As you are now the owner of this document which should have come to you for free, please consider making a donation of £1 or more for the upkeep of the (Radar) website which holds this document. I give my time for free, but it costs me money to bring this document to you. You can donate here https://blunham.com/Misc/Texas

Many thanks.

Please do not upload this copyright pdf document to any other website. Breach of copyright may result in a criminal conviction.

This Acrobat document was generated by me, Colin Hinson, from a document held by me. I requested permission to publish this from Texas Instruments (twice) but received no reply. It is presented here (for free) and this pdf version of the document is my copyright in much the same way as a photograph would be. If you believe the document to be under other copyright, please contact me.

The document should have been downloaded from my website https://blunham.com/, or any mirror site named on that site. If you downloaded it from elsewhere, please let me know (particularly if you were charged for it). You can contact me via my Genuki email page: https://www.genuki.org.uk/big/eng/YKS/various?recipient=colin

You may not copy the file for onward transmission of the data nor attempt to make monetary gain by the use of these files. If you want someone else to have a copy of the file, point them at the website. (https://blunham.com/Misc/Texas). **Please do not point them at the file itself as it may move or the site may be updated.**

It should be noted that most of the pages are identifiable as having been processed by me.

I put a lot of time into producing these files which is why you are met with this page when you open the file.

If you find missing pages, pages in the wrong order, anything else wrong with the file or simply want to make a comment, please drop me a line (see above).

It is my hope that you find the file of use to you.

Colin Hinson
In the village of Blunham, Bedfordshire.

COLIN HINSON.

The Engineering Staff of
TEXAS INSTRUMENTS LIMITED
Semiconductor Group

TMS 9918
VIDEO DISPLAY PROCESSOR
PRELIMINARY DATA MANUAL

PRELIMINARY
MARCH, 1980

TEXAS INSTRUMENTS
LIMITED

IMPORTANT NOTICES

Texas Instruments reserves the right to make changes at any time in order to improve design and to supply the best product possible.

TI cannot assume any responsibility for any circuits shown or represent that they are free from patent infringement.

If you have any technical questions concerning the TMS 9918, consult your local TI systems engineer or the HOTLINE at (713) 776-6632

1.0 INTRODUCTION

1.1 DESCRIPTION

This specification defines the functional, electrical, and timing requirements for the TMS 9918 Video Display Processor (VDP). The VDP is an N-Channel MOS LSI device used in video systems where data display on a raster-scanned home color television is required. The VDP generates all necessary video, control, and synchronization signals, and additionally controls the storage, retrieval, and refresh of display data in the dynamic screen refresh memory. The interfaces to the microprocessor, refresh memory, and the TV are defined so as to require a minimum of additional electronics.

The VDP has three video color display modes: pattern graphics, multicolor, and text. The text mode provides twenty-four 40-character rows in two colors and is intended for computer oriented displays. The multicolor mode provides an unrestricted 64 × 48 dot display in 15 colors. The pattern graphics mode provides a 256 × 192 pixel display and is primarily intended for use in game and educational applications. All 15 colors are also available in the pattern graphics mode.

The video display consists of thirty-five display planes: external video, backdrop, pattern, and sprites 0-31. The planes are vertically stacked with the external video being the bottom or innermost plane. The backdrop plane is the next plane, and the 32 sprite planes are the top planes. A sprite is a special animation pattern used in the pattern graphics and multicolor modes.

The VDP uses 4K or 16K RAS/CAS memories for storage of the display parameters and for microprocessor temporary data storage.

1.2 FEATURES

- Single-Chip Interface To Color TV's (Excluding RAM and RF Modulator)
- 256 × 192 Resolution on TV Screen; 15 Unique Colors and Transparent
- General 8-bit Bidirectional Interface to CPU
- Direct Wiring to 4K or 16K RAS/CAS Dynamic RAM Memories
- Automatic and Transparent Refresh of RAM's
- General Purpose Memory Mapped CPU Interface
- External Video Input Capability
- Easy to Use for Text, Graphics, Animation
- Unique Planar Representation Allows 3-D Simulation
- Standard 40-Pin Plastic Package
- N-Channel Silicon-Gate Technology

1.3 TYPICAL APPLICATIONS

- Color Computer Terminals
- Home Computers

- Drafting/Design Aids
- Teaching Aids
- Industrial Process Monitoring
- Home Educational Systems
- Animation Aids

The following example of a typical application may help introduce the user to the TMS 9918 VDP. Figure 1 is a block diagram of a typical application. Each of the ideas presented below is described more fully in later sections of this manual.

The VDP basically has three interfaces: CPU, color television, and dynamic refresh RAM (VRAM) whose contents define the TV image. The TMS 9918 VDP also has 8 write-only registers and a read-only status register.

The VDP communicates with the CPU via an 8-bit bidirectional data bus. Three control lines, decoded off the CPU address and enable lines, determine interpretation of the bus. Through the bus, the CPU can write to VRAM, read from VRAM, write to VDP registers, and read the VDP status. The VDP also generates an interrupt signal after every refresh of the TV display, which may be useful to the CPU.

The dynamic RAM interface consists of direct wiring of 8 4K × 1 or 8 16K × 1 dynamic RAS/CAS-type RAMs to the TMS 9918 VDP.

The interface to the TV can consist of wiring of the VDP's Composite Video Output pin (suitably buffered) to the input of a color or black-and-white monitor, or may use an appropriate RF modulator in order to feed the signal into a TV's antenna terminals.

The VDP can be in one of three modes of operation, each of which affect the manner in which the VRAM is mapped onto the television screen. In Pattern mode, characters are mapped onto the screen in 8 × 8 pixel blocks, yielding 24 lines of 32 blocks (or "pattern positions") each. In Text mode, there are 24 lines of 40 blocks, each of which is 6 × 8 pixels. In Multicolor mode, there are 48 lines of 64 blocks, each of which is composed of 4 × 4 pixels, all of one solid color. In addition to these, objects call Sprites can be superimposed onto the television image. Furthermore, signals entering the VDP through the External Video Input can be used as a background to VDP-generated images.

1.4 DEFINITIONS

The following definitions will be useful in understanding the use of the TMS 9918 VDP:

pixel — the smallest point on the TV screen that can be independently controlled;

NTSC — National Television Standards Committee which specifies the television signal standard for the USA;

VRAM — Video RAM; refers to the dynamic RAMs that connect to the VDP and whose contents define the TV image.

sprite — an object whose pattern is relative to a specified X, Y coordinate and whose position can therefore be controlled by that coordinate with a positional resolution of one pixel.

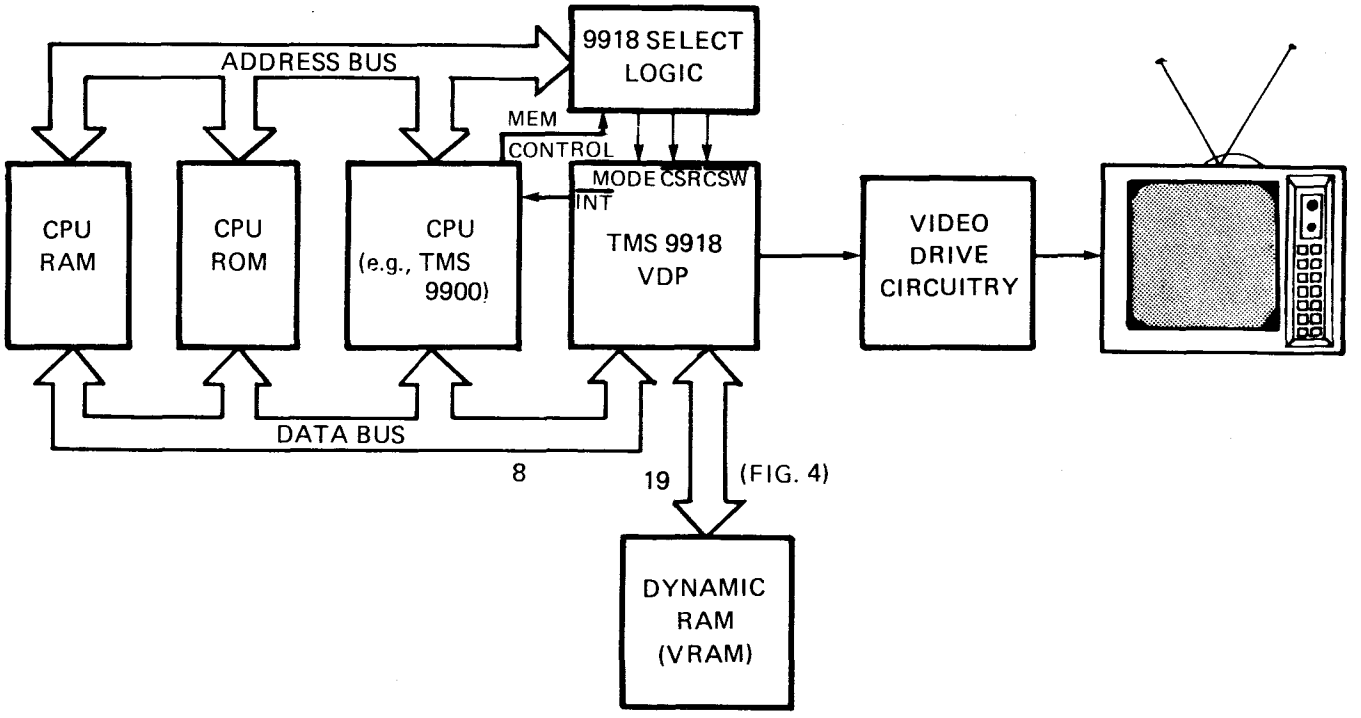


FIGURE 1. SYSTEM BLOCK DIAGRAM

2.0 ARCHITECTURE

The TMS 9918 Video Display Processor (VDP) is designed to provide a simple interface between a microprocessor and a raster-scanned color television. Shown in Figure 2 is a block diagram of the major portions of the VDP architecture. Described below are details of the various interfaces to the VDP: CPU, VRAM, and color television.

2.1 CPU INTERFACE

The VDP interfaces to the CPU using an 8-bit bidirectional data bus, three control lines, and an interrupt as shown in Figure 1. Through this interface the CPU can do 4 operations: Write data bytes to VRAM, Read data bytes from VRAM, Write to one of the 8 VDP write-only registers, and Read the VDP Status register. Each of these operations requires one or more data transfers to take place over the CPU/VDP data bus interface. The interpretation of the data transfer is determined by the 3 control lines of the VDP. It should be noted that the CPU can communicate with the VDP simultaneously and asynchronously with the VDP's TV screen refresh operations. The VDP takes care of memory management and allows periodic windows of CPU access to VRAM even in the middle of a raster scan.

2.1.1 CPU Interface Control Signals

The type and direction of data transfers are controlled by the $\overline{\text{CSW}}$, $\overline{\text{CSR}}$, and MODE inputs. $\overline{\text{CSW}}$ is the CPU-to-VDP write select; when it is active (low), the 8-bits on D0-D7 are strobed into the VDP. $\overline{\text{CSR}}$ is the CPU-from-VDP read select; when it is active (low), the VDP outputs 8-bits on D0-D7 to the CPU. $\overline{\text{CSW}}$ and $\overline{\text{CSR}}$ should never be simultaneously low. If both are low, the VDP outputs data on D0-D7 and latches in invalid data.

