as a read or write at X'0825'. Likewise, reading or writing at X'0826' has the same effect as reading or writing at X'0827'.

The two main USART registers each have distinct "read" and "write" halves, with distinct functions. Since each register also occupies two adjacent words of memory, one may pretend that the "read" half of a USART register is one memory location, while the "write" half is a different register. Thus, the USART Data Register can be regarded as two distinct memory words. The firmware always reads input data from address X'0824', while it always writes output data to address X'0825'. Likewise, to ascertain the USART's status, the firmware reads from X'0826', while to program the USART it writes to address X'0827'.

USART Register Functions. The USART Command and Data Registers have the following functions:

- **Input Data.** When the USART receives a character of data on its RXD pin, it stores that character in the "read" half of the USART Data Register and causes the Host Port Interrupt Requestor to request an interrupt. (The Processor is expected to respond to the interrupt by reading from the USART Data Register.)

- **Output Data.** To send a character of text out the TDATA line, the Processor writes into the "write" half of the USART Data Register.

  (Once the character has been sent, the USART will cause the Host Port Interrupt Requestor to request an interrupt. When the Processor services the interrupt, (by reading from the USART Status Register), it learns that the character has been sent and can then write the next character into the USART Data Register.)

- **USART Status.** By reading from the USART Status Word (the "read" half of the USART Command Register), the Processor can learn whether certain events have occurred:

  Bit 0: TXREADY. This bit is high (binary 1) when the USART is ready to accept another character. (Because the USART is "double buffered," it can accept another character at the same time as it is transmitting the previous character.)

  Bit 1: RXRDY. This bit is high when when the USART is ready to receive a character.

  Bit 2: TXE (Transmitter Empty). This bit is high when the USART is empty (has sent all the characters it has been given to transmit). The USART can accept a character, start sending it, and, while sending, signal TXREADY and accept another character. TXE will not be sent, however, until all characters given the USART have been transmitted.

  Bit 3: PE (Parity Error). The PE bit is set when the USART detects a parity error in a character it has received.

  Bit 4: OE (Overrun Error).

  Bit 5: FE (Framing Error). The FE bit is set when the USART fails to detect a stop bit in a character it has received.

  Bit 6: SYNDET. This signal is not used. (It is reserved for use in synchronous data communications.)

Bit 7: Tells whether a high or low is present at the USART's DSR pin (USART Pin 22). This pin need not be connected to the RS-232 DSR line, and in the 4025 it is connected to the RING line instead. Thus, by reading Bit 7 of the USART Status Word, the Processor can ascertain whether the modem is presenting a RING signal to the terminal.

- **USART Programming.** By writing into the USART Programming Word (the "write" half of the USART Command Register), the Processor can pass commands to the USART.

  After the USART is reset, the first word written here is the *mode instruction word.* This word determines the USART's operating mode.

  The 4025 operates only in asynchronous mode. In that mode, all succeeding words written at this address are treated by the USART as *command instruction words*, as described earlier under "Host Port Firmware."

Baud Rate Word

The Baud Rate Word (Schematic 9-1) is an eight-bit latch. It is loaded from the data bus when clocked by HBRW (Host Baud Rate Write).

Baud Rate Generator

The Baud Rate Generator (Figure 6-8, Schematic 9-1) provides two clocks, RXC and TXC, for the receive and transmit sections of the USART. These are "16X clocks;" their frequencies are 16 times the receive and transmit baud rates. The data in the Baud Rate Word selects which of several standard frequencies are used for the RXC and TXC clocks. (Baud Rate Word bits 0-3 select the TXC frequency; Bits 4-7 select the RXC frequency.)

The heart of the Baud Rate Generator is an integrated circuit (Fairchild F4702 or equivalent). This IC includes: a 2.4576 MHz oscillator, frequency dividers to provide 16X clocks for the standard baud rates, and a data selector which selects one of these clocks for output.

Used by itself, the F4702 could provide one of the two USART clocks. Circuitry external to the F4702, however, makes it do double duty, providing two different clock frequencies. This circuitry takes advantage of the fact that the F4702 has internal clock signals for all the standard baud rates, even though only the selected clock is presented to the F4702's Z output.

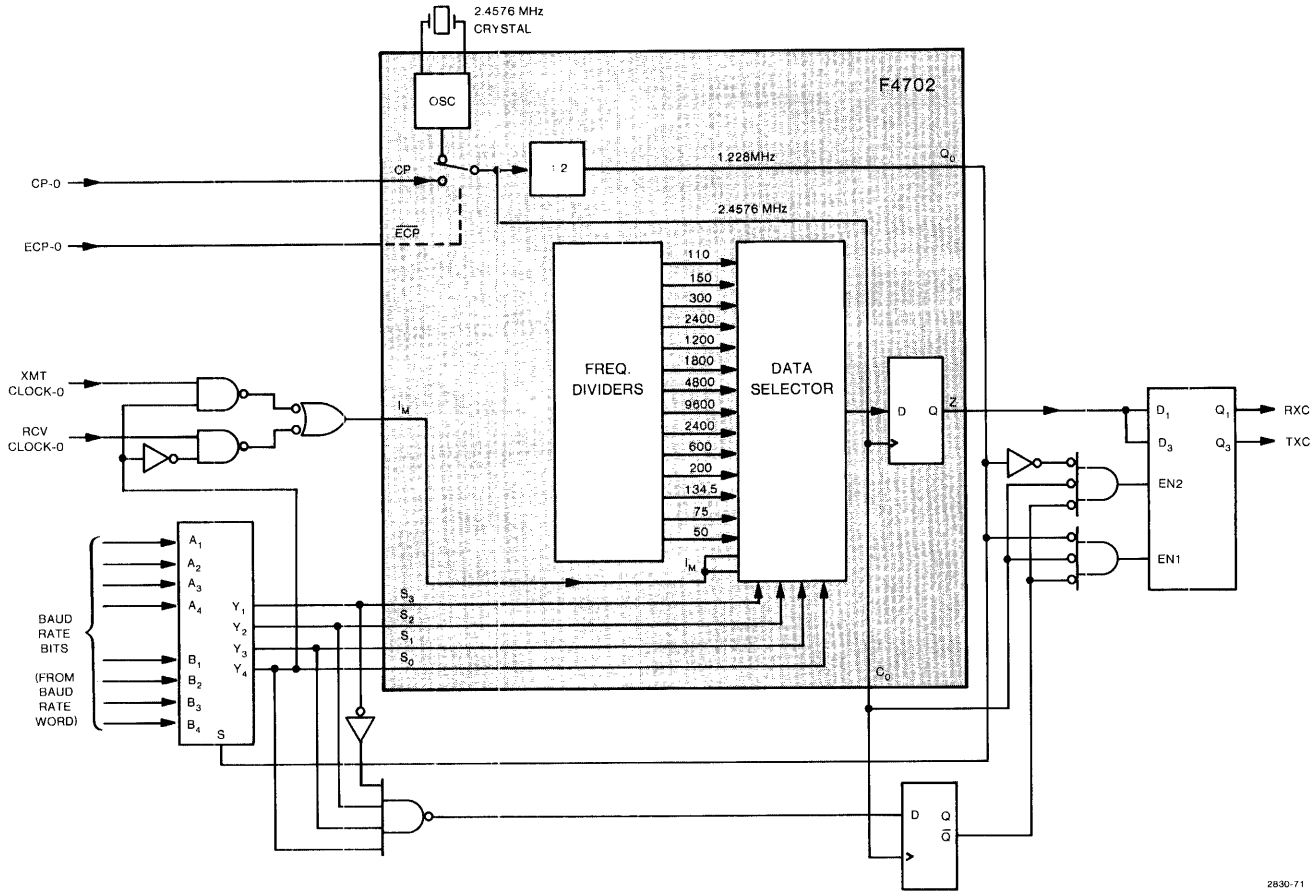


Figure 6-8. Baud Rate Generator.

The way it works is this:

The F4702 Q0 1.2288 MHz output rapidly switches the F4702's internal data selector between two of its internal clocks. The C_O 2.4576 MHz output latches the state of the selected clock into one of two flip-flops in a 74LS75. The state of one internal clock is sampled and held on the 74LS75's Q1 output, which drives the RXC line. The other of the two internal clocks is sampled and held on the Q3 output, which drives the TXC line.

As just mentioned, the F4702 Q0 signal determines which of the two internal clocks is being sampled. To do this, the Q0 output steers two devices:

- First, the F4702 Q0 output steers a 74LS273 data selector. When Q0 is low, the data selector places the "receive" bits of the baud rate word on the F4702's speed select inputs S0-S3. When Q0 is high, the data selector places the "transmit" baud rate bits on Pins S0-S3.

  When Q0 is low, the internal data selector samples the "receive" baud rate clock; the next oscillator C_O pulse places the state of the "receive" clock on the Z output. When Q0 is high, the internal data selector samples another baud rate clock: the "transmit" baud rate clock. The Next C_O pulse places the state of the "transmit" clock on the Z output.

  Thus, the F4702's Z output is multiplexed between the desired two clock signals.

- Secondly, the Q0 output enables one or the other of the two flip-flops which sample the F4702 Z output. When Q0 is low, it enables the Q1 flip-flop, which samples and holds the receive clock waveform. When Q0 is high, it enables the Q3 flip-flop, which samples and holds the transmit clock waveform.

  Thus, the Q0 output is used to control the demultiplexing of the F4702 Z output signal. This demultiplexing separates the the two clock signals being alternately presented on the Z output pin. The receive clock is gated to the RXC line, and the transmit clock is gated to the TXC line.

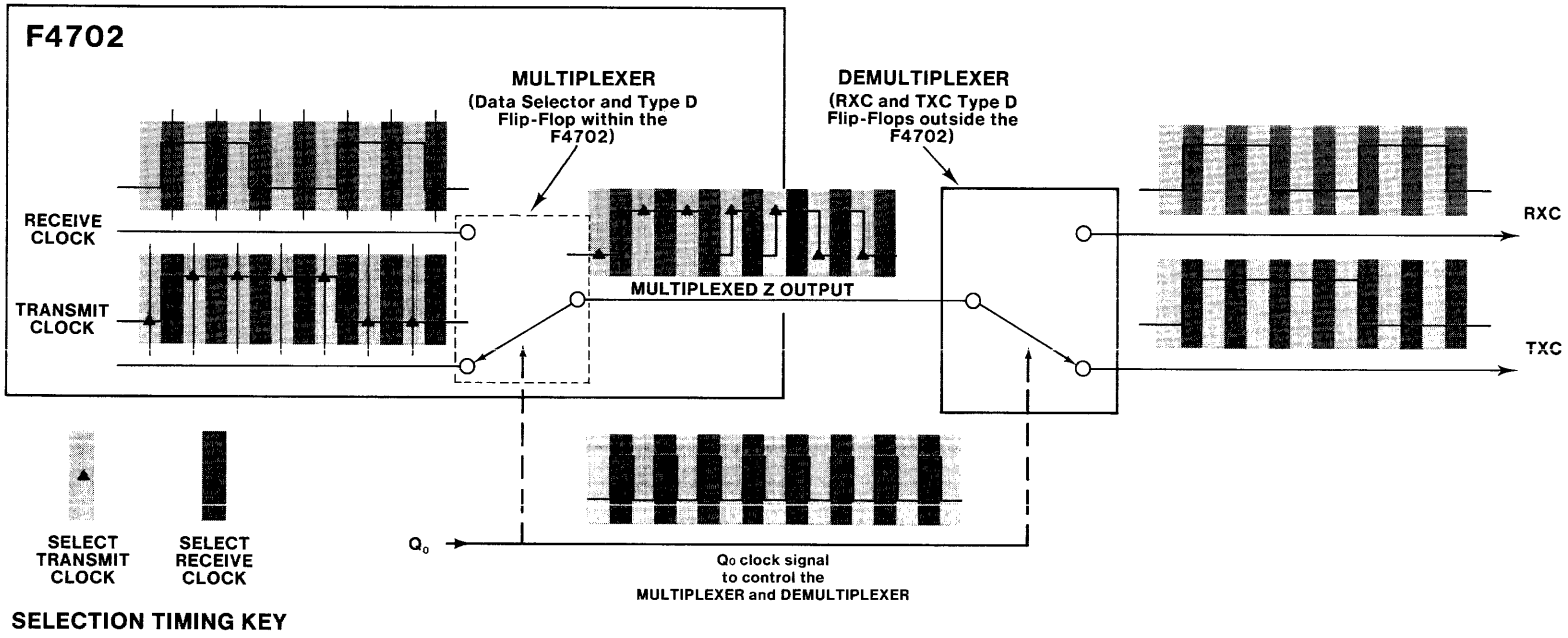
Figure 6-9 summarizes this process. The Q0 line steers a multiplexer (most of whose parts are within the F4702 integrated circuit). The multiplexer output is the F4702 Z output, which is then de-multiplexed to provide the RXC and TXC clock signals. The demultiplexer is steered by the same signal (that is, Q0) as the multiplexer.

The F4702 S0-S3 selector inputs have sixteen possible settings. Three of these have special meanings, as follows:

- Settings binary 0000 and 0001 pass the signal on the F4702 I_M pin on the the Z output pin. Logic external to the F4702 monitors the S0 input line. When S0 is low, the external RCV CLOCK is applied to the I_M input; when S0 is high, the external XMT CLOCK is applied.

- Two settings of S0-S3 cause the F4702 to provide a 2400 baud clock. One of these (binary 0111) is not used for this purpose by the firmware. Instead, this setting (0111) is detected by a NAND gate. The NAND gate's output is delayed slightly by a type D flip-flop to match the delay in the F4702 output caused by a similar flip-flop within the F4702. The delayed NAND gate output disables the 74LS75, which stops the baud rate clock.

The result is that the Baud Rate Generator provides 16X clocks for the USART according to the bits stored in the Baud Rate Word, as summarized in Table 6-3.



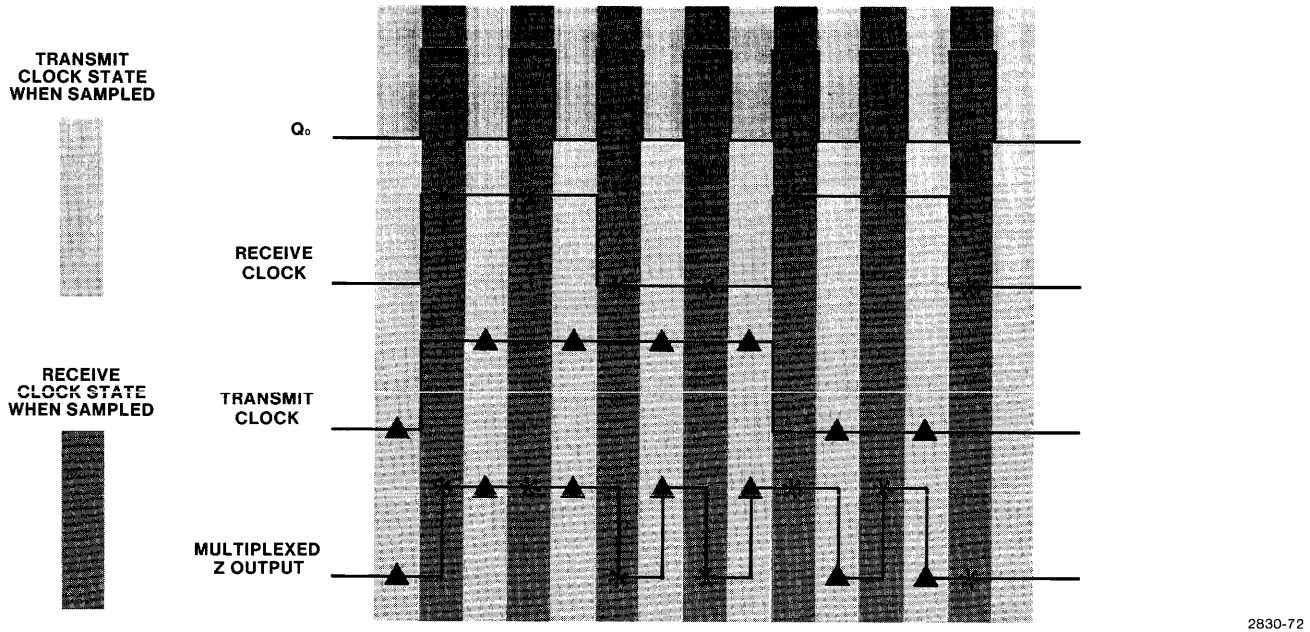


Figure 6-9. Multiplexing and Demultiplexing the Baud Rate Clocks.

Table 6-3

BAUD RATE GENERATOR PROGRAMMING

Baud Rate Word Bits	Baud Rates
Transmit: bits 7 6 5 4 Receive: bits 3 2 1 0	
0 0 0 0	RCV CLOCK external clock
0 0 0 1	XMT CLOCK external clock
0 0 1 0	50 baud
0 0 1 1	75 baud
0 1 0 0	134.5 baud
0 1 0 1	200 baud
0 1 1 0	600 baud
0 1 1 1	0 baud (stops the clock)
1 0 0 0	9600 baud
1 0 0 1	4800 baud
1 0 1 0	1800 baud
1 0 1 1	1200 baud
1 1 0 0	2400 baud
1 1 0 1	300 baud
1 1 1 0	150 baud
1 1 1 1	110 baud

RS-232 Receivers and Drivers

The RS-232 Receivers and Drivers are inverting buffers which interface between the RS-232 signal levels (typically + 12 V for "high" and -12 V for "low") and the TTL signal levels used within the the Data Communications Interface Board (0 V to 0.6 V for "low" and + 3.5 V to + 5.0 V for "high"). The RS232 Receivers and Drivers appear on schematic pages 9-1, 9-2, and 9-3.

RDATA Logic

The RDATA Logic appears in Schematic 9-3. It controls the local echoing of transmitted data, as follows:

- If data is being transmitted (in which case RTS is true) and the Data Communications Board's Host Port is to provide local echo (Line Protocol Bit LP1 is set), then the RDATA signal is not passed to the USART RXD input. Instead, the data being transmitted from the USART TXD output is looped back and presented to the RXD input.

- Otherwise, it is the data coming in on the RDATA line which is presented to the USART's RXD input.

Line Status Word (Read)

The "read" half of the Line Status Word is a 74LS368 tri-state buffer (Schematic 9-3). When enabled by the HLSR signal, the states of DSR, SDCD, DCD, and CTS are placed on the data bus.

Line Status Word (Write)

The "write" half of the Line Status Word is a 74LS175 integrated circuit holding four type D flip-flops. When clocked by the HLSW signal, it reads data lines D4 to D7 and stores them in its flip-flops. The flip-flop outputs drive the following signals in the Host Port:

- Bit 4 drives the RS-232 DTR (Data Terminal Ready) line.
- Bit 5 can be used to drive the SRTS (RS-232 Pin 11) line, if strap "C-D" is wired in position C. Normally, however, this strap is wired in position D, and the Half Duplex Logic drives this line in parallel with the SRTS (RS-232 Pin 19) line.
- Bit 6 drives the REQSRTS line. (REQSRTS requests the Half Duplex Logic to send the SRTS signal to the modem.)
- Bit 7 drives the REQRTS line. (REQRTS requests the Half Duplex Logic to send the RTS signal to the modem.)

Line Protocol Word

The Line Protocol Word is a write-only register with only two bits (data bits 0 and 1). It consists of two type D flip-flops (Schematic 9-3) which are loaded with data bits 0 and 1 when clocked by the HLPW (Host Line Protocol Write) signal. The flip-flop outputs drive the LP0 and LP1 lines, which specify which of several alternate line protocols is to be used. Bit LP1 controls the local echo feature, and the two bits together inform the Half Duplex Logic whether Half Duplex, Half Duplex with Supervisor, or Full Duplex mode is to be used.

Table 6-4

LINE PROTOCOL BITS

LP1	LP0	Data Communication Mode
0	0	Full Duplex, no local echo. Any echo must be provided by the host (remote echo).
0	1	Half Duplex with Supervisor. Any echo is provided by the firmware, not the Host Port circuitry.
1	0	Full Duplex, local echo provided by the Host Port.
1	1	Half Duplex without supervisor, local echo provided by the Host Port.

Host Port Status Word (Read)

The "read" half of the Host Port Status Word (Schematic 9-3) uses five tri-state buffers in a 74LS367. When enabled by the HSWR signal, these buffers place bits on the data bus, as follows:

- Bit 1: XMTRDY. Set when the USART transmitter is ready to accept another character.
- Bit 3: RXRDY. Set when the USART has received a character and is ready for the Processor to read that character from its Data Word.
- Bit 5: LSC (Line Status Change). Indicates when the state of one of the RS-232 signal lines has changed. (When the Processor reads from the Host Port Status Word, this bit is automatically cleared.)
- Bit 6: TXE (Transmitter Empty). Indicates when the USART has finished sending all the characters it has been given to transmit.
- Bit 7: HIRQST. This bit is set whenever the Host Port is requesting an interrupt.

Debouncer

The Debouncer prevents transient noise (such as that due to switch contact bounce) from being detected as line status changes and causing Processor interrupts. The Debouncer does this for the RING, DSR, DCD, CTS, and SDCD RS-232 signal lines, and for the SYNDET signal from the USART.

The Debouncer (Schematic 9-3) consists of six type D flip-flops in a in a 74C174 integrated circuit. The flip-flops act as one-stage shift registers, so that their outputs lag one LCLK pulse behind their inputs. In addition, the slow response time of the 74C174's CMOS circuitry removes any short-duration noise. Consequently, the Q outputs of the flip-flops are noise-free, but lag slightly behind the D inputs. The LSC Detector takes advantage of this slight delay.

LSC Detector

The LSC (Line Status Change) Detector is shown in Schematic 9-3. It consists of several NOT-XOR gates and a flip-flop.

Each NOT-XOR gate is connected to an a Debouncer input and the corresponding Debouncer output. When the status of that particular line changes, there will be a brief period when, due to the propagation delay through the Debouncer, the input and output have different states. When that occurs, the NOT-XOR gate's open-collector output goes low.

The NOT-XOR gates are connected in a "wired-OR"configuration to drive the preset input of the flip-flop. Thus, the flip-flop is set whenever a line status change occurs on the RING, DSR, SYNDET, DCD, CTS, or SDCD signal lines. The flip-flop will be cleared when the Processor reads from the Host Port Status Word (by the HSWR signal) or on power-up or MASTER RESET (by the CLEAR signal).

Host Port Interrupt Requestor

The Host Port Interrupt Requestor (Schematic 9-3) consists of three open-collector NAND gates connected in a wired-OR configuration. These gates send the HIRQST (Host Port Interrupt Request) signal whenever any of the following occur:

- The XMTRDY interrupt is enabled (Host Port Status Word Bit 0 is set) and the XMTRDY signal occurs.
- The RXRDY interrupt is enabled (Host Port Status Word Bit 2 is set) and the USART sends the RXRDY signal.
- The LSC interrupt is enabled (Host Port Status Word Bit 4 is set) and the LSC Detector sends the LSC signal.

Line Protocol Logic

(See Schematic 9-2.)

The Line Protocol Logic controls the "handshaking" (exchange of control signals) between the terminal and its modem. Different handshaking rules apply in each of the three communications modes: full duplex, half duplex normal (without supervisory channel), and half duplex with supervisor.

- **Full Duplex Mode.** This mode is selected when the Line Protocol Bit LP0 is zero. The RS-232 RTS, SRTS, and DTR signals are under direct firmware control. (The bits in the write half of the Line Status Word are passed directly to their respective RS-232 signal lines.) The USART's TXRDY signal is passed directly to the XMTRDY line, so that the Processor sees an OUTRDY state whenever the USART is ready to accept another character.

- **Half Duplex, Local Echo.** (LP1 and LP0 are both ones.) RTS, SRTS, and DTR are under direct firmware control. The Processor sees an OUTRDY condition only when the USART is sending its TXRDY signal and modem is sending CTS. Whenever RTS is true, transmitted data is echoed back to the receiver.

- **Half Duplex With Supervisor, No Echo.** The DTR signal remains under direct firmware control. The RTS and SRTS signals are controlled by the Line Turnaround circuit, described later. The Processor sees an OUTRDY only when all the following conditions are met: (a) the Line Turnaround Logic is in its "transmit" state (RTS true, SRTS false), (b) the USART is ready (TXRDY true), and (c) the modem is signalling "clear to send" (CTS true).

  When a BREAK is received (and the Line Turnaround Logic is in the "receive" state), SRTS will go false to notify the host computer. (As SRTS goes false, the modem stops sending the secondary carrier. This causes the computer's modem to stop sending the SDCD signal to the computer.)

A 74LS153 dual data selector controls the OUTRDY and SRTS signals. Line Protocol Bits LP0 and LP1 steer the data selector as follows:

- LP1 = 0, LP0 = 0; Full Duplex, No Echo. The data selector's C0 inputs are selected. OUTRDY is driven by the USART's TXRDY signal, and SRTS is driven by the REQSRTS bit from the Line Status Word.

- LP1= 1, LP0= 0 or = 1; Full Duplex, Local Echo or Half Duplex, Local Echo. The data selector's C2 or C3 inputs are selected. OUTRDY is sent only when the USART is empty (TXRDY true) and CTS is true. As before, SRTS is driven by the REQRTS bit from the Line Status Word.

- LP1= 0, LP0= 1; Half Duplex Supervisor. The data selector's C1 inputs are selected. OUTRDY and SRTS are driven by the Line Turnaround Logic.

The RTS signal is driven by a logic tree which uses a 74LS51 AND-OR-INVERT gate, a negative-logic OR gate (inverting input, inverting output OR gate), and an inverter.

- Whenever LP0= 0 or LP1= 1, half duplex supervisor mode is *not* being used. In this case, the RTS signal is driven by the REQRTS signal from the write half of the Line Status Word.

- When LP0= 1 and LP1= 0, the terminal is in half duplex supervisor mode. In this case, RTS is governed by an output of the Line Turnaround Logic.

Line Turnaround Logic

(See Schematic 9-2.)

The Line Turnaround Logic functions when the terminal is in "half duplex supervisor" mode. In this mode, data transfer occurs in only one direction at a time; while the 4025 is sending to the computer, the computer cannot send back to the 4025—and vice versa. The Line Turnaround Logic controls the process of changing the direction of data transfer.

The Line Turnaround Logic (Schematic 9-2) has four different states or conditions during the data transmission cycle. These states are represented by the four possible states of flip-flops Q0 and Q1:

State Name	Flip-Flops	Description
	Q1 Q0	
A	0 1	"Receive" state.
B	1 1	Changing from "receive" to "transmit."
C	1 0	"Transmit" state.
D	0 0	Changing from "transmit" to "receive."

The Line Turnaround Logic is implemented as a "state machine." Flip-flops Q0 and Q1 designate the states. TTL logic gates change the state machine from one state to another by controlling flip-flop inputs. Figure 6-10 is a state diagram summarizing the design of this state machine. The following text explains the state diagram in more detail:

- **State A ("Receive" State).** The Line Turnaround Logic remains in "receive" state so long as the modem continues to send the DCD (Data Carrier Detect) signal. (DCD true means the computer's modem is using the data communications channel.)

  When the computer stops sending data to the terminal, it turns off its data carrier. The terminal's modem responds by turning off the DCD signal. When the Line Turnaround Logic sees DCD drop, it exits State A and enters State B.

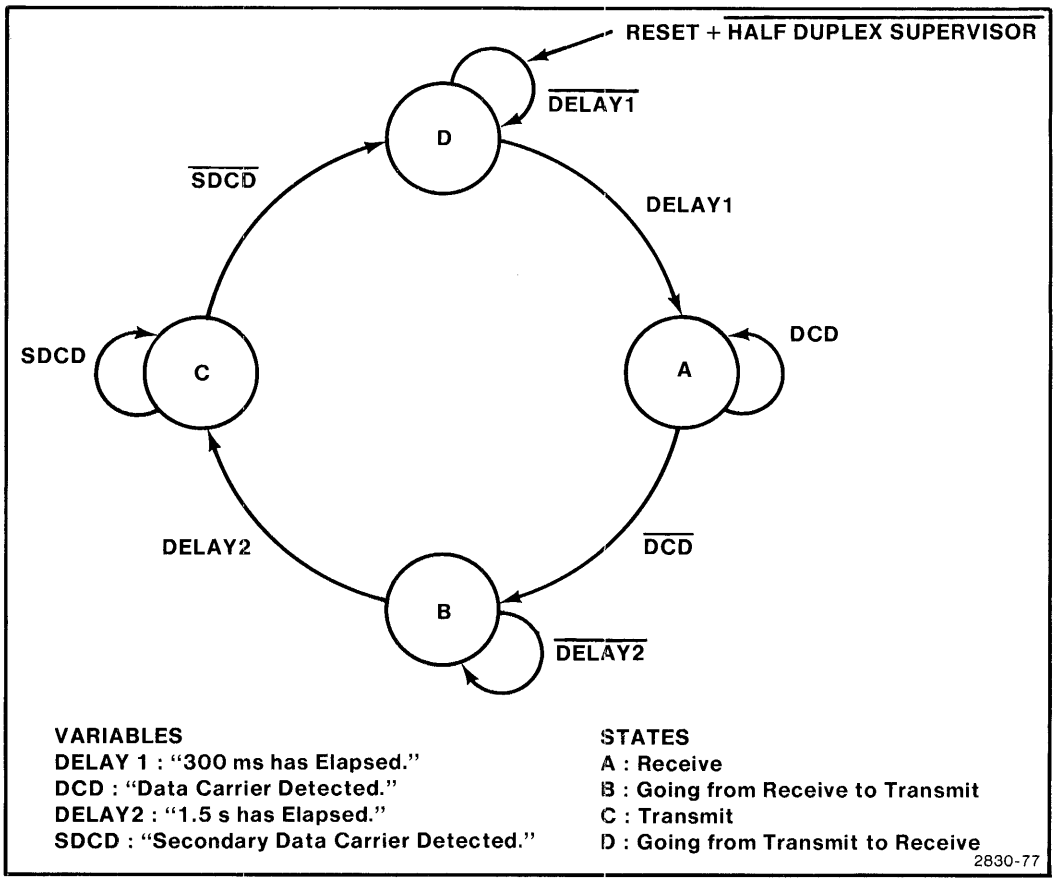


Figure 6-10. Line Turnaround State Diagram.

- **State B (Changing From Receive to Transmit).** In State B, the Line Turnaround Logic sends the RTS signal, requesting the modem to place a carrier tone on the telephone line. The Line Turnaround Logic then waits about 1.5 seconds in State B for a secondary "supervisory" carrier on the telephone line. The secondary carrier indicates that the computer is ready to receive data from the terminal.

  After the 1.5 second delay, the Line Turnaround Logic exits State B and enters State C.

- **State C ("Transmit" State).** In State C, the Line Turnaround Logic continues to send the RTS signal. It is during this state the the terminal transmits data to the computer. The Line Turnaround Logic remains in "transmit" state so long as the SDCD signal is present.

  Should the computer need to use the data communications channel, it will disable the secondary carrier tone. Consequently, the 4025's modem will stop sending SDCD. The Line Line Turnaround Logic waits until the USART has emptied its transmit buffers and sent the TXE signal. As soon as SDCD is false and TXE is true, the Line Turnaround Logic exits State C and enters State D.

- **State D (Changing From Transmit to Receive).** In State D, the Line Turnaround Logic stops sending RTS and waits about 300 ms to give the computer's modem a chance to respond by placing a carrier on the telephone line. After the 300 ms delay, the Line Turnaround Logic exits State D and enters State A.

State Machine Operation. On power-up, the CLEAR signal resets both flip-flops, putting the state machine in State D. The J and K flip-flop inputs are: J1= 0, K1= 1, J0= DELAY1, K0= 0. The 200 ms timer, triggered by flip-flop Q1's inverting output, starts to time a 200 ms pulse. During the 200 ms period, DELAY1 is false, keeping J0= 0. With J0= K0= 0, Flip-flop Zero does not change state, and the state machine stays in State D.

When the one-shot times out, however, DELAY1 goes true, and J0 goes high. The next LCLK pulse sets the flip-flop, and the state machine enters State A.

In State A, the "receive" state, we have J0= 1, K0= 0, J1 = not DCD, K1= 0. That is, flip-flop Q0 will not change state since its J and K inputs are both low, and flip-flop Q1 will not change state so long as DCD is true (and "not DCD" at its J input is low). Thus, so long as the computer's modem sends the data carrier (and the terminal's modem asserts DCD), the state machine remains in the "receive" state.

When DCD goes false, Q1's J input goes high. The next LCLK pulse sets the flip-flop, and the state machine enters State B.

Entering State B, triggers the one-second timer. This one-shot sends DELAY2 false until it times out, whereupon it sends DELAY2 true. In State B, the flip-flop J and K inputs are: J1= "not DCD", K1= 0, J0= 0, K0= DELAY2. Since flip-flop Q1 is already set in State B, what happens to its J input is irrelevant; the flip-flop stays set whether DCD is true or false. The K input of flip-flop Zero, however, does matter. So long as DELAY2 is false, the Q0 flip-flop stays cleared. When the one-shot times out, K0= DELAY2 goes true, and the flip-flop is set. This moves the state machine into State C.

In State C (the "transmit" state), J1= 0, K1= "not SDCD and TXE", J0= 0, and K0= 1. Since J0= 0 and K0= 1, flip-flop Q0 stays cleared. Flip-flop Q1 stays set only as long as its K1 input stays low. That is, the state machine stays in State C only so long as either the USART transmit buffer is not empty (TXE false) or a secondary "supervisory" carrier is being received from the computer's modem (SDCD true, SDCD false).

Should the host computer need to use the data communication line, it will send a "break" by turning off the secondary carrier. SDCD goes false (and "not SDCD" goes true). As soon as the 4025 USART is done transmitting (signalled by TXE), J1 goes true. The next LCLK pulse will then clear the flip-flop, putting the state machine in State A.

As noted before, the state machine stays in State A for 300 ms, and then enters State B, the "receive" state.

KEYBOARD PORT

Introduction

The keyboard sends and receives data in a serial format. It repeatedly sends (on the KDATA line) a 128-bit "character" whose individual bits tell whether keyswitches are open or closed. The Keyboard Port monitors the incoming stream of KDATA bits and interrupts the Processor whenever a key is pressed or released. The Processor then learns which key has been pressed or released by reading from the Keyboard Data Word.

CLOCK and SYNC signals from the Keyboard Port keep the stream of KDATA bits synchronized with the Character Counter which decodes them.

Likewise, the Keyboard Port uses a serial format to send "lights and bell" data to the keyboard on the LDATA line. It provides the SCLK (Shift Clock) signal for clocking the LDATA bits into the keyboard's shift register and latches.

Keyboard Port Firmware

The following description explains how the circuitry is used. Two topics are described: how the Processor reads characters typed on the keyboard, and how it controls the keyboard lights and bell.

Receiving Characters From the Keyboard.

As you press or release a key, the following occurs:

1. One of the bits changes in the 128-bit "character" being sent from the keyboard on the KDATA line. The bit's new state shows the new position of the key.

2. The Keyboard Port monitors the stream of KDATA bits, and notes the change in the bit representing that key. It stores an 8-bit word in the "read" half of the Keyboard Data Word (address X'0832'). Seven of the bits name the key which moved. The eighth bit tells whether the key was pressed (bit = 1) or released (bit = 0).

3. Simultaneously, the Keyboard Port requests a Processor interrupt.

4. The Processor interrupts its current task and reads from address X'0832'. It consults a table (stored elsewhere in memory) to ascertain the key's definition. The definition may be an ASCII character (a binary number between 0 and 127) or a function key code (a binary number between 128 and 255). If the key has been programmed, its definition may be a string of several ASCII characters or function key codes.

5. The Processor stores the key's definition in its input queue (another part of memory).

6. The Processor returns from the interrupt handling routine and resumes its previous task.

Later the Processor examines its input queue and does one or more of the following:

- Places characters from the input queue in the display list, so that they may be displayed on the screen.
- Places characters from the input queue in the transmit buffer, so that they may be sent to the host computer.
- Interprets strings of characters as commands and executes the commands.

Controlling Lights and Bell

The Processor controls the keyboard lights and the terminal's bell (which is in the keyboard) by writing into the Keyboard Data Word, address X'0832'. In this word, Bit 0 is set if the bell is to be rung, and Bits 4 through 7 each control one of the four function key lights. Bits 1 through 3 are unused.

Thus, to ring the bell the Processor first writes into X'0832', setting Bit 0 = 1. To turn off the bell, it writes into X'0832', setting Bit 0 = 0.

Keyboard Port Circuitry

Figure 6-11 is a block diagram for the Keyboard Port. The following circuit blocks are for receiving keyswitch information from the keyboard: Clock Counter, Keyswitch Counter, Sync Generator, Key Status Change Detector, and Keyboard Data Word (Read). The remaining blocks control the keyboard lights and bell: Keyboard Data Word (Write), SCLK Generator, Shift Register. Each block is discussed in turn.

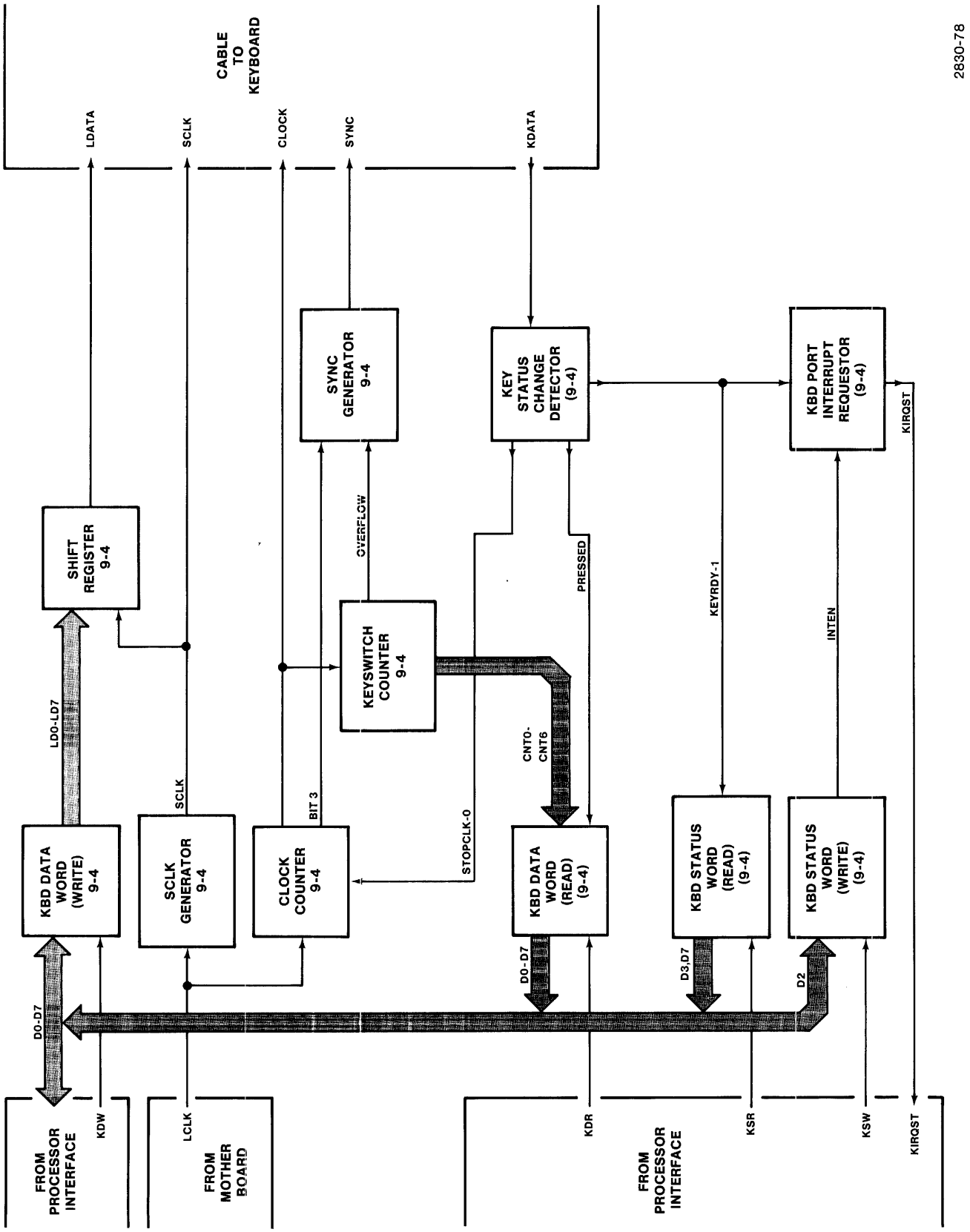


Figure 6-11. Keyboard Port Block Diagram.

Clock Counter

(See Schematic 9-4.)

The Clock Counter divides the 2.048 MHz clock (LCLK) by 16 to provide the 128 kHz CLOCK signal for the Keyboard and the Keyboard Port. The CLOCK is stopped (counter cleared) whenever a change in position of any keyswitch is detected (KEYRDY condition). (When the Processor reads from Keyboard Data Word, the KEYRDY condition is cleared and the CLOCK starts again.)

The counter is also cleared on power-up or when MASTER RESET is pressed, and the Processor may clear it by writing into the Keyboard Reset Word.

Keyswitch Counter

(See Schematic 9-4.)

The Keyswitch Counter is stepped by the CLOCK pulses together with a similar counter within the keyboard's Key Scanner. The least significant seven bits of the count (outputs $1Q_A$ to $2Q_C$) specify which key is currently being scanned. That is, these seven bits tell which key's position is currently being reported on the KDATA line.

The counter's most significant bit (output $2Q_D$) is used to trigger the Sync Generator.

Sync Generator

(See Schematic 9-4.)

The Sync Generator consists of two flip-flops which provide a SYNC signal for the Keyboard and a KBDSYNCD signal for the keyboard Port.

The first flip-flop is set when the Keyswitch Counter's $2Q_D$ output goes high. At this time, the Keyswitch Counter's least significant bits are all zero, and the similar counter in the keyboard's Key Scanner is reset to zero. The flip-flop output is the SYNC pulse for the keyboard's Key Scanner.

The second flip-flop drives the KBDSYNCD (Keyboard Synchronized) line. On power-up, or after the Processor resets the Keyboard Port, this flip-flop is cleared, sending KBDSYNCD false. The first SYNC pulse to occur sets the flip-flop, sending KBDSYNCD true. Until KBDSYNCD goes true, the Host Port will not generate KEYRDY interrupts.

Key Status Change Detector

(See Figure 6-12, Schematic 9-4.)

The Key Status Change Detector looks for changes in the keyswitch positions reported on the KDATA line. On detecting a valid change in keyswitch position, it sets a flip-flop and sends the KEYRDY signal. This stops the CLOCK (clears the Clock Counter) so that the Character Counter holds its current count, which can be read from the Keyboard Data Word. If KEYRDY interrupts are enabled, the Processor reads from that word, thus learning which key's position has changed.

To examine changes in any key's position, the Key Status Change Detector must compare each KDATA bit with the KDATA bit 128 CLOCK pulses earlier. (The KDATA bit 128 CLOCK pulses earlier represents the same key's position the previous time it was scanned.) In fact, to eliminate the effects of keyswitch bounce, the circuit compares the key's position with its position on several previous scans of the keyboard. To do this, it uses a type 5055 integrated circuit, which contains four 128-stage shift registers.

The KDATA signal goes to the D input of the 5055. After 128 CLOCK pulses, it emerges from the D output. Thus, the D output shows the "past history" of the key currently being scanned: its position on the previous scan of the keyboard. Provided KBDSYNCD is true, the D output is delayed three more 128-clock pulse periods in the A, B, and C sections of the 5055. Thus the 5055's D, A, B, and C outputs show the state of a key (the key currently being scanned) during the previous four times it was scanned.

Two four-input NAND gates monitor the shift register outputs. They detect when a key has held the same state for three consecutive scan periods (the D, A, and B outputs agree) but its state was different the scan period before that (the C output differs from the D, A, and B outputs). One NAND gate detects low-to-high transitions (keys being pressed); the other gate detects high-to-low transitions (keys beng released). By requiring that the key's state be the same on three consecutive scans of the keyboard, the gates reject false readings due to switch contact bounce.

The NAND gate outputs are fed to an OR gate, which provides a VALID signal when a valid change of keyswitch position has occurred. VALID sets the KEYRDY flip-flop. The KEYRDY signal, in turn, stops the Clock Counter, so that the keyswitch code is held static in the Keyswitch Counter. KEYRDY also interrupts the Processor, so that it may read the Keyboard Data Word.

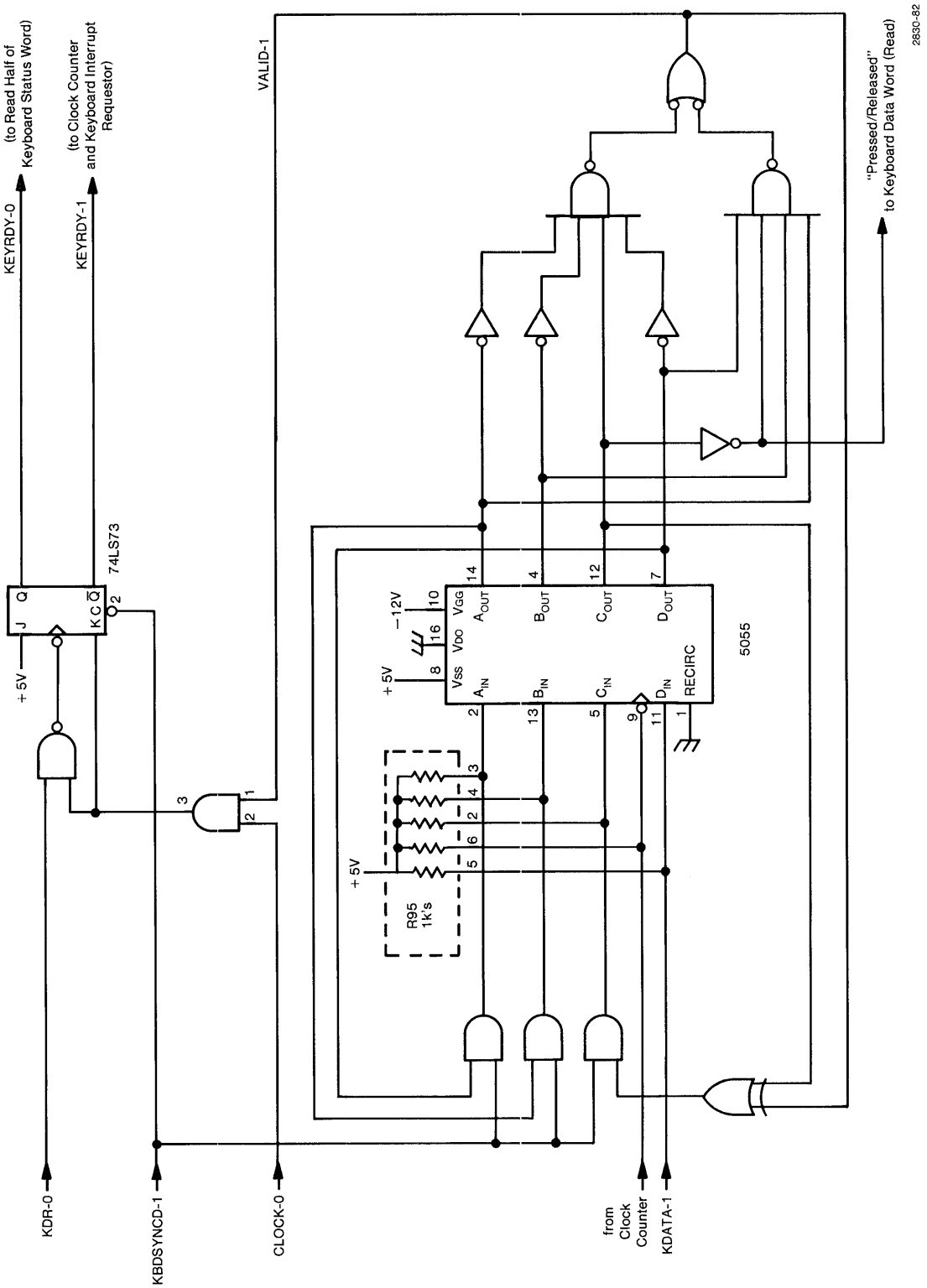


Figure 6-12. Key Status Change Detector.

Keyboard Data Word (Read)

(See Schematic 9-4.)

The "read" half of the Keyboard Data Word is a 74LS241 integrated circuit holding eight tri-state buffers. When the Processor (in response to a KEYRDY interrupt) reads from address X'0832', the KDR signal enables these buffers, placing the Keyboard Data Word on the data bus. The least significant seven bits of this word (from the Keyswitch Counter) show which key was pressed or released. The most significant bit (from the Key Status Change Detector) shows whether the key was pressed (Bit 7 = 1) or released (Bit 7 = 0).

Keyboard Data Word (Write)

(See Schematic 9-4.)

The "write" half of the Keyboard Data Word is a type 74LS273 latch. When the Processor writes into address X'0832', the KDW signal clocks this latch, storing the eight bits being presented on the data bus lines D0-D7. The latch outputs Q_1 to Q_8 drive the Shift Register's parallel inputs.

Shift Register

(See Schematic 9-4.)

The Shift Register converts the parallel data from the "write" half of the Keyboard Data Word into the serial LDATA signal for the keyboard. It is loaded by the LOAD signal from the SCLK Generator, and its data is shifted out by the SCLK (Shift Clock) signal.

SCLK Generator

(See Figure 6-13, Schematic 9-4.)

The SCLK Generator generates the SCLK and LOAD waveforms. LOAD is used only to load the Shift Register in the Keyboard Port. SCLK, on the other hand, serves both the Keyboard Port and the keyboard itself. SCLK has three functions:

- It clocks LDATA bits out of the Keyboard Port's Shift Register.
- It clocks those same bits into a shift register in the keyboard.
- It is used to load a latch in the keyboard, in which the LDATA bits are stored.

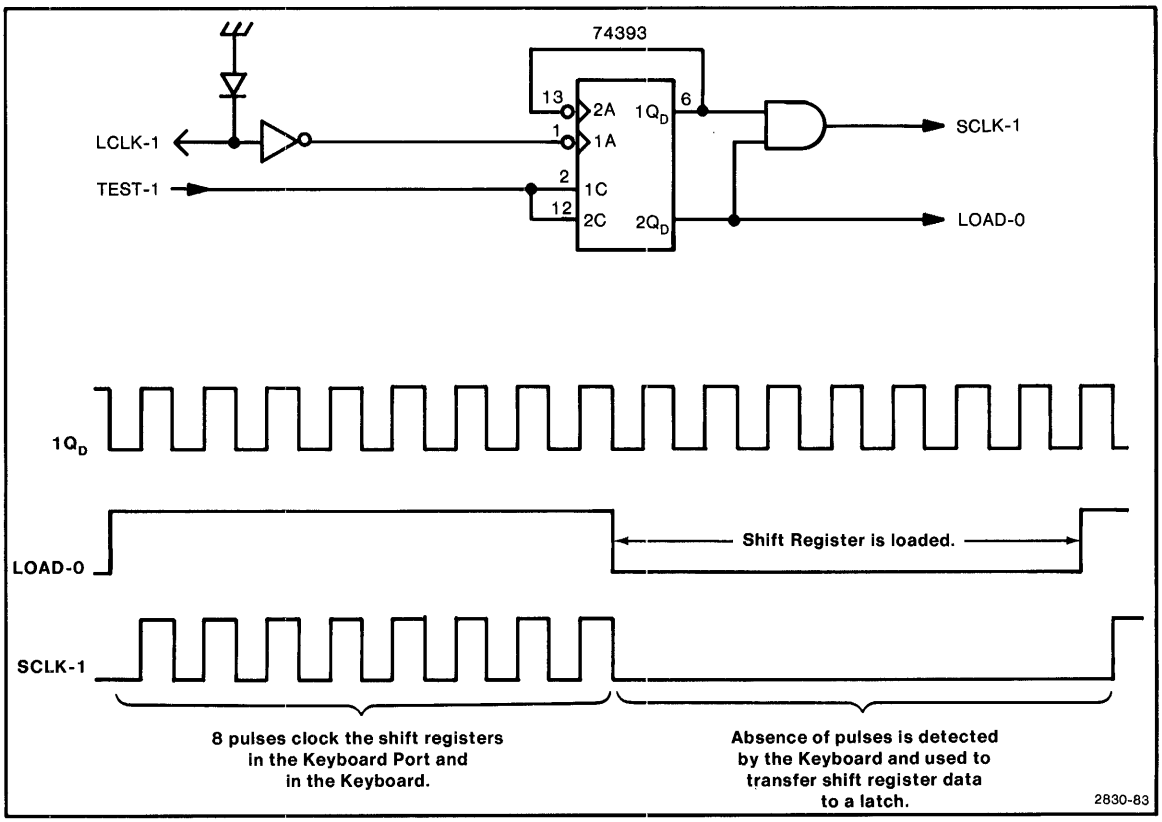


Figure 6-13. SCLK Generator and Waveforms.

Each group of eight SCLK pulses shifts the LDATA bits out of the Shift Register, sending them to the keyboard. The inverted SCLK waveform then clocks these same bits into another shift register within the keyboard.

After the eight LDATA bits have been shifted out, a LOAD pulse occurs, and the SCLK pulses are turned off. LOAD re-loads the Keyboard Port's Shift Register. The *absence* of LCLK pulses during this time is detected by the keyboard, which transfers the word just clocked into *its* shift register into a latch. (This is described in more detail in Section 4.)

The SCLK generator is comprised of a 74393 dual divide-by-16 counter and an AND gate. The first half of the counter divides the 2.048 MHz LCLK signal by 16, giving a 128 kHz square wave at its $1Q_D$ output. This is again divided by 16 in the second half of the counter, to provide the LOAD waveform at the $2Q_D$ output. The AND of these two waveforms is the SCLK waveform.

Section 7

4024
PROCESSOR/COMMUNICATIONS BOARD

INTRODUCTION

The 4024 Processor/Comm Board contains circuitry like that on the 4025 Processor and Communications Interface Boards. However, the 4024 circuitry is on one circuit board rather than two.

(See Figure 7-1.)

The Processor/Comm circuitry may be grouped under five major headings: Processor-Related Circuitry, I/O Address Decoders, Firmware ROMs, Host Port Circuitry, and Keyboard Port Circuitry. The Processor-Related Circuitry and the Firmware ROMs correspond to circuitry on the 4025 Processor Board, while the Host and Keyboard Port Circuitry correspond to circuits on the 4025 Communications Interface Board. The I/O Address Decoders are used by both parts of the Processor/Comm Board.

Before reading this section, you should read Sections 5 and 6, which describe the 4025 Processor and Data Communications Boards. This section describes the differences between the 4025 circuitry and the simpler circuitry on the 4024 Processor/Comm Board.

PROCESSOR-RELATED CIRCUITRY

(See Schematic 6-1.)

The Processor-Related Circuitry includes: Microprocessor, Clock and Reset Circuitry, Memory Address Buffers, Bus Address Buffers, Data and Control Buffer, and Memory Data Buffer. The circuitry closely resembles corresponding circuitry on the 4025 Processor Board. The differences are:

- The 4024 does not use a vectored interrupt handling scheme. Instead, all interrupts direct the Processor to address X'0000', where a ROM holding a "jump" instruction causes the Processor to branch to interrupt polling routines in the system firmware ROMs. Because the 4024 uses "polled interrupts" (rather than "vectored interrupts"), it lacks the following 4025 circuitry: Restart Address Modifier, Restart Instruction Register, Interrupt Address Counter, Interrupt Priority Register, and IRQ Sampler.

- The 4024 lacks provision for DMA operations; there is no DMA Control circuitry and no BUSEN signal.

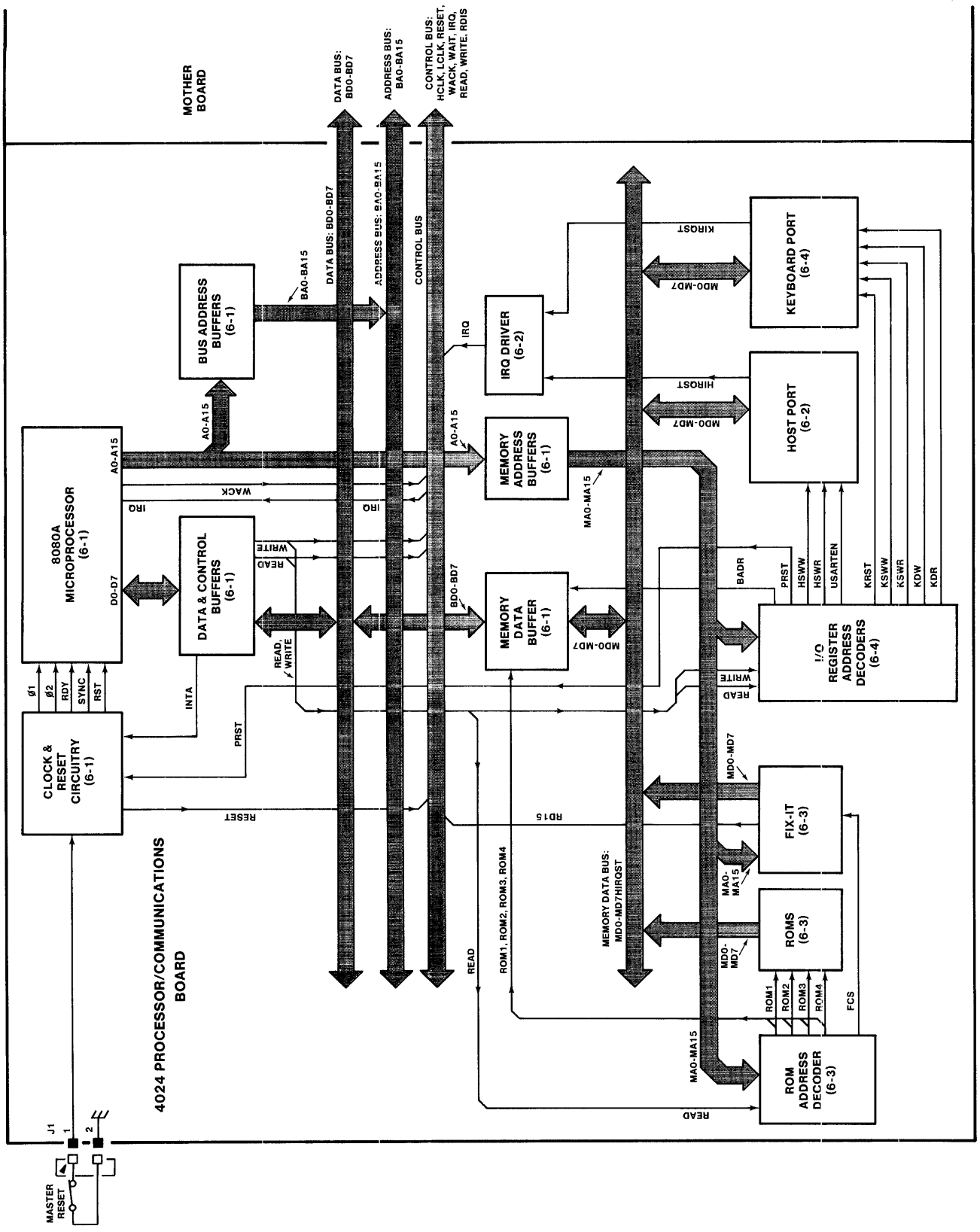


Figure 7-1. 4024 Processor/Comm. Board Block Diagram.

Microprocessor

The 8080 microprocessor is the same as that used in the 4025. See the section on the 4025 Processor Board for descriptions of the signals on its different pins.

Clock and Reset Circuitry

The Clock and Reset Circuitry (Schematic 6-1) resembles the Clock and Reset Circuitry on the 4025 Processor Board. It uses a type 8224 clock generator circuit and logic gates to generate these signals:

- Phase One and Phase Two clocks for the microprocessor.
- On-board (Phase Two TTL CLOCK) and off-board (LCLK) TTL versions of the Phase Two clock.
- On-board (OSC) and off-board (HCLK) versions of the 18.432 MHz high-frequency clock.
- The RST signal to reset the microprocessor on power-up.
- On-board (CLEAR) and off-board (RESET) signals to clear the terminal's circuitry.

Memory Address Buffers

The Memory Address Buffers (Schematic 6-1) buffer the microprocessor address lines A0-A15, driving an address bus internal to the Processor/Comm Board (lines MA0-MA15). These address buffers are always enabled.

Bus Address Buffers

(See Schematic 6-1.)

The Bus Address Buffers buffer the microprocessor address lines A0-A15, driving the main terminal address bus lines BA0-BA15 on the Mother Board. These buffers are always enabled.

Data and Control Buffer

(See Schematic 6-1.)

The Data and Control Buffer is a type 8228 system controller IC, designed for use with the 8080A microprocessor. It buffers the microprocessor data lines D0-D7, interfacing between them and the Mother Board data bus lines BD0-BD7. It also generates the WRITE, READ, and INTA (Interrupt Acknowledged) control signals.

Memory Data Buffers

(See Schematic 6-3.)

The Memory Data Buffers consists of a pair of 74LS243 bi-directional tri-state buffer ICs. During read operations from any of the ROMs, the Fix-It, or the I/O registers on this board, the buffers transfer data from the internal memory data bus (lines MD0-MD7) to the terminal's main data bus (lines BD0-BD7). During writes to I/O registers on the board, the buffers transfer from BD0-BD7 to the MD0-MD7 lines.

