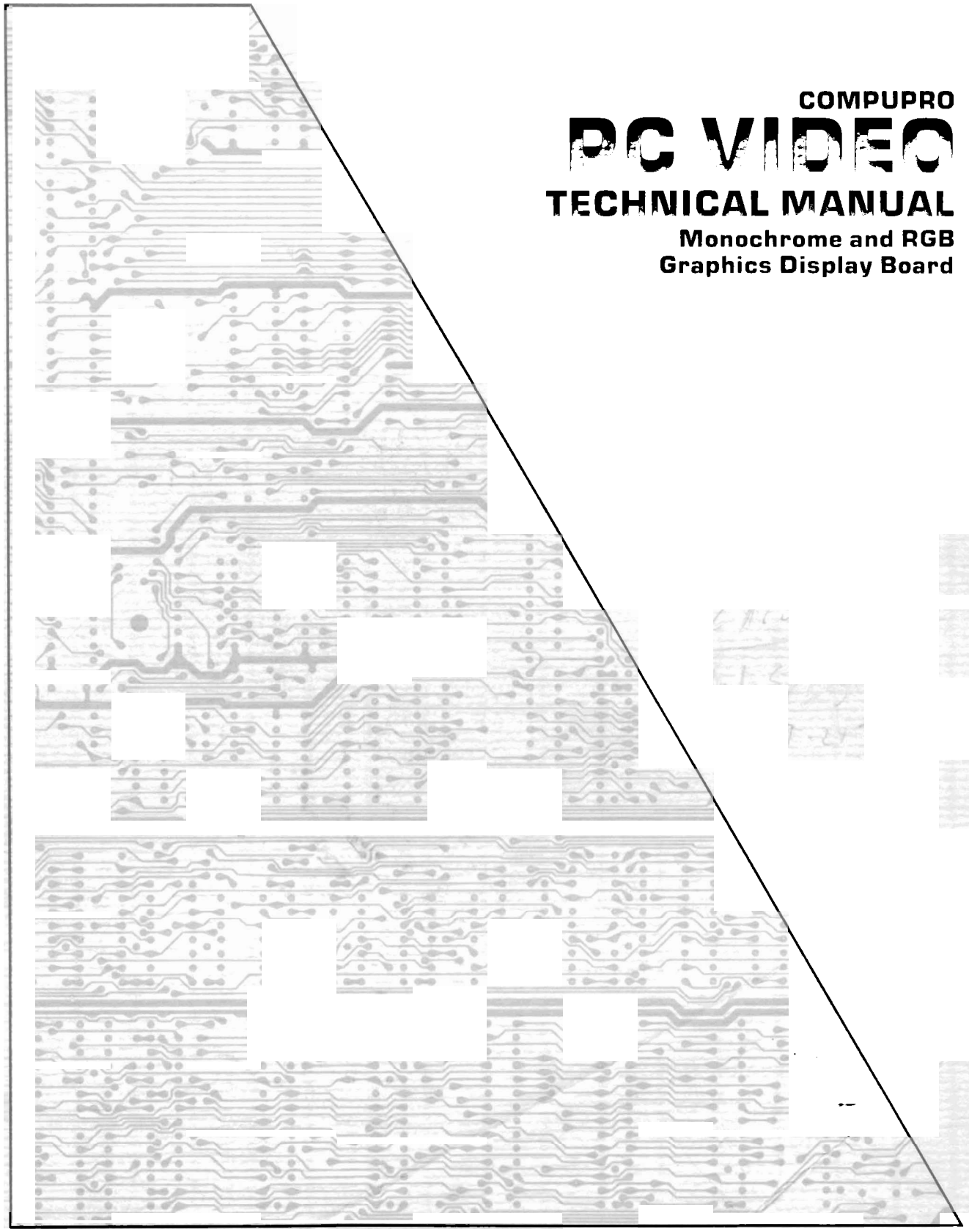
VIASYN™

COMPUPRO

PC VIDEO

TECHNICAL MANUAL

Monochrome and RGB Graphics Display Board



A493

$20.00

CONTENTS

LIST OF TABLES

ABOUT THE PC VIDEO BOARD

VIASYN's PC Video board is a high speed, bit-mapped, monochrome and RGB graphics display board. The PC Video interface operates with the IBM®-PC compatibility module included in CompuPro's Concurrent DOS™ 8-16™ operating system, as well as with Digital Research's GSX graphics interface software (included with Concurrent DOS 8-16).

The PC Video board has a monochrome as well as color display mode. A keyboard and light pen may also be attached.

The PC Video board meets all IEEE 696/S-100 bus specifications and includes the following features:

* 16 Kbytes of static CMOS RAM
* 24 address bits for memory, 16 address bits for I/O ports
* Automatic wait state generation
* Programmable 6845 video display controller with switch-set master mode
* Up to 8 boards can be accessed independently with unique addresses

HARDWARE SECTION

This section begins with a discussion of the types of display monitors, keyboards, and light pens that may be attached to the PC Video Graphics interface board. The choice of which display monitor to use should be a careful one as it will determine how to set the primary configuration switches and jumpers. This primary configuration will also select the permissible Master Display Modes. The Hardware Section is designed to be read completely, from start to finish, before any unusual installation or attempts at debugging.

ATTACHING DEVICES AND CABLES TO THE PC VIDEO BOARD

Choosing a Display Monitor

The PC Video graphics interface can operate with three different types of display monitors. Two of these, the Color and Monochrome displays, have very specific interface characteristics. An attempt to attach a display monitor, which does not conform to these characteristics, may cause problems beyond which VIASYN is normally able to support. **In particular, NOT ALL RGB type monitors will work with this interface board.**

Display monitors which may work correctly are "IBM-PC compatible." Check your monitor to see if the manufacturer states that it is IBM-PC compatible. This means that it has a 9 pin "D type" subminiature male connector to plug into the back of a computer. If the cable is of some other type, the monitor is probably not going to work in conjunction with the PC Video interface. If it is a Color monitor, and more than one display type is supported by the monitor, then it is essential that it be connected or switch selected to the RGB-I (Red, Green, Blue type #1) display configuration. Even if a particular monitor meets all of these qualifications, it still does not guarantee that the display monitor will operate in an entirely correct manner. To help remove any ambiguities in the selection process to obtain a fully functional display monitor, VIASYN has tested and found the models listed below to operate with 100% compatibility.

Table 1: Display Monitor Switch Settings

Recommended Color (RGB) Display Monitors:

1. IBM PC Color/Graphics Display Monitor
2. Zenith Data Systems model ZVM-135

If you are using one of these monitors or compatible ones, use the following switch settings on your PC Video board:

DIP switch S1 paddle #10 set to ON = Normal Vertical Sync
DIP switch S2 paddle #9 set to OFF = Color Mode

Recommended Monochrome Display Monitors:

1. IBM PC Monochrome Display

If you are using this monitor or one compatible, use the following switch settings on your PC Video board:

DIP switch S1 paddle #10 set to OFF = Invert Vertical Sync
DIP switch S2 paddle #9 set to ON = Monochrome Mode

DIP switch S1, paddle 10 selects the vertical sync polarity, which is opposite for Color and Monochrome display monitors. If you have a non-IBM monitor, this switch may need to be set the other way to make the screen synchronize its vertical display.

DIP switch S2, paddle 9 selects between the Color and Monochrome Display Monitor Modes by routing the master timing chain signals and selecting the correct portion of the character generator PROM. Once selected, ONLY the associated Color or Monochrome Master Display Modes may be programmed successfully into the board. If Color Mode is chosen, none of the Monochrome Display Modes will produce anything useful, and if Monochrome Mode is chosen, none of the Color Display Modes produce anything useful. A major difference between the CompuPro PC Video Graphics interface and the IBM Monochrome Adaptor is that the PC Video board will support either high or medium resolution graphics (similar to the IBM Color Graphics Adaptor) on a monochrome monitor. The IBM Monochrome Adaptor will not support either graphics mode.

Jumper J1 is an adjustment for the horizontal sync signal output to the display monitor. Changing this jumper from A-C to B-C will cause the data displayed on screen to move one character to the left. This adjustment exists to help set up various monitors to display default IBM-PC initialization sequences with the display data centered on the screen. If you have any difficulty with the position of the data image on the screen, first check the monitor alignment, but if necessary the horizontal position is readily adjusted by altering the values of certain CRT controller registers through software. If the shunt at jumper J1 is removed, there will be no horizontal sync output and the screen data becomes scrambled.

Jumper J1: A-C selects normal horizontal sync signal
B-C selects delayed horizontal sync signal

The third type of display monitor that may be used with your PC Video board is commonly connected to the PC Video interface via a single coaxial cable. The signal generated by the board and transmitted through the cable is RS-170 composite video compatible. The interface produces this signal on a second connector, independently of the data output on the 9 pin TTL

level connector. The interface board produces the RS-170 signal at a 1 Volt peak-peak level into a 75 ohm load. This is the normal signal level for most composite video display monitors. Note that the composite video signal is **Black and White only.** This means that no color information is encoded by the interface. There is a wide variety of "off the shelf" display monitors that will accept and operate correctly using this output signal.

Composite video type display monitors will usually have raster timing similar to the Color Mode, particularly when the interface has been programmed to match the standard IBM-PC display modes. DIP switch S2, paddle 9 should probably be set to the ON position. If you wish to use the PC Video interface with non-standard display modes or programming, then set the switch according to whichever master frequency (14.31318 or 16.257 MHz), and character generator is appropriate for your monitor and application. Jumper J1 and DIP switch S1, paddle 10 have no effect on the composite video output signal.

NOTE: It is possible to connect an RF modulator to the RGB output lines to generate composite color video. However, VIASYN does not support customized interface modules or connections.

Choosing a Keyboard

Nearly any keyboard which is IBM-PC plug compatible is also compatible with the PC Video interface board. The only restriction is that the keyboard draw no more that 0.5 amperes at 5 Volts from the interface board. If more current is needed, connect the keyboard to an external 5 Volt power supply.

Choosing a Light Pen

The light pen input cable connection provided at the top of the PC Video interface board has several possible interconnect configurations. In addition to ground, +5 Volts and +12 Volts power supply outputs which may be used to power the light pen, the light pen section of the connector has two LS-TTL level inputs. One of these LS-TTL inputs responds by generating a data available status signal for polled applications or an interrupt and is positive edge triggered. The other input is passed directly to a status port without latching or conditioning.

If you are considering using a light pen as an important factor in your system configuration, please be aware that it will not operate correctly with a "high persistence" display monitor. The IBM Monochrome Display Monitor has just such a high persistence screen and is incompatible with the use of a light pen. Most color display monitors do not have high persistence screens. Check the documentation that came with your monitor to determine which type of screen your monitor has.

Connecting Cables to the PC Video Board

The output from the PC Video board is derived from two connectors at the upper right hand side of the board. One of these is a 3 pin connector for Black and White composite video RS-170 output. Pin #2, the middle connection, is connected as the signal and pins #1 and #3 are connected to ground. This output should be connected via coaxial "RG" type cable, and may be of any length so long as the opposite end is terminated into 75 ohms.

The other input/output connector may have one of two physical cable configurations, depending on the intended use of the PC Video board. If only the RGB TTL output to either a Monochrome or Color IBM type monitor is needed, use only the DB-9 female connector at the rear of the box. It may be connected by using ribbon cable to a 10 pin header. This header connector should be attached to the left hand side of the PC Video main output connector to cover exactly ten pins.

If both a monitor and a keyboard need to be attached, use CompuPro cable #36 which has a 26 pin header for the PC Video board output connector. This cable splits in half, with the left half becoming a DB-9 female connector for TTL video output. The right half has signals for both the Light Pen and Keyboard inputs. Only the Keyboard signals are connected for cable #36, and terminate in a DIN-5 female connector, the same as used by IBM type keyboards. You may obtain keyboard extension cables from any computer supply store.

To use a light pen with the PC Video board, at present it is necessary for you to construct your own internal cable using the pinouts described in the following table:

Table 2: PC Video Main Output Connector

(Top View)

	14	15	16	17	18	x	20	21	22	23	24	25	26
Pin:	1	2	3	4	5	x	7	8	9	10	11	12	13

Pin #	Connector:	Function:
1	Video	Signal ground
2	Video	Signal ground
3	Video	Red gun control output signal
4	Video	Green gun control output signal
5	Video	Blue gun control output signal
6	No connection	
7	Keyboard	Power/signal ground
8	Keyboard	Power/signal ground
9	Keyboard	RESET* (negative true) output
10	Keyboard	KCLK* clock bi-directional signal
11	Keyboard	KDATA* keyboard input data signal
12	Keyboard	Power supply, +5 Volts
13	Keyboard	Power supply, +5 Volts at less than 0.5 amp
14	Video	Intensity control output signal
15	Video	Black/White control output signal
16	Video	Horizontal Sync pulse output signal
17	Video	Vertical Sync pulse output signal
18	No connection	
19	No connection	
20	Light Pen	Power/signal ground
21	Light Pen	Power/signal ground
22	Light Pen	LPEN-SW* light pen switch input signal
23	Light Pen	LPEN-INPUT* light pen pulse input signal
24	Light Pen	Power supply, +5 Volts, < 0.5 amp inc. keyboard
25	Light Pen	Power supply, +12 Volts at less than 0.1 amp
26	No connection	

Most IBM type Color or Monochrome monitors come with about 4 feet of cable for connection directly into the rear of the computer. This may be extended by another 4 feet by using a specially constructed extension cable available at most computer supply stores where monitors are sold. The monitor cables should not be extended beyond a single extra 4 foot length (a total of less than 10 feet because an "impedence mismatch" may occur).

The terms "impedence matched" or "impedence mismatched" are used when describing how a cable system carries its signal. If a cable and the devices at both ends are impedence matched, the signal carried between the two devices will not be distorted as it passes through. This allows the cable to be of any length. This is true for the PC Video RS-170 output, the interconnection cable and most RS-170 compatible monitors, where care has been taken so that all three have been impedence matched to 75 ohms.

Since the output needed to drive IBM type monitors is of "TTL signal level" type, it is not matched to any normal "RG type" transmission cable in either the monitor or the PC Video board output. Because of this, the length of cable which is used will affect this impedence mismatch and cause some of the signal to be reflected back from the monitor to the PC Video board. At certain precise lengths of cable (which are dependent on the "characteristic impedence" of the cable and the video clock master frequency), the entire signal will be reflected. Nothing will appear at the monitor even though the PC Video board is functioning perfectly. Unless you are extremely familiar with this phenomena and the methods and calculations needed to correct for it, **DO NOT** attempt to construct your own extension cable for the display monitor.

VIDEO DISPLAY INTERFACE MEMORY ADDRESSING

The PC Video graphics display interface has a dual port array of static RAM accessed by both the internal CRT controller and the external host processor. Internally, this memory appears as 16-bits by 8K words for display of alphanumeric or graphic data. To the host processor, the array occupies an area of either 16K bytes or 8K words of memory address space, depending on whether byte or word data transfers are performed.

The display memory array is addressed in multiples of 16K bytes by switch selection to determine the matching values of A14 through A21 on the IEEE 696/S-100 bus. Address lines A22 and A23 are decoded in the address selection PAL which is programmed to respond to an address match with both bits equal to zero as the standard release. DIP switch S1, paddles 1 thru 9, are used to select the read/write mode and base address of the display memory on the PC Video interface as shown in the following table:

Table 3: Switch Settings for Memory Addressing

"S1" SWITCH POSITION	ADDRESS BIT	
1	(Write Only Memory mode, ON = Read/Write)	
	A23 = A22 = '0'	
2	A21	
3	A20	ON = '0'
4	A19	
5	A18	
6	A17	OFF = '1'
7	A16	
8	A15	
9	A14	

Wait states are automatically generated by the PC Video interface to synchronize access by the host processor with an available data transfer window. These windows occur between successive internal data word transfers timed by the raster display process. There are no switch settings for the number of wait states because the synchronization time is variable. The delay period generated is dependent on the relationship of a bus request occuring with respect to the internal master timing cycle of the PC Video interface, but will be no longer than 850 nanoseconds for the worst case.

NOTE: Most DMA devices cannot directly read or write the RAM on the PC Video board. Due to the wait states inserted by the arbitration circuitry, most non-buffered DMA devices will overrun while reading or writing to the PC Video RAM.

The interface memory array may also behave as "Write Only Memory" (WOM) for the special case where there is other RAM that must appear in the same address space as the PC Video interface board.

If switch S1, paddle 1 is set to ON, then the PC Video memory will only respond to write operations from the bus. To function correctly, there must be other Read/Write RAM occupying the same address. This memory must also be of "8-16" type to match the response of the PC Video interface board. Write Only Memory mode will run somewhat faster than normal because no wait states are needed for display memory read operations.

To permit the use of more than one PC Video interface addressed at the same base location, the memory array on each board may be enabled or disabled by its bank select register. Board selection is performed via an output to relative port #0 which sets the state of each board's bank select register. When the register value is zero, the individual PC Video interface is disabled and will not respond to any data transfers in the memory address space. The bank select register function is given in detail in the "Video Display Interface Port Mapping" section of this manual.

When more than one PC Video interface is addressed in the same memory address space **AND** there is also additional RAM addressed in this space as well, then the phantom memory control should be enabled. This is done by setting DIP switch S2, paddle #10 to the ON position. **In all other cases this switch should be in the OFF position.** Although this and all other current memory products from VIASYN are designed to run with processor speeds up to 12 MHz, there is a potential problem in using the phantom line at speeds exceeding 6 MHz due to its "open collector" signal type. This problem may exist particularly if all boards in a system adhere precisely to the IEEE 696/S-100 standard. The standard is the source of the potential problem because it specifies termination sourcing for speeds up to 6 MHz only. To operate at higher speeds, the phantom line needs a greater amount of pull-up than is specified in the bus standard to overcome its "RC" delay. If you do not absolutely need to use memory phantoming, don't activate it.

VIDEO DISPLAY INTERFACE MEMORY MAPPING

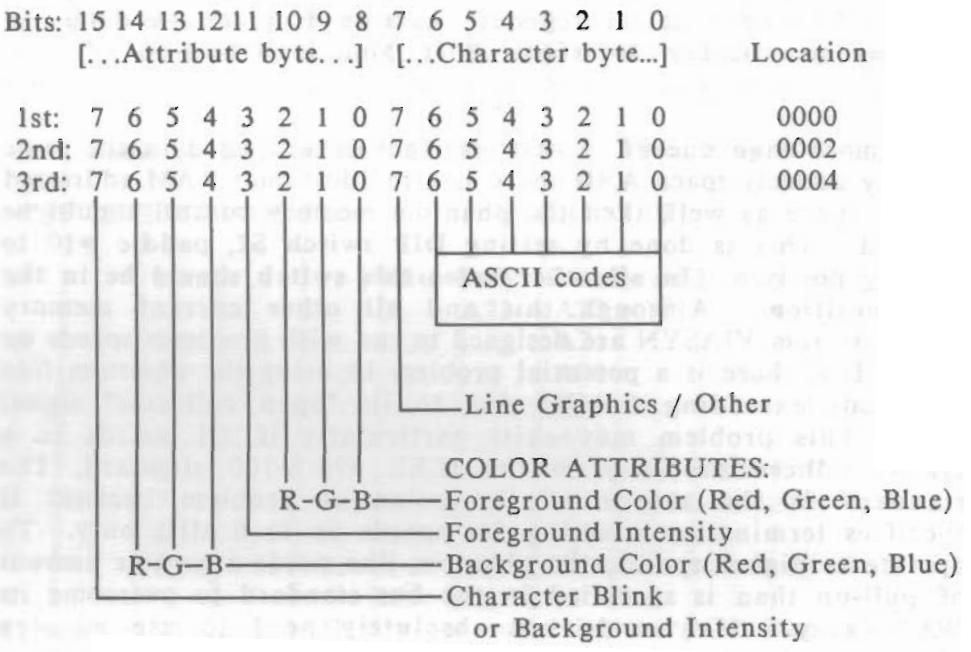
Alphanumeric Display Modes: (80w x 25h High Resolution)
(40w x 25h Low Resolution)

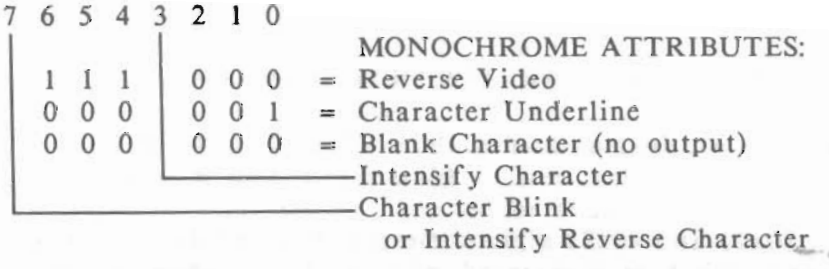
Both 80x25 and 40x25 alphanumeric display modes have two bytes in a word structure associated with each character displayed. The low order byte of each character word is the representation value for the character itself, which is usually its ASCII code value. The representation value addresses an area of PROM programmed with the correct bit representations for each scan line to display that character. If the PC Video interface is configured for color monitor output, the character generation PROM produces 7x7 dot characters with single line descenders, which are displayed within an 8x8 dot box. If configured for monochrome monitor output, then the character generation PROM produces 7x9 dot characters with 2 line descenders, which are displayed in a

9x14 box. The monochrome monitor does not support display in 40x25 resolution mode.

The high order byte of a character word contains the control attributes of that character, which differs between color and monochrome modes. For color displays, the attribute byte affects the foreground and background colors and the blink/intensity mode. For monochrome mode the attribute byte affects the reverse video, blink/intensity and underlining modes. The relative locations within the display memory array and internal structure of alphanumeric character words are listed in the following table:

Table 4: Alphanumeric Word Structure in Display Memory





Starting at the display memory base address, the first character is displayed in the upper left corner of the screen. Each sequential character word in memory is displayed to the right of the previous one,until the rightmost possible position of a screen line (column #80) is reached. The next character word then appears as the first character on the left of the next line downward. Mapping of the display memory to the screen positions has linear one to one position correspondence with no gaps. This means that the controller will display only the first 2000 (80x25) or 1000 (40x25) character word locations starting from the base address.

Additional pages of memory may be displayed by altering the "start address" programmed in the 6845 CRT controller. This programmed value determines the base address of the upper left corner character in the display memory. The start address does not affect the number of lines or columns displayed, only the locations in the display memory that will appear on screen. By changing this value in the CRT controller, it is possible to display up to 4 different 80x25 screens without altering the display memory array. Refer to the 6845 CRT controller chip data specifications in the "Theory of Operation" section of this manual for more detailed information.

Line Drawing and Other Character Graphic Displays

Halfway between alphanumeric and true graphics display modes is the realm of "line drawing" graphics. In both color and monochrome character PROMs, there are 40 alphanumeric representations which display combinations of various single and double line segments. Only horizontal and vertical line segments are available. These segments extend to the edges of the character box, producing large unbroken structures when combined on screen. The character generator PROM on the PC Video graphics board will display all of the representations shown in the following table:

Table 5: Display Character Representations

Lower Half of Character Generator PROM
(MONOCHROME Mode Display Representations)

Upper Half of Character Generator PROM
(COLOR Mode Display Representations)

High Resolution (640w x 200h) Graphics Display Mode

In true graphics modes, the display memory map is quite different from the alphanumeric modes explained above. Due to logic which routes the CRT controller scan addresses in a different manner from alphanumeric display modes, the memory map splits into 2 blocks of equal size. The graphic display logic will obtain data from alternating memory blocks depending on whether the output is appearing on an even or odd numbered line. The first scan line displayed, #0, and all other even numbered lines, come from the area of the block located at the true base or lower half of display memory. The second scan line displayed, #1, and all other odd numbered lines, are located in the block at the true base plus 8K bytes (2000 hexadecimal), which is the upper half of display memory.

Within each scan line, the rule for determining the memory locations that will be displayed is similar to that for alphanumeric modes. Both graphic modes display the same number of words as for 40 column alphanumeric mode on a line by line basis. The difference is that the actual data bits are displayed rather than a multiple scan line representation from a character generator. The display structure is byte wide, which means that first byte #0, then byte #1, is output from the same word.

Each discrete dot displayed in a graphics mode is called a "pixel," short for "picture element," and is also sometimes referred to as a "pel". For 640 pixel wide graphics, each bit in a byte is a dot, or pixel. These are displayed from the left-most screen dot as the most significant bit linearly to the right for the least significant bit of each byte. A graphics scan line

is completed when 40 words separated into 80 bytes have been displayed. The next scan line, odd numbered, is found at the same base address as the previous even numbered line plus 2000 hexadecimal. When the next even numbered scan line is displayed, its first memory location starts at the byte following the last one displayed on the previous even scan line. If that still seems confusing, a diagram showing the memory locations displayed from the top of the screen downwards is shown below:

Table 6: 640w and 320w x 200h Graphics Memory Map

Upper Left Corner	Upper Right Corner
Byte Locations Displayed	
0, 1, 2, 3, ...	77, 78, 79
8192,8193,8194,8195, ...	8269,8270,8271
80, 81, 82, 83, ...	157, 158, 159
8272,8273,8274,8275, ...	8349,8350,8351
. . . continues for a total of 200 scan lines.	

Medium Resolution (320w x 200h) Graphics Display Mode

For 320 pixel wide graphics mode, the pattern of bytes output is the same as for 640 wide, but uses 2 bits per pixel instead of one. Each byte displayed is made up of 4 pixels of 2 bits in the following pattern:

Table 7: 320w x 200h Graphics Mode Pixel Structure

Data Bit:	D7	D6	D5	D4	D3	D2	D1	D0
Pixel Bit:	b1	b0	b1	b0	b1	b0	b1	b0
Pixel Number:	[#0]		[#1]		[#2]		[#3]	

The pixels are displayed left to right, lowest to highest number on screen as they are shown in the above pattern. Because each pixel is comprised of 2 bits, it is possible to output any one of 4 colors for each display dot. This collection of colors that can be displayed by a graphics mode is called its "palette." The color which is generated by each bit pattern is determined as follows: each pixel bit is connected directly to one of two primary color controls. Pixel bit "b0" drives the color green, and pixel bit "b1" drives the color red.

Two possible color palettes are available. They are dependent upon the state of the third primary color control, blue. Its state is controlled by the palette select register (bit #5 of relative port #9) described in the I/O port map portion of this section. If the 320 pixel black/white control is enabled in the master display mode register (bit #2 of relative port #8), it will override the palette select control, giving a third possible palette, titled "black and white." If enabled, the 320 pixel black/white control gates the color blue to be identical to the color green, which follows the value of the "b0" pixel bit. Thus, if the green pixel bit is a one, or ON, then the color blue

is also on for that pixel. The following tables show how the color blue output, and resulting display color are determined for the three different master 320 wide graphics modes.

Table 8: 320w x 200h Graphics Mode Color Palettes

Palette #1: "Color Mode" with Palette Select = '0'

Port #8 B/W Mode	Port #9 Palette	b1	b0	Blue	Pixel Color
0	0	0	0	x	(Background)
0	0	0	1	0	Green
0	0	1	0	0	Red
0	0	1	1	0	Yellow

Palette #2: "Color Mode" with Palette Select = '1'

Port #8 B/W Mode	Port #9 Palette	b1	b0	Blue	Pixel Color
0	1	0	0	x	(Background)
0	1	0	1	1	Cyan
0	1	1	0	1	Magenta
0	1	1	1	1	White

Palette #3: "Black and White Mode"

Port #8 B/W Mode	Port #9 Palette	b1	b0	Blue	Pixel Color
1	x	0	0	x	(Background)
1	x	0	1	1	Cyan
1	x	1	0	0	Red
1	x	1	1	1	White

where "x" = any value

The rationale behind calling the third palette "black and white" will not be immediately obvious from its display on a color RGB monitor because it is not the intended output device for this mode. If the individual color control bits are each assigned an equal intensity value and they are summed together algebraically, there will be 4 possible intensity levels for each pixel. The "b1" pixel bit controls only the red output and has an assigned "weight" of '1', while the "b2" pixel bit controls both the green and blue outputs giving it an assigned weight of '2'. The various bit combinations will therefore generate an output having 4 different "gray levels." The various intensities effectively map a color graphics display onto a black and white only display monitor. Although there is no logic on the PC Video interface which directly generates this mutilevel intensity output, it may be obtained by using a few resistors and diodes connected to the RGB output signals.

The "background" color in Table 8 is defined by the outside border color controls (the lowest 4 bits of relative port register #9), and may be any one of 16 possible colors.