MODE determines the source or destination of a read or write data transfer. MODE is normally tied to a CPU low order address line. (A14 for TMS 9900).

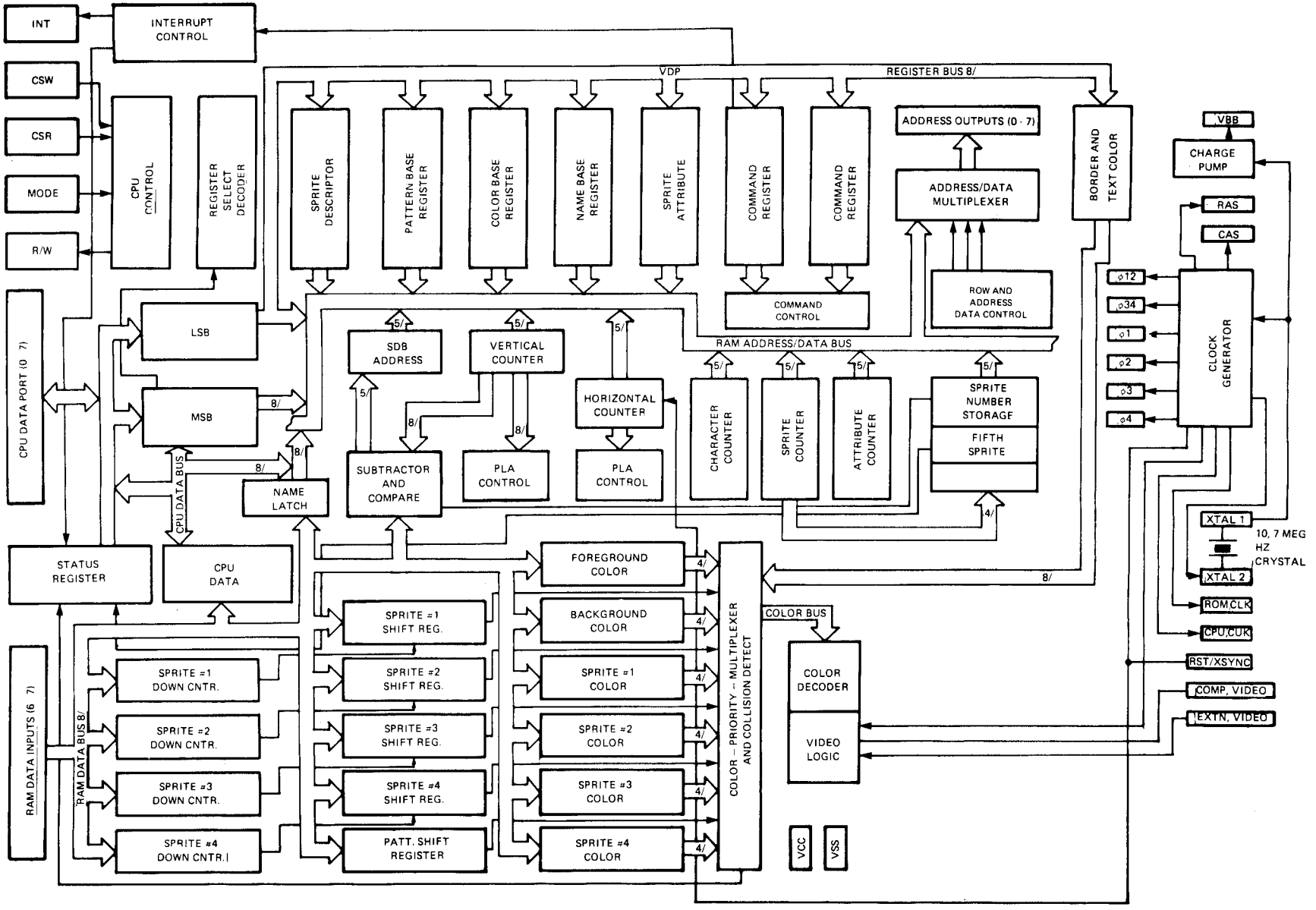


FIGURE 2. TMS 9918 VDP BLOCK DIAGRAM

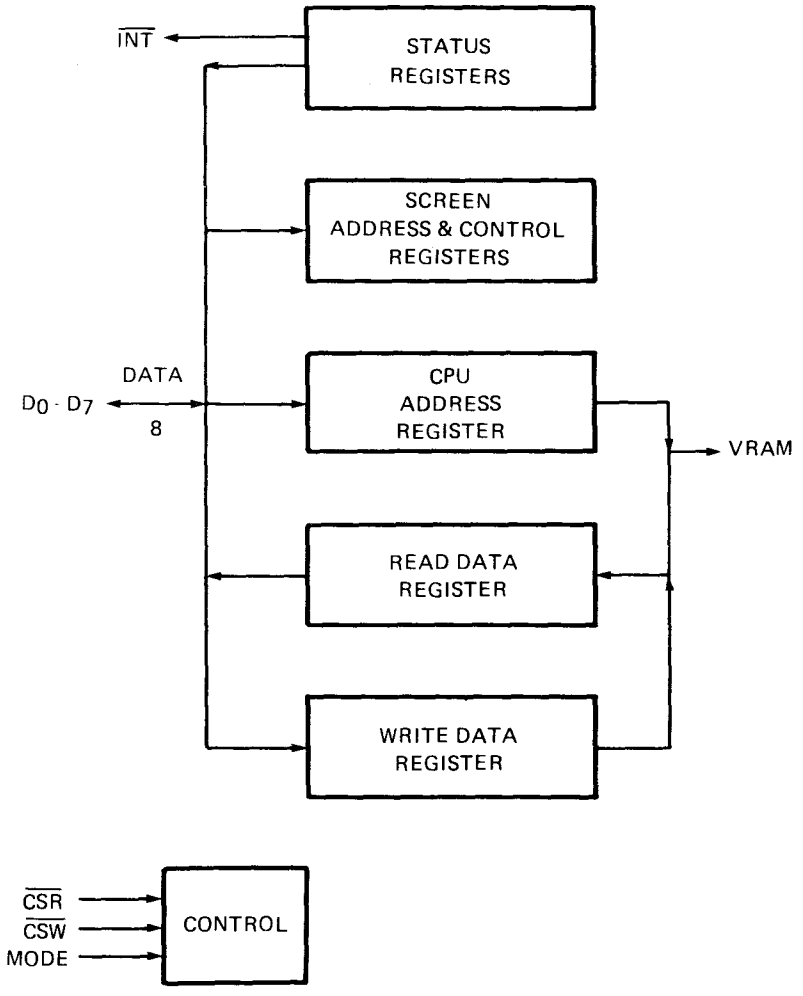


FIGURE 3. VDP TO CPU INTERFACE

2.1.2 CPU Write To VDP Register

The VDP has eight write-only registers and one read-only status register. The write-only registers are used to control the VDP operation and to determine the way in which VRAM is allocated. The status register contains interrupt, sprite coincidence and fifth sprite status flags.

Each of the 8 VDP write-only registers can be loaded using two 8-bit data transfers from the CPU. Table 1 describes the required format for the two bytes. The first byte transferred is the data byte, and the second byte transferred controls the destination. The most significant bit of the second byte must be a '1', the next four bits are '0's, and the lowest three bits are the destination register number. The MODE input is high for both byte transfers.

2.1.3 CPU Write To VRAM

The CPU transfers data to the VRAM through the VDP using a 14-bit autoincrementing address register. Two bytes transfers are required to set up the address register. One byte transfer is then required to write the data to the addressed VRAM byte. The address register is then autoincremented. Sequential VRAM writes require only one byte transfer since the address register is already set up. During set up of the address register, the two most significant bits of the second address byte must be '0' and '1' respectively. MODE is high for both address transfers and low for the data transfer. $\overline{CSW}$ is used in all transfers to strobe the 8-bits into the VDP. See Table 1.

TABLE 1. CPU/VDP DATA TRANSFERS

OPERATION	BIT 0	1	2	3	4	5	6	7	$\overline{CSW}$	$\overline{CSR}$	MODE
WRITE TO VDP REGISTER											
BYTE 1 DATA WRITE	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	0	1	1
BYTE 2 REGISTER SELECT	1	0	0	0	0	RS_0	RS_1	RS_2	0	1	1
WRITE TO VRAM											
BYTE 1 ADDRESS SET UP	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	0	1	1
BYTE 2 ADDRESS SET UP	0	1	A_0	A_1	A_2	A_3	A_4	A_5	0	1	1
BYTE 3 DATA WRITE	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	0	1	0
READ FROM VDP REGISTER											
BYTE 1 DATA READ	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	1	0	1
READ FROM VRAM											
BYTE 1 ADDRESS SET UP	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	0	1	1
BYTE 2 ADDRESS SET UP	0	0	A_0	A_1	A_2	A_3	A_4	A_5	0	1	1
BYTE 3 DATA READ	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	1	0	0

2.1.4 CPU Read From VDP Status Register

The CPU can read the contents of the status register with a single byte transfer. MODE is high for the transfer. $\overline{CSR}$ is used to signal the VDP that a read operation is required.

2.1.5 CPU Read From VRAM

The CPU reads data from the VRAM through the VDP using the autoincrementing address register. Two byte transfers are required to set up the address register. One byte transfer is then required to read the data from the addressed VRAM byte. The address register is then autoincremented. Sequen-

tial VRAM data reads require only one byte transfer since the address register is already set up. During set up of the address register, the two most significant bits of the second address byte must be '0's. MODE is high for the address byte transfers and low for the data transfers. The VDP requires approximately 8 microseconds to fetch the VRAM byte following either the address register set up or another data transfer. $\overline{CSW}$ is used to strobe in the 2 address bytes; $\overline{CSR}$ is used to signal the VDP to fetch the data out of VRAM.

2.1.6 VDP Interrupt

The VDP $\overline{INT}$ output pin is used to generate an interrupt at the end of each active-display scan, which is about every 1/60 second (color burst frequency/60192). The $\overline{INT}$ output is active when the Interrupt Enable bit (IE) in VDP register 1 is a '1' and the F bit of the status register is a '1'. Interrupts are cleared when the status register is read.

2.1.7 VDP Initialization

The VDP is externally initialized whenever the $\overline{RESET}$ input is active (low). The external reset synchronizes all clocks to its falling edge, sets the horizontal and vertical counters to known states, and clears VDP registers 0 and 1. The video display is automatically blanked since the BLANK bit in VDP register 1 becomes a '0'. The VDP, however, continues to refresh the VRAM even though the display is blanked. While the RESET line is active, the VDP does not refresh VRAM.