MEMORY MAP AND I/O ADDRESS DECODERS

Processor/Communications Board Memory Map

Figure 7-2 shows the Host Port memory map. Included on the 4024 Processor/Comm Board are:

- System Firmware ROMs 1, 2, 3, and 4. ROM 1 occupies addresses X'1000' to X'1FFF', ROM 2 occupies addresses X'2000' to X'2FFF', etc. These ROMs contain most of the 4024's firmware programs.
- ROM 0 at addresses X'0000' to X'00FF.' This ROM contains tables from which such things as parity, command character, etc. are set on power-up. It also contains (at address X'0000') a "jump" instruction directing the Processor to the interrupt polling routines.
- I/O registers:
    - Address X'0801': Processor-Initiated Reset. (Not used.)
    - Addresses X'0820'-X'0823': Host Port I/O Registers. These include the Host Port Status Word (X'0820'), the USART Data Word (X'0822'), and the USART Command/Status Word (X'0823').
    - Addresses X'0830'-X'0832': Keyboard Port I/O Registers. These include the Keyboard Status Word (X'0830'), the Keyboard Reset line (X'0831'), and the Keyboard Data Word (X'0832').

Tables 7-1 and 7-2 list the function of each bit in the Host Port and Keyboard Port I/O Registers.

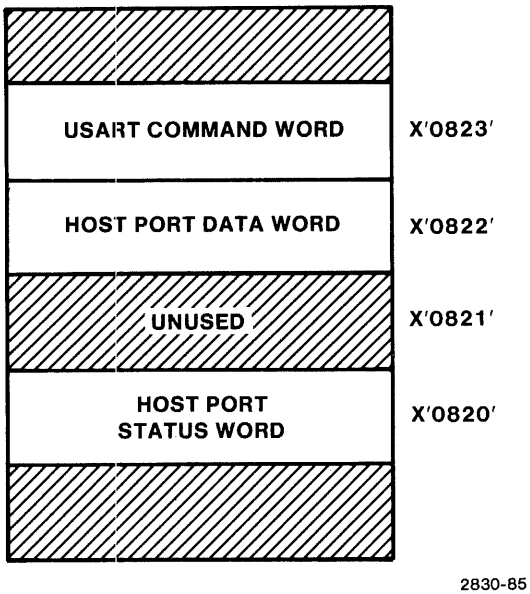


Figure 7-2. 4024 Host Port Memory Map.

Table 7-1

4024 HOST PORT I/O REGISTERS

Register Name	Address	Function	
Host Port Status Word	X'0820'	Read	Bit 0: Unused Bit 1: TXRDY (Transmitter Ready) Bit 2: Unused Bit 3: RXRDY (Receiver Ready) Bits 4-5: Unused Bit 6: TXE (Transmitter Empty) Bit 7: Interrupt Flag
		Write	Bit 0: TXRDY Interrupt Enable Bit 1: Unused Bit 2: RXRDY Interrupt Enable Bit 3: Unused Bit 4: TXE Interrupt Enable Bits 5-7: Unused
(Unused)	X'0821'		
USART Data Word	X'0822'	Read	Bits 0-7: Read Data
		Write	Bits 0-7: Write Data
USART Command and Status Words	X'0823'	Read	(USART Status Word) Bit 0: TXRDY (Transmitter Ready) Bit 1: RXRDY (Receiver Ready) Bit 2: TXE (Transmit Empty) Bit 3: PE (Parity Error) Bit 4: OE (Overflow Error) Bit 5: FE (Framing Error) Bit 6: Sync Detect Bit 7: Ring Indicator
		Write	(USART Programming and Command Words) (See the USART Circuit Description in the Section on the 4025 Communications Interface Board.)

Table 7-2

4024 KEYBOARD PORT I/O REGISTERS

Register Name	Address	Function	
Keyboard Port Status Word	X'0830'	Read	Bits 0-2: Unused Bit 3: INRDY (Input Ready) Bit 4-6: Unused Bit 7: KIRQST
		Write	Bits 0-1: Unused Bit 2: INRDY Interrupt Enable Bits 3-7: Unused
Keyboard Reset	X'0831'	Read	Bits 0-7: Unused
		Write	Bits 0-7: Unused; any write to this address resets the Keyboard Port.
Keyboard Data Word	X'0832'	Read	Bits 0-7: Read Data
		Write	Bits 0-7: Write Data
(Unused)	X'0833'		

I/O Address Decoders

The I/O Address Decoders (Schematic 6-4) provide signals to trigger read or write operations from the different I/O registers on the Processor/Comm Board. Inputs to the decoders are: IOADR, MA0-MA9, READ, and WRITE. The output signals are:

- **BADR (Board Addressed).** Signals an address in the part of memory address space reserved for I/O registers on the Processor/Comm Board. That is, signals an address in the ranges X'0800'-X'0803', X'0820'-X'0823', or X'0830'-X'0833'.
- **PRST (Processor-Initiated Reset).** Signals a read or write at address X'0801'.
- **HSWW (Host Port Status Word Write).** Signals a write to X'0820'.
- **HSWR (Host Port Status Word Read).** Signals a read from X'0820'.
- **USARTEN (USART Enable).** Signals read or write at X'0822' or X'0823'.
- **KSW (Keyboard Status Write).** Signals a write to X'0830'.

- **KSR (Keyboard Status Read).** Signals a read from X'0830'.
- **KRST (Keyboard Reset).** Signals a write to X'0831'.
- **KDW (Keyboard Data Write).** Signals a write to X'0832'.
- **KDR (Keyboard Data Read).** Signals a read from X'0832'.

ROM CIRCUITRY

(See Schematic 6-3.)

The Firmware ROM Circuitry includes these blocks: ROM Address Decoder, ROMs, and Fix-It.

ROM Address Decoder

The ROM Address Decoder decodes the internal memory address lines MA8-MA15, providing ROM0, ROM1, ROM2, ROM3, and ROM 4 enable signals. It also provides the FCS (Fix-It Chip Select) signal to enable the Fix-It FPROM, and the IOADR signal for the Mother Board. All these signals except ROM0 have counterparts on the 4025 Processor Board, and function identically to those counterparts.

ROMs

The ROMs hold the 4024's system firmware. As in the 4025, ROM 1 occupies addresses X'1000'-X'1FFF', ROM 2 occupies addresses X'2000'-X'2FFF', etc.

ROM 0 has no counterpart in the 4025. It occupies addresses X'0000'-X'00FF', the lowest 256 memory addresses. Whenever an interrupt occurs, the 4024 Processor automatically branches to address X'0000'. At X'0000' in ROM 0 there is a "jump" instruction which steers the Processor to the interrupt handling subroutines in system firmware ROM 1.

Fix-It

The Fix-It is similar to that on the 4025 Processor Board. An FPLA monitors address lines MA2-MA15 looking for "flagged addresses." When it detects such an address, its D1 output disables the ROMs, enables the Fix-It FPROM, and switches the FPROM address inputs from the address bus to FPLA outputs D2-D7. The result is that each flagged word of memory is replaced by a corresponding word in the Fix-It FPROM.

For more information on the Fix-It, refer to the Fix-It circuit description in the Section 5.

HOST PORT CIRCUITRY

(See Schematic 6-2.)

The Host Port Circuitry is much simpler than that in the 4025. It operates only in full duplex mode, so there are no Line Protocol or Line Turnaround circuits. Its baud rate is set by movable straps, so there is no Baud Rate Word.

The Host Port Circuitry consists of: USART, Baud Rate Generator, RS-232 Receiver and Drivers, Host Port Status Word (Write), Host Port Status Word (Read), Host Port Interrupt Requestor. The IRQ Driver, which is also used by the Keyboard Port, will be discussed here as well.

USART

The USART is the same type 8251A integrated circuit as is used in the 4025 Communications Interface Board. For a description of the USART registers, see Section 6.

In the 4024, the USART C/D pin is driven by MA0 (rather than MA1 as in the 4025). The result is that the USART Command and Status Registers occupy only one memory address, and the USART Data Registers likewise occupy only one memory location. This alters the Host Port's memory map slightly; see Figure 7-2.

The 4024 USART's RXD pin gets its data directly from the RS-232 Receiver on the RDATA line. The 4024 has no hardware for local echoing of transmitted data. (In local echo mode, the echo is provided by the firmware.)

Baud Rate Generator

The Baud Rate Generator borrows from the Keyboard Port Clock a signal at 1/8 of the 18.432 MHz oscillator frequency. A 74LS161 counter divides this signal by 15 to provide a signal at 153.6 kHz. Since 15.36 kHz is 16 times 9600 baud, this signal is a 16X clock for the USART (when operating at 9600 baud). This 9600 baud 16X clock is presented to Pin 1 of J4.

A 74393 counter further divides the 9600 baud clock to provide 16X clocks for operation at 4800, 2400, 1200, 600, 300, 150, and 75 baud; these clocks are presented to Pins 2 through 8 of J4.

To provide a 110 baud clock, a 74LS161 counter divides the 1200 baud clock by 11. The result is within 1% of the exact frequency. A strap at connector J7 directs either the 110 baud clock or the 75 baud clock to Pin 8 of J4.

Pins 1 and 2 of J5 connect to the USART's RXC pin, and J5 Pins 3 and 4 connect to the TXC pin. Jumpers from the appropriate pins of J4 to J5 set the transmit and receive baud rates.

RS-232 Receiver and Drivers

The RS-232 Receiver and the RS-232 Drivers use transistors to interface between the RS-232 signal levels (plus or minus 12 volts) and the TTL signal levels (about + 5 V and ground). The RS-232 Receiver takes the RDATA signal and converts it to a TTL RXD signal for the USART. Likewise, the RS-232 Drivers take the TTL TXD signal and convert it to the RS-232 TDATA signal. Also, pullup resistors in the RS-232 Driver circuit drive the RTS (Request to Send) and DTR (Data Terminal Ready) signal true whenever the terminal power is on.

Host Port Status Word (Write)

As in the 4025, the write half of the Host Port Status Word consists of type D flip-flops in a 74LS175 integrated circuit. When the Processor writes into the Host Port Status Word (address X'0820'), the HSWW (Host Status Word Write) signal clocks these flip-flops, storing the bits on data lines MD0, MD2, and MD4. These are the interrupt enable bits for the TXRDY, RXRDY, and TXE (Transmitter Empty) interrupts.

Host Port Interrupt Requestor

As in the 4025, the read half of the Host Port Status Word consists of tri-state buffers in a 74LS367 integrated circuit. When the Processor reads from address X'0820', the HSWW (Host Status Word Write) signal enables these buffers. The buffers then place the states of the TXRDY, RXRDY, TXE, and HIRQST lines on data bus lines MD1, MD3, MD6, and MD7.

IRQ Driver

The IRQ driver is a single gate which drives the Mother Board IRQ (Interrupt Request) line low whenever a HIRQST (Host Port Interrupt Request) or KIRQST (Keyboard Port Interrupt Request) occurs. IRQ is driven low as soon as HIRQST or KIRQST occurs.

Host Port Status Word (Read)

As in the 4025, the read half of the Host Port Status Word consists of tri-state buffers in a 74LS367 integrated circuit. When the Processor reads from address X'0820', the HSWW (Host Status Word Write) signal enables these buffers. The buffers then place the states of the TXRDY, RXRDY, TXE, and HIRQST lines on data bus lines MD1, MD3, MD6, and MD7.

KEYBOARD PORT CIRCUITRY

(See Schematic 6-4.)

The 4024 Keyboard Port circuitry is like that on the 4025 Communications Interface Board. The circuit blocks are: Keyboard Port Clock (on Schematic 6-2), Keyswitch Counter, Sync Generator, Key Status Change Detector, Keyboard Data Word (Read), Keyboard Data Word (Write), Shift Register, and SCLK Generator.

These circuits all perform identically to the corresponding circuits in the 4025 Communications Interface Board. See Section 6 for circuit descriptions.

Section 8

DISPLAY MEMORY BOARD

This section describes the Display Memory Board, used in both the 4024 and the 4025.

DISPLAY MEMORY BOARD CONCEPTS

Before examining the Display Memory Board in detail, let's look at the functions it performs.

Display Memory

The display memory occupies the upper half of the 8080's memory address space. (If 4K of display memory is installed, it occupies addresses X'F000' X'FFFF'. With the maximum 32K of memory installed, it occupies X'8000' to X'FFFF'.)

The display memory serves as a general-purpose "pool" of read/write memory. For instance, it is in display memory that the Processor stores input/output queues and the definitions of the programmed function keys. However, the most important thing stored in display memory is the display list.

Display List

The display list is a sequence of instructions controlling what is sent to the Video Display Unit. (The Processor controls the display by adding to this list or deleting instructions from it.) Included in the display list are:

- All text in the workspace and monitor scrolls of memory.
- Field attribute codes. These include visual attribute codes, character font attribute codes, and logical attribute codes.
- End-of-line markers.
- Cursor position data.
- Commands for the Tracker.

Tracker

The "Tracker" circuitry reads the display list and sends the Display Controller the information it needs.

As text is inserted (with the INSERT MODE or INSERT LINE keys), the Processor links the new text into the display list by inserting "jump" instructions. These direct the Tracker to the memory locations where the new text is stored. At the end of the new text, additional JUMPs direct the Processor back to the older parts of the display list. Thus, as the display list grows, it wanders around in display memory.

The Tracker reads jump instructions, end-of-line markers, visual field attribute codes, etc., from the display list and operates on them as commands. In a sense, the Tracker is a specialized computer which operates on the display list as data.

As the Tracker reads the display list, it composes a line of text for the Display Controller. This includes not only the characters in that line of text but also the visual field attribute codes and character font information to indicate how the line is to be displayed. It passes this information to the Display Controller on the "paired bus" lines (PB0-PB11).

While the Display Controller scans one row of text, the Tracker is sending it the next row.

Input and Output

To the Processor, display memory acts like any other part of the 8080's address space. To access a word in display memory, the Processor places the word's address on the terminal's main address bus (BA0-BA15), and the data is transferred on the main data bus (BD0-BD7).

Unknown to the Processor, the Tracker is also accessing the display memory and causing display information to be sent to the Display Controller Board. The Tracker accesses display memory through lines internal to the Display Memory Board, without affecting the terminal's main address or data busses. Display information passes to the Display Controller on the "paired bus" lines of the Mother Board (PB0 to PB11).

CIRCUIT DESCRIPTIONS

Refer to Figure 8-1. The Display Memory Board includes these major blocks:

- Display RAM
- RAM Controller
- RAM Address Steering
- Tracker
- Bus Interface

Display RAM

The display memory is contained in RAM (Random Access Memory) and consists of either eight or sixteen devices, each holding either 4K or 16K bits, as follows:

- 4K of memory: 8 4Kx1 devices to provide 4096 8-bit words.
- 8K (option 20): 16 4Kx1 devices, 8192 words.
- 16K (Option 21): 8 16Kx1 devices to provide 16,384 words.
- 32K (Option 22): 16 16Kx1 devices, 32,768 words.

These RAMs appear in Schematic 5-3; Figure 8-2 shows their pinouts. The 4K RAMs have six address lines (A0-A5) which alternate between carrying one half of the twelve-bit address and carrying the other half of that address. Likewise, the 16K RAMs have seven address lines which alternate between two halves of a 14-bit address. The *row address* is the first half of the address placed on a RAM's address pins; the *column address* is the second half to be placed on those pins.

Figure 8-3 shows the RAM read, write, and refresh timing.

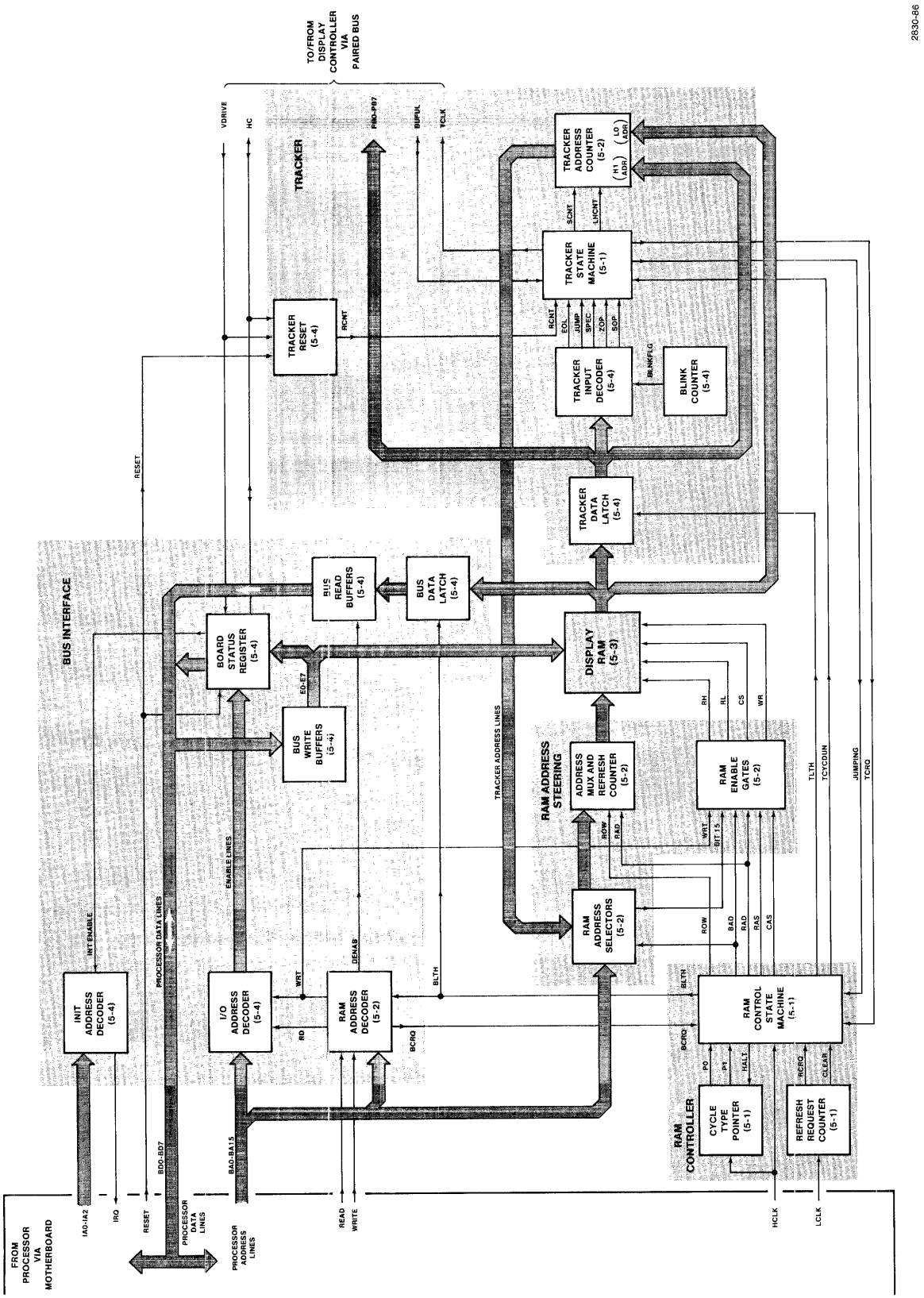


Figure 8-1. Display Memory Board Block Diagram.

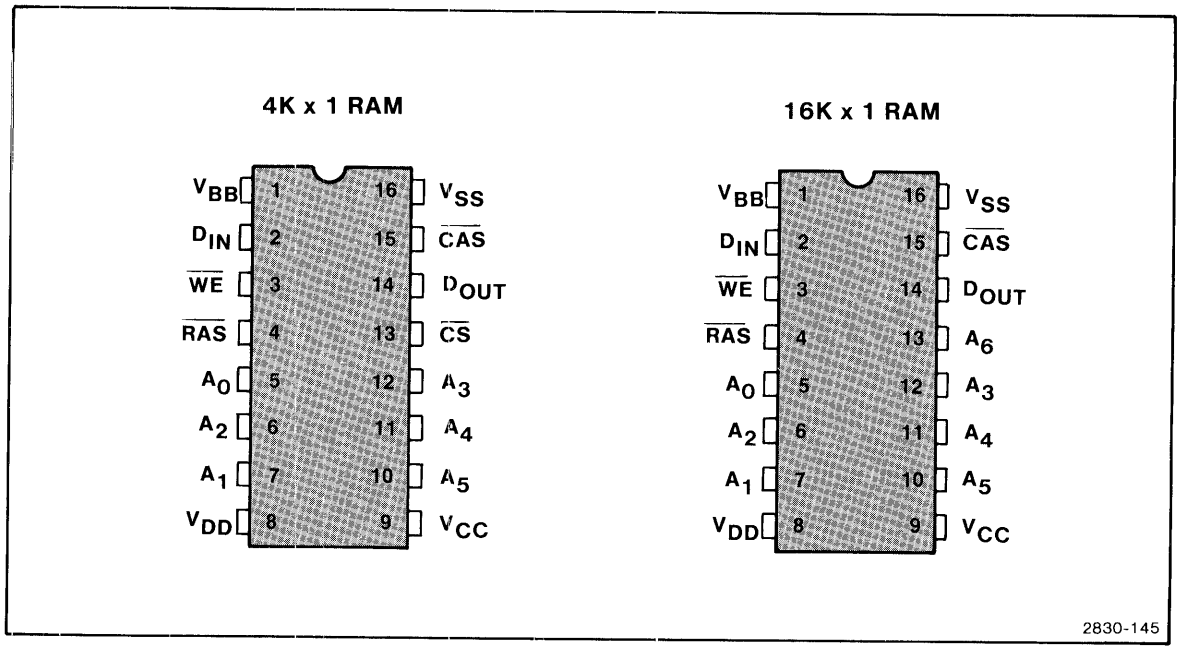


Figure 8-2. Pinouts for 4K and 16K RAMs.

RAM Controller

(See Figure 8-1, Schematic 5-1.)

The RAM Controller provides signals to control the RAMs during read, write, or refresh cycles. It also provides periodic read cycles to refresh the dynamic RAMs. During a read or write cycle, it provides the correct RAS (Row Address Strobe) and CAS (Column Address Strobe) signals to control the multiplexing of the RAM address inputs.

The RAM Controller inputs are:

- **HCLK.** The 18.432 MHz clock.
- **LCLK.** The 2.0428 MHz clock.
- **BCRQ (Bus Cycle Request).** A signal that the Processor requests a memory read or write operation.
- **TCRQ (Tracker Cycle Request).** A signal that the Tracker requests a read operation.
- **JUMPING.** A Tracker signal indicating that the byte being read is the second byte of a jump command and should be loaded directly into the Tracker Address Counter rather than into the Tracker Data Latch.

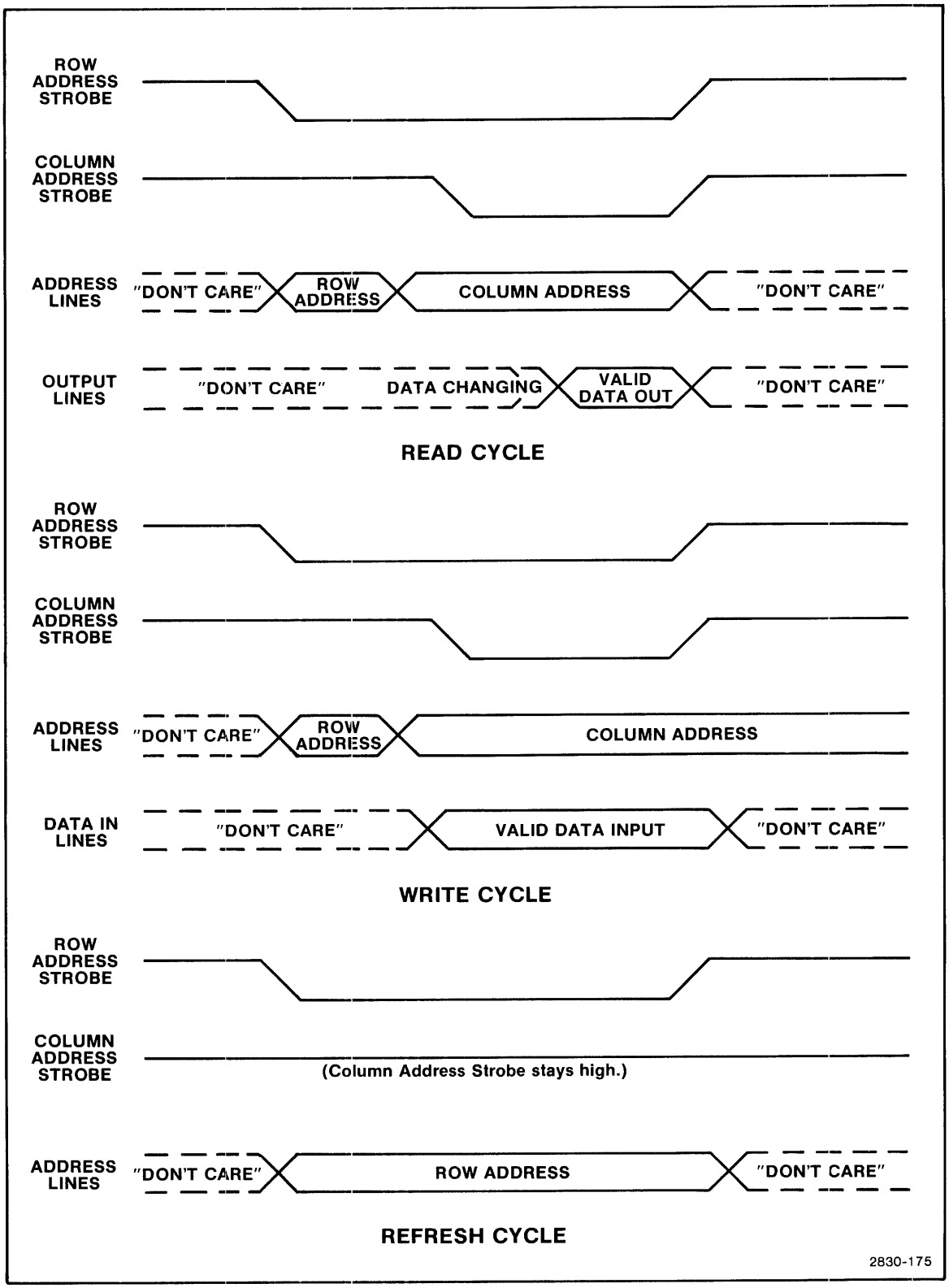


Figure 8-3. RAM Waveforms.

The RAM Controller outputs are:

- **RAS (Row Address Strobe).** A signal asserted whenever a RAM is to read its address inputs.

- **CAS (Column Address Strobe).** A signal asserted whenever the second half of the RAM address (the column address) has settled, and the RAM is to read that column address from its address inputs.

- **RAD (Refresh Addressed).** A signal that the Ram Controller's Cycle Request Pointer is pointing to "refresh cycle." During a refresh cycle, this signal causes the RAM address to be taken from a "refresh address counter" rather than from the Tracker or the terminal address bus.

- **BAD (Bus Addressed).** A signal indicating that the Cycle Request Pointer is pointing to a "bus cycle." This causes the RAM address to be taken from the terminal's main address bus.

The RAM Controller is divided into three parts: the Cycle Type Pointer, the RAM Control State Machine, and the Refresh Request Counter. Each of these is discussed separately.

Cycle Type Pointer

(See Schematic 5-1.)

The Cycle Type Pointer consists of the flip-flops labeled P0 and P1 in Schematic 5-1. These comprise a two-bit binary counter which repeatedly counts from 0 to 3. The counter "points" to the type of RAM cycle which may occur at any instant. On the counts of 0 and 2, the pointer designates a *bus cycle*—a RAM cycle requested by the Processor. On the count of 1, it points to a *Tracker Cycle*—a RAM cycle requested by the Tracker. On the count of 3 it designates a *Refresh Cycle*—a RAM read cycle requested by the Refresh Request Counter.

The P0 and P1 flip-flops control data selector U115, which selects whether the RAM Control State Machine takes its cycle request inputs from Refresh Request Counter U311, from the BCRQ (Bus Cycle Request) line, or from the TCRQ (Tracker Cycle Request) line.

The P0 and P1 flip-flop outputs are also decoded to generate the RAD (Refresh Cycle Addressed) and BAD (Bus Cycle Addressed) signals.

RAM Control State Machine

(See Schematic 5-1.)

The RAM Control State Machine consists of the flip-flops labeled Q0 to Q4, and their associated gates. The flip-flop states make a code designating which state the State Machine occupies:

Q4	Q3	Q2	Q1	Q0	State
0	0	0	0	0	A
0	0	0	0	1	B
0	0	0	1	1	C
0	0	1	1	1	D
0	1	1	1	1	E
1	1	1	1	1	F
1	1	1	1	0	G
1	1	1	0	0	H
1	1	0	0	0	I
1	0	0	0	0	J

As the State Machine moves through these states, it provides waveforms to drive the RAMs in a read or write cycle. The states are as follows:

State A: Advance the Cycle Type Pointer. The next state is B.

State B: Ask, "Is the pointed-to device requesting a RAM cycle?" If "Yes," the next state is C. If "No," the next state is A.

(If no device requests a RAM cycle, the State Machine alternates between States A and B. When a device requests a cycle, the State Machine waits until the Pointer designates the requesting device, and then initiates the RAM cycle by entering State C.)

State C: Assert RAS (Row Address Strobe) and place the row address on the RAM address bus. (The row address is the first half of the RAM address to be placed on the RAM address bus.) The next state is D.

State D: Continue as in State C. The next state is E.

State E: Continue asserting RAS. On the RAM address bus, replace the row address with the column address. The next state is F.

State F: Continue asserting RAS and placing the column address on the RAM address bus. By now the column address has settled; assert CAS (Column Address Strobe). The next state is G.

State G: Continue sending RAS and CAS and holding the column address on the RAM address bus. The next state is H.

State H: Continue as in States F and G. The next state is I.

State I: Continue asserting RAS and CAS. By now the RAM output data should have settled; latch that data. If this is a "Bus Cycle," latch the data in the Bus Latch for later transfer to the Processor over the main data bus. (Send the BLTH signal.) If this is a "Tracker Cycle," put the data in the Tracker Latch (send the TLTH signal) unless the Tracker is on the second half of a "jump" (JUMPING true); if this is the case, put the data in the Tracker Address Counter (send the LCNT signal).

The next state is J.

State J: Turn off any signals you are sending. This concludes the RAM cycle.

The next state is A.

The gates scattered throughout the RAM Control State Machine decode its states, providing the output signals and the flip-flop inputs to advance from one state to the next.

Refresh Request Counter

(See Schematic 5-1.)

The Refresh Request Counter in the RAM Contoller is driven by LCLK, the terminal's 2.048 MHz clock. It periodically provides, through jumper SB, a signal to data selector U115 that it's time for another RAM refresh cycle. Setting SB to position 16 allows 16K RAMs, if installed, to be refreshed more frequently than 4K RAMs would be.

RAM Address Steering

The RAM Address Steering circuitry interfaces the RAM address lines with the sources of addresses to be placed on those lines. Such an interface is needed for two reasons:

- There are three possible sources for an address to be placed on the RAM address bus: the main terminal bus (used by the Processor), the Tracker Address Counter (used by the Tracker), and the refresh address counter (to be discussed later). There must be a way to switch the RAM address bus between these three sources of address information.

- As mentioned before, the RAM addresses are multiplexed; only seven RAM address lines are used to carry 14 address bits. There must be a way to switch the RAM address lines between the seven "row address" bits and the seven "column address" bits.

The RAM Address Steering circuitry (Figure 8-1, Schematic 5-2) consists of the following blocks: RAM Address Selectors, Address Multiplexer & Refresh Counter, and RAM Enable Gates.

RAM Address Selectors

Several "data selector" devices (U471, U481, U375, U365, U475) select whether a RAM address comes from the Tracker Address Counter or the main terminal bus. These data selectors are steered by the BAD (Bus Cycle Addressed) signal from the RAM Controller. They pass 14 address bits on to U485.

Address Multiplexer and Refresh Address Counter

U485, an Intel 3242 (or equivalent), is a combination RAM address multiplexer and refresh address counter. When the RAD (Refresh Cycle Addressed) signal is false, U485 places either the row address bits or the column address bits on the RAM address bus, depending on the state of the ROW signal.

When RAD and ROW are both asserted, U485 places the row address of the row to be refreshed on the RAM address bus. The row address is provided by a Refresh Address Counter (within U485), which is advanced on each refresh cycle.

RAM Enable Gates

Several gates associated with the RAM Address Steering circuitry provide signals to enable the two banks of RAMs:

- CS drives the RAM column address strobe pins whenever CAS is asserted and it is not a refresh cycle (RAD not true).
- RL (RAS for the "low" bank of RAMs) drives the row address strobe pins of the "low" bank of RAMs whenever the RAM Controller is addressing one of those RAMs.
- RH serves a similar function for the "high" bank of RAMs.
- WR (Write) drives the RAM write enable pins during a write cycle initiated by the Processor (a "bus write cycle").

Tracker

(See Figure 8-1; also see Schematics 5-1, 5-2, and 5-4.)

The Tracker is the circuitry that tracks the display list through memory. The Tracker operates on the "jumps," "no ops," and "end-of-line markers." It passes the remaining data (characters and field attribute codes) to the Display Controller.

Display List Commands

The commands on which the Tracker operates are bytes in the display list. Figure 8-4 shows their formats.

Characters. The most common items in the display list are the characters of text. Each character specifies one of 128 dot patterns for the character font currently being displayed. (The default font is font zero, for which the dot patterns are representations of ASCII characters.)

Field Attribute Codes. There are three types of field attribute codes: visual attribute codes, character font attribute codes, and logical attribute codes. The Tracker passes the visual and character font attribute codes on to the Display Controller text.

Logical attribute codes are inserted in the display list only for the Processor's benefit; they do not affect how text is displayed. Because of this, the Processor inserts, before each logical attribute code, a command ("super no op") for the Tracker to skip over that code.

Characters

msb | 0 | C | C | C | C | C | C | C | lsb

C = Character (0 to 127)

Spare

| 1 | 0 | 0 | X | X | X | X | X |

X = Reserved

Tracker Control

| 1 | 0 | 1 | 0 | 0 | F | F | F |

F = Function

0 - EOL (End of Line)
1 - ZOP (Zero No-op)
2 - SOP (Super No-op)
3 - MOP (Maybe No-op)
4 - Not Used
5 - Not Used
6 - NOP
7 - Not Used

Visual Field Attribute

| 1 | 0 | 1 | 0 | 1 | U | E | I |

U = Underline (0 - Off, 1 - On)
E = Background (0 - Normal, 1 - Enhanced)
I = Video (0 - Normal, 1 - Inverted)

Character Set

| 1 | 1 | 0 | C | C | C | C | C |

C = Character Set 0 to 31

Jump

1	1	1	A	A	A	A	A
B	B	B	B	B	B	B	B

A,B = Jump Address - the tracker will jump to the memory address:

1	A	A	A	A	A	B	B	B	B	B	B	B	B	0	0

Notes: In addition to the above instructions, the first character code after EOL will be interpreted as the location of the cursor on the line being built. Values of 0 to 79 will put the cursor in columns 0 to 79. Values of 80 to 127 will put NO cursor on that line.

An EOL will set Intensity to NORMAL, Underline OFF, and Character Set to 0 (standard ASCII set).

2830-87

Figure 8-4. Display List Instruction Formats.

Tracker Control Commands. The Tracker Control Commands are: EOL, NOP, ZOP, SOP, MOP, and JUMP.

EOL (End of Line): An EOL marks the end of a row of text. When the Tracker encounters an EOL, it briefly pulls the BUFUL (Buffer Full) line low, signalling the Display Controller to reset the character font and visual attributes to character font zero, "standard" visual attribute. The Tracker then repeatedly sends ASCII spaces (character number 32) to the Display Controller. This continues until the Display reaches the end of its Row Buffer. The Display Controller then pulls BUFUL low and holds it low. This causes the Tracker to wait until BUFUL goes high again, whereupon the Tracker starts assembling the next row of text.

NOP (No Op): Upon reading a NOP, the Tracker goes on to the next byte (word of memory) without doing any other operation.

ZOP (Zero No Op): When the Tracker reads a ZOP, it reads the same word over and over, until it finds there a command other than ZOP. ZOP is used during modification of JUMP commands, to keep the Tracker from reading a partially modified JUMP, because the JUMP is two bytes long. In order to modify a JUMP, the Processor first changes the first byte to a ZOP. It then changes the second byte to the second byte of the new JUMP command; finally, it changes the ZOP to the first byte of the new JUMP command.

SOP (Super No Op): A SOP tells the Tracker to ignore (skip over) both the SOP and the next word in the display list.

MOP (Maybe No Op): The MOP instruction is used to blank visual attribute fields. There is a flag (BLNKFLG) which changes levels every quarter second. When the flag is high, Tracker decodes a MOP as a SOP: it skips over both the MOP and the following word of the display list. When the flag is low, Tracker decodes a MOP as NOP: it skips over only the MOP.

AOP (Anti No Op): This is the opposite of MOP; it functions like a NOP when BLNKFLAG is high, and like a SOP when BLNKFLG is low.

JUMP: This is a two-byte command which causes the Tracker to jump to another location in the display list.

The Tracker consists of the Tracker Data Latch, Tracker Input Decoder, Blink Counter, Tracker State Machine, Tracker Reset Circuitry, and Tracker Address Counter. Each is discussed in turn.

Tracker Latch and Input Decoder

(See Schematic 5-4.)

Data read from RAM during a Tracker RAM cycle is latched in U351 (the Tracker Latch) and becomes the display list byte on which the Tracker operates. The Tracker Input Decoding gates examine this word and provide the Tracker State Machine with the following signals:

- **EOL:** The word just read from the display list is an end-of-line marker.
- **ZOP:** The word just read from the display list is a "zero no op" command.
- **SOP:** The word just read from the display list is a "super no op" command.
- **SPEC:** The word just read from the display list is a "special Tracker command": EOL (End-of-Line), ZOP (Zero No Op), SOP (Super No Op), MOP (Maybe No Op), NOP (No Op), or the first half of a JUMP.
- **JUMP:** The word just read from the display list is the first half of a two-byte JUMP command.

Blink Counter

The Blink Counter (U411, Schematic 5-4) is clocked by VDRIVE pulses from the Display Controller, except during hard copy operations. On every 128 VDRIVE pulses, its BLNKFLG (Blink Flag) output toggles an exclusive-OR gate in the Tracker Input Decoding circuitry.

As BLNKFLG toggles, the Tracker Input Decoding alternates its interpretation of MOP (maybe no op) and AOP (anti no op) instructions. MOP is used for "blinking" between two sets of visual attributes.

Tracker State Machine

(See Schematic 5-1.)

The heart of the Tracker is the Tracker State Machine. This is a specialized processor which acts on the signals from the Tracker Input Decoders and the RAM Controller, and provides signals to:

- Request Tracker cycles from the RAM Controller (the TCRQ signal).

- Inform the RAM controller when the word to be read is the second half of a Tracker JUMP command (JUMPING).

- Load the high order bits of the Tracker Address Counter (send the LHCNT signal). (The low-order bits are loaded by the RAM Controller's LCNT signal.)

- Inform the Display Controller when to clock another word into its input buffer (TCLK) and when that buffer is full (BUFUL). (The BUFUL signal makes the Display Controller reset the current visual field attribute to "standard" and the character font to "font 0."

Latch U21 and ROM U15 are the heart of the state machine. LCLK, the terminal low frequency clock, clocks the latch to advance the state machine from one state to the next. The latch outputs provide address inputs for the 32 x 8 ROM. Three of the ROM inputs (U15 Pins 10, 11, and 12) give the "present state address." The other two ROM inputs (Pins 13, 14) are the data inputs for that state. Three of the ROM output bits (Pins 1, 2, 3) specify the next state. Two output bits (Pins 7, 8) control data selector U111, selecting which two signals will be examined as inputs in the next state. The remaining three ROM output bits (Pins 4, 5, 6) determine which signals the State Machine sends to the RAM Controller, Display Controller, etc.

ROM address input B (Pin 11) serves a special purpose: it is high when the next Tracker RAM cycle will be reading the second byte of a jump command. This is passed as a signal (JUMPING) to the RAM Controller so that the RAM controller loads the byte into the Tracker Address Counter, rather than into the Tracker Data Latch.

Demultiplexer U315 decodes the ROM output bits (ROM outputs Y6, Y5, Y4) and, together with some associated gates and flip-flops, generates the Tracker output signals:

- When ROM outputs Y6, Y5, Y4 are 0, 0, 0, the demultiplexer clears the TCRQ flip-flop, U215A. The flip-flop then sends a TCRQ (Tracker Cycle Request) signal to the RAM Controller. (The flip-flop is reset by a TCYCDUN, Tracker Cycle Done, signal from the RAM Controller.)

- When ROM outputs Y6, Y5, Y4 are 0, 0, 1, an SCNT signal is sent (to advance the Tracker Address Counter), provided the byte in the Tracker Data Latch is not a ZOP command.

- When ROM outputs Y6, Y5, Y4 are 0, 1, 0, the Tracker Address counter is advanced if the byte in the Tracker Data Latch is a SOP command.

- When ROM outputs Y6, Y5, Y4 are 0, 1, 1, the Tracker Address counter is advanced, unconditionally. Also, the TCLK (Tracker Clock) signal is sent to the Display Controller; the Display Controller then accepts the byte waiting for it on the paired bus lines (outputs of the Tracker Latch).

- When ROM outputs Y6, Y5, Y4 are 1, 0, 0, the BUFUL signal is sent to the Display Controller.

- When ROM outputs Y6, Y5, Y4 are 1, 1, 0, the Tracker State Machine does nothing; it sends no signals, but just advances to its next state.

- When ROM outputs Y6, Y5, Y4 are 1, 1, 1, the Tracker sends a TCLK (Tracker Clock) signal to the Display Controller. (The Display Controller then seizes the byte presented it on line PB0-PB7, the outputs of the Tracker Latch.)

Tracker Reset Circuitry

(See Schematic 5-4.)

The Tracker Reset Circuitry provides the RCNT (Reset Counter) signal, which resets the Tracker to start again at the first line of text to be displayed. RCNT loads X'F000' into the Tracker Address Counter. (This address holds a JUMP directing the Tracker to the start of the first line of text to be displayed.) RCNT also clears the state address latch in the Tracker State Machine; this re-initializes the state machine to a known starting state.

The Tracker Reset Circuitry has three inputs, corresponding to the three events which can make the Tracker start over at the top of the screenful of text:

- The Mother Board RESET line provides a signal on power-up or when MASTER RESET is pressed.

- The paired bus VDRIVE line provides a signal at each vertical retrace.

- The paired bus HC line provides a signal when a hard copy is in progress.

A three-input negative-logic OR gate (V131C) detects when any of these three conditions occurs. This gate's output is ANDed with the CLK signal to provide the RCNT signal.

In the 4024/4025, the HCU (Hard Copy Unit) strap is set to NORM rather than VID. With the strap in this setting, VDRIVE pulses are passed on to the three-input OR gate only when there is no hard copy in progress.

When a hard copy occurs, an AND gate (U241D) prevents VDRIVE signals from reaching the OR gate to generate RCNT pulses. Also, two type D flip-flops provide a brief pulse, which does reach the OR gate, generating a single RCNT pulse.

Tracker Address Counter

(See Schematic 5-2.)

The Tracker Address Counter holds the address of the next byte which the Tracker will read from the display list. The Tracker advances this counter to go from one byte in the display list to the next.

When the Tracker encounters the first byte of a jump command, it leaves that byte stored in the Tracker Latch. It then requests another RAM cycle, sending the JUMPING signal as it does so. JUMPING tells the RAM Controller to take the byte it reads from RAM and load it into the low-order bits of the Tracker Counter. (Usually, a byte read during a Tracker Cycle is loaded into the Tracker Latch; JUMPING causes it to go into the Tracker Address Counter instead.)

At the end of the Tracker Cycle, TCYCDUN goes high. This clocks the TCRQ flip-flop, turning off the TCRQ signal. Also, if JUMPING is true, the positive edge of TCYCDUN clocks flip-flop U215B in the Tracker. This flip-flop then sends the LHCNT signal, which loads bits stored in the Tracker Latch into the high-order bits of the Tracker Address Counter.

Bus Interface Circuitry

(See Figure 8-1, Schematics 5-2 and 5-4.)

The Bus Interface circuitry interfaces the Display Controller Board to the the main terminal bus. It includes: RAM Address Decoder, Bus Data Latch, Bus Read Buffers, Bus Write Buffers, I/O Address Decoder, Status Register, and Interrupt Request Circuitry.

RAM Address Decoder

(See Schematic 5-2.)

The RAM Address Decoder decodes the high-order bits of the terminal data bus to see whether display memory is being addressed. If it is, and either WRITE or READ is asserted, it sets the Bus Cycle Request and Wait flip-flops to perform a "handshake" function.

As these flip-flops are set, a BCRQ (Bus Cycle Request) signal goes to the RAM Controller, and a WAIT signal goes to the Processor. When the requested RAM cycle is complete, BLTH (which latches the data into the Bus Data Latch) resets the flip-flops.

Bus Write Buffers

(See Schematic 5-4.)

The Bus Write Buffers are inverters which buffer the Mother Board data lines BD0-BD7, driving the on-board data lines E0-E7.

Bus Data Latch and Bus Read Buffers

(See Schematic 5-4.)

During a Processor-initiated RAM cycle (bus cycle), the data read from the RAMs is latched into the Bus Data Latch by the BLTH signal from the RAM Controller. The DENAB (Data Enable) signal (from the Bus Interface circuitry on Schematic 5-2) causes the Read Buffer to place this data on the main terminal bus. This DENAB signal occurs when (a) the Display Memory Board has been addressed, as detected by the Board Address Decoder, and, moreover, (b) the Processor has sent a READ signal on the main terminal bus.

I/O Address Decoder

(See Schematic 5-4.)

A network of gates detects reads or writes to addresses X'0810' or X'0811'. (The gates do not decode all the address lines: they treat any even address in the range X'0810' to X'081F' the same as address X'0810'; likewise, any odd address in that range is regarded as address X'0811'.) Address X'0810' holds the Status Register, while X'0811' is used for clearing VDRIVE interrupt requests.

Table 8-1 lists the Display Memory Board I/O Register functions.

Table 8-1

DISPLAY MEMORY BOARD I/O REGISTERS

Address	Register	Functions		
X'0810'	Status Register	Write:	Bit 0:	INTEN (INTEN= 1: enable interrupts)
			Bit 1:	Unused
			Bit 2:	HC (To start a hard copy, set HC to 1.)
			Bits 3-7:	Unused
		Read:	Bit 0:	INTEN
			Bit 1:	VDRIVE detected
			Bit 2:	HC (HC= 1: hard copy in progress)
			Bits 3-6:	Unused
			Bit 7:	VDRIVE interrupt being requested
X'0811'	VDRIVE Detector Reset	Write:	Writing into X'0811' clears the VDRIVE Detect Flip-Flop.	
		Read:	(The "read" half of address X'0811' is unused.)	

Status Register

(See Schematic 5-4.)

The Status Register occupies address X'0810' (and all other even addresses between X'0810' and X'081F'). By writing into this register, the Processor can enable interrupts from the Display Memory Board or initiate a hard copy. By reading from this register, the Processor can learn:

- Whether "VDRIVE detected" interrupts are enabled.
- Whether a VDRIVE pulse has occurred.
- Whether a "VDRIVE detected" interrupt has occurred.
- Whether a hard copy operation is in progress.

The Status Register includes: VDRIVE Detect Flip-Flop, IRQ Enable Flip-Flop, Status Register Read Buffer, and Hard Copy Start circuitry.

VDRIVE Detect Flip-Flop. VDRIVE pulses from the Display Controller occur 60 times per second; they are used as a timer by the system firmware. Each pulse sets the VDRIVE Detect Flip-Flop. The flip-flop is cleared by a write to address X'0811'.

IRQ Enable Flip-Flop. The Processor enables VDRIVE interrupts by writing a one into Bit 0 of the Status Register. This clears a type D flip-flop; when the flop is cleared, VDRIVE interrupts are enabled. Writing a zero into the same bit sets the flip-flop, disabling the interrupts.

Status Register Read Buffer. Four tri-state buffers are enabled by a read from address X'0810'. When enabled, they place the following information on four bits of the Processor's data bus:

- Bit 0 tells whether VDRIVE interrupts are enabled.
- Bit 1 tells whether a VDRIVE pulse has occurred, that is, whether the VDRIVE Detect Flip-Flop is set.
- Bit 2 Tells whether a hard copy is in progress, that is, whether the HC line on the Mother Board is low.
- Bit 7 is true (a logical 1) if VDRIVE interrupts are enabled and a VDRIVE pulse has been detected.

Hard Copy Start. Three gates (two of them functioning as Schmidt trigger inverters) detect when the Processor writes a 1 into bit 2 of the Status Register. When this occurs, the HC (Hard Copy) line briefly goes low. This starts the hard copy. Circuitry on the Display Controller Board and the 4631 Hard Copy Unit will hold HC low until the hard copy operation is complete.

Interrupt Request Circuitry

(See Schematic 5-4.)

The Interrupt Request Circuitry consists of three gates and a strap setting. In the 4025, the strap is set to VEC (for "vectored interrupts"). With the strap in this position, the gates drive the IRQ line low only when (a) a VDRIVE signal has been detected with VDRIVE interrupts enabled and, moreover (b) the interrupt address lines IA0-IA2 hold Display Memory Board's interrupt address (binary 010).

In the 4024, the strap is set to NON (for "non-vectored interrupts"). With the strap in this position, the interrupt address lines are ignored; IRQ will be grounded as soon as VDRIVE is detected with VDRIVE interrupts enabled.

Section 9

DISPLAY CONTROLLER

INTRODUCTION

This section describes the Display Controllers in the 4024 and 4025 terminals. The following are discussed:

- **Display Controller Concepts.** A general overview of the 4024 and 4025 display controllers, showing the major circuit blocks and their functions. Differences between the two display controllers are listed.
- **4025 Display Controller.** A detailed description of the 4025 Display Controller.
- **4024 Display Controller.** A less detailed description of the 4024 Display Controller, with emphasis on its differences from the 4025 Display Controller.

DISPLAY CONTROLLER CONCEPTS

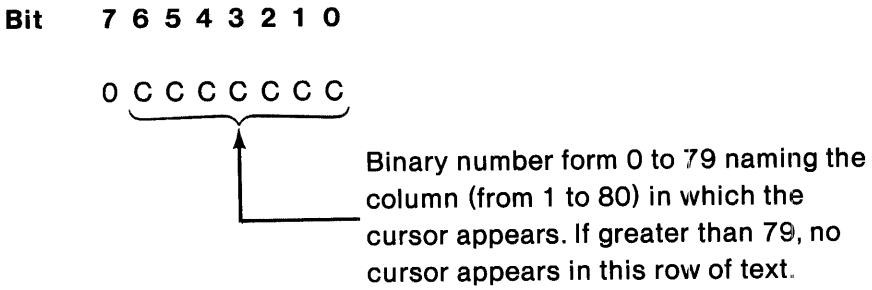
Information from the Paired Bus

The Mother Board's paired bus lines (PB0-PB11) connect the Display Memory Board and the Display Controller Board. The Tracker (on the Display Memory Board) uses these lines to send information about each row of text to be displayed. For each line, this information includes a cursor control character, up to 80 characters of text, and any visual or character font field attribute codes for that text.

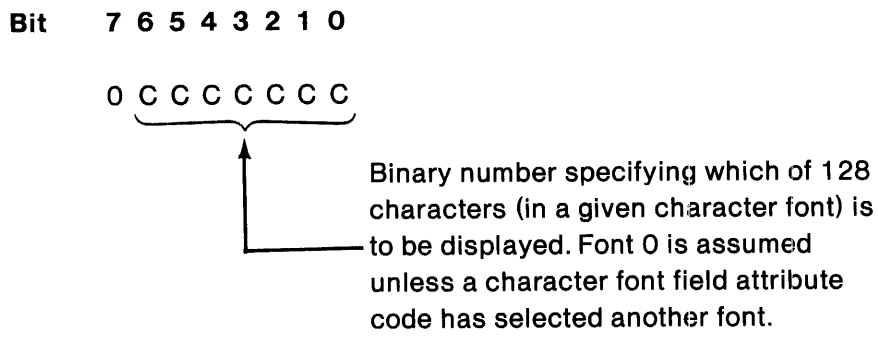
When the Tracker has a byte ready on the paired bus, it sends a TCLK (Tracker Clock) signal on paired bus line PB8. When it has sent all the characters for a row of text, it pulses the BUFUL line (PB9). This resets the visual and character font attributes in the Display Controller so that the Tracker may fill the rest of the line with ASCII *space* characters.

The bytes transferred in this manner have the following formats:

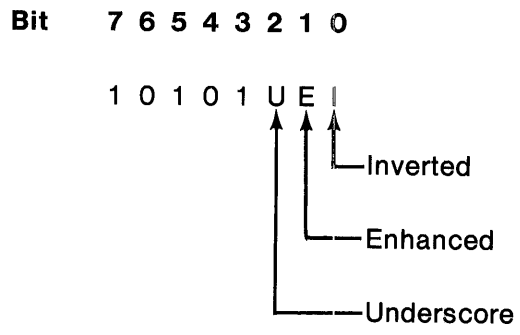
- **Cursor Control Character.** (The first character in each row of text.)



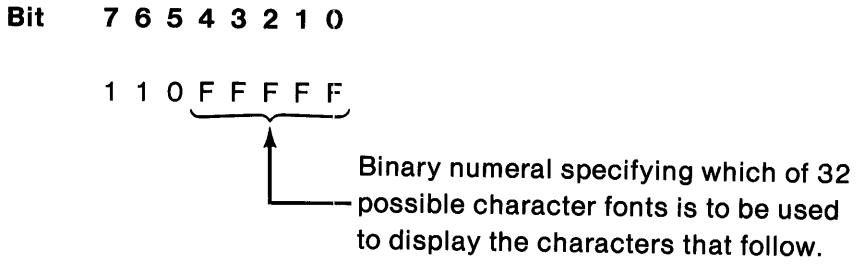
- **Text Characters.**



- **Visual field attribute codes.** Specify whether the following characters are to be displayed with an underscore, an "enhanced" background, or an "inverted" background.



- **Character font field attribute codes.**



Raster Scan Display

The Video Display Unit uses a 60-Hz interlaced scan (Figure 9-1) like those of American television sets. The screen has 525 raster lines, of which 476 are visible. (The other lines are blanked during vertical retrace.) First the even-numbered lines are scanned, and then the odd-numbered ones. A *field* is a complete scan of half the raster lines (the odd half or the even half). A *frame* is two consecutive fields: a complete scan of the entire screen. Thirty frames (sixty fields) are scanned each second.

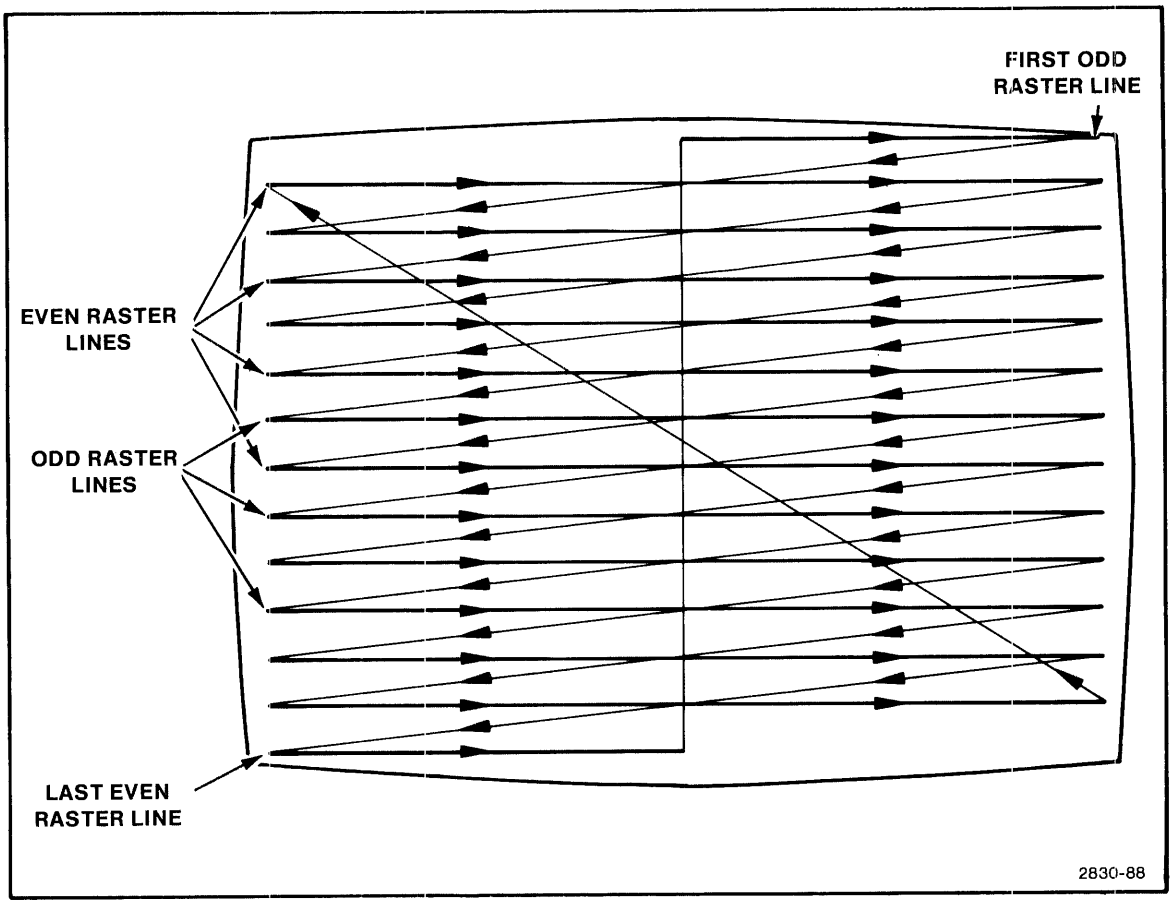


Figure 9-1. Raster Scan Display.

Simplified Block Diagram

Figure 9-2 is a simplified block diagram for the 4024/4025 Display Controllers. The major blocks are:

- Display List Decoder
- Row Buffers
- Video Generator
- Timing and Control Circuitry

Row Buffers

As the Display Controller generates the video signal for each frame, it scans 34 rows of text. To scan a row, it must already have that row in one of its two Row Buffers (Buffer A and Buffer B). (While the Display Controller scans one buffer, it fills the other buffer with characters received from the Tracker.)

4025 Row Buffers

In the 4025, each Row Buffer holds eighty 16-bit words. Each word has 7 bits for the character number, 5 bits for the font number, 3 visual attribute bits, and 1 cursor position bit.

4024 Row Buffers

In the 4024, each Row Buffer holds eighty-one 8-bit words. The first word holds the cursor control character, which specifies in which column (if any) the cursor is to appear. The remaining 80 words each have 7 bits for the character number and one visual attribute bit.

Scanning and Swapping

The Display Controller scans Buffer A seven times, generating seven video raster lines. Then it swaps buffers, and scans Buffer B to generate seven more raster lines. This continues until the entire field has been scanned. The Display Controller then generates the video signal for the next field, starting again at the top row of text on the screen. It scans the even-numbered lines during one field, and the odd-numbered lines during the next field.

Each character cell in the display consists of 14 rows of eight dots each, numbered from 0 to 13. The seven even-numbered dot rows are scanned during each even field, and the seven odd-numbered rows during the following odd field. Thus, during each frame (two fields), all 14 dot rows are scanned.

Display List Decoder

Refer to Figure 9-2. On each TCLK pulse, the Display List Decoder examines the byte coming from the Tracker on paired bus lines PB0-PB7. If the byte is a character, the Decoder loads it into one of the Row Buffers—whichever one is not being scanned at the moment. If the byte is a visual attribute code, it loads the visual attribute bits into a latch; as subsequent characters are loaded into the buffer, their visual attribute bits are set to correspond to the contents of the latch. Once a visual attribute is set for a line of text, it remains set until (a) a new visual attribute code is found, or (b) the end of the text row is reached.

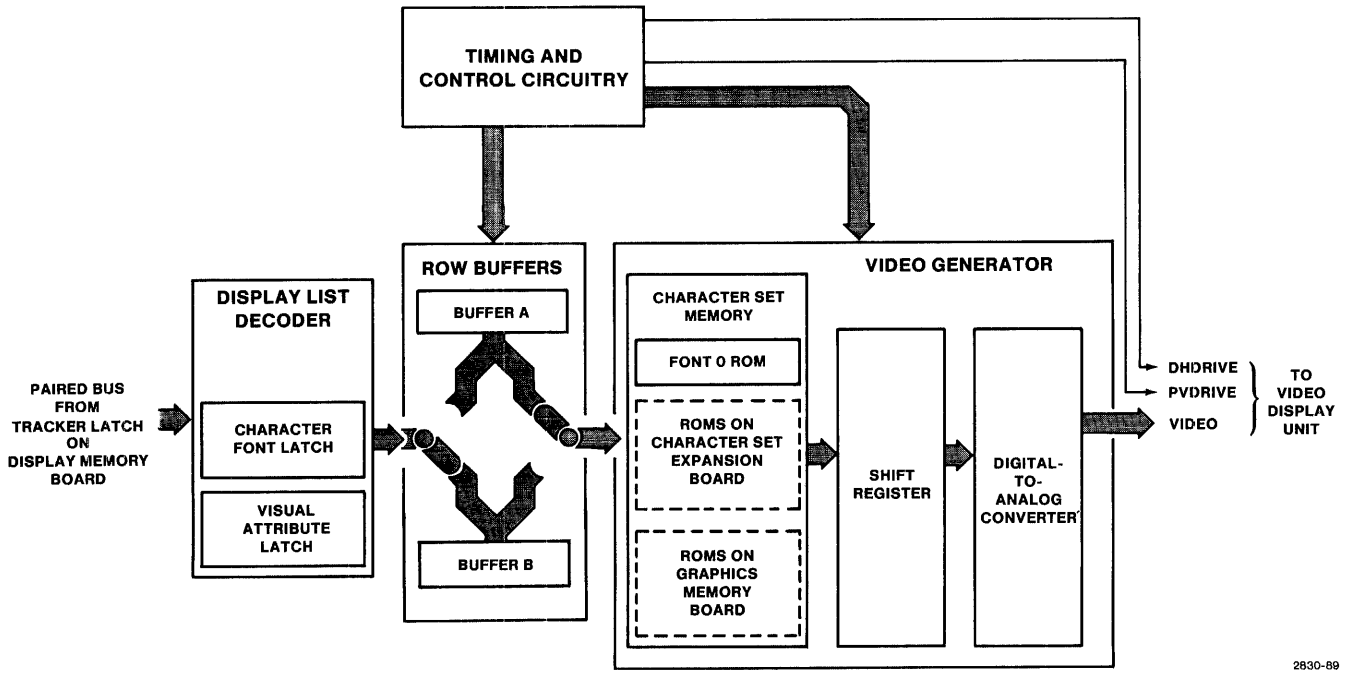


Figure 9-2. Simplified Display Controller Block Diagram.