Low Resolution (160w x 100h) Graphics Display Mode

The lowest resolution graphics mode available (160x100) actually uses a hybrid display memory map combining aspects of both alphanumeric and true graphics modes. There are a total of 200 scan lines displayed in all color display modes which indicates that each pixel in low resolution graphics appears in 2 different scan lines of the monitor. This is effected in the CRT controller by altering the character height register from the normal 8 scan lines per character down to 2 scan lines. To obtain the correct number of scan lines for the total screen, the CRT registers are further modified to display 80 characters by 100 character lines instead of the normal 25. Another special case adjustment of alpha mode is used to divide each character into 2 horizontal pixels. The entire array of 8000 alpha mode character bytes is first initialized with the value 0DD hexadecimal, whose display representation is as follows:

Table 9: 160w x 100h Graphics Character Raster

Bits:	7 6 5 4 3 2 1 0	
Displayed	1 1 1 1 0 0 0 0	Scan line #1
PROM cells:	1 1 1 1 0 0 0 0	Scan line #2
	1 1 1 1 0 0 0 0	
	1 1 1 1 0 0 0 0	
	1 1 1 1 0 0 0 0	Scan lines #3 thru #7 are not
	1 1 1 1 0 0 0 0	displayed for low resolution
	1 1 1 1 0 0 0 0	160 x 100 graphics mode.
	1 1 1 1 0 0 0 0	
	1 1 1 1 0 0 0 0	Character Code = 0DDh

This fixed array of character bytes is translated by the PROM character generator to display 2 different pixel dots, each representing either the foreground or background colors of the corresponding attribute byte. Finally, the individual display dots are controlled by altering either the foreground (left pixel dot) or background (right pixel dot) color and intensity bits in the corresponding attribute byte. This corresponds to 4 bits per pixel. This mode is highly inefficeint because all even numbered bytes must have a fixed code value. This graphic mode can display 16 colors for either pixel (foreground/background nibbles) which permits full color graphics.

Available Display Colors

Throughout this manual, we mention that 16 colors can be displayed by various alphanumeric and graphic modes. To assist you in determining what colors are actually generated and in what manner, the following table shows the 4 video output control signals in all combinations and the associated colors that they produce.

Table 10: The 16 Possible PC Video Colors Generated

Intensity	Red	Green	Blue	Color Produced:
0	0	0	0	Black
0	0	0	1	Dark Blue
0	0	1	0	Dark Green
0	0	1	1	Dark Cyan
0	1	0	0	Dark Red
0	1	0	1	Dark Magenta
0	1	1	0	Brown
0	1	1	1	Light Gray
1	0	0	0	Dark Gray
1	0	0	1	Light Blue
1	0	1	0	Light Green
1	0	1	1	Light Cyan
1	1	0	0	Light Red
1	1	0	1	Light Magenta
1	1	1	0	Yellow
1	1	1	1	White

Observed Color Distortion and Display Monitor Adjustment

The true color appearing on most RGB monitors is dependent on several factors outside the control of the PC Video interface board. The most noticable is the change in color as the "intensity" control on the display is adjusted. You will see that yellow appears as gold, then brown as the intensity is decreased. The characterics of colors generated by pairs of the primary colors may also be different due to alternate phosphors used in monitor manufacture, or if the monitor's internal color intensity balance is incorrectly adjusted or aligned.

The second effect that will alter the observed color is well known to artists and others who often deal with color perception. It is possible to have any colored area perceived differently than the color that would be normally expected. This perception is dependent upon the visual area that a color occupies in relation to other adjacent area sizes and colors. This change in perceived color occurs because the human eye/brain nerve network performs a type of optical integration over the entire picture it receives. It then adjusts its perception of various colors such that the most reasonable picture results. What the eye/brain integrator believes to be reasonable is sometimes radically different from the "absolute" picture. An optical illusion can, for example, change the appearance of an interior color from green to red, when the entire surrounding visual region is altered in both color and intensity, even though the interior color remains physically the same!

VIDEO DISPLAY INTERFACE PORT ADDRESSING

The PC Video graphics display interface uses a block of 16 input/output port addresses for control and status communication with the host processor. These 16 port addresses are bank selectable and switch setable within certain range limits. This address range may be changed only by replacing the associated address select PAL chip. The base address of the first port is switch selectable in multiples of 16, and appears as the middle two nibbles of a 16-bit port address. All 16-bits of any I/O port address are always decoded, and the standard release PAL for this board is programmed with the upper nibble equal to zero. DIP switch S2, paddles 1 thru 8 are used to select the base address of the input/output ports on the PC Video display interface as shown in the following table:

Table 11: Input/Output Port Address Switch Settings

"S2" Switch Position	Address Bit	
	A15 = A14 = A13 = A12 = '0'	
1	A11	
2	A10	
3	A9	ON = '0'
4	A8	
5	A7	
6	A6	OFF = '1'
7	A5	
8	A4	

VIDEO DISPLAY INTERFACE PORT MAPPING

Although the PC Video board will respond to access at all 16 relative port addresses, only 9 of these perform any useful operation. These 9 ports will be referred to in this manual as relative ports 0 - 12 as shown below:

Table 12: Relative Port Address Functions

Relative Port:	Direction:	Function:
0 (00h)	Write	Bank select register -- always active
1 (01h)	Write	Address area select register
4 (04h)	Write	CRT controller address register
5 (05h)	Write	CRT controller data register
5 (05h)	Read	CRT controller data register
8 (08h)	Write	Master display mode register
9 (09h)	Write	Background color, palette select register
10 (0Ah)	Read	Video display, keyboard status buffer
11 (0Bh)	Write	Clear keyboard status, set clock register
12 (0Ch)	Write	Set keyboard / light pen status register
12 (0Ch)	Read	Keyboard input data register

NOTE: All of these I/O ports, except relative port 0, are also conditionally enabled by the on-board bank select register status. See the following bank select register discussion for more details.

Port #0 -- Bank Select Register (write only)

The purpose of this register is to permit as many as eight PC Video interface boards to occupy the same I/O port and memory address spaces. This capability is needed if all displays are to be running IBM-PC compatible software. By utilizing the bank select register, individual interface boards may be enabled at different times for operation in a multi-user environment. If the status of the bank select register is set to '0', the bus interface for all memory and I/O addresses, except the bank select register, is disabled. At reset, all PC Video boards have their bank select register status set to '1', making all of them appear at their selected addresses. The data bit used to control a particular board is selected by placing a shunt on the corresponding bit position of jumper block J3 as shown below:

Table 13: Bank Select Jumper Block, J3

Data Bit:	D0	D1	D2	D3	D4	D5	D6	D7
Bank Select Number:	[]	[]	[]	[]	[]	[]	[]	[]
Jumper Block J3	[]	[]	[]	[]	[]	[]	[]	[]

Port #1 -- Address Area Select Register (write only)

This register is connected to three of the address select decode bits via exclusive-or gates. It permits limited alteration of the established board address value under software control. The state of the address area select register is controlled by the least significant bit of the byte written to this I/O port. The switch settings affected by this function are memory address bit A15, (S1, paddle 8), and I/O port address bits A5 and A6 (S2, paddles 7 and 6). A system reset will clear this register to zero causing the affected address bits to follow normal switch settings. If the jumper at J2 is placed at B-C instead of A-C, then this register is disabled and PC Video board addressing will always match switch settings.

Table 14: I/O Port and Address Values for IBM Adaptors

Video Interface Type:	I/O Ports (Switch 2) *	Display Memory Address (Switch 1) *
		* = difference bits
IBM Color Adaptor	03Dx	0B8000 (hexadecimal)
IBM Monochrome Adaptor	03Bx	0B0000

The purpose of this register is to permit software testing at both of the IBM Color and Monochrome standard address locations without removing the PC Video interface from the system bus. As an example of how this works, set the board memory and I/O port addresses by switches at the standard values (shown in the preceeding table) for the IBM Color Adaptor video interface. If an output to relative port #1 contains a '1' in the least significant bit, then the address area select register will alter the board address select circuitry such that future accesses from the bus are possible only by using the Monochrome Adaptor locations. To return to the Color Adaptor address space, it is necessary only to output a '0' to relative port #1 in the Monochrome Adaptor address space.

Port #4 -- CRT Controller Address Register (write only)
and
Port #5 -- CRT Controller Data Registers (read and write)

There are two ports that the 6845 CRT controller chip responds to on the PC Video interface board. The one at relative port #4 is write only and is used to select one of 18 internal data registers of the chip. Relative port #5 is both write for register control set-up and read for status sense operations. The following table gives the functions of each of the 6845 CRT controller's internal registers.

Table 15: 6845 CRT Controller Register Functions

Register Number:	Data Direction:	Function:	Units:
0 (00h)	Write	Horizontal total	Character
1 (01h)	Write	Horizontal displayed	Character
2 (02h)	Write	Horizontal sync position	Character
3 (03h)	Write	Horizontal sync width	Character
4 (04h)	Write	Vertical total	Char. Row
5 (05h)	Write	Vertical total adjust	Scan Line
6 (06h)	Write	Vertical displayed	Char. Row
7 (07h)	Write	Vertical sync position	Char. Row
8 (08h)	Write	Interlace mode	
9 (09h)	Write	Max scan line address	Scan Line
10 (0Ah)	Write	Cursor start	Scan Line
11 (0Bh)	Write	Cursor end	Scan Line
12 (0Ch)	Write	Display start address	High Byte
13 (0Dh)	Write	Display start address	Low Byte
14 (0Eh)	Read/Write	Cursor address	High Byte
15 (0Fh)	Read/Write	Cursor address	Low Byte
16 (10h)	Read	Light pen address	High Byte
17 (11h)	Read	Light pen address	Low Byte

Initialization values for these registers for the common display modes are given in the Software Section of this manual. Further details on the function of individual registers within the 6845

CRT controller chip, along with the reprinted data sheets, are given in the "Theory of Operation" section of this manual.

Port #8 -- Master Display Mode Register (write only)

This 6-bit register controls the primary display modes of the PC Video interface board. These signals determine the master timing for display dots and select their signal routing to produce the various possible graphic and alphanumeric display types. The names and bit positions within this mode register are shown in the following table.

Table 16: Relative I/O Port #8 Control Bit Functions

Data Bit:	Function:
7	None
6	None
5	Enable character blink/Foreground intensity attribute
4	Special mode bit enables 640 X 200 graphics display
3	Video display output enable
2	Black/White display type for 320 X 200 color graphics
1	Graphics/Alphanumeric display mode select
0	High/Low resolution master timing mode select

The Master Display Mode control values for both color and monochrome monitors are listed in the following table:

Table 17: Useful Master Display Mode Control Values

Color Display Monitor:

Bits: 5 4 3 2 1 0	Function:
1 0 1 x 0 0	40 x 25 Alphanumeric
1 0 1 x 0 1	80 x 25 Alphanumeric
x 0 1 1 1 0	320 x 200 Graphics, B/W
x 0 1 0 1 0	320 x 200 Graphics, color
x 1 1 1 1 0	640 x 200 Graphics, B/W

Monochrome Display Monitor:

Bits: 5 4 3 2 1 0	Function:
1 0 1 x 0 1	80 x 25 Alphanumeric
x 0 1 1 1 0	320 x 200 Graphics, B/W
x 1 1 1 1 0	640 x 200 Graphics, B/W

"x" = don't care

To produce a correct raster display, both the master display mode register and the 6845 CRT controller registers #0 thru #15 must be initialized in a consistent manner. This is usually done by sequentially loading all of these registers in the same routine

to minimize the time period that the display monitor is subjected to abnormal sync pulse timing. Default initialization sequences showing these register values are covered, along with an initialization routine, in the Software Section of this manual.

Port #9 -- Background Color and Palette Select Register (write only)

This 6-bit control register is only functional when using color display monitors. It selects the master color display characteristics by selecting the color displayed in the outside border region area of the monitor for all but 640 x 200 graphics mode. It selects the foreground intensity level, and also selects one of two palettes available for medium resolution (320 x 200) graphic display mode as indicated in the following table:

Table 18: Relative I/O Port #9 Control Bit Functions

Data Bit:	Function:	
7	None	
6	None	
5	Palette select for color 320 x 200 graphics display	
4	High/Low intensity for foreground graphics colors	
3	High/Low intensity	\\
2	Red color enable	\\ Outside border area
1	Green color enable	// color control bits
0	Blue color enable	//

Port #10 -- Video Display and Keyboard Status Buffer (read only)

This 4-bit master status port returns only those specific hardware signals that are not available from the CRT controller. Two of the bits concern the keyboard and light pen data input status and may be polled as part of a keyboard data input routine. The other two bits contain the state of the display enabled and vertical sync signals, which may be used to synchronize display memory alterations with screen blanking periods.

Table 19: Relative I/O Port #10 Status Bit Functions

Data Bit:	Function:
3	Vertical Sync status (raster active indicator)
2	Light Pen switch activated status (not latched)
1	Keyboard/Light Pen data available status (latched)
0	Monitor Display Enabled (display not blanked indicator)

Port #11 -- Clear Keyboard Status and Set Clock Register (write only)

This control port acts to reset both the keyboard data port and its data available status by writilg any value to it. It also has a single bit register controlled by the value of the most significant bit of the data byte sent to it. The register controls an open collector gate attached to the keyboard clock signal which, in nopmal operation, synchronizes data sent from an IBM compatible keyboard. By precisely controlling the ON/OFF period of vhis register, it enables the clock signal to be used in the reverse direction as a simple asynchronous data communications line. This is needed by a few specialized keyboards which are able to download data from the computer to perform additional functions such as speech recognition.

Port #12 -- Set Keyboard/Light Pen Status Register (write only)

By writing any data value to this port, the keyboard/light pen data available status bit is set, indicating that data is ready. It really has only one minor function, which is to hold off the light pen from making any additional false interrupts while the current light pen signal is being processed.

Port #12 -- Keyboard Input Data Register (read only)

Reading this port sets the data available status register, but also returns data sent by the keyboard or light pen. Both devices may be connected at the same time to this data and status/interrupt circuit, so either may be the source of the data available flag being set. This is resolved by reading the keyboard data returned and testing for all zeros. The keyboard will never generate a zero data value, but a zero is forced into the data register whenever the keyboard clear port is written. Note that all keyboard input data register read operations should be followed by a write to the clear keyboard status port. This will enable another keyboard or light pen signal to be received by the PC Video board.

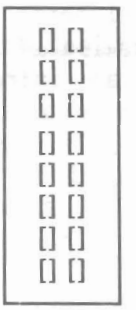
INTERRUPTS

The PC Video interface is capable of operating in either a polled status or interrupt driven mode for data input. Operation in the interrupt driven mode is particularly suited for multi-user or multi-tasking environments, such as with Concurrent DOS 8/16. When functioning in interrupt mode, the output of the light pen/keyboard status register is connected to drive one of the vectored interrupt lines (VI0* thru VI7*) on the IEEE 696/S-100 bus. This is accomplished by inserting the correct shunt on jumper block J4, positions 0-7, corresponding to the desired interrupt vector number. The interrupt vector position numbers are identified on interrupt jumper block, J4, as follows:

Table 20: Vector Interrupt Jumper Block, J4

Interrupt Jumper Block J4:

CompuPro standard



VI7
VI6
VI5
VI4
VI3
VI2
VI1
VI0

If correctly jumpered, an interrupt will be generated when the data ready status is set from either source of input: keyboard or light pen. **All CompuPro software, which operates interrupt driven from this board, expects that vectored interrupt #5 has been shunted.**

SOFTWARE SECTION - CODE EXAMPLES

This section discusses the control setup and data input/output sequences needed for correct operation of the PC Video interface board. It is intended only as an introduction to producing functional software. Every operational detail of the PC Video interface software is not covered. The 8086 source code examples presented are derived from actual driver routines that perform these functions within related CP/M operating systems from VIASYN. They are expected to execute correctly with only a few code additions necessary to fill out a completed driver. For example, code and data segment areas in the sample routines have no references to any base or origin because they are not complete programs. To produce runable code, it is necessary to create a main program, which then calls the sample routines presented in this section to perform the indicated functions.

More specific details of software operation may be obtained by listing a TMXBIOS (CP/M-86 or CP/M 8-16 BIOS source code) that has been assembled with the PCVIDEO variable equated "true" in the file ACTIVE.EQU. Please read the associated CP/M-86 or CP/M 8-16 documentation for the information necessary to assemble and use TMXBIOS.

COMMON EQUATES AND DATA STORAGE VARIABLES

Common Equate statements start the sample 8086 code and are used by the routines which follow in the rest of the Software Section of this manual. Included are base specifications for both Color and Monochrome configurations that match those of the IBM-PC display adaptors. Also included are mnemonics for the PC Video relative input/output port offsets and the four input status port bits. The remainder of the equates deal with other types of hardware base specifications and ASCII control characters.

The Common Data Storage Area deals with those variables which are utilized by more than one type of routine. This starts with PCVBASE, which holds the current base input/output port address. By making all ports referenced by input/output instructions relative to a storage variable, all of the routines apply equally to both Monochrome and Color configurations of the PC Video board. Similarly, PCVWIDE as a variable permits both 40 and 80 column wide alphanumeric displays as well as graphics modes to be selected without altering or directing the flow of various routines.

```
;*************************************************
;*                                               *
;*   PC VIDEO GRAPHICS BOARD EQUATES             *
;*                                               *
;*************************************************

;
;     COLOR Mode Equates:
;
PCVCSEG    EQU   0B800h  ;Memory segment base
PCVCOLP    EQU   03D0h   ;Base port address
PCVCCUR    EQU   0607h   ;Cursor raster limits   (underline)
;          0007h                                 (block cursor)
;
;     MONOCHROME Mode Equates:
;
PCVMSEG    EQU   0B000h  ;Memory segment base
PCVMONP    EQU   03B0h   ;Base port address
PCVMCUR    EQU   0B0Ch   ;Cursor raster lines    (underline)
;          000Ch                                 (block cursor)
;
;     PC Video Relative I/O Port Offset Equates:
;
PCVBANK    EQU   00h     ;Bank select register output port
PCVFLIP    EQU   01h     ;Base address switch flip reg. output port
PCV6845A   EQU   04h     ;Register address port of 6845 CRT Controller
PCV6845D   EQU   05h     ;Register data input/output port of 6845 CRTC
PCVMASTR   EQU   08h     ;Master Display Mode latch output port
PCVBGRND   EQU   09h     ;Background Color latch output port
PCVSTATUS  EQU   0Ah     ;Status Buffer input port
PCVLPCLR   EQU   0Bh     ;Clear Light Pen / Keyboard status/data regs.
                         ; and Keyboard Clock Control reg. output port
PCVLPSET   EQU   0Ch     ;Set Ready Flag / Keyboard Data input port
;
; Input Status Port (PCVSTATUS) bit definitions.
;
VRT        EQU   1000b   ;Vertical retrace bit
LPEN_SW    EQU   0100b   ;Light Pen Switch activated bit
LPEN_RDY   EQU   0010b   ;Light Pen / Keyboard data byte ready bit
HRT        EQU   0001b   ;Horizontal retrace (display enable) bit
;
;     Priority Interrupt Controller Equates:
;
MASTER_PIC_0  EQU  50h   ;System support I master PIC port #0
MASTER_PIC_1  EQU  51h   ;System support I master PIC port #1
SEOI       EQU   01100000b ;Specific End Of Interrupt
;
;     ASCII Control Character Equates:
;
ETX        EQU   03h     ;Control-C for system restart
X_ON       EQU   11h     ;XON /DC1 control code
X_OFF      EQU   13h     ;XOFF/DC3 control code
DEL        EQU   7Fh     ;Delete
```

```
;*****************************************
;*                                       *
;* COMMON DATA STORAGE LOCATIONS         *
;*                                       *
;*****************************************

;
    DSEG
;
PCVBASE DW   PCVCOLP ; PCVMONP ;Color/Monochrome base I/O port addr
                               ; (Dynamically alterable to either)
PCVWIDE DB     80            ; 40  ;Horiz columns (40 for graphics)
```

KEYBOARD/LIGHT PEN STATUS AND DATA INPUT ROUTINES

This section contains four routines for Keyboard or Light Pen data available status and data input. The sample routines have been only minimally condensed from their originals so that the full set of potential software/hardware interactions is apparent. The only inline instruction code not completely present is the scan code and ASCII character input buffering for keyboard interrupts. This is so dependent on operating system parameters and handshaking that it would have greatly enlarged the coded examples were it fully incorporated. Instead, a set of comments clearly indicates the location of this buffering code and the functions that must be performed.

The only other condensation that affects the execution of the routines in this section involves the ASCII code translation tables. These are excessively large tables near the end of the data storage area for these routines, and have been reduced to simple "Reserve Storage" statements to simplify the examples. Each table storage definition indicates the nature of the original translation byte codes. These need to be correctly filled out if the code is to function in a real system.

Keyboard/Light Pen Initialization Routine

There are two parts to this initialization sequence; first the PC Video board keyboard data/status is cleared, which is followed by enabling the associated interrupt at Priority Interrupt Controller (PIC). The keyboard/light pen status and data input registers need flushing to permit the keyboard input logic to function correctly at the first keystroke. Master PIC initialization for vector interrupt #5 is optional, but if present should come after the keyboard data/status flush sequence to prevent any spurious scan code data for the first keyboard character input.

KEYBOARD/LIGHT PEN DATA INPUT INTERRUPT ROUTINE

Keyboard or light pen data inputs can interrupt the host CPU if enabled and physically connected on the PC Video board via a jumper at VI5. On interrupt, this routine saves the CPU register state and switches the data segment context to local storage.

A consequence of the non-interrupt data input routine is that the interrupt line at VI5 is released when the keyboard data/status registers are cleared. The final portion of the interrupt handling routine resets the Master PIC so it can correctly receive another VI5 interrupt and restores the CPU registers to their entering states. The exit instruction, IRET, performs a far return to the interrupted code and restores the entering flag registers, including interrupt enable.

Keyboard/Light Pen Data Available Status Routine

Console Data Input Available status determination is performed by this routine for the keyboard and light pen inputs. The result of sampling the PC Video status input port is returned directly in the high byte of the accumulator. It is masked by the keyboard/light pen data available bit in the low byte of the accumulator, which returns a non-zero value with the Zero Flag reset if the status bit indicates input data is ready at the keyboard/light pen data byte holding register.

Keyboard Coded Data/Light Pen Position Input Routines

Console Character Data Input (or a logically equivalent function) is performed in this routine for both the keyboard and light pen data inputs. The data available status common to both input sources is polled until it indicates a ready condition. This permits the keyboard scan code register to be read and sampled to determine the input data source. Two different data input functions are performed in the remainder of this routine, depending on the input source given by the keyboard scan code value. A zero scan code value indicates the light pen triggered the input, and causes the data input handler to execute. Any other (non-zero) scan code value must be the result of a keyboard character input, and causes the keyboard input function to execute.

Light Pen Position Data and Status Input Routine

A light pen trigger input causes the CRT controller (CRTC) to latch its present character address internally, which can be read to determine the display screen position of the light pen. The raw position is obtained by reading registers #16 and #17 of the CRTC as high and low bytes of the character address. This is adjusted to compensate for input latching delays, and divided by the current column count to give an "X,Y" set of coordinates for the light pen position.

The Light Pen Switch input indicates whether the current position coordinates are flagged by the operator for special significance (determined by the application software package). Raw status is input again and saved followed by clearing of the scan code, light pen and input data available status registers. These registers are not cleared earlier because they prevent a second light pen input from triggering a change in the CRTC during the data input of the original position. Finally, the raw input status is masked for the light pen switch bit, and the accumulator and zero flag are set according to the indicated switch status.

Keyboard Scan Code Data Input and Translation Routine

The scan code is first duplicated into the high byte of the accumulator, where it will be returned from this routine. Because the scan code translation routine has several different exit points, data input is re-enabled early by clearing both the keyboard scan code and input data available status registers. The flushed registers permit another data input to be accepted into either the keyboard or light pen holding registers.

Scan codes are stripped of the 8th "release" indicator bit and checked against the upper limit of valid scan code numbers. If they are invalid, the low byte of the accumulator is set to 0FFh and further processing is aborted.

Next, the seven scan codes for "mode" keys are checked against the current input. If any of these mode key scan codes match, then the associated "depress" and "release" control bits for that key are obtained from the associated tables. The correct table's control byte is selected by sampling the "release" bit of the current scan code. The control mode is then updated by exclusive-or of that control byte with the stored control mode byte. If the scroll lock mode key is the current scan code, it is tested for "depress" status and returns the XON/XOFF control codes to toggle output scrolling. All other mode key conditions result in a "non-action" keystroke (zero) set in the low byte of the accumulator.

ASCII translation of the scan code will occur if none of the above mode key scan codes match. The scan code is first tested for "depress" status. If it is not active, the "non-action" result code is placed in the low accumulator byte, and the routine is exited. Initial translation table base selection for all scan codes is determined by the status of the control and altmode keys, as indicated by the control mode storage byte. If either mode is active, the associated table base is selected with no secondary considerations.

Secondary selection of the translation table base is made differently for each of the following three groups of scan codes:

Codes 1 thru 58 are the normal alphanumeric keys; 59 thru 68 indicate the function keypad; and 69 thru 83 belong to the numeric/arrow keypad. Shift mode status is the basis for this secondary selection level, determined by the left and right shift keys and caps lock for alphanumeric scan codes or number lock for the numeric/arrow keypad. The status bit combinations of these mode keys become word offsets in mode tables in order to select the correct table base pointer.