2.2 VDP/VRAM INTERFACE

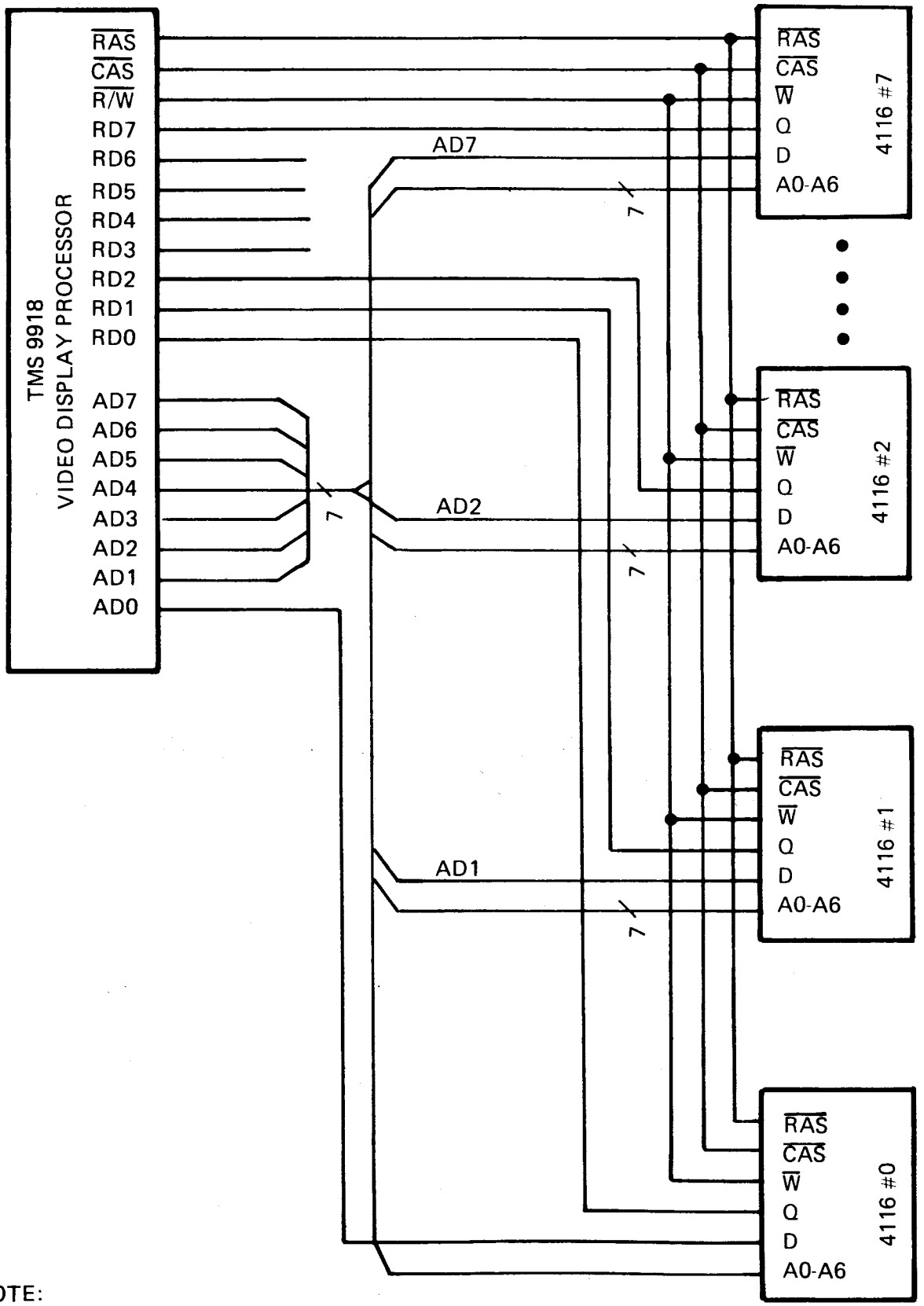
The VDP can access up to 16,384 bytes of VRAM using a 14-bit VRAM address. The VDP fetches data from the VRAM in order to process the video image as described later. The VDP also stores data in or reads in data from the VRAM during a CPU-VRAM data transfer. The VDP automatically refreshes the VRAM.

2.2.1 VRAM Interface Control Signals

The VDP-VRAM interface consists of two unidirectional 8-bit data busses and three control lines as shown in Figure 4. The VRAM outputs data to the VDP on the VRAM read data bus (RD0-RD7). The VDP outputs both the address and data to the VRAM over the VRAM address/data bus (AD0-AD7). The VRAM row address is output when $\overline{RAS}$ is active (low), and the column address is output when $\overline{CAS}$ is active (low). Data is output to the VRAM when R/W is active (low).

2.2.2 VRAM Memory Types

The VDP can utilize either 4027 type 4K or 4116 type 16K dynamic RAMs. The 4/16K bit in VDP register 1 is a '0' for 4027 type RAMs and a 1 for 4116 type RAMs. The VRAM can be expanded to 16K by either switching from 4027's to 4116's: Note that 4027's and 4116's have a minor difference in the way in which they are wired to the VDP, If one wishes, a jumper could be used to select the VRAM type. In the 4027 position, all $\overline{CE}$ pins on the 4027's are tied to ground, and in 4116 mode the A6 lines on the 4116's (the same pin as $\overline{CE}$ on 4027's) are all tied to AD1 on the TMS 9918.



NOTE:
ADO is the most significant bit of the AD bus,
AD7 is least significant. RDO is the most significant
bit of the RD bus, RD7 is the least significant.
AD7 from the VDP connects to AO of the 4116.

FIGURE 4. VRAM INTERFACE

Fast memories are required for use with the VDP. When using 4027 or 4116's, the devices should have dash numbers -15.

2.4.1 Register 0

The External Video (EV) bit is the least significant bit in register 0. It is a '1' when the external video option is required and '0' otherwise. All other bits are reserved for future use and must be '0's.

2.4.2 Register 1

Register 1 contains eight VDP option control bits.

BIT 0 — 4/16K
'0' for 4027 type (4K) RAMs; '1' for 4116 type (16K) RAMs.

BIT 1 — BLANK
'0' to blank active display area; '1' to enable display.
Blanking causes the display to show border color only.

BIT 2 — IE
'0' to disable VDP interrupt; '1' to enable.

BIT 3 — M1 (mode bit 1); M1 and M2 determine the operating mode of the VDP:

M1	M2	
0	0	0: Pattern Mode;
0	1	1: Multicolor Mode;
1	0	0: Text Mode;

2.3 VDP/TV INTERFACE

The Composite Video Output signal coming from the VDP is used to drive an NTSC color television or monitor. This signal incorporates all necessary horizontal and vertical synchronization signals, as well as luminance and chrominance information. In monitor applications, the requirements of the monitor should be studied in order to determine if the VDP can be connected directly to it. In some cases, it may be necessary to provide a simple interface circuit in order to match the VDP's output voltages with the monitor's specifications.

If it is desired to drive a standard television that is not outfitted with a composite video input, the signal can be run into the televisions antenna terminals by using an appropriate RF modulator on the VDP output. Once again, care should be taken to ensure proper matchup between VDP, RF modulator, and TV.

2.4 WRITE-ONLY REGISTERS

The eight VDP write-only registers are shown in Figure 5. They are loaded by the CPU as described in Section 2.1.2. Registers 0 and 1 contain flags to enable or disable various VDP features and modes. Registers 2 through 6 contain values which specify starting locations of various sub-blocks of VRAM. The definitions of these sub-blocks are described in Section 2.7. Register 7 is used to define backdrop and text colors.

The following is a description of each register.

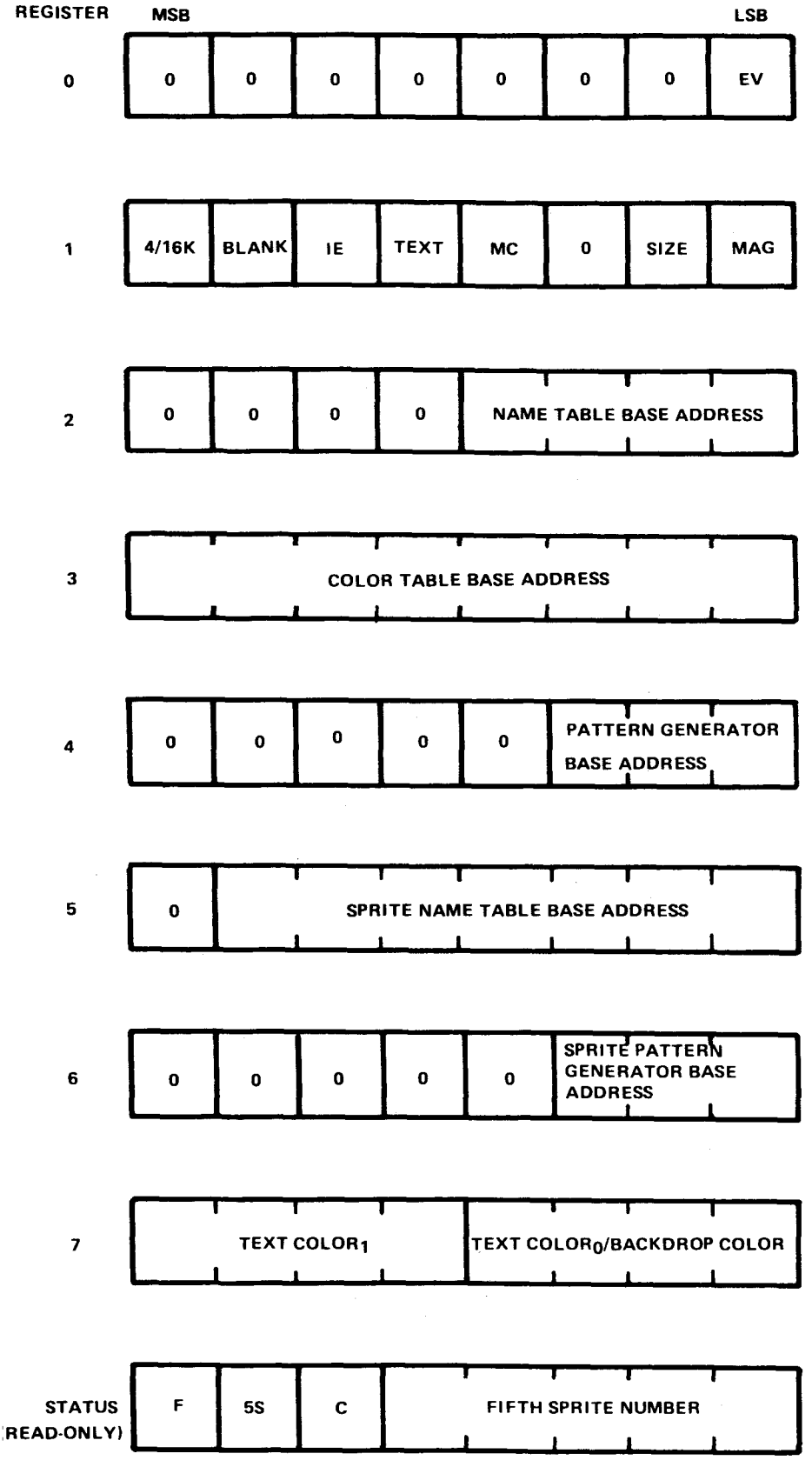


FIGURE 5. VDP REGISTERS

BIT 4	M2 (mode bit 2);
BIT 5	
BIT 6	SIZE '0' for SIZE0 (8 × 8 bit) sprites; '1' for SIZE1 (16 × 16) sprites
BIT 7	MAG (magnification) '0' for MAG0 sprites; '1' for MAG1 sprites

2.4.3 Register 2

Register 2 defines the base address of the Name Table sub-block. The range on its contents is from 0 to 15. The contents of the register form the upper 4 bits of the 14-bit Name Table addresses; thus the Name Table base address is equal to (register 2) * 400 (hex).

2.4.4 Register 3

Register 3 defines the base address of the Color Table sub-block. The range on its contents is from 0 to 255. The contents of the register form the upper 8 bits of the 14-bit Color Table addresses; thus the Color Table base address is equal to (register 3) * 40 (hex).