Video Generator

The Video Generator consists of the Character Set Memory, a Shift Register, and the Digital-to-Analog Converter.

Character Set Memory

The character dot matrix patterns are stored in the Video Generator's character set memory. (This memory is not directly accessible by the Processor.) The Character Set Memory includes a ROM on the Display Controller Board holding dot patterns for the standard ASCII characters. It may also include:

- ROMs on the 4025 Character Set Expansion Board for alternate character fonts (rulings characters, math characters, etc.). (In the 4024, the rulings ROM, if installed, is on the Display Controller Board.)

- RAMs on the 4025 Graphics Memory Board holding characters defined by the user (with the SYMBOL command) or by the Processor (as it executes VECTOR commands to draw graphs).

Each word of character set memory holds eight bits—the eight dots of one row of a character's dot matrix. The address lines which access that word specify the font number (from 0 to 31), the character number in that font (from 0 to 127), and the row number (from 0 to 13) of a particular row in that character's dot matrix.

To generate the raster lines for a row of text, the Video Generator scans the Row Buffer holding that row of text. As it reaches each character, it reads its character set memory to learn which dots to display for the current raster line. It scans the row seven times, generating seven raster lines. Then it goes to the next row, swaps row buffers, and scans the new buffer.

(While the Video Generator is scanning one buffer, the Display List Decoder fills the other buffer with the characters for the next row of text.)

Shift Register

As the Video Generator reads (from its character set memory) the dot pattern for one row of a character's dot matrix, it places that dot pattern into a Shift Register. The dots are then shifted out of the register, converting the information from parallel data to serial data. The data is processed by the Digital-to-Analog Converter, becoming the video signal for the Video Display Unit.

Digital-to-Analog Converter

The Digital-to-Analog Converter combines the following signals to make a z-axis signal for the Video Display Unit:

- The string of dots (serial data) coming from the Shift Register.
- The visual field attribute bits for the current character. (These bits determine whether the background is "enhanced" (brighter than normal) or "inverted," and whether the character is displayed with an underscore.)
- The cursor position information—whether a cursor is to appear at the character currently being scanned.

Video Pipeline

To generate the video signal corresponding to a particular row of a character's dot matrix, the Video Generator must do a number of things, one after the other:

1. The Video Generator takes the "character number" and "font number" bits from the Row Buffer and the "raster number" bits from the Timing and Control Circuitry. It places these bits on the address lines of its Character Set Memory.
2. After a delay while it responds, the Character Set Memory places a byte on the parallel inputs of the serial-to-parallel Shift Register. This byte tells which dots are to be turned on in a particular row of the character's dot matrix.
3. The Shift Register receives the byte on its parallel input and shifts it out, converting the data to a serial format.

To accomplish these operations in time required to scan a single character, the Video Generator functions as a sort of "pipeline" (Figure 9-3). For instance, suppose the word THE is in the row buffer being scanned.

1. On one negative edge of the column clock (the COLCLK signal), the letter T is placed on the character set memory address lines.
2. By the next negative COLCLK edge, the dot pattern for one row of the T's dot pattern will have settled on the memory output lines. This COLCLK pulse clocks the bit pattern into the Shift Register, and simultaneously clocks the letter H onto the character set memory address lines.

3. By the third COLCLK edge, that row of the T's dot pattern will have been shifted out of the Shift Register. (A faster clock is used to shift the bits out of the Shift Register.) This third COLCLK pulse loads the bit pattern for H into the Shift Register and applies the ASCII code for E to the character set memory address lines.

Three characters are in the pipeline at any given time. The first character (to enter the pipeline) is being shifted out of the shift Register. The second character is being read from Character Set Memory and presented to the Shift Register parallel inputs. The third character is being read from the current Row Buffer and presented to the Character Set Memory address lines.

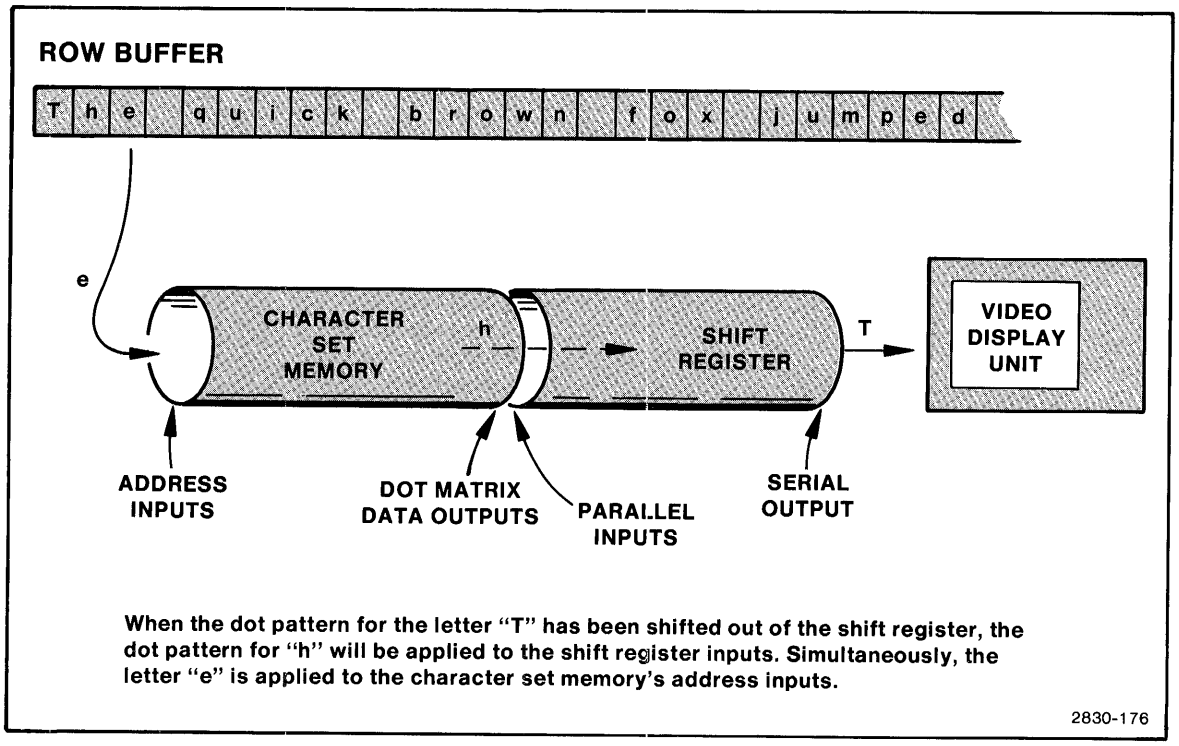


Figure 9-3. Video Pipeline.

Differences Between the Display Controllers

The 4024 Display Controller is simpler than the 4025 Display Controller. It provides only the "enhanced" visual attribute, has only two character fonts ("standard" and "rulings"), and makes no provision for hard copies or an external video monitor.

In the 4024 Display Controller, character font 1 (the rulings characters) is not treated by the Video Generator as a separate character font. Instead, the rulings characters replace the "snoopy" characters (the two-letter menmonics for control characters) and are treated as part of font 0, the standard font. (For each "font 1" character it encounters, the 4024 Processor places a corresponding font 0 control character in the display list.

In the 4025, the first character of each row (the cursor control character) is latched into a separate cursor counter. Thus, the Row Buffers hold only the 80 characters which are to be displayed. In the 4024, however, the cursor character is loaded into the first word of the Row Buffers, which are 81 characters long.

In the 4024, words of the Row Buffers each have eight bits—one bit indicates whether an enhanced background is to be displayed, and the other seven bits name the ASCII character. In the 4025, each Row Buffer word has 16 bits—seven for the ASCII character, five for the character font number, three visual attribute bits, and a bit to indicate whether the cursor is to be displayed with the character.

4025 DISPLAY CONTROLLER

Figure 9-4 is a block diagram for the 4025 Display Controller. It resembles Figure 9-2, but differs from that diagram as follows:

- It shows circuits peculiar to the 4025 as well as those common to both the 4024 and the 4025. For instance, Figure 9-4 includes the Hard Copy Circuitry; since the 4024 lacks hard copy capability, that block was omitted from the generalized block diagram in Figure 9-2.

- It is more detailed.

The 4025 Display Controller appears in Schematics 10-1 to 10-5. In addition, parts of those schematics appear as illustrations in this section.

Display List Decoder

Whenever the Tracker has a byte of data for the Display Controller, it places that byte on paired bus lines PB0-PB7 and sends a TCLK pulse on paired bus line PB8. On each such TCLK pulse, the Display List Decoder examines the byte on PB0-PB7.

- Bytes of the form 0CCCCCCC (most significant bit zero, other bits anything) are ASCII characters. Bits CCCCCCC are sent to whichever Row Buffer currently is being loaded. (The first ASCII character coming from the Tracker for each row of text is the cursor control character; it is loaded into the cursor counter rather than the Row Buffer.)

  In addition, the Display List Decoder sends the following to the Row Buffer: five bits naming a character font, three bits naming visual attributes, and one bit to tell whether the cursor is to be displayed with that character. Thus, for each character, the Display List Decoder sends 16 bits of information to the Row Buffer.

- The first ASCII character in a row of text, however, is loaded into the Cursor Counter rather than sent to the Row Buffer. This character is treated as a number telling where on the line the cursor should be displayed.

- If the byte is of the form 10101UIE, then it is a visual attribute code. The Display List Decoder loads the least significant bits (UEI) into its Visual Attribute Latch.

- If the byte is of the form 110FFFFF, then it is a character font field attribute code. The byte's five least significant bits (FFFFF) are loaded into the Display List Decoder's Font Attribute Latch.

- If the byte is not of the forms 0CCCCCCC,10101UEI, or 110FFFFF, then the Display List Decoder ignores it.

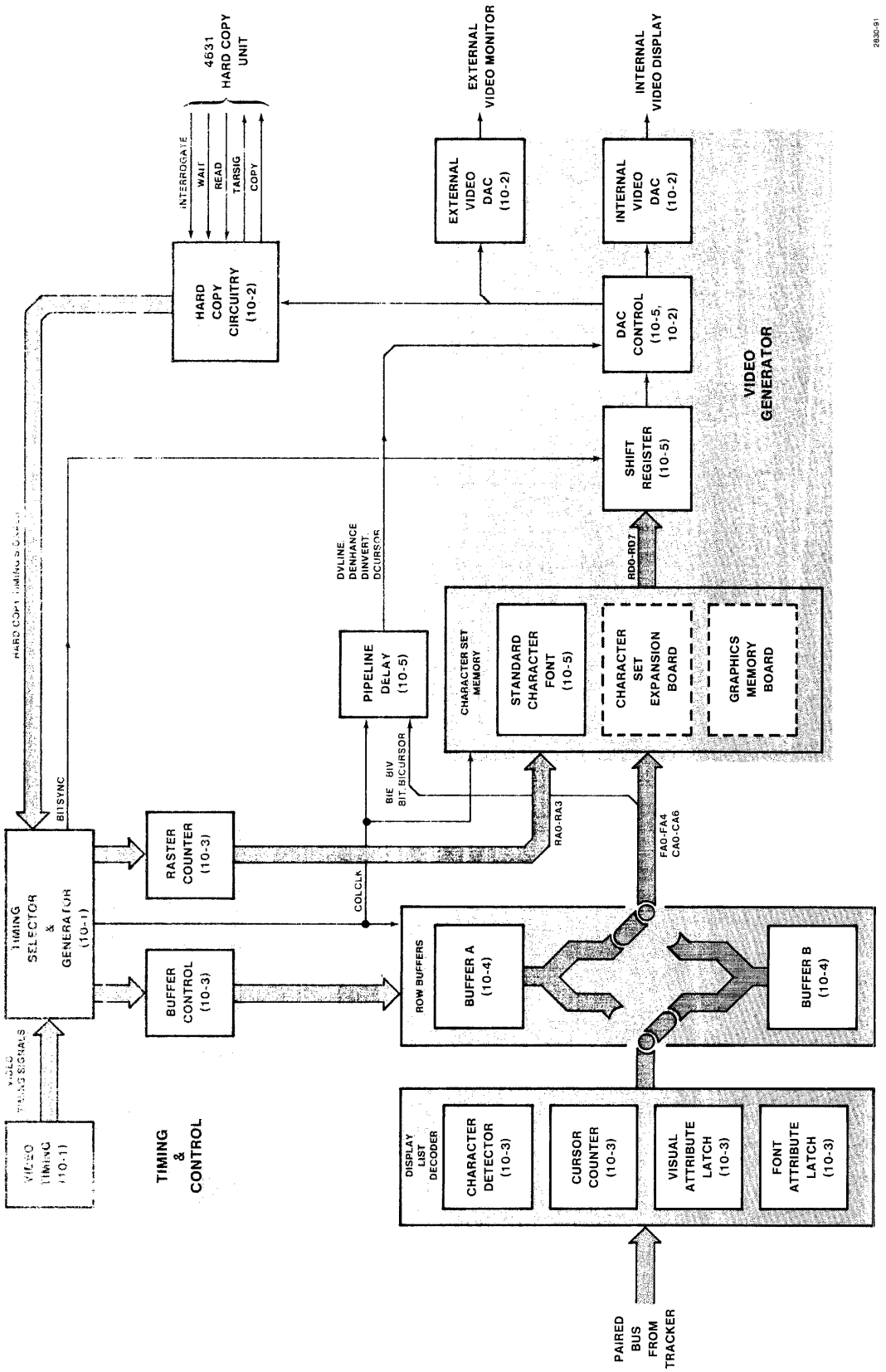


Figure 9-4. 4025 Display Controller Block Diagram.

The Display List Decoder consists of the following blocks: Visual Address Latch, Font Attribute Latch, Cursor Counter, and Character Detector. (See Figures 9-4, 9-5, and Schematic 10-3.) Whenever a character is loaded into the Row Buffer, the following bits of information are sent to that buffer:

- **BICA0-BICA6 (Buffer Input Character Address).** These bits come directly from paired bus lines PB0-PB6.
- **BIU (Buffer Input Underline), BIE (Buffer Input Enhanced), and BII (Buffer Input Inverted).** These bits come from the Visual Attribute Latch.
- **BICURSOR (Buffer Input Cursor).** This bit specifies whether a character is to be displayed with a cursor. It comes from the Cursor Counter.
- **BIFA0-BIFA4 (Buffer Input Font Address).** These bits come from the Font Address Latch.

Visual Attribute Latch

(See Figure 9-5, Schematic 10-3.)

Paired bus bits PB7-PB3 are decoded. If they are 10101 and a TCLK pulse occurs, the Display List Decoder recognizes the byte on the paired bus as a visual field attribute code and loads the least significant bits (PB2-PB0) into a latch. The latch outputs are the current visual attribute bits. They remain in effect until either of the following occurs:

- The latch is loaded again by another visual field attribute code.
- The Tracker pulses the BUFUL line, marking the end of the current row of text.

Font Address Latch

(See Figure 9-5, Schematic 10-3.)

Two gates decode the most significant three paired bus bits (bits PB5-PB7). When they are 110 and a TCLK pulse occurs, the gates clock bits PB0-PB4 into the Font Address Latch. The font address in the latch remains in effect in effect until either of the following occurs:

- The latch is loaded again by another character font field attribute code.
- A pulse on the BUFUL (Buffer Full) line signals the end of the current row of text.

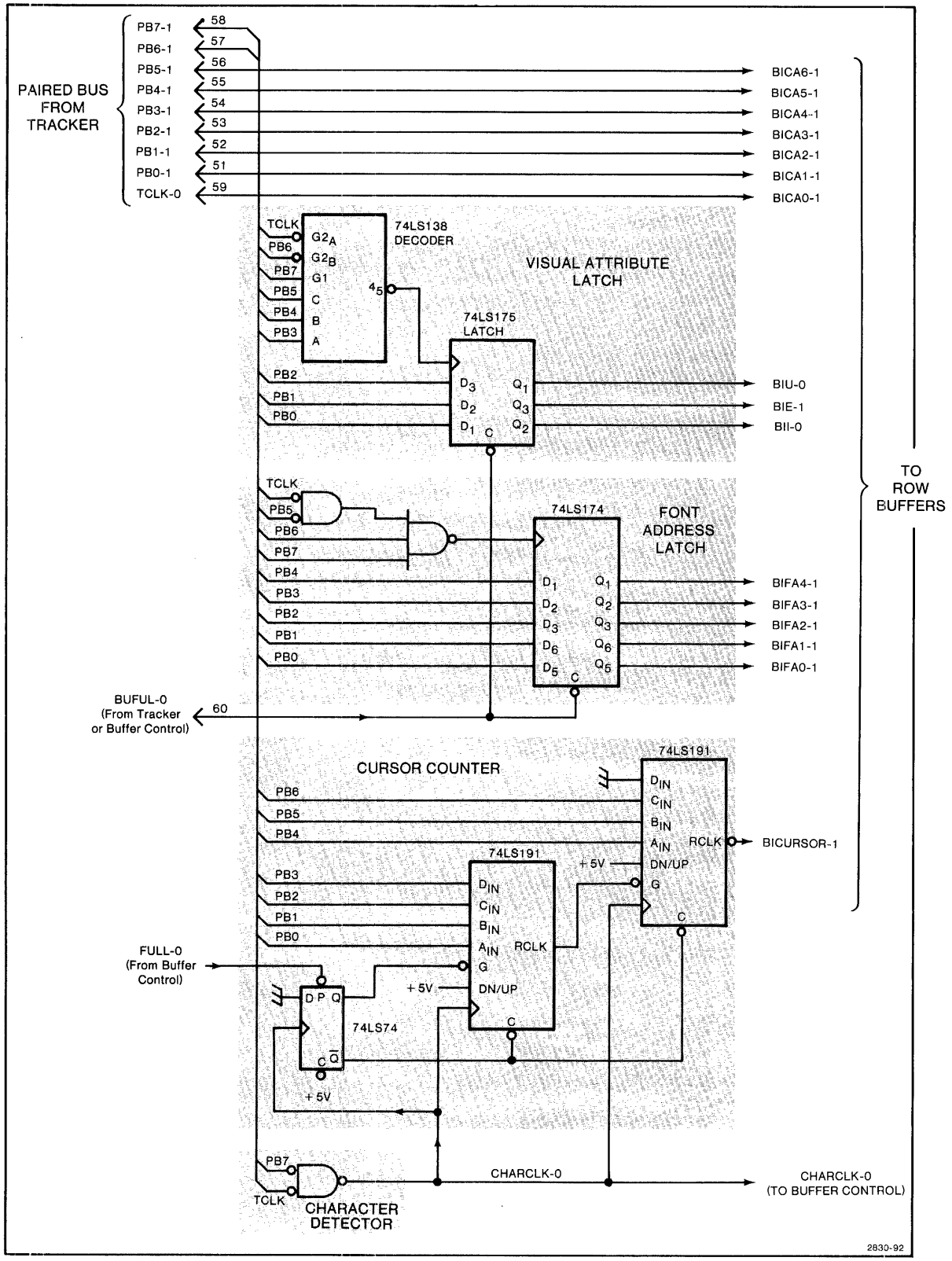


Figure 9-5. 4025 Display List Decoder.

Character Detector

(See Figure 9-5, Schematic 10-3.)

Whenever the Tracker places an ASCII character on the paired bus, it sets bit PB7 to zero and sends a TCLK pulse. The Character Detector gate detects this event and sends a CHARCLK (Character Clock) pulse. The CHARCLK pulse loads the character into the current Row Buffer and clocks the Cursor Counter.

Cursor Counter

A 74LS74 flip-flop keeps track of whether the first character of a row of text has occurred. At the end of the previous row of text, a FULL signal from the Buffer Control circuitry has set this flip-flop, causing it to send the COUNT signal false. With COUNT false, the two 74LS91 counters are ready to be loaded.

As the Tracker sends the cursor-control character (which begins the row of text), a CHARCLK pulse loads bits PB0-PB6 into the 74LS191s. If the number loaded is non-zero, then the second counter's RCLK output drives the BICURSOR line high. (A high on the BICURSOR line indicates that the next character will be displayed *without* a a cursor.)

The first CHARCLK pulse also clears the COUNT flip-flop, making it send the COUNT signal. As more characters come from the Tracker, their CHARCLK pulses decrement the counter. If the count reaches zero, the BICURSOR goes low; the next character will be displayed with a cursor.

Since the cursor control character loads the Cursor Counter, it determines where the cursor appears. If the cursor control character is zero (ASCII NUL character), then the cursor appears in column one of the row. If the cursor control character is 79 (ASCII O character), the cursor will be in column 80. If the cursor control character is 80 or more, then no cursor appears in that row of text.

4025 Row Buffers

The Row Buffers appear in Figure 9-6 and Schematic 10-4. Each buffer consists of two 128-by-8 RAMs (MCM6810), an address counter (74LS161 and 74LS196), and four bidirectional data buffers (8833).

As the Row Buffers are identical, the following description refers only to Buffer A. Buffer B operates identically, although its signals are BUFB, BUFBSEL, etc.

Buffer Architecture

Buffer A has two RAMs, connected to form a memory of 128 16-bit words. Of the 128 words, only 80 are actually used, as the maximum length of a row of text is 80 characters.

The BUFASEL line from the Buffer Control functions as a chip select to enable the RAMs. The CNTACLR line clears the counters at the start of each load or scan operation.

The BUFA line determines whether the buffer is to be written into, or read from. When BUFA is true (low), the RAMs are enabled for writing and the bidirectional data buffers steer data from the buffer input lines to the RAM data pins. When BUFA goes false (high), the RAMs are in "read" mode, and the data buffers drive the Character Set Memory address lines.

The buffer input lines from the Display List Decoder are:

- BICA0-BICA6 (Buffer Input Character Address lines). These specify the character to be displayed.
- BIFA0-BIFA4 (Buffer Input Font Address lines). These specify the font.
- BIU, BIE, and BII. Specify if the character is to be displayed with the *underline*, *enhanced*, or *inverted* visual attributes. Note that BIU and BII are true low, while BIE is true high.
- BICURSOR, which says whether the cursor is to be displayed with the character.

The corresponding buffer output data lines are used by the Character Set Memory as address inputs. These lines are: CA0 to CA6, FA0 to FA4, ULINE, ENHANCE, INVERT, and CURSOR.

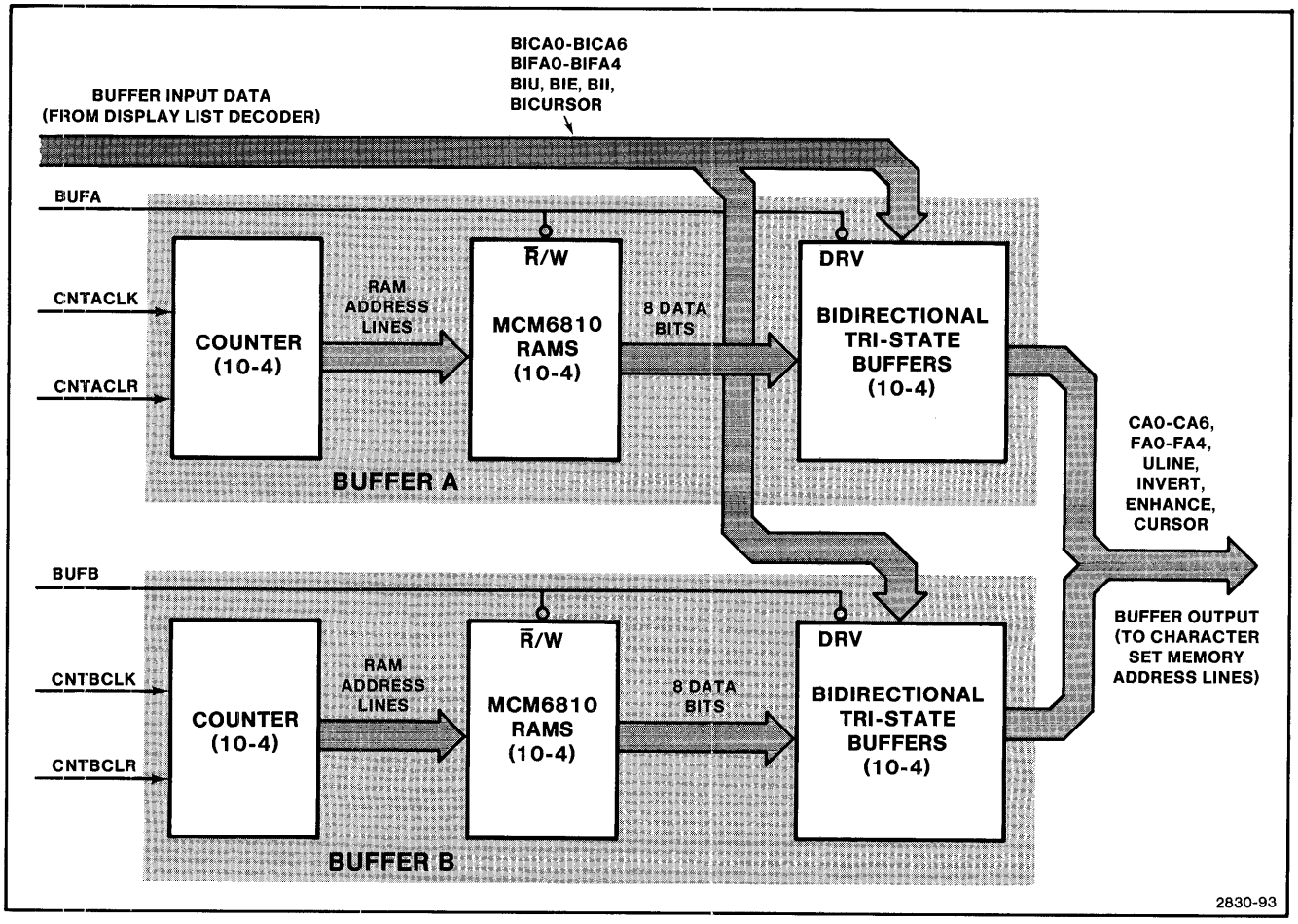


Figure 9-6. 4025 Row Buffers.

Loading a Buffer

While Buffer A is being loaded with characters from the Tracker, each CHARCLOCK pulse from the Display List Decoder generates simultaneous BUFASEL and CNTACLK pulses (true low) from the Buffer Control circuitry. During this time (while the buffer is being loaded), BUFA-0 is held low, putting the RAMs in "write" mode. As BUFASEL-0 goes low, the data on the buffer input lines is loaded into the RAMs. Then, as CNTACLK goes high, the buffer address counter is advanced.

The 74LS161 is a divide-by-16 counter; on every 16 CNTACLK pulses, it sends a clock pulse to the 74LS196. The 74LS196 has two halves: a divide-by-five counter (outputs Q_B, Q_C, and Q_D), and a divide-by-two counter (output Q_A). The divide-by-five counter counts from 0 to four; on the fifth pulse from the 74LS161 (which corresponds to the 80th CNTACLK pulse), it resets to zero and clocks the Q_A counter. The Q_A output goes high, sending the BUFAFUL signal. BUFAFUL tells the Buffer Control circuitry that Buffer A has been loaded with 80 characters and it is time to swap buffers.

Scanning a Buffer

While Buffer A is being scanned to generate memory addresses for the Video Generator's Character Set Memory, the BUFA line is held high, so that the RAM's internal output drivers are enabled and the bidirectional data buffers are always in "receive" mode. Also, BUFASEL is held low, so that the Row Buffer is always enabled for reading.

As the COLCLK signal advances the scan from one character to the next, the Buffer Control sends a CNTACLK pulse to advance the buffer address counters. As before, at the end of the row of text the 74LS196 sends a BUFAFUL pulse; this tells the Timing and Control circuitry that one scan of the row has been completed. After seven such scans, the Buffer Control circuitry will swap buffers.

Control Signals

Figure 9-7 shows the Buffer A timing. Note that:

- When the buffer is being loaded, BUFA is held low, and BUFASEL and CNTACLK are pulsed.
- When the buffer is being scanned, however, BUFA is held high, BUFASEL is held low, and only CNTACLK is pulsed.

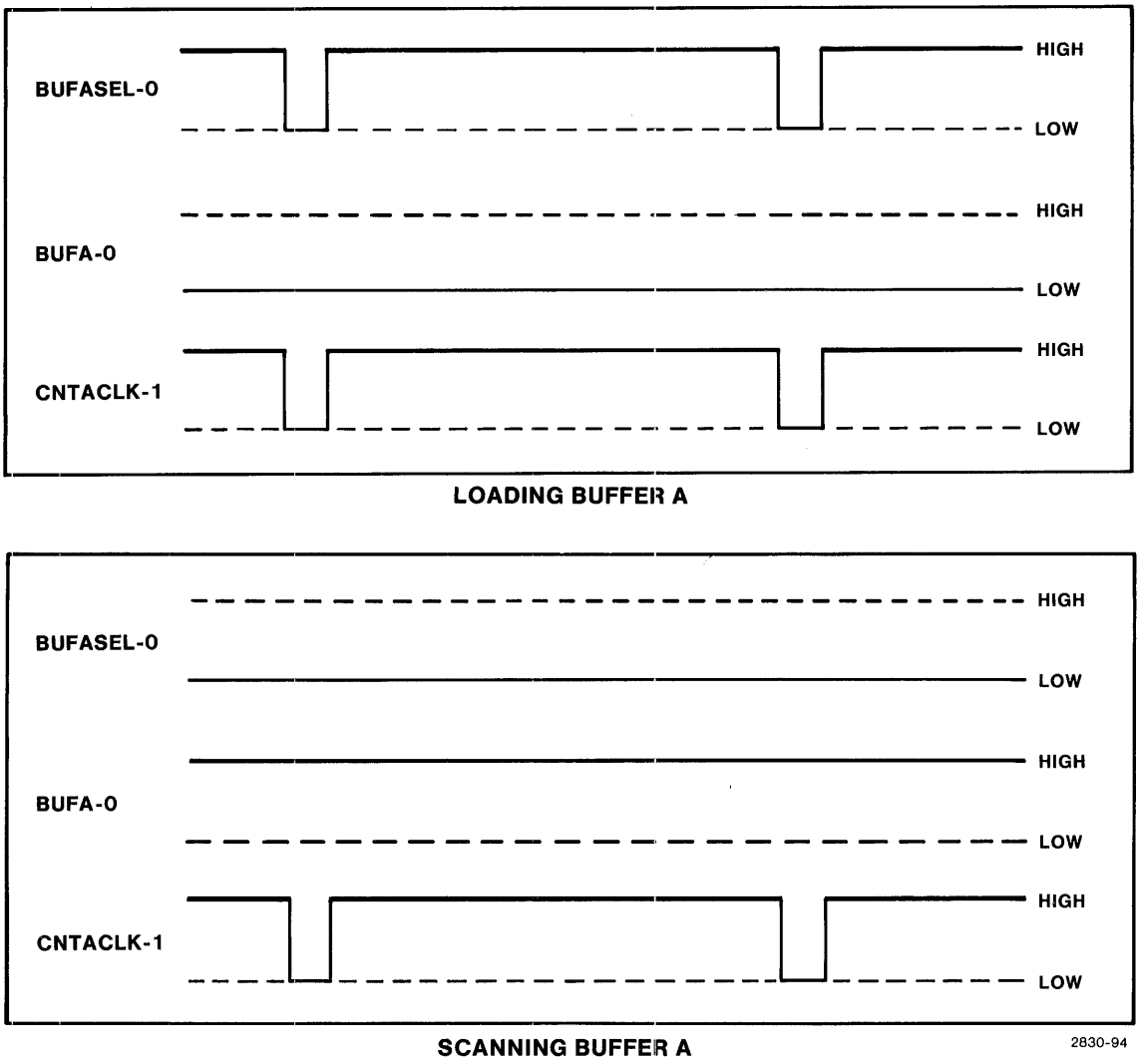


Figure 9-7. Buffer Control Waveforms.

Video Generator

The 4025 Display Controller's Video Generator includes these blocks: Character Set Memory, Shift Register, Pipeline Delay, DAC Control, Internal Video DAC, and External Video DAC. (While the Hard Copy Driver may be considered part of the Video Generator, it will be discussed separately.)

Character Set Memory

The Character Set Memory (Figure 9-8) has three major blocks: Standard Character Font, Optional ROM Character Fonts, and RAM Character Fonts. The Standard Character Font resides on the Display Controller Board; the other fonts, if installed, are on the Character Set Expansion Board (Option 31) and the Graphics Memory Board (Option 35).

The Character Set Memory address inputs are: the FA0-FA4 (Font Address) and CA0-CA6 (Character Address) signals from the Display List Decoder; and the RA0-RA4 (Raster Address) signals from the Timing and Control circuitry. In addition, COLCLK regulates the flow of information through the Video Pipeline. These signals are provided as inputs to the Standard Character Font circuitry; they are also sent (via connector J31) to the Character Set Expansion and Graphics Memory Boards.

HDRIVE, VDRIVE, BLANK, and BGATE are also used on the Graphics Memory Board; they are provided via connector J31.

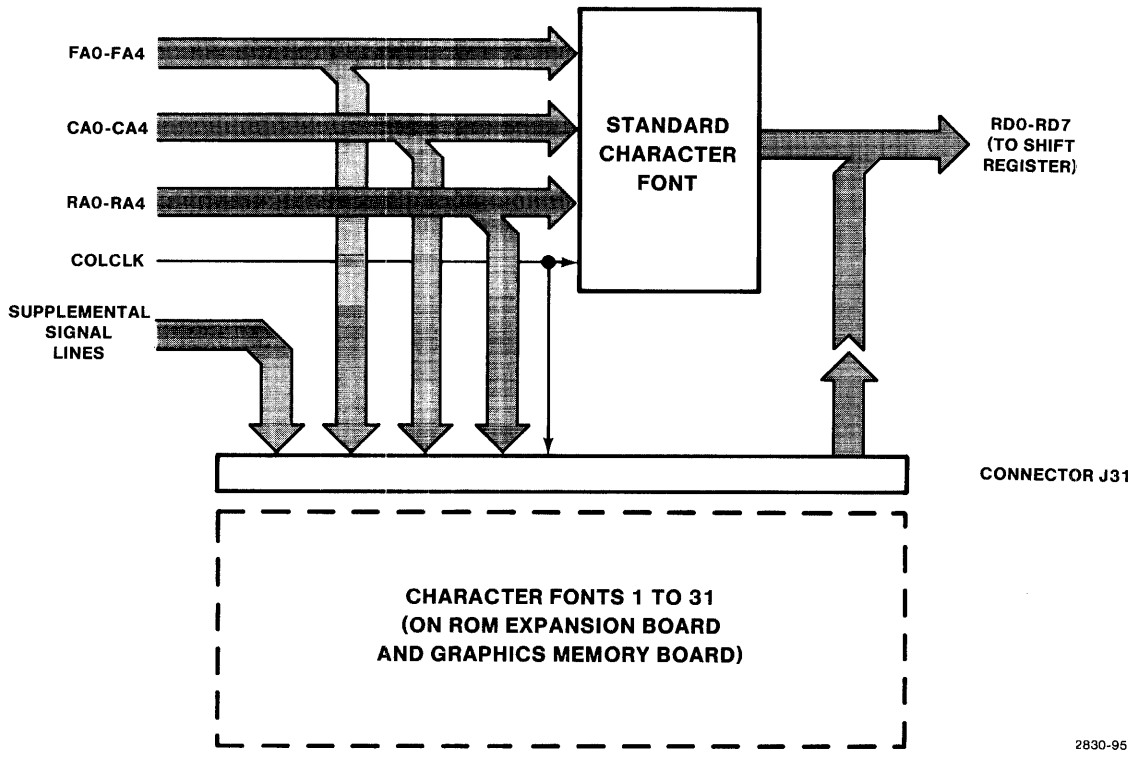


Figure 9-8. 4025 Character Set Memory.

Standard Character Font. Figure 9-9 and Schematic 10-5 show the Standard Character Font circuitry. This circuitry includes:

- An MCM6575 character generator ROM. This ROM holds dot patterns for each of the ASCII characters, including "snoopy" characters (two-letter mnemonics for the ASCII control characters).
- A 7483 adder. This adjusts the raster address inputs for the character generator ROM.
- 74LS174 latches for the character address inputs.
- 74LS367 tri-state buffers for the ROM output bits RD0-RD7.
- Font decoding gates to recognize when the standard font (font zero) is specified.
- A type D flip-flop to delay the FONT0 signal by one COLCLK pulse.

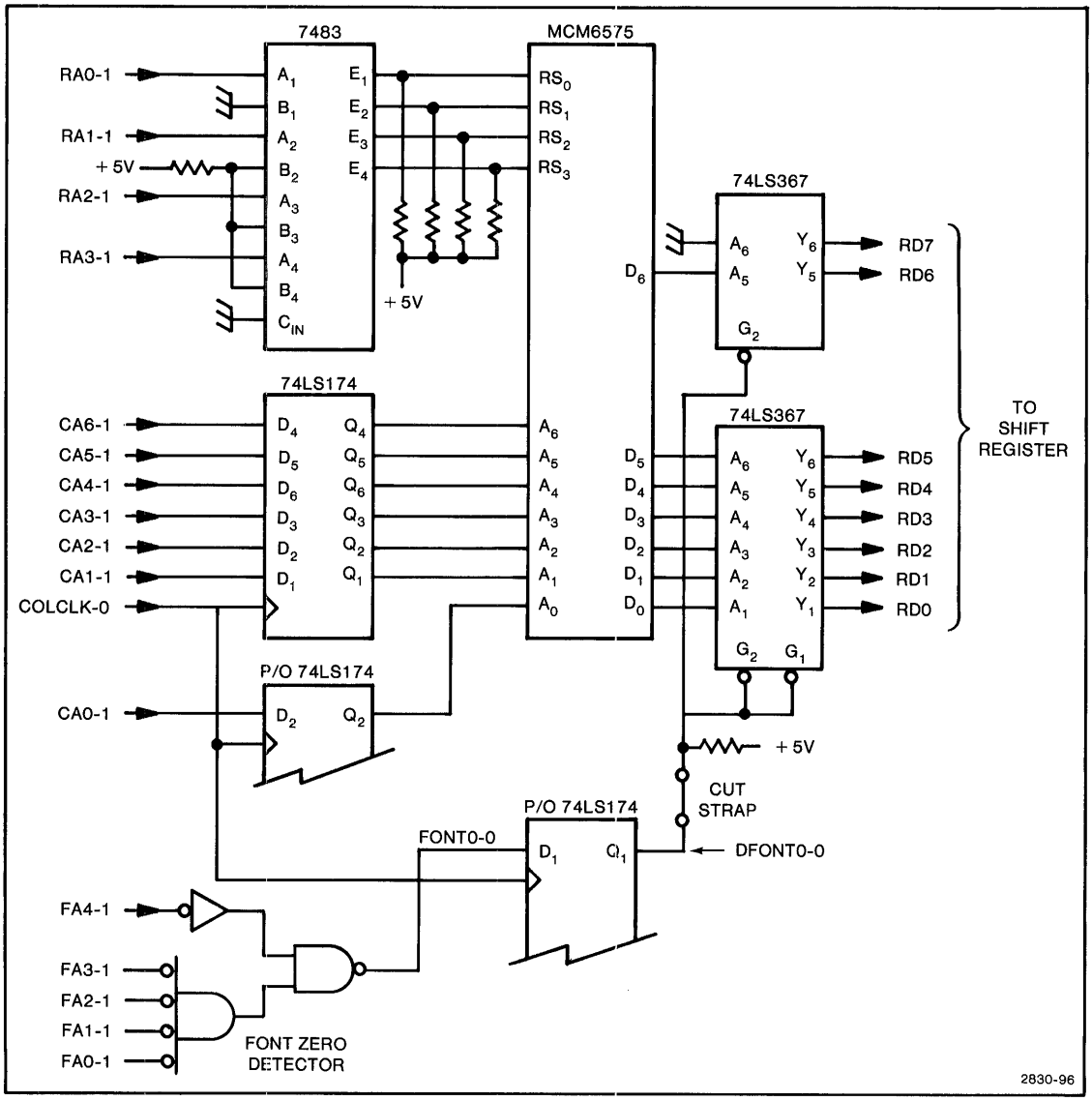


Figure 9-9. Standard Character Font.

The character generator ROM holds dot patterns like that in Figure 9-10A. When the proper raster address is supplied on the RS0-RS3 pins, and an ASCII character's seven bits are supplied on the A0-A7 pins, the ROM provides the dot pattern for one raster line of that character. Note that the dot pattern from the MCM6575 is part of a 7-by-14 dot matrix, and starts at the top row in that matrix: raster line 0.

However, we want the dot pattern to be part of an 8-by-14 matrix, and to be shifted down two rows in that matrix. (See Figure 9-10B.) The extra column of dots is provided by hard-wiring the eighth bit of each Standard Character Font word to zero. Shifting the dot pattern downward in the cell requires additional circuitry: the 7483 adder.

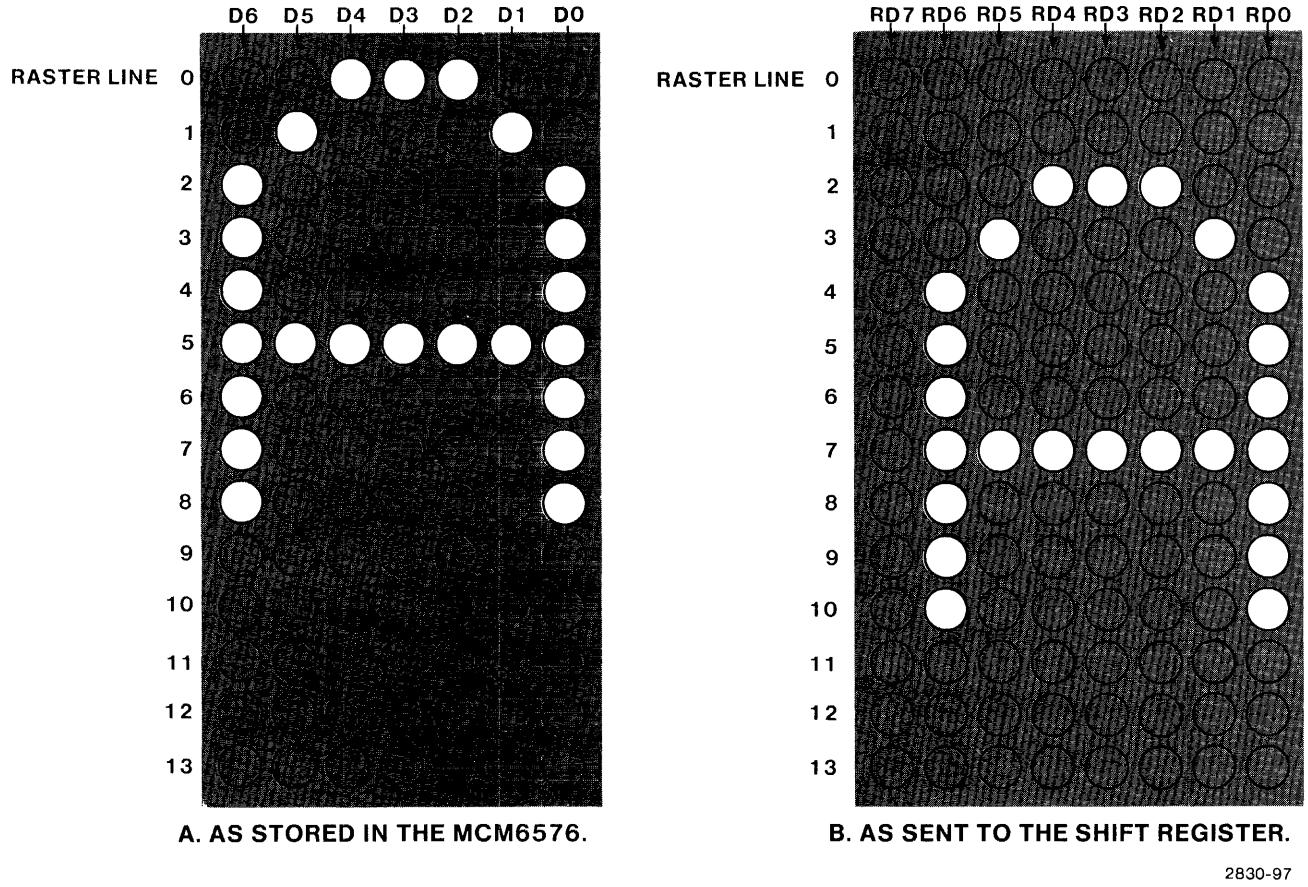


Figure 9-10. A Character's Dot Matrix Pattern.

The 7483 adds in modulo 16 binary arithmetic. To each raster address on the RA0-RA3 lines, it adds 14 (binary 1110). In modulo 16 arithmetic, adding 14 is the same as subracting 2. Thus, the 7483 decreases each raster address by two before applying it to the character generator ROM. This shifts each character's dot pattern downward by two raster lines.

The 74LS174 latches hold the character address (CA0-CA6) during each COLCLK cycle. (As described earlier, each character is applied to the Character Set Memory on one COLCLK pulse, and its dot pattern is read from that memory on the following COLCLK pulse.)

The 74LS367 tri-state buffers take the seven output bits from the character generator ROM, together with an eighth bit (hard-wired to zero), and apply them to output lines RD0 to RD7 (Read Data Bit 0 to Read Data Bit 7). These buffers are enabled on the next COLCLK pulse after a Font Zero character has been applied to the Character Set Memory address inputs.

The Font Zero Detector (two gates and an inverter) monitors the font address lines FA0-FA4. When FA0-FA4 are all low, it sends the FONT0 signal. A type D flip-flop (part of the Pipeline Delay circuitry) delays FONT0 by one COLCLK pulse. The delayed signal (DFONT0) arrives at the tri-state buffers at the same time as the dot matrix data for the character being displayed.

It is possible to use a different character font than that provided in the MCM6575. In that case, the standard character font would reside in ROMs on the Character Set Expansion Board, and the Standard Character Font Circuitry on the Display Controller Board must be disabled. Cutting a strap in the DFONT0 signal line accomplishes this.

Other Character Fonts. Fonts 1 to 31, if installed, reside on either the Character Set Expansion Board or the Graphics Memory Board. These are accessed through the ribbon cable attached to connector J31. Table 9-1 lists the signals at this connector.

Table 9-1

SIGNALS AT DISPLAY CONTROLLER CONNECTOR J31

To Other Boards	Pin No.	From Other Boards	Pin No.
Font Address Lines		Dot Matrix Information	
FA0-1	27	RD0-1	1
FA1-1	11	RD1-1	18
FA2-1	28	RD2-1	2
FA3-1	12	RD3-1	19
FA4-1	29	RD4-1	3
		RD5-1	20
Character Address Lines		RD6-1	4
		RD7-1	21
CA0-1	7		
CA1-1	24	Alphabetic/Graphic Information	
CA2-1	8		
CA3-1	25	GRAPH-0	13
CA4-1	9		
CA5-1	26	Supplemental Signal Lines	
CA6-1	10		
		HDRIVE-1	15
Raster Address Lines		VDRIVE-1	16
		BLANK-0	31
RA0-1	5	BGATE-0	14
RA1-1	22	HCIP-0	32
RA2-1	6		
RA3-1	23		
Column Clock			
COLCLK-1	30		

Shift Register

The Shift Register (Schematic 10-5) converts parallel data to serial data. On each COLCLK pulse, the Character Set Memory presents data bits RD0-RD7 to the Shift Register parallel inputs. These bits represent one row of dots in some character's dot matrix pattern. Before the next COLCLK pulse, the Shift Register (clocked by SHIFTCLK) shifts the bits out its serial port; the data become part of the video signal.

Bit RD7 is loaded into the Shift Register's "H" flip-flop, so that it is shifted out first. Bit RD0, loaded into the "A" flip-flop, is the last bit shifted out. Since the Video Display Unit scans from left to right, this puts bit RD8 at the left edge of the character cell, and bit RD0 at the right edge.

BITSYNC and SHIFTCLK. Figure 9-11 shows the COLCLK and BITSYNC waveforms. BITSYNC is synchronized with the COLCLK column clock; it provides a brief low pulse just before COLCLK goes high. This pulse loads bits RD0-RD7 from the Character Set Memory into the Shift Register. (This is done just before the next COLCLK pulse presents a new character address to the Character Set Memory inputs.) During the next character time, SHIFTCLK clocks the row of dots out of the Shift Register.

SHIFTCLK Sources. The SHIFTLCK signal has one of two sources, depending on whether the character being displayed is a graphic character or an alphanumeric character. Graphic characters fully occupy their character cells, while alphanumeric characters are compressed (horizontally) to leave space between adjacent characters of the display.

When graphic characters are displayed, SHIFTCLK runs at about 12 MHz (exactly eight times the COLCLK frequency), so the eight dots of each row of a graphics cell are equally spaced across that cell. For alphabetic characters, however, SHIFTCLK runs at about 14 MHz. The higher clock rate compresses each character slightly in its cell, giving a more space between adjacent alphanumeric characters.

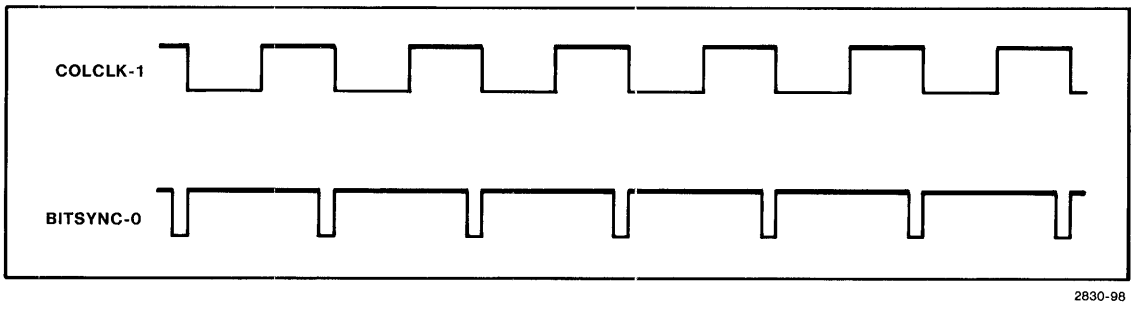


Figure 9-11. COLCLK and BITSYNC.

Pipeline Delay

To display a given character, the character information is applied to the Character Set Memory on one COLCLK pulse. At the following COLCLK pulse, the dot pattern (from memory) is applied to the Shift Register. The serial data bits are shifted out (to become the video signal) before the next COLCLK pulse.

However, the ULINE, ENHANCE, INVERT, and CURSOR signals from the Row Buffers do not pass through the Character Set Memory or the Shift Register. Consequently, they are not delayed by those circuit blocks; without circuitry to provide an equal delay, these signals would reach the end of the Video Pipeline prematurely, before the dot pattern bits of the character with which they are associated.

The Pipeline Delay circuitry (Schematic 10-5) provides the necessary delay. It consists of several short shift registers, made from type D flip-flops. The flip-flops are clocked by COLCLK, the signal which controls the flow of data through the Video Pipeline.

The signals are delayed as follows:

- The INVERT, ENHANCE, and CURSOR signals are delayed two COLCLK pulses. the delayed signals are DINVERT, DENHANCE, and DCURSOR.

- When ULINE and a raster address of 12 (binary 1100 on lines RA3, RA2, RA1, RA0) is detected by the **ULINE Detector** (comprised of two gates), the ULINE Detector's output signal is also delayed by two COLCLK pulses. The delayed output is the DULINE signal.

The delayed signals are then sent to the D-to-A converters.

Other flip-flops provide similar delay or latching functions in the Standard Character Font circuitry. One flip-flop latches the CA0 signal at the input to the character generator ROM. Another flip-flop delays FONT0 by one COLCLK pulse so that the tri-state output buffers are not enabled prematurely. These two flip-flops were discussed in connection with the Standard Character Font.

DAC Control

The DAC (Digital-to-Analog Converter) Control circuitry appears in Schematics 10-5 and 10-2. This circuitry takes the serial Shift Register output and combines it with the delayed visual attribute signals (DULINE, DENHANCE, DINVERT, DCURSOR) and signals from the Timing and Control Circuitry (BLACK1, BLACK2, BLANK, and CHOPCLK) to control the Digital-to-Analog Converters. The signals generated by the DAC Control are:

- **DACBLANK.** Causes the DACs to blank the video output.

- **DACENHANCE.** Tells the DACs to display characters with an enhanced background. Those dots of a character's dot matrix which otherwise would be completely turned off are partially turned on.

- **CHOPPED.** The "chopped video" signal tells the DACs when to turn on the crt beam to create the lighted dots of the display.

- **DIM.** The DIM1 and DIM2 signals are logically identical, but go to different DACs. These signals are used when a character is being displayed with inverted background; they prevent the inverted background from being abnormally bright (as bright as the cursor).

The circuitry for generating each of these signals is described in turn.

DACBLANK. The DACBLANK signal is generated by a 74LS00 functioning as inverting-input, non-inverting output OR gate. The DACBLANK signal is enabled by a BLANK signal from the Timing and Control Circuitry, or by a HCIP (Hard Copy In Progress) signal from the Hard Copy Timing circuitry.

DACENHANCE. Figure 9-12 and Schematic 10-5 show the two gates which generate the DACENHANCE signal. When either BLACK1 or BLACK2 (from the Timing and Control Circuitry) is true, BLACK is enabled. If DENHANCE (Delayed ENHANCE) is *true*, and BLACK signal is *false*, then the AND gate sends the DACENHANCE signal.

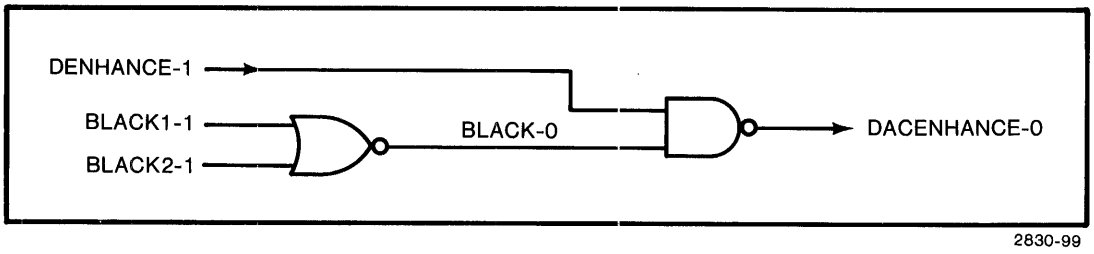


Figure 9-12. Generating the DACENHANCE Signal.

CHOPPED. Figure 9-13 shows the gates which generate the CHOPPED (Chopped Video) signal for the DACs. (These gates appear partly on Schematic 10-5 and partly on Schematic 10-2.)

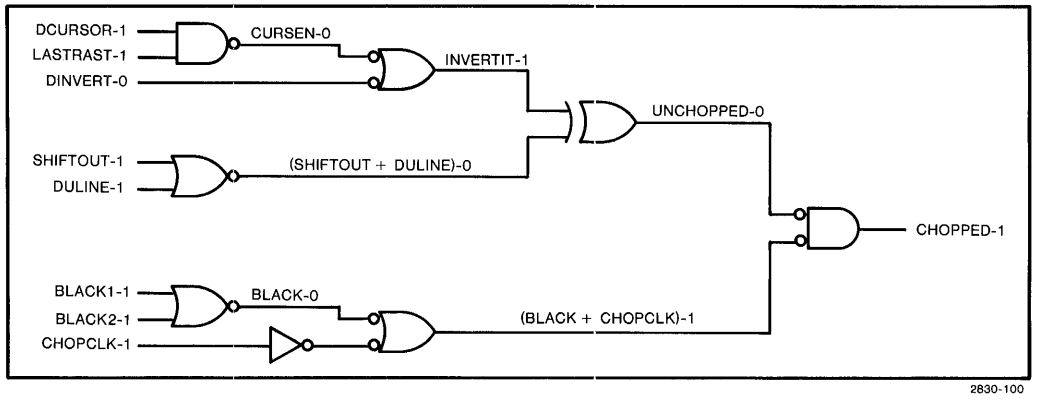


Figure 9-13. Generating the Chopped Video Signal.

An AND gate detects when DCURSOR and LASTRAST (Last Raster) are true. DCURSOR indicates that the character being scanned is to be displayed with a cursor. LASTRAST indicates that raster line 12 or 13 of that character is currently being scanned. (During an even field, raster line 12 is the last of the seven raster lines scanned for that character. During an odd field, line 13 is the last raster line scanned.) When the conditions are met, the AND gate sends the CURSEN (Cursor Enable) signal.

If the cursor is to be displayed (CURSEN true) or the character is to be displayed with the "inverted" visual attribute (DINVERT true), then an OR gate sends the INVERTIT signal. When INVERTIT is true, dots that otherwise would be turned on are turned off, and dots that otherwise would be turned off are turned on.

An OR gate detects when a dot would normally be turned on, that is, when the SHIFTOUT signal is true or an underline is to be displayed (DULINE true). The OR gate's output (SHIFTOUT + DULINE) is supplied to one input of an exclusive-OR gate, and the INVERTIT signal to the other input. The exclusive-OR gate provides the signal inversion according to the state of INVERTIT. The exclusive-OR gate output is UNCHOPPED, the "unchopped video" signal.

To prevent the vertical lines in a character from having a lower average intensity that the horizontal lines (Figure 9-14A), the video signal is "chopped" to separate adjacent dots on the same horizontal line (Figure 9-14B). The Timing and Control Circuitry provides a CHOPCLK (Chop Clock) signal for this purpose. (Also provided are BLACK1 and BLACK2 signals to blank the display.)

Two OR gates and an inverter monitor the BLACK1, BLACK2, and CHOPCLK signal lines and indicate (with a "BLACK+ CHOPCLK" signal) when any of these signals is true. An AND gate combines the UNCHOPPED video signal with BLACK+ CHOPCLK (BLACK or CHOPCLK), creating the CHOPPED video signal. CHOPPED causes the DACs to display characters as in Figure 9-14B rather than as in Figure 9-14A.

The CHOPCLK signal (used to generate the CHOPPED video) is only present when SHIFTOUT is true. Since SHIFTOUT is false when a cursor is to be displayed, this makes the cursor appear brighter than normal characters.

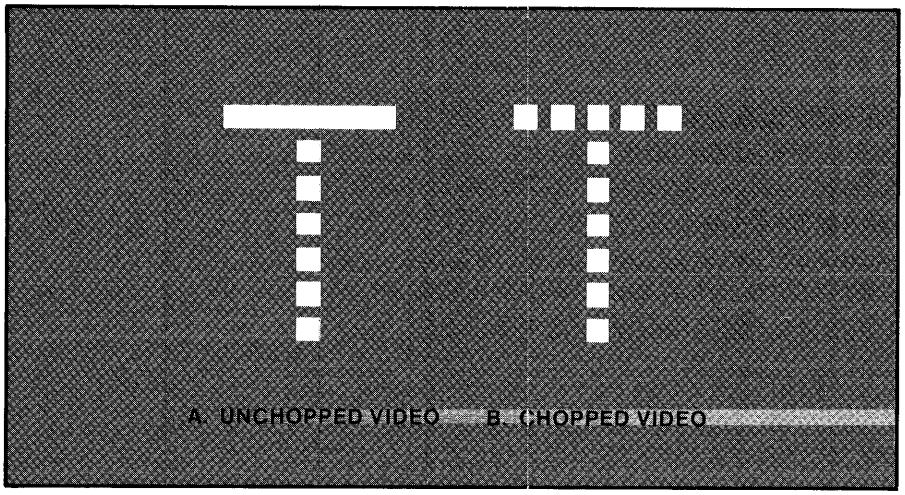


Figure 9-14. Effect of Chopping the Video Signal.

DIM. The DIM signals (DIM1 and DIM2) are used when an inverted background is being displayed. As we have seen, the cursor is brighter than normal dots of the display because normal dots are chopped (CHOPCLK is running), and the cursor dots are not. However, we want inverted backgrounds to be dimmer than the cursor, despite the fact that during all backgrounds (including inverted backgrounds), the CHOPCLK is turned off. This is accomplished by the DIM signals.

When SHIFTOUT is false (displaying the "background" of a character) and CURSEN is false (not displaying the cursor), two NAND gates send the DIM1 and DIM2 signals. If an inverted character is being displayed, DIM1 will turn down the crt beam on the internal video display, while DIM2 turns down the beam for any external video monitor. If a normal (noninverted) character is being displayed, the beam will be turned off anyway (by the CHOPPED video signal being false), so the DIM signals have no visible effect.

Internal Video DAC

(See Figure 9-15.)

The Internal Video Digital-to-Analog Converter (DAC) takes the DACBLANK, DACENHANCE, CHOPPED, and DIM2 signals and converts them to an INTERNAL VID signal for the Video Display Unit. Essentially, the circuit works by connecting different values of resistors from the base of the PNP transistor to ground. When none of the resistors is grounded, the transistor base has its maximum voltage, the INTERNAL VID output has its maximum voltage, and the crt beam is turned on to full intensity. By connecting different values of resistors to ground, the crt beam's intensity can be decreased by different amounts.