In either translation table selection case, the resulting ASCII character code placed in the accumulator low byte is translated via the scan code offset into the selected table base. The high byte of the accumulator will always be the original scan code input from the PC Video board.

```
;*************************************************************
;*                                                           *
;*KEYBOARD / LIGHT PEN DATA INPUT ROUTINES*
;*                                                           *
;*************************************************************


LPEN_DELAY  EQU  2      ;Light pen response position latch is delayed
                        ;by constant offset relative to CRTC position
;
   CSEG      ;Code Segment Area
;
;=======================================
;
;     Keyboard / Light Pen  Initialization Routine:
;
KBDINIT:MOV     DX,PCVBASE          ;Get base of PC Video I/O ports
        ADD     DX,PCVLPCLR         ;Add clear regs./set clock port offset
        XOR     AL,AL               ;Clear keyboard clock control bit (D7)
        OUT     DX,AL               ;Clear status of ready flag for input,
                                    ; enable keyboard clock
        MOV     AX,DS               ;Get current data segment
        MOV  CS:LOCAL_DSEG,AX       ;Save for interrupt routine
        PUSHF ! CLI                 ;Save flags, disable interrupts
        IN   AL,MASTER_PIC_1        ;Get current PIC interrupt enable bits
        AND     AL,11011111b        ;Add vector interrupt #5 enable
        OUT  MASTER_PIC_1,AL        ;Update PIC interrupt enable register
        PUSH    AX !   POP AX       ;Delay fast CPUs (too fast for PIC)
        MOV     AL,SEOI+5           ;Send VI5 specific end of interrupt
        OUT  MASTER_PIC_0,AL        ; to PIC in case input latch was set
        POPF                        ;Recover flags, re-enable interrupts
        RET
;

;=======================================
;
;     Keyboard / Light Pen Input Interrupt Routine:
;
LOCAL_DSEG      DW      0           ;Storage for local data segment value
```

```
;
PCVINTR:PUSH AX ! PUSH BX ! PUSH CX ! PUSH DX
        PUSH SI !  PUSH DS              ;Save incoming registers
        MOV  AX,CS:LOCAL_DSEG           ;Get value of local data segment
        MOV     DS,AX                   ;set up as current
        CALL    VI5DATA                 ;Get keyboard/light pen data input
        OR      AH,AH                   ;See if light pen input
        JNZ     VI5_KEY                 ;Buffer Key scan code/ASCII if not
        MOV  LPEN_POSITION,BX           ;Update current light pen position
        MOV  LPEN_SWITCH,AL             ;and input switch status
        JMPS    VI5SEOI                 ;Finish interrupt for VI5
;
VI5_KEY:CMP     AL,0FFh                 ;See if invalid key input
        JE      VI5SEOI                 ;Just finish interrupt if so
        OR      AL,AL                   ;See if non-action keystroke
        JE      VI5SEOI                 ;Just finish interrupt if so
;
;  ##########################
;
; Put Keyboard Scan Code / ASCII Code word pair "AH,AL" in
; console input buffer and bump the associated buffer load pointer.
; This code is very dependent on the buffering scheme chosen.
;
;  ##########################
;
VI5SEOI:MOV     AL,SEOI+5               ;Send VI5 specific end of interrupt
        OUT  MASTER_PIC_0,AL            ;to PIC to clear interrupt for next
        POP DS  ! POP SI                ;Recover incoming registers
        POP DX ! POP CX ! POP BX ! POP AX
        IRET                            ;Return from interrupt (restore flags)
;
;=======================================
;
;       Keyboard / Light Pen  Data Available Status Routine:
;
;Exit:  AH =    Status byte from PC Video board.
;       AL =    0 and Z-Flag set if input data byte not available.
;
PCVISTAT:MOV    DX,PCVBASE      ;Get base address of PC Video ports
         ADD    DX,PCVSTATUS    ;Add status input port relative offset
         IN     AL,DX           ;Input keyboard / light pen status
         MOV    AH,AL           ;Save status value in "AH"
         AND    AL,LPEN_RDY     ;Mask to get data available bit
         RET                    ;Return zero flag set by input status
;
;=======================================
;
;     Keyboard Data Byte / Light Pen Position Input Routine:
;
;Exit:  AH =    Keyboard Scan Code (zero if Light Pen input).
;   if Keyboard Data input:          ( AH = non-zero scan code )
;       AL =    Keyboard ASCII code if valid key depression.
;       AL =    0 if keyboard non-action key depressed.
;       AL =    0FFh if invalid keyboard scan code value.
;   if Light Pen data input:         ( AH = zero )
```

```
;       BX =    Light Pen position ( BH = "Y" row, BL = "X" column ).
;       AL =    0/4, Z-Flag set according to Light Pen Switch status.
;
PCVIDATA:CALL   PCVISTAT        ;Get status of PC Video data available
        JZ      PCVIDATA        ;Loop until input data byte ready
VI5DATA:MOV     DX,PCVBASE      ;Get base address of PC Video ports
        ADD     DX,PCVLPSET     ;Add keyboard data input port offset
        IN      AL,DX           ;Input serial data byte from keyboard
        OR      AL,AL           ;See if zero value for light pen input
        JNZ     KEYDATA         ;Perform keyboard data input if not
;
;---------------------------------------
;
LPENDATA:MOV    DX,PCVBASE      ;Get base address of PC Video ports
        ADD     DX,PCV6845A     ;Add 6845 CRTC register address offset
        MOV     AL,16           ;Reg #16 = light pen position hi byte
        OUT     DX,AL           ;Set up register number in CRTC
        INC     DX              ;Point to 6845 CRTC data port PCV6845D
        IN    AL,DX ! MOV AH,AL ;Light pen position high byte in "AH"
        DEC     DX              ;Point back to CRTC reg address port
        MOV     AL,17           ;Reg #17 = light pen position low byte
        OUT     DX,AL           ;Set up register number in CRTC
        INC     DX              ;Point to 6845 CRTC data port again
        IN      AL,DX           ;Light pen position low byte in "AL"
        SUB   AX,LPEN_DELAY     ;Remove light pen response delay bias
        MOV     BL,PCVWIDE      ;Column count for active display mode
        DIV  BL ! MOV  BX,AX    ;Convert position to "Y,X" in "BH,BL"
        CALL    PCVISTAT        ;Get status of PC Video data available
        XCHG    AH,AL           ;Put status byte in "AL", clear "AH"
        PUSH    AX              ;Save status, data bytes
        MOV     DX,PCVBASE      ;Get base address of PC Video ports
        ADD     DX,PCVLPCLR     ;Add clear regs./set clock port offset
        XOR     AL,AL           ;Clear keyboard clock control bit (D7)
        OUT     DX,AL           ;Clear status for next, enable kbd clk
        POP     AX              ;Recover status, data bytes
        NOT     AL              ;Invert status bits for neg signal
        AND     AL,LPEN_SW      ;Mask light pen switch LPEN SW* input
        RET                     ;Return zero flag accordingly
;
;---------------------------------------
;
KEYDATA:MOV     AH,AL           ;Save keyboard scan code byte in "AH"
        PUSH    AX              ;Save scan code bytes
        MOV     DX,PCVBASE      ;Get base address of PC Video ports
        ADD     DX,PCVLPCLR     ;Add clear regs./set clock port offset
        XOR     AL,AL           ;Clear keyboard clock control bit (D7)
        OUT     DX,AL           ;Clear status for next, enable kbd clk
        POP     AX              ;Recover scan code bytes
        AND     AL,7Fh          ;Mask to remove key release bit
        CMP     AL,83           ;Test scan code for valid key number
        JLE     KEYTEST         ;Continue with mode key test if valid
        MOV     AL,0FFh         ;Flag "AL" with invalid scan code ID
        RET
;
```

```
KEYTEST:MOV  SI,offset MODE_KEYS ;Point to table of mode key codes
        MOV    CX,NMODES         ;Number of mode key scan codes to test
        CLD                      ;Clear dir flag for forward search
        REPNE  SCASB             ;Search for matching table scan code
        JNE    KEY1XLT           ;Translate to ASCII if not found
        MOV  BL, 1 * NMODES[SI]  ;Key depress update bit code in "BL"
        MOV  AL, 2 * NMODES[SI]  ;Key release update bit code in "AL"
        TEST   AH,80h            ;See if key depression or release
        JNZ    FIXMODE           ;Use release bit code if 8th bit set
        MOV    AL,BL             ;Key depression update code in "AL"
FIXMODE:XOR  KEY_MODE_BITS,AL ;Flip correct mode bit for this key
        JCXZ   SCROLOK           ;Special scrl lock output if that key
NON_ACTION_KEY:
        XOR    AL,AL             ;Clear "AL" to indicate non-action key
        RET
;
SCROLOK:OR     AL,AL             ;See if scroll lock key depression
        JZ     SCROLMD           ;Do non-action mode output if not
        TEST KEY_MODE_BITS,AL ;Test if scroll mode bit set this pass
        MOV    AL,X_OFF          ;XOFF/DC3 code to disable scrolling
        JNZ    SCROLMD           ;if scroll mode bit set
        MOV    AL,X_ON           ;XON /DC1 code to re-enable scrolling
SCROLMD:RET
;
;.....
;
KEY1XLT:TEST   AH,80h            ;See if key depression or release
        JNZ    NON_ACTION_KEY;Non-action keystroke if release
        MOV  BL,KEY_MODE_BITS ;Get current mode control bits
        AND    BX,06h            ;Mask cntl and alt bits (x2 for word)
        CMP    BX,06h            ;See if both bits active
        JNE    KEY2XLT           ;Continue ASCII translation if not
        CMP    AL,DEL            ;See if ctrl/alt/del (system reset)
        MOV    AL,ETX            ;Use a ctrl-C if so
        JNE    NON_ACTION_KEY;Non-action keystroke if not
        RET
;

KEY2XLT:MOV  BX,KEYTABLE[BX]   ;Get correct translate table base ptr
        OR     BX,BX             ;See if either cntl or alt bit set
        JNZ    XLATE_KEY         ;Translate in selected table if so
        MOV  BL,KEY_MODE_BITS ;Current mode control bits in "BL"
        CMP    AL,58             ;See if Alphanumeric key scan code
        JLE    XLATE_ALPHA       ;Do its translation if so
        CMP    AL,68             ;See if Function key scan code
        JLE    XLATE_FXN         ;Translate if so
                                 ;Key is in Arrow/Numeric keypad
        AND    BX,70h            ;Mask NumLock, R./Shift,, L./Shift bits
        MOV  CL,3  ! SHR  BX,CL  ;Shift right 3 bits to get word offset
        MOV  BX,ARROW_TABLE[BX];Get correct translate table base ptr
        JMPS   XLATE_KEY         ;Translate using table base in "BX"
;
XLATE_ALPHA:                     ;Alphanumeric Key (scan code 1-58)
        AND    BX,38h            ;Mask R./Shift, L./Shift,, Caps Lock bits
        SHR  BX,1  ! SHR  BX,1   ;Shift right 2 bits to get word offset
```

```
        MOV  BX,SHIFT_TABLE[BX] ;Get correct translate table base ptr
;
XLATE_FXN:                      ;Function keys do not use shift mode
XLATE_KEY:                      ;Xlate table base "BX", scan code "AL"
        XLAT   BX               ;ASCII code for this key in "AL"
        RET                     ;Non-zero scan code "AH", ASCII "AL"


    ************************************************************
    *                                                          *
    *          KEYBOARD / LIGHT PEN LOCAL                      *
    *              DATA STORAGE AREA                           *
    *                                                          *
    ************************************************************


   DSEG         ;Data Segment Area
;
LPEN_POSITION   RW    0     ;Light Pen position coordinate storage
LPEN_X          DB    0     ;"X" column coordinate
LPEN_Y          DB    0     ;"Y" row coordinate
LPEN_SWITCH     DB    0     ;Non-zero if light pen switch active
;
;Mode: Scrl/Lk Num/Lk  R/Shft  L/Shft  Caps/Lk  Alt    Ctrl    ---
;  Bit:    7       6       5       4       3      2       1      0
;
KEY_MODE_BITS   DB      00000000b       ;Mode control bit storage
;
;                    Ctrl  L/Shft R/Shft Alt   Caps  Num   Scrl
MODE_KEYS       DB   29,   42,    54,    56,   58,   69,   70
depress         DB   02h,  10h,   20h,   04h,  08h,  40h,  80h
release         DB   02h,  10h,   20h,   04h,  00h,  00h,  00h
;
NMODES  EQU  length MODE_KEYS   ;Number of mode keys in table
;
KEYTABLE:                       ;Alt   Cntl
        DW  offset NORMTBL      ; 0     0
        DW  offset CTRLTBL      ; 0     1
        DW  offset ALT_TBL      ; 1     0
        DW  offset CTRLTBL      ; 1     1
;
SHIFT_TABLE:                    ;R/Shft L/Shft  Caps Lock
        DW  offset NORMTBL      ; 0      0        0
        DW  offset SHFTTBL      ; 0      0        1
        DW  offset SHFTTBL      ; 0      1        0
        DW  offset NORMTBL      ; 0      1        1
        DW  offset SHFTTBL      ; 1      0        0
        DW  offset NORMTBL      ; 1      0        1
        DW  offset SHFTTBL      ; 1      1        0
        DW  offset NORMTBL      ; 1      1        1
;
ARROW_TABLE:                    ;Num    R/Shft  L/Shft
        DW  offset NORMTBL      ; 0      0       0
        DW  offset SHFTTBL      ; 0      0       1
        DW  offset SHFTTBL      ; 0      1       0
```

```
        DW  offset SHFTTBL      ; 0     1       1
        DW  offset SHFTTBL      ; 1     0       0
        DW  offset NORMTBL      ; 1     0       1
        DW  offset NORMTBL      ; 1     1       0
        DW  offset NORMTBL      ; 1     1       1
;
NORMTBL RB      58      ;Translation table from scan code to ASCII
        RB      10      ;10 Function Keys
        RB      15      ;15 Arrow / Numeric Keys
SHFTTBL RB      83      ;Upper case shift table
CTRLTBL RB      83      ;Control key table
ALT_TBL RB      83      ;Altmode key table
```

VIDEO DISPLAY DATA OUTPUT ROUTINES

Video Display Timing chain initialization is demonstrated by the PCVINIT routine, which initiates both a Color and a Monochrome display interface. (Actual useful operation requires two separate PC Video boards addressed at different memory and I/O port bases.) Video output initialization involves filling the entire display memory with a word value, followed by setting up all of the Mode control I/O port registers and then 16 of the CRT controller's internal registers.

The sample routine presented also updates storage locations used by the alphanumeric mode screen drivers to maintain synchronization. Both timing chains are initialized by the same routines. At the end of the sequence, the main video driver reference addresses and parameters (not included here) have all been switched to match the new type of display.

An attempt is made by the example routines to synchronize timing chain alterations with the vertical sync pulse by locking on to this signal to turn the video output on and off (BEAMON, BEAMOFF, and VSYNC subroutines). This is attempted because a randomly timed re-initialization of the PC Video's 6845 CRTC will produce an erratic amount of separation between vertical sync pulse phases. This in turn causes the video display output to "roll" for one or two screens. This is visually unappealing, which leads to an attempt to correct the situation. Unfortunately, there are data combinations which can be programmed into the CRTC, causing it to not produce any vertical syncs. Consequently, these synchronization routines have timeouts built in to prevent a lockout condition.

There are nine initialization sequence strings at the end of the data storage area, one for each of the various common Color and Monochrome display modes. All of the values in these strings were chosen to enable the PC Video board to correctly drive the associated IBM-PC display monitor.

Unfortunately, inclusion here of video display drivers demonstrating both alphanumeric and graphic modes (or even a sophisticated version of just one of them) is well beyond the space permitted. The video display output driver within the TMXBIOS code for CP/M-86 is quite sophisticated in that it emulates DEC VT-52, Qume QVT-102 and Lear-Siegler ADM-3A terminals at the same time. In addition, the driver contains many of the control sequences used for graphics mode control in conjunction with Digital Research GSX (Graphics System Extension) GDOS and GIOS interfaces. These characteristics probably make it the best reference for writing or modifying PC Video output driver code. The video display driver within TMXBIOS is primarily contained in the file TMXPCV.DVR, with common equates found in the file PCVIDEO.EQU.

```
;*******************************************************
;*                                                     *
;*   PC VIDEO DISPLAY DATA OUTPUT ROUTINES             *
;*                                                     *
;*******************************************************


;       Common Equates:
;
BLANK     EQU   0720h   ;Initial Space character/attribute
PCVHIGH   EQU   25      ;CRT screen height in rows
;
;=======================================
;
;       PC Video Display Output Initialization Routine:
;
PCVINIT:MOV   SI,offset COLOR_ALPHA_80     ;Point to color init string
        CALL  INICOLOR          ;Perform color init sequence
        MOV   SI,offset MONO_ALPHA_BW_80 ;Point to monochrome string
;       jmps  inimono           ;Finish monochrome init sequence
;
;------------------------------
;
INIMONO:MOV   AX,PCVMSEG    ;Monochrome base segment address
        MOV   DX,PCVMONP    ;Monochrome base port on PC Video
        JMPS  PCVINI1       ;Complete initialization sequence
;
INICOLOR:MOV  AX,PCVCSEG    ;Color base segment address
        MOV   DX,PCVCOLP    ;Color base port on PC Video
;       jmps  pcvini1       ;Complete initialization sequence
;
;=======================================
;
;       Set up Initial Video Memory and Port Values:
;
;Entry: AX =    Base segment of PC Video display memory,
;       DX =    Base address I/O port for PC Video board,
;       SI =    Pointer to video initialization data string.
;
PCVINI1:PUSH  ES            ;Save entering extra segment
        MOV   PCVPORT,DX    ;Save base port address for reference
        MOV   PCV_SEG,AX    ;Save base segment for video display
        MOV   ES,AX         ;Update extra to new video memory seg
        CALL  BEAMOFF       ;Turn off video display at vert sync
        CLD                 ;All string operations in forward dir
              LODSW         ;Get fill word from init string
        MOV   PCVATTR,AH    ;Update Attribute byte for scrolling
        MOV   CX,8192       ;Fill entire size of screen (8K words)
        MOV   DI,0          ;Start at first location
        REP   STOSW         ;Init video display with fill words
        POP   ES            ;Restore entering extra segment
        CALL  VSYNC         ;Sync up new timing chain to vert sync
              LODSW         ;Get master mode, color select values
        PUSH  SI            ;Save 6845 reg value string pointer
        ADD   DX,PCVMASTR   ;Advance to master mode control port
```

```
        PUSH    AX                  ;Save mode, color select values
        AND     AX,11101111b        ;Start with video enable bit off
        OUT     DX,AL               ;In new master mode (Relative Port #8)
                                    ;Starting 6845 reg number in "AH" = 0
        MOV     CX,16               ;16 registers in controller to load
        SUB     DX,(PCVMASTR-PCV6845A) ;Get 6845 addr  (Relative Port #4)
PCVINI2:MOV  AL,AH ! OUT DX,AL      ;Set up reg number via address port
        INC     DX                  ;Point to 6845 data (Relative Port #5)
                LODSB               ;Get data for CRT controller register
        OUT     DX,AL               ;Send data byte to 6845
        DEC     DX                  ;Point back to register address port
        INC     AH                  ;Next register number to load
        LOOP    PCVINI2             ;Loop until all 16 registers set up
        POP     AX                  ;Recover master mode, color select
        ADD  DX,(PCVMASTR-PCV6845A) ;Point to master mode control port
        OUT     DX,AL               ;Set it up          (Relative Port #8)
        MOV     MODEREG,AL          ;Save a copy of the mode control
        INC     DX                  ;Point to background color reg. port
        MOV     AL,AH               ;Get the color selection
        OUT     DX,AL               ;Setup color select (Relative Port #9)
        MOV     COLSREG,AL          ;Save a copy of the color selection
        POP     SI                  ;Recover 6845 reg value string pointer
        MOV     CL,1[SI]            ;Get horizontal chars displayed count
        MOV     CH,PCVHIGH          ;Get predetermined vertical line count
    if (PCVHIGH eq 0)
        MOV     CH,6[SI]            ;Get vertical lines displayed count
    endif
        MOV     SCREEN,CX           ;Save current active dimensions
        MOV     AL,15[SI]           ;Get specified cursor starting address
        MOV     AH,14[SI]
        DIV     CL                  ;Divide by screen width to get "X,Y"
        XCHG    AH,AL               ;Row number in "AH", column in "AL"
        MOV     CURSOR,AX           ;Update cursor position storage
        MOV     AL,11[SI]           ;Get cursor end line number
        MOV     AH,10[SI]           ;Get cursor start line number
        MOV     CURSROW,AX          ;Update storage to match
        SUB     AX,AX               ;Show successful return flags
        RET
;
;=======================================
;
; Enable the video beam (normal display output).
;
BEAM_ON:PUSH    AX !   PUSH DX ;Save entering registers
        MOV     AL,MODEREG     ;Get current mode with display bit on
        JMPS    BEAMCHG        ;Output to master mode reg. and return
;
;---------------------------------------
;
; Disable the video beam (blank entire screen).
;
BEAMOFF:PUSH    AX !   PUSH DX ;Save entering registers
        CALL    VSYNC          ;Wait for vertical sync pulse to occur
        MOV     AL,MODEREG     ;Get current (old) master mode control
        AND     AL,11101111b   ;Reset mode with video enable bit off
```

```
BEAMCHG:MOV     DX,PCVPORT      ;Point to master mode control port
        ADD     DX,PCVMASTR
        OUT     DX,AL           ;Set up new value ( Relative Port #8 )
        POP     DX !   POP AX   ;Recover entering registers
        RET
;
;=========================================
;
; Wait for Vertical Sync Pulse from video controller.
;
VSYNC:  PUSH AX ! PUSH CX ! PUSH DX
        MOV     DX,PCVPORT      ;Point to video status port address
        ADD     DX,PCVSTATUS    ;( Relative Port #10 )
        MOV     CX,-1           ;Init timeout counter
VSYNC0: CALL    VSYNC3          ;Delay and test status of vert sync
        LOOPNZ  VSYNC0          ;Loop until absent (pulse start)
        JCXZ    VSYNC2          ;Abort if timeout encountered
        MOV     CX,-1           ;Init timeout counter
VSYNC1: CALL    VSYNC3          ;Delay and test status of vert sync
        LOOPZ   VSYNC1          ;Loop until vertical sync occurs
VSYNC2: POP DX ! POP CX ! POP AX
        RET
;
VSYNC3: PUSH    AX !   POP AX   ;Delay for CPU timeout
        IN      AL,DX           ;Get status of video board
        TEST    AL,VRT          ;Test for vertical sync pulse
        RET
```

```
*********************************************************
*                                                       *
*                VIDEO DISPLAY OUTPUT                   *
*                                                       *
*               VARIABLE STORAGE AREA                   *
*                                                       *
*********************************************************
```

```
   DSEG
;
PCV_SEG  DW     0       ;Base segment of screen memory
PCVPORT  DW     0       ;Base address of current I/O ports
;
PCVSAVE  DW     0       ;Hold cursor coordinates save storage
CURSOR   DW     0       ;Current cursor "Y,X" position
CURSROW  DW     0       ;Display limits for current cursor
SCREEN   RW     0       ;Current screen dimensions
HSCREEN  DB     80      ;Screen width
VSCREEN  DB PCVHIGH     ;Screen height
MODEREG  DB     0       ;Current mode register value
COLSREG  DB     0       ;Current palette register value
PCVATTR  DB     0       ;Current character attributes
PCVMODE  DB     03h     ;Line wrap on, cursor on
;
```

```
;=======================================
;
;     6845 CRT Video Controller Configuration Data Tables:
;
CCURLX  EQU  PCVCCUR shr 8       ;Upper byte of color cursor limit
CCURLZ  EQU  PCVCCUR and 0FFh    ;Lower byte of color cursor limit
MCURLX  EQU  PCVMCUR shr 8       ;Upper byte of monochrome cursor limit
MCURLZ  EQU  PCVMCUR and 0FFh    ;Lower byte of monochrome cursor limit
;
;=======================================
;
;     Init String:                        Mode: Description: Adaptor Type:
;
COLOR_ALPHA_80:                  ;Color  80 column   Color
  DW  BLANK, 0029h               ;Fill Char, Backgrnd/Master Mode Regs
  DB  71h, 50h,    5Ah,    0Ah  ;Horizontal Total, Disp, Sync, Width
  DB  1Fh, 06h,    19h,    1Ch  ;Vertical Total, Adjust, Disp, Sync
  DB  02h, 07h,CCURLX,CCURLZ    ;Scan Mode, Max Line, Cursor Start/End
  DB  00h, 00h,    00h,    00h  ;Start addr (H/L), Cursor addr (H/L)
;
COLOR_ALPHA_BW_80:               ;B/W    80 column   Color
  DW  BLANK, 002Dh
  DB  71h, 50h,    5Ah,    0Ah  ;R0   R1   R2   R3
  DB  1Fh, 06h,    19h,    1Ch  ;R4   R5   R6   R7
  DB  02h, 07h,CCURLX,CCURLZ    ;R8   R9   R10  R11
  DB  00h, 00h,    00h,    00h  ;R12  R13  R14  R15
;
COLOR_ALPHA_40:                  ;Color  40 column   Color
  DW  BLANK, 0028h
  DB  38h, 28h,    2Bh,    0Ah,  1Fh, 06h, 19h, 1Ch
  DB  02h, 07h,CCURLX,CCURLZ,    00h, 00h, 00h, 00h
;
COLOR_GRAPH_320:                 ;Color  320 X 200   Color
  DW  0000h, 000Ah
  DB  38h, 28h,    2Bh,    0Ah,  7Fh, 06h, 64h, 70h
  DB  02h, 01h,    26h,    07h,  00h, 00h, 00h, 00h
;
COLOR_GRAPH_BW_320:              ;B/W    320 X 200   Color
  DW  0000h, 200Eh
  DB  38h, 28h,    2Bh,    0Ah,  7Fh, 06h, 64h, 70h
  DB  02h, 01h,    26h,    07h,  00h, 00h, 00h, 00h
;
COLOR_GRAPH_BW_640:              ;B/W    640 X 200   Color
  DW  0000h, 071Eh
  DB  38h, 28h,    2Bh,    0Ah,  7Fh, 06h, 64h, 70h
  DB  02h, 01h,    26h,    07h,  00h, 00h, 00h, 00h
;
MONO_ALPHA_BW_80:                ;B/W    80 column   Monochrome
  DW  BLANK, 002Dh
  DB  61h, 50h,    52h,    0Fh,  19h, 06h, 19h, 19h
  DB  02h, 0Dh,MCURLX,MCURLZ,00h, 00h, 00h, 00h
;
MONO_GRAPH_BW_320:               ;B/W    320 X 200   Monochrome
  DW  0000h,       200Eh
  DB  30h, 28h,    29h,    0Fh,  7Fh, 06h, 64h, 70h
```

```
  DB   02h, 01h,   2Bh,     0Ch,   00h, 00h, 00h, 00h
;
MONO_GRAPH_BW_640:          ;B/W   640 X 200   Monochrome
  DW   0000h, 071Eh
  DB   30h, 28h,   29h,     0Fh,   7Fh, 06h, 64h, 70h
  DB   02h, 01h,   2Bh,     0Ch,   00h, 00h, 00h, 00h
```

THEORY OF OPERATION

This section is provided to enable someone with a good understanding of TTL logic to follow the hardware functions performed by the PC Video interface board.

Because of the "raster processor" nature of the CRT controller and its dual port display memory, PC Video board logic is centered around a set of clocks and state machine cycles. These control all of the display output sequencing as well as synchronization to externally generated IEEE 696/S-100 bus read and write data transfer operations. A board logic discussion will follow the master frequency signals as they are processed to become another logical function in the central timing chain.

Precise presentation of some time dependent functions is made difficult by the myriad Master Display Modes which vary the basic timing and control cycles. Where possible, general timing sequences and relationships are described to explain logic functions which have mode dependent timing. Timing cycles and control signal data appears in table format where useful, with principal timing diagrams and CRT controller chip data sheets (found at the rear of this section) referenced to aid general case discussions with specific examples.

MASTER CLOCK AND DISPLAY MODE SELECTION

The PC Video interface is always set to one of two master clock frequencies according to the type of display monitor attached. Four gates of U11, a 74LS04, are used to generate these two different frequencies for use as master timebase clocks. The first crystal controlled frequency (X1, 16.257 MHz as chosen by IBM) is selected to drive a Monochrome display monitor. The second crystal frequency (X2, 14.31318 MHz) is the fourth harmonic of the US standard television (NTSC) color burst frequency, and is selected for Color display monitors.

Primary display mode selection is made by DIP switch S2, paddle #9, which controls the MXSEL signal level. MXSEL is high (S2, 9 OFF) for a Color type display monitor, and low (S2, 9 ON) for a Monochrome type display monitor. MXSEL determines which crystal oscillator frequency becomes the unit timebase by controlling the multiplexer chip, U23 (74LS153). U23 selects the master clock (MCLK) and dot clock (DCLK) frequencies used for all other local timing.

The level of control signal MXSEL also determines which group of Master Display Mode values should be latched, by software, into relative input/output port #8. When any of the valid Master Display Modes are output to this I/O port, the PC Video board can produce functional raster output signals which display memory data at the selected type of monitor. The valid bit patterns for both Color and Monochrome Master Display Modes are shown in the following table:

Table 21: Master Display Modes

Bits: 5	4	3	2	1	0	Monitor:	Display	Function:
MXSEL:								
0	1	0	1	x	0	1	MONO	80 x 25 Alphanumeric
0	x	0	1	1	1	0	MONO	320 x 200 B/W Graphics
0	x	1	1	1	1	0	MONO	640 x 200 B/W Graphics
1	1	0	1	x	0	0	COLOR	40 x 25 Alphanumeric
1	1	0	1	x	0	1	COLOR	80 x 25 Alphanumeric
1	x	0	1	1	1	0	COLOR	320 x 200 B/W Graphics
1	x	0	1	0	1	0	COLOR	320 x 200 Color Graphics
1	x	1	1	1	1	0	COLOR	640 x 200 B/W Graphics

Relative I / O Port #8	Signal Function:	Signal Name:
	High/Low resolution mode	-- HRES
	Graphics/Alphanumeric display	-- GRPH
	Black/White graphics mode	-- SBW1
	Video display output enable	-- VIDENB
	640 X 200 black/white graphics mode	-- SBW2
	Character blink / Foreground intensity	-- ENBBLINK

MXSEL performs two other functions where differences exist between Color and Monochrome modes. It selects which half of the character generator PROM, U6 (an 8K byte 2764-2), will be used for alphanumeric character scan line construction by controlling the most significant address line input (U6-2). MXSEL also functions as an enable/disable control for portions of the Attribute decoder PAL (U36), where the special Monochome alphanumeric display control signals are generated.

CHARACTER CLOCK CYCLE SELECTION AND TIMING

Character clock (CCLK) cycle timing is generated entirely within a single 16R8 PAL at U37, referred to as the Clock/Timing PAL. This PAL has 8 output registers synchronously latched by the positive going edge of MCLK. It is programmed to behave as a "state machine". State machine logic is divided into two groups of functions, with both the Counter/Sequencer and Data Transfer groups of the Clock/Timing PAL having four register outputs each.

The Counter/Sequencer group may define a maximum of 16 logically different time states, and functions to produce the basic character clock cycle. Three different character clock timing cycles having periods of 8, 9 or 16 master clock (MCLK) ticks may be generated by the Clock/Timing PAL. The signals MXSEL, GRPH, and HRES, from the Master Display Modes, act as multiplexer controls to determine which of the three distinct timing cycles is selected. The period of CCLK is determined by the number and period of master clock states in each character clock cycle. This is dependent on the Master Display Mode signals according to the following table.

Table 22: Character Clock Cycle Selection Chart

MXSEL	GRPH	HRES	MCLK/ CCLK	Cycle Period (nanoseconds) MCLK:	CCLK:	Display Type:	Display Mode:
0	0	0	< 16 >	--	--	< invalid >	
0	0	1	9 x	61.51 =	553.6	Mono	80x25 Alpha
0	1	0	16 x	61.51 =	984.2	Mono	Graphics
0	1	1	< 8 >	--	--	< invalid >	
1	0	0	16 x	69.87 =	1117.9	Color	40x25 Alpha
1	0	1	8 x	69.87 =	558.9	Color	80x25 Alpha
1	1	0	16 x	69.87 =	1117.9	Color	Graphics
1	1	1	< 8 >	--	--	< invalid >	

Design Constraints Affecting the Character Clock Cycle

The PC Video board maintains complete compatibility with the IBM-PC Color and Monochrome interface adapters at the logical bus interface level(all I/O port and memory addressing is identical).