2.4.5 Register 4

Register 4 defines the base address of the Pattern, Text or Multicolor Generator sub-block. The range on its contents is 0 through 7. The contents of the register form the upper 3 bits of the 14-bit Generator addresses; thus the Generator base address is equal to (register 4) * 800 (hex).

2.4.6 Register 5

Register 5 defines the base address of the Sprite Name Table sub-block. The range on its contents is from 0 through 127. The contents of the register form the upper 7 bits of the 14-bit Sprite Name Table addresses; thus the base address is equal to (register 5) * 80 (hex).

2.4.7 Register 6

Register 6 defines the base address of the Sprite Pattern Generator sub-block. The range on its contents is 0 through 7. The contents of the register form the upper 3 bits of the 14-bit Sprite Pattern Generator addresses; thus the Sprite Pattern Generator base address is equal to (register 6) * 800 (hex).

2.4.8 Register 7

The upper four bits of register 7 contain the color code of color1 in the text graphics mode. The lower four bits contain the color code for $color_0$ in the text graphics mode and the backdrop color in all modes. See Table 2 for color codes.

2.5 STATUS REGISTER

The VDP has a single 8-bit status register which can be accessed by the CPU. The status register contains the interrupt pending flag, the sprite coincidence flag, the fifth sprite flag, and the fifth sprite number, if one exists. The format of the status register is shown in Figure 5; a discussion of the contents follows.

The status register may be read at any time to test the F, C, and 5S status bits. Reading the status register will clear the interrupt flag, F. Asynchronous reads will, however, cause the frame flag (F) bit to be reset and therefore missed. Consequently, the status register should be read only when the VDP interrupt is pending.

2.5.1 Interrupt Flag (F)

The F status flag in the status register is set to '1' at the end of the raster scan of the last line of the active display. It is reset to a '0' after the status register is read or when the VDP is externally reset. If the Interrupt Enable bit in VDP register 1 is active, a '1', the VDP interrupt output (INT) will be active (low) whenever the F status flag is a '1'.

2.5.2 Coincidence Flag (C)

The C status flag in the status register is set to a '1' if two or more sprites "coincide". Coincidence occurs if any two sprites on the screen have one or more overlapping pixels. Transparent colored sprites, as well as those that are partially or completely off the screen are also considered. Sprites beyond the Sprite Name Table terminator ($D0_{16}$) are not considered. The 'C' flag is cleared to a '0' after the status register is read or the VDP is externally reset.

2.5.3 Fifth Sprite Flag (5S) and Number

The 5S status flag in the status register is set to a '1' whenever there are five or more sprites on a horizontal line (lines 0 to 192) and the frame flag is equal to a '0'. The 5S status flag is cleared to a '0' after the status register is read or the VDP is externally reset. The number of the fifth sprite is placed into the lower 5 bits of the status register when the 5S flag is set and is valid whenever the 5S flag is '1'. The setting of the fifth sprite flag will not cause an interrupt.

2.6 OSCILLATOR AND CLOCK GENERATION

The VDP is designed to operate with a 10.738635 MHz ±50 PPM crystal input to generate the required internal clock signals. A fundamental frequency, parallel mode crystal is used as the frequency reference for the internal clock oscillator, which is the master time base for all system operations. This master clock is divided by two to generate the pixel clock (5.3 MHz) and by three to provide the CPUCLK (3.58 MHz). The GROMCLK is developed from the master clock divided by 24.

2.6.1 Color Phase Generation

The 10.7+ MHz master clock and its complement are used to generate an internal 6-phase 3.579545 MHz (±10 Hz) clock to provide the video color signals and the color burst reference for use in developing the composite video output signal. While the VDP signals are not exact equivalents to the standard NTSC colors, the differences can easily be adjusted by the color and tint controls of the target color television.

2.6.2 Video Sync and Control Generation

The vertical and horizontal control signals are generated by decoding the outputs of the horizontal and vertical counters. The horizontal counter is driven by the pixel clock, and the horizontal counter in turn increments the vertical counter. Table 3 gives the relative count values of the screen display parameters. Within the active display area during pattern graphics mode, the three least significant bits of the horizontal counter address the individual picture element of each pattern displayed. Also, during the vertical active display period, the three least significant bits of the vertical counter address each individual line in the 8 × 8 patterns. The multicolor and text modes use the counters similarly.

The VDP operates at 264 lines per frame and approximately 60 frames per second in a non-interlaced mode of operation.

TABLE 3. SCREEN DISPLAY PARAMETERS

HORIZONTAL		PIXEL CLOCK CYCLES	
		PATTERN OR MULTICOLOR	TEXT
HORIZONTAL ACTIVE DISPLAY		256	240
RIGHT BORDER		15	25
RIGHT BLANKING		8	8
HORIZONTAL SYNC		26	26
LEFT BLANKING		2	2
COLOR BURST		14	14
LEFT BLANKING		8	8
LEFT BORDER		13	19
	TOTAL	342	342

VERTICAL		LINE
VERTICAL ACTIVE DISPLAY		192
BOTTOM BORDER		26
BOTTOM BLANKING		3
VERTICAL SYNC		3
TOP BLANKING		13
TOP BORDER		27
	TOTAL	264

2.7 VIDEO GRAPHICS

The VDP displays an image on the television which can best be envisioned as a set of display planes sandwiched together. Figure 6 shows the definition of each of the planes. Objects on planes which are closest to the viewer have a higher priority; in cases where 2 entities on 2 different planes are occupying the same spot on the screen, the entity on the higher priority plane will show at that point. For an entity on a specific plane to show through, all planes in front of that plane must be transparent at that point. The first 32 planes each may contain a sprite. Sprites are objects whose positions on the screen are defined by horizontal and vertical coordinates in VRAM. The areas of the Sprite planes outside of the Sprite itself are transparent. Since the coordinates of the sprite are in terms of pixels, the sprite can be positioned and moved about very accurately. Sprites are available in 3 sizes; 8 × 8 pixels, 16 × 16 pixels, and 32 × 32 pixels. Behind the Sprite planes is the Pattern plane. This is the plane used for textual and fixed graphics images. Behind the Pattern plane is the backdrop, which is larger in area than the other planes, so that it forms a border around the other planes. The last plane is the external video plane, whose image is defined by the external video input pin. The backdrop consists of a single color which is used for the display borders and as the default color for the active display area. The default color is stored in the VDP register 7. When the backdrop color register contains the transparent code, the backdrop automatically defaults to black when the external video mode is not selected.

The thirty-two sprite planes are used for the thirty-two sprites in the multicolor and pattern graphics modes. They are not used in the text mode and are automatically transparent. Each of the sprites can cover an 8 × 8, 16 × 16, or 32 × 32 pixel area on its plane. Any part of the plane not covered by the sprite is transparent. All or part of each sprite may also be transparent. Sprite 0 is on the outside or highest plane, and sprite 31 is on the plane immediately adjacent to pattern plane. Whenever a pixel in a sprite plane is transparent, the color of the next plane can be seen through that plane. If, however, the sprite pixel is non-transparent, the colors of the lower planes are automatically replaced by the sprite color. As a restriction on the number of sprites on a line. Only four sprites can be active on any horizontal line. Additional sprites on a line will be automatically made transparent for that line. Only those sprites which are active on the display will cause the coincidence flag to set. The VDP status register provides

Contd Page 15

TABLE 2. COLOR ASSIGNMENTS

COLOR (HEX)	COLOR	LUMINANCE (DC VALUE)	CHROMINANCE (AC VALUE)
0	TRANSPARENT	0.00	—
1	BLACK	0.00	—
2	MEDIUM GREEN	.60	.60
3	LIGHT GREEN	.80	.53
4	DARK BLUE	.47	.73
5	LIGHT BLUE	.67	.60
6	DARK RED	.53	.53
7	CYAN	.80	.73
8	MEDIUM RED	.67	.73
9	LIGHT RED	.80	.73
A	DARK YELLOW	.87	.53
B	LIGHT YELLOW	1.00	.40
C	DARK GREEN	.47	.60
D	MAGENTA	.60	.47
E	GRAY		—
F	WHITE	1.00	—
—	BLACK LEVEL	0.00	—
—	COLOR BURST	0.00	.40
—	SYNC LEVEL	−0.40	—

2.7.1 Pattern Mode

The VDP is in Pattern mode when both M1 and M2 bits in VDP register 1 are zero. In the Pattern mode, the pattern plane is divided into a grid of 32 across by 24 down pattern positions. (See Figure 7). Each of the pattern positions contains 8 × 8 pixels. The tables in VRAM used to generate the pattern plane are the Pattern Name Table, the Pattern Generator Table, and the Pattern Color Table. Figure 8 illustrates the mapping of these tables into the pattern plane. A total of 2848 VRAM bytes are required for the pattern name, color, and generator tables. Less memory is required if all 256 possible pattern definitions are not required; the tables can be overlapped to reduce the amount of VRAM needed for pattern generation. Examples of VRAM memory allocation are provided in Appendix B.

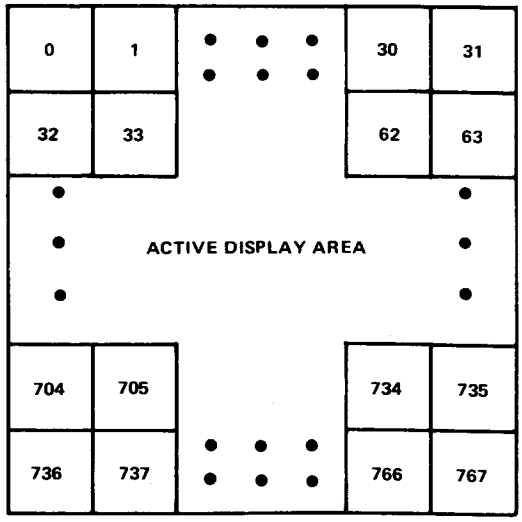
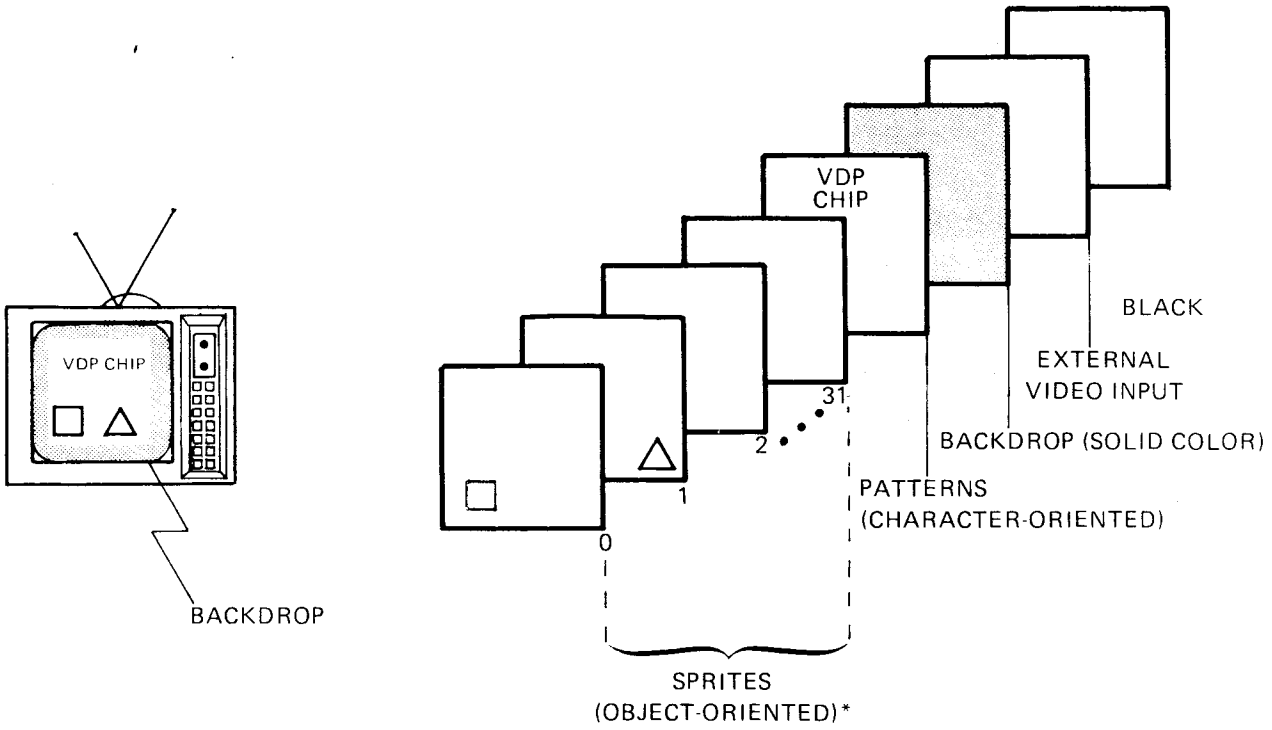