Minimum crt beam intensity occurs when DACBLANK is true. DACBLANK turns on an inverter which effectively shorts the transistor base to ground.

When CHOPPED is true, DIM2 is false. CHOPPED being true prevents the 330 and 75 ohm resistors being grounded, and DIM2 being false prevents the 680 Ω resistor from being grounded. Thus, the transistor base has its maximum voltage, and the crt beam is turned on to full intensity.

When CHOPPED goes false, we are looking at the "turned off" dots of a character's dot matrix. If these are to displayed with an enhanced background, the DACENHANCE signal will cause the 74155 $2Y_0$ output to ground the 330 ohm resistor. Otherwise, it is the 75 ohm resistor which is grounded. Since the 330 ohm resistor will not pull the transistor base voltage as close to ground as will the 75 ohm resistor, having DACENHANCE true causes the crt beam not to be completely turned off. This generates the partially lighted background for the "enhanced" visual atrribute.

DELHCCBG Flip-Flop. A type D flip-flop sends the DELHCCBG (Delayed Hard Copy Color Burst Gate) signal for the Timing and Control Circuitry. Each HCCBG pulse clears the flip-flop; the next INTERROGATE pulse sets it again. It stays set, sending DELHCCBG, until the end of the hard copy raster line, when HCCBG clears it again.

External Video DAC

The External Video DAC operates much like the Internal Video DAC. (It even uses the other half of the same 74155 integrated circuit.) However, to generate a standard composite video signal, it also incorporates the SYNC signal from the Timing and Control Circuitry.

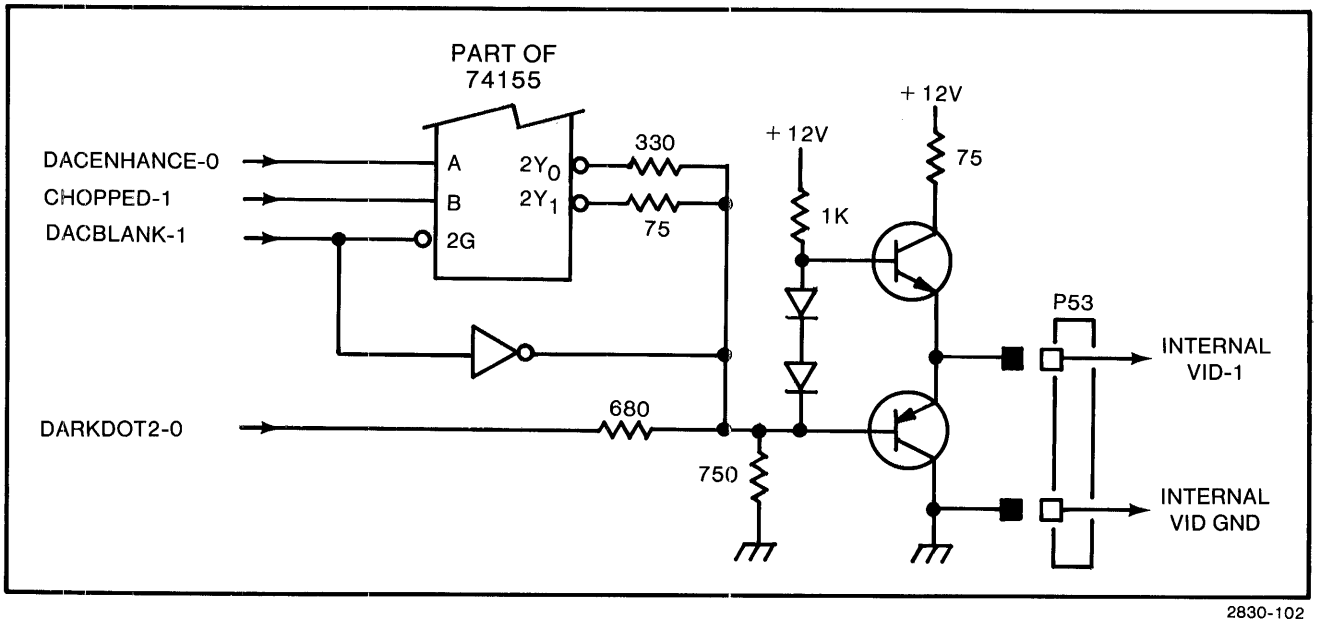


Figure 9-15. Internal Video Digital-to-Analog Converter.

Timing and Control Signals

The Timing and Control circuitry provides signals which regulate the operation of the Video Display Unit and of other parts of the Display Controller. First, we'll examine these Timing and Control signals; after that, we'll look at the circuitry which generates them.

These signals may be classified according to their repetition rate. The "vertical retrace" signals recur 60 times a second (60 Hz), at the start of each new field of the raster scan. The "text row" signals recur at about 2300 Hz, whenever the Display Controller finishes scanning one row of text and starts on the next row. "Horizontal retrace" signals recur at about 16 kHz, at each scan of a new raster line. "Character time" signals recur at about 1.566 MHz, whenever the scan of a raster line progresses from one character cell to the next. Finally, the "dot time" signals recur at about 12 or 14 MHz, as the crt beam moves from one dot of a character cell to the next.

Several of the timing signals come from an MM5320 camera sync generator integrated circuit, and get their names from signals used in television cameras.

Vertical Retrace Signals

(See Figure 9-16.)

VDRIVE, DVDRIVE, and FLDRST. The VDRIVE (Vertical Drive) signal comes from the TV camera sync generator. During normal video operation, each VDRIVE pulse produces a FLDRST (Field Reset) pulse to reset the Display Controller for the start of a new field of the display. (During a hard copy operation, FLDRST is driven by the HCRST—Hard Copy Reset—signal from the Hard Copy Circuitry.)

VDRIVE is sent to the Tracker on paired bus line PB10. VDRIVE tells Tracker when to start over at the top row of text on the screen.

DVDRIVE (Display Vertical Drive) is logically identical to VDRIVE. It is the vertical sync pulse for the Video Display Unit.

VDRIVE and DVDRIVE are active for the duration of nine horizontal lines (about 571 μs).

BLANK. The BLANK (Composite Blanking) signal comes from the TV camera sync generator. It contains a vertical blanking interval which lasts for 21 horizontal lines (about 1.333 ms). (BLANK also includes horizontal blanking signals, which recur at each horizontal retrace.)

VTIME. The VTIME-0 (Video Time) signal is false (high) during the vertical blanking interval.

PTIME. The PTIME (Picture Time) signal indicates that the Display Controller has a picture ready to be displayed. PTIME is false during the vertical retrace (VTIME) and for one more raster line. PTIME goes true, enabling the display, at the end of the first horizontal blanking interval following a vertical blanking interval.

RA0. In each character cell, the 14 rows of dots are displayed on 14 different raster lines. Each row of dots has a "raster address" in the range from 0 to 13, whose binary address appears on raster address RA0 to RA3. Because of the interlaced scan, bit RA0 is constant throughout each field of the display. During an even field, RA0 is zero; during an an odd field, it is one. Thus RA0 changes at every vertical retrace, and may be classified as a "vertical retrace" signal.

During a hard copy, the interlaced scan is not used. RA0 is then driven by HCFIELD (Hard Copy Field), from the Hard Copy Circuitry.

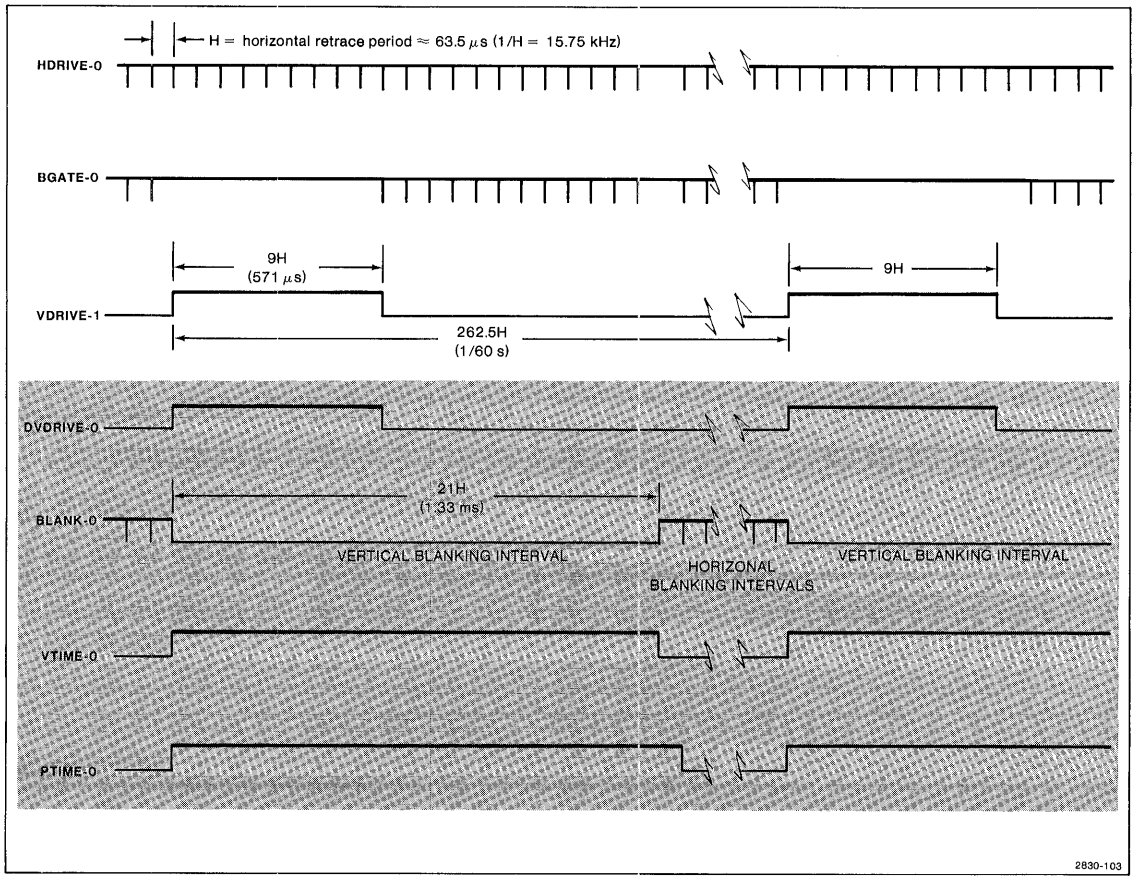


Figure 9-16. Vertical Retrace Signals.

Text Row Signals

(See Figure 9-17.)

BUFA and BUFB. The BUFA (Buffer A) and BUFB (Buffer B) signals control the two Row Buffers. When BUFA is true (low), Buffer A is being loaded with characters from the Tracker. When BUFA is false (high), Buffer A is being scanned by the Video Generator to generate the video signal. The two signals are complementary; when Buffer A is being loaded, Buffer B is being scanned. BUFA and BUFB change state when the Row Buffers are swapped.

BUFUL. The BUFUL (Buffer Full) line on the paired bus can be driven by the Display Controller or by the Tracker (on the Display Memory Board). BUFUL is false while the Tracker is sending characters to the Display Controller.

When the Tracker encounters an end-of-line marker (prior to the 80th character), it pulses BUFUL. This makes the Display Controller revert to the standard character font and the standard visual attribute for the remaining characters on that row of text.

The Tracker then continues to send characters to the Display Controller. This time, the characters are all ASCII spaces. When the rest of the Row Buffer is filled with ASCII spaces, the Display Controller asserts BUFUL. It keeps sending BUFUL until the start of the next line.

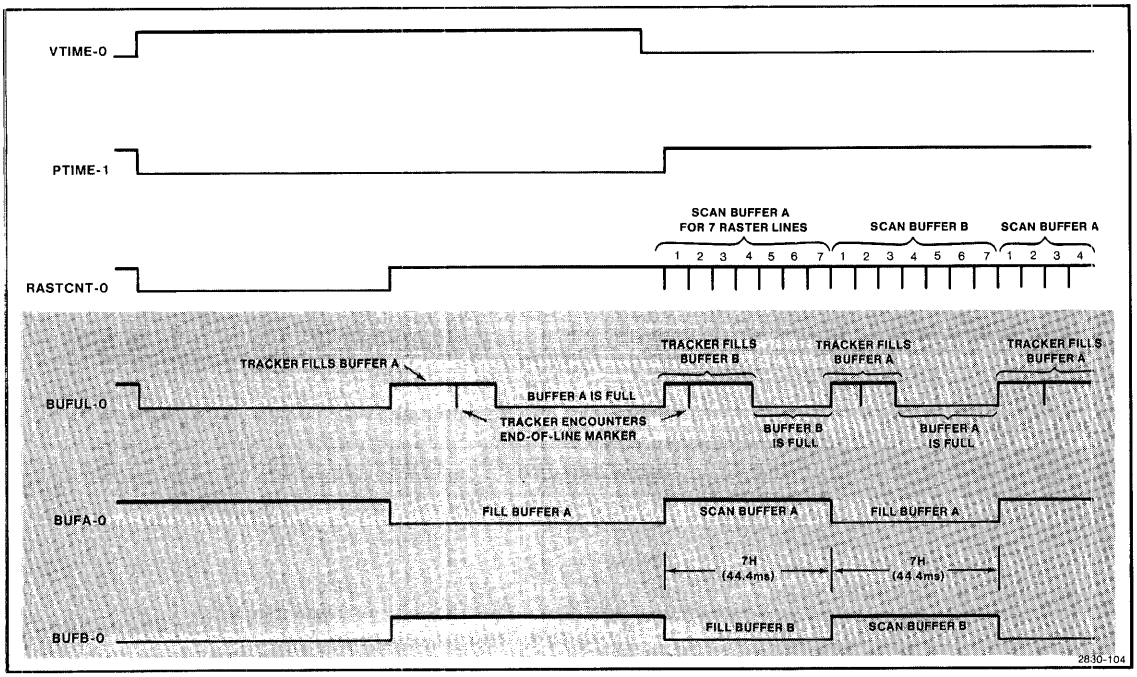


Figure 9-17. Text Row Signals.

Horizontal Retrace Signals

(See Figure 9-18.)

HDRIVE. HDRIVE (Horizontal Drive) is one of the signals provided by the camera sync generator IC. It defines the horizontal retrace period.

SYNC. SYNC is the composite sync signal from the camera sync generator. As well as horizontal sync pulses during each HDRIVE interval, SYNC includes a variety of special-purpose pulses during the vertical retrace. Those special-purpose pulses are intended for use by tv cameras; they are ignored by the 4025, and will not be discussed here.

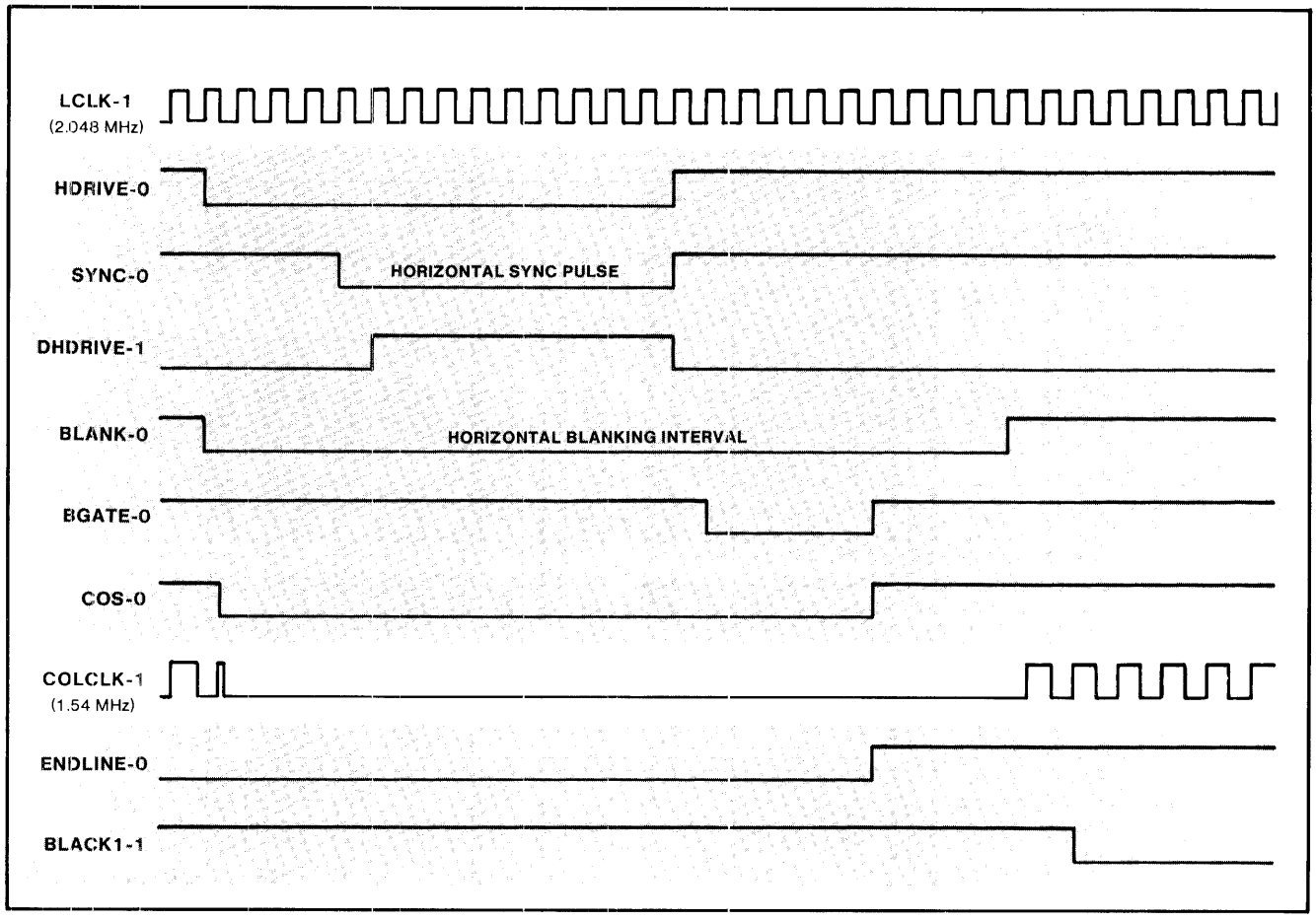


Figure 9-18. Horizontal Retrace Signals.

DHDRIVE. DHDRIVE (Display Horizontal Drive) is the logical AND of HDRIVE and SYNC. Those SYNC pulses occurring during the horizontal retrace interval generate DHDRIVE horizontal sync pulses for the Video Display Unit.

BLANK. The BLANK (Composite Blanking) signal from the camera sync generator provides a horizontal blanking pulse throughout the HDRIVE period.

BGATE. The BGATE (Color Burst Gate) signal from the camera sync generator occurs toward the end of each horizontal blanking pulse. The name of the signal comes from its use in color television cameras. In the 4025, it functions as a convenient timing signal to show that the horizontal blanking period is about to end.

COS. The COS (Clear Out Stop) pulse occurs at the first COLCLK pulse after the start of a horizontal retrace. COS flushes the cursor and underline bits from the Pipeline Delay flip-flops. It also reloads the COLCLK counter, pulling the COLCLK line low. (COLCLK is disabled while COS is active.)

ENDLINE. The ENDLINE (End of Line) signal begins one COLCLK pulse after the Row Buffer reaches the end of its 80 characters. In Figure 9-19, this event is marked as time t_2. The ENDLINE pulse ends at time t_4, at the end of the BGATE pulse preceding the next raster line.

BLACK1. The BLACK1 signal blanks the display between the last character on one raster line and the first character on the next raster line. BLACK1 begins one COLCLK pulse after ENDLINE goes true (time t_3 in Figure 9-19). The BLACK1 pulse ends when the new raster line's first character is ready to be shifted out of the Shift Register. (This is at time t_6 in Figure 9-19.)

RA0-RA3. The four raster address lines RA0 to RA3 designate which row of a character's dot matrix is being scanned. The least significant bit RA0 is constant throughout a field of the raster scan. However, the most significant three bits RA1 to RA3 count from 0 to 6 as the scan advances from line to line. Thus, during an even scan, the total raster address counts as follows: 0, 2, 4, 6, 8, 10, 12. During an odd scan, it counts 1, 3, 5, 7, 9, 11, 13. The Timing and Control Circuitry includes a Raster Counter which increments the RA1 to RA3 bits as each new raster line begins.

RASTCNT. The RASTCNT clock advances raster address lines RA1 to RA3. During normal video operations, a RASTCNT pulse occurs with every BGATE pulse shortly before the end of the horizontal blanking period. During a hard copy, however, RASTCNT is driven by the HCRSTCNT (Hard Copy Raster Count) signal from the Hard Copy Circuitry.

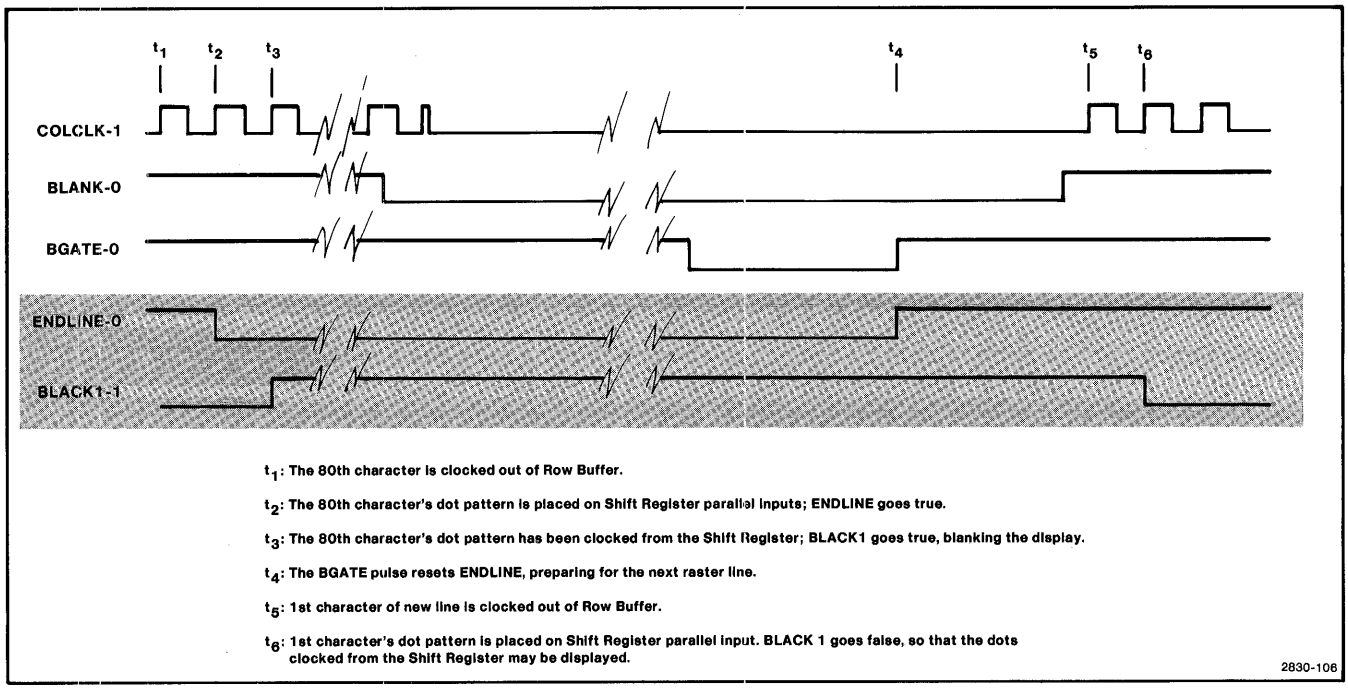


Figure 9-19. ENDLINE and BLACK1 Signals.

Character Time Signals

(See Figure 9-20.)

The "character time" signals recur 80 times in each horizontal raster line, as the Display Controller advances from character to character in scanning the Row Buffer.

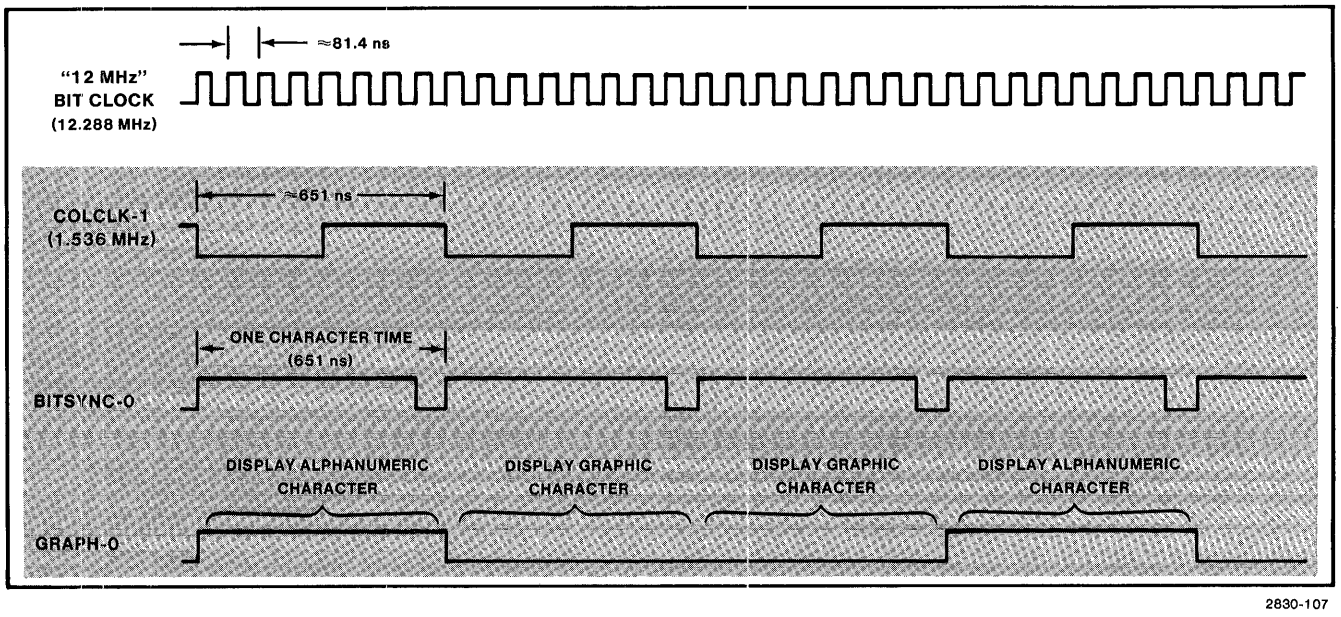


Figure 9-20. Character Time Signals.

COLCLK. The COLCLK (Column Clock) signal regulates the flow of data through the video pipeline. It clocks the counter for whichever Row Buffer is being scanned, causing it to advance to the next character in the buffer. COLCLK clocks the buffer outputs (character and font address bits, and visual attribute bits), and the raster address bits RA0-RA3 onto the Character Set Memory address lines. COLCLK clocks the dot pattern data out of the Character Set Memory.

BITSYNC. The BITSYNC (Bit Syncronizing) signal occurs at the last eighth of each character time. BITSYNC loads the Shift Register with the output of the Character Set Memory. It is also used to keep the 14 MHz "scrunch clock" in synchronization with the characters coming through the video pipeline (in synchronization with COLCLK).

GRAPH. Another "character time" signal is GRAPH, which is true when the character being scanned is a graphic or rulings character, rather than an alphanumeric character. GRAPH comes from the Character Set Memory and tells the Display Controller when to use the 14 Mhz "scrunch clock" for clocking data bits out of the Shift Register. When GRAPH is false, an alphanumeric character is being displayed, and the scrunch clock is used. When GRAPH is true, a graphic or rulings character is being displayed, and the 12 MHz bit clock is used.

Dot Time Signals

(See Figure 9-21.)

SHIFTCLK, 12 MHz, and 14 MHz. The SHIFTCLK (Shift Clock) signal clocks dot pattern bits out of the shift register.

If the character being clocked out of the shift register is a graphics or ruling character, then the 12 MHz bit clock drives the SHIFTCLK line. This clock, at 12.288 MHz, runs at eight times the COLCLK frequency. Thus, in each raster line of a graphics or rulings character, the eight dots are evenly spaced across the character cell.

If the character is an alphanumeric character, the 14 MHz "scrunch clock" is used. This clock runs at about nine times the COLCLK frequency. Thus, in each raster line of an alphanumeric character, the eight dots are "scrunched" into 8/9 the width of the character cell. This gives extra space between adjacent alphanumeric characters. The BITSYNC signal keeps the 14 MHz clock synchronized with the COLCLK pulses.

During a hard copy operation, SHIFTCLK is driven by the HCBITCLK signal from the Hard Copy Timing circuitry. That signal is described later in this section, under "Making Hard Copies."

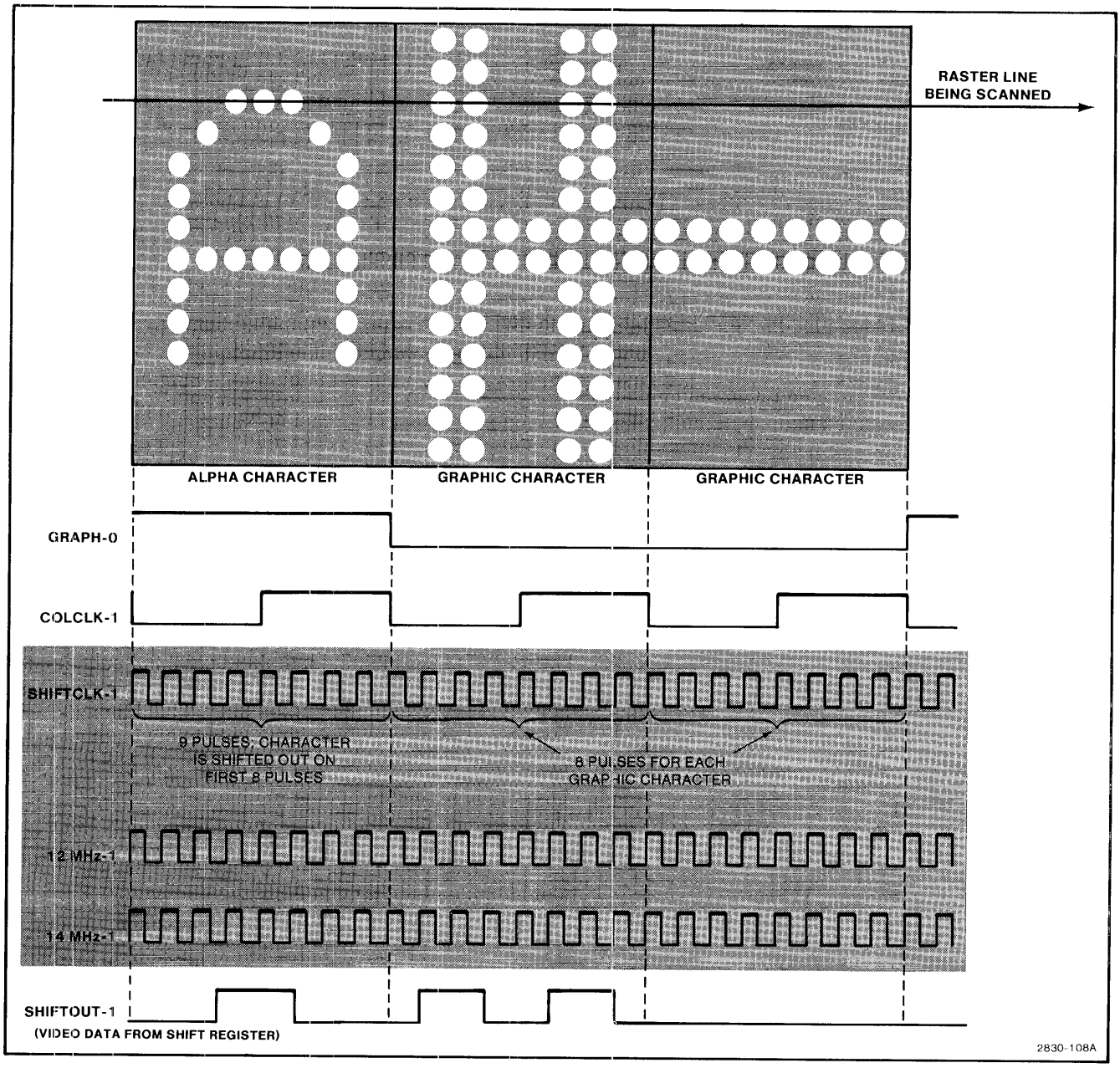


Figure 9-21. Dot Time Signals.

CHOPCLK. The CHOPCLK signal, like SHIFTCLK, is derived from the 12 MHz or 14 MHz bit clock according to whether the character being displayed is a graphics character or an alphanumeric character. Because of propagation delays through the Shift Register and other gates, CHOPCLK arrives at the DAC Control circuit slightly out of phase with the video data from the Shift Register. CHOPCLK is used by the DAC Control circuitry to "chop" the video signal into separate dots. (This was described earlier, under "DAC Control Circuitry.")

Timing and Control Circuitry

Now that we've described the timing and control signals, let's look at the circuitry which generates them. The Timing and Control Circuitry is comprised of four blocks: Video Timing, Timing Selector and Generator, Raster Counter, and Buffer Control. In addition, several timing signals used during hard copy operations come from the Hard Copy Circuitry, described later in this section.

Video Timing Circuitry

(See Figure 9-22.)

The Video Timing Circuitry generates signals used during normal video operation, but not during hard copies. Most of these signals come from a tv camera sync generator: HDRIVE, VDRIVE, BLANK, SYNC, etc. The 12 MHz bit clock, however, is generated by counting down the terminal's HCLK (18.432 MHz) clock.

Sync Generator. The MM5320 tv sync generator is clocked from LCLK, the terminal's 2.048 MHz clock. Type D flip-flops in a 74LS174 keep the sync generator outputs synchronized with LCLK. A 74LS167 buffers HDRIVE, VDRIVE, BLANK, and BGATE.

ASTOP. The ASTOP (Asynchronous Stop) signal is the logical OR of HDRIVE and VDRIVE. It is delayed one COLCLK pulse by a flip-flop in the Pipeline Delay Circuitry, becoming the SSTOP (Synchronous Stop) signal for the Timing Selector and Generator. (During normal video operation, SSTOP drives the STOP line.)

12 MHz Bit Clock. The 12 MHz bit clock is generated by a divide-by-3/2 (multiply-by-2/3) circuit using two JK flip-flops, two exclusive-OR gates, and an AND gate. For every three HCLK pulses, this circuit provides two pulses out. Since HCLK runs at 18.432 MHz, the circuit's output is at 12.288 MHz.

Timing Selector and Generator

The Timing Selector and Generator includes data selectors (to select between the Video Timing Circuitry and the Hard Copy Circuitry as the source of signals), some flip-flops and gates (to generate specific output signals), two counters (for generating the COLCLK signal and the raster address bits RA1 to RA3), and an oscillator (to generate the 14 MHz "scrunch clock").

Data Selectors. The Data Selectors include two 74LS257's (Schematic 10-1) and a 74LS153 (Schematic 10-5). They are steered by the HCBUSY signal, which indicates whether a hard copy is in progress.

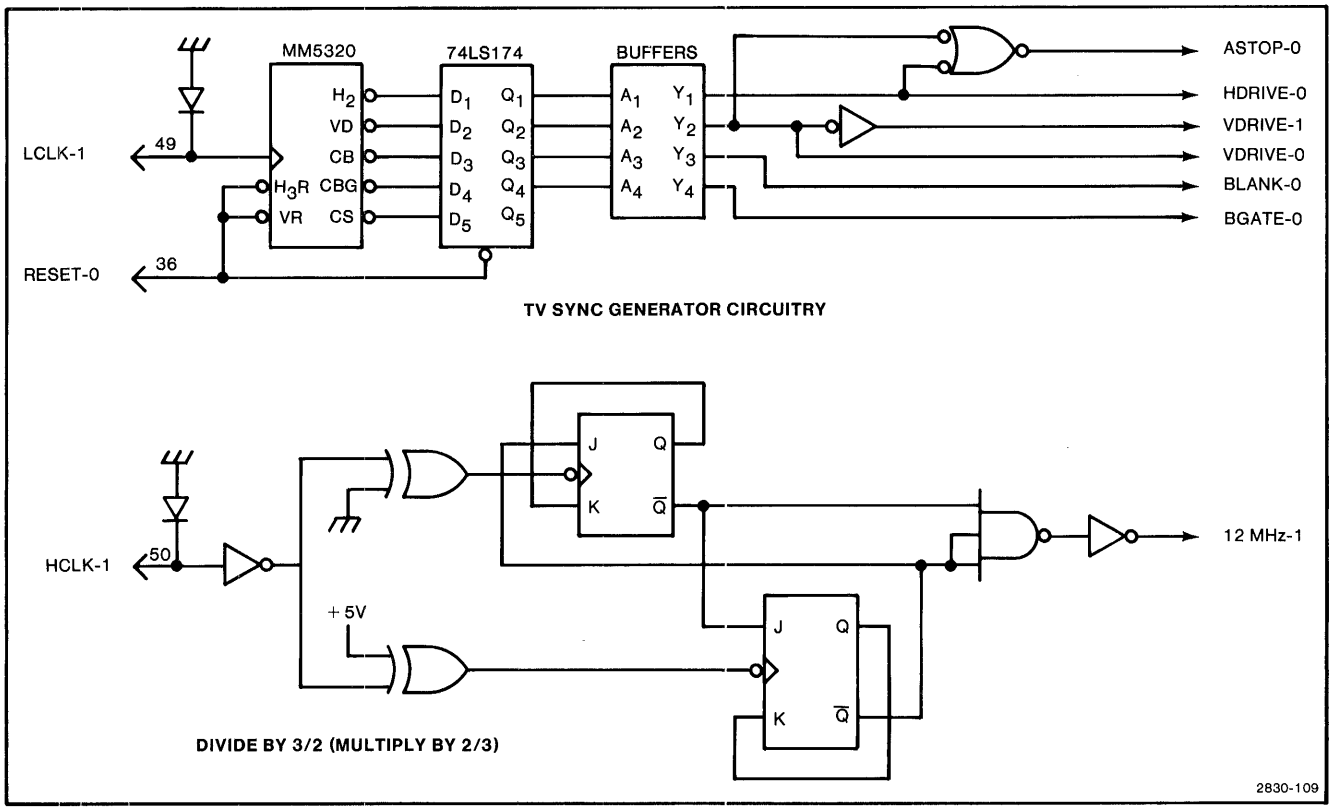


Figure 9-22. Video Timing Circuitry.

RASTCNT. The raster address lines RA1 to RA3 are driven by the Raster Address Counter (Schematic 10-3). This counter is clocked by RASTCNT, from an OR gate driven by two of the data selector outputs (Schematic 10-1). A RASTCNT pulse occurs at the end of each vertical retrace (VDRIVE pulse) or the end of the hard copy (HCRST pulse). During normal video operation (HCBUSY false and VTIME, Video Time, true), RASTCNT pulses occur at the end of each raster line (at the BGATE pulse). During hard copies, RASTCNT pulses occur at each HCRSTCNT (Hard Copy Raster Count) pulse from the Hard Copy Circuitry.

RA0. Raster address bit RA0 specifies whether the field being displayed is an even field or an odd field. During hard copy operations, this signal follows the HCFIELD (Hard Copy Field) signal from the Hard Copy Circuitry.

During normal video operation, RA0 is generated in a different manner.

Recall that a full frame of the raster scan has 525 lines, an odd number. VDRIVE pulses occur at the end of each field (half-frame): every 525 lines. Because of this, the VDRIVE pulses cannot always occur at the end of a raster line.

Instead, the VDRIVE pulse at the start of an odd field occurs in the middle of a raster line. That is, the VDRIVE pulse that begins an even field occurs at the end of a raster line (when ENDLINE is true), but the VDRIVE pulse that begins an odd field occurs in the middle of a raster line (when ENDLINE is false).

To generate the RA0 signal, a flip-flop detects whether the VDRIVE pulse occurs when ENDLINE is true (start of even field) or false (start of odd field.) A type D flip-flop is used, with ENDLINE going to its D input, and VDRIVE clocking it. The flip-flop output is the RA0 signal: a zero during even fields, and a one during odd fields.

BITCLK and COLCLK. The COLCLK counter (Schematic 10-1) generates the COLCLK signal by dividing down a BITCLK signal. During normal video, BITCLK is the 12.288 MHz clock from the Video Timing Circuitry. During a hard copy, BITCLK is the HCBITCLK (Hard Copy Bit Clock) signal.

During normal video, the COLCLK counter is cleared by the COS (Clear Out Stop) pulse. (COS occurs two COLCLK pulses into the horizontal blanking interval.) COLCLK remains disabled until the end of the blanking interval. The COLCLK counter starts each raster line at the count of zero.

The counter counts from 0 to 15 during each two character times. COLCLK is low on the counts of 0 through 3 and 8 through 11; it is high on the counts of 4 through 7 and 12 through 15. At the counts of 7 and 15, a NAND gate drives the BITSYNC line low. (BITSYNC keeps the 14 MHz "scrunch clock" synchronized.)

Scrunch Clock. A free-running oscillator made from two gates drives the 14 MHz clock. Two NAND gates force the clock high just before the start of a new character, when BITCLK and BITSYNC are both low; this keeps the scrunch clock in sync with the COLCLK pulses.

FLDRST. The FLDRST (Field Reset) pulse is driven by VDRIVE during normal video, and by HCRST (Hard Copy Reset) signal during a hard copy.

ASTOP, SSTOP, STOP, and COS. At the end of each raster line (or at the middle of the last raster line in an even field, when the VDRIVE pulse occurs), a gate in the Video Timing Circuitry sends the ASTOP (Asynchronous Stop) signal. This goes to the SSTOP flip-flop in the Pipeline Delay Circuitry. The flip-flop output is the SSTOP (Synchronous Stop) signal. During normal video operation, SSTOP drives the STOP line. (During a hard copy, STOP is driven by DELHCCBG, Delayed Hard Copy Color Burst Gate.) STOP clears the COS flip-flop, sending the COS (Clear Out Stop) signal. COS, in turn, clears several type D flip-flops in the Pipeline Delay circuitry. (Since one of these flip-flops drives the SSTOP line, the STOP pulse is brief.) Other flip-flops cleared are those driving the DFONT0, DCURSOR, and DULINE lines. This flushes old cursor and underline information out of the video pipeline.

STOP also drives an OR gate (Schematic 10-3), which in turn sends the CLRLINLTCH (Clear Line Latch) signal. (CLRLINLTCH clears the ENDLINE flip-flop discussed next.)

ENDLINE. A type D flip-flop sends ENDLINE (End of Line). The flip-flop sends ENDLINE when cleared by the CLRLINLTCH (Clear Line Latch) signal. (CLRLINLTCH occurs at the end of a raster line, or when the BUFAFUL or BUFBFUL signal comes from the Row Buffer being scanned.)The flop continues sending ENDLINE until it is set again. The flop will not set during a vertical retrace interval (VTIME must be true). Provided VTIME is false, the flop sets on the BGATE pulse preceding the next raster line. Thus, ENDLINE is true from the STOP pulse at the end of one raster line to the BGATE pulse preceding the next raster line.

ENDLINE serves two functions. First, as described above, it is sampled at each VDRIVE pulse to determine whether the coming field is even or odd (whether RA0 is to be zero or one). Secondly, the Buffer Control Circuitry passes ENDLINE to the buffer being scanned as the CNTACLR or CNTBCLR signal. This resets the Row Buffer's counter start at the first character in the row of text.

BLACK1. The BLACK1 signal suppresses the video display during horizontal retraces. It is sent by a type D flip-flop. The flip-flop is set, sending BLACK1, by the STOP signal at the end of a raster line. At the start of the new raster line, the COLCLK counter provides a ripple carry output together with the second COLCLK pulse in that line; the ripple carry clears the BLACK1 flip-flop. Thus BLACK1 is sent from the STOP pulse at the end of one raster line until the second COLCLK pulse in the next raster line.

SHIFTCLK and CHOPCLK. A 74S153 data selector (Schematic 10-5) sends the SHIFTCLK and CHOPCLK signals. During normal video operation, the signals are identical: they are driven by the 12 MHz bit clock if GRAPH is true, indicating a graphics character, and by the 14 MHz scrunch clock if GRAPH is false. During hard copies, CHOPCLK is turned off, and the SHIFTCLK is driven by HCBITCLK (Hard Copy Bit Clock).

Raster Counter

(See Schematic 10-3.)

The Raster Counter provides the following signals: RA1 to RA3, LASTRAST, and BLACK2.

The counter, a 74LS161, is loaded synchronously on the next RASTCNT pulse after a vertical retrace (VTIME signal false), or after its RA1-RA3 outputs have reached the count of seven (LASTRAST true).

RA1 to RA3. The counter's least significant bits drive the RA1 to RA3 raster address lines. These, together with RA0, specify which raster line is being scanned.

LASTRAST. Two gates decode the raster address bits RA1 to RA3 and send the LASTRAST (Last Raster) signal when they reach the count of six. LASTRAST loads the counter again at the next RASTCNT pulse; it also tells the DAC Control Circuitry when it is OK to display the cursor.

BLACK2. The Q_D counter output is used, with an external type D flip-flop, to generate the BLACK2 signal.

When a hard copy is initiated, there is a minimum delay before signals may be sent to the hard copy unit. (This delay gives the hard copy unit's anit-burn circuit time to turn off.) The delay is accomplished by the external flip-flop and the "part D" flip-flop in the 74LS161. During this delay, the latter lip-flop sends the BLACK2 signal, suppressing any signals which otherwise might be sent to the hard copy unit.

(This is described later in this section, in connection with the Hard Copy Circuitry.)

Buffer Control

(See Schematic 10-3.)

The Buffer Control Circuitry provides signals to control the Row Buffers: BUFASEL, BUFBSEL, CNTACLK, CNTBCLK, CNTACLR, CNTBCLR, BUFA, BUFB. It also provides the BUFUL signal for the Display List Decoder and the Display Memory Board's Tracker circuitry.

BUFA and BUFB. The BUFA and BUFB signals are logical complements, and are driven by the same flip-flop. BUFA is true when Row Buffer A is being loaded and Row Buffer B is being scanned; BUFB is true when Buffer B is being loaded and Buffer A is being scanned.

Data Selectors. Two 74LS157 data selectors are steered by the BUFB signal. These data selectors drive the BUFASEL, BUFBSEL, CNTACLK, CNTBCLK, CNTACLR, and CNTBCLR control lines for the Row Buffers.

BUFASEL is the chip select line for Buffer A's RAMs. When Buffer A is being scanned, BUFASEL is always true (TTL low). When Buffer A is being loaded, BUFASEL is true only during the CHARCLK pulses (which clock characters into the RAMs). Likewise, BUFBSEL is true when Buffer B is being scanned, but true only during CHARCLK pulses when Buffer B is being loaded from the Tracker.

CNTACLK is the clock for Buffer A's character position counter. When Buffer A is being loaded, CNTACLK is driven by the CHARCLK signal from the Display List Decoder. When Buffer A is being scanned, CNTACLK is driven by the COLCLK signal which regulates the flow of data in the video pipeline. Likewise, CNTBCLK is driven by CHARCLK or COLCLK according to whether Buffer B is being loaded or scanned.

CNTACLR is used to reset Buffer A's character position counter. When Buffer A is being scanned, CNTACLR occurs with each ENDLINE pulse. When Buffer A is being loaded, CNTACLR occurs when the Cursor Counter is being loaded. Likewise, CNTBCLR is driven by ENDLINE pulses when Buffer B is being scanned, and by the "load Cursor Counter" pulse when Buffer B is being loaded.

BUFUL. Whether or not the Display Controller sends BUFUL is controlled by a flip-flop, the Buffer Full Latch. The flip-flop is set, sending BUFUL, when the Row Buffer (being loaded) reaches its last character (BUFAFUL or BUFBFUL from the Row Buffers). The flip-flop may also be set by FLDRST, which marks the end of a video or hard copy scan. The BUFUL flip-flop is cleared by the first RASTCNT pulse following a LASTRAST signal from the Raster Counter. That is, the flip-flop is cleared when the Row Buffers are next swapped.

The BUFUL line can also be driven by the Tracker (on the Display Memory Board). In normal operation, Tracker briefly pulses BUFUL when it encounters an end-of-line marker. Later, when the remainder of the buffer is filled with ASCII spaces, the BUFUL flip-flop goes set and holds BUFUL true until the Row Buffers are swapped.

Making Hard Copies

Hard Copy Procedure

Refer to Figure 9-23, which shows the interconnections between circuit blocks used to make hard copies. Hard copies are made as follows:

1. An HCOPY command is received (from keyboard or host computer), or the COPY switch is pressed on the 4631 Hard Copy Unit.

2. In either case, the Mother Board's HC (Hard Copy) line goes low. (This is the same as paired bus line PB11.)

   If the HCOPY command was used, the Processor writes into a status register (address X'0810') on the Display Memory Board. One of that register's bits then grounds the HC line.

   If the 4631's COPY switch was used, the 4631 grounds its READ output, which is connected by a diode to the HC line.

3. The following then occurs:

   - The 4025 sends a COPY signal to the 4631. The 4631 starts copying: it sends the READ signal (if it was not already doing so) and starts sending INTERROGATE pulses.

   - The 4025 triggers a one-shot, grounding HC until the 4631 can respond. (By the time the one-shot times out, the task of grounding HC will have been taken over by the READ signal from the 4631.)

4. The 4631 Hard Copy Unit sends a string of INTERROGATE pulses. These step the 4025 Hard Copy Circuitry across the hard copy raster lines. For each INTERROGATE pulse, the 4025 drives the 4631's TARSIG input high or low, according to whether the current hard copy dot is to be dark or light.

   (The TARSIG signal from the 4025 emulates the "target signal" from a TEKTRONIX 4010 series terminal.)

5. When the 4631 completes its copy cycle, it releases the READ line. This tells the 4025 that the hard copy is done.

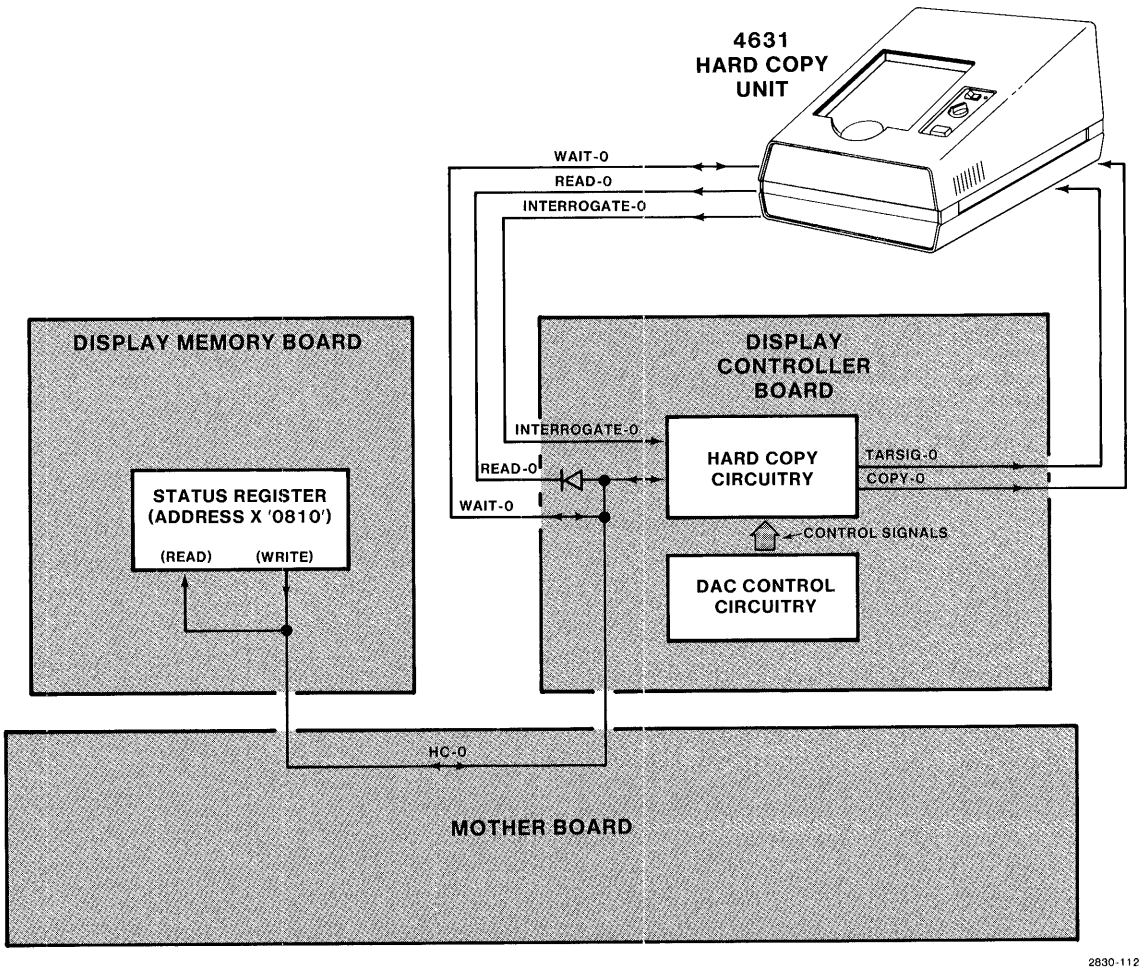


Figure 9-23. Hard Copy Interconnections.

TARSIG pulses for the 4631 Hard Copy Unit are generated in much the same way as is the VIDEO signal for the internal video display:

- The Tracker loads the Row Buffers with characters from the display list. While one buffer is being loaded, the other is being scanned to generate the hard copy.
- Character and font data (Row Buffer outputs CA0-CA6 and FA0-FA4), together with raster address data (RA0-RA3 from the Timing and Control Circuitry) goes to the Character Set Memory's address inputs. That memory responds with dot pattern information on its data outputs.
- The Shift Register converts the parallel data from the Character Set Memory to a serial bit stream. That stream of bits becomes the TARSIG signal for the hard copy unit.

However, there are some important differences between the 4631 hard copy unit and the terminal's internal video display:

- The hard copy unit operates at a much slower rate than does the video display. INTERROGATE pulses from the hard copy unit recur at a rate of about 500 kHz, while the video analog, the SHIFTCLK pulses driving the Shift Register, recur at about 12 or 14 MHz. Consequently, during hard copy operations, all the display timing signals come from separate hard copy timing circuitry. The hard copy timing signals are derived from the INTERROGATE pulses.

- The hard copy unit advances from one raster line of the display to the next by mechanically moving photographic paper past its crt. It cannot use an interlaced scan, but must scan the lines in sequence.

- The hard copy unit has much finer resolution than the video display, and the hard copy raster lines are much closer together. Because of this, each video dot appears as six "hard copy dots," as in Figure 9-24. The six hard copy dots comprise a rectangle with three dots on each of two adjacent hard copy raster lines.

  Thus, to generate the TARSIG signal, each video raster line is scanned twice. During each of the two scans, three successive INTERROGATE pulses query the state of each video dot.

- The 4631 Hard Copy Unit expects a TARSIG signal which is either on or off. Thus, one cannot "turn TARSIG one-third on" to display a character with an enhanced background.

  Instead, when displaying an "enhanced background" character, one-third of the background "hard copy" dots are turned on.

Hard Copy Circuitry

The Hard Copy Circuitry (Figure 9-25 and Schematic 10-2) includes these blocks: Hard Copy Starter, COPY Line Driver, HCIP flip-flop, Interrogate Pulse Conditioner, Retrace Detector, Hard Copy Raster Counter, Interrogate Counter, DELHCCBG Flip-Flop, and TARSIG Generator.

Some of the circuitry in the Raster Counter is also used during hard copies. After describing the Hard Copy Circuitry on Schematic page 10-2, we'll return to look at the Raster Counter (Schematic 10-3).

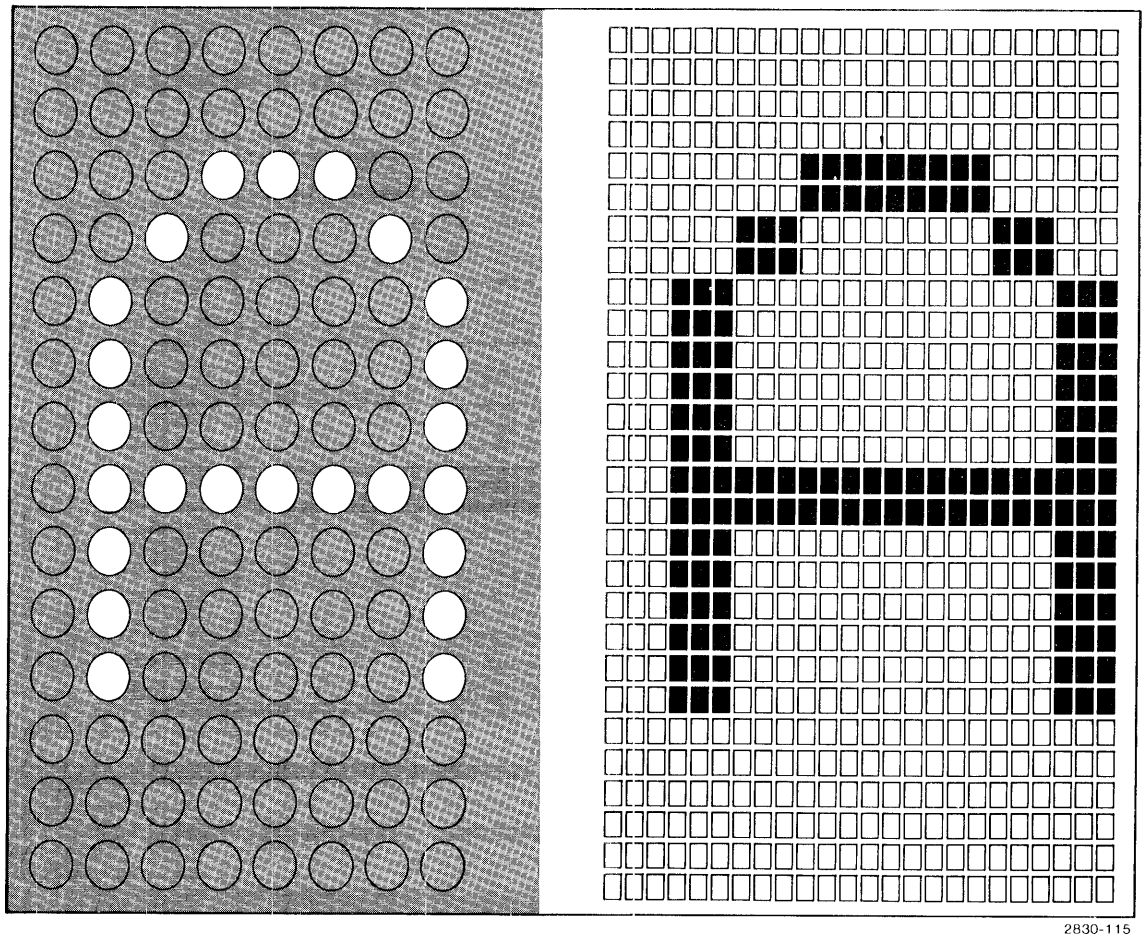


Figure 9-24. Video and Hard Copy Representations of a Character.

Hard Copy Starter. The Hard Copy Starter is a one-shot multibrator. When a hard copy operation is started. the HC line is driven low (by circuitry on the Display Memory Board, or by a READ signal from the 4631 Hard Copy Unit). This negative transition is conditioned by a Schmidt trigger inverter and used to trigger the one-shot. So long as the one-shot is set, it sends the HCRST (Hard Copy Reset) signal. This clears other circuits in the Hard Copy Circuitry and the rest of the Display Controller; it also causes the COPY Line Driver to send a COPY signal to the 4631 Hard Copy Unit.

The Hard Copy Unit responds by starting to make a hard copy and pulling the HC line low. Thus, by the time the one-shot times out, the task of holding HC low will have been taken over by the Hard Copy Unit.

COPY Line Driver. The COPY Line Driver buffers the HCRST pulse and drives the COPY line to the Hard Copy Unit.