To avoid many of the problems associated with the IBM adapters, performance of the following two major tasks by the PC Video board served as design constraints:

(1) Displaying the local memory data array in a regular pattern on the monitor screen without "hash" or other visual interruptions.

(2) Communicating with the host processor so the host may read or write various data at memory or input/output port locations addressed on the PC Video board. It is highly desirable that communication with the host processor proceed with minimum delay at the host processor bus level.

The character clock cycle is the period of time during which a display character or graphics word is manipulated and placed at the video outputs to appear on the monitor screen. Local display memory data selection is controlled by the display raster process. This process consists of a sequence of character clock cycles combined to form horizontal scan lines, and scan lines combined to form the vertical retrace pattern. This display raster is generated by the 6845 CRT controller chip. The CRT controller chip uses the character clock period as its unit timebase. It scans the display memory addresses continuously, even though it also creates an external "blanking" control signal. This signal disables the video outputs so character data does not appear at the display monitor during the horizontal and vertical retrace periods.

The first design constraint says that each character must appear as selected by the display raster when the video output is enabled. This requires that an internal data transfer read of display memory be performed for each character clock cycle.

As you can see from Table 22, character clock periods vary from a low of 553.6 nanoseconds to a high of 1117.9 nanoseconds. The shortest cycle period is the factor limiting what may be done during each character clock. The choice of 120 nanosecond static CMOS RAM for the display memory permits a data transfer to be completed in four of the shortest period master clock ticks. This allows the completion of two synchronous data transfers during the shortest character clock period, satisfying the requirements of both design constraints.

Character Clock Cycle Function Overview

> **NOTE:** During the following discussion of functions performed by the Clock/Timing PAL, refer to the **Character Clock Cycle** timing diagrams found at the end of the Theory of Operation section.

Each character clock cycle may be logically divided into two different functional operations or "data transfer phases". The first logical function reads a memory word selected by the CRT controller and transfers it to the data dot latches and output logic. The dot output logic will then display the latched data sequentially over exactly one character clock period -- from the "latch display word" phase of a character clock cycle until the start of the same phase in the next cycle. The video output of the current display word is performed dot by dot (by DCLK, the dot clock). This occurs **concurrently** with the acquisition of the next display word.

The PC Video board must always have display data ready to be latched at the same time slice within each character clock cycle in order to maintain a consistant display. The memory read portion of the display data transfer operation requires four master clock cycles; 1 for address setup and 3 for data stabilization and register latching. To simplify the Clock/Timing PAL logic, this operation occurs at the same phase of each character clock cycle. This is independent of the selected character clock cycle length, and has been programmed to occur during the first four master clock "states" of every CCLK cycle.

The second data transfer phase is reserved for host processor communication. It is performed only if a transfer request from the host is active at that time. The period during which a host processor request is sampled is called its "initiation window". The transfer sequence will begin execution only if requested prior to the closing edge of that window. A host processor request coming too late in one character clock cycle is held until the next cycle. Execution of the internal data transfer will begin at the first opportunity after the next raster character data transfer. Since each external bus data transfer operation executes in the four master clocks which immediately

follow a request into an open window, the host processor will be delayed for the shortest time possible.

For the shortest character cycle state count, 8 master clocks, there is time for only a single sequence of events (4 master clocks) to complete a second internal data transfer necessary for host communication. The host processor request window is open only for a single master clock tick at State #3, the state machine cycle which immediately preceeds this phase.

A character clock cycle having 9 master clocks is generated by adding a "stretch" clock state at the beginning of the second (or host processor) data transfer phase. The Clock/Timing PAL logic also stretches the request initiation window by one master clock. This makes it possible for a "late" request from the host (detected in the stretch cycle after State #3) to be executed in the current character clock cycle, rather than held off until the next.

If the character clock cycle contains 16 master clocks, then the host request initiation window is open for 8 of them and closed during the final 4 of the second phase. Some or all of the final 4 clocks are occasionally needed to perform the actual data transfer operation prior to the start of the next character cycle. Having 12 master clocks for the second phase functions permits the host to perform two separate data transfers within the same character clock cycle, as happens for block moves when using a very fast host processor. This means that actually three data transfers take place in a single character clock cycle, 1 for the display data read and 2 for host processor read or write operations.

State Machine Registers - Counter/Sequencer Logic

The Clock/Timing PAL at U37 has 4 of its 8 registers programmed to function together as a 4-bit counter/sequencer. As a counter, it can define 16 discrete time states with these four register outputs, which are labeled Q1, Q2, Q3, and CCLK. Although outputs Q2 and Q3 are not connected to anything outside the PAL, they are used internally to determine the current state of the counter/sequencer.

The output signal, Q1, is a square wave at a frequency equal to MCLK/2. It is gated through the clock select multiplexer (74LS153) at U23. Q1 is selected as the dot clock (DCLK) whenever a Master Display Mode with 16 master clocks per character clock cycle is activated (see Table 21). Dividing the master clock by two to generate the dot clock produces 8 display dot time periods per character clock cycle. This matches the number of data bits from the character generator and simplifies the dot display logic for alphanumeric display modes. Additional information on the use of DCLK is covered in the section on Video Dot Multiplexing.

The character clock signal, CCLK, is inverted to drive the CRT controller timebase input (U15, pin 21) of the same name. The leading edge of CCLK also synchronizes, via U25 (74LS174 6-bit latch), the CRT controller outputs for horizontal and vertical sync, display enable, and cursor position. CCLK is programmed to be active high during the portion of each character clock cycle where external host processor data transfer operations may be performed. Consequently, it is connected directly to the PROM character generator output enable (U6, pin 22) to help reduce its power comsumption.

State Machine Registers - Data Transfer Control Logic

In addition to the four counter/sequencer registers, there are four other state machine output register signals programmed into the Clock/Timing PAL. The PAL signals, ADX, CSRAM*, S/L* and DSX, are augmented by register logic elements U24 (DXFER) and U2 (BDRDY) to form the completed state machine. Taken together, these signals control the display memory data transfer sequences as a part of each character clock cycle. A list and brief description of each control signal function is shown in the following table:

Table 23: State Machine Data Transfer Signals

Signal Name:	Function:
ADX	Dual port address multiplexer control.
CSRAM*	Chip select for dual port static RAM.
S/L*	Shift/Load control for latching of display word.
DSX	Start internal data transfer sequence for host I/O.
DXFER	Data transfer latch/buffer enable for host I/O.
BDRDY	Board ready, internal data transfer complete/lockout.

Two of these signals, ADX and CSRAM*, are used for both raster character read and external host processor read/write data transfer operations. ADX selects whether the host processor bus address (A0 thru A13 when ADX is high) or the CRT controller address (RA0, CA0 thru CA12 when ADX is low) will be passed through the address multiplexer logic (U26, U29, U30 and U31) to drive the display memory address lines.

Control signal, CSRAM*, functions to help reduce power consumption by disabling the display RAM chips when no data transfers are active. Both character (U17) and attribute (U16) display memory bytes are affected by its operation. CSRAM* is always activated to a low signal level at the beginning of a raster character data transfer phase. It will become inactive for the early portion of the host processor data transfer phase if no request is pending. If a host to display memory data transfer is active on board as indicated by RAMADR and DBACT both active high, CSRAM* remains asserted until that transfer has been completed. CSRAM* is always re-activated during the final 2 MCLK cycles of each character clock cycle to assist data stabilization in the next raster character data transfer.

Raster Character Data Transfer Phase

Each character clock cycle will prepare, for output, a single 16-bit word of display memory addressed by the CRT controller. Machine State #0 marks the beginning of this raster character (or graphics word) data transfer phase. At this time, the address multiplexers are switched by ADX (U37, pin 18) to select the CRT controller character address lines CA0 thru CA11 (74ALS157 MUX U29, U30 and U31), and either CA12 or RA0. The graphics control signal, GRPH, into multiplexer U26 (74LS153), selects CA12 for alphanumeric and RA0 for graphics Master Display Modes. This change in the raster address most significant bit (MSB) selected also causes the observed change in display memory mapping between these modes.

If an alphanumeric display mode is active, the character byte RAM then addresses 8 bits of the character generator PROM at U6 (200 nanosecond 2764-2) to produce a byte of representation dots for the current scan line. The remainder of the character generator address lines are controlled by MXSEL (switch S2, paddle 9 select for color/monochrome characters) and the scan line raster addresses RA0 thru RA3 from the CRT controller.

The Shift/Load (S/L*) control signal is used to latch display memory data fetched by the raster character data transfer phase. During Machine State #3 of every character clock cycle, S/L* is set active low to enable a broadside load of the settled display data. At the leading edge of State #4, the new display data is latched into the (74LS166) 8-bit shift registers U7, U8 and U9 by the rising edge of DCLK. The attribute byte data latch U20 (74LS273) and the ROW12 scan and character generator 9th bit extension registers of U3 (74LS74) are latched by the rising edge of S/L* at the start of State #4. Latching of the character and attribute data bytes is the final step in the raster character data transfer phase. Further processing of this data is covered in the section on "Video Dot Multiplexing".

Host Processor Requested Internal Data Transfer Phase

Data transfers, via the IEEE 696/S-100 bus, require synchronization of the host processor bus cycle to the local PC Video board character clock cycle. The host processor synchronization is performed by the Address decoder PAL (U41) as detailed in the section on "Bus Interface Logic". This PAL holds the logic state of all external bus address, data, and control signals by delaying the host processor with wait states. The necessary internal data transfer operation is then sequenced by the Clock/Timing PAL (U37) and the appropriate data is latched. BDRDY (Board Ready) is activated at the end of this internal data transfer phase and indicates that the internal and external cycles have been synchronized. This signals the Address decoder PAL wait state logic to release the bus hold and allows the host processor to finish the current bus cycle.

An internal data transfer request is flagged when the signal DBACT (Data Bus Active) becomes high with either IOADDR or RAMADR from the Address decoder PAL (U41) being active high. This indicates a new host processor data transfer request for the PC Video board I/O port or memory data. DBACT is generated by the logical "OR" of the bus signals PDBIN and inverted PWR*, which indicate an active bus read or write cycle. The internal data transfer sequence actually begins when a transfer request occurs during an open initiation window (WINDOW active high) shown as a pseudo signal in the character clock cycle timing diagrams at the end of the Theory of Operation section.

DSX (Data Transfer Sequence Start) becomes active high as the first step when the state machine begins to perform a data transfer sequence for a host processor read or write to either memory or an I/O port. The Clock/Timing PAL (U37) sets DSX to a '1' in the master clock period immediately following a valid request in an open initiation window, which initiates the transfer sequence. DSX is returned to a low state in the MCLK period following BDRDY (Board Ready) becoming active high. BDRDY active high prevents DSX from becoming active high again, thus preventing secondary transfer sequences for the same request.

The actual transfer of internal data is performed one MCLK period after DSX becomes active. The control signal DXFER (Data Transfer) is set active high via U24 (74LS166 8-bit shift register), by performing a parallel load of control bits to be shifted during the first MCLK positive edge which follows DSX going high. Once DXFER has gone active high, all remaining MCLK cycles will perform shift operations instead of a broadside load. DXFER remains active until a zero bit is shifted out by the correct number of MCLK cycles. The shift count of '1' bits is 8 when either CRT controller port is selected (approx. 500 nsec) and 2 for all other memory or I/O port data transfers.

The falling edge of DXFER sets the completion status at the logical true output of U2 (a 74LS74 flip-flop), producing the synchronization signal BDRDY. Once set, BDRDY will remain high during the remainder of the read or write operation of the host processor. BDRDY indicates that the internal data transfer associated with the current bus request has been completed. This prevents additional internal transfer cycles of the same data and flags the Address decoder PAL to release the host processor from its wait state. BDRDY will fall when its register is cleared by the DBACT control signal going low. This indicates that the host has completed the current bus cycle.

When beginning to perform a data transfer sequence for read or write of display memory, ADX is set high at the same time as DSX. However, it will remain inactive if the transfer request is for an I/O port. It becomes active high in the master clock period immediately following a valid request in an open initiation

window. This occurs one MCLK prior to DXFER going high. It will remain a '1' for one MCLK after DXFER goes low, thus bracketing the actual data transfer enable period to ensure stable memory addressing. If not actively performing the internal sequence to transfer requested host data, ADX always resides in the inactive low or '0' state.

Interaction Effects Between Adjacent Data Transfer Phases

For alphanumeric display modes, data from the character generator PROM has a minimum of 4 master clocks (States #0-#3) in which to settle for each raster character before being latched for serial dot output. This also include delays created by the address multiplexer and the corresponding unsettled character RAM data output/PROM address input lines. This minimum settling period occurs only if the host processor performed a data transfer that took place in the last 4 master clocks of the previous character clock cycle.

Using the corresponding crystal frequencies gives minimum settling periods of 246 nanoseconds for Monochrome and 279.5 nanoseconds for Color display modes. This contrasts with the combined maximum settling delays of 10 nanoseconds for the (74ALS157) address multiplexers, 120 nanoseconds for RAM and 200 nanoseconds for PROM giving a worst case total of 330 nanoseconds needed for character dot representation output settling. These combined worst case conditions can cause some visible "flickering" of characters on the display monitor, but the conditions required for this to occur are extreme.

This combination only appears while continuous reading or writing of the PC Video board is performed by the host processor. This is rarely encountered except for cases of screen scrolling or total update, where the screen image is usually altered radically enough that any flickering is not noticable. If this condition must be avoided, a software driver may sample the Display Enable status signal (Port #10, bit #0) to synchronize read and write operations to the video output blanking periods.

Since ADX will be set high only if a host processor data transfer operation was completed in the prior character clock cycle, there is usually no change in the address multiplexer selection from one raster character transfer phase to the next. Under ordinary circumstances, the display address is stable. The RAM chips are enabled by CSRAM* two master clock cycles prior to Machine State #0. This produces 6 master clock states in which the alphanumeric character dot representation normally settles. This results in settling times of 369 nanoseconds for Monochrome and 419 nanoseconds for Color display modes -- both of which exceed the 330 nanosecond maximum that might be needed.

For both medium and high resolution graphics, display memory data is transferred and latched directly without interpretation, giving a maximum settling delay of 120 nanoseconds. This is less

than the minimum settling times for both Color and Monochrome modes, so there is no possible flickering caused by adjacent data transfer operations for either of the graphics display modes.

IEEE 696/S-100 BUS INTERFACE LOGIC

Internal master clear signals, RESET and RESET*, are made active by either of the IEEE 696/S-100 bus reset lines, SLVCLR* or PRESET*, going to a low TTL level. This forces all control and timing registers to be cleared and necessitates a software controlled re-initialization of the PC Video board.

The S-100 bus address lines, A0 thru A15, are routed to at least 2 different logic elements each. Unbuffered address lines are connected to (74ALS157) selectors, (25LS2521) comparitors and PAL logic. These combine to give worst case loading characteristics that are less than the maximum specified. To comply with bus loading requirements, address lines A0, A13, A14 and A15 are buffered by half of U42, (a non-inverting 74LS244). For the same reason, SOUT, PDBIN and PWR* are buffered by portions of U39 (74F04), but are inverted to decrease propagation delays.

Bus Address Decoder Logic

The S-100 bus address detection of valid PC Video display memory space is primarily handled by an (25LS2521) 8-bit comparitor, U38, and by 8 positions of the associated DIP switch, S1. External bus address lines, A14 thru A21, form inputs to the other side of this 8-bit comparitor. The output driver of U38 is enabled by an active low value on control signal PSEL* from the Bank Select register flip-flop, U2. If PSEL* is inactive high, then the comparitor output strobe, RAMSEL*, is disabled and no external bus addresses will be recognized and indicated as valid. When PSEL* is active low, RAMSEL* will become active low when the external bus address matches the selected space for the video board display memory.

Valid external bus address detection of PC Video I/O Port space is similar to valid memory space detection, but utilizes another 8-bit comparitor (25LS2521) at U40 and 8 positions of DIP switch S2 for I/O address space selection. External bus address lines, A4 thru A11, are the inputs to the other side of this 8-bit comparitor. The output driver is enabled by an active low value on BA15, the buffered output of bus address line A15. This forces PC Video board I/O port logic to decode a full 16-bit address. To be valid, the high order bit (A15) is a zero. Block enable of the I/O port address logic for bank selection via PSEL is performed in a later decoding stage. The comparitor output signal, IOSEL*, becomes active low whenever the external bus address matches the space of the selected video board I/O port base.

Both external bus address comparitors have X-OR gates (U14, 74LS86) that may flip the preset values of a few of the DIP switches prior to becoming comparison inputs. These X-OR gates invert the comparitor "static" inputs for A15 in memory space, and for A5 and A6 in I/O port space when the output of the Address Flip register (U13) is set to a '1'. When this flip-flop (U13) is set to a '0', both address spaces selected by DIP swithes S1 and S2 are passed through the X-OR gates to the comparitors without inversion. For more details on the overall function of this logic, refer to the description of "Relative Port #1" in the Hardware Section of this manual.

The outputs of both address comparitors, RAMSEL* and IOSEL*, feed into a Master Address Decoder 14H4 PAL (U41). This PAL also handles handshaking for most of the bus control signal lines used by the PC Video board. In addition, the PAL decodes those extra bus address lines (A23 and A22 in memory space; A12, A13 and A14 in I/O space) that the 25LS2521 comparitors omitted due to their limited number of inputs. Other S-100 bus lines decoded by the Address Decoder PAL are the processor status signals SINP, SOUT, SMEMR, SWO* and SXTRQ*. By using logical combinations of the real-time values of these processor status signals, the PAL determines if the PC Video board is currently being addressed by the bus.

Address Decoder PAL output signal, IOADDR, becomes active high when any of the 16 possible I/O ports is correctly addressed by the bus. The corresponding output signal, RAMADR, becomes active high when any location of PC Video display memory is correctly addressed by the bus. Either of these "address active" output signals will remain asserted until the host processor fully completes the current bus cycle. Full bus cycle completion is indicated by deactivation of the corresponding processor status signal, SINP, SOUT, SMEMR or SWO*, which initiated the current "address active" signal.

The RAMADR active high signal is inverted by transister Q4 to produce an "open collector" low level output capable of driving an S-100 bus signal line. This transistor output may be connected to the PHANTOM bus signal pin via DIP switch S2, paddle #10 (active in the ON position.) The PHANTOM bus signal will disable all other system RAM when active low, and therefore must be used carefully during normal run-time bus operations. Under most circumstances it is not used. Refer to the discussion on "Display Memory Addressing" in the Hardware Section of this manual to determine the correct setting for this switch.

Automatic Wait State Generation

When either the IOADDR or RAMADR signal becomes active, the Address Decoder PAL (U41) also asserts its "not ready" output (pin 15) to an active high level. This is inverted by transistor Q3 to produce another open collector output which drives the PRDY (Processor Ready) bus line to a low level. Assertion of PRDY to a low state indicates that the PC Video board is not ready to

proceed to completion of the requested data transfer. This causes the host processor to wait for an indefinite period of time.

Further progress of the data transfer operation is held at the bus by PRDY until the PC Video board completes the corresponding internal data transfer sequence. The internal transfer is sequenced by the Clock/Timing PAL and related state machine logic. The length of time that PRDY will be asserted is variable. It is dependent upon the relative timing and phase of both the host processor bus cycle and the PC Video board's Character Clock cycle. This causes the PC Video board to generate a variable number of wait states for the host processor. At the same time, it permits the automatic synchronization of both timing cycles independent of any switch setting or CPU processor speed.

Completion of the internal data transfer sequence is flagged by BDRDY, the local control signal. BDRDY is asserted high when internal and external data transfers have been synchronized. BDRDY asserted high indicates that the bus cycle may be completed by the host processor, causing the Address Decoder PAL logic to release the PRDY bus signal to a high state. This release permits the host processor to complete the current bus data transfer cycle and advance to the next bus operation.

16-Bit Wide Data Transfer Acknowledge Logic

The value selected by the "Write Only Memory," or WOM mode, (setting at DIP switch S1, paddle #1) modifies a few functions of the Address Decoder PAL (U41) logic. WOM mode is disabled by setting this switch to the ON position, corresponding to a low level signal into the Address decoder PAL.

Disabling the WOM mode enables the Address Decoder PAL logic to perform the S-100 bus signal handshaking for 16-bit data transfers. The Address PAL samples bus line SXTRQ* for host processor 16-bit transfer requests during any valid display memory address period (RAMADR high). The PAL will respond to these requests by asserting high the output on pin #16 (U41), "sixteen acknowledge". This PAL output is inverted by transistor Q2 to produce another open collector control for an S-100 bus signal. The transistor output asserts the SIXTN* bus signal to a low level, which indicates that the PC Video board has acknowledged the use of a 16-bit wide data path for this bus transfer operation.

Write Only Memory (WOM) Mode Control Logic

If the WOM input signal to the address decoder PAL is active high (DIP switch S1, paddle 1 OFF,) then all S-100 bus memory read requests will be ignored by the PC Video board. Display memory write requests will still be honored as before, allowing the

video display to be updated. This permits memory data in an overlapping static RAM board to be read instead of the corresponding PC Video board memory. This will avoid the automatic wait state generation caused by synchronizing display memory to the host bus cycle. It also avoids any potential bus conflict caused by data assertion from two different memory boards existing in the same address space with differing data values or bus timing.

Enabling the WOM mode also causes the address decoder PAL to internally disable the 16-bit transfer acknowledge logic, which sets the "sixteen acknowledge" output low. Bus signal, SIXTN*, will never be asserted. This permits the 16-bit transfer acknowledge to be asserted by another RAM board which occupies the same memory address space as the PC Video board.

The logic which gates the internal data paths to buffer 8- or 16-bit wide memory data transfers to the external S-100 data buses is controlled separately by the Memory Select decoder PAL (U27), and remains active independent of WOM mode status. The PC Video board will still perform either 8- or 16-bit memory write operations in WOM mode. The bit width of the data transfer is dependent on the status of bus signal SIXTN*, which should be asserted by the co-existent static RAM board. This permits the PC Video board to follow the type of memory data transfer indicated by another read/write RAM board. The PC Video memory (in WOM mode) is therefore able to function correctly with both byte wide and "8-16" type static RAM boards.

S-100 Bus to Local Data Bus Buffer and Control Logic

The PC Video board has three internal byte wide data buses which are connected to the S-100 data bus with buffers or latches. These internal data buses are called General Purpose (DB0-DB7), Character Memory Byte (CC0-CC7), and Attribute Memory Byte (AT0-AT7) data paths. The exact composition and function of each of these is detailed further in later sections of this manual.

This section is concerned with those logic elements which interact with the host processor via the S-100 data input and data output buses. Logic and buffer elements that interconnect the three internal data buses are included because they function as an integral portion of data transfer with the host processor. The six buffer or latch elements which form these interconnections are all controlled by the Memory Select Decoder PAL (U27). The following table of interconnect logic elements includes the two RAM chips because memory read/write operations are controlled by these same Memory Select Decoder PAL signals.

Table 24: Local and Host Data Bus Interconnection Elements

Element ID:	Type:	Bus "X" to "Y" Direction:	Control Signals for Enable:	Direction:	Latch:
U16	6264	ATx <==> RAM		ATMWR*	ATMWR*
U17	6264	CCx <==> RAM		CCMWR*	CCMWR*
U32	74LS245	BDx <==> ATx	ATMSEL*	ATMWR*	
U33	74LS245	BDx <==> CCx	CCMSEL*	CCMWR*	
U43	74LS244	DOx ==> BDx	DILEN*		
U44	74LS244	DIx ==> ATx	DIHEN*		
U45	74LS373	DOx <== CCx	DOHEN*		DXFER
U46	74LS373	DIx <== BDx	DOLEN*		DXFER

Bus Key: DOx = IEEE 696/S-100 Data Output Bus
DIx = IEEE 696/S-100 Data Input Bus
BDx = PC Video board General Purpose Bus
CCx = PC Video board Character Memory Bus
ATx = PC Video board Attribute Memory Bus

Memory Select Decoder and Data Bus Buffer Control Logic

A host processor bus cycle can perform any one of eight major types of data transfer operations with the PC Video board. These include both read and write transfer directions with an I/O port byte, a Character or Attribute Memory byte, or a Display Memory word (both Character and Attribute bytes taken together.) The Memory Select decoder is a (10L8) PAL programmed to correctly route data between the three internal data bus paths and the host processor data buses for each transfer type.

The four control outputs of this PAL for Character and Attribute Memory data bus select and write operations (CCMSEL*, CCMWR*, ATMSEL* and ATMWR*) are enabled only while the ADX input is active high. This prevents assertion of any other data during acquisition of Raster Character data for display output. This also applies to the DIHEN* signal, which is used to directly buffer the host DIx data bus as the high order output byte during 16-bit display memory write operations.

The remaining control signals for buffering data between the General Purpose internal bus and the S-100 bus, (DILEN*, DOLEN* and DOHEN*), are asserted for the entire period that a valid read or write cycle is active. The Memory Select Decoder PAL output signals are shown in the following table along with the combinations of active levels that perform each type of data transfer operation. Transfers between the host and display memory utilize the local General Purpose bus as an intermediate pass thru buffer to either the Character or Attribute Memory buses. The pattern of control signals asserted reflects this additional buffering.

Table 25: Memory Select Decoder PAL Output Signals

Signal:	Function:	State Patterns for Valid Host Data Transfer Types:							
CCMSEL*	character memory select								
CCMWR*	character memory write								
ATMSEL*	attribute memory select								
ATMWR*	attribute memory write								
DOHEN*	data out high byte								
DIHEN*	data in high byte								
DOLEN*	data out low byte								
DILEN*	data in low byte								
Host Data Transfer Type:									
I/O Port	Byte Read	1	0	1	1	1	1	1	1
I/O Port	Byte Write	0	1	1	1	1	1	1	1
Character	Byte Read	1	0	1	1	1	1	1	0
Character	Byte Write	0	1	1	1	1	1	0	0
Attribute	Byte Read	1	0	1	1	1	0	1	1
Attribute	Byte Write	0	1	1	1	0	0	1	1
Display	Word Read	1	0	1	0	1	0	1	1
Display	Word Write	0	1	0	1	0	1	0	0

Local General Purpose Data Bus Functional Elements

The local General Purpose memory and I/O port data bus ties the various programmable logic elements together on the PC Video board. As described in the previous section, this local bus is logically connected to the external host processor by an input buffer and an output latch. Both the Character and Attribute Memory data buses are also logically connected to the General Purpose bus via a bidirectional buffer. These four buffer elements account for all of the host processor and display memory interface logic connected to this local General Purpose bus.

The remaining logic elements connected to the local General Purpose bus (less the 4 inter-bus buffer elements) are all latches, registers or buffers for the various PC Video board input/output port functions. Most of these are control mode latches or registers, but a keyboard data input register and status buffer as well as the 6845 CRT controller are also connected. The set of IC's attached to the General Purpose data bus is listed in the following table, which identifies the relevant characteristics of each logic element.

Table 26: General Purpose Data Bus BD0-BD7 Connections

ID:	Data Bus Direct:	Bit Usage:	IC type:	Functional description:	
U2	Input	via J3	74LS74	Bank Select Control register	Port
U13	Input	BD0	74LS74	Address Flip Control register	Port
U15	I/O	BD0-BD7	6845	CRT Raster Display Controller	Port
U18	Input	BD7	74LS74	Keyboard Clock Control reg.	Port
U19	Output	BD0-BD7	74LS322	Keyboard Data Input shift reg	Port
U32	I/O	BD0-BD7	74LS245	Attribute Memory Bus buffer	ATx
U33	I/O	BD0-BD7	74LS245	Character Memory Bus buffer	CCx
U42	Output	BD0-BD3	74LS244	Keyboard Data / Raster Status	Port
U43	-->	BD0-BD7	74LS244	IEEE 696 Data Bus input buffer	DOx
U46	<--	BD0-BD7	74LS373	IEEE 696 Data Bus output latch	DIx
U47	Input	BD0-BD5	74LS174	Master Display Mode register	Port
U48	Input	BD0-BD5	74LS174	Background Color Control reg.	Port

Input/Output Port Select Decoder and Data Buffer Control

All of the input/output ports addressed by the PC Video board are strobed by the I/O Port Select decoder PAL (U28). This PAL works in conjunction with the Memory Select decoder PAL to perform data transfers at the selected port element. The Memory Select PAL handles the routing of internal and external data buses to buffer the local General Purpose data bus to the host processor data input and data output buses. The I/O Port Select PAL then gates the desired port logic element onto the General Purpose bus to complete the data path connection with the host processor.