FIGURE 7. PATTERN GRAPHICS NAME TABLE MAPPING

Contd P 17



*NOT AVAILABLE IN TEXT MODE.

FIGURE 6a. VDP DISPLAY PLANES

a flag bit and the number of the fifth sprite whenever this occurs. The pattern plane is used in the text, multicolor, and pattern graphics modes for display of the graphic patterns or characters. Whenever a pixel on the pattern plane is non-transparent, the backdrop color is automatically replaced by the pattern plane color. When a pixel in the pattern plane is transparent, the backdrop color can be seen through the pattern plane.

The VDP has three video color display modes: Pattern mode, Text mode, and Multicolor mode. The pattern mode causes the Pattern plane to be broken up into groups of 8 × 8 pixels, called pattern positions. Since the full image is 256 × 192 pixels, there are 32 × 24 pattern positions on the screen in Pattern mode. Within each pattern position, 2 unique colors are allowed. In Text mode, the pattern plane is broken into groups of 6 × 8 pixels, called text positions. There are 40 × 24 text positions on the screen in this mode. In Text mode sprites do not appear on the screen and two colors are defined for the entire screen. In multicolor mode, the screen is broken into a grid of 64 × 48 positions, each of which is 4 × 4 pixels. Within each position, one unique color is allowed.

The VDP registers define the base addresses for several sub-blocks within VRAM. These sub-blocks form tables which are used to produce the desired image on the TV screen. The Pattern Name Table, the Pattern Generator Table and the Pattern Color Table are used to form the Pattern plane, while the Sprite Name Table and the Sprite Generator Table are used to form the Sprites. The contents of these tables must all be provided by the microprocessor. Animation is achieved by altering the contents of VRAM in real time.

The VDP can display the fifteen colors shown in Table 2. The VDP colors also provide eight different gray levels for displays on monochrome televisions; the luminance values in the table indicate these levels, 0. 00 being black and 1.00 being white. Whenever all planes are of the transparent color at a given point, the color shown at that point will be black.

The video display consists of thirty-five display planes: external video, backdrop, pattern, and sprite planes 0-31. The planes are stacked with the external video the innermost or lowest plane. The sprite 0 plane is the outside or highest plane.

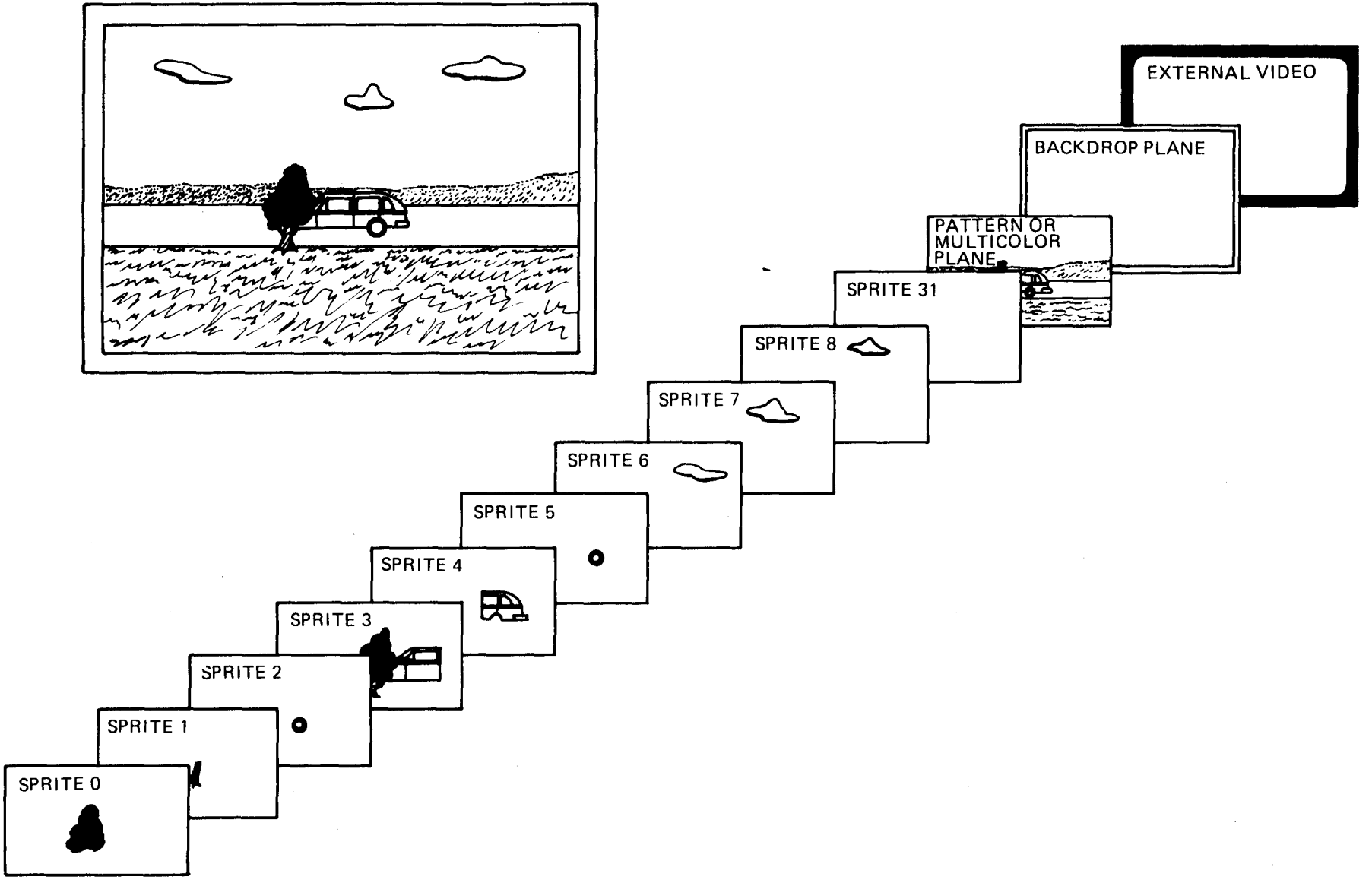


FIG. 6b VDP DISPLAY PLANES

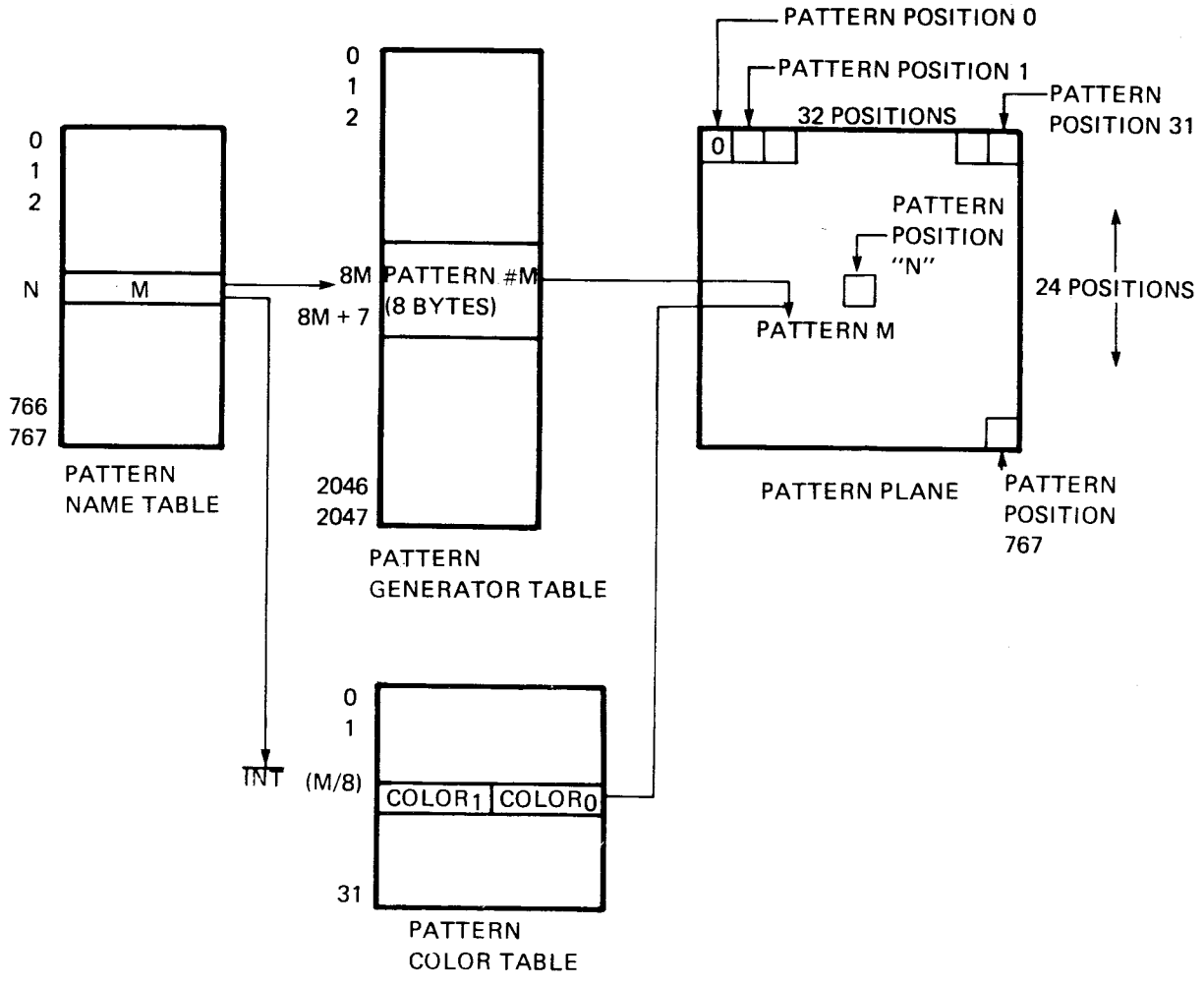
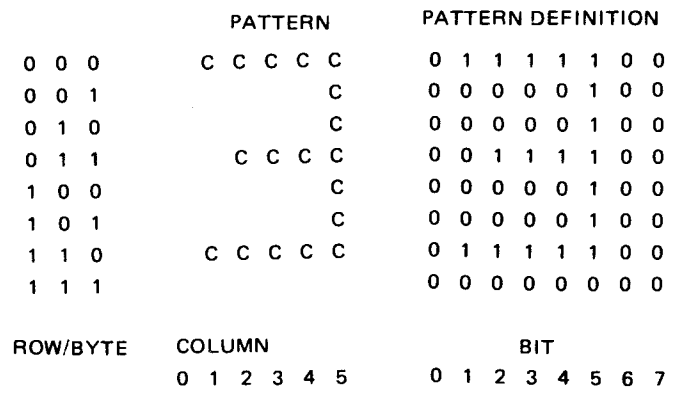


FIGURE 8. PATTERN MODE MAPPING

The Pattern Generator Table contains a library of patterns that can be displayed in the pattern positions. It is 2048 bytes long, and is arranged into 256 patterns, each of which is 8 bytes long, yielding 8 × 8 bits. Each 8-byte pattern defines a pattern in that all of the one's in the pattern can be made one color (color 0), while all the zeros are made another color (color 1).