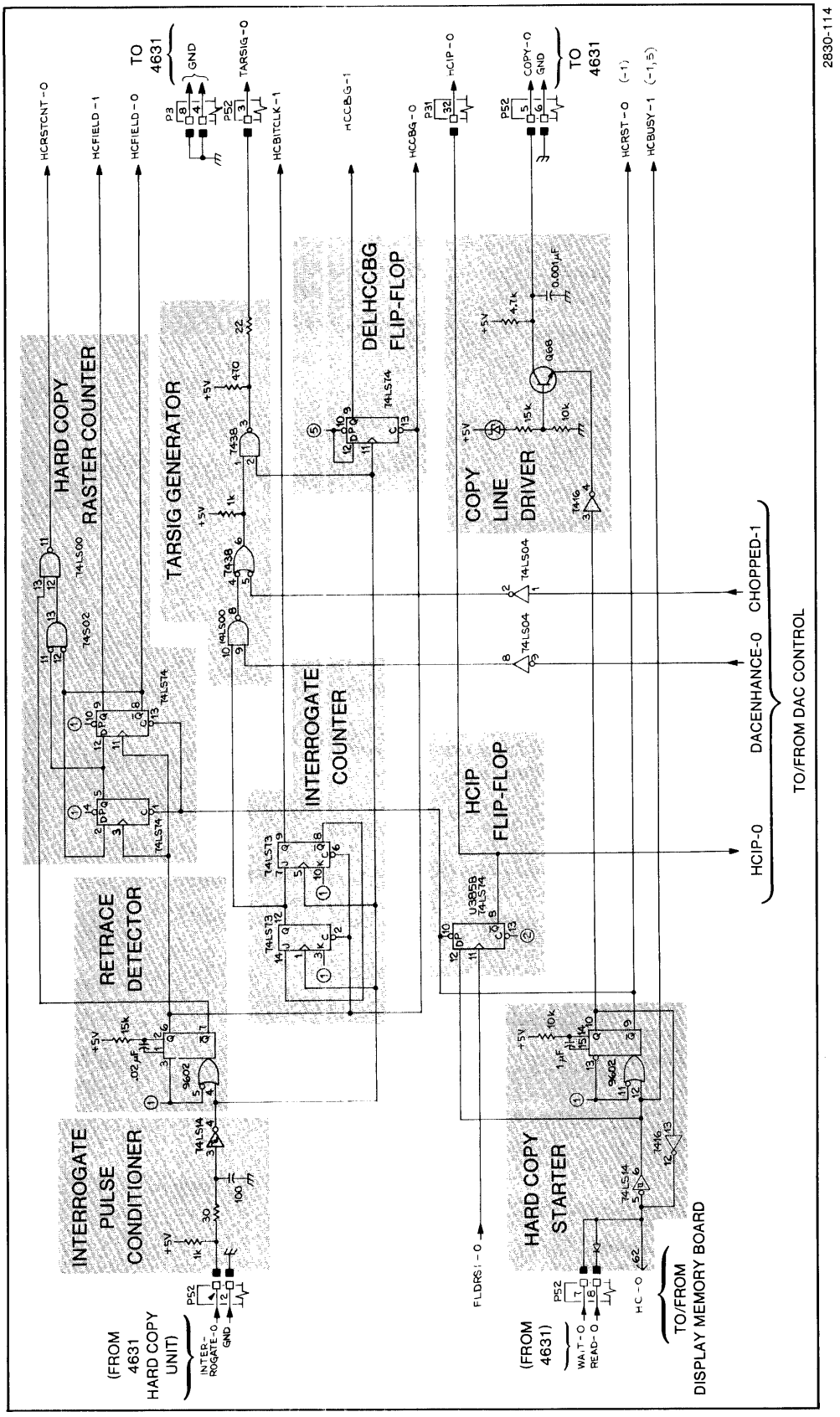


Figure 9-25. Hard Copy Circuitry.

HCIP Flip-Flop. The HCIP flip-flop drives the HCIP (Hard Copy In Progress) line. The flip-flop is set by the HCRST pulse at the start of a hard copy. It is cleared by the FLDRST (Field Reset) signal from the Timing and Control circuitry, provided the hard copy unit is finished (HC line has gone high).

Interrogate Pulse Conditioner. The Interrogate Pulse Conditioner consists of a resistor-capacitor network and a Schmidt trigger gate. The R-C network filters out random noise on the INTERROGATE line. The Schmidt Trigger gate "cleans up" the INTERROGATE pulses and prevents double-triggering on noisy pulses.

Retrace Detector. During each hard copy raster line, the Retrace Detector one-shot is repeatedly triggered by INTERROGATE pulses from the hard copy unit. As long as those pulses keep coming, the one-shot never gets a chance to time out.

However, at the end of each raster line, there is short period when no INTERROGATE pulses occur. When this happens, the one-shot times out and sends the HCCBG (Hard Copy Color Burst Gate) signal. (This signal gets its name because it replaces the BGATE signal during a hard copy.) BGATE and HCCBG are both used as timing signals to mark the end of a raster line. BGATE occurs after a video raster line, just before the next video line, while HCCBG occurs after a hard copy raster line, shortly before the next hard copy line.)

HCCBG is the clock for the Hard Copy Raster Counter.

Hard Copy Raster Counter. The Hard Copy Raster Counter generates the HCFIELD (Hard Copy Field) and HCRSTCNT (Hard Copy Raster Count) signals.

During hard copy operations, HCFIELD (Hard Copy Field) is used by the Timing Selector and Generator to drive raster address bit RA0. HCFIELD changes on every second HCCBG pulse. This ensures that each video raster line is scanned twice, producing two hard copy raster lines.

The HCRSTCNT (Hard Copy Raster Count) signal is sent after every second hard copy raster line to increment the Raster Counter. When each video raster line has been scanned twice (to generate two hard copy raster lines), the HCRSTCNT signal clocks the Raster Counter, causing it to increment raster address lines RA1-RA3.

The Hard Copy Raster Counter consists of two type D flip-flops, connected as a divide-by-two ring counter to generate the HCFIELD signal. Two gates compare the output of that counter with the ENDHCRAST signal, and generate one HCRSTCNT pulse after each two hard copy raster lines.

Interrogate Counter. As shown in Figure 9-24, each video dot is mapped into six hard copy dots, three on each of two adjacent hard copy raster lines. Thus, on each of those two hard copy raster lines, it takes three INTERROGATE pulses to advance from one video dot to the next. The Interrogate Counter accomplishes this: its inputs are INTERROGATE pulses, and its primary output is the HCBITCLK (Hard Copy Bit Clock), which is used to advance the Shift Register from one video dot to the next.

The Interrogate Counter consists of two J-K flip-flops, connected in a divide-by-three configuration.

DELHCCBG Flip-Flop. A type D flip-flop sends the DELHCCBG (Delayed Hard Copy Color Burst Gate) signal for the Timing and Control Circuitry. Each HCCBG pulse clears the flip-flop; the next INTERROGATE pulse sets it again. It stays set, sending DELHCCBG, DELHCCBG, until the end of the hard copy raster line, when HCCBG clears it again.

TARSIG Generator. The TARSIG Generator corresponds to the Digital-to-Analog Converters in the Video Generator. It takes the DACENHANCE and CHOPPED signals from the DAC Control circuitry, and uses them to generate the TARSIG signal for the hard copy unit.

TARSIG is true (at ground potential) whenever either of these conditions are met:

- A lighted dot of the video display is to be copied (the CHOPPED video signal is true).

- An "enhanced background" dot of the video display is to be copied *and* the Interrogate Counter is on the middle one of the three INTERROGATE pulses for that video dot.

Figure 9-26 shows how a character with enhanced background is copied on the 4631 Hard Copy Unit. Each dot of the video display corresponds to a rectangle of six dots on the hard copy unit. If the the video dot is turned full on, all six of the hard copy dots are turned on. If the video dot is an "enhanced background" dot, the middle two of the six hard copy dots are turned on.

Special Circuitry in the Raster Counter for Use in Hard Copies. Refer to Schematic 10-3, which shows the Raster Counter in its upper right corner. The "D" flip-flop in the Raster Counter's 75LS161 binary counter plays a special role, as does the external type D counter which drives the 74LS161 "D" input.

It takes a little while for the "anti-burn" circuit in the 4631 Hard Copy Unit to turn off. Consequently, the 4025 must ignore the first few INTERROGATE pulses until the 4631 is ready to record on its paper. This is accomplished by the external type D flip-flop (U361A), the "D" part of the 74LS161 (U255), and the gate (U261A) that drives the SWAP line for the Buffer Control circuitry.

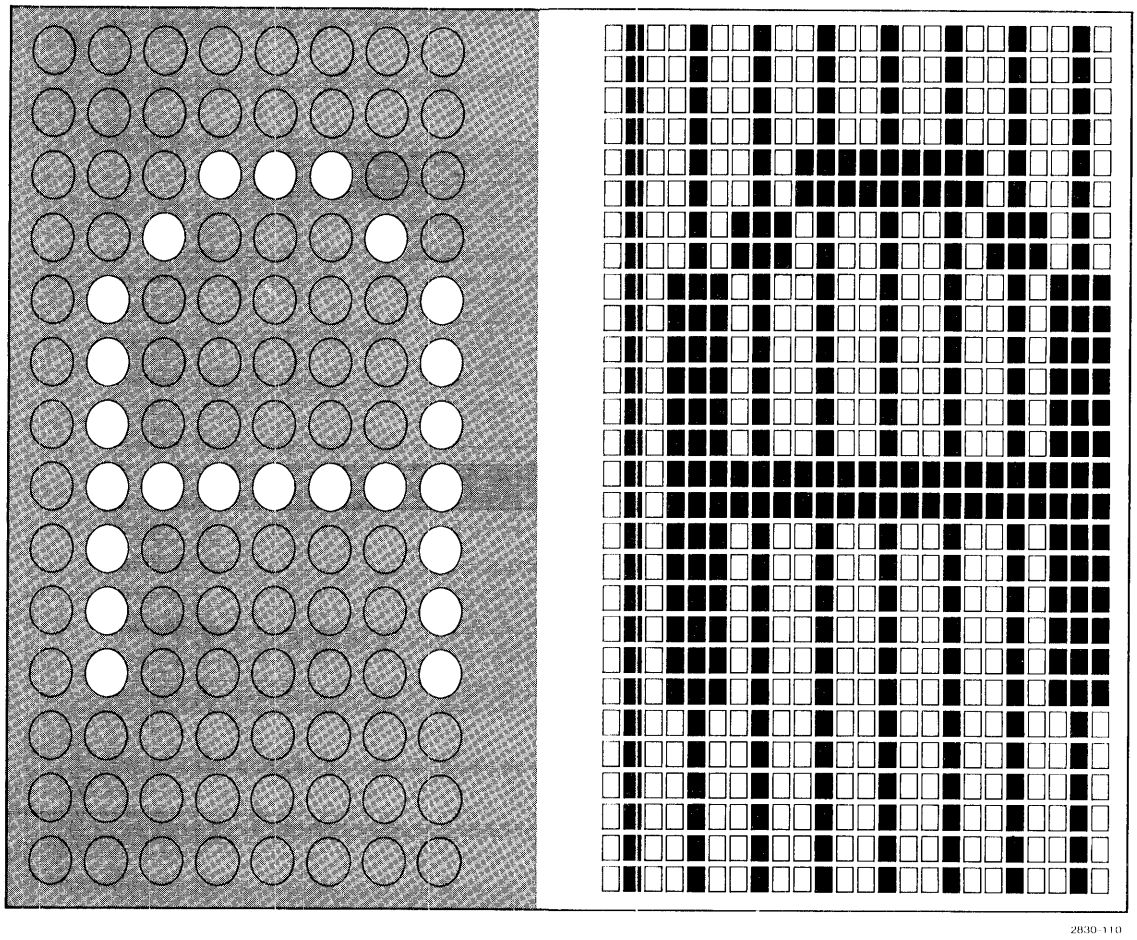


Figure 9-26. Video and Hard Copy Representations of a Character Displayed with Enhanced Background.

At the first HCCBG pulse during a hard copy, PTIME has not yet gone true. This first HCCBG pulse sets U361A. The Q output of U361A does two things: it turns off gate U261A (so that SWAP pulses no longer reach the Buffer Control circuit); and it provides a high at the "D" input of the 74LS161 counter (loading a "1" into the 74LS161's "D" flip- flop).

Disabling the SWAP signal prevents the buffers from being swapped. Consequently, the Display Controller "idles"; it scans the same Row Buffer over and over.

The "1" in the "D" bit of 74LS161 sends the BLACK2 signal. BLACK2 causes the display to be blanked. Hence, while the Display Controller is "idling" (scanning the first Row Buffer again and again), the display is turned off and the hard copy unit records nothing.

This idle state persists until PTIME goes low. As PTIME goes low, it clears flip-flop U361A, so that the D input of the 74LS161 goes low. On the next LASTRAST pulse, the 74LS161's D flip-flop is cleared, and BLACK2 signal is turned off. This ends the "idle state." From now on, the Raster Counter functions normally, the SWAP signals reach the Buffer Control circuit as usual, and a hard copy is produced.

4024 DISPLAY CONTROLLER

The 4024 Display Controller closely resembles the 4025's. To avoid tedious repetition, the following description is very brief. It concentrates on those aspects of the 4024 Display Controller which differ from the 4025.

Block Diagram

Figure 9-27 is a block diagram for the 4024 Display Controller. The board is functionally divided into four major blocks, each of which has sub-blocks: the Timing and Control Circuitry, the Display List Decoder, the Row Buffers, and the Video Generator. Each of these blocks is described separately.

Timing and Control Circuitry

The Timing and Control Circuitry includes: Video Timing, Timing Generator, Buffer Control, and Raster Counter.

The Video Timing Circuitry resembles that in the 4025 Display Controller. It includes a camera sync generator integrated circuit, which generates a number of timing signals for the video display and for other parts of the Display Controller. It also includes a divide-by-1.5 circuit which generates the 12 MHz bit clock for the shift register and the Timing Generator.

The Timing Generator generates signals analogous to those generated by the Timing Selector and Generator in the 4025. Some of the more important of these signals are: COLCLK, which regulates the flow of data through the video pipeline; BITSYNC, which loads the shift register at the start of each character cell; the 14 MHz "scrunch clock;" and the BLANK and BLACK signals to blank the video display.

The Buffer Control circuitry provides signals to control the Row Buffers. These are like signals already described for the 4025 Display Controller.

The Raster Counter, again, is like that in the 4025 Display Controller.

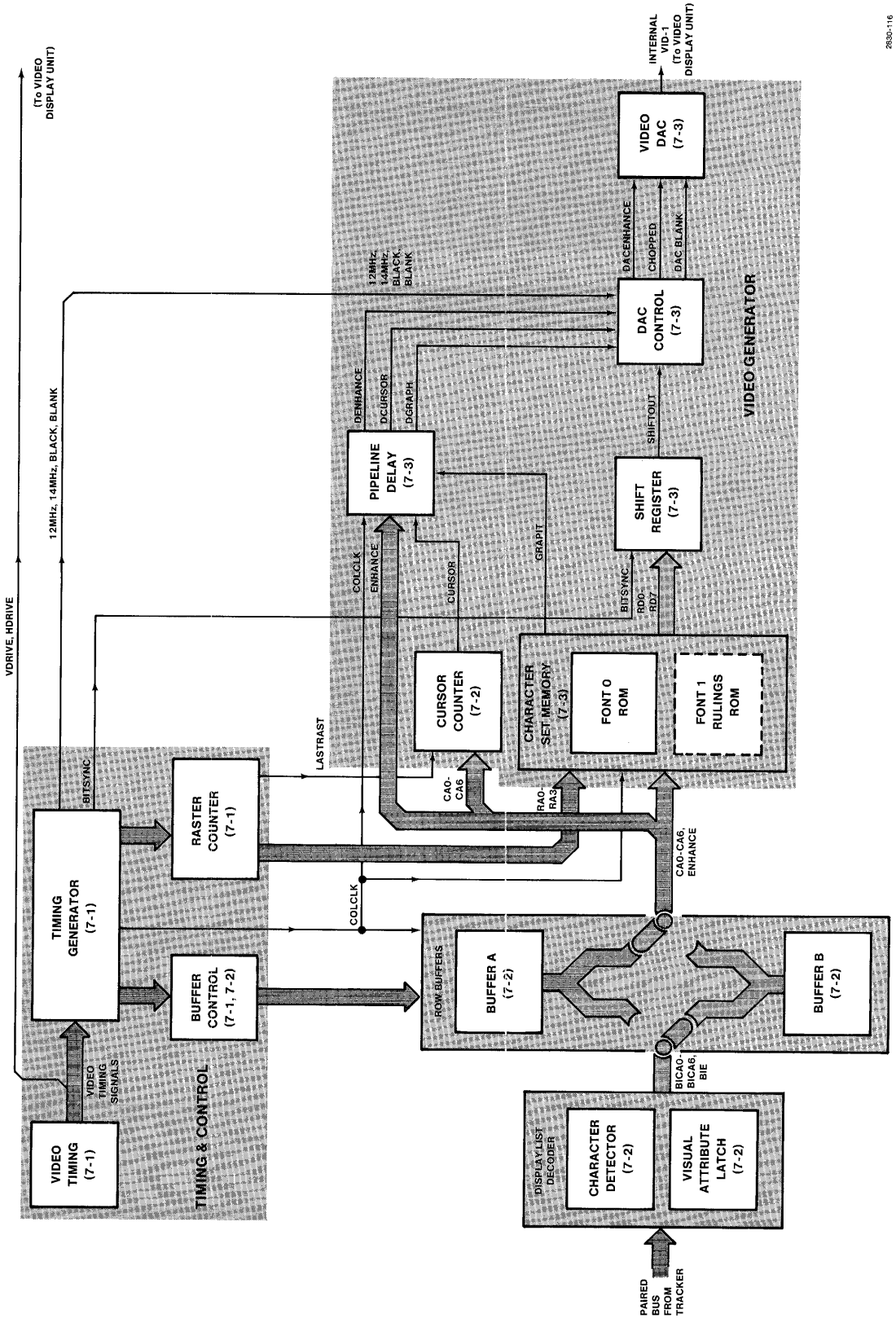


Figure 9-27. 4024 Display Controller Block Diagram.

Display List Decoder

The 4024's Display List Decoder lacks a Cursor Counter and Font Address Latch, and its Visual Attribute Latch is simpler than the 4025's.

The 4024 cursor information is loaded as the the first character in each row buffer instead of being loaded as a cursor bit in each Row Buffer word. Because of this, the 4024's Cursor Counter is not in the Display List Decoder, but in the Video Generator.

The 4024 has only one alternate font: font one, the rulings font. The 4024 hardware treats this font as part of the standard font: as an alternate way of displaying ASCII control characters. (Normally, the Processor never places control characters in the display list, so they are never displayed. They are only displayed in "snoopy mode," after a !SNOOPY YES command has been given. When rulings are installed in the 4024, they replace the "snoopy" characters in the standard ASCII font.) Because the hardware treats rulings characters as if they were "snoopy" characters of font zero, the display list does not use font attribute codes.

Since the 4024 only has two visual attributes ("standard" and "enhanced"), the 4024 pays attention only to the "enhanced" bit in visual attribute codes. The Visual Attribute Latch has only one output: the BIE (Buffer Input Enhanced) line.

Row Buffers

Each Row Buffer holds 81 eight-bit words. Each word's seven least-significant bits comprise a seven-bit ASCII character; the most-significant bit specifies whether that character is to be displayed on an enhanced background.

The first ASCII character stored in each Row Buffer is the cursor control character. When the Buffer is scanned, that character is used to load the Cursor Counter in the Video Generator. The remaining 80 characters in the Buffer are the visible characters displayed on the screen.

Video Generator

As shown in Figure 9-27, the Video Generator consists of these blocks: Character Set Memory, Shift Register, DAC Control, Video DAC, Cursor Counter, and Pipeline Delay. Of these, the Shift Register, DAC Control, Video DAC, and Pipeline Delay circuits closely resemble those in the 4025.

Character Set Memory

(See Schematic 7-3.)

The Character Set Memory consists of an address latch (U105), the Font 0 ROM (U5), an adder (U15) to adjust the row address inputs to that ROM, and an output buffer(U20) to place the ROM output data on the Shift Register inputs RD0-RD7. If rulings characters are installed, they occupy another ROM, U25, which is only enabled (by gate U110A) when an ASCII control character is detected.

The address latch, Font 0 ROM, adder, and output buffer function as in the 4025.

If rulings (Option 32) are not installed in 4024, connectors J16 and J17 each have Pin 2 connected to Pin 3 by a strap. This causes the FONT0 signal always to be true and the GRAPH signal always to be false. FONT0 being true means that the standard character font ROM is selected, even for ASCII control characters. GRAPH being false means that all characters are regarded as alphanumeric characters, and the 14 MHz "scrunch clock" is selected.

When Option 32 is installed, J16 Pin 1 is connected to J17 Pin 1, and J16 Pin 2 is connected to J17 Pin 2. With the connectors strapped this way, control characters (which are detected from their most significant bits by U110A) cause FONT0 to go false and GRAPH to go true.

With FONT0 false, the output buffers for U5 are disabled and U25 is enabled. (U25 holds the rulings characters.) Hence, with Option 32 installed, all control characters are displayed as rulings characters.

With GRAPH true, the 12 MHz clock is enabled (instead of the 14 MHz scrunch clock). This lets the rulings characters adjoin each other, without intervening spaces.

Cursor Counter

(See Schematic 7-2.)

At the start of a Row Buffer scan, the Cursor Counter is loaded (with the contents of the cursor control character) by the CURSLD signal from the Timing and Control circuitry. Each COLCLK pulse decrements the counter. If the counter reaches zero, the current character is to be displayed with a cursor. When the counter reaches zero and LASTRAST is true, an AND gate turns on the CURSOR signal.

If the cursor counter does nor reach zero (because it was loaded with a number greater than 79), no cursor is displayed for that row of text.

Section 10

OPTION 1 — HALF DUPLEX (4025 ONLY)

Option 1 consists of a ROM holding firmware for the Half Duplex and Half Duplex with Supervisor modes of data communication.

With Option 1 installed, the DUPLEX and DISCONNECT commands are added to the 4025's command set. These commands are summarized in Table 10-1. For more information, see the 4024/4025 Programmer's Reference Manual.

For installation procedures, see Volume 2 of this Service Manual.

Table 10-1

OPTION 1 HALF DUPLEX COMMANDS

!DIS <CR>	Sends a signal to the modem, disconnecting the 4025 from the communications line. (The terminal turns off the DTR ("data terminal ready") signal on the RS-232 interface for about one second. This causes the modem to disconnect from the communications line.)
!DUP <CR> **!DUP F <CR>**	Sets the 4025 for full duplex.
!DUP H <CR> **!DUP H S <CR>** **!DUP H S L <CR>**	Sets the 4025 for half duplex with supervisor. In buffered mode the prompt condition is line turnaround only.
!DUP H S P <CR>	Sets the 4025 for half duplex with supervisor. In buffered mode the prompt condition is the prompt string plus line turnaround.
!DUP H N <CR> **!DUP H N L <CR>**	Sets the 4025 for half duplex normal. In buffered mode the prompt string is line turnaround only.
!DUP H N P <CR>	Sets the 4025 for half duplex normal. In buffered mode the prompt condition is the prompt string plus line turnaround.

Section 11

OPTION 2 — CURRENT LOOP INTERFACE (4024 AND 4025)

INTRODUCTION

NOTE

For installation procedures, see Volume 2.

The Current Loop Interface (Option 2) lets you connect a 4024 or 4025 to its host computer with a "current loop" cable. This cable has conductors for two independent circuits, or loops. One circuit (the transmit loop) carries data to the computer; the other (the receive loop) carries data from the computer to the terminal.

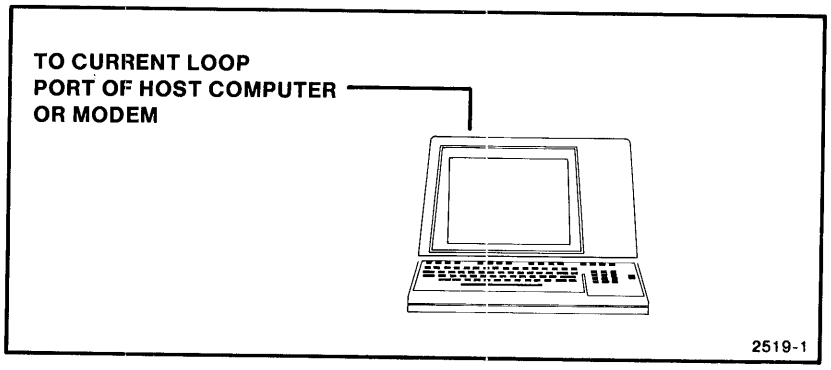


Figure 11-1. Current Loop Connecting Terminal to Host Computer.

NOTE

Several 4020 series terminals may be connected in series on the same current loop cable, as in Figure 11-2. If they are, however, they must all have the Option 10 Polling Interface); otherwise, the terminals will interfere with each other. (Usually, one terminal will also be equipped with an Option 11 Polling Controller). Also, the connecting cables are wired differently for use with Option 10 than for use with Option 2.

This section of the manual is limited to describing Option 2; for information about the Current Loop Interface as used with Option 10, see the section describing Options 10 and 11.

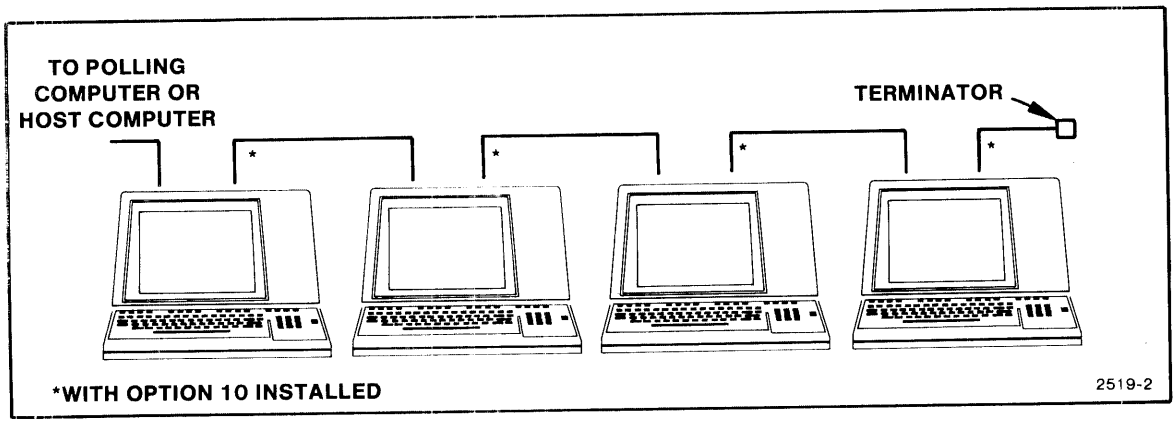


Figure 11-2. Multiple Terminals on the Same Current Loop.

The Current Loop Interface can operate at data rates up to 9600 baud (provided the computer's or modem's current loop port is also capable of communicating at that rate).

NOTE

Some host computers have debouncing capacitors in their current loop ports. These capacitors are intended to filter out electrical noise from the selector magnets and mechanical switches in mechanical teleprinters. Such filtering is not required with the 4020 Series Computer Display Terminals. Indeed, it may severely limit the frequency response of the current loop, resulting in much lower data rates.

Should you encounter such a problem, the remedy may be to remove the debouncing filters in the computer's or modem's current loop port. Consult service personnel of the computer or modem manufacturer for advice concerning any such modifications.

THEORY OF OPERATION

Figure 11-3 is a block diagram of the Current Loop Interface. The figure assumes that both the transmit and receive current loops are set to PASSIVE mode. (The ACTIVE/PASSIVE straps are described in the installation procedures in Volume 2 of this manual. In ACTIVE mode, the Current Loop Interface provides the 20 mA current source to drive the current loop. In PASSIVE mode, it relies on an external device—the computer or modem—to provide the source of current for the loop.)

The current in the receive loop controls the Current Loop Interface's receive circuitry, which drives the terminal's RDATA line. Current flowing in the loop is interpreted as a "mark," or binary one; the absence of current is interpreted as a "space," or binary zero. Binary ones send RDATA negative; binary zeroes send RDATA positive.

Likewise, the TDATA signal from the terminal controls the Current Loop Interface's "transmit" circuitry, which keys the transmit loop. A negative voltage on TDATA is interpreted as a mark or binary one, causing current to flow in the transmit loop. A positive voltage on TDATA is interpreted as a space or binary zero, shutting off the flow of current.

Note that, with the Current Loop Interface in PASSIVE mode, conventional current (a flow of imaginary positive charges) enters the Current Loop Interface through the T+ lead and leaves it through the T- lead. Likewise, conventional current enters the R+ lead and leaves the R- lead.

Figure 11-4 shows the Current Loop Inteface set to ACTIVE mode in both the transmit and receive loops. Compared to PASSIVE mode, there are only two differences:

- It is the Current Loop Interface, rather than the computer (or modem) which provides the source of current in the loops.
- Conventional current (a flow of imaginary positive charges) leaves the T+ lead and returns by the T- lead. Likewise, conventional current leaves the R+ lead and returns by the R- lead.

For more details on the Current Loop Interface's circuitry, refer to Schematic 11-1 in Volume 2 of this Service Manual.

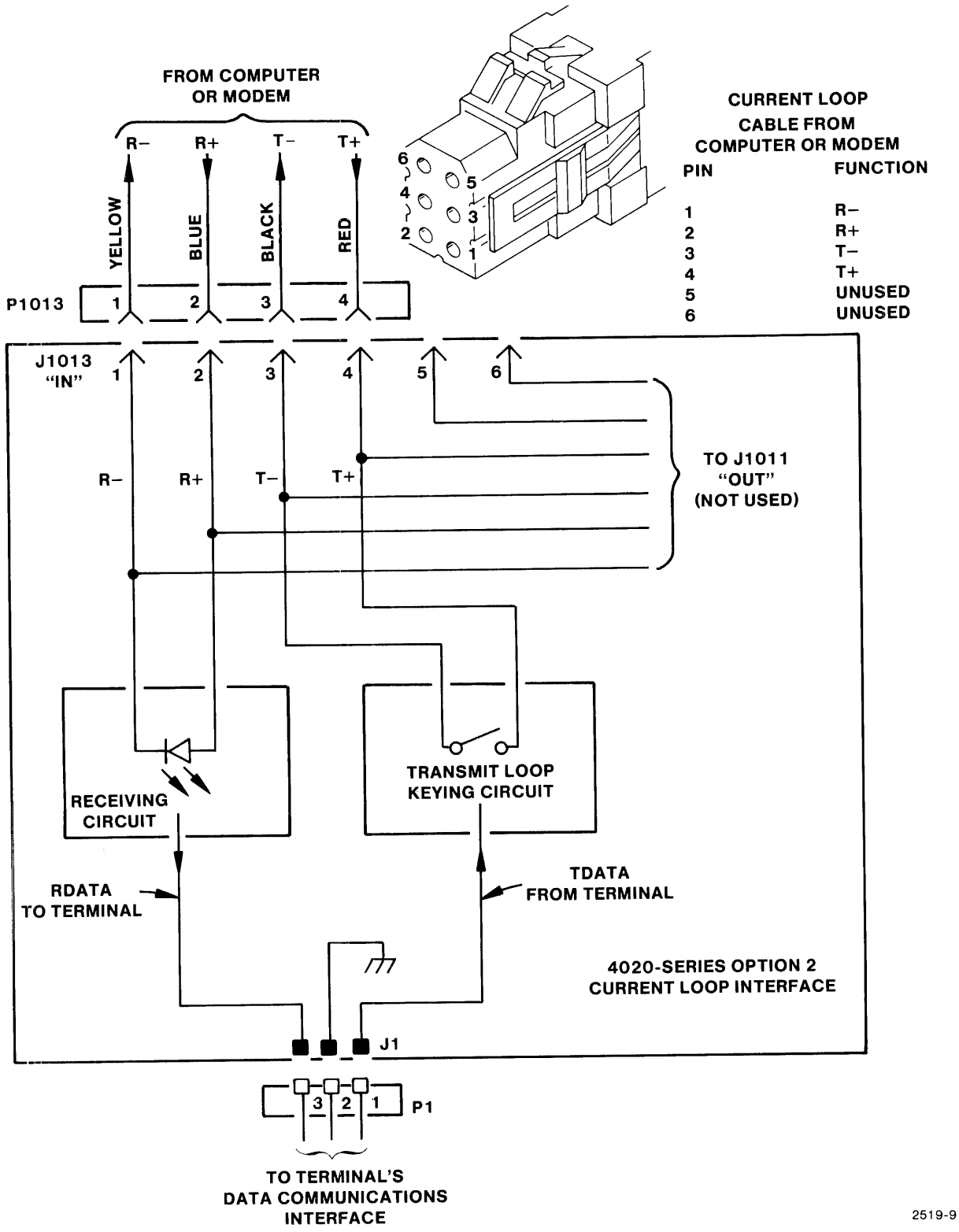


Figure 11-3. Option 2 Block Diagram in PASSIVE Mode.

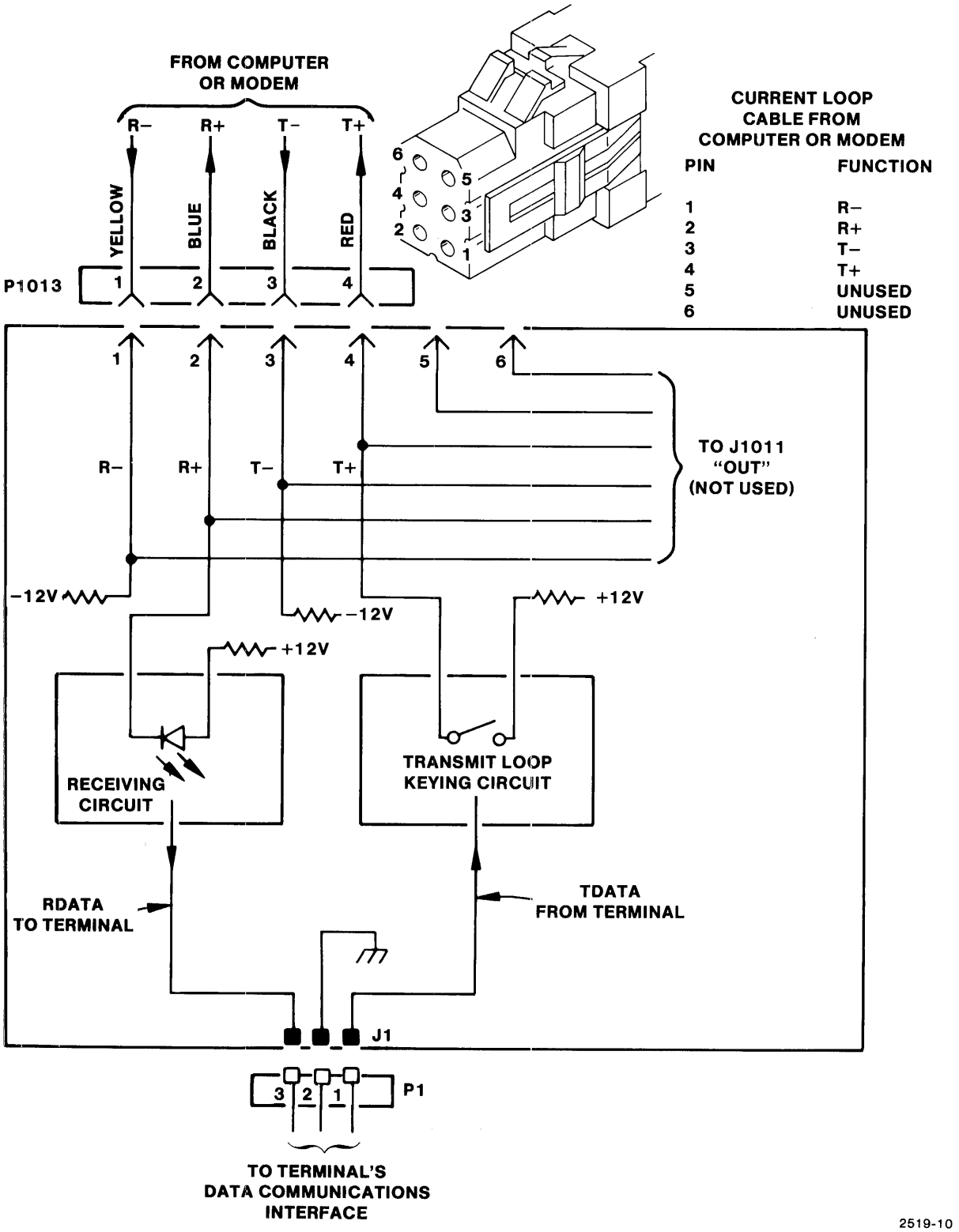


Figure 11-4. Option 2 Block Diagram in ACTIVE Mode.

Section 12

OPTION 3 — RS-232 PERIPHERAL INTERFACE

INTRODUCTION

The RS-232 Peripheral Interface is a circuit board which provides a serial data port for communicating with RS-232 peripheral devices. The firmware which drives Option 3 is available separately as Option 36, Peripherals ROMs. (These same ROMs can also drive the Option 4, GPIB Peripheral Interface.)

NOTE

The "receive" capabilities of the RS-232 Peripheral Interface Board are not used by the Option 36 firmware which drives it. Consequently, the RS-232 Peripheral Interface can only be used with a printer (or other output-only device).

The circuitry on the RS-232 Peripheral Interface Board is designed around a large scale integrated circuit, the 8251A USART (Universal Synchronous/Asynchronous Receiver/Transmitter). Besides serial input and output, the circuit provides two inputs and two outputs for handshaking purposes, an external clock input, and a transmitter clock output.

SUMMARY OF CHARACTERISTICS

- Baud Rates: 75 to 9600, jumper selected; An external clock may be selected.
- Inputs: Serial data; two handshaking flags; external clock.
- Outputs: Serial data; two handshaking flags; transmit baud rate clock.
- Interrupts: Transmitter ready; receiver ready; flag status change; priority level jumper selectable (0 to 7).
- Data Format: The USART may be programmed to support several synchronous and asynchronous formats. (See the note below.)

NOTE

As set up by 4025 Option 36 (ROM containing firmware for driving the TEKTRONIX 4642 Printer), the RS-232 Peripheral Interface provides asynchronous communications with eight data bits per character, no parity, and two stop bits per character. The external clock must be 16 times the desired baud rate.

CIRCUIT DESCRIPTIONS

USART

The 8251A USART converts parallel data bytes from the Processor into a serial bit stream. The USART also performs a number of coordinating functions.

The USART (Figure 12-1) can be divided functionally into five circuits: the Data Bus Buffer, the Read/Write Control Logic, the Transmitter, the Receiver, and the Modem Control.

Data Bus Buffer

The Data Bus Buffer transfers data between the USART's internal data bus and the circuit board's data bus. The bytes can be transmitter data, commands to the USART, or status information from the USART.

Read/Write Control Logic

The Read/Write Control Logic co-ordinates routing and storage of data within the USART. Its inputs are:

RESET	When RESET goes high, the USART enters an "idle" state. This prepares it to receive a set of control bytes from the Processor to select its mode of operation. Any time the RESET line on the Mother Board is pulled low, the USART is reset.
CLK	The CLK (Clock) signal times various functions within the USART.
WR	The WR (Write) signal informs the USART that information is to be written into it.
RD	Informs the USART that information is to be read from it.
C/D	The C/D (Control/Data) signal, in conjunction with WR and RD, tells the USART whether the data being transferred is character, control, or status information.
CS	A chip select signal to enable the USART for reading or writng.

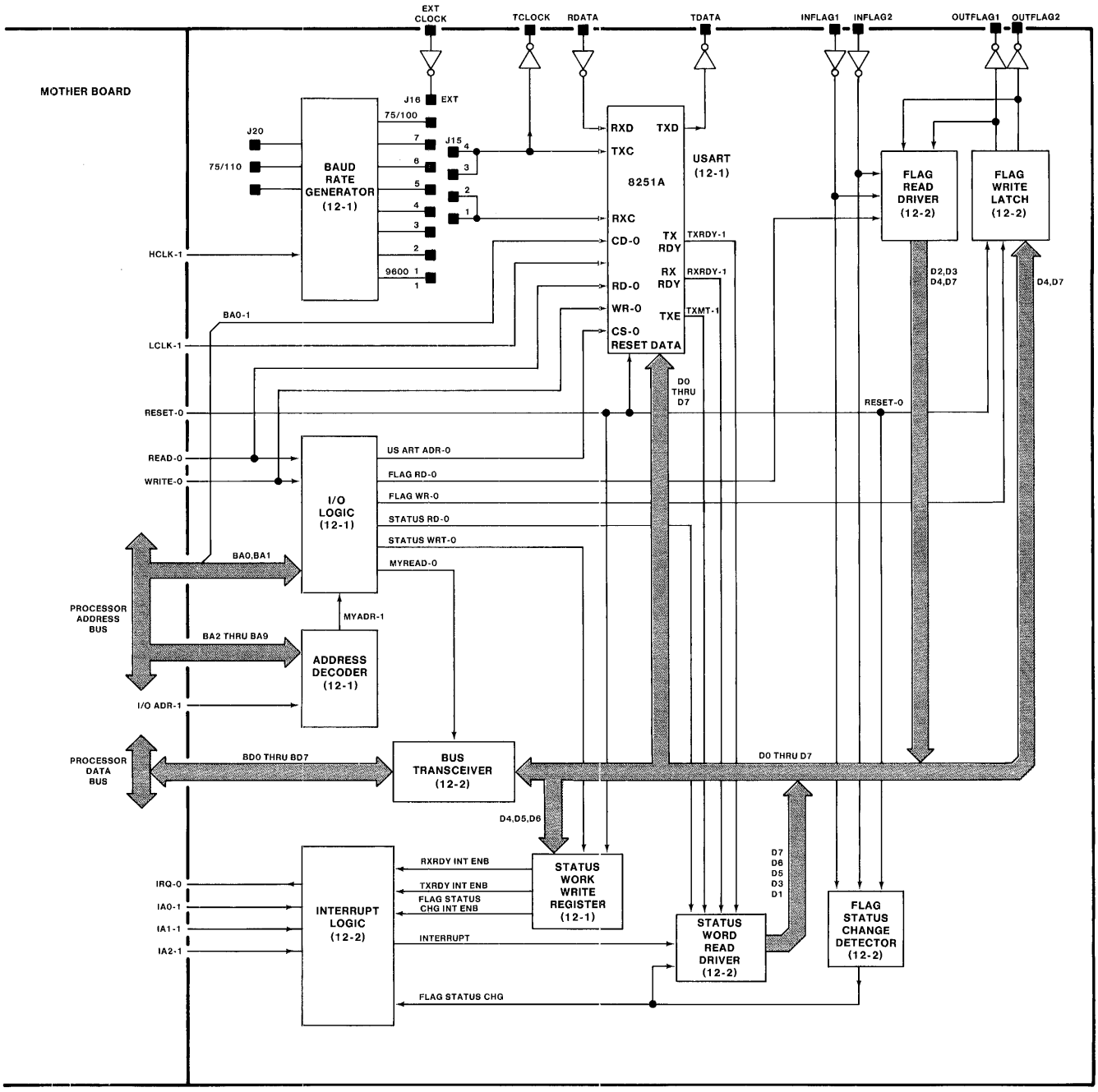


Figure 12-1. RS-232 Peripheral Interface Board Block Diagram.

Table 12-1 explains how the WR, RD, C/D, and CS signals work together to control data flow to and from the USART.

Table 12-1

USART I/O TRUTH TABLE

Signal State				Function
C-1/D-0	RD-0	WR-0	S-0	
0	0	1	0	Receiver data to data bus
0	1	0	0	Data bus to transmitter
1	0	1	0	USART status to data bus
1	1	0	0	Data bus to USART control

Transmitter

The Transmitter accepts data bytes in parallel, then shifts them out bit-serially from the TXD output. It provides the following inputs and outputs.

TXD — TXD (Transmitter Data) is in the high state when the transmitter has no data to send. As soon as the transmitter receives a character, TXD goes low for one bit period to produce a start bit. Then the data bits are transmitted . After the data bits have been transmitted, the line is held high for two bit periods to produce the stop bits.

TXRDY — TXRDY signals that the transmitter is ready to accept the next data character. It is used to generate an interrupt and can also be read through bit one of the board status word.

TXE — TXE (Transmitter Empty) signals that the transmitter buffer is empty. This can be read through bit 6 of the board status word.

TXC — TXC (Transmitter Clock) controls the rate at which the data is transmitted. The falling edge of TXC shifts the data out of TXD. The firmware programs the USART to use a 16X clock (a clock whose frequency is 16 times the baud rate).

Receiver

Since the Option 36 firmware uses the RS-232 Interface only to drive a printer, the USART's receiver is not used.

Modem Control

The USART includes "modem control" circuitry. However, this circuitry is not used.

Address Decoder

Nine 74LS266 Exclusive-OR gates (Schematic 12-1) make up the address decoder. These gates have open-collector outputs which are connected together to form a wired-AND function. All the outputs must go high to produce the MYADR signal. The I/O Address is determined by the presence or absence of jumpers at one of the inputs to eight of the Exclusive-OR gates. If a jumper is installed, the input is connected to ground. Otherwise, the input is pulled up to 5 volts through a resistor. The other inputs to these gates are connected to the address bus lines BA2 through BA9. If the state of BA2 through BA9 matches the state of the jumpered inputs, the outputs of these eight gates go can go high. A ninth gate is connected to IOADR. Its output can go high any time IOADR is true. IOADR is true any time the processor addresses a location beween X'0800' and X'0BFF'. The jumpers select a base address within this range. In the 4025, jumpers to ground are installed at positions A2, A3, A4, A5, A8, and A9. Therefore, the valid addresses are X'08C0', X'08C1', X'08C2', and X'08C3'. These are the RS-232 Peripheral Interface's Status Word, Flag Word, I/O Data Word, and USART Word, respectively. The two low-order bus address bits are decoded by the I/O logic to select these locations.

Table 12-2

RS-232 PERIPHERAL INTERFACE I/O REGISTERS

Bus Address	Bit	Description
X'08C0'		Status Word (Read)
	0	(Unused)
	1	TXRDY (USART transmitter ready for data)
	2	(Unused)
	3	RXRDY (USART receiver has data ready for the Processor)
	4	(Unused)
	5	Flag Status Change
	6	TXE (USART transmitter empty)
	7	Interrupt Condition Present
		Status Word (Write)
	0	TXRDY Interrupt Enable
	1	(Unused)
	2	RXRDY Interrupt Enable
	3	(Unused)
	4	Flag Status Change Interrupt Enable
	5-7	(Unused)
X'08C1'		Flag Word (Read)
	0-1	(Unused)
	2	INFLAG1
	3	INFLAG2
	4	OUTFLAG1
	5-6	(Unused)
	7	OUTFLAG2
		Flag Word (Write)
	0-3	(Unused)
	4	OUTFLAG1
	5-6	(Unused)
	7	OUTFLAG2
X'08C2'		Data Word (Read)
	0-7	The Processor reads from the USART receive buffer at this address
		Data Word (Write)
	0-7	The Procesor writes at this address to give the USART a character to transmit
X'08C3'		USART Status and Programming (Read)
	0-7	The Processor reads the USARTX's status from this address
		USART Status and Programming (Write)
	0-7	The firmware programs the USART by having the Processor write into this word

I/O Logic

The I/O logic (Schematic 12-1) consists of several gates and a 74LS155 multiplexer which examine the MYADR, BA0, BA1, WRITE, and READ signals to determine where read or write operations are to be performed. It sends the following signals to control these operations:

STATUSWR Selects the Status Word Write Register.

USARTADR The chip select signal for the USART. This enables reading or writing in the USART.

FLAGWR Enables writing into the Flag Write Latch (the "write" half of the Flag Word).

STATUSRD Enables the Status Word Read Driver (the "read" half of the Status Word).

MYREAD Controls the direction of data transfer through the Bus Transceiver.

Table 12-3 explains the conditions under which these output functions are active.

Table 12-3

RS-232 PERIPHERAL INTERFACE I/O LOGIC TRUTH TABLE

READ-0	WRITE-0	BA0-0	BA1-0	MYADR-1	Function
1	0	0	0	1	STATUSWR-0
X	X	X	1	1	USARTADR-0
1	0	1	0	1	FLAGWR-0
0	1	1	0	1	FLAGRD-0
0	1	0	0	1	STATUSRD-0
0	X	X	X	1	MYREAD-0

X= Irrelevant

Status Word Write Register

The Status Word Write Register (Schematic 12-1) consists of four D-type flip-flops. Initially, the register is cleared by the RESET-0 line. When a write operation takes place at X'800', STATUSWR-0 causes the contents of data lines D0, D2, and D4 to be loaded into the register (see the I/O address map).

Status Word Read Driver

The Status Word Read Driver (Schematic 12-2) consists of a 74LS367 "hex bus driver" IC. When a read occurs at X'08C0', STATUSRD enables the Driver, causing several status bits to appear on the data bus (see the I/O address map).

Flag Write Latch

Refer to Schematic 12-2. The Flag Write Latch consists of two type D flip-flops in a 74LS74 integrated circuit. When the Processor writes into the Flag Word, FLAGWR clocks these flip-flops, loading them with data bits D4 and D7. The flip-flop outputs drive RS-232 bus drivers, which drive "flag" signals OUTFLAG1 and OUTFLAG2 for external devices. The OUTFLAG1 and OUTFLAG2 signals can also be read through the Flag Read Driver.

Flag Read Driver

The Flag Read Driver consists of a 74LS368 Bus Driver. A read from X'08C1' causes the FLAGRD signal to enable the 74LS368, which places the flag bits on the bus (see the I/O address map).

Flag Status Change Detector

Refer to Schematic 12-2. The Flag Status Change Detector consists of several inverters, Exclusive-OR gates, and a flip-flop. When a change takes place in INFLAG1 or INFLAG2, this change (after passing through an RS-232 receiver) is seen at the inputs of an Exclusive-OR gate. The change is seen immediately at one input. However, it is delayed in getting to the other input because it must pass through an inverter, charge a capacitor, then pass through three more inverters on its way to the input. When the status of an input flag line changes state, then, there is a brief period when the inputs of its Exclusive-OR gate differ. During this period, the Exclusive-OR gate pulls its output low, setting the Q input of the 74LS174 high. This causes a flag status interrupt to occur (provided that interrupt has been enabled). The state of this flip-flop can be read through the Status Word. It is automatically reset when the Processor reads from the Status Word or when a system reset occurs.

Interrupt Logic

The Interrupt Logic (Schematic 12-2) consists of several AND gates and Exclusive-OR gates. Jumpers at the Exclusive-OR inputs set the interrupt address for the board. Jumpers are installed for those interrupt address bits which are to be set to zero. The open-collector outputs of the gates form a "wired-AND"; when the IA0-IA2 interrupt address lines match the jumper-set states, the wired-AND goes high, causing one input of a NAND gate to be pulled high. The other input to this NAND gate goes true whenever one of the three interrupt conditions occurs (provided they are enabled by the Processor). The ultimate result is that IRQ (Interrupt Request) will be pulled low.

For example, if a one is written into bit 0 of the status word, TXRDY interrupts are enabled. Whenever the TXRDY signal goes true, the output of a NAND gate connected to Q6 of the Status Word Write Register must go low. This output then ripples through an OR gate, a NAND gate (assuming the selected interrupt priority is on the IA0-IA2 lines), and an Exclusive-OR gate to pull the IRQ line low.

Baud Rate Generator

The Baud Rate Generator (Schematic 12-1) is composed of four counters and several inverters. The HCLK 18.432 MHz clock is divided down to provide 16X clocks for the USART.

The first counter divides the HCLK frequency by 8. This counter's Q_C output goes to a 74LS161 where it is further divided by 15. The Q_D output of this counter is used for the 9600 baud 16X clock. It is also the input to a seven stage counter. The seven outputs of this divider are used for the other baud rate clocks. The 110 baud output is derived by dividing the 1200 baud output by 11 in a fourth counter (74LS161).

Bus Transceiver

The Bus Transceiver (Schematic 12-1) consists of two 74LS243 bidirectional drivers. These control the direction of data flow between the circuit board and the Processor data bus. When MYREAD is true (low), data flows to the Processor bus. When MYREAD is false (high), data on the processor bus is transferred to the board's internal bus.

FIRMWARE

The firmware which drives the Option 3 RS-232 Peripheral Interface resides (together with firmware for Option 4) in the Option 36 Peripherals ROMs. The following description of this firmware should help you understand how the RS-232 Peripheral Interface circuitry is used.

Initializing the RS-232 Peripheral Interface

On power-up, or when MASTER RESET is pressed, the Option 36 firmware causes the Processor to do the following:

1. Read location X'08C0' (Board Status Word) to see if the RS-232 Peripheral Interface Board is installed. (If X'FF' is read from this location, the board is *not* installed.)

2. Store X'80' at X'08C1' (the Flag Word). This sets OUTFLAG1 and OUTFLAG2 to the spacing state.

3. Reset the USART. To do this, the Processor:

   A. Writes X'00' at X'08C3'.

   B. Waits 32 microseconds.

   C. Does A and B two more times.

   D. Writes X'40' at location X'08C3' (the USART Control Word). This sets the INTERNAL RESET bit in the USART's status register and prepares it to receive the mode instruction.

4. Write X'CE' at location X'08C3' (the USART Control Word). This is the mode instruction. It sets up the USART for asynchronous communications with eight data bits per character, no parity, and two stop bits.

5. Wait 32 microseconds.

6. Write X'15' at location X'08C3'. This is a command instruction. It resets all of the USART's error flags and enables its receiver and transmitter.

Sending Characters to the Printer

When sending characters to the printer, the firmware causes the Processor to do the following:

1. The Processor gets a character from the printer queue and writes it at X'08C2' (the USART data port).

2. If this character is a control character (a character such as *carriage return* or *line feed* requiring extra mechanical motion by the printer), the Processor does one of two things. (Which of these it does is determined by an earlier SET command given to the terminal.)

   - If the terminal has been set to use flagged simplex protocol (!SET PR 0), the Processor will wait for the printer to send DTR (Data Terminal Ready) true. (The RS-232 DTR line is connected to the board's INFLAG1 input.) While waiting for DTR, the Processor enables line status change interrupts. As each interrupt occurs, it tests the Flag Word (X'08C1') to see if INFLAG1 (that is, DTR) has gone true. If DTR is true, then output interrupts are enabled and line status change interrupts are disabled.

   - The terminal may have been set to use a fixed delay after control characters. (For instance, !SET PR 3 would cause the terminal to wait 0.3 seconds after a control character before sending the next character.) If this is the case, the Processor goes into a time delay routine after sending each control character out. The purpose of this time delay is to allow the printer sufficient time to do a carriage return/line feed.

4. If the printer queue is empty, the Processor disables interrupts. (Input interrupts are always disabled, because the Option 36 firmware makes no provision for receiving characters through the RS-232 Peripheral Interface.)

Section 13

OPTION 4 — GPIB PERIPHERAL INTERFACE

INTRODUCTION

The GPIB Peripheral Interface Board (4025 Option 4) provides the circuitry to interface the 4025's Processor with the GPIB (General Purpose Interface Bus). The GPIB firmware (microprocessor programs) is in the Option 36 Peripherals ROMs.

The hardware (Option 4) is quite versatile; it can perform any allowed GPIB function. However, the firmware (Option 36) is more limited; it provides for communication *only* with the TEKTRONIX 4924 Digital Cartridge Tape Drive and the TEKTRONIX 4662 Interactive Digital Plotter.

NOTE

For installation procedures, see Volume 2 of this Service Manual.

ABOUT THE GPIB

The GPIB is a standard interface for programmable instrumentation, defined in IEEE Standard 488-1975. The following description summarizes the pertinent parts of that standard.

Talkers, Listeners, and Controllers

There are three types of GPIB devices: *talkers*, *listeners*, and *controllers*. In any particular data transfer, the talker is the device which sends the data, and the listeners are the devices receiving it. The controller supervises the data transfer; it determines which device is to talk and which are to listen. (The Option 36 firmware requires that the 4025 be the only controller on the GPIB.)

GPIB Interface

The actual interface (between terminal and peripheral devices) is through a GPIB cable. This is a shielded, 24-wire cable which attaches to a standard connector (Figure 13-1) on the peripheral devices. Eight of the cable's wires are grounds; the other sixteen are logically divided into three distinct groups: a data bus, a management bus, and a transfer bus.

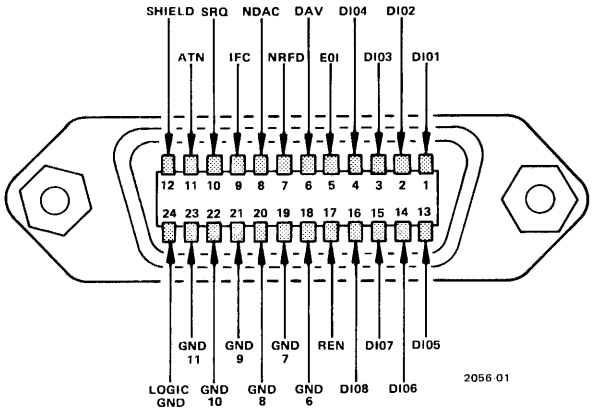


Figure 13-1. GPIB Connector.

All devices on the GPIB are connected in parallel, and all its signal lines are active low, passive high. A line is low if any GPIB device pulls it low (i.e., to ground) and high only if all devices let it float to a TTL high (i.e., + 3.4 V). In other words, devices on the GPIB are connected in a "wired-OR" configuration.

Data Bus

The GPIB data bus contains eight bidirectional signal lines. One byte of information (eight bits) is transferred over the bus at a time. DIO1 (Data In-Out bit 1) is the least significant bit, and DIO8 the most significant bit. Each byte represents a device address, a universal command, or a datum. (Device addresses and universal commands are distinguished from data by having the ATN line—in the management bus—activated while they are sent. With ATN asserted, certain bytes are reserved for universal commands and others for device addresses.)

Management Bus

The GPIB management bus consists of five signal lines for controlling data transfers. The lines are: ATN, SRQ, IFC, REN, and EOI.

ATN (Attention). The GPIB controller (that is, the 4025) sends the ATN signal when assigning devices as listeners or talkers (sending device addresses) or when giving commands simultaneously to all devices on the bus (sending universal commands). Only device addresses and control messages can be transferred over the data bus when ATN is true. After ATN goes false (high), only those devices assigned as listeners and talkers can take part in the data transfer.

SRQ (Service Request). Any device on the bus can request the controller's attention by sending SRQ active low.

IFC (Interface Clear). The GPIB controller may send IFC true to put all devices on the GPIB in a known quiescent state. (The 4025 does this during power-up or MASTER RESET.)

REN (Remote Enable). The REN signal is used in some GPIB applications to transfer devices from manual operation to remote control.

EOI (End or Identify). Talkers may use the EOI signal to mark the end of a data transfer. (The talker sends EOI while sending the last data byte in the sequence.)

Transfer Bus

Each time a byte is transferred over the data bus, the talker and listeners execute a handshake sequence using the three transfer bus signal lines: NRFD, DAV, and NDAC.

NRFD (Not Ready For Data). A low NRFD signal means that one or more assigned listeners are not ready to receive the next byte. When all the listeners have released NRFD, the NRFD line goes high. This tells the talker that it may place the next byte on the data bus.

DAV (Data Valid). The talker sends DAV low shortly after placing a valid byte on the data bus. A true (low) DAV signal tells each listener to capture the data presented on the data bus. The talker is inhibited from sending DAV when NRFD is low.

NDAC (Not Data Accepted). The NDAC signal is held low by each listener until it has captured the byte currently presented on the data bus. When all listeners have captured the byte, NDAC goes high. This tells the talker that it may remove the byte from the data bus.

Handshake Sequence

Figure 13-2 shows the transfer bus "handshaking" that regulates the exchange of data bytes on the data bus.

Initially, the listeners hold NDAC (Not Data Accepted) low, and the talker leaves DAV (Data Valid) high. One or more of the listeners may be holding NRFD (Not Ready For Data) low, indicating that it is not yet ready to accept a data byte.

When all listeners are ready for data, NRFD goes high. The talker then places a data byte on the data bus, waits briefly for the data to settle, and pulls DAV low. The low DAV signal indicates that valid data is available on the data bus.

The listeners then capture the data. Before beginning to accept the byte, each listener pulls NRFD low, indicating that it is not ready for the talker to place another byte on the data bus. Then the listeners read the data, and when done, release NDAC. When the slowest listener has captured the data, NDAC goes high; this tells the talker that all listeners have received the byte.

The talker then releases the DAV line and changes the data byte on the data bus. The listeners, sensing DAV go high, pull down NDAC, preparing for the next data byte.

The process then repeats for successive data bytes.

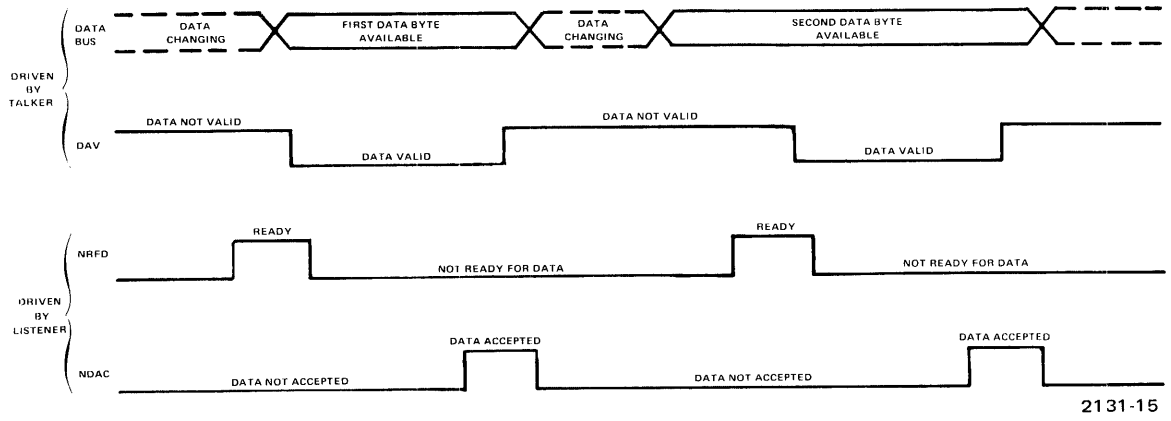


Figure 13-2. GPIB Transfer Bus Handshake Sequence.

GPIB Protocol: Addressing Devices for a Data Transfer

To cause a data transfer to occur, the 4025 (functioning as the GPIB controller) does the following:

1. First, it asserts ATN. This tells all other devices on the bus to stop whatever they are doing and pay attention to the bytes which the 4025 is about to send. While ATN is asserted, all devices on the bus must participate as "acceptors" in the data transfer handshaking. Each device examines the bytes it accepts, watching for its own *primary talk address* or *primary listen address*.

2. Next, the 4025 assigns one of the devices on the bus to be a listener. This it does by sending that device's primary listen address. (If the 4025 is to be the listener, it addresses itself internally, without sending its own listen address on the bus.)