The I/O Port Select PAL is programmed to decode the S-100 bus address lines, A0 thru A3, in conjunction with the data direction signals, PDBIN and PWR. This programming will activate the correct control strobe corresponding to a recognized relative port number. Each control strobe is activated only for that portion of time that the data transfer signal, DXFER, is active high (2 or 8 master clocks.) The control strobes generated by the I/O Port Select PAL are given in the following table, along with the function they perform, and the specific logic elements affected.

Table 27: Input/Output Port Control Strobes

Strobe Name:	Relative Port #:	Strobe Function:	Loc:	Logic Type:
SETPSEL*	0	Load Bank Select reg.	U2	1/2 ff
SETADDR*	1	Load Address Flip reg.	U13	1/2 ff
CS6845*	4,5	Load Address, R/W Data	U15	CRT chip
SETMODE*	8	Load Master Display Mode	U47	6 bit reg
SETBKGND*	9	Load Background Color	U48	6 bit reg
STATUS*	10	Input Status buffer	U42	4 bit
KBDCLR*	11	Load Keyboard Clock reg.	U18	1/2 ff
		Clear Keyboard Data	U19	8 bit S/R
		Clear Data Avail. Flag	U13	1/2 ff
		Clear Light Pen Flag	U18	1/2 ff
KBDENB*	12	Input Keyboard Data	U19	8 bit S/R
		Preset Data Avail. Flag	U13	1/2 ff
		Preset Light Pen Flag	U18	1/2 ff

The function of the I/O Port Select PAL is modified by the control signal, PSEL, from the Bank Select register (flip-flop U2). If the Bank Select register is set so that PSEL is a '1', then all of the port control strobes are active as described above. If PSEL is a '0', then all of the port control strobes of the PAL except SETPSEL* are disabled. This permits the Bank Select register to enable or disable all of the control ports on the PC Video board with the singular exception of itself. If the function of SETPSEL* were also disabled, it would then be impossible to enable the board again without activating either the SLVCLR* or PRESET* S-100 bus reset line. This would force a total reset of the PC Video board.

CRT CONTROLLER AND VIDEO DISPLAY RASTER TIMING LOGIC

Video display raster timing generated by the CRT controller (CRTC) is based on the Character Clock (CCLK*, U15 pin 21) unit timebase. In the data sheets reproduced at the end of the Theory of Operation section, each of the 6845 CRTC internal registers, R0 thru R17, is described in detail. Several related timing diagrams for raster scanning are also included. Please review these data sheets carefully to get a basic understanding of the internal logic and controls of the CRTC chip. The next several paragraphs are included to help clarify or amplify on some of the information presented in the CRTC data sheets.

The 6845 CRTC data sheets refer to the PC Video board's internal dual port display memory as "Refresh Memory" because the video display monitor must be refreshed continuously to maintain a visible image. Although this will also automatically refresh dynamic RAM used for display memory, DRAM requires more circuitry at both higher speed and cost to be able to perform the dual data transfer cycle in each character clock necessary for high speed, flicker free displays.

At the beginning of each horizontal scan line, the Character Address, placed on lines CA0 thru CA12, is that of the left-most displayed character. This address is incremented on each CCLK* negative going edge to select the next highest character address. This is usually the next horizontal character needing display. The character address continues to be incremented linearly beyond that of the right-most displayed character until the horizontal total value stored in CRTC register R0 has been reached.

Synchronous with the change in Character Address at the end of each horizontal scan line, the Raster Address placed on lines RA0 thru RA3 is incremented. For non-interlaced displays, it continues to increment by one at the end of each scan line until one less than the maximum raster address value in CRTC register R9 has been reached. In this case, the next raster address will then be zero. Both the character and raster addresses are reset to zero at the first vertical line displayed in each video frame. This occurs immediately after the number of horizontal scan lines has reached the vertical total value held in CRTC registers R4 and R5.

CRT Controller I/O Port Interface Timing and Control Logic

The 6845 CRT controller chip (U15) requires some unusual interface control logic due to its "clocked" internal buffer. This buffer is connected to the PC Video board's local General Purpose data bus. For the 6845 to pass a data byte in either direction, it is necessary first to enable its Chip Select (pin 25) with an active low signal. Secondly, the data must be clocked in or out by strobing its Enable (pin 23) with a high level signal pulse for a minimum of 450 nanoseconds (nsec). Minimum times of 140 nanoseconds setup, and 10 nsec hold require Chip Select to be enabled prior to and after the transition edges of the Enable pulse.

To satisfy these requirements, the 6845 Chip Select control signal, CS6845*, differs from the other control strobes generated by the I/O Select Decoder PAL (U28). CS6845* is not gated as usual by DXFER, the data transfer strobe signal, but instead is enabled low at just the condition of IOADDR active high and either relative port address #4 or #5 decoded from the external address bus. The other signals with required setup times are SOUT* (U15, pin 22) and BA0 (U15, pin 24). Both of these signals are stable prior to IOADDR. DXFER is enabled a minimum of 2 master clocks (140 nsec Color, 123 nsec Monochrome) after PDBIN or PWR* becomes active, either of which must follow IOADDR becoming stable. The total will exceed the minimum setup time.

DXFER remains enabled for 8 master clocks whenever CS6845 (inverted from CS6845* by U5) is selected. This gives either a 559 nanosecond Color or 492 nsec Monochrome data transfer pulse. This is logically ANDed with CS6845 by a gate of U1 (74LS00) followed by another inverter of U5 to produce the 6845 Enable control signal found at U15, pin 23.

CRT Controller Output Synchronization Interface and Timing

Please refer to the Character Clock Cycle timing diagrams at the end of this section for the following discussion.

Each Raster Character is latched for output by the rising edge of Shift/Load (S/L*), which signals the end of the first data transfer phase of each CCLK cycle. This occurs at the same time that the signal, CCLK, makes a positive going transition, and is independent of the number of MCLKs in the specific CCLK cycle. (Synchronous positive going transition edges of CCLK and S/L* are not accidental.)

The Character Clock Cycle also drives the unit timebase input of the 6845 CRT controller (CRTC) that produces the display raster timing. The CRTC is clocked to the next display state by the negative going edge of CCLK* (U15, pin 21) which is CCLK inverted by U39 (a 74F04). This is identical to a positive going transition of the logically true CCLK signal.

The combined results of the two previous paragraphs state that each raster character is latched for dot multiplexed output by the same clock edge that advances the character address of the display memory used to locate that raster character. This means that the character currently being displayed was actually addressed by the CRTC in the CCLK cycle immediately preceeding it. A small amount of juggling with the various output delay and setup times involved in CRTC addressing and latching of an alphanumeric display character will quickly convince you that this is unavoidable.

The 6845 CRTC produces four output signals for external hardware synchronization to the display data scanned by the Character and Raster Addresses. Since the data addressed for display is actually output to the video monitor in the CCLK cycle which follows, the timing of these four synchronization signals is corrected to match by latching them on the positive going edge of CCLK into U25 (a 74LS174). This is also needed to stabilize any drift and compensate for the large variation in delays in the CRTC outputs from the CCLK* input transition. The synchronization signals and their delayed outputs are described in the following table. All four of these signals are utilized in the final stages of output Dot Multiplexing.

Table 28: CRTC External Hardware Synchronization Signals

Signal:	Delayed 1 CCLK:	Description:
HSYNC	DLYHSYNC	Horizontal osc. syncronizing pulse
VSYNC	DLYVSYNC	Vertical osc. synchronizing pulse
DSPENB	DLYDSP	Display Enable gate video dot output
CURSOR	DLYCUR	Cursor Enable character dot override

Two of these three signals, HSYNC (J1-AC) or DLYHSYNC (J1-BC) and DLYVSYNC, are buffered to drive the phase and frequency of the video display monitor horizontal x-axis and vertical y-axis scanning oscillators. Changing jumper J1 from AC to BC will delay the horizontal sync pulse by one character clock period. This moves the display left by one character width.

Delayed Display Enable control (DLYDSP) gates the video intensity, or z-axis to map the valid data of each video scan frame onto the active display surface. Valid scan frame data includes only those portions within the programmed horizontal and vertical "displayed" limits. This prevents extraneous data scanned during horizontal or vertical retraces from being made visible because it lies outside those display limits.

Delayed Cursor Enable control (DLYCUR) is another z-axis modifier which can be used to visually indicate a special location without affecting the actual data at that location. DLYCUR flags single character wide portions of any horizontal scan lines which have a Character Address matching that programmed into the "Cursor Position" registers R14 and R15. The horizontal scan lines must also have a Raster Address which falls between the limits programmed into "Cursor Raster" registers R10 and R11.

CRT Controller Synchronization Status Logic

Two of the CRTC external hardware synchronization signals are made available at the Status Port buffer (U42, part of a 74LS244). DLYVSYNC appears directly at bit #3, and can be used to tie display memory block moves to this blanking period. It also can be polled to synchronize re-initialization of the timing chain by programming the CRTC to prevent the screen from rolling when switching between alpha and graphics modes.

DLYDSP is inverted at an otherwise spare output of the Multiplexer PAL (U49), and buffered by U42 to become bit #0 of the Status Input Port. This bit is often used to test that the timing chain is indeed valid and active, but has little other use outside of maintaining IBM-PC compatibility.

DISPLAY MEMORY DATA BUSES AND RASTER CHARACTER LATCH LOGIC

When both the Character (signals CC0 thru CC7) and Attribute (AT0 thru AT7) Memory local data buses are taken together, they form the full 16-bit wide Display Memory bus. This section is concerned with the logic elements of the full display memory bus. These logic elements are latched at the Raster Character phase of each character clock cycle. They are divided into two logical sets, each associated with one of the byte wide local memory buses and include all relevant signals latched for later output by the Video Dot Multiplexing logic.

Character Memory Data Bus and Related Logic

The Display Character Memory bus is a byte wide data path. It is connected to the General Purpose data bus via a bidirectional buffer (U33, a 74LS245), as well as the S-100 bus for 16-bit wide memory output transfers via a 3-state output latch, (U45, a 74LS373). Gating of these buffers to effect external data transfers has been described previously. The set of Character Memory bus connections includes those Raster Character logic elements that are latched as a consequence of the CCx bus data. These are shown in the following table:

Table 29: Character Byte Data Bus CC0-CC7 Connections

ID:	Data Bus Direct:	Bit Usage:	IC type:	Functional description:
U3	from	U4,U6	74LS74	Latch for 9th bit of Scan Byte
U4	Input	CC6,CC7	74LS20	Line Drawing character detector
U5	Input	CC5	74LS04	Inverter for character bit
U6	Input	CC0-CC7	2764	Character Scan Byte Generator PROM
U7	from	U3,U6	74LS166	Character Scan Byte shift register
U8	Input	CC0,2,4,6	74LS166	Graphics Bit #0 shift register
U9	Input	CC1,3,5,7	74LS166	Graphics Bit #1 shift register
U17	I/O	CC0-CC7	6264	Character Byte Display Memory RAM
U33	I/O	CC0-CC7	74LS245	General Purpose Data Bus buffer
U45	<--	CC0-CC7	74LS373	IEEE-696/S-100 Data Bus buffer

Character Memory data is latched in two different sets of shift registers for later output as either alphanumeric or graphic data dots. This bus directly drives the 4 high order input bits of U8 and U9 to become graphics output signal bits CB0 (even) and CB1 (odd). For the generation of alphanumeric display dots, the data from the character memory data bus is first passed through the Character Generator PROM (U6) to produce the decoded "scan byte" dots for each relative scan line of a character row.

Each of the 256 characters that can be displayed is represented by 16 bytes in the CG PROM. One byte is for each of the horizontal scan lines of that character (8 scan bytes for Color, 14 for Monochrome,) with the master scan byte group (Color or Monochrome half of the PROM) selected by MXSEL. The specific byte of data dots selected by the PROM depends on the row address number controlled by RA0 thru RA3, as generated by the CRT controller chip. Each of these character scan bytes is latched by U7 to produce the serialized character dots signal, SDOTS, and routed to the Attribute decoder PAL (U36) for further processing prior to actual output.

For Monochrome displays, each character scan byte is actually constructed and displayed as 9 dots. Under most circumstances the ninth bit displayed is a zero, which most often is used to separate one character horizontally from another. This means that the character generator (U6), can utilize a full 8-bits of horizontal scan data (instead of just 7 for Color) for each character. This results in greater resolution for the Monochrome

display. The exception to this "zero 9th bit" rule is the "Line Drawing" graphic characters, which need to utilize this 9th dot to produce graphics representations without blank gaps between adjacent horizontal characters. To correct for this condition, the 8th dot value is repeated as the 9th dot for these characters.

To accomplish this, the range of line drawing characters (C0 thru DF hexadecimal) is logically "ANDed" with the character scan dot #8 via 4-input NAND gate (U4, a 74LS20). Its output is latched via flip-flop (U3, a 74LS74), to become the 9th bit of the character serial dots (SDOTS) shift register. The output of U3 will follow the CG PROM (pin 11) 8th bit signal level whenever the character memory byte has bits CC7 and CC6 high and CC5 low. In all other cases the output of U3 will be low for normal intercharacter spacing.

Attribute Memory Data Bus and Related Logic

The Display Attribute Memory bus is also a byte wide data path and connected to the General Purpose data bus via another bidirectional buffer (U33, a 74LS245). It differs from the Character Memory bus in that it is connected directly with the S-100 bus for 16-bit wide memory input data transfers via an input buffer (U44, a 74LS244). Again, these buffers effect external data transfers. The complete set of Attribute Memory bus connections is shown in the following table. This table also includes those logic elements (U3, U4 and U5) producing latched signals that affect attribute bit decoding prior to Video Dot Multiplexing.

Table 30: Attribute Byte Data Bus AT0-AT7 Connections

ID:	Data Bus Direct:	Bit Usage:	IC type:	Functional description:
U3	from	U4	74LS74	Latch for Row 12 Scan Byte
U4	Input	U5, RA2,3	74LS20	Row 12 Scan Byte/Line detector
U5	Input	RA0,RA1	74LS04	Inverter for character bit
U8	Input	AT0,2,4,6	74LS166	Graphics Bit #0 shift register
U9	Input	AT1,3,5,7	74LS166	Graphics Bit #1 shift register
U16	I/O	AT0-AT7	6264	Attribute Byte Display Memory RAM
U20	Input	AT0-AT7	74LS273	Attribute Byte holding latch
U32	I/O	AT0-AT7	74LS245	General Purpose Data Bus buffer
U44	-->	AT0-AT7	74LS244	IEEE 696/S-100 Data Bus buffer

Attribute Memory data is latched in two different sets of registers for later output as either alphanumeric or graphic data dots much the same as for Character Memory data. The attribute bus directly drives the remaining 4 low order inputs to U8 and U9 to become graphics bits CB0 and CB1. Generation of both graphic and alphanumeric display dots utilizes attribute bus data first latched by an 8-bit holding buffer (U20), producing data bits AT0X thru AT7X corresponding to inputs AT0 thru AT7.

The values of RA0 thru RA3 are decoded in 2 inverters of U5 and the remaining section of NAND gate U4 to detect the presence of "row 12". This is then latched in flip-flop U3 at the same time as attribute bits are held in U20. This "row 12" signal, along with latched attribute bits AT0X-AT2X and AT4X-AT6X, is used by the Attribute Decoder PAL to generate the underline function for Monochrome display modes.

VIDEO DOT MULTIPLEXING AND OUTPUT DRIVERS

For a review of relevant event timing prior to actual output dot construction, refer back to the sections on the Character Clock Cycle Function and Raster Character Transfer Phase. The specific data latched for this construction is covered in the sections on display memory data bus logic. This section is concerned with the timing and routing of those latched display memory data bits (video dot multiplexing) to make them produce the desired image on the display monitor.

Attribute Decoder Logic for Alphanumeric Modes

The Attribute decoder PAL (U36) is a 14H4 positive true logic gate array that performs the function of a "pre-multiplexer" to generate three formal output signals (ADOTS, REVERSE and INTENS) to the final stage of video dot mux logic. A fourth output of this PAL generates the CBLINK signal which is used entirely locally for the character blink logic.

There are two types and rates of blinking alphanumeric display functions on the PC Video board, Character Blink and Cursor Blink. The blink rate of both of these is governed by a dual 4-bit binary counter (U12, a 74LS393). The pin 1 input to the staged dividers is DLYVSYNC, the vertical sync pulse signal. This is usually programmed in the 6845 CRTC to produce pulses at a rate of about 30 per second to maintain a valid display for most monitors. The first divider stage is fully utilized to produce the BLINKER Character Blink Rate signal at 1QD pin 6, having 2 blinks per second. This is fed into input pin 13 of the second stage, which is tapped at 2QA pin 11, the first division output to produce the CURBLINK Cursor Blink Rate signal at about 1 blink per second. The Cursor Blink Rate connects via Jumper J5 to input pin 3 of the Attribute PAL (U36). If Jumper J5 is removed, a steady non-blinking cursor is produced.

The control signal, ENBBLINK, from the Master Display Mode register, directs the Attribute decoder PAL as to the manner in which the AT7X attribute bit will be utilized for alphanumeric character display. If ENBBLINK is set to a '0' signal level, the signal level of AT7X will be passed on as the character background intensity bit, INTENS, to the final decoder mux logic. This will also disable the character blink function logic which is in the Attribute PAL.

If ENBBLINK is enabled by being set to a '1' signal level, INTENS will be forced to an inactive '0' level, showing only a low level of character background intensity. With ENBBLINK enabled, the Attribute PAL character blink signal, CBLINK, will follow the signal level of attribute bit AT7X. This is logically ANDed in U1 (74LS00) with the character blink rate, BLINKER. This produces the negative true logic enabled character blink, CXBLINK input back into U36 pin 13. The last portion of the character blink function logically ANDs the negative true signal, CXBLINK, with the serial character (SDOTS) to form the normal alphanumeric dots (ADOTS) signal. The character blink function will cause an alpha character to be periodically "blanked" in its display cell so that only the background color is present at the time of the blink.

When activated, the cursor blink function differs from the character blink in the manner in which the alpha character will be periodically blanked. The cursor blink function also blinks at half the character blink rate. Cursor blink forces a programmed range of raster lines of the particular character's display cell to the foreground color instead of its background. When DLYCUR flags the raster lines of a character with CURBLINK high, the cursor blink function forces ADOTS high, overriding normal output with a series of solid character raster dots which produces the observed cursor.

Both underline and REVERSE video functions are specific to the Monochrome Alphanumeric display mode. They become enabled only when MXSEL is set by switch S2, paddle 9 to a '0' signal level. To help compensate for the lack of colors, attribute bits AT0X-AT2X, AT4X-AT6X, and ROW12 raster address underline flag are decoded in the Attribute PAL to produce the functions shown in the following table.

Table 31: Special Attribute Function Decoding

Function:	MXSEL	AT7	AT6	AT5	AT4	AT3	AT2	AT1	AT0	ROW12
underline	0	---	0	0	0	---	0	0	1	1
REVERSE	0	---	1	1	1	---	0	0	0	---

REVERSE is output from the Attribute PAL as a high signal when the attribute bit pattern above is detected. It is logically combined with the ADOTS signal to form an exclusive-or function in the mux logic to actually reverse the polarity of dots displayed.

The underline function is activated at a single raster address line of each character (Row #12 of 13) when the attribute bits match the pattern in the above table. The display output is modified in a similar manner to that of cursor blink function. When the underline function is activated, it overrides the normal output logic and forces ADOTS high for the single raster line at row 12 of the particular character's display cell. This produces a single displayed underline at the foreground intensity for that character.

Video Dot Multiplexer and Decoder Logic

The video dot multiplexer is centered around the Multiplexer PAL (U49). This produces the composite black and white video dot signal, B/W. It also drives control inputs MUXA, MUXB and CSTRB* from a pair of dual 4-to-1 line multiplexers (U34 and U35, both 74LS153's).

The multiplexer array selects a group of four inputs that are all derived from the same type or class of source signal using two control signals, MUXA and MUXB. The matrix showing this relationship and the associated class titles is given in the following table.

Table 32: Dot Multiplexer Select Matrix

MUXB	MUXA	Blue:	Green:	Red:	Intensity:	Type or Class:
0	0	AT0X	AT1X	AT2X	AT3X	Alphanumeric Foreground
0	1	AT4X	AT5X	AT6X	INTENS	Alphanumeric Background
1	0	SELBLU	CB0	CB1	BAKHI	320 Graphics Foreground
1	1	OVRBLU	OVRGRN	OVRRED	OVRHI	320 Graphics Background
1	1	OVRBLU	OVRGRN	OVRRED	OVRHI	Outside Border except for
1	1	OVRBLU	OVRGRN	OVRRED	OVRHI	640 Graphics Foreground

During main screen area display indicated by DLYDSP high, MUXB selects between Alphanumeric and Graphic input groups, and MUXA selects between Foreground and Background color groups within the class of input group. When DLYDSP falls to a low value to indicate display of the outside border area, both MUXB and MUXA are forced high. This selects the 320w x 200h graphics normal background input group, which makes it the outside border for both alphanumeric and 320w x 200h graphics.

Both MUXA and MUXB are again forced high when the SBW2 control indicates that 640w x 200h graphics is active. This makes 320w x 200h graphics background into the 640w x 200h foreground. The multiplexer outputs are enable strobed with CSTRB* as the control for foreground/background multiplexing. When CSTRB* is inactive high, all four mux outputs are held low, which makes 640w x 200h background and outside border colors always black. CSTRB* is active low in all modes except 640w x 200h graphics (SBW2 active) to enable the video dot mux outputs at all times.

Alphanumeric modes (GRPH inactive low) cause the MUXA control to follow the exclusive-or of REVERSE and ADOTS signals from the Attribute Decoder PAL (U36). The X-OR function and selection of these signals to drive MUXA is performed by logic in the Multiplexer PAL (U49). The multiplexer selects from the two groups of latched attribute bits and INTENS.

Graphics modes are indicated by GRPH active high. Graphics modes utilize graphics serial dot bits, CB0 and CB1, as inputs to affect the foreground/background multiplex control signals instead of ADOTS. For 320w x 200h graphics, MUXA is used as the dot multiplexer control signal, and is the logical "OR" of the two graphics bits CB0 and CB1. For 640w x 200h graphics, CSTRB* is used as the dot multiplexer control, and is driven by CB0 when DCLK is low and by CB1 when DCLK is high. The multiplexer inputs which are selected for graphics color output are detailed in the following section.

The composite black and white output signal, B/W, from the Multiplexer PAL follows the inverted MUXA signal level for all modes except 640w x 200h graphics, where it follows the inverted signal level of CSTRB*.

Background Color Register and Graphics Color Logic

The four bits which are selected normally for 320w x 200h graphics background and for the outside border display area (OVRHI, OVRRED, OVRGRN and OVRBLU) are latched by the Background Color register port. These bits are independent of any display memory or other dynamic process on the PC Video board and are therefore used when a static control color is needed for output.

The Background Color register contains two other controls as shown in the following table. BAKHI is actually the 320w x 200h graphics mode foreground intensity control bit. OVRHI is the corresponding background intensity control.

Table 33: Background Color Register Functions

Bit:	Signal:	Function:
5	COLORS	320 x 200 Graphics palette select
4	BAKHI	Intensity foreground level for graphics
3	OVRHI	Intensity background level
2	OVRRED	Red background color
1	OVRGRN	Green background color
0	OVRBLU	Blue background color

SELBLU is the multiplexer input for the graphics foreground color blue. It is decoded in the Multiplexer PAL (U49). If the Master Display Mode control signal (SBW1) is active high, then the graphic serial dot signal (CB0) is passed thru to become SELBLU. This causes both the green and blue display signals to both follow CB0, which is the black/white medium resolution graphics display mode.

If control signal SBW1 is inactive low, then medium resolution color graphics mode is active, and the SELBLU multiplexer input is independent of the other foreground color inputs. SELBLU will follow the signal level of the COLORS control bit of the Background Color register to select one of two (with or without blue graphic dots) foreground color palettes.

TTL Level Video Output Signal and Driver Logic

Buffer U22 (74LS244) drives all of the signal lines at output connecter #1 for both Monochrome and Color type of display monitors. Capacitors are attached across ground and video frequency outputs Red, Green, Blue, Intensity and Black/White to help reduce reflections caused by TTL drive level outputs. These outputs have unmatched impedences with the interface cable and display monitor. This can cause the appearance of ghosts and other related interference effects. The relationship between the buffered signals appearing at Connector #1 at the top of the PC Video board and the pins used in either type of display monitor interface cable are given in the following table.

Table 34: TTL Level Video Display Output Connections

PC Video Board Video Output Connector #1		Video Display Interface Cable		
Signal Name:	Pin #:	Signal Name:	Color Pin #:	Monochrome Pin #:
Signal Ground	1	GND	1	1
Signal Ground	2	GND	2	2
Red	3	R	3	---
Green	4	G	4	---
Blue	5	B	5	---
Intensity	14	I	6	6
Black/White	15	B/W	---	7
Horiz Sync	16	H	8	8
Vertical Sync	17	V	9	9

The signal for the horizontal sync is selected by jumper J1 to be either HSYNC or DLYHSYNC. If HSYNC is selected (J1-AC), then the output will be closest to that of the IBM-PC interface adaptors. However, it is subject to some drift with temperature and the particular 6845 CRTC chip installed. If DLYHSYNC is selected (J1-BC), then the horizontal sync pulse is delayed by 1 character clock (CCLK) cycle and consequently shifts the video display outputs 1 character to the left at the monitor. It is then latched to gain stabilization.

The vertical sync pulse may be either positive or negative true by the setting of switch S1, paddle 10. If OFF, S1, paddle 10 output gives a high input to pin 2 of U14 (a 74LS86 exclusive-or gate). This X-OR gate inverts DLYVSYNC to produce an inverted (negative true) vertical sync at the buffer output as needed by the IBM-PC Monochrome display monitor. If S1, paddle 10 is ON, the resulting low level signal into U14 causes DLYVSYNC to be passed thru without logical inversion. This produces the positive true vertical sync used by Color type display monitors.

Composite Video Output Signal Logic

The black and white composite video output will produce an EIA RS-170 (NTSC) 1 Volt P-P signal level into a 75 ohm load. This is accomplished by the use of only TTL logic and a 3 diode (D1-D3), 4 resistor (R5-R8) network. The output may be at one of four voltage levels depending on the signal levels of the four digital inputs LB/W, HILITE, DLYHSYNC and DLYVSYNC according to the following table.

Table 35: Composite Video Output Signal Levels

Output Voltage:	LB/W	HILITE	DLYHSYNC	DLYVSYNC	Description:
0.04	0	x	0	1	Vertical Sync
---	1	x	0	1	< Illegal >
0.04	0	x	1	0	Horizontal Sync
---	1	x	1	0	< Illegal >
0.27	0	0	0	0	Black Video Level
0.27	0	0	1	1	Vertical Sync
0.79	1	0	0	0	Normal White Level
1.00	1	1	0	0	Hilite White Level

NOTES: (1) Output voltages above assume a 75 ohm load.
(2) x = don't care.

KEYBOARD AND LIGHT PEN INPUT

The keyboard clock control register is formed by a 74LS74 flip-flop (U18) and one of the 74LS266 open collector output X-OR gates of U10. U18 is clocked on the positive edge of KBDCLR* and latches the value of DB7 from the General Purpose data bus. This is inverted by the OC X-OR gate. If the DB7 bit is latched as a '1', the keyboard clock (KCLK) signal is forced to a low state at the external keyboard interface (Connector #1 pin 10). If the latched bit is a '0', then KCLK is free to function normally and synchronize serial data sent from the keyboard.

KCLK is inverted by one gate of U39 (74LS04), and then delayed by 2 CCLK cycles (1-2 microseconds) by using 2 bits of the raster character synchronization latch (U25). This produces the delayed keyboard clock signal (DLYKCLK). DLYKCLK is used to shift in serial data from the keyboard. The added delay ensures that these data input signals are stable prior to being latched.

Keyboard and Light Pen data input logic is centered around a 3-state output 8-bit shift register (U19, a 74LS322) and two 74LS74 flip-flops (U18 and U13). Flip-flop U18 is used to catch the positive going edge of the LPEN INPUT* signal produced by external logic at the light pen itself. This appears at pin 23 of Connector #1. This transition is latched by U18. The negative true output (U18, pin 6) drives the Preset input of U13. This asserts the control signal KBDSTB high at the logical true output (pin 9). The positive going transition of KBDSTB also latches

the current Raster Address into the Light Pen registers within the 6845 CRT controller chip, and may be used as the locator values for light pen position determination.