The full eight bit pattern name is used to select one of the 256 pattern definitions in the Pattern Generator Table. The table is a 2048 bytes block in VRAM beginning on a 2K byte boundary. The starting address of the table is determined by the generator base address in VDP register 4. The base address forms the three most significant bits of the 14-bit VRAM address for each pattern generator table entry. The next eight bits are equal to the eight bit name of the selected pattern definition. The lowest three bits of the VRAM address are equal to the row number within the pattern definition.

Eight bytes are required for each of the 256 possible unique 8 × 8 pattern definitions. The first byte defines the first row of the pattern, and the second byte defines the second row. The first bits of each of the eight bytes define the first column of the pattern. The remaining rows and columns are similarily defined. Each bit entry in the pattern definition selects one of the two colors for that pattern. A '1' bit selects the color code ($color_1$) contained in the most significant four bits of the corresponding color table byte. A '0' bit selects the other color code ($color_0$). An example of pattern definition mapping is provided in Figure 9.



VDP register 7 entry: 71_{16}

color code 7 is cyan (signified above by 'C')

color code 1 is black (signified above by a space)

Note: Bit 0 is the most significant bit of each data byte.

FIGURE 9. PATTERN DISPLAY MAPPING

The color of the one's and zero's is defined by the Pattern Color Table. It contains 32 entries each of which is one byte long. Each entry defines two colors: the most significant 4 bits of each entry define the color of the one's, and the least significant 4 bits define the color of the zero's. The first entry in the color table defines the colors for patterns #0 to 7; the next entry for patterns #8 to 15, and so on. See Table 4 for assignments. Thus one can display 32 different pairs of colors simultaneously.

The Pattern Name Table is located in a contiguous 768 byte block in VRAM beginning on a 1K byte boundary. The starting address of the name table is determined by the 4-bit name table base address field in VDP register 2. The base address forms the upper four bits of the 14 bit VRAM address. The lower 10 bits of the VRAM address point. An example of pattern name table addressing is given in Appendix A.

Each byte entry in the name table is the name of or the pointer to a pattern definition in the Pattern Generator Table. The upper 5 bits of the 8 bit name identify the color group of the pattern. There are 32 groups of 8 patterns. The same two colors are used for all 8 patterns in a group; the color codes are stored in the VDP color table. The color table is located in a 32 byte block in VRAM beginning on a 64 byte boundary. The table starting address is determined by the 8-bit color table base address in VDP register 3. The base address forms the upper eight bits of the 14-bit color table entry VRAM address. The next bit is a '0' and the lowest five bits are equal to the upper five bits of the corresponding name table entries. An example of color table addressing is provided in Appendix A.

Note that since the tables in VRAM have their base addresses defined by the VDP registers, a complete switch of the values in the tables can be made by simply changing the values in the VDP registers. This is especially useful when one wishes to time-slice between two or more screens of graphics.

Once the Pattern Generator Table is loaded with a pattern set, one need only manipulate the Pattern Name Table contents in order to change the appearance of the screen. Alternatively, one might wish to have a dynamically changing set of patterns throughout the course of a graphics session. Since all tables are in VRAM, this is easily accomplished.

For textual applications, the desired character set is typically loaded into the Pattern Generator first. For example, one might load the official USASCII character set into the Pattern Generator in such a way that the pattern numbers correspond to the 8-bit ASCII codes for that pattern; e.g., the pattern for the letter "A" would be loaded into pattern number 41_{16} in the Pattern Generator. Next the Pattern Color Table would be loaded up with the proper color set. When one wished to print a textual message on the screen, then, one would merely write the proper ASCII codes out to the Pattern Name Table.

Graphics can be achieved using the Pattern plane. To display an object of size 8 × 8 pixels or smaller, only one pattern would need to be defined. To display a larger figure, the figure should be broken up into smaller 8 × 8 squares. Then multiple patterns can be defined, and the Pattern Generator and Pattern Name Table set up appropriately. Note that rough motion of objects is done very easily by merely updating entries in the Pattern Name Table.

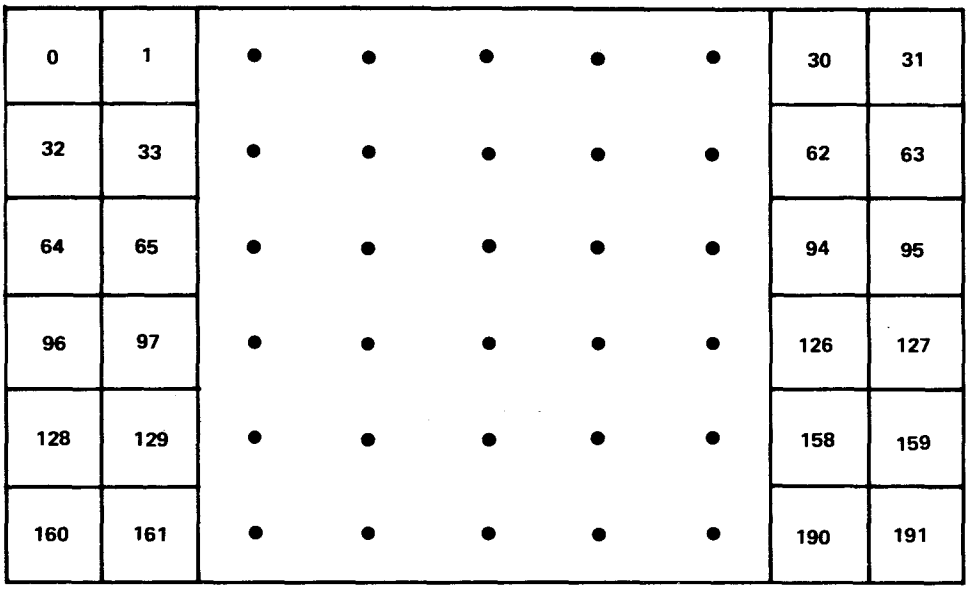
TABLE 4. PATTERN COLOR TABLE

Byte No.	Pattern No.
0	0..7
1	8..15
2	16..23
3	24..31
4	32..39
5	40..47
6	48..55
7	56..63
8	64..71
9	72..79
10	80..87
11	88..95
12	96..103
13	104..111
14	112..119
15	120..127
16	128..135
17	136..143
18	144..151
19	152..159
20	160..167
21	168..175
22	176..183
23	184..191
24	192..199
25	280..207
26	208..215
27	216..223
28	224..231
29	232..239
30	240..247
31	248-255

A total of 2848 VRAM bytes are required for the pattern name, color, and generator tables. Less memory is required if all 256 possible pattern definitions are not required; the tables can be overlapped to reduce the amount of VRAM needed for pattern generation. Examples of VRAM memory allocation are provided in Section 3.3.

2.7.2 Multicolor Mode

The VDP is in Multicolor mode when M1 and M2 in VDP register 1 are zero and one respectively. It provides an unrestricted 64 × 48 color square display. Each color square contains a 4 × 4 block of pixels. The color of each of the 3072 color squares can be any of the 15 video display colors plus transparent. Consequently, all 15 colors can be simultaneously used in the multicolor mode. The backdrop and sprite planes are active in the multicolor mode.



NOTE: Each numbered block above corresponds to an entry in the multicolor name table. Each block consists of 16 color squares (2 columns x 8 rows).

The multicolor name table is exactly the same as that for the pattern mode, a grid of 32 columns across by 24 rows down of cell positions. The way in which each cell position is filled is different from the pattern mode.

The names are used to address the pattern generator table entries. Each entry contains 8 bytes, each byte is used for two 4 bit color codes. Since each cell on the screen is 8 × 8 pixels, 2 bytes are required to define the 4 colors for the 4 squares (each square is 4 × 4 pixels) to fill each cell. Pattern name table entries 0 to 31 (Row 0), 128 to 159 (Row 4), 256 to 287 (Row 8), 384 to 415 (Row 12), 512 to 543 (Row 16), and 640 to 671 (Row 20) all use the first 2 Bytes (Bytes 0 and 1) of the pattern generator entry pointed to by the Name. Pattern name table rows (each row with 32 entries) 1, 5, 9, 13, 17, and 21 all use the second two bytes (bytes 2 and 3) of the pattern generator entry. Pattern name table rows 2, 6, 10, 14, 18, and 22 all use the third two bytes (bytes 4 and 5) and name table rows 3, 7, 11, 15, 19, and 23 all use the fourth two bytes of the pattern generator entry.

There are a number of ways to use this indirect mapping scheme, the following is a suggested way to memory map the multicolor squares:

The technique used is to define the pattern name table such that the pattern generator table maps onto the screen. Groups of four pattern name table entries are formed into multicolor blocks. A block is composed of 4 pattern name table entries which are in the same display column and in adjacent rows. Note the entries in a block are 32 bytes apart in the pattern name table since there are 32 columns per row). Each name table entry in the same block is given the same name, called the block number. The blocks are numbered as shown in Figure 12, with block 0 containing pattern name table entries 0, 32, 64, and 96, and block 191 containing pattern name table entries 671, 703, 735, and 767. The mapping is shown in Figure 13 and block display is shown in Figure 14.