3. Next, the 4025 assigns one of the devices to be a talker: it sends that device's primary talk address over the GPIB. (If the 4025 is to be the talker, it addresses itself internally, without sending its own talk address on the bus.)

4. The 4025 releases ATN. Only those devices which have been addressed as talker or listener are now permitted to take part in the data transfer.

   The device just addressed as talker then starts sending data bytes over the bus. The device addressed as listener (usually there is only one) receives the data. The data transfer is governed by the three-wire handshaking described earlier: the talker drives the DAV line and the listener drives NRFD and NDAC.

5. As the talker sends the last byte in the data sequence, it simultaneously drives the EOI (End or Identify) line low. This signals the 4025 that the data transfer is finished.

6. After the last data byte has been transferred, the 4025 pulls the ATN line low. While holding ATN low (true), it sends two bytes over the bus: the UNTALK and UNLISTEN bytes. These cause the devices currently addressed as talker and listener to "unaddress themselves." Once the UNTALK and UNLISTEN bytes have been sent, the 4025 turns off the ATN signal.

GPIB Code Chart

During ordinary data transfers over the GPIB, the 8-bit bytes are usually ASCII characters. When the 4025 is asserting ATN, however, the bytes transferred take on special meanings: they may be special commands (such as SPE, "serial poll enable"), or device addresses (such LA1, "primary listen address for device number 1"). These alternate meanings are listed in the GPIB Code Chart, Table 13-1.

Table 13-1

GPIB CODE CHART

DIO7 DATA BUS DIO6 DIO5 DIO4 DIO3 BITS DIO2 DIO1				0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
				ADDRESSED COMMANDS	UNIVERSAL COMMANDS	PRIMARY LISTEN ADDRESSES		PRIMARY TALK ADDRESSES		SECONDARY ADDRESSES	
0	0	0	0	NUL 0	DCE 16	SP LA0 32	0 LA16 48	@ TA0 64	P TA16 80	' SA0 96	p SA16 112
0	0	0	1	SOH GTL 1	DC1 LLO 17	! LA1 33	1 LA17 49	A TA1 65	Q TA17 81	a SA1 97	q SA17 113
0	0	1	0	STX 2	DC2 18	" LA2 34	2 LA18 50	B TA2 66	R TA18 82	b SA2 98	r SA18 114
0	0	1	1	ETX 3	DC3 19	# LA3 35	3 LA19 51	C TA3 67	S TA19 83	c SA3 99	s SA19 115
0	1	0	0	EOT SDC 4	DC4 DCL 20	$ LA4 36	4 LA20 52	D TA4 68	T TA20 84	d SA4 100	t SA20 116
0	1	0	1	ENQ PPC 5	NAK PPU 21	% LA5 37	5 LA21 53	E TA5 69	U TA21 85	e SA5 101	u SA21 117
0	1	1	0	ACK 6	SYN 22	& LA6 38	6 LA22 54	F TA6 70	V TA22 86	f SA6 102	v SA22 118
0	1	1	1	BEL 7	ETB 23	' LA7 39	7 LA23 55	G TA7 71	W TA23 87	g SA7 103	w SA23 119
1	0	0	0	BS GET 8	CAN SPE 24	(LA8 40	8 LA24 56	H TA8 72	X TA24 88	h SA8 104	x SA24 120
1	0	0	1	HT TCT 9	EM SPD 25	) LA9 41	9 LA25 57	I TA9 73	Y TA25 89	i SA9 105	y SA25 121
1	0	1	0	LF 10	SUB 26	* LA10 42	: LA26 58	J TA10 74	Z TA26 90	j SA10 106	z SA26 122
1	0	1	1	VT 11	ESC 27	+ LA11 43	; LA27 59	K TA11 75	[TA27 91	k SA11	{ SA27 123
1	1	0	0	FF 12	FS 28	, LA12 44	< LA28 60	L TA12 76	\ TA28 92	l SA12 108	! SA28 124
1	1	0	1	CR 13	GS 29	- LA13 45	= LA29 61	M TA13 77	] TA29 93	m SA13 109	} SA29 125
1	1	1	0	SO 14	RS 30	. LA14 46	> LA30 62	N TA14 78	^ TA30 94	n SA14 110	~ SA30 126
1	1	1	1	SI 15	US 31	/ LA15 47	? UNL 63	O TA15 79	_ UNT 95	o SA15 111	DEL 127

- Shaded codes are those usable by the 4025 with Options 4 and 36.

- LAn — Primary Listen Address for device n
- TAn — Primary Talk Address for device n
- UNL — UNLISTEN command
- UNT — UNTALK command
- SPE — SERIAL POLL ENABLE command
- SPD — SERIAL POLL DISABLE command

- KEY

ASCII Character
GPIB Code
Decimal

2830-123

I/O REGISTERS

The GPIB Interface has four I/O registers through which it communicates with the Processor. These registers and their functions are listed in Table 13-2.

Table 13-2

GPIB BOARD I/O REGISTERS

Register Name	Address	Function		
GPIB Board Status Word	X'0840'	Read	Bit 0:	LSC (Line Status Change) interrupts are enabled.
			Bit 1:	An LSC interrupt has occurred.
			Bit 2:	HAND interrupts are enabled.
			Bit 3:	A HAND interrupt has occurred.
			Bits 4-6:	Unused
			Bit 7:	An interrupt has occurred.
		Write	Bit 0:	Enable LSC interrupts.
			Bit 1:	Unused
			Bit 2:	Enable HAND interrupts.
			Bits 3-7:	Unused
Trigger Word	X'0842'	Read		Reading from this address clears any LSC interrupts.
		Write		Writing to this address sends the SHAKE signal to the Handshake Logic.
GPIB Data Word	X'0844'	Read	Bits 0-7:	Data byte received from the GPIB.
		Write	Bits 0-7:	Data byte to send on the GPIB.

Table 13-2 (cont)

GPIB BOARD I/O REGISTERS

Register Name	Address	Function		
GPIB Control Word	X'0846'	Read	Bit 0:	DAV (Data Valid)
			Bit 1:	EOI (End or Identify)
			Bit 2:	SRQ (Service Request)
			Bit 3:	ATN (Attention)
			Bit 4:	IFC (Interface Clear)
			Bit 5:	REN (Remote Enable)
			Bit 6:	NOBODY: True when "nobody is listening" on the GPIB. (Warns the firmware not to try to talk on the GPIB.)
			Bit 7:	HAND: True when a data transfer is awaiting Processor action. Depending on whether the interface is in input or output mode, the Processor should either read or write at the GPIB Data Word.
		Write	Bit 0:	HOLD: Set true to prevent completion of the current transfer bus handshake.
			Bit 1:	EOI (End or Identify)
			Bit 2:	SRQ (Service Request)
			Bit 3:	ATN (Attention)
			Bit 4:	IFC (Interface Clear)
			Bit 5:	REN (Remote Enable)
			Bit 6:	XMIT: Handshake Logic steering bit. Set true to send data bytes, false to receive them.
			Bit 7:	MYGPIBADR (My GPIB Address): Tells the Interface that it has been addressed as a talker or listener and should take part in GPIB data transfers.

GPIB FIRMWARE

The firmware controls the GPIB lines by reading and writing from the I/O registers just described. To illustrate the process, we'll consider three primitive operations: controlling and monitoring the GPIB lines, sending a data byte, and receiving a data byte. (More complicated procedures are made up of many of these primitive operations.)

Controlling and Monitoring the GPIB Lines

To control many of the GPIB signal lines, the firmware writes into the GPIB Control Word, X'0846'. For instance, to send the ATN signal, it writes a 1 into bit 3 of this word; to turn off ATN, it writes a 0 there.

Control Word bits 1 through 5 control the five management bus lines. Bit 0 drives the HOLD signal, which is sent when it is necessary to delay completion of a transfer bus handshake cycle. (HOLD forces NRFD true.) Bit 6 steers the Handshake Logic (described later in this section) between "transmit" (source handshake) and "receive" (acceptor handshake) modes. Bit 7 controls the MYGPIBADR signal, by which the firmware tells the hardware that it has been addressed as a listener or talker for an upcoming GPIB data transfer.

To monitor the state of the GPIB lines, the firmware reads repeatedly from the GPIB Control Word. Bits 1 to 5 of this word tell the states of the five management bus lines. Bit 0 tells the state of the DAV line on the transfer bus. Bit 6 provides a warning flag (the NOBODY signal) whenever there are no addressed listeners attached to the bus. Bit 7 (the HAND signal) goes true when action is required of the Processor in order to complete a transfer bus handshake.

NOTE

Circuitry is provided to generate Processor interrupts whenever the states of signal lines change (Line Status Change Interrupt), or whenever Processor action is needed to complete a transfer bus handshake (HAND Interrupt). However, this circuitry is unused; the Option 36 firmware disables GPIB Board interrupts and monitors the signal lines by repeatedly reading from the GPIB Control Word.

Sending a Data Byte

To send a data byte over the GPIB, the firmware does the following:

1. First, it writes into the GPIB Control Word, setting the XMIT bit true. This steers the Handshake Logic into its "transmit" mode.

2. Next, the firmware reads from the GPIB Control Word to be sure that the NOBODY signal is not being sent. (If NOBODY is true, then there are no listeners present on the GPIB, and the firmware aborts the attempt to send a data byte.)

3. Next, it checks the state of the HAND bit in the GPIB Control Word. HAND goes true when the GPIB "listeners" are ready for the 4025 to place a byte on the data bus.

4. When HAND goes true, the firmware writes a data byte into the GPIB Control Word. This places the data byte on the GPIB data bus.

5. After waiting for the data to settle, the firmware writes to the Trigger Word. This sends a SHAKE signal to the hardware. The hardware then completes the transfer bus handshake cycle. (When all the GPIB listeners have accepted the byte, HAND goes true again.)

6. The firmware repeats Steps 2 to 5 until all the data bytes have been sent.

Receiving a Data Byte

To receive a GPIB data byte, the firmware does the following:

1. First, it writes into the GPIB Control Word, setting the XMIT bit false. This steers the Handshake Logic into "receive" mode.

2. The firmware monitors the HAND bit in the GPIB Control Word. (The firmware repeatedly reads from this word to check its status.)

3. When the HAND bit goes true, the firmware reads the data byte from the GPIB Data Word and places that data byte in the appropriate input queue.

4. Next, it writes to the Trigger Word, sending the SHAKE signal. This causes the GPIB handshaking to continue. When the next byte from the talker is ready in the GPIB Data Word, the Handshake Circuitry will set the HAND bit true again.

5. Steps 2 through 4 are repeated for successive data bytes.

CIRCUIT DESCRIPTIONS

Refer to Figure 13-3, the GPIB Interface Board block diagram. The circuitry includes these circuit blocks: Bus Data Buffer, Address Decoder, GPIB Bus Transceivers, GPIB Data Buffer and Latch, GPIB Control Word (Write), Debouncers, HAND Decode, Hello Gate, GPIB Control Word (Read), ATN Mask, Handshake Steering, and Handshake Logic.

Also included are "interrupt control" circuit blocks: Line Status Change Detector, Interrupt Mask Logic, Interrupt Requestor, and the buffers which form the "read" half of the GPIB Board Status Word. *Since the Option 36 firmware disables GPIB Board interrupts, most of this circuitry is unused.*

Bus Data Buffer

The Bus Data Buffer (Schematic 18-1) consists of bidirectional tri-state buffers which interface the Mother Board data bus (BD0-BD7) with the GPIB Board's internal data bus (D0-D7). These buffers are steered by the PROCREAD (Processor Read) signal from the Address Decoder.

Address Decoder

The Address Decoder (Schematic 18-1) monitors the BA0-BA9, IOADR, READ, and WRITE lines, and detects when the Processor is reading or writing to the various I/O registers on the GPIB Interface Board. Its outputs are:

- **DWSTROBE.** Indicates a write to the GPIB Data Word.
- **DRSTROBE.** Indicates a read from the GPIB Data Word.
- **SHAKE.** Generated by a write to the Trigger Word; indicates that the Processor has performed the handshaking function requested by the HAND signal.
- **LSCCLR.** Generated by a read from the Trigger Word; causes the LSC (Line Status Change) interrupt bit to be cleared.
- **CWSTROBE.** Indicates a write to the GPIB Control Word.
- **CRSTROBE.** Indicates a read from the GPIB Control Word.

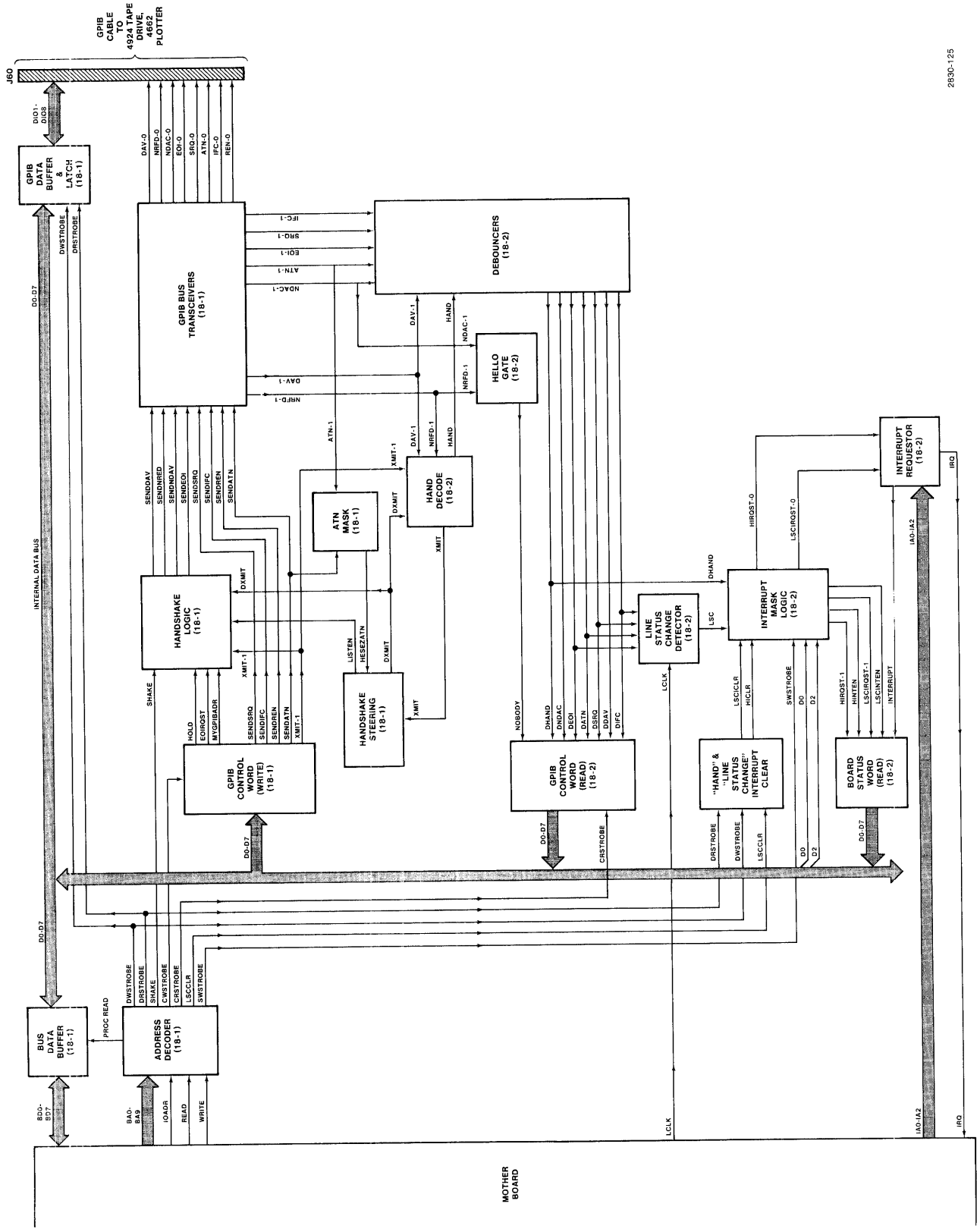


Figure 13-3. GPIB Peripheral Interface Board Block Diagram.

GPIB Transceivers

The GPIB Transcievers (Schematic 18-1, Figure 13-4) are special integrated circuits which interface the TTL logic on the GPIB Board with the active-low GPIB signal lines. Each has four terminals (A, B, C, D) which connect to the GPIB, four "incoming data" terminals (AI, BI, CI, DI) which drive on-board versions of the GPIB signals, and four "outgoing data" terminals (AO, BO, CO, DO) which drive the GPIB lines when enabled by a low at the "enable" pin.

GPIB Data Buffer and Latch

The GPIB Data Buffer and Latch (Schematic 18-1) includes the "read" and "write" halves of the GPIB Data Word, and GPIB transceiver ICs to interface them to the GPIB data bus.

The "read" half of the GPIB Data Word is an IC containing eight tri-state buffers. When the Processor reads from the GPIB Data Word, the DRSTROBE (Data Read Strobe) signal enables these buffers, placing the current GPIB data byte on the board's internal data bus. The Data Bus Buffers then relay the data to the BD0-BD7 Mother Board data lines.

The "write" half of the GPIB Data Word is an IC containing eight type D flip-flops. When the Processor writes to the Data Word, the DWSTROBE (Data Write Strobe) signal clocks these flip-flops, storing the data in them. The DXMIT signal (from the Handshake Steering logic) enables the GPIB Transceiver ICs, placing the data on the GPIB data bus.

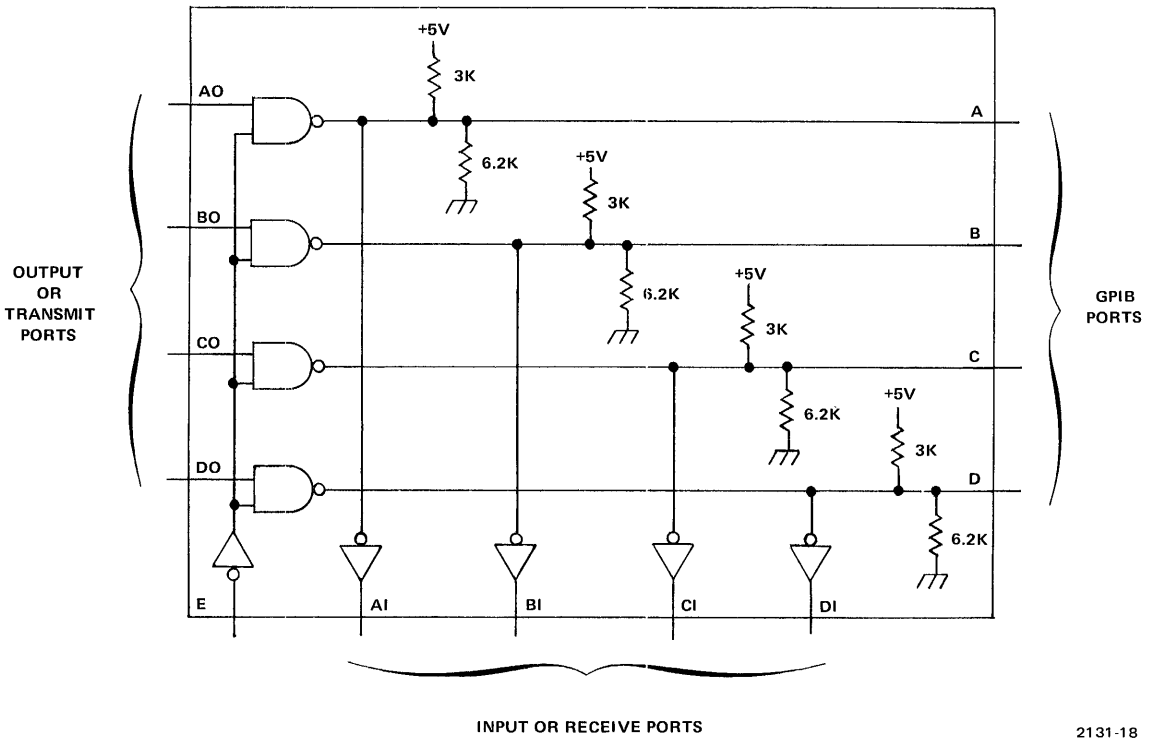


Figure 13-4. GPIB Transceiver.

"Write" Half of GPIB Control Word

The "write" half of the GPIB Control Word (Schematic 18-1) is an IC with eight type D flip-flops. When clocked by CWSTROBE (Control Word Write Strobe), the flip-flops store the data on the board's internal data bus. The flip-flops drive these signal lines:

- **HOLD.** A signal to the Handshake Logic, causing it to force the NRFD (Not Ready For Data) signal true. This prevents completion of the current transfer bus handshake.
- **EOIRQST (EOI Request).** Requests the Handshake Logic to send the EOI (End or Identify) signal on the GPIB. (If ATN is true, the EOI signal is suppressed.)
- **SENDSRQ, SENDATN, SENDIFC, SENDREN.** These signals directly drive the GPIB Transceivers, sending the SRQ, ATN, IFC, and REN signals.
- **MYGPIBADR (My GPIB Address).** A signal to the Handshake Logic that the 4025 is to take part as a talker or listener in data transfers.

Debouncers

Before incoming GPIB signals are passed on to the rest of the GPIB Board circuitry, they are "debounced" (removed of short-duration noise). The Debouncers (Schematic 18-2) for six of the signal lines use an MC14490 "hex contact bounce eliminator" IC; for the seventh signal line (IFC), a network of gates and smoothing capacitors is used. The debounced outputs are: DNDAC (Debounced Not Data Accepted), DEOI (Debounced End or Identify), DDAV (Debounced Data Valid), DATN (Debounced Attention), DSRQ (Debounced Service Request), DIFC (Debounced Interface Clear), and DHAND (Debounced HAND).

HAND Decode

The HAND Decode logic (Schematic 18-2) monitors two of the transfer bus handshake lines (NRFD and DAV).

When *sending* data (DXMIT true), HAND goes true when NRFD (Not Ready For Data) goes false. In this case, HAND tells the Processor, "The listeners are ready for data; go ahead and give them another data byte."

When *receiving* data (XMIT false), HAND goes true when DAV (Data Valid) goes true. In this case, HAND tells the Processor, "The talker has placed a valid data byte on the bus; go ahead and read the byte from the GPIB Data Word."

Hello Gate

The Hello Gate (Schematic 18-2) monitors the two lines NRFD and NDAC (driven by listeners in a GPIB data transfer). So long as there is a GPIB listener present, at least one of these signals will be true; but if neither of NRFD or NDAC is true, then "there's nobody out there listening," and the gate sends the NOBODY signal. The Processor reads the NOBODY bit (from the GPIB Control Word) before attempting to send data over the GPIB.

"Read" Half of GPIB Control Word

The "read" half of the GPIB Control Word (Schematic 18-2) consists of tri-state buffers. When enabled by the CRSTROBE (Control Word Read Strobe) signal, these buffers place bits on the data bus bits (a) to tell the state of several GPIB signal lines (EOI, DAV, ATN, SRQ, IFC, REN signals), (b) to warn when there are no listeners on the bus (NOBODY signal), and (c) to tell the Processor when action is required of it to complete a GPIB handshake (HAND signal).

ATN Mask

The ATN Mask gate (Schematic 18-1) monitors the GPIB ATN signal and the SENDATN line from the GPIB Control Word. If ATN is true, and it is *not* the 4025 which is sending it true (SENDATN is false), this gate sends HESEZATN (He Says Attention). This signal is used by the Handshake Steering logic.

NOTE

The Option 36 firmware requires that the 4025 be the only controller (and thus the only device allowed to send ATN) on the GPIB. Thus, HESEZATN should always be false.

Handshake Steering

The Handshake Steering logic (Schematic 18-1) enables the "source handshake" part of the Handshake Logic (by sending the DXMIT signal), if the firmware has asked to transmit data (XMIT true) and there is no other GPIB device sending the ATN signal (HESEZATN is false).

If an external controller sends ATN (HESEZATN is true) or if the 4025 is not sending data (XMIT is false), the logic enables the "acceptor handshake" part of the Handshake Logic (by sending LISTEN true).

NOTE

Since the 4025 is the only controller allowed on the GPIB, HESEZATN should always be false.

Handshake Logic

The Handshake Logic (Schematic 18-1) includes circuitry for performing the GPIB "source handshake" function when sending data bytes, and for performing the "acceptor handshake" function when receiving data bytes.

Source Handshake

The 4025 sends data bytes over the GPIB as follows:

1. When all listeners are ready for a data byte, the GPIB NRFD (Not Ready For Data) signal goes false. This is detected by the HAND Decode logic, which sets the HAND bit in the GPIB Control Word, inviting the Processor to place the data byte on the GPIB.

2. Meanwhile, the Processor has been reading repeatedly from the Control Word. When the HAND bit goes true, it writes into the GPIB Data Word, placing a data byte on GPIB data lines DIO1-DIO8.

3. After the data has settled, the Processor writes to the Trigger Word. This causes the Address Decoder to send the SHAKE signal to the Handshake Logic.

4. SHAKE sets the source handshake flip-flop (U270A in Schematic 18-1). With the Handshake Logic steered to "source handshake" mode (DXMIT true), the flip-flop output is passes through an AND gate, which sends SENDDAV to the GPIB Bus Transceivers. This causes the 4025 to send the DAV (Data Valid) signal on the GPIB.

5. The GPIB listeners set NRFD true again and read the data byte.

6. When the slowest listener has accepted the data, NDAC (Not Data Accepted) goes false. The debounced version of this signal (DNDAC) clears the source handshake flip-flop, causing the 4025 to stop sending DAV.

7. When all listeners are ready for the next data byte, NRFD goes false again. Steps 1 through 6 then repeat for successive data bytes.

Acceptor Handshake

The 4025 receives data bytes from the GPIB as follows:

1. When the talker has placed a byte on the GPIB data bus (DIO1-DIO8), it sends the DAV (Data Valid) signal. With the GPIB Board steered to "receive" mode (XMIT false), the HAND Decode logic sets the HAND bit in the GPIB Control Word.

   Several logic gates (U190D, U180A, U290A, U290D) send SENDNDAC as long as either DAV or HAND is true. This causes the 4025 to send the NDAC (Not Data Accepted) signal on the GPIB.

2. Meanwhile, the Processor has been reading repeatedly from the GPIB Control Word. When HAND goes true, it reads from the GPIB Data Word.

3. Having read the data, the Processor writes to the Trigger Word. This causes the Address Decoder to send the SHAKE signal to the Handshake Logic.

4. SHAKE sets the acceptor handshake flip-flop (U270B in Schematic 18-1). As the flip-flop is set, its inverting output goes false; this turns off the SENDNDAC signal. This causes the 4025 to stop sending NDAC (Not Data Accepted).

5. When all listeners have accepted the data, the GPIB NDAC line goes false (high). This tells the talker that the listeners have the data. The talker then turns off DAV.

6. When DAV goes false, the HAND Decode logic stops sending the HAND signal. With both HAND and DAV false, logic gates in the Handshake Logic stop sending the SENDNRFD signal (unless the HOLD bit in the GPIB Control Word is set).

   Also, as DAV goes false, its debounced version (DDAV) clears the acceptor handshake flip-flop. With the flip-flop cleared, SENDNDAC goes true, and the 4025 sends the NDAC (Not Data Accepted) signal on the GPIB.

7. When all listeners have stopped sending NRFD, the talker places another data byte on the GPIB and sends DAV true again. Steps 1 through 6 then repeat for successive data bytes.

Interrupt Control Circuitry

The GPIB Board's interrupt control circuitry includes: Line Status Change Detector, Interrupt Mask Logic, Interrupt Requestor, and the "read" half of the GPIB Board Status Word. This circuitry provides for two types of Processor interrupts (Line Status Change Interrupts and HAND Interrupts).

NOTE

The Option 36 firmware disables these interrupts. Therefore, most of this circuitry is unused.

LSC Detector

The LSC Detector (Schematic 18-2) is unused.

Interrupt Mask Logic

The Interrupt Mask Logic (Schematic 18-2) includes interrupt latches (flip-flops U360A and U360B) and latches for the interrupt enable bits (U220A and U220B). The firmware disables interrupts by writing zeroes into bits 0 and 2 of the GPIB Status Word. This causes the Address Decoder to send SWSTROBE (Status Word Strobe), loading the zeroes (bits 0 and 2 of the word written) into the interrupt enable latches. This sets the HINTEN (HAND Interrupt Enable) and LSCINTEN (Line Status Change Interrupt Enable) bits false. With HINTEN and LSCINTEN false, the HAND and LSC interrupts are both disabled.

"Read" Half of Board Status Word

The "read" half of the Board Status Word (Schematic 18-2) is unused.

Interrupt Address Decoder

With interrupts disabled, the Interrupt Address Decoder is unused.

Section 14

OPTIONS 10 AND 11 — POLLING

INTRODUCTION

This section provides a brief overview of the TEKTRONIX 4020 Series polling system using the Option 10 Polling Interface and Option 11 Polling Controller. It does not go into deep detail; for more information, see the 4020 Series Polling Reference Manual. For installation procedures, see Volume 2 of this Service Manual.

Tektronix Polling System

Figure 14-1 shows the polling system. The host computer is connected (via a telephone line and modems) to one or more Tektronix polling arrays. (Figure 14-1 shows a host connected to two such arrays.) In each polling array, several 4024, 4025, or 4027 terminals are connected in series using a current loop cable much like the cable used with Option 2. (However, this cable's connectors are wired differently than those used with Option 2.)

Option 10 Polling Interface

Each terminal (called a "display station") has an Option 10 Polling Interface. The Option 10 Interface includes the following:

- A Current Loop Interface Board like that supplied with Option 2.
- A ROM providing firmware to allow the terminal to cope with the polling configuration. This ROM replaces one of the ROMs on the 4024 Processor/Comm Board or 4025 Processor Board.

Option 11 Polling Controller

One of the terminals (a 4025 or 4027) has an Option 11 Polling Controller installed. The Polling Controller borrows power from that terminal; functionally, however, the Polling Controller is an entirely separate device.

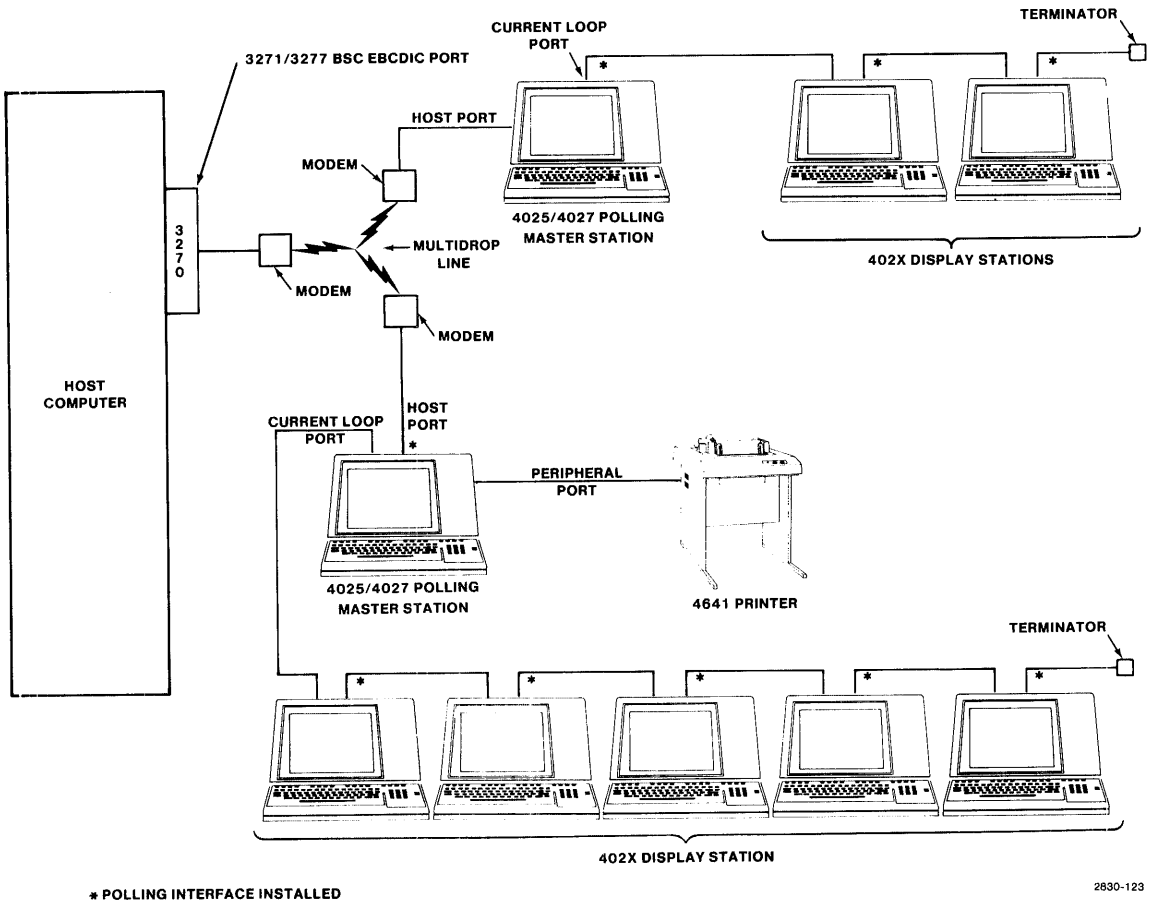


Figure 14-1. 4020-Series Polling System.

The Polling Controller supervises the flow of data between the different display stations and the computer. When the computer has a message for one of the display stations, it sends that message to the Polling Controller. The message is in a standard IBM message format, and is sent using the EBCDIC telegraph code. The Polling Controller relays the message to the particular display station, converting it to a Tektronix message format using the ASCII telegraph code.

The Polling Controller also controls the flow of messages from the individual display stations to the computer. It "polls" each display station in turn, to see if that display station has a message for the computer. It converts such messages from the Tektronix message format and the ASCII telegraph code to the IBM message format and the EBCDIC telegraph code, and relays them the host computer.

The Polling Controller communicates with the host through its own RS-232 interface. It communicates with the display stations (including the one in which it is installed) using its own interface and the polling cable. Each display station (including the one in which the Option 11 Polling Controller is installed) interfaces to the current loop through an Option 10 Polling Interface.

THE CURRENT LOOP AND OPTION 10

Figures 14-2 and 14-3 show how the Polling Controller and the display stations are connected on the polling cable. Figure 14-4 shows the cable in more detail, as well as a block diagram for the Current Loop Interface through which each terminal is connected to the polling cable. Like the "current loop cable" used with Option 2, this "polling cable" has two separate circuits: a "transmit loop" and a "receive loop." The transmit loop carries data from the display stations to the Polling Controller, while the receive loop carries data from the Polling Controller to the display stations.

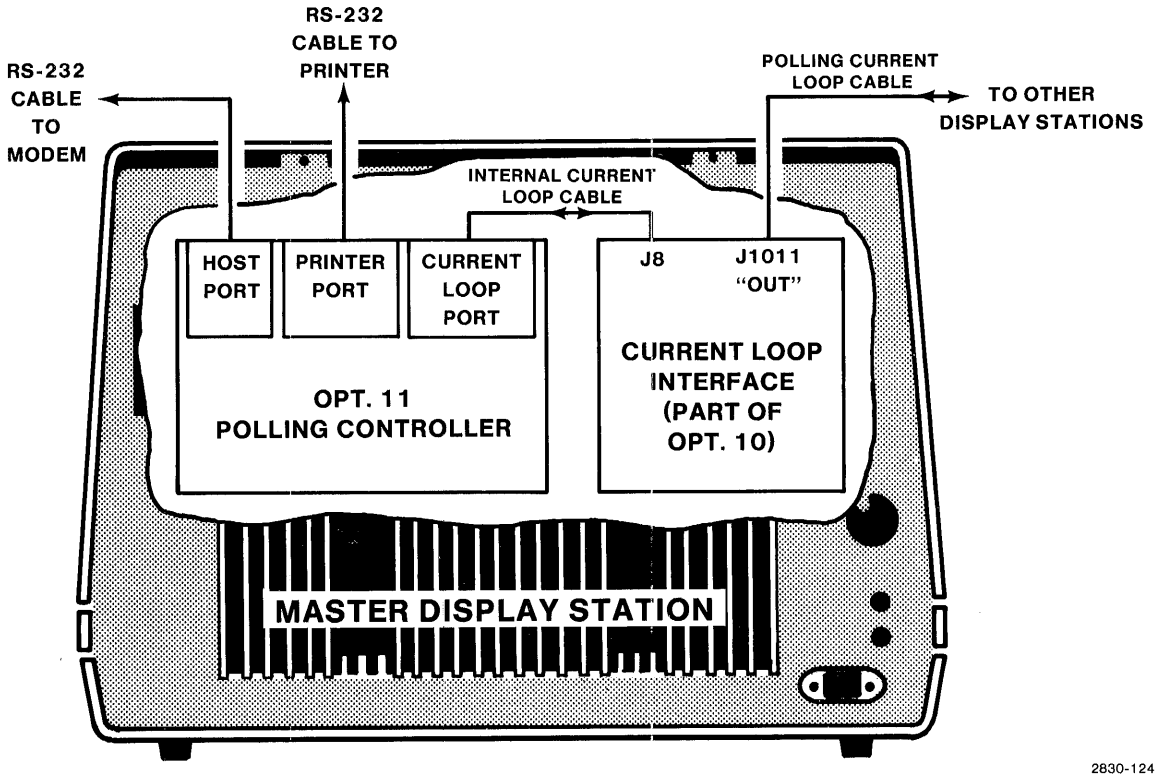


Figure 14-2. Master Display Station Interconnections.

NOTE

The "polling cable" used with Option 10 and Option 11 has its connectors wired differently from the "current loop cable" used with Option 2.

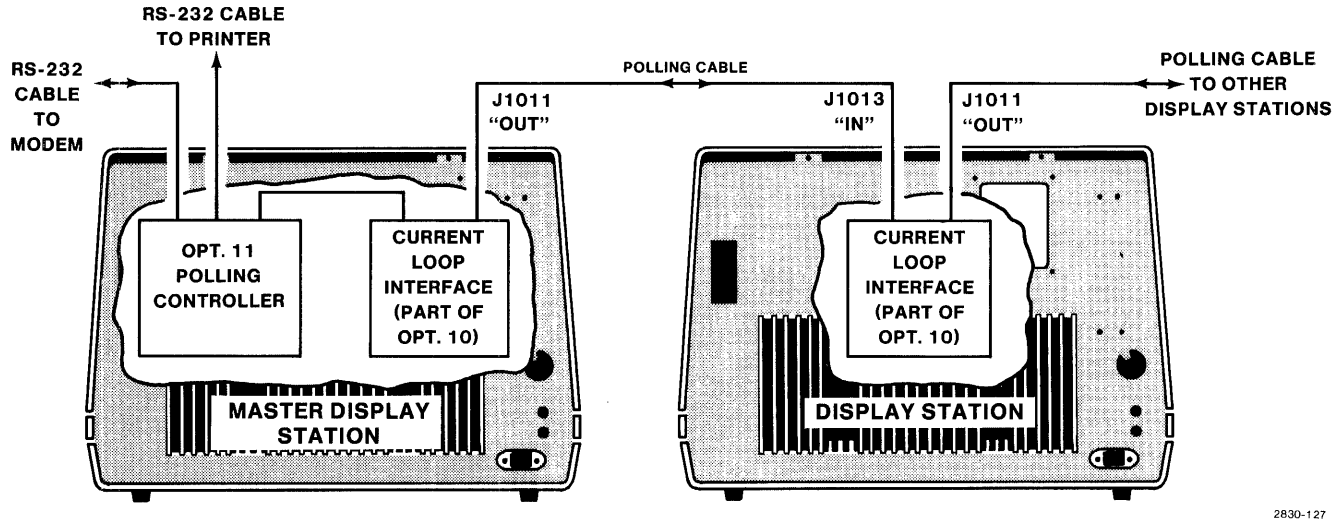


Figure 14-3. Display Station Interconnections.

Refer to Figure 14-4. The Polling Controller controls the flow of current in the receive loop. In each terminal, the Current Loop Interface (part of Option 10) has a receiving circuit which responds to the receive loop current, driving the terminal's RDATA line. Current flowing in the loop is interpreted as a *mark*, or binary one; the absence of current is interpreted as a *space*, or binary zero. Binary ones send RDATA negative; binary zeroes send RDATA positive.

Any terminal can send data on the transmit loop to the Polling Controller. (However, only one terminal may do so at a time. When a terminal is not transmitting to the controller, it holds its transmit loop keying circuit closed; this provides a complete circuit through the remaining terminals and back to the Polling Controller.) In each terminal, the Current Loop Interface has a transmit loop keying circuit which responds to the TDATA line from the terminal. A negative voltage on TDATA is interpreted as a mark or binary one, closing the "key" and permitting current to flow in the transmit loop. A positive voltage is interpreted as a space or binary zero, opening the "key" and shutting off the flow of current. All terminals except the one permitted to transmit hold their TDATA lines in mark condition.

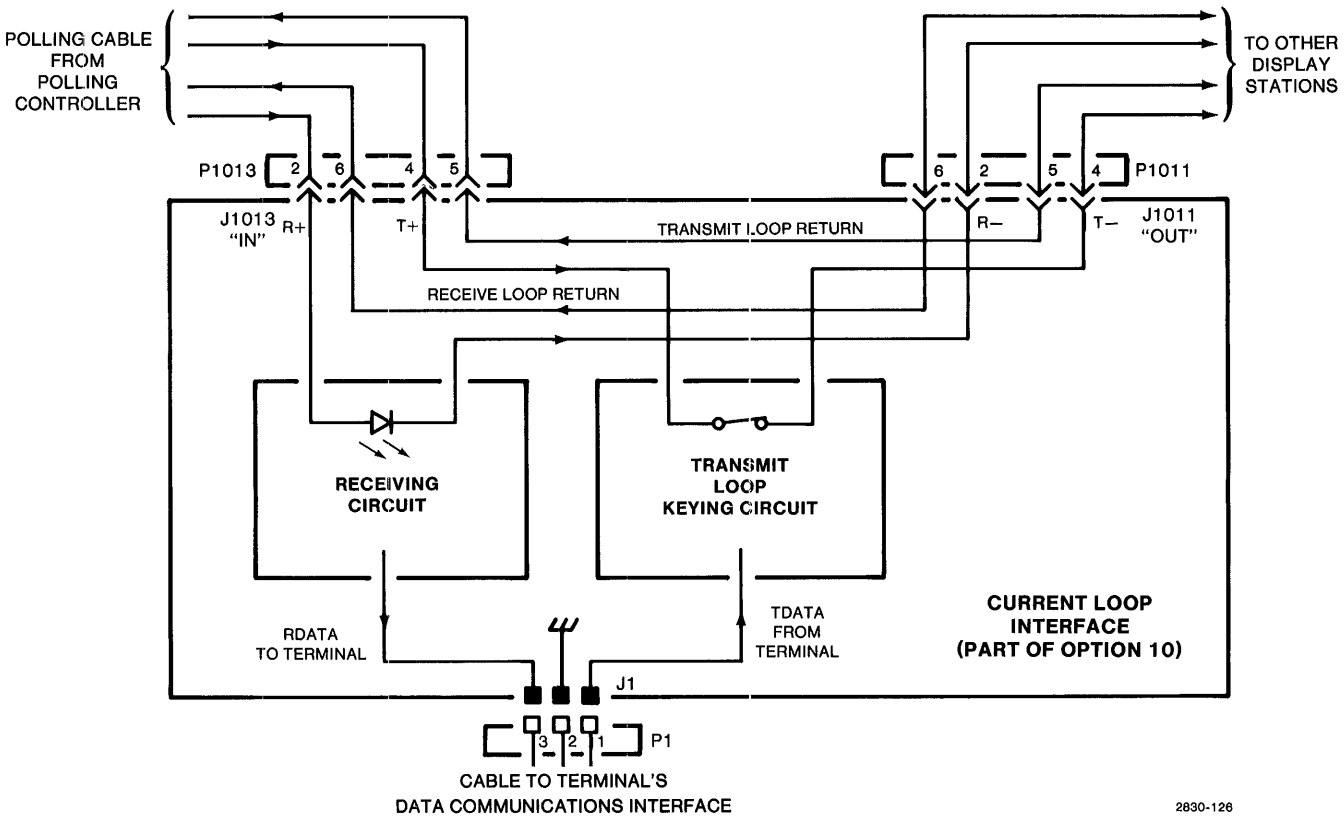


Figure 14-4. Current Loop Interface, as Used with Option 10.

The Current Loop Interface Board (part of Option 10) is the same as that provided with Option 2. For more information on its circuitry, see Schematic 12-1. When used in the polling loop, the board is strapped to PASSIVE mode for both the transmit and receive loops. (The Polling Controller's internal current loop interface provides the current source for the current loops.)

Note that when the Polling Controller sends a message to any one of the terminals on the loop, all the other terminals also hear that message on their RDATA lines. Each terminal's polling firmware (part of Option 10) ignores those messages not addressed to it. Likewise, the Option 10 polling firmware prevents the terminal from sending data on the "transmit" loop except when enabled to do so by the Polling Controller.

Section 15

OPTIONS 23 TO 26 — GRAPHICS MEMORY (4025 ONLY)

INTRODUCTION

This section describes the operation of the Graphics Memory Board. For installation information, see Volume 2 of this Service Manual.

Parts Included

Options 23, 24, 25, and 26 consist of the following:

- A ROM holding the graphics firmware. This ROM is installed on the Option 35 ROM Expansion Board.
- The Graphics Memory Board. This board holds the RAMs in which graphic character information is stored, and support circuitry for those RAMs. Also included are I/O registers through which the Processor can cause dots to be turned on or off in the character cells stored in the graphics RAMs.
- Varying amounts of RAM memory installed on the Graphics Memory Board. Option 23 includes 4K of graphics RAM memory; Options 24, 25, and 26 include 8K, 16K, and 32K of memory, respectively.

Note that Options 23 to 26 include both firmware (the graphics ROM) and hardware (Graphics Memory Board). Each of these is discussed in turn.

4025 GRAPHICS CONCEPTS: WHAT THE FIRMWARE DOES

To draw a vector on the 4025's screen, the operator (or host computer) sends commands to the terminal. As a minimum, these must include a GRAPHIC command (to set up a graphic area in the workspace), and a VECTOR command (to draw a line in that graphic area).

For instance, the user might type these two commands:

```
!GRAPHIC 2, 11, 20, 49 <CR>
!VECTOR 0,0 10,10 50,100 <CR>
```

("<CR>" represents pressing the RETURN key; "!" is the command character.)

GRAPHIC Command

To execute the GRAPHIC command, the microprocessor re-arranges its display list, which is recorded in RAM on the Display Memory Board. In doing so, it creates a *graphic area*: a block of adjacent character cells ("graphic cells").

Figure 15-1 illustrates the graphic area that would be created in response to a !GRAPHIC 2,11,20,49 command. The graphic area occupies rows 2 through 11, columns 20 through 49: a total of 300 character cells. Also shown is typical line which might be drawn in that graphic area in response to a !VECTOR command.

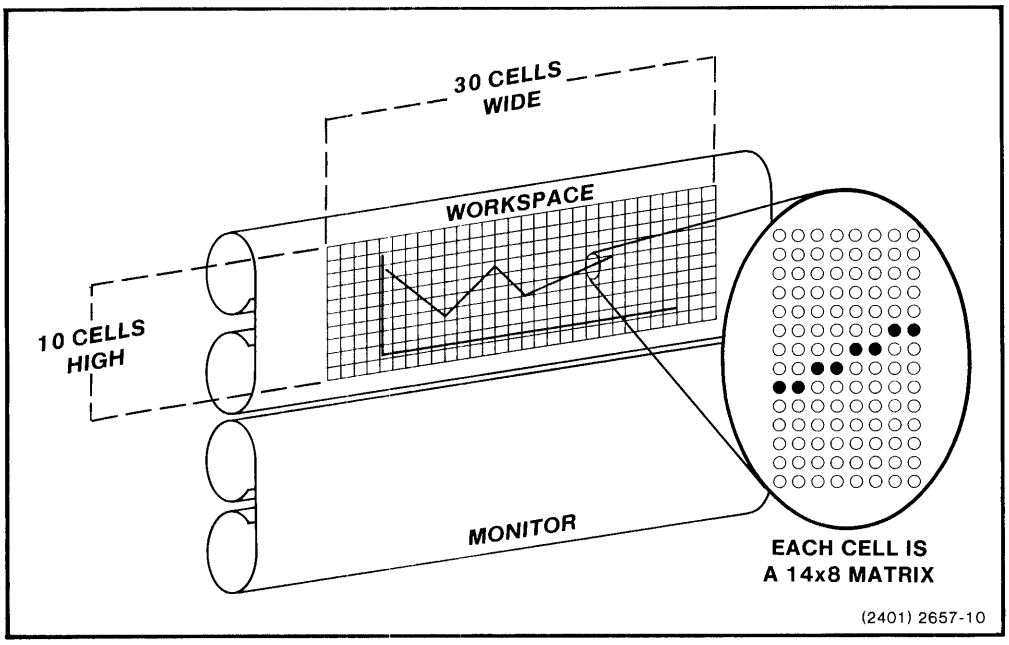


Figure 15-1. A Graphic Area.

Each graphic cell has *two* bytes in the display list. The first byte is a character font field attribute code, and the second byte is an ASCII character. The field attribute code specifies one of up to 32 character fonts. The ASCII character specifies one of the dot patterns (characters) stored in that font. Initially, each graphic cell is set to font zero, character number 32: the standard character font, displaying an ASCII "space" character.

VECTOR Command

Once a graphic area is defined, the user may issue VECTOR commands. These draw lines on the screen by changing the dot patterns for the graphic character cells. The dot patterns are stored in RAMs on the Graphics Memory Board; thus the VECTOR command (unlike the GRAPHIC command) involves activity on that board. To execute a VECTOR command, the Processor does the following:

1. First, it calculates through which cells of the graphic area the vector will pass.

2. For each such cell, it reserves a character in one of the high-numbered fonts in the Display Controller's Character Set Memory. Normally, character zero of font 31 is used first; then characters 1 through 31 of font 31 are used. After that, the Processor uses font 30, and so forth.

   (Of course, if a graphic cell already has a vector drawn through it, then it will already have been assigned a character in Character Set Memory. The Processor does not change that assignment.)

3. The Processor computes which dots should be turned on in each graphic character cell.

4. Next, it must turn on the appropriate dots in that character's dot pattern. This involves writing into the high-numbered character fonts, which reside on the Graphics Memory Board.

   However, the Processor cannot write directly into those fonts, because the Character Set Memory does not lie in the Processor's memory address space. (In other words, the Processor's address bus does not drive the CA0-CA6, FA0-FA4, and RA0-RA3 lines, which are the Character Set Memory address lines.)

   However, there are I/O registers on the Graphics Memory Board which do lie in the Processor's address space. The Processor writes into these I/O registers; this causes circuitry on the Graphics Memory Board to alter the dot patterns in the specified character cells.

5. Once it has turned on the correct dots in a character's dot pattern, the Processor links that character into the display list. For instance, it might change the display list so that a cell which formerly was displayed as "font zero, character 32" might now be be displayed as "font 31, character 16."

   Displaying the graphic character is done automatically by the Tracker and the Display Controller.

I/O Registers

The Processor controls the dot patterns stored in graphics memory by writing into registers on the Graphics Memory Board. These registers are:

- **Graphics Memory Status Word (X'0870').** By writing into the Status Word, the Processor enables or disables interrupt requests from the Graphics Memory Board. By reading from this word, it learns the states of several signal lines on the board.

- **Font Address Word (X'0871').** The Processor writes into the Font Address Word to specify which character font is to be accessed.

- **Character Address Word (X'0872').** The Processor writes into the Character Address Word to specify which character in that font is to be accessed.

- **Row Address Word (X'0873').** The Processor writes into the Row Address Word to specify a particular row of dots in that character's dot matrix.

- **Dot Data Word (X'0874').** The Processor writes into the Dot Data Word to specify the the dot pattern in one row of one character's dot matrix. It reads from that word to learn the current dot pattern for that row of the dot matrix.

- **Read Trigger Word (X'0875').** The Processor initiates a "read" operation by writing into the Read Trigger Word. This causes the Graphics Memory Board to read from the location specified in the Font, Character, and Row Address Words. When the read operation is complete, the data appears in the "read" half of the Dot Data Word and the Graphics Memory Board requests a Processor interrupt.

- **Write Trigger Word (X'0876').** The Processor initiates a "write" operation by writing into the Write Trigger Word. This causes the Graphics Memory Board to take the data in the "write" half of the Dot Data Word and record it at the location specified in the Font, Character, and Row Address Words. When the write operation is complete, the Graphics Memory Board requests a Processor interrupt.

Table 15-1 lists the functions of the Graphics Memory I/O registers.

Table 15-1

GRAPHICS MEMORY BOARD I/O REGISTERS

Address	Register Name	Functions	
X'0870'	Graphics Memory Status Word	Write	Bit 0: INTEN (Interrupt Enable) Bits 1-7: Unused
		Read	Bit 0: Unused Bit 1: READY Bits 2-3: Unused Bit 4: R-1/W-0 Bit 5: SCRUNCH Bit 6: Unused Bit 7: IFLAG (Interrupt Flag)
X'0871'	Font Address Word	Write	Bits 0-4: FAD0-FAD4 (Font Address) Bits 5-7: Unused
		Read	Bits 0-7: Unused
X'0872'	Character Address	Write	Bits 0-6: CAD0-CAD6 (Char Address) Bit 7: WSCRUNCH
		Read	Bits 0-7: Unused
X'0873'	Row Address Word	Write	Bits 0-3: RAD0-RAD3 (Row Address) Bits 4-7: Unused
		Read	Bits 0-7: Unused
X'0874'	Dot Data Word	Write	Bits 0-7: Dot Pattern Data
		Read	Bits 0-7: Dot Pattern Data
X'0875'	Write Trigger Word	Write	Writing here causes a write into Character Set RAM
		Read	Bits 0-7: Unused
X'0876'	Read Trigger Word	Write	Writing here causes a read from Character Set RAM
		Read	Bits 0-7: Unused

Firmware Details

Now that we've described the I/O registers, let's see how the firmware uses them. We'll consider the following procedures: initialization on power-up, drawing a vector, and erasing the graphic area. Then we'll look in more detail at how the firmware writes dot patterns into characters of graphic character fonts, and how it reads dot patterns from those characters.

Initialization

When the 4025 is powered-up, or when MASTER RESET is pressed, among other things the firmware does the following:

1. It tests to see how much graphics memory is installed.

2. For all the graphic memory present, the firmware clears the first 15 bytes of each character cell. (Because of the graphics memory addressing scheme, each character cell occupies 16 adjacent bytes of memory. The first 14 bytes hold dot patterns for the 14 rows of the character's dot matrix. The 15th byte is used as a flag to indicate whether that character has been referred to in the display list. The 16th byte is unused.)

Drawing a Vector

To draw a vector (execute a VECTOR command), the firmware does the following:

1. It determines where in the display to turn on dots. That is, it ascertains through which cells of the graphic area the vector will pass.

2. For each cell through which the vector passes, the firmware looks in the display list to see which font and character are displayed there.

3. If the character displayed there is not a character in one of the graphic character fonts, the firmware finds an unused graphic character and puts its font number and character number in the display list. It sets that character's 15th byte to indicate that the character is now being used in the display.

4. For each row of dots in the character, the firmware then:

   (a) reads the row of dots,

   (b) adds new dots to the row just read, and

   (c) writes the altered row of dots back into the byte of graphics memory.

Erase

To execute an ERASE G (erase graphic area) command, the firmware goes through the display list for the graphic area; wherever in it finds a graphic character in the display list, it replaces that character with font 0, character number 32: the ASCII space character. Each such graphic character removed from the display list is also returned to the "pool" of available graphic characters for use in drawing new vectors. This is done by clearing the 15th of the 16 bytes of graphics memory reserved for that character. (This 15th byte holds a flag telling whether the character has been used to hold graphic information. Clearing this byte makes the character available for re-use.)

An ERASE W (erase workspace scroll) command, on the other hand, does not pay particular attention to the graphic area. It erases the workspace part of the display list, but does not go into graphics memory and release character cells so that they may be used for drawing new vectors. It is, therefore, advisable to use ERASE G, rather than ERASE W, for erasing the graphic area.

Character Read Cycle

In the procedures just described, the firmware must often read a byte from graphics memory. It performs the read as follows:

1. First, it waits for the READY flag. That is, it repeatedly reads from the Graphics Memory Board Status Word (address X'0870') until the READY bit goes true.

2. The firmware writes the font number to be accessed into the Font Address Word (X'0871').

3. It writes the character number (within that font) into the Character Address Word (X'0872').

4. It writes the row number (within that character) into the Row Address Word (X'0873'). (Rows 0-13 specify rows of dots in the character's dot matrix. Row 14 holds a flag indicating whether the character has been linked into the display list. Row 15 is unused.)

5. The firmware writes to the Read Trigger Word (X'0875').

   This causes the Graphics Memory Board to read from graphics memory and place the byte it reads in the Dot Data Word. When the read operation is complete, the Graphics Memory Board sets READY true and requests an interrupt from the microprocessor.

6. When the interrupt is received, the firmware reads from the Status Word (X'0870') to be sure that READY is true.

7. After READY is true, the firmware reads the data from the Dot Data Word (address X'0874').

Character Write Cycle

When the firmware must write a dot pattern into a byte of graphics memory, it does the following:

1. First, the firmware waits for the READY flag.

   (If the firmware has just read from the location to which it will be writing, then the location's address is already set up in the board's I/O registers. In that case, the firmware omits the following three steps and goes on to Step 5.)

2. The firmware writes the font number into the Font Address Word (X'0871').

3. It writes the character number into the Character Address Word (X'0872').

4. It writes the row number into the Row Address Word (X'0873').

5. At this point, the Font Address, Character Address, and Row Address words hold the address of the byte of graphics memory to be accessed. The firmware now writes a dot pattern into the Dot Data Word (X'0874').

6. With the data in the Dot Data Word, the firmware writes to the Write Trigger Word (X'0876'). This triggers the circuitry on the Graphics Memory Board, causing that circuitry to perform a "write" operation.

GRAPHICS MEMORY BOARD CIRCUITRY

Block Diagram

The Graphics Memory Board (Figure 15-2) consists of the following major blocks: Dot Pattern Memory, Scrunch Memory, RAM Control, RAM Cycle Requestor, RAM Address Circuitry, Timing Circuitry, and Processor Interface.

The Dot Pattern Memory, as its name implies, holds the dot patterns for characters in the graphic character fonts.

The Scrunch Memory holds one bit for each character in a graphic font. This bit, the *scrunch bit*, designates whether the character is to be displayed as a graphic character (using the Display Controller's 12 MHz clock), or as an alphanumeric character (using the 14 MHz "scrunch clock").

The RAM Control Circuitry is a state machine which provides waveforms to drive the RAMs during "read," "write," or "refresh" cycles.

The RAM Cycle Requestor provides control signals which cause the Display Memory Board to execute three kinds of memory access cycles: CSM cycles, I/O cycles, and refresh cycles. The CSM (Character Set Memory) cycles each involve a "read" from the Dot Pattern and Scrunch Memories, to provide the Display Controller with the data it needs to display a character on the screen. The I/O cycles involve reads or writes at the request of the Processor; it is by these cycles that the contents of the Dot Pattern and Scrunch Memories are changed. The Refresh cycles periodically keep the RAMs from "forgetting" their data.

The RAM Address Circuitry sets up the address in RAM for the part of memory being accessed. This involves selecting the address from one of three sources, depending on the type of RAM access cycle being requested. The addresses for CSM cycles come from the Character Set Memory address lines (FA0-FA4, CA0-CA6, RA0-RA3). For I/O cycles, the RAM addresses come from the I/O registers in the Processor Interface. For refresh cycles, the addresses come from a counter within the RAM Address Circuitry.

The Timing Circuitry provides timing signals for the rest of the Display Memory Board.

The Processor Interface Circuitry holds the I/O registers through which the Processor communicates with the Graphics Memory Board. Also included are: "handshaking" flip-flops to control the exchange of data between the Processor and Graphics Memory, and circuitry to request Processor interrupts.