To assist in the detection of an intended or "static" Light Pen input versus one that was accidentally triggered or "dynamic", a second Light Pen signal, LPEN SW* (Connector #1, pin 22) is buffered by part of a 74LS244 (U42) to become bit #2 of the status register input at relative port #10. This may be polled by software to distinguish between static and dynamic positional inputs as flagged by the user.

Flip-flop U13 also acts as the ninth bit of a 9-bit shift register. The lower 8 bits come from the keyboard character byte input register (U19). This logical 9-bit shift register is clocked by DLYKCLK. DLYKCLK is generated from the keyboard synchronization clock signal, KCLK. The serialized character data (KDATA) from the keyboard appears at Connector #1 pin 11. It is directly connected to the serial data input of shift register U19.

Each character sent by the keyboard consists of a "start" bit followed by 8 data bits. The start bit is always a '1', which is shifted into U13 when the 8th character data bit (9 bits total) has been received. The logical true output of U13 is KBDSTB. KBDSTB is inverted by an open collector X-OR gate (U10) to force KDATA to a low state at the keyboard interface (Connector #1 pin 11). Feeding the inverted start bit back to the keyboard (via KDATA) effectively generates a trailing "stop" bit for each character transmitted. When the keyboard attempts to transmit the start bit of the next character, it samples the serial data line and finds that it is still low. It will wait until the start bit is asserted true before attempting to clock the remainder of that character data. This feedback interaction prevents keyboard characters from overflowing the PC Video keyboard input buffer.

Assertion of KBDSTB to a high state when either a Light Pen or Keyboard data input is completed also causes an interrupt to occur if selected by the interrupt vector level select header (J4). The selected vector interrupt line is asserted low by transister Q1 which inverts the state of KBDSTB. Additionally, KBDSTB is buffered by U42 to become bit #1 of the status register. This bit may be polled by software as the PC Video data byte available status flag.

Keyboard character data bytes are read by the host processor by performing an input from relative port #12. This asserts KBDENB* active low, which enables the parallel data bits of U19 onto the local General Purpose data bus. Each keyboard character data input read by the host is followed by an output to relative port #11. This asserts KBDCLR* active low, and clears all three register elements (U19, U18 and U13) of the Keyboard/Light Pen data input logic. When U13 is cleared, it releases the keyboard serial data line. This permits the next keyboard character to be

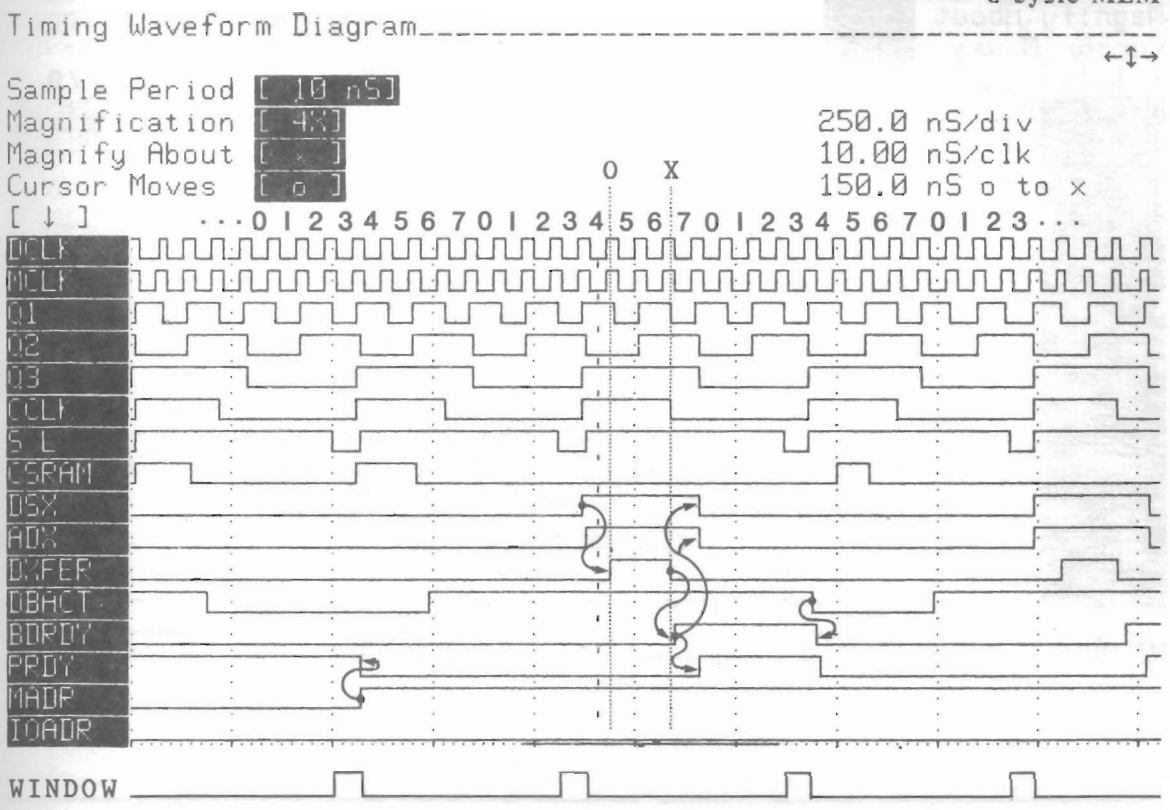
serially transmitted to the PC Video board. The transition of KBDSTB to a low state also releases the vector interrupt line on the S-100 bus. This is the first step of an "interrupt acknowledge" sequence to prepare the system for the next keyboard or light pen interrupt.

Light Pen and Keyboard data inputs are distinguished from each other by the value returned from the keyboard character input operation. Since the keyboard data register is always cleared prior to each data byte input and the light pen input does not alter its value, a light pen input will always return a zero keyboard character byte. Conversely, IBM-PC compatible keyboards do not transmit a keyboard scan character which has a value of zero.

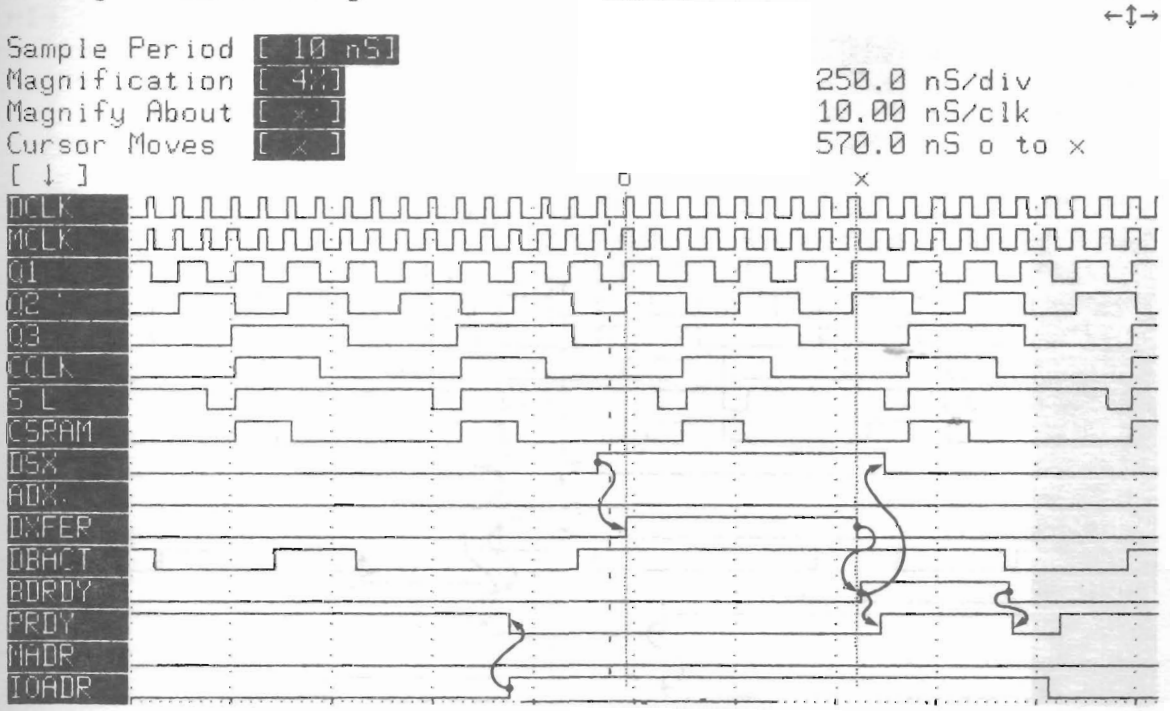
TIMING DIAGRAMS

The following timing diagrams are included here to assist you in understanding the operation of the PC Video board.

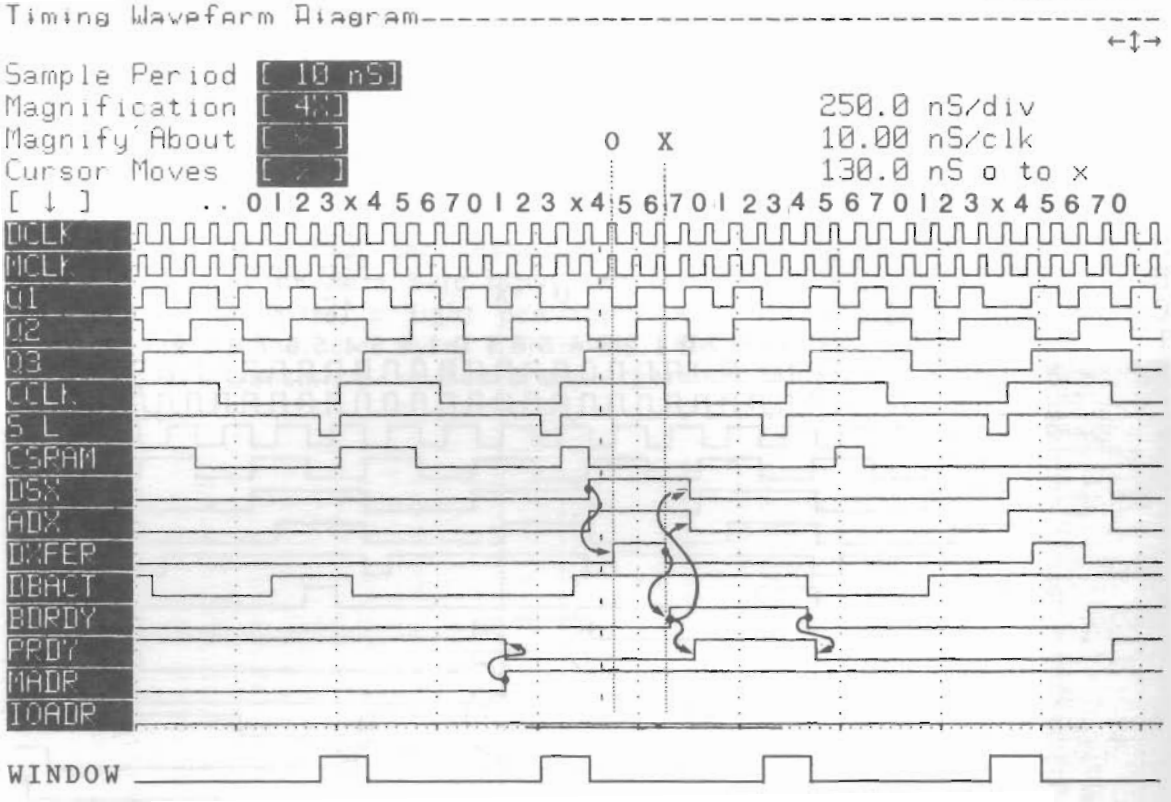
8 cycle MEM



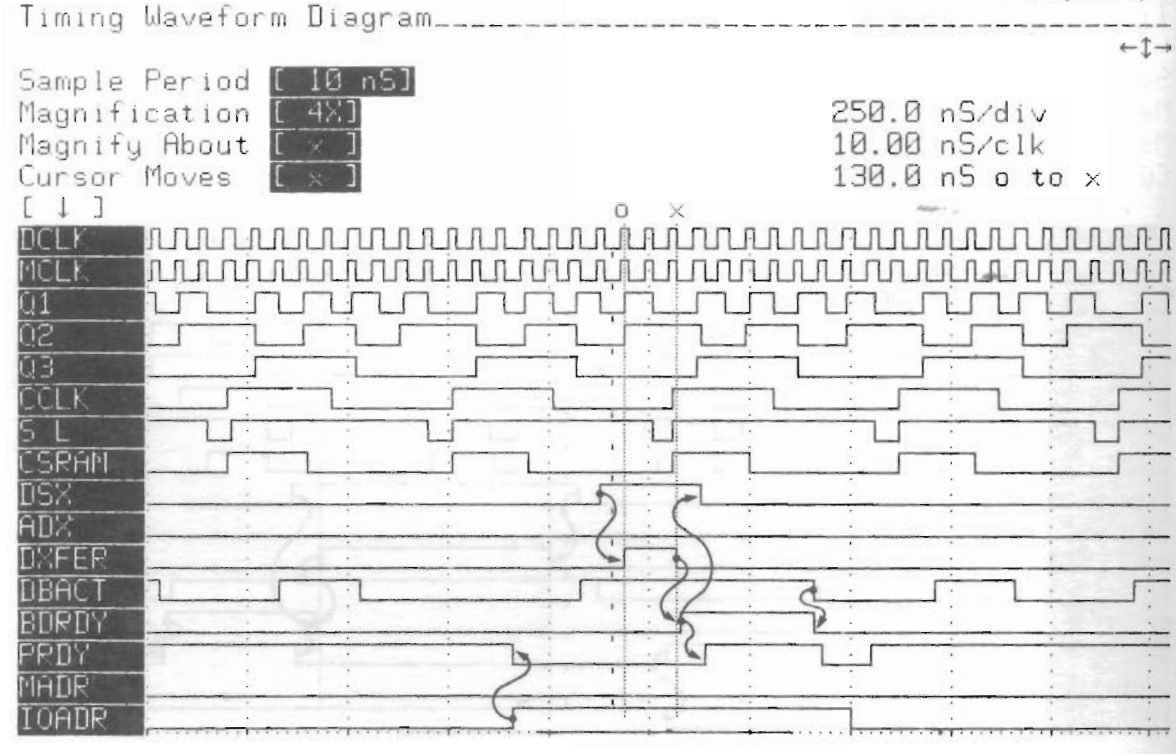
8 cycle I/O (6845)

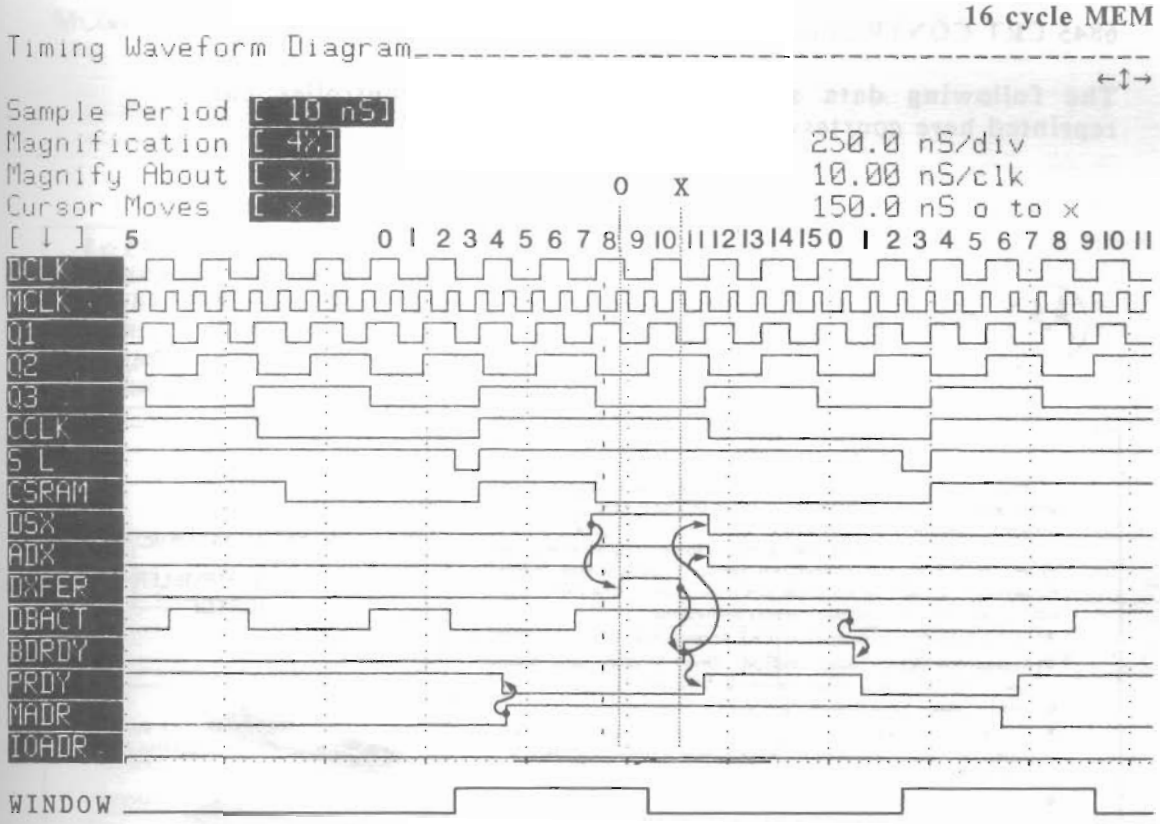


9 cycle MEM



9 cycle I/O




Timing Waveform Diagram
Sample Period [10 nS]
Magnification [4X]
Magnify About [x]
Cursor Moves [x]
250.0 nS/div
10.00 nS/clk
150.0 nS o to x
0 X
[↓] 5 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1 2 3 4 5 6 7 8 9 10 11
DCLK
MCLK
Q1
Q2
Q3
CCLK
S L
CSRAM
DSX
ADX
DXFER
DBACT
BDRDY
PRDY
MADR
IOADR
WINDOW

6845 CRT CONTROLLER CHIP DATA SHEET

The following data sheet on the 6845 CRT controller chip is reprinted here courtesy of Motorola Inc.

MOTOROLA

MC6845 (1.0 MHz) **MC6845☆1** (1.0 MHz)
MC68A45 (1.5 MHz) **MC68A45☆1** (1.5 MHz)
MC68B45 (2.0 MHz) **MC68B45☆1** (2.0 MHz)

CRT CONTROLLER (CRTC)

The MC6845 CRT Controller performs the interface between an MPU and a raster-scan CRT display. It is intended for use in MPU-based controllers for CRT terminals in stand-alone or cluster configurations.

The CRTC is optimized for the hardware/software balance required for maximum flexibility. All keyboard functions, reads, writes, cursor movements, and editing are under processor control. The CRTC provides video timing and refresh memory addressing.

- Useful in Monochrome or Color CRT Applications
- Applications Include "Glass-Teletype," Smart, Programmable, Intelligent CRT Terminals; Video Games; Information Displays
- Alphanumeric, Semi-Graphic, and Full-Graphic Capability
- Fully Programmable Via Processor Data Bus. Timing May Be Generated for Almost Any Alphanumeric Screen Format, e.g., 80 × 24, 72 × 64, 132 × 20
- Single +5 V Supply
- M6800 Compatible Bus Interface
- TTL-Compatible Inputs and Outputs
- Start Address Register Provides Hardware Scroll (by Page, Line, or Character)
- Programmable Cursor Register Allows Control of Cursor Format and Blink Rate
- Light Pen Register
- Refresh (Screen) Memory May be Multiplexed Between the CRTC and the MPU Thus Removing the Requirements for Line Buffers or External DMA Devices
- Programmable Interlace or Non-Interlace Scan Modes
- 14-Bit Refresh Address Allows Up to 16K of Refresh Memory for Use in Character or Semi-Graphic Displays
- 5-Bit Row Address Allows Up to 32 Scan-Line Character Blocks
- By Utilizing Both the Refresh Addresses and the Row Addresses, a 512K Address Space is Available for Use in Graphics Systems
- Refresh Addresses are Provided During Retrace, Allowing the CRTC to Provide Row Addresses to Refresh Dynamic RAMs
- Programmable Skew for Cursor and Display Enable (DE)
- Pin Compatible with the MC6835

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}*	−0.3 to +7.0	V
Input Voltage	V_{in}*	−0.3 to +7.0	V
Operating Temperature Range MC6845, MC68A45, MC68B45 MC6845C, MC68A45C, MC68B45C	T_A	T_L to T_H 0 to 70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

THERMAL CHARACTERISTICS

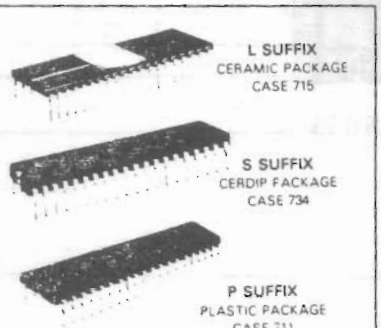
Characteristic	Symbol	Value	Rating
Thermal Resistance Plastic Package Cerdip Package Ceramic Package	θ_{JA}	 100 60 50	°C/W

*This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \le (V_{in}$ or $V_{out}) \le V_{CC}$.

MOS

(N-CHANNEL, SILICON-GATE)

CRT CONTROLLER (CRTC)