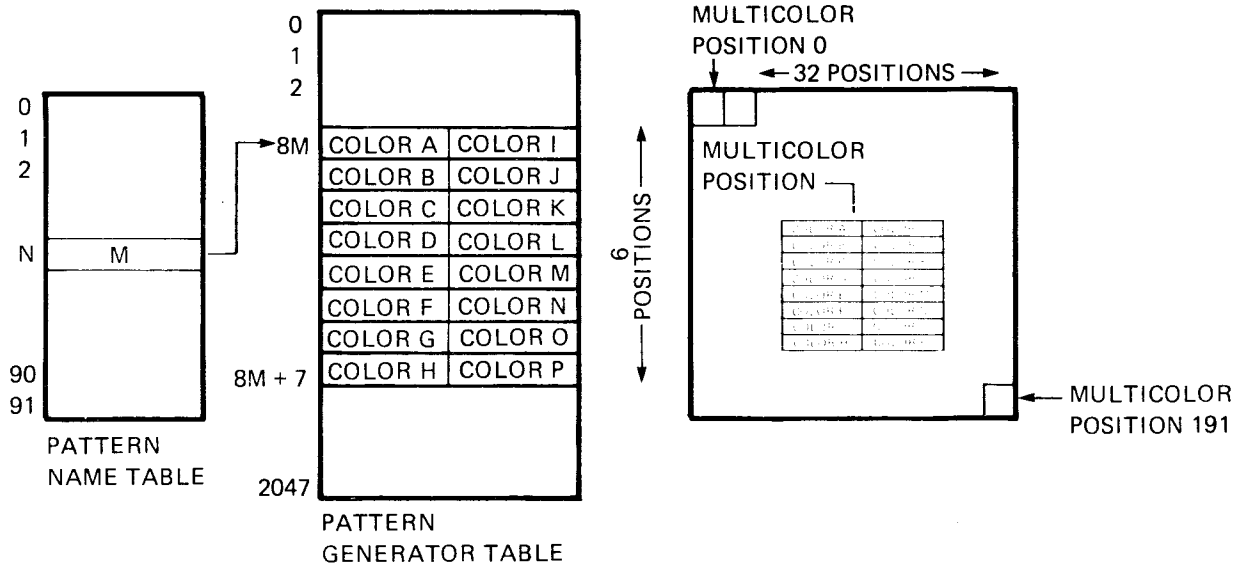


FIGURE 11. MULTICOLOR MODE MAPPING

The Pattern Generator Table consists of 192 entries, each of which is 8 bytes long; the total size is 1536 bytes. The values of the entries in the Pattern Name Table specify which of the 192 entries in the Pattern Generator Table to get the color information out of. The eight bytes of the chosen entry out of the Pattern Generator Table choose the sixteen colors to appear in the block on the screen. The mapping of the 8 bytes to the 16 colors is very straightforward, as shown in Figure 12.

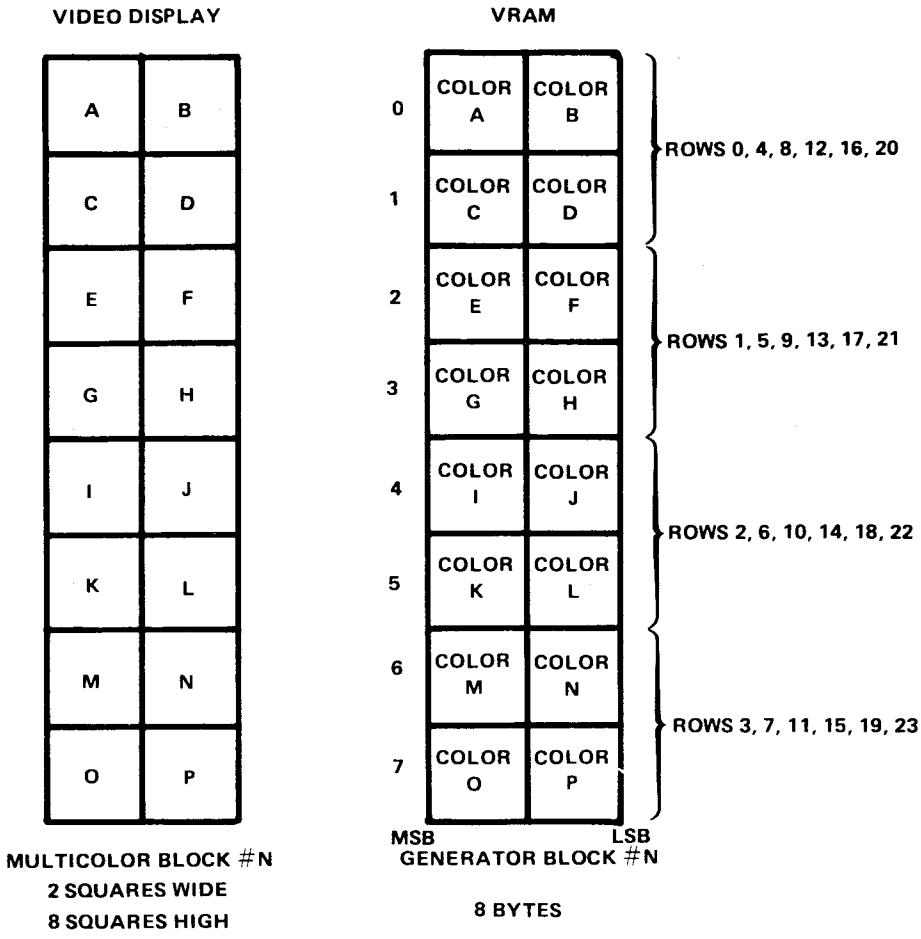


FIGURE 12. MULTICOLOR BLOCK DISPLAY

A total of 1728 VRAM bytes are required for the multicolor name and color generator tables. It should be noted that the tables begin on even 1K and 2K byte boundaries respectively and are therefore not contiguous. An example of multicolor VRAM memory allocation is provided in Appendix B.

2.7.3 Text Mode

The VDP is in Text mode when M1 and M2 in VDP register 1 are one and zero respectively. In the Text mode, the screen is divided into a grid of 40 text positions across and 24 down. (See Figure 13). Each of the text positions contains 6 pixels across and 8 pixels down. The tables used to generate the pattern plane are the Pattern Name Table and the Pattern Generator Table. There can be up to 256 unique patterns defined at any time. The pattern definitions are stored in the pattern generator table in VRAM and can be dynamically changed. The VRAM contains a Pattern Name Table which maps the pattern definitions into each of the 960 pattern cells on the pattern plane (Figure 14). Sprites are not available in text mode.

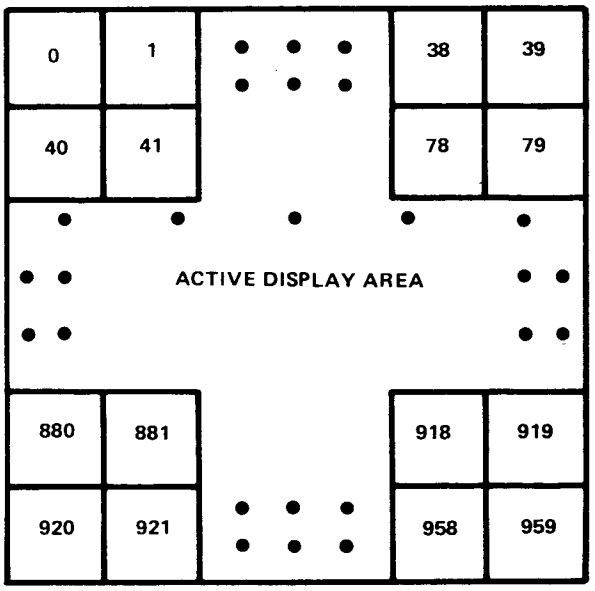


FIGURE 13. PATTERN GRAPHICS NAME TABLE

As in the case of Pattern mode, the Pattern Generator Table contains a library of text patterns that can be displayed in the text positions. It is 2048 bytes long, and is arranged in 256 text patterns, each of which is 8 bytes long. Since each text position on the screen is only 6 pixels across, the least significant 2 bits of each text pattern are ignored, yielding 6 × 8 bits in each text pattern. Each block of 8 bytes defines a text pattern in that all the one's in the text pattern take on one color when displayed on the screen, while all the zero's take on another color. These colors are chosen by loading VDP register 7 with the color1 and color0 in the left and right nibbles respectively (see Section 2.4).

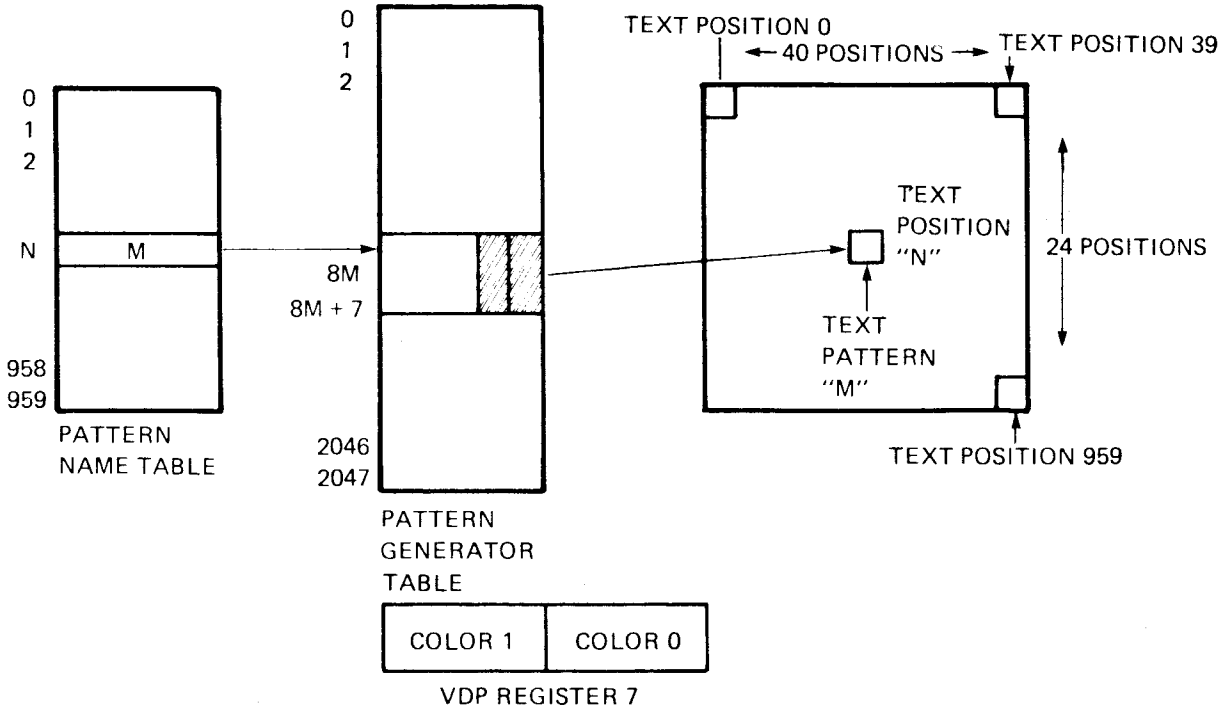


FIGURE 14. MAPPING OF VRAM INTO THE PATTERN PLANE IN TEXT MODE

In the Text mode, the Pattern Name Table determines what text pattern goes where on the screen. There are 960 entries in the Pattern Name Table, each one byte long. There is a one-to-one correspondence between text pattern positions on the screen and entries in the Pattern Name Table (40*24 = 960). The first 40 entries corresponds to the top row of text pattern positions on the screen, the next forty to the second row, and so on. The value of an entry in the Pattern Name Table indicates which of the 256 text patterns is to be placed at that spot on the Pattern plane. The pattern name table is located in a contiguous 960 byte block in VRAM beginning on a 1K byte boundary. The starting address of the name table is determined by the 4-bit name table base address field in VDP register 2. The base address forms the upper four bits of the 14 bit VRAM address. The lower 10 bits of the VRAM address point to one of the 960 pattern cells. The name table is organized by rows. An example of pattern name table addressing is given in Appendix A. Each byte entry in the name table is the pointer to a pattern definition in the Pattern Generator Table. The same two colors are used for all 256 patterns; the color codes are stored in VDP register 7.