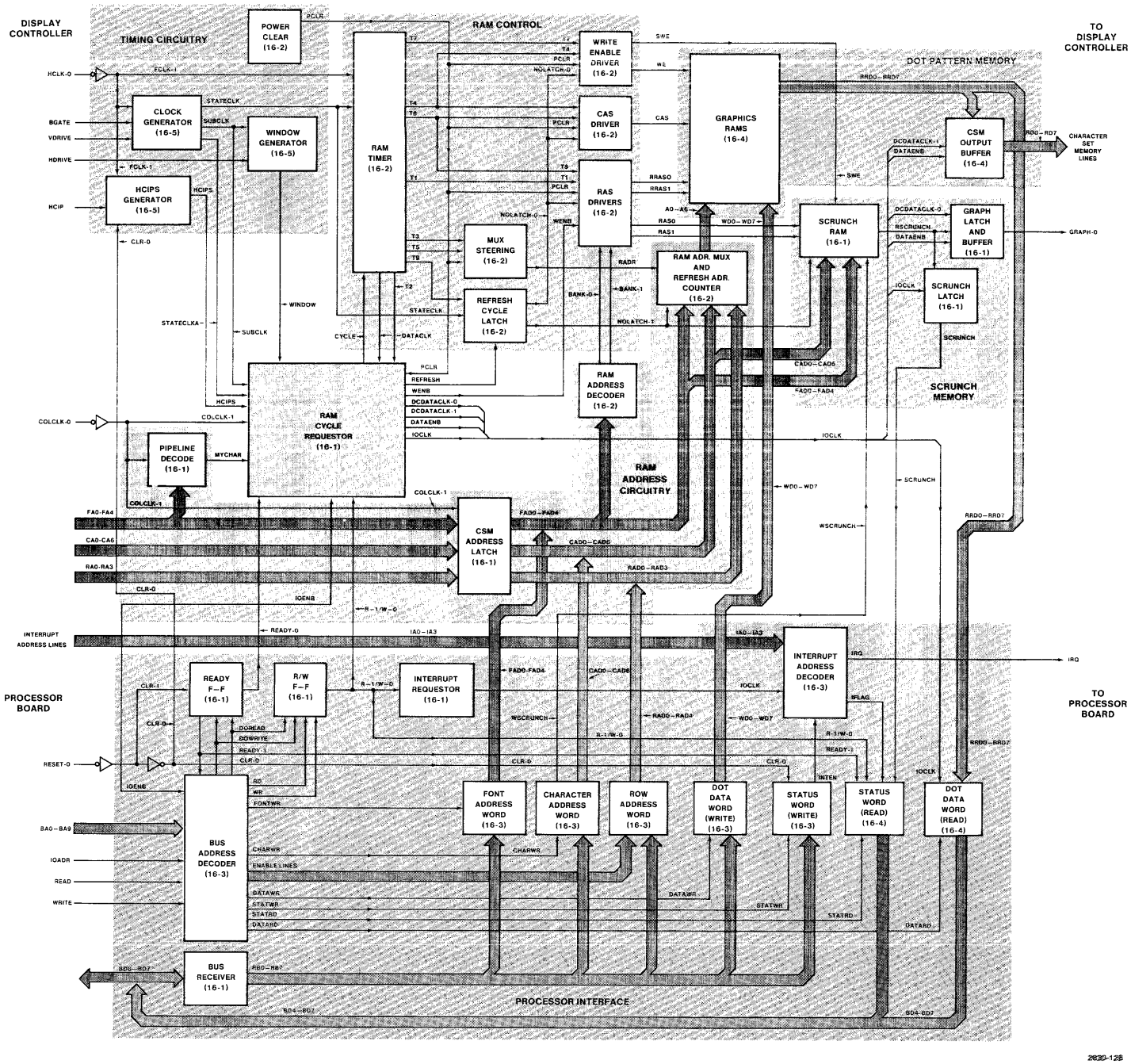


Figure 15-2. Graphics Memory Board Block Diagram.

Dot Pattern Memory

The Dot Pattern Memory consists of the Graphics RAMs and the CSM Output Buffer, both of which appear in Schematic 16-4. The Graphics RAMs hold the dot pattern data, while the CSM (Character Set Memory) Output Buffer places data from those RAMs on the Character Set Memory data lines, RD0-RD7.

Graphics RAMs

(See Schematic 16-4)

The Graphics RAMs consist of either eight or sixteen devices, each holding either 4K or 16K bits, as follows:

- 4K of Graphics Memory (Option 23): 8 4Kx1 devices to provide 4096 8-bit words.
- 8K (Option 24): 16 4Kx1 devices to provide 8192 words.
- 16K (Option 25): 8 16Kx1 devices to provide 16,384 words.
- 32K (Option 26): 16 16Kx1 devices to provide 32,768 words.

These RAMs appear in Schematic 16-4. Figure 15-3 shows their pinouts, and Figure 15-4 shows the waveforms necessary to drive them. These devices are like those on the Display Memory Board, and they are driven in a similar manner.

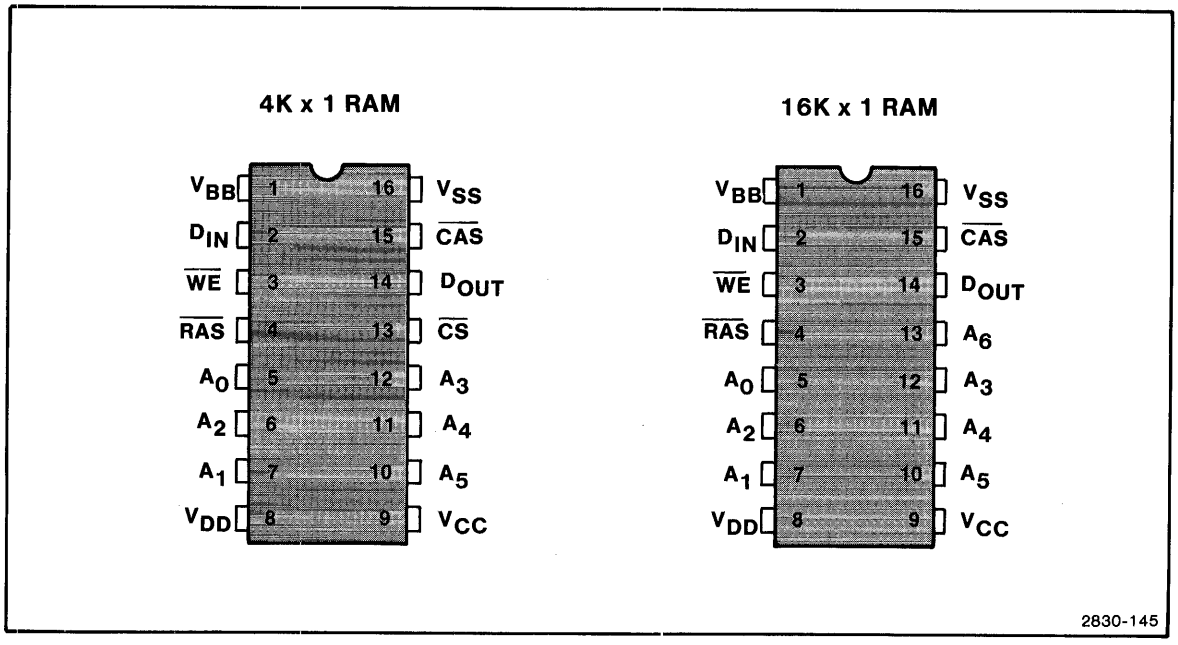


Figure 15-3. Pinouts for 4K and 16K RAMs.

Characters

msb | 0 | C C C C C C C | lsb

C = Character (0 to 127)

Spare

| 1 0 0 | X X X X X |

X = Reserved

Tracker Control

| 1 0 1 0 0 | F F F |

F = Function

0 - EOL (End of Line)
1 - ZOP (Zero No-op)
2 - SOP (Super No-op)
3 - MOP (Maybe No-op)
4 - Not Used
5 - Not Used
6 - NOP
7 - Not Used

Visual Field Attribute

| 1 0 1 0 1 | U E I |

U = Underline (0 - Off, 1 - On)
E = Background (0 - Normal, 1 - Enhanced)
I = Video (0 - Normal, 1 - Inverted)

Character Set

| 1 1 0 | C C C C C |

C = Character Set 0 to 31

Jump

| 1 1 1 | A A A A A |
| B B B B B B B B |

A,B = Jump Address - the tracker will jump to the memory address:

| 1 A A A A A B B B B B B B B 0 0 |

Notes: In addition to the above instructions, the first character code after EOL will be interpreted as the location of the cursor on the line being built. Values of 0 to 79 will put the cursor in columns 0 to 79. Values of 80 to 127 will put NO cursor on that line.

An EOL will set Intensity to NORMAL, Underline OFF, and Character Set to 0 (standard ASCII set).

2830-87

Figure 15-4. RAM Waveforms.

The RAMs are organized in two banks of eight devices each. Bank 0 uses the RRAS0 (RAM Row Address Strobe, bank 0) for its row address strobe, while Bank 1 uses RRAS1. The two banks share their other inputs: CAS (Column Address Strobe), A0-A6 (address lines), WE (Write Enable), WD0-WD7 (Write Data lines), and RRD0-RRD7 (RAM Read Data lines). The two banks of RAM sockets are filled with RAMs according to the amount of graphics memory installed, as follows:

- 4K of Graphics Memory (Option 23): Bank 0 is filled with 4K RAMs. Bank 1 is left empty.
- 8K of Graphics Memory (Option 24): Banks 0 and 1 are both filled with 4K RAMs.
- 16K of Graphics Memory (Option 25): Bank 0 is filled with 16K RAMs. Bank 1 is left empty.
- 32K of Graphics Memory (Option 26): Banks 0 and 1 are both filled with 16K RAMs.

CSM Output Buffer

(See Schematic 16-4)

The CSM (Character Set Memory) Output Buffer is a type 74LS374 integrated circuit. This device includes a set of eight type D latches to save the RAM outputs at the end of a CSM RAM cycle. It also includes eight tri-state buffers which place the latched data on the Character Set Memory data lines (RD0-RD7) at the appropriate time. The type D latches are clocked by DCDATACLK (Display Controller Data Clock), while the tri-state output buffers are enabled by the DATAENB (Data Enable) signal. Both these signals come from the RAM Cycle Requestor.

Scrunch Memory

For each character in a graphic character font, there is a *scrunch bit* which specifies whether the character is to be displayed as an alphanumeric character (using the Display Controller's 14 MHz scrunch clock), or as a graphics character (using the 12 MHz clock). These scrunch bits are stored in the Scrunch Memory.

The Scrunch Memory address lines (Figure 15-2) are the on-board versions of the font address and character address lines: FAD0-FAD4 and CAD0-CAD6. As with the Dot Pattern Memory, there are two banks of RAM, but here each bank consists of only a single device: a type 2102 2Kx1 dynamic RAM.

Data to be written into Scrunch Memory comes on the WSCRUNCH (Write Scrunch data) line from the Processor Interface.

The bit read from Scrunch Memory is saved in one of two latches, depending on whether it is destined for the Display Controller or the Processor. If the current memory cycle is a CSM cycle, then the scrunch data bit is intended for the Display Controller; it is latched into the GRAPH latch by the DCDATACLK signal, and placed on the GRAPH output line by the DATAENB signal. If the current memory cycle is an I/O cycle, then the scrunch bit is intended for the Processor; it is latched into the Scrunch Latch by the IOCLK signal, and saved there for the Processor to read from the "read" half of the Status Word.

RAM Control Circuitry

(See Schematic 16-2)

The RAM Control Circuitry is a state machine which provides the correct waveforms to drive the RAMs during read, write, and refresh cycles. It includes the following blocks: RAM Timer, RAS Drivers, CAS Driver, Write Enable Driver, Multiplexer Steering, and Refresh Cycle Latch.

RAM Timer

(See Schematic 16-2)

The RAM Timer consists mainly of a shift register made from type D flip-flops. The shift register is clocked by FCLK (Fast Clock), the on-board version of the terminal's 18.432 MHz clock.

When the RAM Cycle Requestor wants a RAM cycle to begin, it sends the CYCLE signal. On the next STATECLK pulse, a low is latched into the Cycle Request Flip-Flop, and the RAM cycle begins. Succeeding FCLK pulses clock the "low" from each stage of the shift register to the next; this produces the timing signals shown in Figure 15-5: T0, T1, T2, etc.

The timing signals T1 to T9 are used to set and reset the flip-flops which generate the waveforms needed to drive the RAMs. (Figure 15-6 shows these waveforms when writing data into RAM bank 0.)

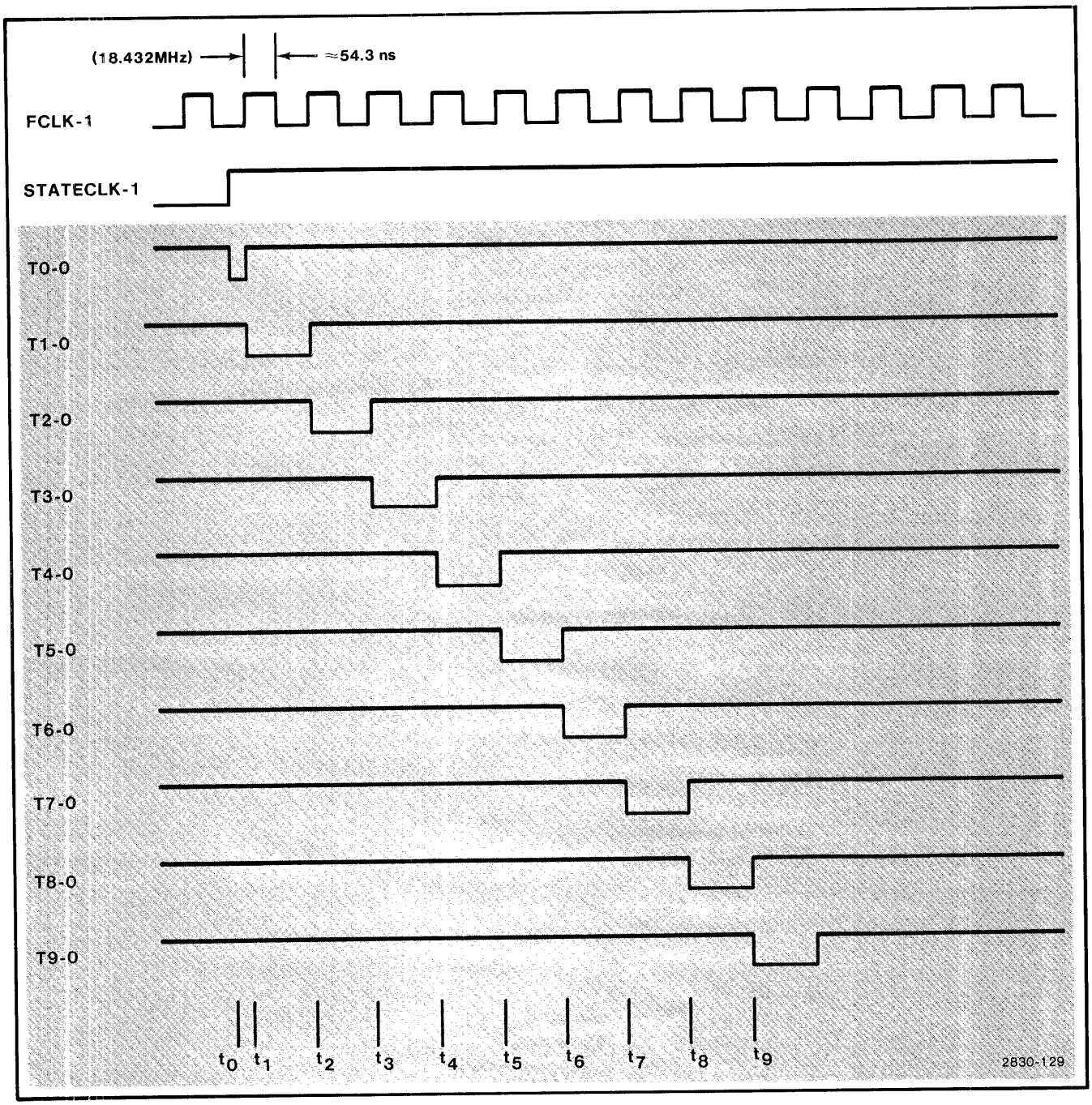


Figure 15-5. RAM Timing Waveforms.

RAS Drivers

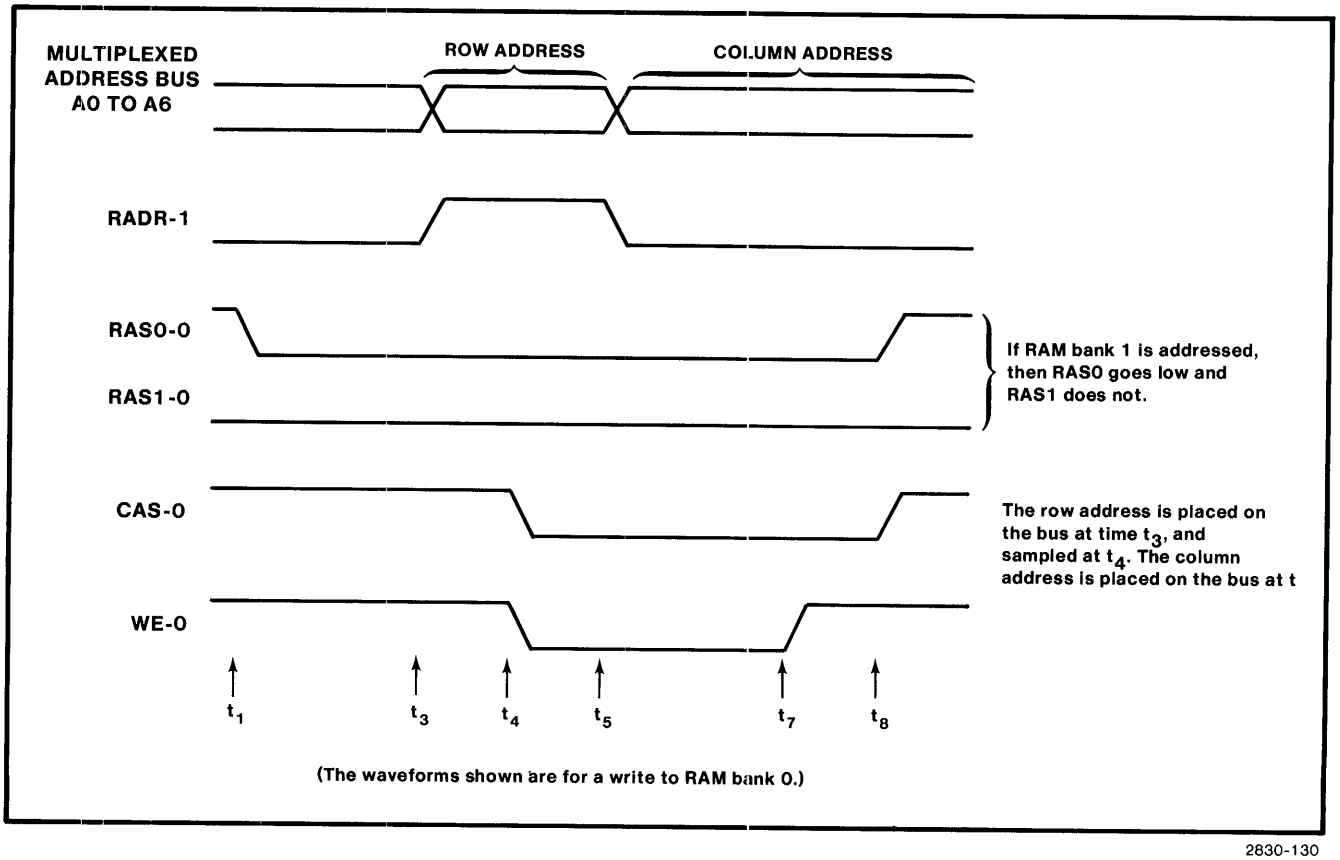
(See Schematic 16-2)

The RAS Drivers are two flip-flops which send the row address strobe signals for the RAMs. One flip-flop sends the RAS0 and RRAS0 signals for RAM bank 0, while the other sends the RAS1 and RRAS1 signals for RAM bank 1.

Suppose the character font being addressed lies in RAM bank 0. Then the BANK0 signal (from the RAM Address Decoder) will be true, and the RAS driver flip-flop for bank 0 will be set at time t_1 and cleared at time t_9. The RAS driver for bank 1, on the other hand, stays cleared. Thus bank 0 of the RAMs gets a Row Address Strobe signal (RRAS0), while bank 1 does not.

The outputs of the RAS drivers are:

- **RAS0 and RAS1.** These signals enable bank 0 and bank 1, respectively, of the Scrunch RAM.
- **RRAS0 and RRAS1.** These signals enable bank 0 and bank 1, respectively, of the Graphics RAMs.



2830-130

Figure 15-6. RAM Control Waveforms During a Write to Bank 0 of Graphics RAM.

CAS Driver

(See Schematic 16-2)

During read or write cycles (but not refresh cycles), the CAS Driver flip-flop is set at time t_4 and cleared at time t_8. This provides the CAS waveform (Figure 15-6) to both banks of Graphics RAMs.

Write Enable Driver

(See Schematic 16-2)

During a write cycle (but not read or refresh cycles), the Write Enable Driver flip-flop is set at time t_4 and cleared at time t_7. This sends the WE (Write Enable) signal to the Graphics RAMs, as shown in Figure 15-6. It also sends the SWE (Scrunch Write Enable) signal to cause a write into the Scrunch RAMs.

Multiplexer Steering

(See Schematic 16-2)

The Multiplexer Steering flip-flop controls the RAM Address Multiplexer (part of the RAM Address circuitry) by sending the RADR (Row Address) signal. At time t_3 the flip-flop is set, making RADR go true; this places the row address bits on the RAM address pins. At time t_5 the flip-flop is cleared; this sends RADR false, placing the column address bits on the RAM address pins.

Refresh Cycle Latch

(See Schematic 16-2)

When the RAM Cycle Requestor initiates a RAM refresh cycle, it sends the REFRESH signal as well as the CYCLE signal. The same STATECLK pulse that starts the RAM Timer also clocks the state of the REFRESH line into the Refresh Cycle Latch. The latch outputs (RCYCLE-0 and RCYCLE-1) inform other circuitry that a refresh cycle is in progress. The latch returns to its initial state at time t_9.

During a refresh cycle, both RAS Drivers operate, so that both banks of Graphics RAMs are refreshed. Also, the Scrunch RAMs (which need not be refreshed) are disabled.

RAM Address Circuitry

The RAM Address Circuitry includes: CSM Address Latch, Pipeline Decode, RAM Address Decoder, and the RAM Address Multiplexer and Refresh Address Counter.

CSM Address Latch

(See Schematic 16-1)

The CSM (Character Set Memory) Address Latch consists of two 74LS374 ICs, which include not only type D latches, but also tri-state buffers on the latch output lines. On each negative transition of COLCLK, the type D latches accept the address being presented on the Character Set Memory address lines, FA0-FA4, CA0-CA6, and RA0-RA3. If an I/O cycle is not being requested, the IOENB signal from the RAM Cycle Requestor will be false; in that case, the tri-state buffers in the 74LS374s place the outputs of the type D latches on the board's address busses FAD0-FA4, CAD0-CAD6, and RAD0-RAD3.

Pipeline Decode

(See Schematic 16-1)

The RAM Cycle Requestor needs to know whether the Display Controller is calling for a dot pattern located on the Graphics Memory Board. Moreover, the RAM Cycle Requestor needs this information before the address is latched into the CSM Address Latch. It is the function of the Pipeline Decode circuit to "look upstream in the video pipeline" and provide this "early warning."

The Pipeline Decode circuit monitors the Character Set Memory address lines, and provides the MYCHAR (My Character) signal when those lines specify a character in one of the fonts stored on the Graphics Memory Board.

A HI/LO jumper and several movable straps (the Pipeline Decode straps) specify which character fonts are installed on the board, so that the Pipeline Decode circuit may respond with the MYCHAR signal whenever it recognizes one of those fonts on the FA0-FA4 font address lines.

Normally, the HI/LO jumper is installed in its HI position, and only fonts 16 to 31 may be installed on the Graphics Memory Board. However, on rare occasions a second Graphics Memory Board may be installed in the terminal. If this is the case, then the second of the two Graphics Memory Boards has its HI/LO jumper installed in the LO position; that Board then holds fonts numbered 15 or less. Figure 15-7 shows the two positions of this HI/LO jumper, as well as the location of the Pipeline Decode straps. Table 15-2 lists the PIPELINE DECODE (and ADDRESS MAPPING) settings for each of the possible installation configurations.

Table 15-2

PIPELINE DECODE AND ADDRESS MAPPING STRAP SETTINGS

First Graphics Memory Board (LO/HI jumpers set to HI)

Memory Option Installed	Fonts Installed	PIPELINE DECODE Straps	ADDRESS MAPPING Straps
		0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7
Option 23 (4K of RAM)	30-31	– – – – – – – X	– – – – – – – 0
Option 24 (8K of RAM)	28-31	– – – – – – X X	– – – – – – 1 0
Option 25 (16K of RAM)	24-31	– – – – X X X X	– – – – 0 0 0 0
Option 26 (32K of RAM)	16-31	X X X X X X X X	1 1 1 1 0 0 0 0

Second Graphics Memory Board (LO/HI jumpers set to LO)

Memory Option Installed	Fonts Installed	PIPELINE DECODE Straps	ADDRESS MAPPING Straps
		0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7
Option 23 (4K of RAM)	14-15	– – – – – – – X	– – – – – – – 0
Option 24 (8K of RAM)	12-15	– – – – – – X X	– – – – – – 1 0
Option 25 (16K of RAM)	8-15	– – – – X X X X	– – – – 0 0 0 0
Option 26 (32K of RAM)	2-15	– X X X X X X X	– 1 1 1 0 0 0 0

KEY: "–" No strap installed.
"X" PIPELINE DECODE strap installed.
"0" ADDRESS MAPPING strap set to select Bank 0.
"1" ADDRESS MAPPING strap set to select Bank 1.

RAM Address Decoder

(See Schematic 16-2)

The RAM Address Decoder performs a function similar to that of the Pipeline Decode circuit: it examines the lines of the address bus, looking for font addresses specifying characters on the Graphics Memory Board. However, the RAM Address Decoder monitors the on-board font address lines FAD1-FAD4, rather than the Character Set Memory address lines FA0-FA4. Also, the RAM Address Decoder detects not only whether the font being addressed lies on the board; it also detects in which of the two banks of RAM that font lies. It is enabled to do this by the ADDRESS MAPPING straps, which select the RAM bank in which each font stored. Table 15-2 shows the normal setting of these straps for each configuration of the board.

Like the Pipeline Decode circuit, the RAM Address Decoder has a HI/LO jumper which is normally installed in its HI position. Should a second Graphics Memory Board ever be installed, the HI/LO strap for that second board would be wired in its LO position.

RAM Address Multiplexer and Refresh Address Counter

(See Schematic 16-2)

The RAM Address Multiplexer and Refresh Address Counter is a type 3242 integrated circuit. This is a special-purpose IC designed for use with the 2104 and 2116 RAMs.

During read or write operations the multiplexer/counter functions as an address multiplexer, presenting either the row address bits or column address bits to the RAM address pins A0-A6. The RADR (Row Address) signal from the Multiplexer Steering circuit controls this. When RADR is true, the 3242 places the row address bits on the A0-A7 lines; when RADR is false, it places the column address bits on those lines.

During a refresh cycle (signaled by the RCYCLE signal from the Refresh Cycle Latch), the "refresh address counter" comes into play. The current count in the refresh address counter is placed on the A0-A6 lines as a row address. All RAM locations sharing that row address are then refreshed. When the refresh cycle ends, the trailing edge of the RCYCLE pulse clocks the internal counter; on the next refresh cycle, the row address sent to the RAMs will be incremented by one.

RAM Cycle Requestor

(See Schematic 16-1)

The RAM Cycle Requestor controls the operation of the Graphics Memory Board during RAM cycles. This circuitry determines whether a CSM cycle, I/O cycle, or refresh cycle is to occur, and sends signals (CYCLE, REFRESH, IOSEL) to initiate these cycles. During a RAM cycle, the RAM Cycle Requestor also performs several miscellaneous logic functions, such as generating the DCDATACLK, IOCLK, and DATAENB signals to control other parts of the Display Memory Board.

During a CSM (Character Set Memory) Cycle, the Graphics Memory Board appears as a collection of ROMs in the Display Controller's Character Set Memory. The Display Controller places a dot pattern address on its FA0-FA4, CA0-CA6, and RA0-RA3 address lines, and then brings the COLCLK-1 line low to latch that address into the CSM Address Latch. The Graphics Memory Board then reads from its RAMs; the next time COLCLK-1 goes low, it places the dot pattern data on the Character Set Memory data lines RD0-RD7, for the Display Controller to read. It also places a GRAPH bit on the GRAPH line, telling the Display Controller whether to display the dots with the 12 Mhz bit clock or the 14 MHz scrunch clock.

During an I/O cycle, the Graphics Memory Board reads from or writes to its RAMs, in response to a request made by the Processor via the I/O registers.

When neither CSM cycles nor I/O cycles are being requested, the RAM Cycle Requestor initiates refresh cycles to keep the RAMs from "forgetting" their data.

To illustrate the operation of the RAM Cycle Requestor (and the Graphics Memory Board as a whole), we'll examine what happens during each of the three types of RAM Cycles.

CSM Cycle

Figure 15-7 shows waveforms during a CSM cycle, when the Display Controller is reading from Graphics Memory. The times t_1, t_2, etc., shown in the figure relate to the following description, and not to the T1, T2, etc., signals within the RAM Control state machine.

At time t_1, COLCLK-1 goes low and the Display Controller advances to the next character in the Row Buffer being scanned. It takes about 500 ns — most of a COLCLK period — for the Row Buffer to respond to its new address.

At time t_2, the Row Buffer outputs have settled. These outputs are the Character Set Memory address lines: FA0-FA4, CA0-CA6, and RA0-RA3. The Pipeline Decode circuitry monitors the FA1-FA4 address lines to see if the Display Controller is calling for a font which is installed on the Graphics Memory Board. If it is, then the Pipeline Decode signal sends a low pulse on the MYCHAR-0 line.

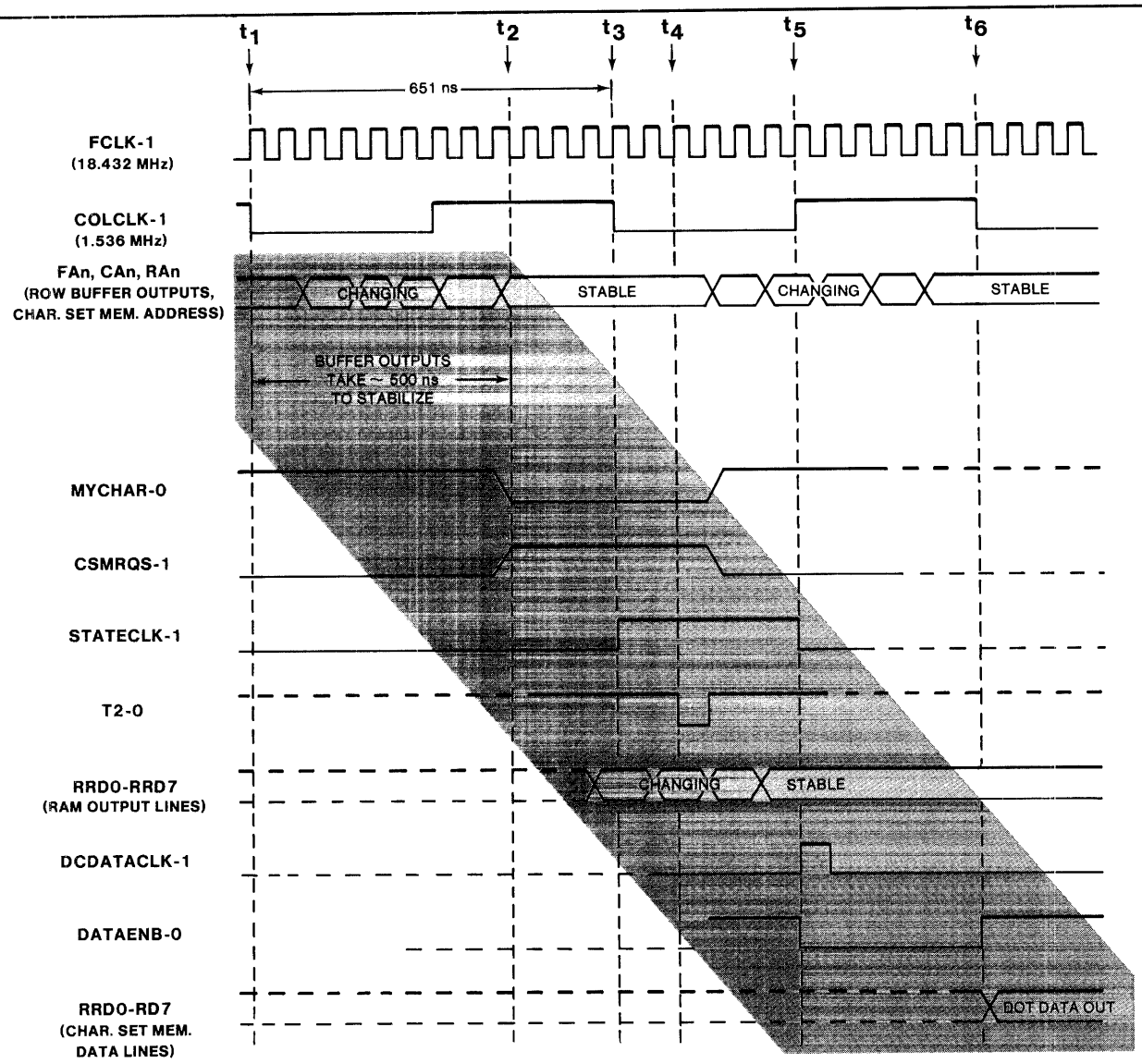
Refer now to Schematic 16-1. Since there is no hard copy in progress (HCIPS is false), gate U425C clears flip-flop U331A, causing it to send the CSMRQS (Character Set Memory Cycle Request) signal. Thus, at time t_2 (or shortly thereafter), the MYCHAR and CSMRQS signals both go true.

At this point, the inputs to the Cycle Request ROM (U421) are as follows: CSMRQS (from U331A) is true; IORQS (from U315A) may be true or false; WINDOW (from the Timing Circuitry) is false; HCIPS is false (there is no hard copy in progress); and SCLOCK (Synchronized Clock) is false. (The latter is true because the SCLOCK flip-flop U431A is being held in its "preset" or "false" state by the zero on the HCIPS line.) ROM U421 is programmed to respond to those inputs by sending the CYCLE signal true and the REFRESH and IOSEL signals false.

Sending CYCLE true, REFRESH false, and IOSEL false is the way the RAM Cycle Requestor requests a CSM cycle. When COLCLK next goes low (time t_3), the Timing Circuitry sends a STATECLK pulse (Figure 15-7) to start the RAM Control state machine running.

Two other things happen as COLCLK-1 goes low at time t_3. First, the COLCLK pulse latches the Character Set Memory address bits into the CSM Address latch. Next, the Timing Circuitry responds to the COLCLK pulse with a STATECLKA pulse. This clocks the IOENB flip-flop (part of the RAM Cycle Requestor, Schematic 16-1), which sends IOENB (I/O Enable) false. The false IOENB signal enables the CSM Address Latch outputs, placing the address on the board's internal address bus: lines FAD0-FAD4, CAD0-CAD6, RAD0-RAD3.

In the COLCLK period between times t_3 and t_6, the RAM Control state machine runs the RAMs through a read cycle.



t_1: As COLCLK-1 goes low, the Display Controller's Row Buffer advances to the next character.

t_2: About 500 ns later, the Row Buffer outputs are stable. The Pipeline Decoder recognizes that a dot pattern on this board is required, and sends the MYCHAR (My Character) signal. MYCHAR causes CSMRQS ("Character Set Message Cycle Request") to go true.

t_3: As COLCLK-1 goes low again, the character set memory address bits are latched onto the Graphics Memory Board's RAM address lines. Also, STATECLK goes high, triggering a new RAM cycle.

t_4: Two FCLK pulses later, the RAM Timer circuit (part of RAM Control) sends T2. This turns off ROMRQS.

t_5: After four more FCLK pulses, DCDATACLK latches the dot data into the RAM Output Buffer.

t_6: As COLCLK-1 goes low again, DATAENB (Data Enable) enables the tri-state RAM Output Buffer. This places the dot data on Character Set Memory data lines RD0-RD7.

2830-132

Figure 15-7. Graphics Memory Timing During a CSM Cycle.

Partway through that cycle (time t_4 in Figure 15-7), the RAM Control circuitry sends its T2 signal, presetting flip-flops U331B and U431B in the RAM Cycle Requestor. This causes the next SUBCLK pulse to reset the CSMRQS flip-flop to its "false" state, ready for the next character to come along the video pipeline.

At t_5, the data on the RAM output lines has stabilized; the RAM Control circuit sends the DATACLK signal. This signal is gated through U311B in the RAM Cycle Requestor (Schematic 16-1) to become the DCDATACLK (Display Controller Data Clock) signal. The DCDATACLK signal clocks the RAM output data into the type D flip-flops in the CSM Output Buffer.

The same DCDATACLK signal clears the DATAENB flip-flop (U321B in the RAM Cycle Requestor, Schematic 16-1). This sends the DATAENB (Data Enable) signal, enabling the CSM Output Buffer and placing the data on the Character Set Memory data lines RD0-RD7.

By the time COLCLK next goes low (t_6), the Display Controller will have loaded the dot pattern data into its Shift Register. As COLCLK goes low, it presets the DATAENB flip-flop, turning off the DATAENB signal. This disables the CSM Output Buffer.

I/O Cycle

The Processor requests an I/O cycle by writing into either the Write Trigger Word or the Read Trigger Word. This clears the Ready Flip-Flop in the Processor Interface (Schematic 16-1). This causes the IORQS Flip-Flop (U315A) to be set on the next STATECLK pulse, sending the IORQS (I/O Cycle Request) signal. However, the I/O cycle request will not be honored until the next horizontal retrace.

When the horizontal retrace occurs, the Timing Circuitry sends WINDOW true. This tells the RAM Cycle Requestor that the time window for I/O cycles has begun. The inputs to the Cycle Request ROM (U421, Schematic 16-1) are as follows: IORQS is true; WINDOW is true; HCIPS is false (there is no hard copy in progress); and SCLOCK is false.

For those inputs, the Cycle Request ROM is programmed to give the following outputs: it sends the CYCLE and IOSEL (I/O Select) signals true, and sends the REFRESH signal false.

With CYCLE true, the next STATECLK pulse tells the RAM Control state machine to start a RAM cycle.

With IOSEL true, the next STATECLKA pulse sets the IOENB flip-flop, sending the IOENB (I/O Enable) signal. This disables the tri-state buffers in the CSM Address Latch and enables those in the I/O registers: the Character Address, Font Address, and Row Address Words. Thus the I/O registers drive the board's internal address lines FAD0-FAD4, CAD0-CAD6, and RAD0-RAD3.

The RAM Control circuitry proceeds to cycle the RAMs through a read or write operation. If it is a read operation, the RAM Control circuit sends the DATACLK signal when the data has stabilized on the RAM output lines RRD0-RRD7 and on the RSCRUNCH line. Gate U211B in the RAM Cycle Requestor takes the DATACLK signal and passes it on as IOCLK (I/O Clock). IOCLK latches the RAM output data into the "read" half of the Dot Data Word and into the Scrunch Latch (which relays the scrunch bit to the "read" half of the Status Word).

Refresh Cycle

When the Display Controller is not trying to access the Graphics RAM for CSM cycles, and when there is no pending I/O cycle request, the RAM Cycle Controller initiates refresh cycles to keep the dynamic RAMs from "forgetting" their data.

Note that it is the Cycle Request ROM (U421, Schematic 16-1) which "decides" (by examining its input lines) whether a CSM cycle or an I/O cycle is required. For most combinations of this ROM's input signals, neither type of cycle is needed; for these input signal combinations, the ROM is programmed to request refresh cycles. This is done by sending the CYCLE and REFRESH signals true, and the IOSEL signal false.

With CYCLE and REFRESH both true, the next STATECLK pulse causes the RAM Control circuitry to give the RAMs a refresh cycle.

Timing Circuitry

The Timing Circuitry (Schematic 16-5) provides several signals for use elsewhere on the Graphics Memory Board. It consists of the following blocks: Clock Generator, HCIPS Generator, and WINDOW Generator.

Clock Generator

The Clock Generator provides three signals: SUBCLK, STATECLK, and STATECLKA.

SUBCLK (Substitute Clock). A number of the Graphics Memory Board timing functions are clocked by COLCLK, the column clock from the Display Controller. COLCLK, though, is not available during vertical retraces. The SUBCLK signal serves as a substitute for COLCLK during those times when COLCLK is unavailable.

Refer to Schematic 16-5. U225 (a 74LS93 binary counter) counts down the FCLK 18.432 MHz clock, providing one SUBCLK pulse for every 16 FCLK pulses.

The SUBLCK pulses may be turned off by clearing U225 with a high at its reset inputs, Pins 2 and 3. This may occur in two ways:

- If the COLCLK column clock is running normally, each COLCLK pulse clears U225. By repeatedly clearing the counter, the COLCLK pulses prevent it from reaching the count of eight and sending SUBCLK pulses. Thus the substitute clock will not run when the normal column clock is available.

- Flip-flop U221A, when set, holds counter U225 cleared, thus turning off the SUBCLK pulses. The flip-flop is set, turning off SUBLCK, follows. In each vertical retrace interval, U15 counts the horizontal retraces (BGATE pulses) which occur after the VDRIVE pulse that begins the vertical retrace. After 11 BGATE pulses have occured, U15's MAX/MIN output goes high. On the next BGATE pulse (if no hard copy is in progress) gates U115D and U325A set the flip-flop, turning on SUBCLK pulses for the remainder of the vertical retrace. The flip-flop will be cleared, however, on the first COLCLK pulse following the vertical retrace interval.

The STATECLK and STATECLKA signals are logically identical, but are sent by different gates. STATECLK pulses occur whenever there are either COLCLK pulses or SUBCLK pulses.

HCIPS Generator

The HCIPS Generator provides the HCIPS (Synchronized "Hard Copy in Progress" signal) for use by other circuits on the board. This circuit consists of a J-K flip-flop connected to function as a type D flip-flop. This flip-flop takes the HCIP (Hard Copy in Progress) signal from the Display Controller and synchronizes it with the SUBCLK substitute clock.

WINDOW Generator

The WINDOW Generator provides the WINDOW signal, which is true for a certain "time window" during horizontal retraces. (It is during this time window that the I/O cycles are performed.)

The WINDOW flip-flop (U411B) is cleared during horizontal scans (when HDRIVE is false). When a horizontal retrace begins, though, HDRIVE goes true and counter U415 starts to count SUBCLK pulses. Its MAX/MIN output is low until the counter reaches the count of 15. With MAX/MIN and HDRIVE-0 both low, gate U425B ceases to clear the WINDOW flip-flop; the first SUBCLK pulse sets that flip-flop and WINDOW goes true.

On the ninth SUBCLK pulse, counter U415's MAX/MIN output goes high, and the WINDOW flip-flop is cleared. Thus the WINDOW signal is only true for eight SUBCLK periods towards the start of a horizontal retrace.

Processor Interface Circuitry

The Processor Interface Circuitry (Figure 15-2) consists of the following circuit blocks:

- **Bus Address Decoder.** The Bus Address Decoder monitors the Processor's address bus, IOADR, READ, and WRITE lines. When the Processor reads or writes to an I/O register on the Graphics Memory Board, this circuit sends the appropriate clock or enabling signal to that register. (For instance, a write to the Font Address Word causes the Bus Address Decoder to send the FONTWR signal, clocking data bits into the Font Address Word I/O register.)

- **I/O Registers.** The "write" I/O registers are the Font Address Word, Character Address Word, Row Address Word, and the "write" halves of the Dot Data Word and the Status Word. The "read" I/O registers are the "read" halves of the Dot Data Word and the Status Word.

- **Bus Receiver.** The Bus Receiver buffers the Processor data bus (lines BD0-BD7). The buffered data appears on the RB0-RB7 lines, which go to the "write" I/O registers.

- **Ready Flip-Flop.** The READY flip-flop controls the READY line, which tells the Processor when it may request an I/O cycle. When the Processor begins an I/O cycle (by writing to the Write Trigger Word or Read Trigger Word), READY goes false. When the I/O cycle is completed, READY goes true again.

- **R/W Flip-Flop.** The R/W flip-flop "remembers" whether the Processor last asked for a "read" or a "write" operation in Graphics Memory. It is set one way or the other when the Processor writes into the Write Trigger Word or the Read Trigger Word.

- **Interrupt Requestor and Interrupt Address Decoder.** When an I/O cycle is complete, the Interrupt Requestor sends the IRQST (Interrupt Request) signal. The Interrupt Address Decoder monitors the interrupt address lines IA0-IA2 and passes the interrupt request on to the Processor (sends the IRQ signal) when the board's interrupt address appears on those line.

Section 16

OPTION 31 — CHARACTER SET EXPANSION (4025 ONLY)

INTRODUCTION

Option 31 for the 4025 is the Character Set Expansion Board. This board holds ROMs for alternate character fonts, such as the rulings ROM (Option 32), or the math character font ROM (Option 34).

The dot patterns for all character fonts are stored in the Display Controller's Character Set Memory. Thus, the Character Set Expansion Board is part of that memory. Like the rest of Character Set Memory, it resides in the Display Controller's address space and is not addressable by the Processor. It communicates with the Display Controller by means of a ribbon cable.

Inputs and Outputs

The inputs to the Character Set Expansion Board are the same input signals used by other parts of the Character Set Memory. These are: COLCLK, FA0-FA4, CA0-CA6, and RA0-RA3.

The outputs are the eight data bits RD0-RD7. These tell which dots are to be turned on in one row of one character's dot matrix.

Another output is GRAPH. This signal tells whether the character being addressed is a graphic character or an alphanumeric character. (Alphanumeric characters are "squished" (compressed) by the Display Controller's 14 MHz scrunch clock, giving a little extra space between adjacent characters. Graphic characters, on the other hand, are not squished, so that adjacent characters are not separated.) GRAPH is true for graphic characters (e.g., Option 32 rulings characters); it is false for alphanumeric characters (e.g., Option 34 math characters).

When each ROM is installed, a SQSH strap for that ROM is set. If it is set to SQSH Y, characters from that ROM are "squished." (When the ROM is addressed, the GRAPH signal will be false.) If the strap is set to SQSH N, characters in the ROM will not be squished. (GRAPH is sent true when the ROM is addressed.)

Modes of Operation

The board can operate in two modes, selected by a strap:

- **Mode 1.** Each character font occupies a single 64-character ROM. Seven-bit characters whose least-significant six bits are the same are displayed identically. The board holds up to six ROMs, so up to six 64-character fonts may be installed on it.

- **Mode 2.** Each character font on the board occupies two 64-character ROMs. Thus, a font can have 128 distinct characters. The board holds up to three 128-character fonts.

Which six fonts are installed on the board in Mode 1, or which three fonts are installed in Mode 2, is determined by straps which set the *base font* for that board. The base font may be Font 0, Font 8, Font 16, or Font 24. Table 16-1 lists the fonts which may be installed in Mode 1 and Mode 2 for different base fonts. Usually the board is strapped to select Mode 1, with Font 0 as the base font; in that case, ROMs may be installed for Fonts 0, 1, 2, 3, 4, and 5. (Font 0, however, is usually reserved for the standard character font ROM installed on the Display Controller Board.)

NOTE

Font 0 is reserved for the the standard character font. Do not install any ROMs in the socket(s) for Font 0 unless you have also disabled the standard character font ROM on the Display Controller Board.

Table 16-1

CHARACTER FONTS ON THE CHARACTER SET EXPANSION BOARD

Base Font	Mode 1		Mode 2	
0	[a]Font 0:	ROM 0	[a] Font 0:	ROMs 0 and 1
	Font 1:	ROM 1	Font 1:	ROMs 2 and 3
	Font 2:	ROM 2	Font 2:	ROMs 4 and 5
	Font 3:	ROM 3		
	Font 4:	ROM 4		
	Font 5:	ROM 5		
8	Font 8:	ROM 0	Font 8:	ROMs 0 and 1
	Font 9:	ROM 1	Font 9:	ROMs 2 and 3
	Font 10:	ROM 2	Font 10:	ROMs 4 and 5
	Font 11:	ROM 3		
	Font 12:	ROM 4		
	Font 13:	ROM 5		
6	Font 16:	ROM 0	Font 16:	ROMs 0 and 1
	Font 17:	ROM 1	Font 17:	ROMs 2 and 3
	Font 18:	ROM 2	Font 18:	ROMs 4 and 5
	Font 19:	ROM 3		
	Font 20:	ROM 4		
	Font 21:	ROM 5		
4	Font 24:	ROM 0	Font 24:	ROMs 0 and 1
	Font 25:	ROM 1	Font 25:	ROMs 2 and 3
	Font 26:	ROM 2	Font 26:	ROMs 4 and 5
	Font 27:	ROM 3		
	Font 28:	ROM 4		
	Font 29:	ROM 5		

[a]Do not install Font 0 on the Character Set Expansion Board unless the Standard Font circuitry has been disabled on the Display Controller Board.

CIRCUIT DESCRIPTIONS

Block Diagram

Figure 16-1 is a block diagram for the Character Set Expansion Board. (See also Schematics 17-1 and 17-2.)

The FA0 to FA4 (Font Address) lines from the Display Controller go to the Base Font Selector and Mode Selector blocks. These blocks, in turn, drive the ROMA0 to ROMA3 (ROM Address) lines; those lines determine which, if any, of the board's ROMs is being addressed.

The Input Latches store the ROM Address and Character Address data from one COLCLK (Column Clock) pulse to the next. The latched ROM Address lines are decoded by the ROM Selector to drive the ROM enable lines, EN0-EN5. The same latched ROM Address lines are fed to the Squish Enabler, which sends the GRAPH signal if the ROM addressed holds graphic information. ("GRAPH true" means that the ROM's character are *not* to be squished.)

Whichever ROM is enabled (if any) responds to the latched character address lines A0-A5 and the Row Address lines RA0-RA3, to access a particular eight-bit byte of character set memory. This byte, which represents one row of dots in one character's dot matrix, is placed on the output "read data" lines RD0-RD7.

Base Font Selector

The Base Font Selector consists of a NAND gate, together with inverters which may be strapped into its input lines. This gate drives the ROMA3 (ROM Address Bit 3) line to help select a particular ROM on the Character Set Expansion Board. The NAND gate output (ROM Address line ROMA3) is *low* whenever a font installed on the board is accessed.

Straps marked "CS4" and "CS3" select the board's base font. Table 16-2 lists the strap settings for the four possible base fonts. (The labels "CS4" and "CS3" come from earlier names for the font address signal lines FA4 and FA3.)

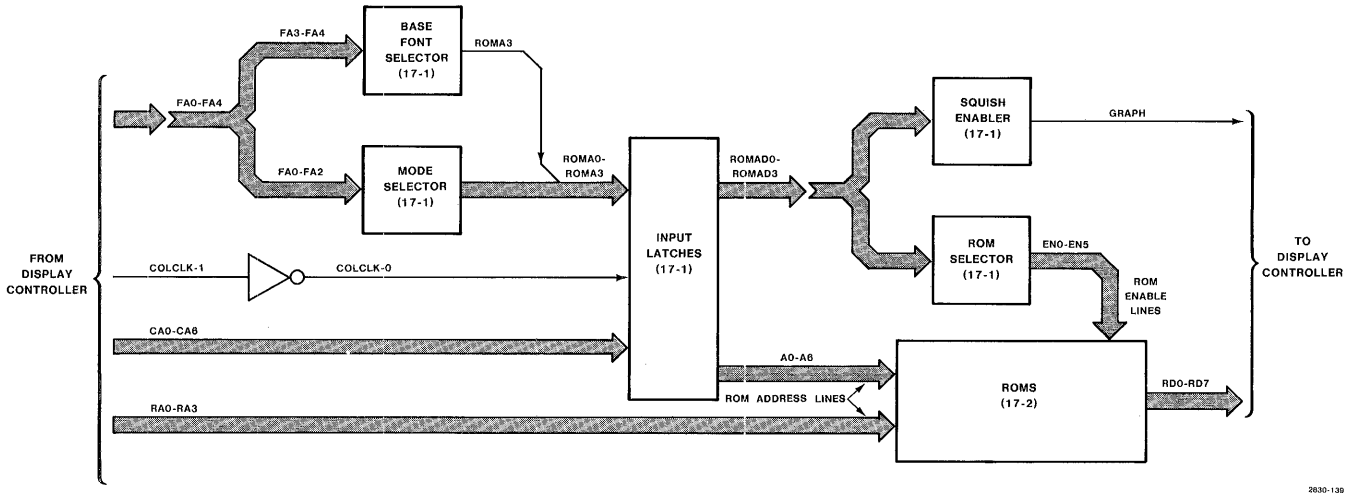


Figure 16-1. Character Set Expansion Board Block Diagram.

Table 16-2

SELECTING BASE FONT ON THE CHARACTER SET EXPANSION BOARD

Base Font	Strap Positions
0	CS4: 0 CS3: 0
8	CS4: 0 CS3: 1
16	CS4: 1 CS3: 0
24	CS4: 1 CS3: 1

Mode Selector

The Mode Selector is a 74LS157 data selector, whose "select" input is controlled by the Mode 1/Mode 2 strap setting.

The board is usually strapped to operate in Mode 1. In this mode the FA0 to FA2 Font Address lines drive the ROMA0 to ROMA2 ROM Address lines, respectively.

In Mode 1, the CA6 Character Address bit is ignored. Consequently, each font's characters 64 through 127 will access the same words in ROM as characters 0 through 63. Thus, each font installed on the board has only 64 distinct characters.

In Mode 2, the CA6 line drives ROM Address line ROMA0; Font Address lines FA0 and FA1 drive ROMA1 and ROMA2. The result is that each character font is split between two 64-character ROMs; the lower-numbered ROM holds characters 0 through 63 (for which CA6 is low), and the higher-addressed ROM holds characters 64 through 127.

In Mode 2, the FA2 Font Address line is ignored. Consequently, each font on the board can be accessed by two font numbers: one for which FA2 is low, and one for which FA2 is high. For instance, suppose the base font is font 8, and ROMs 2 and 3 hold a 128-character special font. According to Table 16-1, this will be font 9; however, these characters can also be accessed as font 13, because fonts 9 and 13 differ only in the FA2 bit in their binary font addresses.

Input Latches

The Input Latches are 74LS174 integrated circuits, which hold type D flip-flops. Each COLCLK pulse from the Display Controller latches the ROM Address and Character Address bits into the 74LS174s, which hold these signals stable until the next COLCLK pulse.

ROM Selector

The ROM Selector is a 74LS138 data selector which decodes the ROM Address lines ROMA0 to ROMA3. The 74LS138 outputs drive the six enable lines EN0 to EN5.

Squish Enabler

The Squish Enabler is a 74156 open-collector data selector driven by the same ROM Address lines as the ROM Selector. The SQSH0 to SQSH5 straps connect its outputs to the GRAPH output line. If one of the ROMs on the board is selected, *and* its SQSH strap is set to "N" (for "no squish"), the GRAPH line will be pulled low; this informs the Display Controller that the character being read is a graphic character, and should be displayed using the 12 MHz bit clock. If the SQSH strap is set to "Y" (for "squish yes"), a pullup resistor on the Display Controller Board pulls the GRAPH line high. With GRAPH high (false), the character will be displayed using the 14 MHz "scrunch clock."

ROMs

The six ROM sockets appear in Schematic 17-2. Each ROM is selected by an enable line, which not only enables the ROM, but also switches on the + 5 volt and + 12 volt power lines to that ROM. When the ROM is not being accessed, it is not supplied with power; this reduces power consumption (and heat generation) on the ROM Expansion Board.

Section 17

OPTIONS 32 AND 34 — RULINGS AND MATH CHARACTERS

OPTION 32: RULINGS CHARACTERS (4024 AND 4025)

NOTE

Refer to Volume 2 of this Service Manual for Option 32 installation procedures.

Option 32 consists of a ROM holding rulings characters (character font number 1). The same ROM is used in both the 4024 and the 4025; however, the placement of the ROM, and the rulings characters available from it, differ between the two terminals.

- In the 4025, the Rulings ROM is installed on the Character Set Expansion Board (Option 31). All 64 rulings characters in the ROM are available for display on the screen.

- In the 4024, the Rulings ROM is installed on the Display Controller Board; no Character Set Expansion Board is needed. However, only 32 of the rulings characters are available for display on the screen.

Figures 17-1 and 17-2 show the rulings characters. Characters 0 through 31 (Figure 17-1) are displayed by both the 4024 and the 4025. However, characters 32 through 63 (Figure 17-2) cannot be displayed by the 4024; these characters are available only in the 4025.

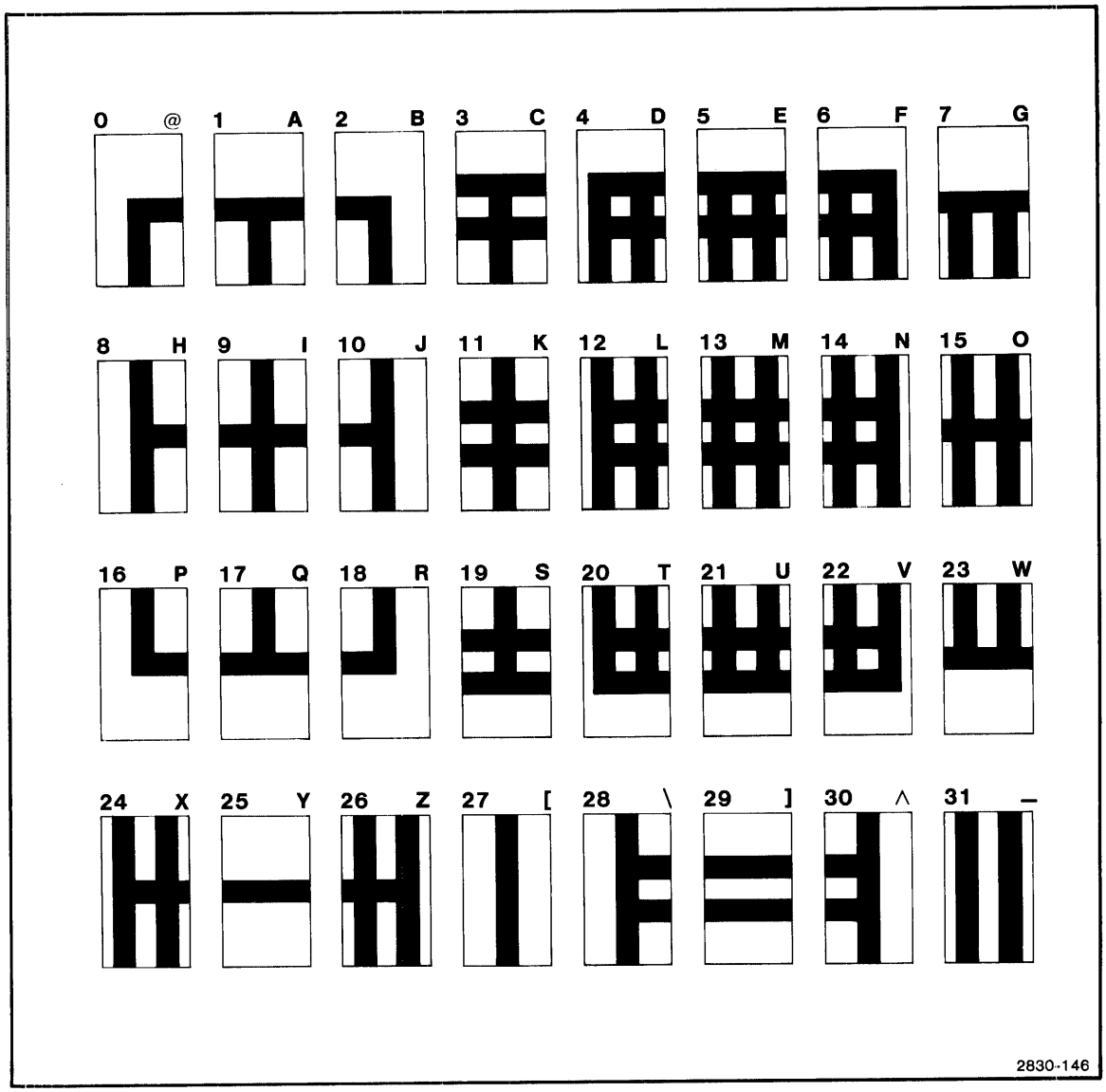


Figure 17-1. 4024/4025 Rulings Characters.

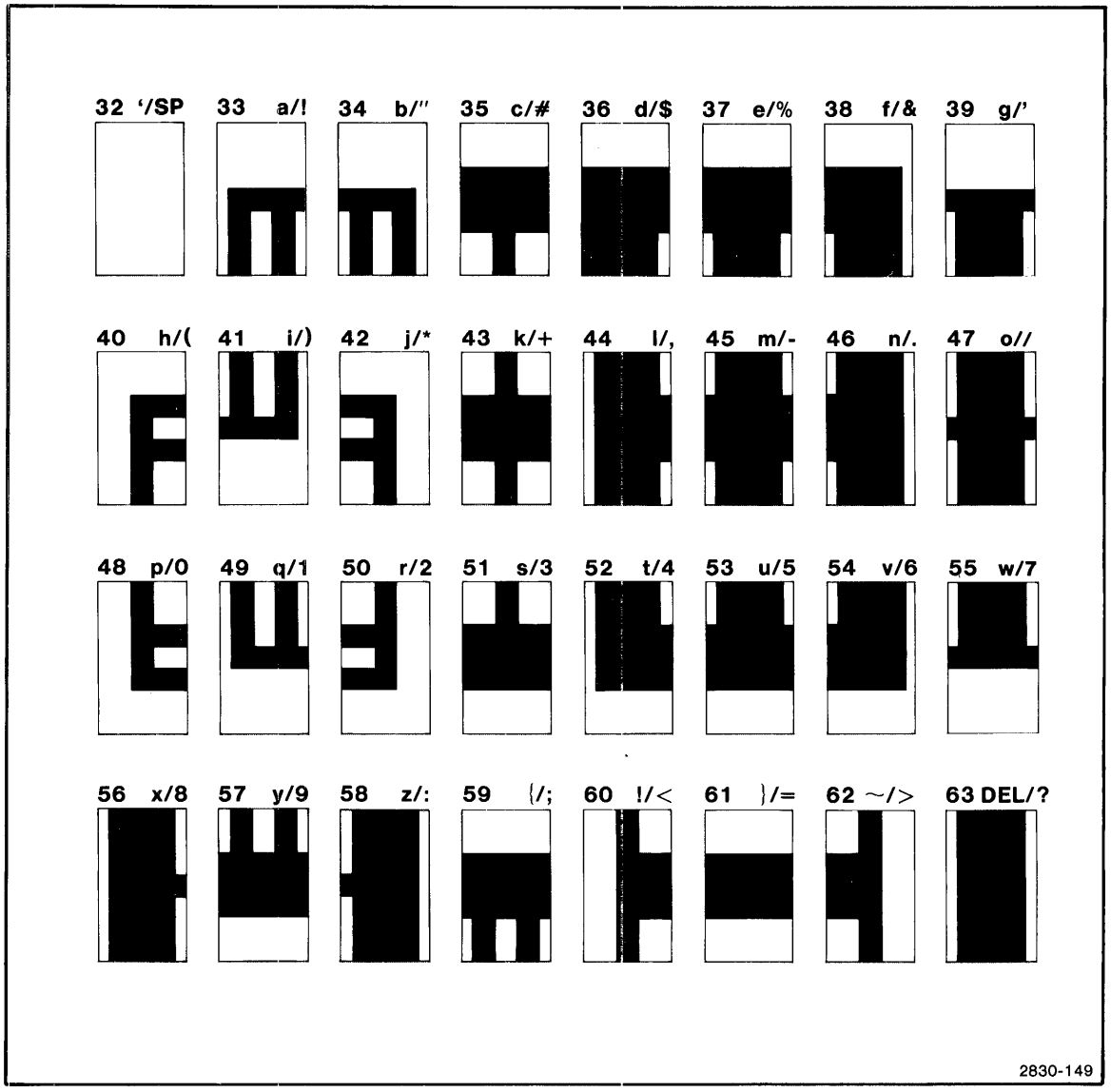


Figure 17-2. 4025 (Only) Rulings Characters.