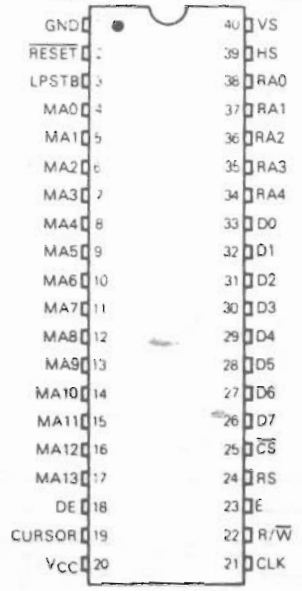
L SUFFIX
CERAMIC PACKAGE
CASE 715

S SUFFIX
CERDIP PACKAGE
CASE 734

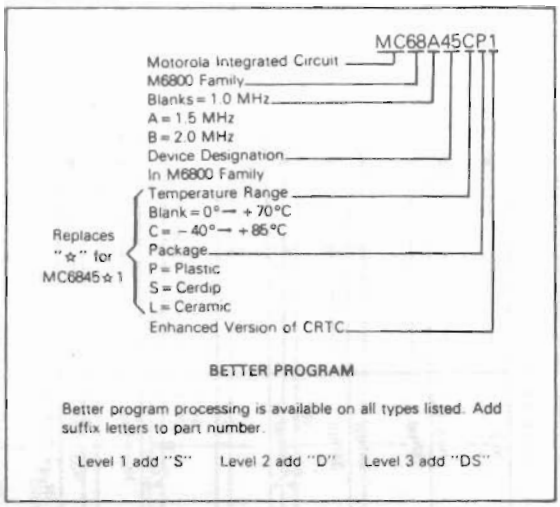
P SUFFIX
PLASTIC PACKAGE
CASE 711

☆ = Package Suffix

FIGURE 1 – PIN ASSIGNMENTS

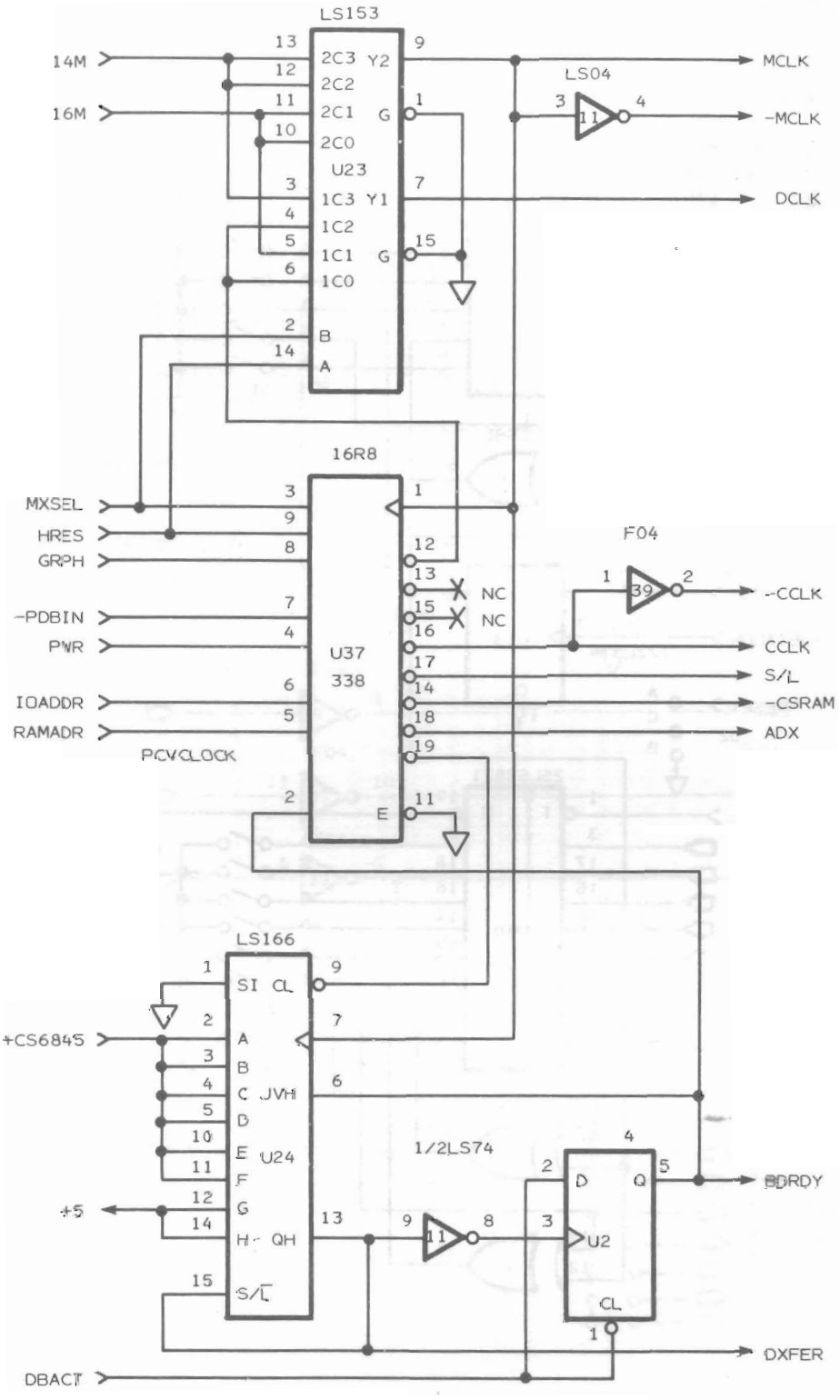


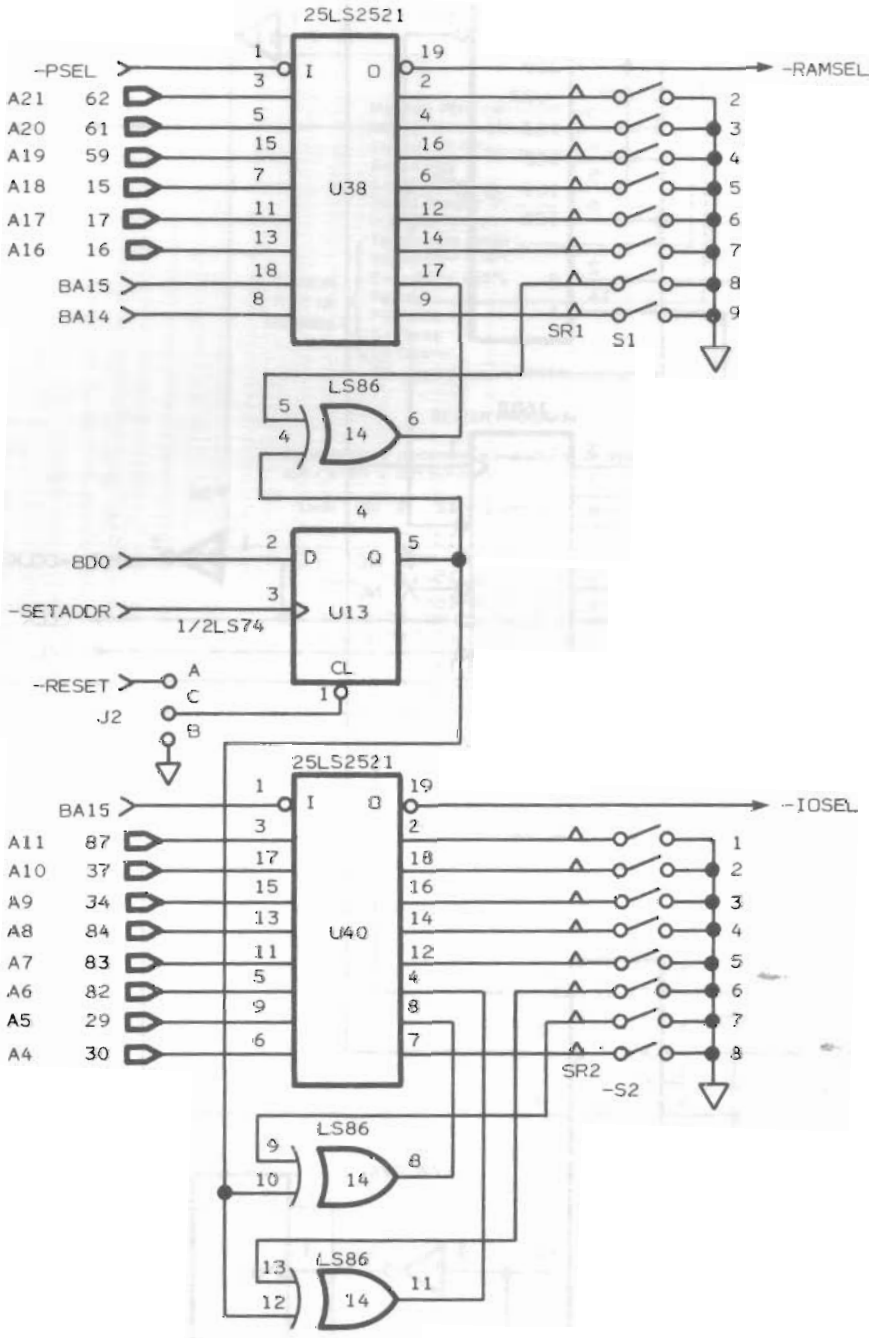
ORDERING INFORMATION



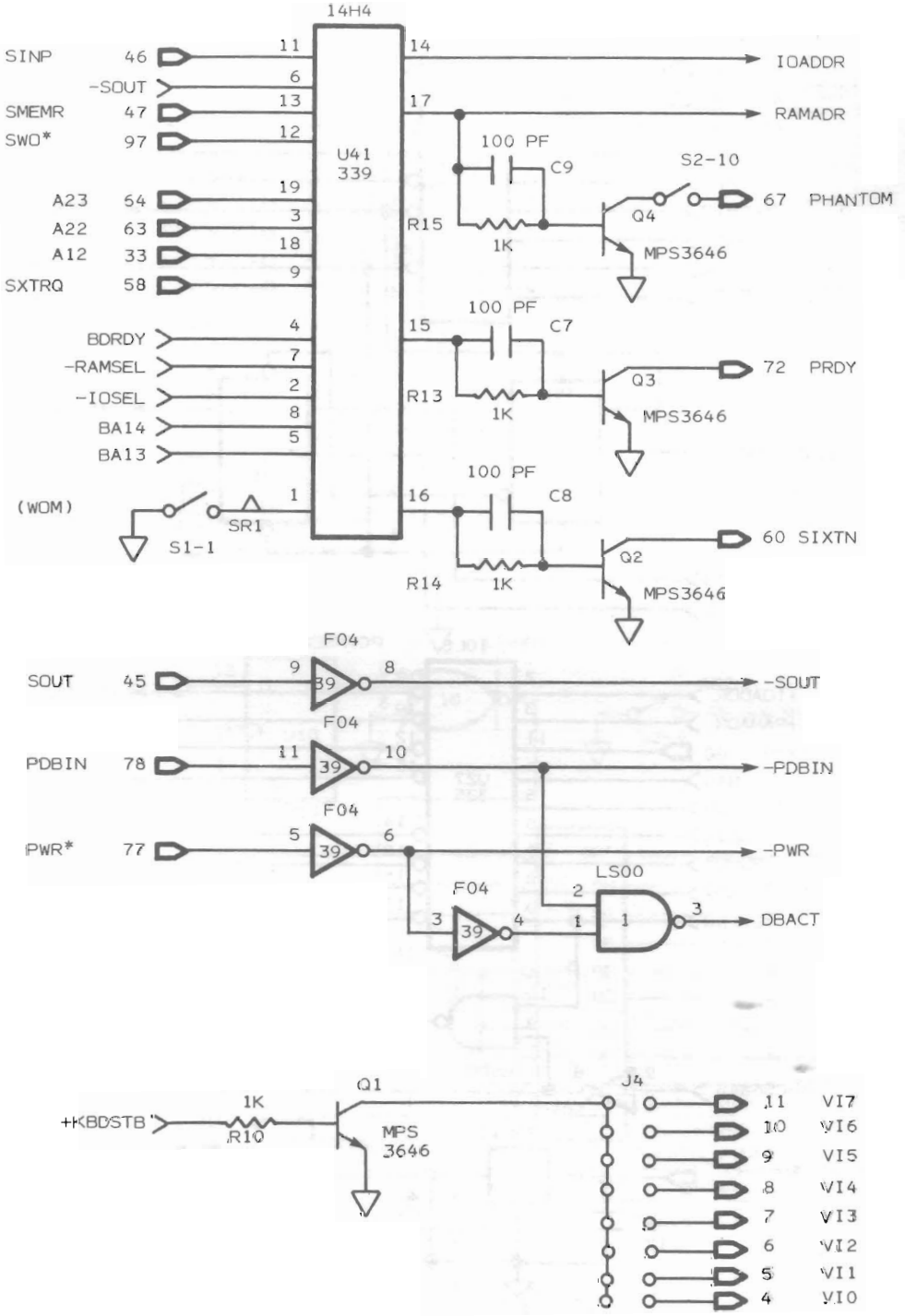
Level 1 "S" = 10 Temp Cycles ← /→ 25 to 150°C);
Hi Temp testing at T_A max.
Level 2 "D" = 168 Hour Burn-in at 125°C
Level 3 "DS" = Combination of Level 1 and 2

LOGIC DIAGRAM

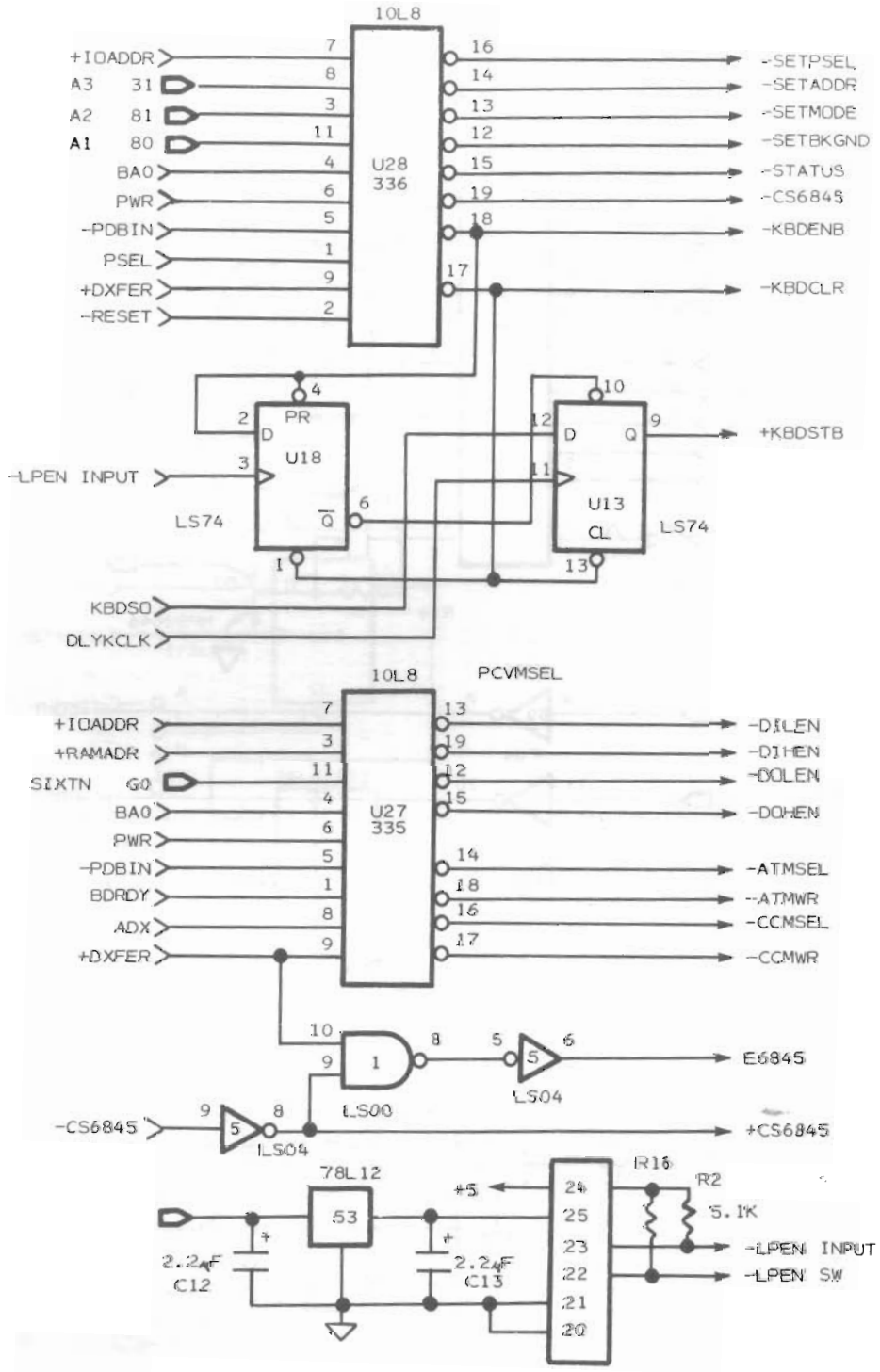




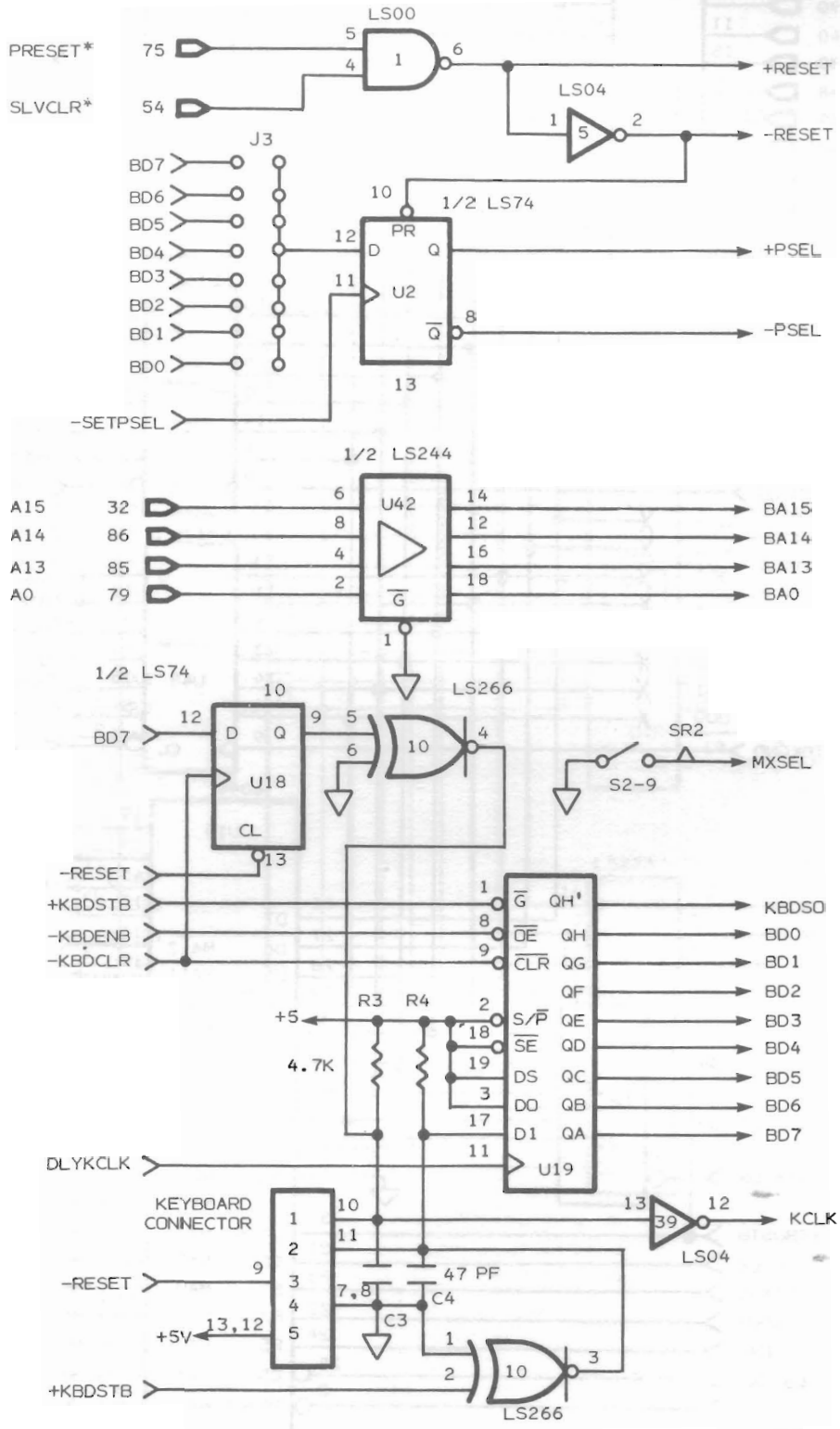
PC VIDEO
229D
VIASYN ©1985



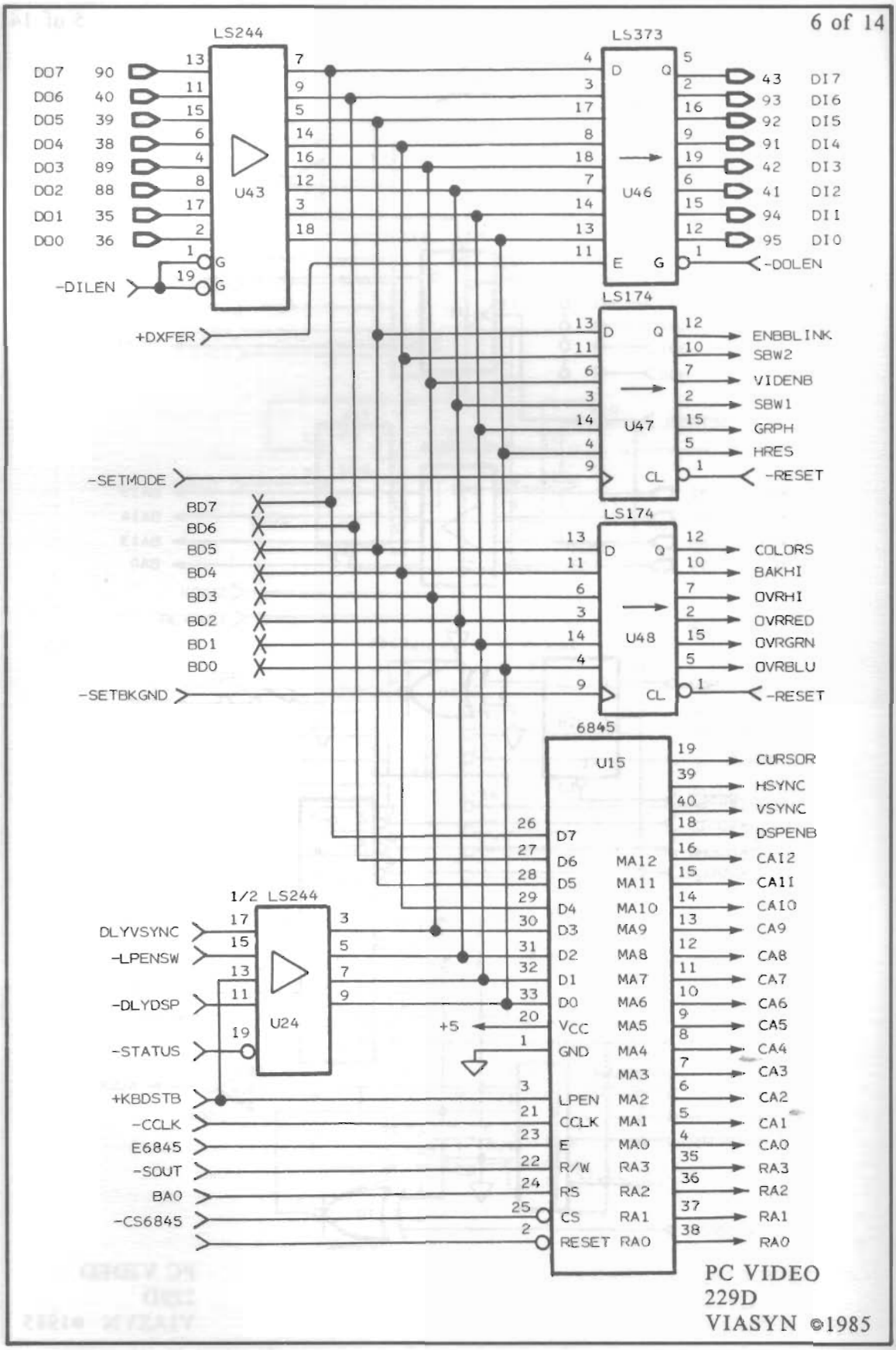
PC VIDEO
229D
VIASYN ©1985

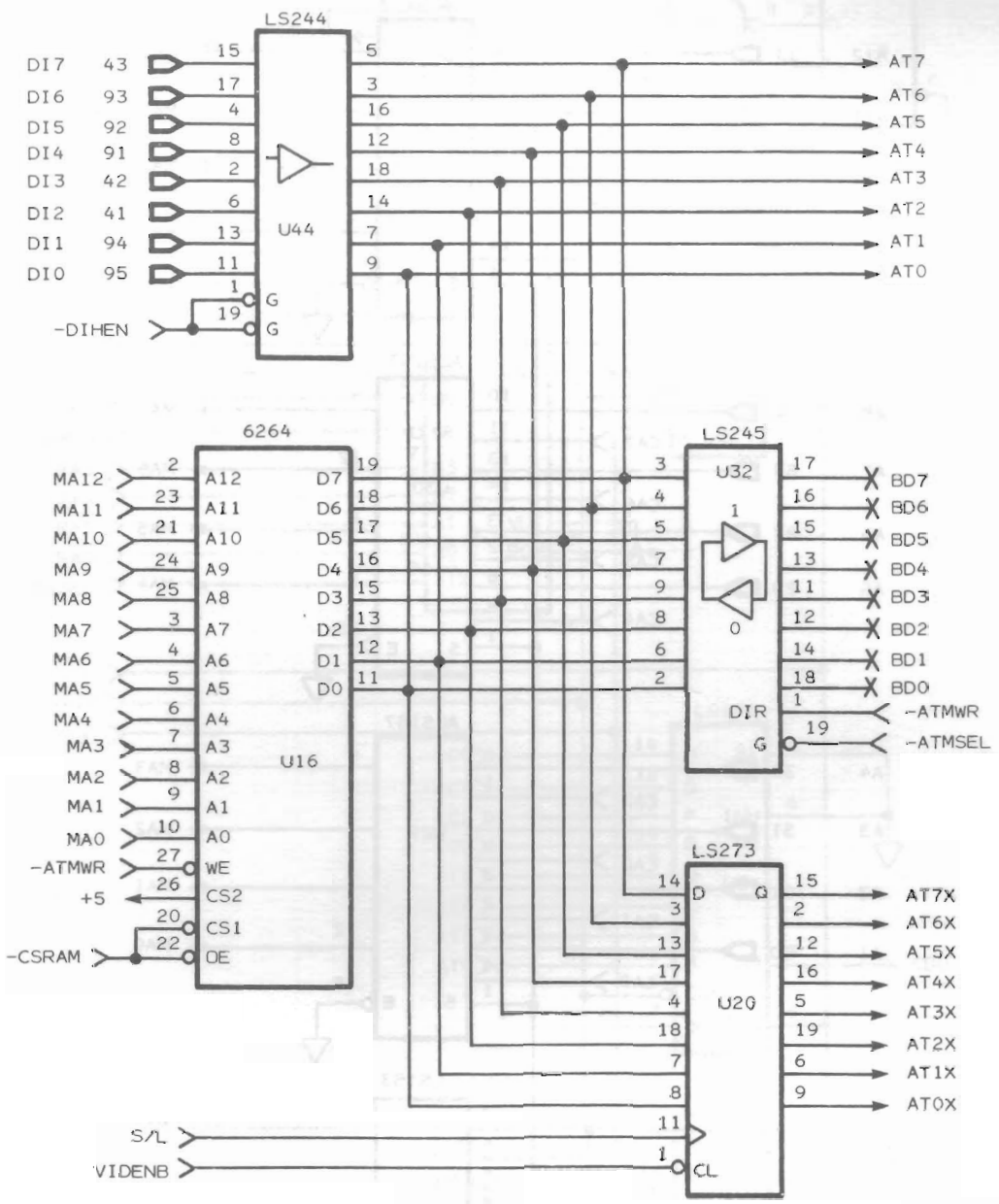


PC VIDEO
229D
VIASYN ©1985

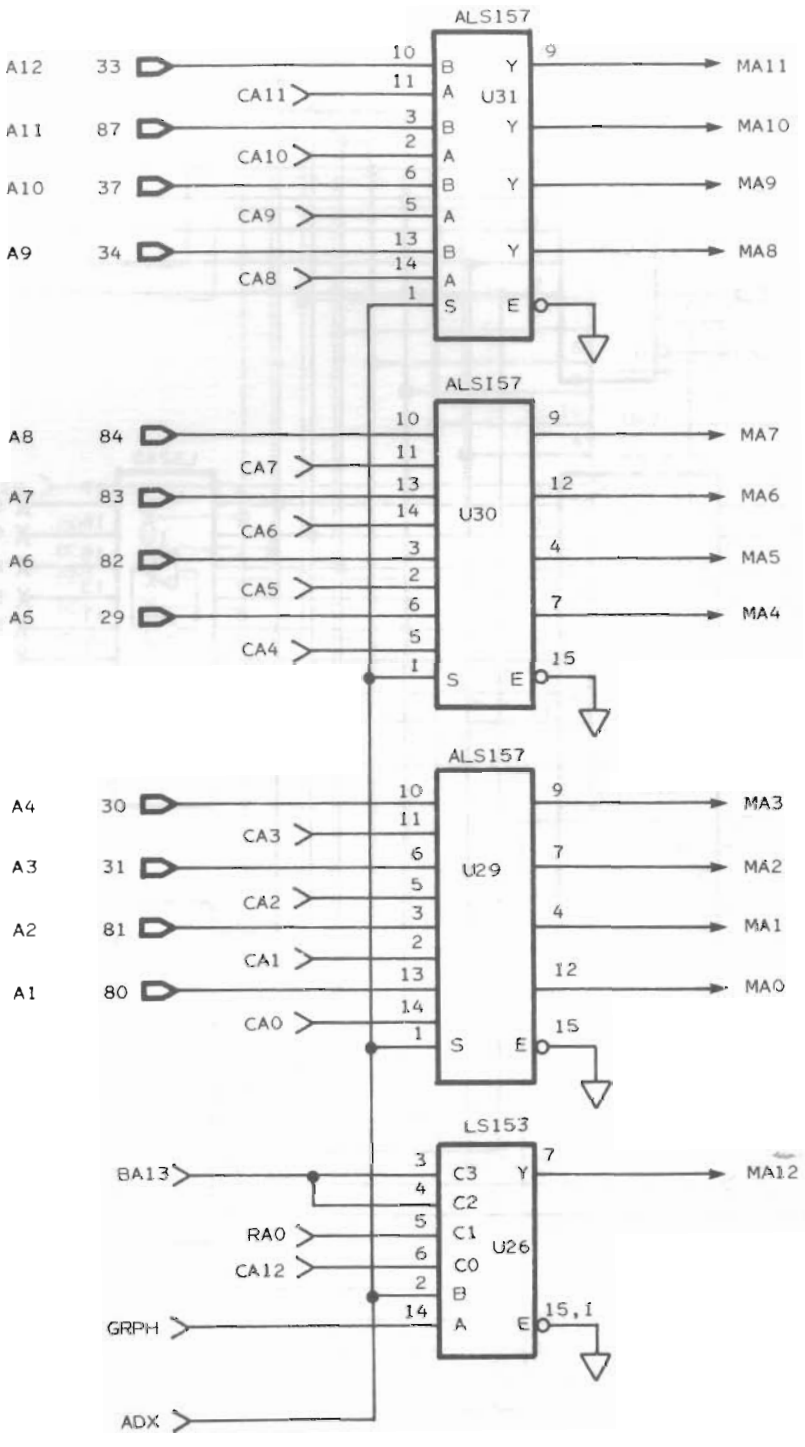


PC VIDEO
229D
VIASYN ©1985

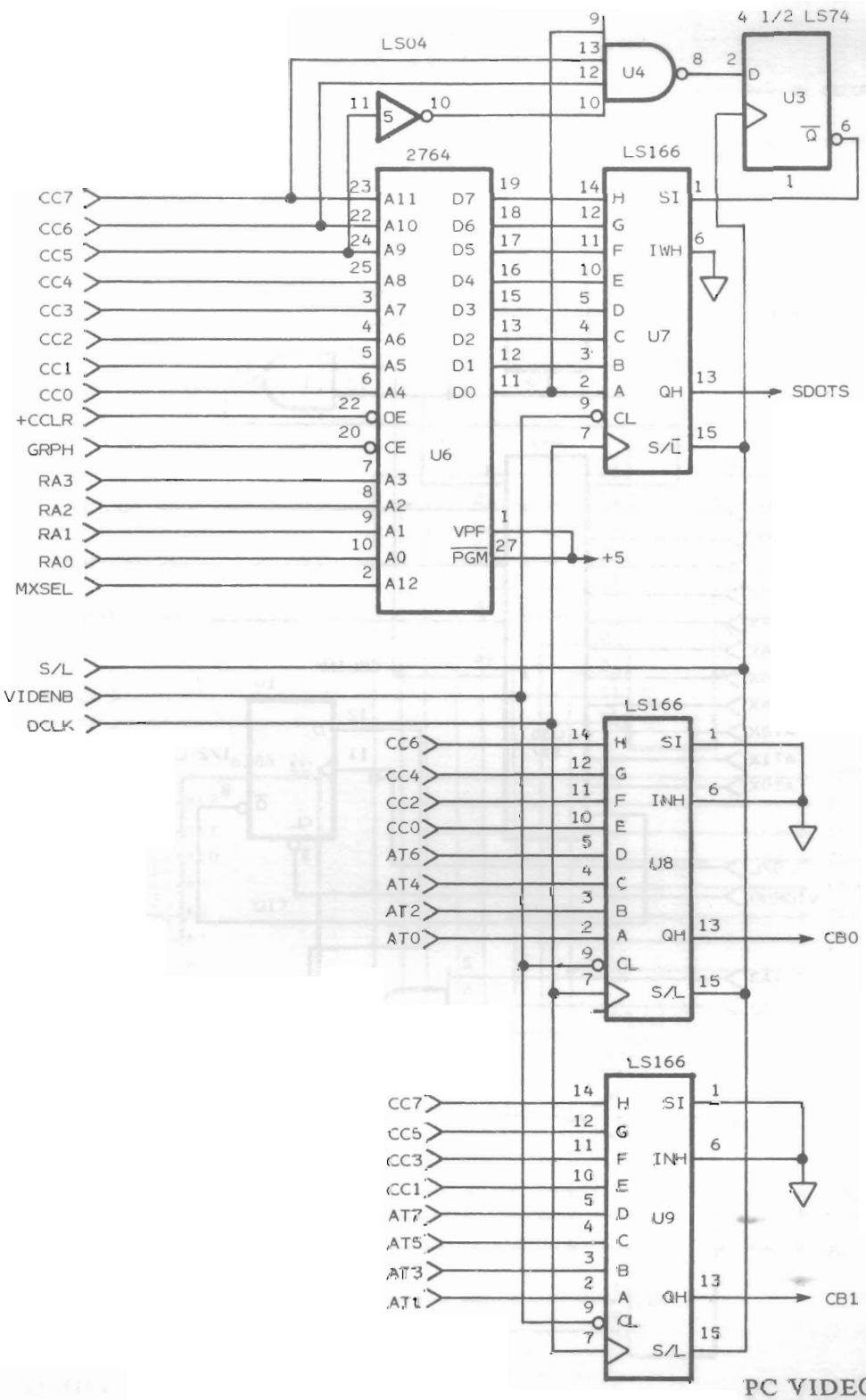

LS244
U43
DO7 90
DO6 40
DO5 39
DO4 38
DO3 89
DO2 88
DO1 35
DO0 36
-DILEN
LS373
U46
DI7 43
DI6 93
DI5 92
DI4 91
DI3 42
DI2 41
DI1 94
DI0 95
-DOLEN
+DXFER
LS174
U47
ENBBLINK
SBW2
VIDENB
SBW1
GRPH
HRES
-RESET
-SETMODE
BD7
BD6
BD5
BD4
BD3
BD2
BD1
BD0
LS174
U48
COLORS
BAKHI
OVRHI
OVRRED
OVRGRN
OVRBLU
-RESET
-SETBKGND
6845
U15
CURSOR
HSYNC
VSYNC
DSPENB
CA12
CA11
CA10
CA9
CA8
CA7
CA6
CA5
CA4
CA3
CA2
CA1
CA0
RA3
RA2
RA1
RA0
1/2 LS244
U24
DLYVSYNC
-LPENSW
-DLYDSP
-STATUS
+5
+KBDSTB
-CCLK
E6845
-SOUT
BA0
-CS6845
PC VIDEO
229D
VIASYN ©1985