OPTION 34: MATH CHARACTERS (4025 ONLY)

NOTE

For Option 34 installation procedures, see Volume 2 of this Service Manual.

Option 34 consists of a ROM holding the math character font. Figure 17-3 shows the characters in this font.

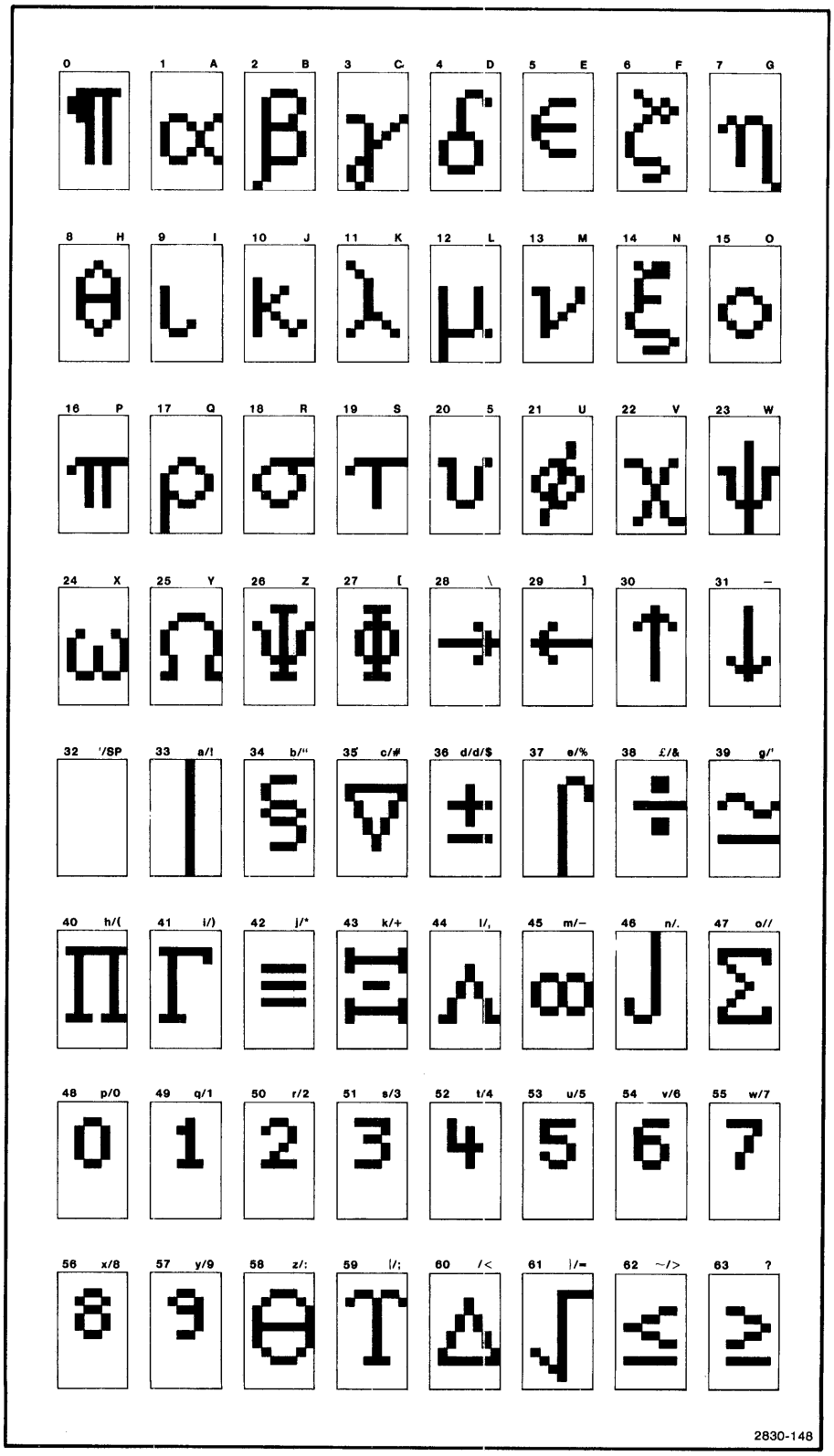


Figure 17-3. Math Characters.

Section 18

OPTION 35 — ROM EXPANSION (4025 ONLY)

INTRODUCTION

Purpose of Option 35

Option 35, ROM Expansion, consists of a circuit board known as the ROM Expansion Board. (On some early terminals, a different board, marked "ROM Option Board," was supplied as Option 35.) The ROM Expansion (or ROM Option) Board has sockets into which optional firmware ROMs may be inserted.

This circuit board also holds a Fix-It for making minor changes to the optional firmware. This Fix-It operates much like the one on the Processor Board.

Two Different Circuit Boards

There are two kinds of Option 35 circuit boards:

- The older type, part number 670-5064-00, called the "ROM Option Board." his board is found in only a few early terminals; it is superseded by the newer type of Option 35 circuit board.

  NOTE

  The older type of board ("ROM Option Board") may only be used with system firmware versions 1.1 or 1.2.

- The newer type, part number 670-5729-XX, called the "ROM Expansion Board." This board holds up to ten ROMs. Eight of the ROMs share addresses X'5000'-X'5FFF'; a *Page Register* selects which of them is actually selected. The other two ROMs occupy addresses X'6000'-X'6FFF' and X'7000'-X'7FFF'.

  NOTE

  The newer type of board ("ROM Expansion Board") may only be used with system firmware versions 1.3 and above.

Since most 4025s have the newer board, the ROM Expansion Board, this section of the manual is devoted to that board. There is, however, a brief description of the ROM Option Board at the end of the section.

ROM EXPANSION BOARD (670-5729-XX)

The ROM Expansion Board appears to the Processor to consist only of memory:

- The optional firmware ROMs at addresses X'5000' to X'7FFF'.
- 768 words of read-only memory at addresses X'0500' to X'07FF'. (This is the part of the Fix-It FPROM that holds patch programs.)
- Write-only memory at addresses X'08F0' to X'08FF'. This is the Page Register, into which the Processor writes to specify which of the "paged ROMs" it will access when reading from addresses X'5000' to X'5FFF'.

Usually, only one ROM Expansion Board is installed in a terminal. However, a second board can be installed. Should that be done, the first of the two ROM Expansion Boards would hold the ROMs at addresses X'6000'-X'6FFF' and X'7000'-X'7FFF', and also eight ROMs sharing addresses X'5000'-X'5FFF'. The second ROM Expansion Board would hold eight more ROMs, also sharing addresses X'50000'-X'5FFF'. The number (from 0 to 15) in the Page Register would determine which of the sixteen possible "paged" ROMs would occupy the shared addresses X'5000'-X'5FFF'. Pages 0 through 7 would be ROMs on the first ROM Expansion Board, and pages 8 through 15 would be ROMs on the second board.

Inputs and Outputs

Inputs to the ROM Expansion Board include: BA0-BA15, IOADR, BD0-BD7, READ, and WRITE. Other inputs are the terminal's RESET line, which clears the Page Register on power-up, and RDIS, by which the Processor Board Fix-It can disable the ROMs on the ROM Option Board.

The ROM Expansion Board's outputs are:

- **BD0-BD7**, the terminal's data bus. Data read from memory is placed on these lines.
- **RDIS (ROM Disable).** When the ROM Expansion Board Fix-It detects a flagged address, it sends the RDIS signal to disable all ROMs, including those on the Processor Board (but not including the Fix-It's own FPROM).

Circuit Descriptions

The ROM Expansion Board (Figure 18-1) includes the following circuit blocks: Address Buffers, Data Buffers, Address Decoder, Page Decoder, I/O Decoder, Page Register, ROMs, and Fix-It.

Address Buffers

(See Schematic 18A-1.)

The address buffers are sixteen 74LS32 logic gates. These gates buffer the BA0-BA15 signals from the terminal's address bus, driving the ROM Option Board's own internal address bus, lines A0-A15.

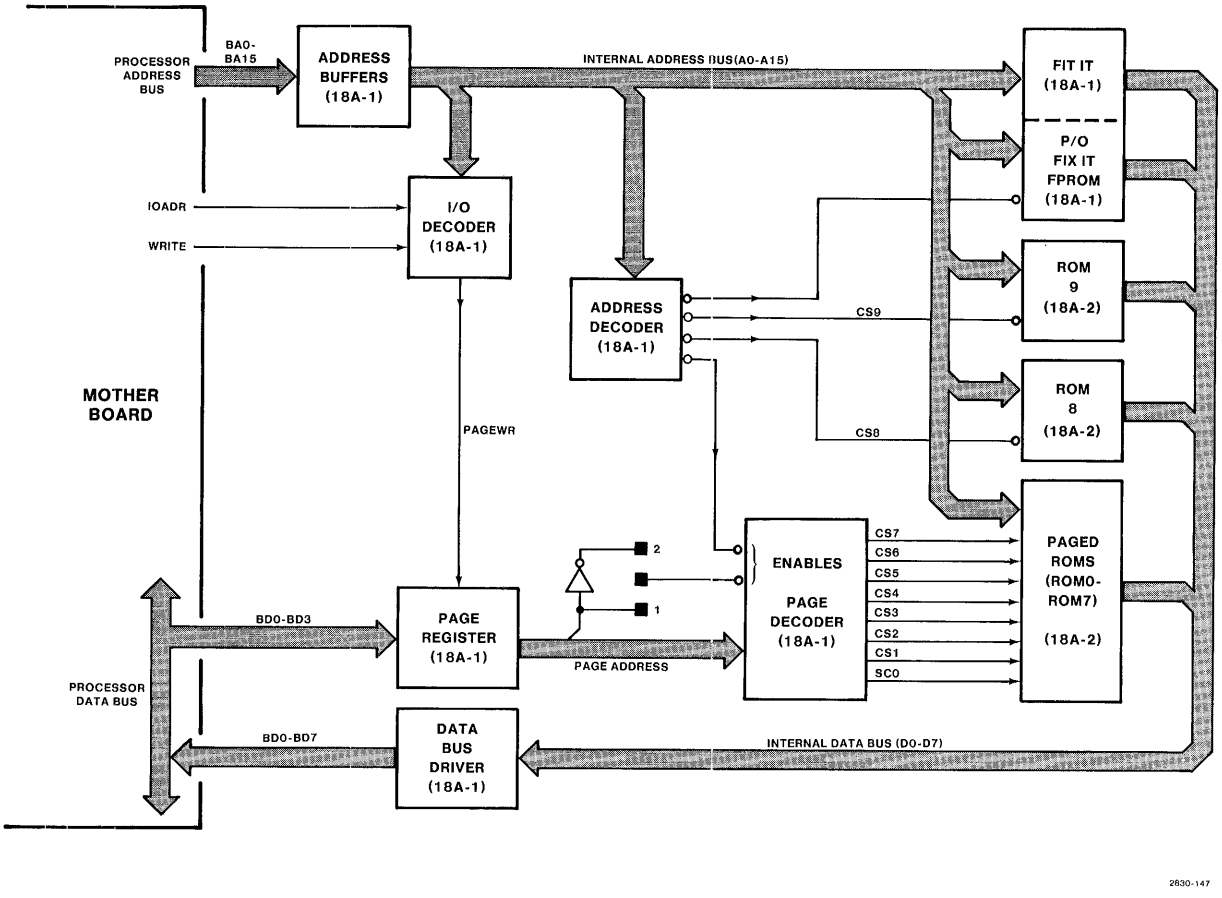


Figure 18-1. ROM Expansion Board Block Diagram.

Data Buffer

The Data Buffer and its enabling gates appear in Schematic 18A-1. Data from the ROMs appear on the board's internal data bus, lines D0-D7. The 74LS244 tri-state buffers send this data on to the terminal's data bus (lines D0-D7) during read operations. Several logic gates enable the Data Buffer only when all the following conditions are met:

- A read operation is in progress. (READ signal is true.)
- One of the ROM Option Board's ROMs is being addressed. (ROM0, ROM1, . . . , or ROM9 signal is true, or the Fix-It FPROM is enabled.)
- The address being accessed is in the lower half of memory (BA15 false).
- ROMs are not being disabled by the Processor Board's Fix-It. (RDIS is false *or* the ROM Option Board's Fix-It has detected a flagged address.)

Address and Page Decoders

(See Schematic 18A-1.)

The Address Decoder detects addresses corresponding to ROM 8 (X'6000'-X'6FFF'), ROM9 (X'7000'-X'7FFF'), or the Fix-It FPROM (X'0500'-X'07FF'). On detecting such an address, it sends a chip select signal to enable the corresponding ROM.

When it detects an address in the range X'5000'-X'5FFF', the Address Decoder sends an enable signal to the Page Decoder. Another enable input for the Page Decoder comes from the Page Register; this signal permits the Page Decoder to be enabled for page numbers from 0 to 7.[1] Provided both enable inputs are true, the Page Decoder reads the least significant three bits in the Page Register and sends a chip select signal to one of the eight paged ROMs, ROMs 0 through 7.

---

[1]**This assumes that the ROM Expansion Board has its "Board Select" strap set to position 1. If two ROM Expansion Boards are installed, the second board would have its Board Select Strap set to "2"; its Page Decoder would be enabled only for page numbers 8 through 15.**

I/O Decoder

(See Schematic 18A-1.)

The I/O Decoder consists of AND gates cascaded to provide a low output only when the following conditions are met:

- The Processor is writing to memory (WRITE signal true).
- The address to which it is writing is in the range X'08F0' to X'08FF'. (The IOADR signal is true, address lines A9 and A8 low, and lines A7, A6, A5, and A4 are all high.)

Those conditions indicate that the Processor is writing to the Page Register. When the WRITE signal goes false at the end of the write operation, I/O Decoder output clocks the Page Register flip-flops, latching the data into the Page Register.

Page Register

(See Schematic 18A-1.)

The Page Register is a 74LS175: four type D flip-flops with a common clock line and a common clear line. The output of the I/O Decoder latches data from BD0-BD3 into the flip-flops whenever the Processor writes to address X'08F0' (or any address in the range X'08F0'-X'08FF'). On power-up or MASTER RESET, the Page Register is cleared by the RESET signal from the Mother Board.

ROMs

The ten ROMs are shown in Schematic 18A-2. Each ROM needs two signals to enable it: a chip select signal for that particular ROM (CS0, CS1, etc.), and the CSALL (Chip Select All) signal. (CSALL is true only if the Fix-It has not detected a flagged address.) Each ROM holds 4096 eight-bit words.

Fix-It

(See Schematic 18A-1.)

The Fix-It works like the Processor Board Fix-It, but is more elaborate because of the paging scheme for ROMs on the ROM Option Board.

The firmware on the ROM Option Board is considered to consist of four-word blocks of memory. Those four-word blocks which are to be changed by the Fix-It have their addresses "flagged" (programmed into the Fix-It's Field Programmable Logic Array.)

When the FPLA detects a flagged address, its F0 output goes low. Provided the FDIS (Fix-It Disable) pin is not grounded, the low on the FPLA's F0 output does three things:

- Steers two 74LS157 data selectors, so that the 82S181 FPROM (Field Programmable Read Only Memory) takes address inputs from the FPLA Pins F2-F7 rather than from the address bus.
- Enables the 82S181 FPROM.
- Disables ROMs elsewhere in the terminal (sends the RDIS signal), and disables the ROMs on the ROM Expansion Board (turns off the CSALL signal).

The FPLA outputs F2-F7, together with address lines A0 and A1 serve as the address inputs to the FPROM. The FPROM places on data lines D0-D7 the word programmed into it at the specified address.

The result is this: At addresses programmed into the Fix-It FPLA, words programmed into the Fix-It FPROM replace the words at those addresses in other ROMs. Thus, the Fix-It can be used to alter instructions in the terminal's firmware.

Strappable Options and Test Points

The ROM Expansion Board includes a number of connectors with shorting straps for configuring the board according to which ROMs which are installed. There are also test points which may be grounded to disable the Fix-It or to force all ROMs into tri-state condition. The connectors for shorting straps and test points are shown in Figure 18-2; their functions are as follows:

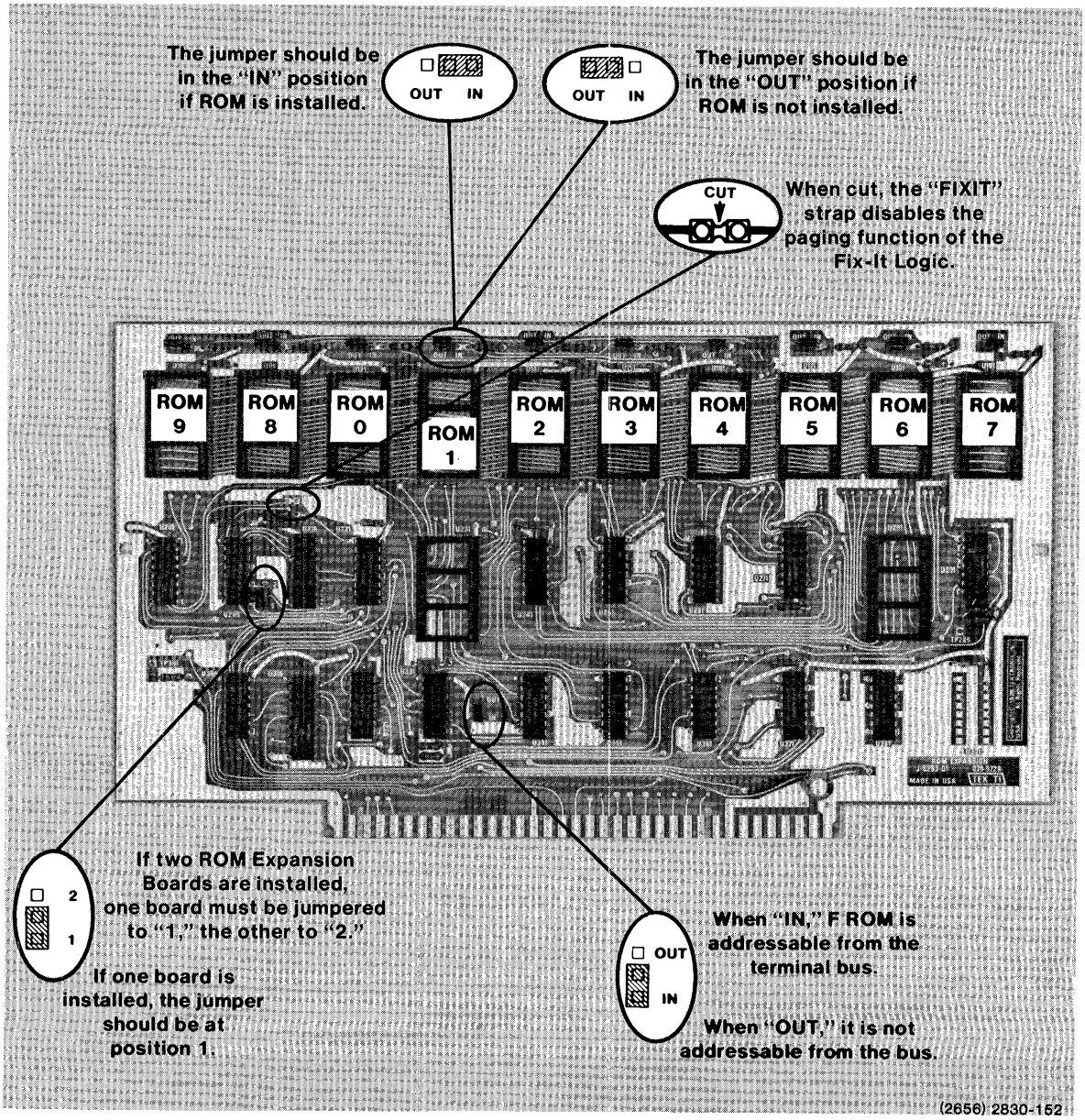


Figure 18-2. ROM Expansion Board Strap Settings.

- **OUT-IN.** For each of the ten ROM sockets there is an OUT/IN strap. This strap is set to IN if a ROM is installed in the socket, otherwise to OUT. These straps are in series with the ROMs' chip select signal lines. When a ROM is strapped IN, its chip select signal from the Address Decoder or Page Decoder is passed on to the ROM; when the ROM is strapped OUT, its chip select input is held high by a pull-up resistor, disabling the ROM.

  There is also an OUT/IN strap for the Fix-It. If two ROM Expansion Boards are installed, the Fix-It on the first board should be strapped IN, and that on the second board should be strapped OUT.

- **Board Select.** If only one ROM Expansion Board is installed, its Board Select strap is set to position 1. If two such boards are installed, the first board's Board Select strap is set to position 1; the second board's strap is set to position 2. In position 1, the board's ROMs 0 through 7 are selected when the number in the page register is from 0 to 7; in position 2, these ROMs are selected when the number is page register is from 8 to 15. That is, the first ROM Expansion Board holds pages 0 to 7, and the second board holds pages 8 to 15.

- **TP285 (TRIST).** Grounding test point TP285 disables all ROMs on the board, forcing them into their tri-state condition.

- **TP301 (FDIS).** Grounding test point TP301 disables the Fix-It.

ROM OPTION BOARD (670-5064-XX)

The older type of Option 35 circuit board is labelled "ROM Option Board," and is found in only a few older 4025s. Unlike the newer board (the ROM Expansion Board), the ROM Option Board has no provision for switching different "pages" of memory into memory addresses X'5000'-X'5FFF'. Consequently, this board cannot be used with the newer firmware options; it is superseded by the ROM Expansion Board, which is described earlier in this section.

The ROM Option Board has sockets to hold four ROMs, at addresses X'5000'-X'5FFF', X'6000'-X'6FFFF', X'7000'-X'7FFF', and X'8000'-X'8FFF'.

The last of these ROM sockets (address X'8000'-X'8FFF') is not used.

Fix-It

The ROM Option Board appears in Schematic 18-1. Like the ROM Expansion Board and the Processor Board, it includes a Fix-It. The Fix-It FPLA (a type IM5200 Field Progammable Logic Array) monitors the fourteen most significant address bits, looking for memory addresses which have been "flagged" (programmed into the FPLA). Each flagged location is a four-word block of memory (because the FPLA ignores the least significant two address bits). When the FPLA detects a flagged address, its F0 output goes low and is used for an error flag. This causes RDIS (ROM Disable) to go low, disabling all ROMs in the terminal except the Fix-It ROM. The FPLA outputs F2 through F7 then addresses the Fix-It FPROM (Field Programmable Read Only Memory) which puts its data on the bus. The address in the Fix-It FPROM corresponding to the output of the FPLA is as follows:

A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	F7	F6	F5	F4	F3	F2	BA1	BA0

Here, A0-A9 are the address inputs of the FPROM, F2-F7 are outputs of the FPLA, and BA0-BA1 are the two least significant bits of the address bus.

Note that the FPROM output, together with the least significant bits of the address bus, addresses only $2^{16} = 256$ words in the FPROM. The FPROM, however, holds a total of 1024 words; the other 768 words are addressed directly from the terminal's address bus. These words occupy addresses X'0500'-X'07FF'.

When the Fix-It detects a flagged address, its F0 output steers two 74LS157 data selectors, switching the FPROM's address inputs A2 through A9 from the memory address bus (lines BA2-BA9) to the FPROM outputs F2-F7.

Section 19

OTHER OPTIONS

OPTION 36 — PERIPHERALS ROMS (4025 ONLY)

Option 36 consists of two ROMs which hold the firmware for use with the RS-232 and GPIB Peripheral Interfaces (Options 3 and 4). These ROMs are installed as ROM2 (U141) and ROM3 (U151) on the Option 35 ROM Expansion Board. Figure 19-1 shows their locations.

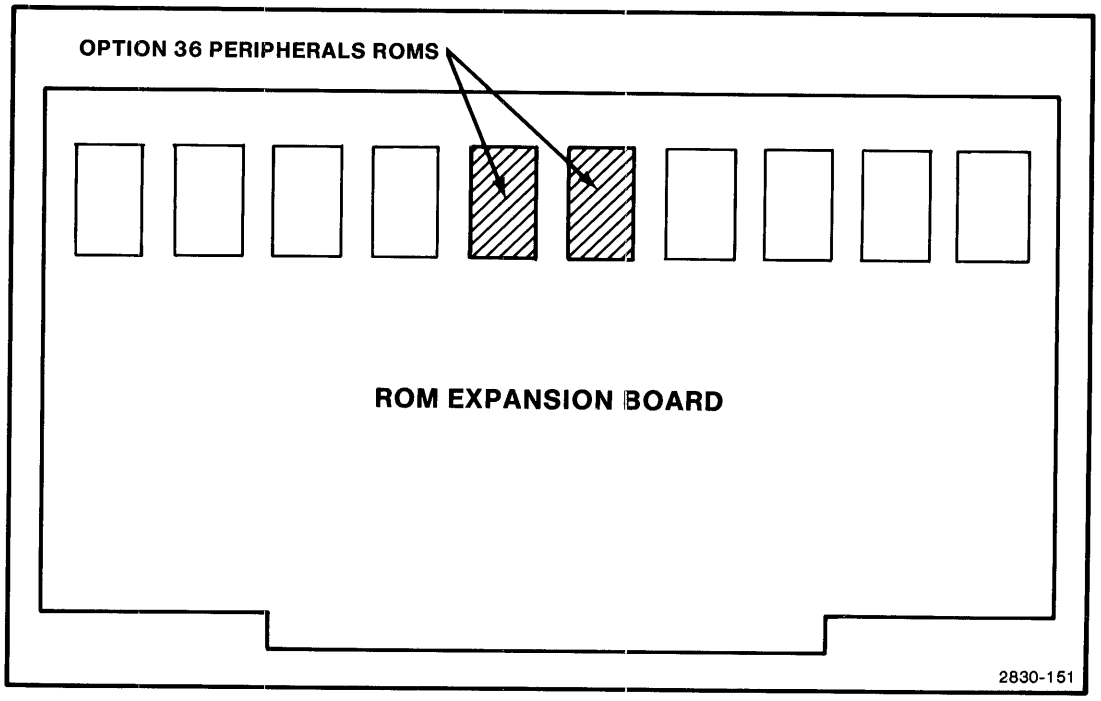


Figure 19-1. Location of Option 36 Peripherals ROM.

OPTION 40 — HARD COPY AND VIDEO OUT (4025 ONLY)

Option 40 consists of connectors and ribbon cables to bring the hard copy and external video signals out to the 4025's rear panel. These are shown in Schematic 19-1.

OPTION 41 — SELF TEST (4025 ONLY)

Option 41 consists of a ROM holding firmware for the 4025 to perform certain diagnostic tests on itself. This ROM is normally installed as ROM9 (U101) on the Option 35 ROM Expansion Board; it occupies memory locations X'7000' to X'7FFF'.

The Option 41 ROM may also be installed in place of System ROM number one (addresses X'1000' to X'1FFF') on the Processor Board. When installed there, it causes the 4025 to cycle continuously through a "test loop" routine for use with the TEKTRONIX 851 Digital Tester.

For information on how to use the Option 41 ROM, see the 4025 Option 41/851 Diagnostic Instruction Manual.

OPTION 48 — 220V, 50Hz

When a 4024/4025 terminal is "equipped with Option 48," it has its power supply wired for operation from 220-240 volt power mains. The procedure for changing the power supply voltage is described in Volume 2 of this Service Manual.

Appendix A

EXAMPLES OF COMMANDS

The following examples of 4024/4025 commands all assume that the command character is the exclamation point. <CR> denotes the RETURN key. For detailed information on those commands, see the 4024/4025 Programmer's Reference Manual.

ALLOCATE (4025 Only; Requires Options 4 and 36)

Initializing an unmmarked tape:

!ALL TA1 0,2,5000<CR> !ALL T 0,2,5000<CR>	Initializes the unmarked tape in tape unit 1, and creates two files (files 1 and 2) of 5000 bytes. If the tape has already been marked, this command destroys all old information on the tape.)

Allocating file space on a marked tape

!ALL TA1 1,2,5000<CR>	Creates two new files on tape unit 1, beginning with file 1. Each new file contains 5000 bytes.
!ALL TA1 7,4,8000<CR>	Creates four new files on tape unit 1, beginning with file 7. Each new file contains 8000 bytes.

ATTRIBUTE (Workspace)

Font Attributes

!ATT 0<CR>	Defines a new field beginning at the cursor position; displays characters from Font 0 (the standard font) in this field.
!ATT 30 E<CR>	Defines a new field beginning at the workspace cursor position: when characters are entered in this field, the 4025 displays the corresponding characters from Font 30, using an enhanced background.

Logical Attributes

!ATT A<CR>	Defines an alphanumeric field beginning at the cursor position.
!ATT N<CR>	Defines a "numeric" unprotected field beginning at the cursor position. In form fillout mode, only characters with ADEs 32-63 can appear in this field.
!ATT P<CR>	Defines a protected field, beginning at the cursor position. In form fillout mode, this field cannot be typed into or erased.
!ATT PM<CR>	Defines a protected modified field beginning at cursor position. This field is transmitted to the computer with each subsequent SEND MOD command.

Visual Attributes

!ATT E<CR>	Displays the field beginning at the cursor position with an enhanced background.
!ATT U<CR>	Displays the field beginning at the cursor position with an underline. (The 4024 lacks this capability; it treats an ATTRIBUTE U command as if it were an ATTRIBUTE S command.)
!ATT I<CR>	Displays the field beginning at the cursor position with an inverted background: dark on light rather than light on dark. (The 4024 lacks this capability; it treats an ATTRIBUTE U command as if it were an ATTRIBUTE E command.)
!ATT S<CR>	Displays the field beginning at the cursor position with the standard visual attribute: neither enhanced, nor underlined, nor inverted.

Combined Attributes (No spaces allowed between parameters):

!ATT E-UI<CR>	Displays the field beginning at the cursor position as blinking between "enhanced" and "underlined and inverted." The dash before the "underlined inverted" attribute creates the blinking attribute.
!ATT 1 PS-E<CR>	Defines a protected field beginning at the cursor position; blinks that field between the visual attributes of "standard" and "enhanced." When characters are entered, this field displays the corresponding Font 1 characters. (Requires Option 32)

BACKTAB

!BAC 5<CR>	Equivalent to pressing the BACKTAB key five times.

BAUD

!BAU 300,1200<CR>	Sets the transmitting baud rate to 300 baud, and the receiving rate to 1200 baud.
!BAU 9600<CR>	Sets both transmitting and receiving baud rates to 9600 baud.

BELL

!BEL<CR>	Sounds the terminal's bell. (This is also done by sending the ASCII BEL character, CTRL G.)

BUFFERED

!BUF<CR> !BUF Y<CR>	Places the terminal in buffered mode.
!BUF N<CR>	Removes the terminal from buffered mode.

CLEAR

!CLE<CR>	Clears all the programmed key definitions; all keys revert to their default definitions.

COMMAND

!COM #<CR>	Changes the command character to the "number sign" (#).
!COM 19<CR>	Changes the command character to the ASCII character whose decimal equivalent is 19: the "DC3," or "device control #3" character.

COPY

Copy to Printer

(On the 4024, requires Option 3; on the 4025, requires Options 3 and 36.)

!COPY WO/N P<CR>	Copies the contents of the workspace to the printer. All attribute codes are ignored and all ruling characters are converted to asterisks.
!COPY H P<CR>	Copies data from the host computer to the printer. The COPY ends when the host sends the current end-of-file string.

Copying from the workspace

!COP WO TA1.3<CR> !COP W T.3<CR>	Copies the workspace contents into file 3 of tape unit 1. If the workspace holds a form, this command copies the entire form (both protected and unprotected fields and their corresponding attribute codes). (4025 only; requires Option 4 and 36)
!COP W TA2<CR>	Copies the workspace contents into the next available file on tape unit 2. (4025 only; requires Options 4 and 36)

!COP W/U TA2<CR>	If the workspace holds a form, copies data from only the unprotected fields into the next available file on tape unit 2. (4025 only; requires Option 4 and 36)
!COP W/N P<CR>	Copies the workspace contents to the printer, ignoring attribute codes and converting all ruling characters to asterisks. (On the 4024, requires Option 3 and a TEKTRONIX 4642 Printer; on the 4025, requires Options 3 and 36, and a printer.)

Copying to the workspace

!COP T.1 W<CR> !COP T.1 <CR>	Copies file 1 of tape unit 1 into the workspace. (4025 only, requires Option 4 and 36)
!COP T.1/N W<CR>	Copies file 1 of tape unit 1 into the workspace, displaying commands as text rather than executing them. (4025 only; requires Options 4 and 36)

Copying from the Host

!COP H TA1.3<CR>	Copies the host file into file 3 of tape unit 1. This file is not displayed on the 4025 screen. (4025 only; requires Options 4 and 36)
!COP H/P TA1.3<CR>	Copies the host file into file 3 of tape unit 1, translating VECTOR, RVECTOR, STRING, and PASS commands into plotter format. (4025 only; requires Options 4 and 36)
!COP H/D/P TA1<CR>	Copies data from the host computer into the next available file of tape unit 1. Displays this data on the screen and translates VECTOR, RVECTOR, STRING, and PASS commands into plotter format before sending to the plotter. (4025 only; requires Options 4 and 36)

Copying from a GPIB Device to a GPIB Device

!COP TA2.3/P PL1<CR>	Copies file 3 of tape unit 2 to plotter 1, translating it to plotter language. The file is not displayed on the 4025 screen. (4025 only; requires Options 4 and 36)

DCHAR

!DCH 5<CR>	Equivalent to pressing the DELETE CHAR key five times.

DELAY

!DEL 20<CR>	Sets (to at least 20 milliseconds) the length of time that the 4025 delays before responding to a prompt from the host computer.

DFONT

!DFO 21<CR>	Deletes character font number 21, so that the memory used for holding those characters may be used instead as graphics memory.

DIRECTORY (4025 Only; Requires Options 4 and 36)

!DIR TA1 MO<CR> !DIR T<CR>	Outputs tape unit 1 file headers to the monitor. Each file header lists the file number, file type, and length of the file (in blocks). (Output device may be any non-GPIB device.)

DISCONNECT (4025 Only; Requires Option 1)

!DIS<CR>	Tells the 4025 to signal the modem to disconnect the data communications line.

DLINE

!DLI 6<CR>	Equivalent to pressing the DELETE LINE key six times; deletes six lines of text from the display.

DOWN

!DOW 15<CR>	Moves the cursor down 15 lines; like pressing the "down" cursor movement key 15 times.

DUPLEX (4025 Only; requires Option 1)

!DUP<CR> !DUP F<CR>	Sets the 4025 for full duplex.
!DUP H<CR> !DUP H S<CR> !DUP H S L<CR>	Sets the 4025 for half duplex with supervisor. In buffered mode the prompt condition is line turnaround only.
!DUP H S P<CR>	Sets the 4025 for half duplex with supervisor. In buffered mode the prompt condition is the prompt string plus line turnaround.
!DUP H N<CR> !DUP H N L<CR>	Sets the 4025 for half duplex normal. In buffered mode the prompt condition is line turnaround only.
!DUP H N P<CR>	Sets the 4025 for half duplex normal. In buffered mode the prompt condition is the prompt string plus line turnaround.

ECHO

!ECH L<CR>	Sets the 4024/4025 for "local echo."
!ECH R<CR>	Sets the 4024/4025 for "remote echo."

EOF

(Always provided in the 4024; in the 4025, requires Option 36.)

!EOF /abc/<CR>	Sets the end-of-file string to the ASCII characters <a>,<b>,<c>. This string is used in COPY commands to mark the end of a file being sent from the host computer.
!EOF<CR>	Sets the end-of-file string to its default value, /*.

EOL

!EOL<CR>	Sets the end-of-line string to "carriage return."
!EOL 13 10<CR>	Sets the end-of-line string to the ASCII characters whose decimal equivalents are 13 and 10: <CR>, <LF> ("carriage return, line feed").
!EOL /abcDEF/ 13 10<CR>	Sets the end-of-line string to to the ASCII characters <a>,<b>,<c>,<D>,<E>,<F>,<CR>,<LF>.

ERASE

!ERA<CR>	If this command comes from the host computer or a peripheral device, the terminal erases the scroll (workspace or monitor) which receives text from the host computer. If this command is typed on the keyboard, the terminal erases whichever scroll receives text from the keyboard.
!ERA W<CR>	Erases the workspace.
!ERA M<CR>	Erases the monitor.
!ERA G<CR>	Erases the contents of the graphic area.

FIELD

!FIE @<CR>	Sets (to "@") the character which, in form fillout mode, precedes fields when they are transmitted to the host computer.
!FIE 02<CR>	Selects the ASCII character whose decimal equivalent is "2" as the character which precedes fields when they are transmitted to the host computer.
!FIE 2<CR>	Selects the numeral 2 as the character to precede fields when they are transmitted to the host computer.

FORM

!FOR<CR> !FOR Y<CR>	Puts the 4024/4025 in form fillout mode.
!FOR N<CR>	Removes the terminal from form fillout mode.

GRAPHIC

!GRA 5,9<CR>	Sets up a graphic area in the workspace, containing all 80 columns of rows 5 through 9.
!GRA 5,9,10,70<CR>	Sets up a graphic area in the workspace, starting at row 5 and ending at row 9, and containing columns 10 through 70 in those rows.

GTEST

!GTE<CR>	Tests each bit of graphic RAM (random access memory). For each font, indicates whether graphic RAM is installed, and whether this RAM is OK. If faulty RAM is found, displays and error code: a decimal number whose binary equivalent has a 1 for each bit which was found in error. For example, the message FONT 30 RAM ERROR 128 would be interpreted as follows: Decimal 128 is the same as binary 10000000, so the fault lies in the RAM holding the most significant bit of bytes in character font 30. If font 30 is installed in Bank 0 of the Graphic Memory Board RAMs, then the fault may be the RAM holding Bank 0's RRD7 bits: U55 on the Graphic Memory Board.

HCOPY (4025 Only; Requires Option 41)

!HCO W<CR>	Copies one 53-line page from the workspace onto a TEKTRONIX 4631 Hard Copy Unit, starting with the first line of the workspace which is in view on the screen.
!HCO 3 M<CR>	Copies three pages from the monitor onto a 4631 Hard Copy Unit. Copy begins with the first line of the monitor which is visible on the screen, and continues for three 53-line pages.
!HCO S<CR>	Copies the contents of the screen onto a 4631 Hard Copy Unit.

HRULE (Requires Option 32)

!HRU 3,5,15,2<CR>	Starts at row 3, column 5 in the workspace, and draws a horizontal ruling through 15 character positions. The ruling is a double line.
!HRU 3,5,15,1<CR> !HRU 3,5,15<CR>	Starts at row 3, column 5 in the workspace, and draws a horizontal ruling through 15 character positions. The ruling is a single line.

ICHAR

!ICH<CR>	Puts the 4024/4025 in insert mode. (The terminal leaves insert mode on receiving a cursor movement character or a command which moves the cursor.)

ILINE

!ILI 3<CR>	Inserts three lines in the text, starting at the current cursor location; equivalent to pressing INSERT LINE three times.

JUMP (Workspace Only)

!JUM<CR>	Moves the cursor to the start of the first line of the workspace.
!JUM 15<CR>	Moves the cursor to the start of the 15th line of the workspace.
!JUM 15,26<CR>	Moves the cursor to row 15, column 26 in the workspace.

KILL (4025 Only; Requires Options 4 and 36)

!KIL TA1.5<CR>	Releases the space used to store file 5 on tape unit 1. This space may then be used to store new information.

LEARN

!LEA Q /abc/<CR>	Programs the shifted version of the Q key (uppercase Q) to mean the string of characters, <a>,<b>,<c>.
!LEA Q /abc/ 13<CR>	Programs the uppercase Q key to mean the string of characters, <a>,<b>,<c>,<CR>. The [fs]CR> ("carriage return") character is represented in the LEARN command by its ASCII decimal equivalent, the number 13.
!LEA F1 /abc/<CR>	Programs function key number 1 to mean the string, <a>,<b>,<c>.
!LEA S1 /abc/<CR>	Programs the shifted version of function key number 1 to mean the string <a>, <b>, <c>.

!LEA F2/!WOR 25!GRA 1,25!MON H;READY/13<CR>

Assigns the delimited command string, followed by a carriage return, to function key number 1.

LEFT

!LEF 5<CR>	Moves the alpha cursor five character positions to the left. If necessary, the cursor wraps around to the previous line. Equivalent to pressing the left cursor key five times.

LINE

!LIN<CR>	Sets the line type for subsequent VECTOR commands to "line type 1" — a solid line.
!LIN 5<CR>	Sets the line type to "line type 5."
!LIN E<CR>	Sets the line type to "line type E." Subsequent VECTOR commands will draw "erase vectors," which erase anything in their paths.
!LIN P<CR>	Sets the line type to "line type P." Subsequent VECTOR commands will plot points, rather than draw lines.

MARGINS (Workspace Only)

!MAR<CR>	Sets the left margin to column 1 of the workspace, and the right margin to column 80.
!MAR 5<CR>	Sets the left margin to column 5, and the right margin to column 80.
!MAR 10,70<CR>	Sets the left margin to column 10, and the right margin to column 70. Typing in column 71 will ring the terminal's bell.

MONITOR

!MON 5 H K<CR>	Creates a workspace. Reserves the bottom 5 lines of the screen for displaying the monitor, leaving the 29 lines above for displaying the workspace. Directs text from the host computer and the keyboard to the monitor.
!MON H<CR>	Directs text from the host computer to the monitor.
!MON K<CR>	Directs text from the keyboard to the monitor.
!MON<CR>	If this command is typed on the keyboard, it directs text from the keyboard to the monitor. If this command comes from the host computer, it directs text from the host computer to the monitor.

PARITY

!PAR E<CR>	Sets the 4024/4025 to even parity.
!PAR O<CR>	Sets the terminal to odd parity.
!PAR N<CR>	Sets parity to "none;" the terminal ignores the parity bit on input, and sets it to "0" on output.
!PAR D<CR>	Sets parity to "data;" the parity bit is available for use as a data bit.
!PAR H<CR>	Sets parity to "high;" the 4024/4025 ignores the parity bit on input, and sets it to "1" on output.

PASS

!PAS /S3.584 5.632/<CR>	Used during a COPY operation with plot translation on effect (/P switch set in COPY command). Passes the plotter-style command string S3.584 5.632, unmodified, to the output device specified in the COPY command.

PERIPHERALS (4025 Only; Requires Option 3 or 4 and Option 36)

!PER M<CR>

Prints a peripherals data list in the monitor. For each device attached to the terminal, this list displays the device mnemonic, the GPIB address, and the option data field. (GPIB address is blank for the printer.) Final entry in the list specifies the end-of file string.

!PER P<CR>

Prints a peripherals data list on the printer.

TA1-TA4	Tape units 1-4
PL1,PL2	Plotters 1,2
P or PR	Printer

PROMPT

!PRO /abc/<CR>

Sets the prompt string to <a>,<b>,<c>. In buffered mode, the terminal must receive this string from the host before it sends each line of text from its transmit buffer.

!PRO 13 10<CR>

Sets the prompt string to the ASCII characters whose decimal equivalents are 13 and 10: <CR>,<LF>.

!PRO<CR>

Sets the prompt string to its "default" value, the single character <LF> ("line feed").

REPORT (From the Host Computer Only)

!REP 00<CR>	Causes the 4024/4025 to send a report to the host computer; indicates the number of unused blocks of memory and the system status byte.
!REP 01<CR>	Reports to the host computer the status of the workspace cursor; gives the row and column location of the cursor and the character stored at that location.
!REP 05<CR>	Reports to the host computer the status of device 5 (tape unit 2); indicates whether that device is present, the last tape error code, and whether the tape unit is SET to 4051 or 4923 compatability.
!REP 12<CR>	Reports the status of plotter 1; indicates whether the plotter is present; if it is present, indicates the ASCII integer value of plotter status word zero (0).
!REP 14<CR>	Reports the status of the printer; indicates whether the printer is present; if it is present, indicates the L/F option and the printer delay.

Peripheral device numbers:

00 — system status block
01 — workspace cursor
04-07 — tape units 1-4
12,13 — plotters 1,2
14 — printer

RDOWN

!RD0<CR>	Rolls the workspace or monitor scroll down five lines. (Equivalent to pressing the "down" scrolling key five times.) The action stops when the top of the scroll is in view on the screen.

RIGHT

!RIG 5<CR>	Moves the cursor five character positions to the right.

RUP

!RUP<CR>	Rolls the workspace or monitor scroll up five lines. (Equivalent to pressing the "up" scrolling key five times.) The action stops when the last line of text already entered in the scroll comes into view on the screen.

RVECTOR

!RVE 5,5,10,10<CR>	"Relative vector" command. Same as a VECTOR command, except the coordinates are centered on the last beam position (end of the last line segment in a VECTOR or RVECTOR command), rather than being centered on the point (0,0).

SEND

!SEN A<CR> !SEN<CR>	If not in form fillout mode, sends to the computer all the data in the workspace, including attribute codes, which are encoded as ATTRIBUTE commands. If in form fillout mode, sends to the computer the contents of all the "blanks" of the form (the unprotected fields).
!SEN M<CR>	If not in form fillout mode, functions like SEND ALL; sends the entire contents of the workspace to the computer. If in form fillout mode, sends to the computer the contents of those blanks ("unprotected fields") which have been modified since the last SEND MOD command. Sends also the contents of those protected fields which have been marked with the "modified" logical attribute.

SET (4025 ONLY; Requires Option 3 or 4, and Option 36)

!SET PL1 15<CR>	Instructs the 4025 that plotter 1 is present at GPIB address 15. This must agree with the address switch settings on the plotter. GPIB addresses are specified for those and only those devices present and powered up on the bus. (Requires Option 4 and 36)
!SET TA1 8 4051<CR> !SET T 8<CR>	Instructs the 4025 that tape unit 1 is present at GPIB address 8, and instructs this tape unit to write data in 4051 compatible format. (GPIB address for a tape unit must be even.) (Requires Option 4 and 36)
!SET TA2 10 4923<CR>	Instructs the 4025 that tape unit 2 is present at GPIB address 10, and instructs this tape unit to write data in 4923 compatible format. (Requires Option 4 and 36)
!SET PR F <CR>	Instructs the 4025 to use the ASCII form feed character [<FF>] as the page separator; when the printer receives a <FF>, it will begin a new page. Also sets the printer delay to 0.3 seconds; after sending a <CR>, <LF>, or <FF> to the printer, the 4025 waits 0.3 seconds before sending another character. (4025 Only; requires Options 3 and 36)
!SET PR L<CR>	Instructs the 4025 that the printer does not treat a <FF> character as the page separator. The 4025 will replace a <FF> with the number of <LF>'s required to begin a new page. (4025 only; requires Option 3 and 36)
!SET PR 0<CR>	Instructs the 4025 to communicate with the printer using flagged simplex protocol. (After sending a <CR>, <LF>, or <FF>, the 4025 waits for a DTR (Data Terminal Ready) signal from the printer before sending another character.) (4025 only; requires Options 3 and 36)

SHRINK (4025 Only)

!SHR<CR> !SHR Y<CR>	Causes the 4025 to "shrink" X- and Y-coordinates in subsequent graphic commands, by a factor of 5/8. This accomodates the 4025 to the range of possible coordinates in 4010-style graphic commands.
!SHR H<CR>	Causes the 4025 to "shrink" Y-coordinates (but not X-coordinates) in graphic commands, by 7/8. This pre-distorts graphs drawn on the 4025's screen, so that they appear in the proper proportion when copied on a 4631 Hard Copy Unit.
!SHR B<CR>	Causes the 4025 to to perform both "ordinary" and "hard copy" shrink operations on coordinates in subsequent graphic commands. Multiplies X-coordinates by 5/8, and Y-coordinates by 35/64.
!SHR N<CR>	Removes the 4025 from "shrink mode," so that X- and Y-coordinates in subsequent graphic commands are not altered.

SNOOPY

!SNO<CR>	Puts the 4024/4025 in "snoopy mode." (In snoopy mode, the "non-printing" ASCII characters are represented on the screen by two-letter abbreviations.)
!SNO N<CR>	Removes the terminal from snoopy mode.

STOPS

!STO 5,10,15<CR>	Puts tab stops in columns 5, 10, and 15.

STRING (4025 Only; Requires Option 23, 24, 25, or 26)

!STR /EARNINGS/<CR>	Inserts the string EARNINGS in the graphics area. The first character in this string appears at the character position containing the graphic cursor.

SYMBOL (4025 Only; Requires Option 23, 24, 25, or 26)

!SYM 8,30,0,0,0,0,2,52,72,72,52,2,0,255<CR>

Defines chararacter 8 of Font 30. Each integer after the "30" (for font 30) is converted to an 8-bit binary equivalent. The 0-1 pattern of this binary equivalent specifies which dots in a given row of the 8X14 dot matrix will be lighted.

SYSTAT

!SYS<CR>

Displays the system status message in the monitor.

TAB

!TAB 2<CR>

Performs the same action as pressing the TAB key twice.

If not in form fillout mode, each tab advances the cursor one tab stop or to the beginning of the next line.

If in form fillout mode, each tab advances the cursor to the next blank (unprotected field) of the form.

TEST

!TES<CR>

Checks the terminal's memory. The terminal erases the entire display list and creates a 34-line window. It then tests each character location, displays the four checksums for the system ROM (Read Only Memory), and prints out characters (Read Only Memory). After this, all four lighted keys are turned on, and all Font 0 and Font 1 characters are displayed in snoopy mode. Then samples of different visual attributes are displayed, the lighted keys are turned off, and the terminal bell is rung. (This resets the terminal.)

UP

!UP 5<CR>

Moves the cursor up five lines; similar to pressing the "up" cursor movement key five times.

VECTOR (4025 Only; Requires Option 23, 24, 25, or 26)

!VEC 0,0,100,100,0,10<CR>	Draws a line in the graphic area of the workspace, starting at point (0,0) and continuing to (100,100); draws another line from there to the point (0,10).

VRULE (Requires Option 32)

!VRU 3,5,15,2<CR>	Starts at workspace row 3, column 5, and draws a vertical ruling downward through 15 character positions. The ruling is a double line.
!VRU 3,5,15,1<CR> !VRU 3,5,15<CR>	Starts at workspace row 3, column 5, and draws a vertical ruling downward through 15 character positions. The ruling is a single line.

WORKSPACE

!WOR 20 H K<CR>	Creates a workspace, and allots the top 20 lines of the screen for displaying it. Directs text from the host computer and the keyboard to that workspace.
!WOR H<CR>	Directs text from the host computer to the workspace.
!WOR K<CR>	Directs text from the keyboard to the workspace.

Appendix B

ASCII CODE CHART

Table B-1 lists the ASCII characters and their decimal equivalents. Table B-2 lists the ASCII control characters, with their "snoopy mode" mnemonics.

Table B-1
ASCII CODE CHART

BITS B4	B3	B2	B1	B7 B6 B5 = Ø Ø Ø	Ø Ø 1	Ø 1 Ø	Ø 1 1	1 Ø Ø	1 Ø 1	1 1 Ø	1 1 1
				CONTROL		HIGH X & Y GRAPHIC INPUT		LOW X		LOW Y	
Ø	Ø	Ø	Ø	NUL 0	DLE 16	SP 32	0 48	@ 64	P 80	\ 96	p 112
Ø	Ø	Ø	1	SOH 1	DC1 17	! 33	1 49	A 65	Q 81	a 97	q 113
Ø	Ø	1	Ø	STX 2	DC2 18	" 34	2 50	B 66	R 82	b 98	r 114
Ø	Ø	1	1	ETX 3	DC3 19	# 35	3 51	C 67	S 83	c 99	s 115
Ø	1	Ø	Ø	EOT 4	DC4 20	$ 36	4 52	D 68	T 84	d 100	t 116
Ø	1	Ø	1	ENQ 5	NAK 21	% 37	5 53	E 69	U 85	e 101	u 117
Ø	1	1	Ø	ACK 6	SYN 22	& 38	6 54	F 70	V 86	f 102	v 118
Ø	1	1	1	BEL 7	ETB 23	' 39	7 55	G 71	W 87	g 103	w 119
1	Ø	Ø	Ø	BS 8	CAN 24	(40	8 56	H 72	X 88	h 104	x 120
1	Ø	Ø	1	HT 9	EM 25	) 41	9 57	I 73	Y 89	i 105	y 121
1	Ø	1	Ø	LF 10	SUB 26	* 42	: 58	J 74	Z 90	j 106	z 122
1	Ø	1	1	VT 11	ESC 27	+ 43	; 59	K 75	[91	k 107	{ 123
1	1	Ø	Ø	FF 12	FS 28	, 44	< 60	L 76	\ 92	l 108	¦ 124
1	1	Ø	1	CR 13	GS 29	- 45	= 61	M 77	] 93	m 109	} 125
1	1	1	Ø	SO 14	RS 30	. 46	> 62	N 78	^ 94	n 110	~ 126
1	1	1	1	SI 15	US 31	/ 47	? 63	O 79	_ 95	o 111	RUBOUT (DEL) 127

2830-142

Table B-2
CONTROL CHARACTERS

Mnemonic	Usual ASCII Abbrev.	Name of Character	Keys to Press
NU	NUL	Null	CRTL- @
SH	SOH	Start of Heading	CTRL-A
SX	STX	Start of Text	CTRL-B
EX	ETX	End of Text	CRTL-C
ET	EOT	End of Transmission	CTRL-D
EQ	ENQ	Enquiry	CTRL-E
AK	ACK	Acknowledgement	CTRL-F
BL	BEL	Bell	CTRL-G
BS	BS	Backspace	CTRL-H
HT	HT	Horizontal Tab	CTRL-I
LF	LF	Line Feed	CTRL-J
VT	VT	Vertical Tab	CTRL-K
FF	FF	Form Feed	CTRL-L
CR	CR	Carriage Return	CTRL-M
SO	SO	Shift Out	CTRL-N
SI	SI	Shift In	CTRL-O
DL	DLE	Data Link Escape	CTRL-P
D1	DC1	Device Control 1	CTRL-Q
D2	DC2	Device Control 2	CTRL-R
D3	DC3	Device Control 3	CTRL-S
D4	DC4	Device Control 4	CTRL-T
NK	NAK	Negative Acknowledgement	CTRL-U
SY	SYN	Synchronization Character	CTRL-V
EB	ETB	End of Transmission Block	CTRL-W
CN	CAN	Cancel	CTRL-X
EM	EM	End of Medium	CTRL-Y
SB	SUB	Substitute	CTRL-Z
EC	ESC	Escape	CTRL-[
FS	FS	Field Separator	CTRL-\
GS	GS	Group Separator	CTRL-]
RS	RS	Record Separator	CTRL- ^
US	US	Unit Separator	CTRL-_ (underscore)

2830-143

Appendix C

SPECIFICATIONS

Table C-1

DISPLAY MECHANISM

Display type	Video monitor.
Screen size	30.0 cm (12.0 in) diagonal.
Usable display area	17.0 cm x 23.0 cm (6.7 in x 9.0 in)
Phosphor type	P-39 green phosphor
Video bandwidth	20 MHz
Raster lines	Standard 525 line scan, with 480 lines displayed.
Scan	30 Hz interlaced scan.
Refresh rate	
Dot	30 times/second
Frame	30 times/second
Field	60 times/second

Table C-2

DISPLAY CHARACTERISTICS

Cursor type	Wide underscore
Character size	7 x 9, in an 8 x 14 dot matrix. (Graphic cells are 8 x 14 matrix.)
Character sets Standard	64/96 upper and lower case ASCII. (A special "snoopy mode" is provided which displays all characters and commands. Snoopy mode enables the display of 128 characters.)
Optional	64 characters each, to a maximum of 6 character sets; or 128 characters each, to a maximum of 3 sets. Can include rulings or math characters, for example. With the graphics option, up to 16 character sets can be defined by the user.
Characters/line	80
Lines/display	34
Total characters/display	2720
Visual attributes	Characters can be displayed with shaded background, inverted fields (dark on light background rather than light on dark background), or underlines. In addition, the display can "blink" (alternate) between combinations of visual attributes.
Logical attributes	Fields can be protected, modified, alphanumeric, or numeric.

Table C-3

INTERFACE CHARACTERISTICS

Transmission speed		The transmitting and receiving baud rates may be independently selected up to 9600 baud. Baud rates can be entered from the keyboard or the host computer.
Interface with host computer		Asynchronous serial data communications, as follows:
Standard		RS-232 full duplex.
Optional	Option 1	RS-232 half duplex, with or without supervisory channel.
	Option 2	20 mA current loop.
	Option 11 (4025 only)	Polling controller (IBM 3270 compatibility). Converts IBM 3270 protocol to 4020-series polling protocol.
Optional interfaces with peripheral devices		
	Option 3	RS-232, transmit only, for printer.
	Option 4	IEEE 488-1975 GPIB (General Purpose Interface Bus), for TEKTRONIX 4924 Digital Cartridge Tape Drive or 4662 Interactive Digital Plotter.
	Option 10	Polling Interface—permits multiple terminal configurations on one host communications line. Terminals are under control of one 4025 with the Polling Controller (Option 11). Host computer interface is RS-232, terminal interface is current loop.
	Option 40	Hard Copy connector for TEKTRONIX 4631 Hard Copy Unit, and standard 75-ohm video output for external video monitor.
Video Signals		Generates 525 line interlaced 60 Hz composite video. Does not accept external synchronizing pulses.
Peripheral Devices		4642 Printer 4631 Hard Copy Unit 4924 Digital Cartridge Tape Drive 4662 Interactive Digital Plotter

Table C-4

BUFFERING AND EDITING CAPABILITIES

Size of display memory	4K bytes standard; may be expanded to 8K, 16K, or 32K (Options 20, 21, and 22.)
Parts of memory	
Monitor	Holds and displays conversational text, such as conversations between the user and the host computer, or commands to the terminal.
Workspace	Holds and displays text or forms. One or several pages of text can be saved in the workspace, edited, and later sent to the host or to a peripheral 4924 tape unit.

Table C-5

KEYBOARD SPECIFICATIONS

Number of keys	86
Lighted keys	4 (Insert Mode, TTY Lock, Numeric Lock, Command Lockout)
Programmable keys	80
Major keyboard functions	Typewriter keys, terminal function control, programmable function keys, numeric pad, cursor control, scrolling.
Tactile features	Curved profile, tactile feedback at typing "home" position.
Relationship to display	Detachable (flexible 8 foot cable)
Editing keys	Delete Character Delete Line Erase and Skip Insert Line Insert Mode

Table C-6

GRAPHICS SPECIFICATIONS

(For 4025's with Option 23, 24, 25, or 26.)

The maximum amount of graphics memory that may be needed depends on the number of graphics "cells" (character positions) in the part of the wokspace used for holding the graph. Display memory is also required to put graphics in the workspace.

More complex graphs require more memory. Seldom, however, is a graphic display so complex and so dense that it uses the maximum amount of graphics memory.

Amount of Graphics Memory	
Option 23	4096 8-bit bytes (covers 256 cells).
Option 24	8192 bytes (covers 512 cells).
Option 25	16384 bytes (covers 1024 cells).
Option 26	32768 bytes (covers 2048 cells).
Line types	Solid lines. Seven styles of dashed lines. Single points. "Erase vectors", which erase previously drawn lines.
Type of display	Dot matrix: each graphics cell (character position) has 14 rows of 8 dots each.
Resolution	28 addressable points/cm (71/inch).
User-definable character sets	That part of graphics memory not used for displaying graphic information may be used for defining alternate character fonts with which to display text. Up to 16 fonts may be defined by the operator or the host computer.
Characters per font	128
Number of fonts available	
Option 23	2
Option 24	4
Option 25	8
Option 26	16

Table C-7

PHYSICAL CHARACTERISTICS

Weight	27.2 kg (60 lb)
Dimensions	
Cabinet	
Height	31.7 cm (12.5 in)
Width	44.5 cm (17.5 in)
Depth	54.0 cm (21.25 in)
Keyboard	
Height	7.6 cm (3.0 in)
Width	45.7 cm (18.0 in)
Depth	23.5 cm (9.25 in)

Table C-8

POWER REQUIREMENTS

Line plug and power cord	15 ampere capability, detached.		
Input Line voltages	VOLTAGE	RANGE	FUSE
Standard	115 Vac	90-100 V (low) 105-126 V (med) 112-136 V (high)	3 A
Option 48	220 Vac	180-220 V (low) 208-252 V (med) 224-272 V (high)	1.5 A
Line frequency	49 to 61 Hz		
Power consumption	295 W maximum at 125 Vac		

INDEX