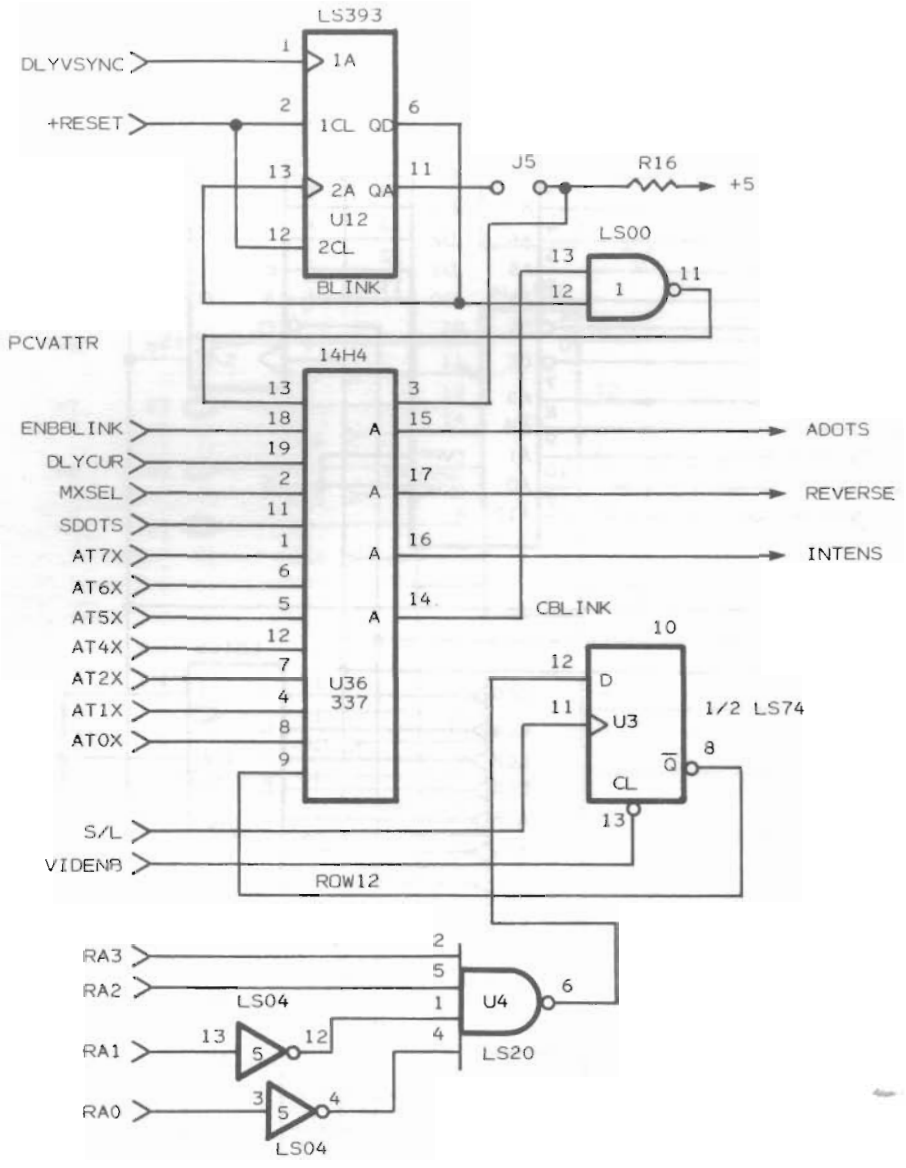


PC VIDEO
229D
VIASYN ©1985

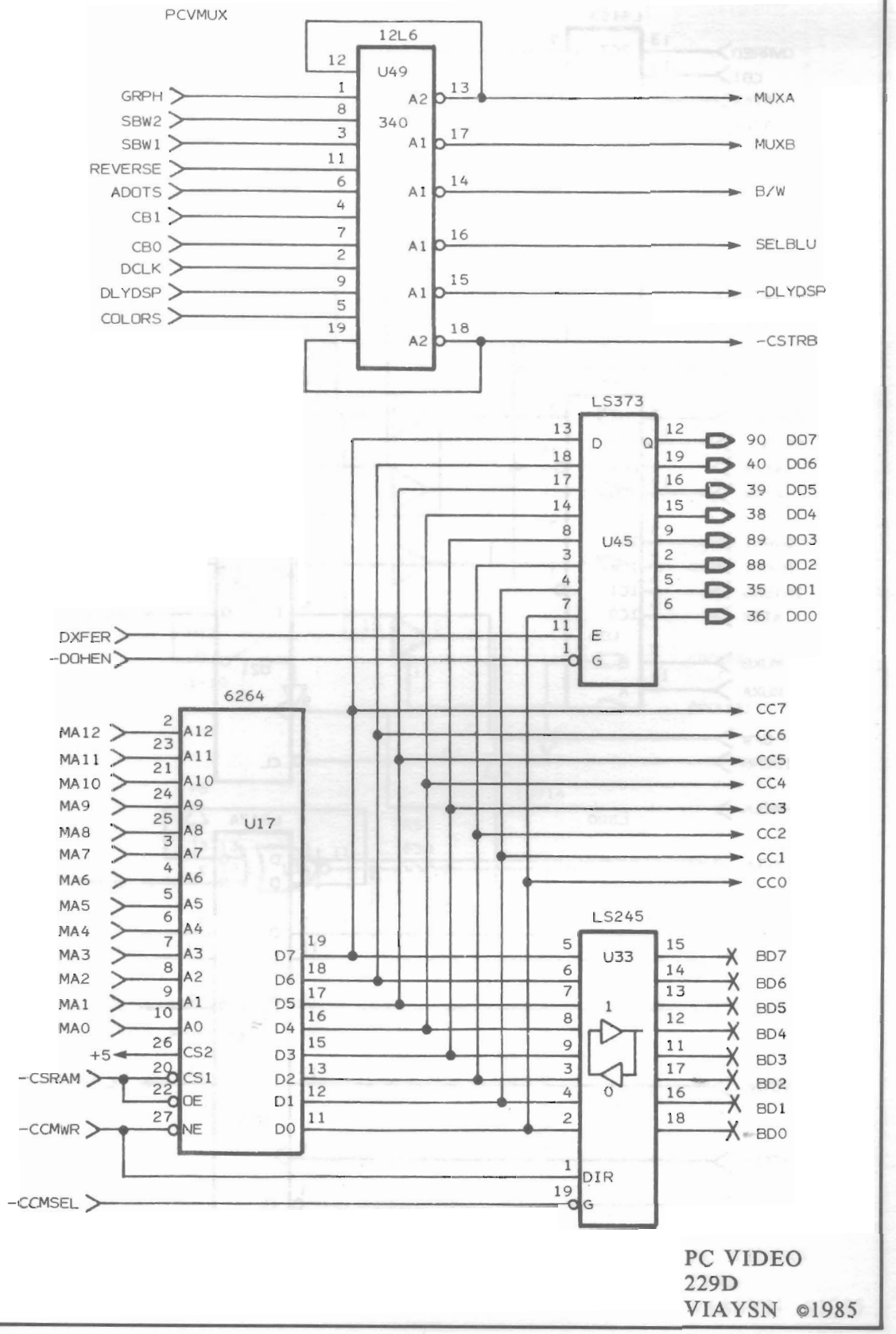


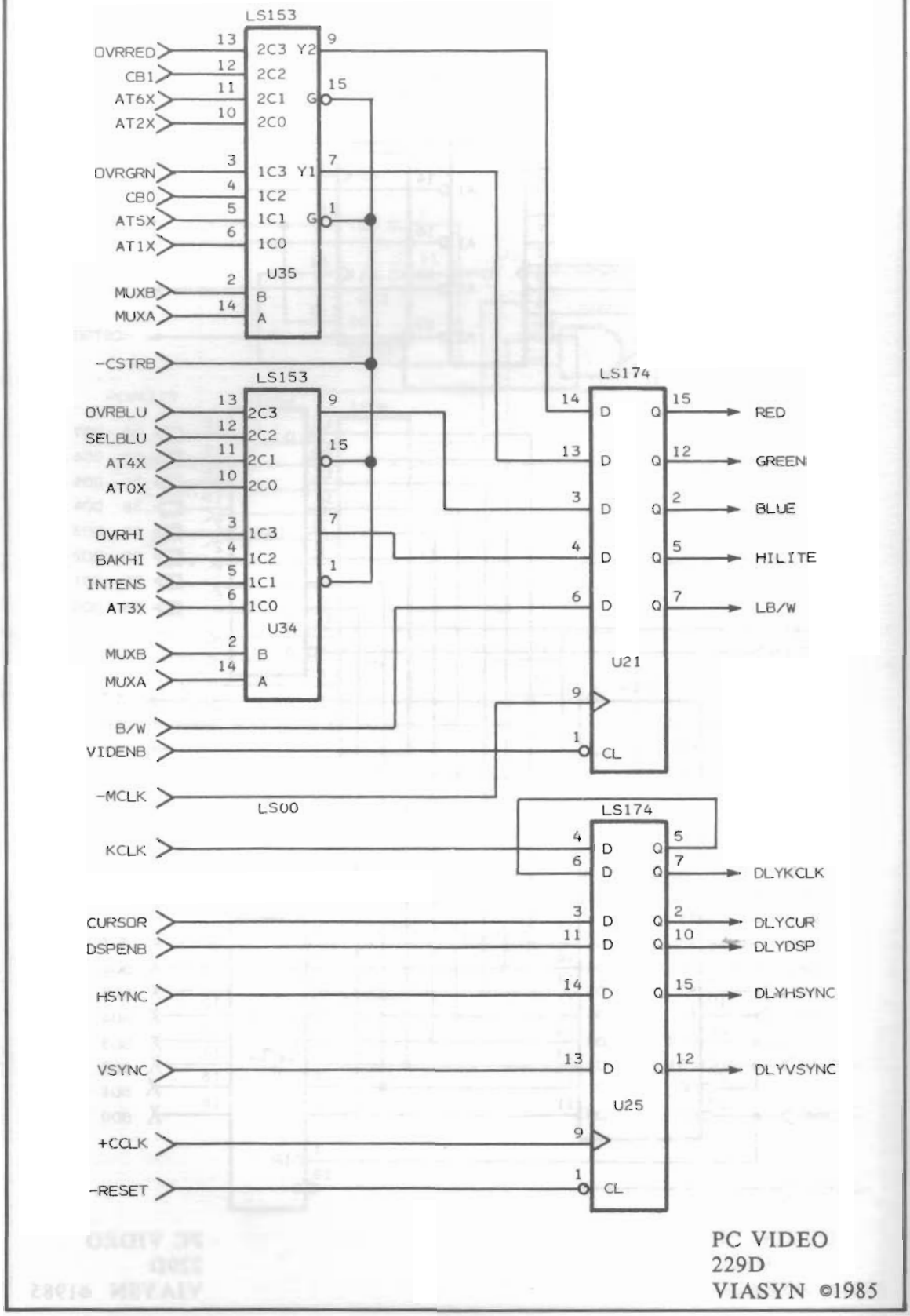
PC VIDEO
229D
VIASYN ©1985

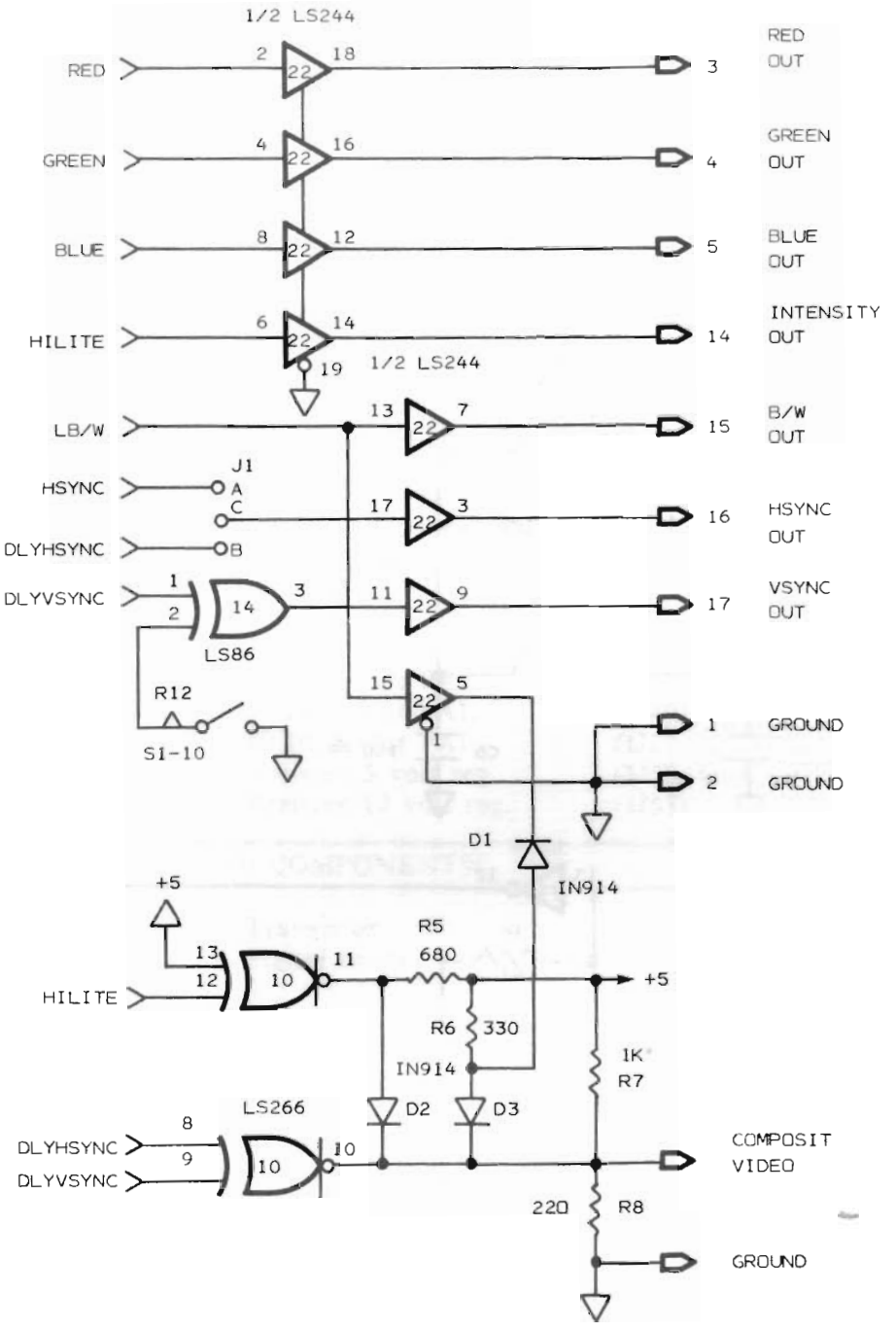

LS04
U4
4 1/2 LS74
U3
2764
U6
LS166
U7
LS166
U8
LS166
U9
CC7
CC6
CC5
CC4
CC3
CC2
CC1
CC0
+CCLR
GRPH
RA3
RA2
RA1
RA0
MXSEL
S/L
VIDENB
DCLK
SDOTS
CB0
CB1
+5
PC VIDEO
229D
VIASYN ©1985



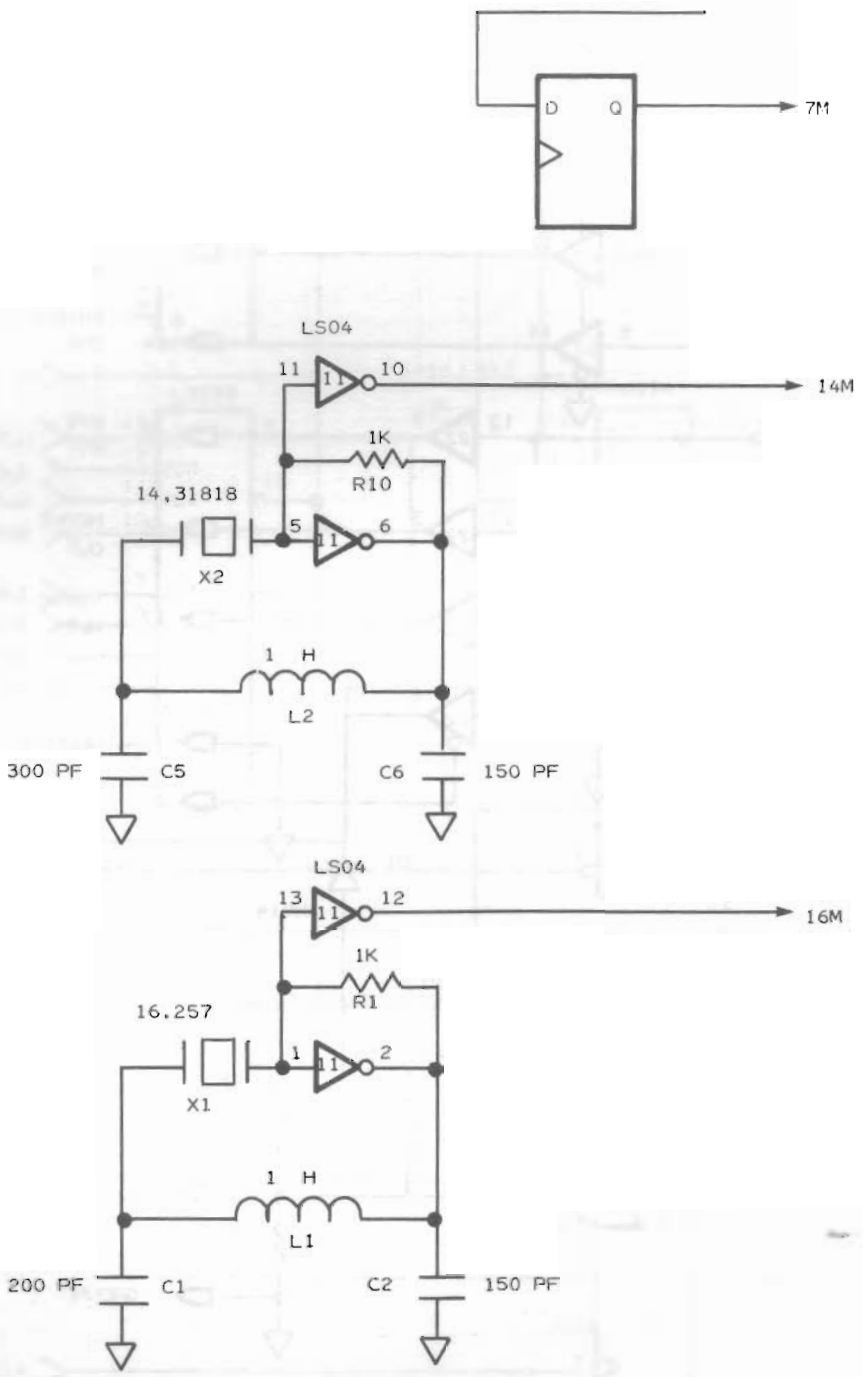
PC VIDEO
229D
VIASYN ©1985


PCVMUX
12L6
U49
340
GRPH
SBW2
SBW1
REVERSE
ADOTS
CB1
CB0
DCLK
DLYDSP
COLORS
MUXA
MUXB
B/W
SELBLU
-DLYDSP
-CSTRB
LS373
U45
DXFER
-DOHEN
DO7
DO6
DO5
DO4
DO3
DO2
DO1
DO0
6264
U17
MA12
MA11
MA10
MA9
MA8
MA7
MA6
MA5
MA4
MA3
MA2
MA1
MA0
+5
-CSRAM
-CCMWR
CC7
CC6
CC5
CC4
CC3
CC2
CC1
CC0
LS245
U33
DIR
-CCMSEL
BD7
BD6
BD5
BD4
BD3
BD2
BD1
BD0
PC VIDEO
229D
VIAYSN ©1985


LS153
OVRRED
CB1
AT6X
AT2X
2C3 Y2
2C2
2C1 G
2C0
OVRGRN
CB0
AT5X
AT1X
1C3 Y1
1C2
1C1 G
1C0
U35
MUXB
MUXA
B
A
-CSTRB
LS153
OVRBLU
SELBLU
AT4X
AT0X
OVRHI
BAKHI
INTENS
AT3X
2C3
2C2
2C1
2C0
1C3
1C2
1C1
1C0
U34
MUXB
MUXA
B
A
LS174
D
Q
RED
GREEN
BLUE
HILITE
LB/W
U21
B/W
VIDENB
CL
-MCLK
LS00
LS174
KCLK
DLYKCLK
CURSOR
DSPENB
DLYCUR
DLYDSP
HSYNC
DLYHSYNC
VSYNC
DLYVSYNC
U25
+CCLK
-RESET
CL
PC VIDEO
229D
VIASYN ©1985



PC VIDEO
229D
VIASYN ©1985



PC VIDEO
229D
VIASYN ©1985

PARTS LIST

INTEGRATED CIRCUITS:

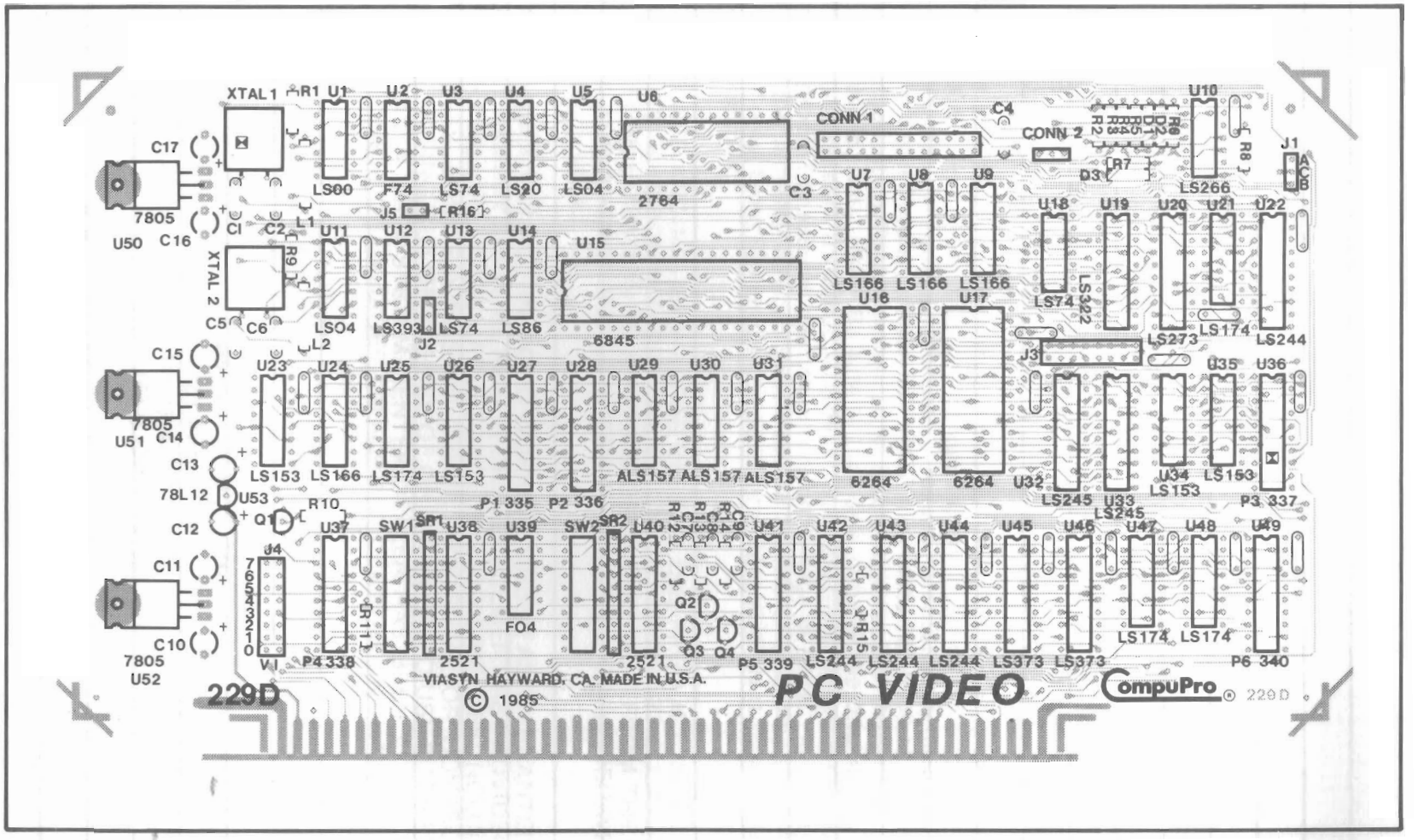
QTY	Description		
1	74LS00	Quad 2 input NAND	(U01)
2	74LS04	Hex inverter	(05, U11)
1	74F04	Hex inverter	(U39)
1	74LS20	Dual 4 input AND	(U04)
4	74LS74	Dual D-type FF	(U02, U03, U13, U18)
1	74LS86	Quad 2 input XOR	(U14)
3	74LS153	Quad 2 line to 1 line	(U26, U34, U35)
1	74F153	Quad 2 line to 1 line	(U23)
3	74ALS157	Dual 4 line to 1 line	(U29, U30, U31)
4	74LS166	8-bit shift register	(U07, U08, U09, U24)
4	74LS174	Hex D-type flip-flop	(U21, U25, U47, U48)
4	74LS244	Octal tri-state buffer	(U22, U42, U43, U44)
2	74LS245	Octal tri-state buffer	(U32, U33)
1	74LS266	Quad 2 input XNOR - OC	(U10)
1	74LS273	Octal D-type flip-flop	(U20)
1	74LS322	Octal shift register	(U19)
2	74LS373	Octal transparent latch	(U45, U46)
1	74LS393	Dual decade counter	(U12)
2	25LS2521	Octal comparator	(U38, U40)
1	2764-2	8K x 8 EPROM (200 nsec.)	(U6)
1	16R8	16 In, 8 Out register PAL	(U37)
2	14H4	14 In, 4 Out PAL	(U36, U41)
1	12L6	12 In, 6 Out PAL	(U49)
2	10L8	10 In, 8 Out PAL	(U27, U28)
3	7805	Positive 5 volt reg.	(U50, U51, U52)
1	78L12	Positive 12 volt reg.	(U53)

OTHER ELECTRICAL COMPONENTS:

4	MPS3646	Transistor	(Q1, Q2, Q3, Q4)
3	1N914	Signal diode	(D1, D2, D3)
2	5.1K Ohm	Sip resistor	(SR1, SR2)
1	220 Ohm	Resistor	(R8)
1	330 Ohm	Resistor	(R6)
1	680 Ohm	Resistor	(R5)
7	1K Ohm	Resistor	(R1, R7, R9, R10, R13, R14, R15)
6	4.7KOhm	Resistor	(R2,R3, R4, R11, R16, R17)
5	1.5uF	Tant cap at 20V	(C10, C12, C13, C14, C16)
3	1.5uF	Tant cap at 10V	(C11, C15, C17)
2	47pF	Disc cap	(C3, C4)
3	100pF	Silver mica cap	(C7, C8, C9)
2	150pF	Silver mica cap	(C2, C6)
1	220pF	Silver mica cap	(C1)
1	270pF	Silver mica cap	(C5)
38		By-pass caps	(all unmarked)
1		14.31818 MHz crystal	(X2)
1		16.257 MHz crystal	(X1)
2		1.0 uH Inductor	(L1, L2)

MECHANICAL COMPONENTS:

QTY	Description	QTY	Description
2	10 position dip switch	2	8 long double row pins
12	14 pin sockets	1	3 long single row pins - right angle
15	16 pin sockets	1	5 long double row pins - right angle
18	20 pin sockets	1	7 long double row pins - right angle
3	28 pin sockets	3	Heatsinks with spacer (.180 hole)
1	40 pin socket	3	6-32 x 5/16" screws
1	2 long single row pins	1	PCB #229B
2	3 long single row pins	2	Card ears



COMPONENT LAYOUT