As its name implies, the Text mode is intended mainly for textual applications, especially those in which the 32 patterns-per-line in Pattern mode is insufficient. The advantage is that 8 more patterns can be fitted onto one line; the disadvantages are that sprites cannot be used, and only two colors are available for the entire screen. With care, the same text pattern set that is used in Text mode can be also used in Pattern mode. This is done by ensuring that the least significant 2 bits of all the character patterns are 0. A switch from Text mode to Pattern mode, then, results in a stretching of the space between characters, and a reduction of the number of characters per line from 40 to 32. As with the Pattern Mode, once a character set has been defined and placed into the Pattern Generator, one need only update the Pattern Name Table in order to produce and manipulate textual material on the screen.

The full eight bit pattern name is used to select one of the 256 pattern definitions in the pattern generator table. The table is a 2048 byte block in VRAM beginning on a 2K byte boundary. The starting address of the table is determined by the generator base address in VDP register 4. The base address forms the three most significant bits of the 14-bit VRAM address for each pattern generator table entry. The next eight bits are equal to the eight bit name of the selected pattern definition. The lowest three bits of the VRAM address are equal to the row number within the pattern definition.

Eight bytes are required for each of the 256 possible unique 6 × 8 pattern definitions. The first byte defines the first row of the pattern, and the second byte defines the second row. The two least significant bits in each byte are not used. It is, however, strongly recommended that these bits be zeroes. Each bit entry in the pattern definition selects one of the two colors for that pattern. A '1' bit selects the color code ($color_1$) contained in the most significant four bits of VDP register 7. A '0' bit selects the other color code ($color_0$) which is in the least significant four bits of the same VDP Register. An example of pattern definition mapping is provided in Figure 15.

A total of 3005 VRAM bytes are required for the pattern name and generator tables. Less memory is required if all 256 possible pattern definitions are not required; the tables can be overlapped to reduce the amount of VRAM needed for pattern generation. Examples of VRAM memory allocation are provided in Appendix B.

2.7.4 Sprites

The video display can have up to 32 sprites on the highest priority video planes. The sprites are special animation patterns which provide smooth motion and multilevel pattern overlaying. The location of a sprite is defined by the top left hand corner of the sprite pattern. The sprite can be easily moved pixel-by-pixel by redefining the sprite origin. This provides a simple but powerful method of quickly and smoothly moving special patterns. The sprites are not active in the text mode. The 32 sprite planes are fully transparent outside of the sprite itself.

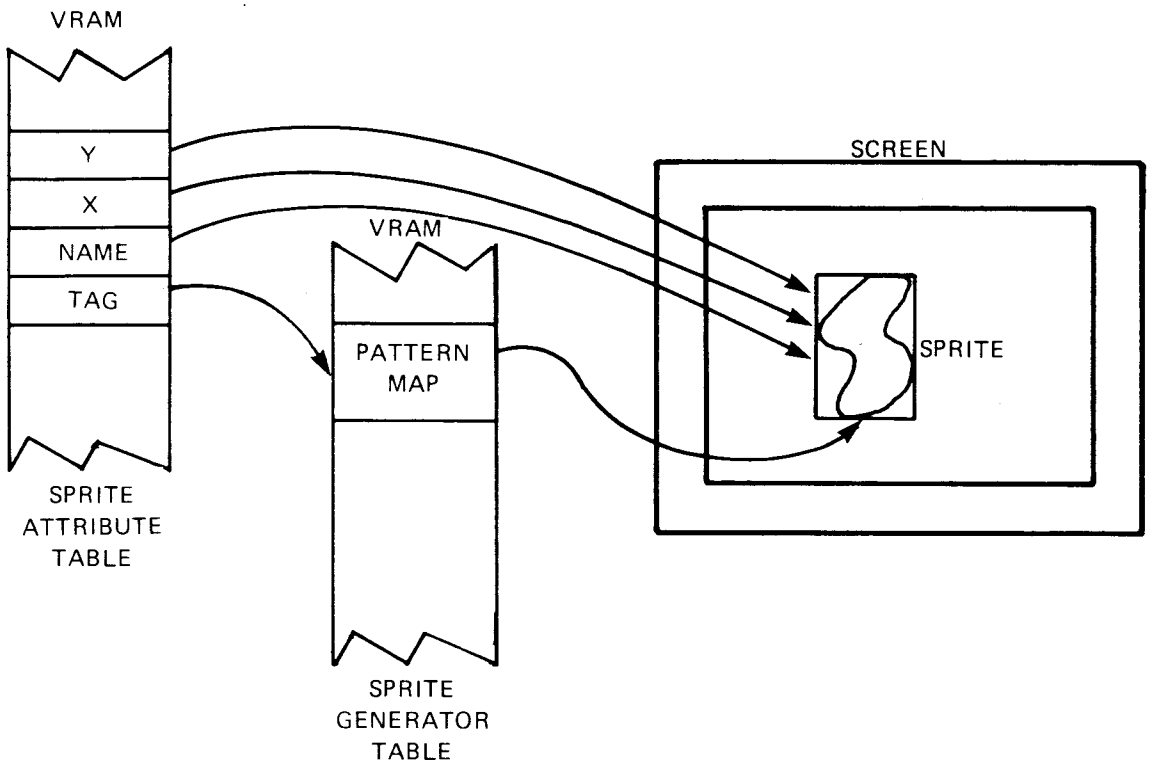


FIGURE 15. SPRITE MAPPING

The sub-blocks in VRAM that define Sprites are the Sprite Attribute Table (Figure 16) and the Sprite Generator Table. These tables are similar to their equivalents in the pattern realm in that the Sprite Attribute Table specifies where the sprite goes on the screen, while the Sprite Generator Table describes what the sprite looks like. Sprite pattern formats are given in Table 5.

Figure 15 illustrates the manner in which the VRAM tables map into the existence of sprites on the display. Since there are 32 sprites available for display, there are 32 entries in the Sprite Attribute Table. Each entry consists of 4 bytes. The entries are ordered such that the first entry corresponds to the sprite on the sprite 0 plane, the next to the sprite on the sprite 1 plane, and so on. The Sprite Attribute Table is 4*32 = 128 bytes long. An ordered list of the sprites is stored in the sprite name table in VRAM. The Sprite Attribute Table is located in a contiguous 128 byte block in VRAM beginning on a 128 byte boundary. The starting address of the attribute table is determined by the 7-bit Sprite Attribute Table base address in VDP register 5. The base address forms the upper seven bits of the 14-bit VRAM address. The next five bits of the VRAM address are equal to the sprite number. The lowest two bits select one of the four bytes in the attribute table entry for each sprite. Each Sprite Attribute Table entry contains four bytes which specify the sprite position, sprite pattern name, and color as shown in Figure 16. An example of sprite name table addressing is provided in Appendix A.

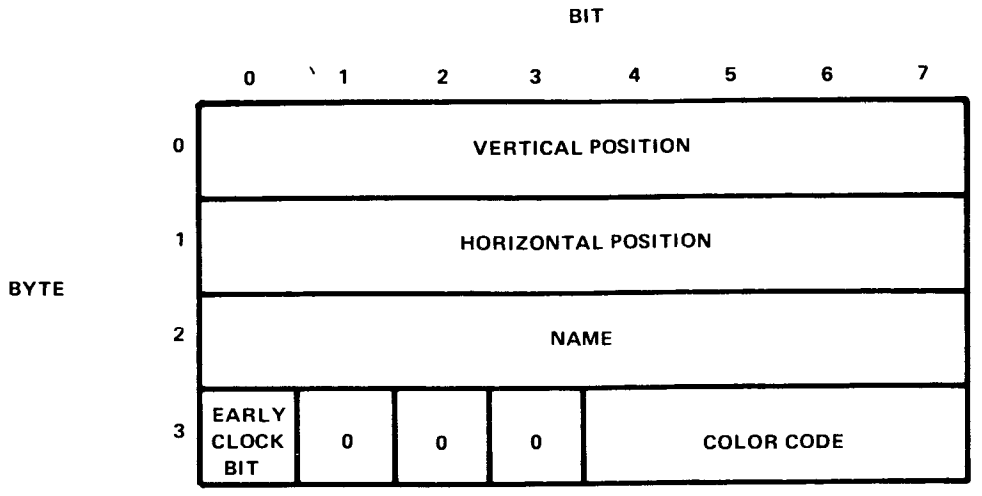


TABLE 5. SPRITE PATTERN FORMATS

SIZE	MAG	AREA	RESOLUTION	BYTES/PATTERN
0	0	8 × 8	single pixel	8
1	0	16 × 16	single pixel	32
0	1	16 × 16	2 × 2 pixels	8
1	1	32 × 32	2 × 2 pixels	32

FIGURE 16. SPRITE ATTRIBUTE TABLE ENTRY

The first 2 bytes of each entry of the Sprite Attribute Table determine the position of the sprite on the display. The first byte indicates the vertical distance of the sprite from the top of the screen, in pixels. It is defined such that a value of 1 puts the sprite butted up at the top of the screen, touching the backdrop area. The second byte describes the horizontal displacement of the sprite from the left edge of the display. A value of 0 puts the sprite butted up against the left edge of the backdrop. Note that it is the upper left pixel of the sprite that all measurements are taken from.

When the first 2 bytes of an entry put a sprite overlapping the backdrop, the part of the sprite that is within the backdrop gets displayed normally, the part of the sprite that overlaps the backdrop gets hidden from view by the backdrop. This allows the animator to cause a sprite to move into the display from behind the backdrop. The displacement in the first byte is partially signed, in that values for vertical displacement between −31 and 0 ($E1_{16}$ to 0) allow a sprite to "bleed in" from the top edge of the backdrop. Likewise, values in the range of 207 to 191 allow the sprite to bleed in from the bottom edge of the backdrop. Similarly, horizontal displacement values in the vicinity of 255 allow a sprite to bleed-in from the right side of the screen. To allow sprites to bleed-in from the left edge of the backdrop, a special bit in the third byte of the Sprite Attribute Table entry is used, as described in a later paragraph.

Byte 3 of the Sprite Attribute Table entry contains the pointer to the Sprite Generator Table that specifies what the sprite should look like. This is an eight bit pointer to the sprite patterns definition in the Sprite Generator Table. The sprite name is similarily to the patterns graphics mode.

Byte 4 of the Sprite Attribute Table entry contains the color of the sprite in its lower 4 bits (see Table 2 for color codes). The most significant bit is the Early Clock bit (EC). This bit, when set to a zero, does nothing. When set to a one, the horizontal position of the sprite is shifted to the left by 32 pixels. This allows a sprite to bleed-in from the left edge of the backdrop. Values for horizontal displacement (byte 2 in the entry) in the range 0 to 32 cause the sprite to overlap with the left-hand border of the backdrop.

The Sprite Generator Table is a maximum of 2048 bytes long beginning on 2K byte boundaries. It is arranged into 256 blocks of 8 bytes each. The third byte of the Sprite Attribute Table entry, then, specifies which 8-byte block to use to specify that sprite's shape. The one's in the Sprite Generator cause the sprite to be defined at that point; zero's cause the transparent color to be used. The starting address of the table is determined by the sprite generator base address in VDP register 6. The base address forms the three most significant bits of the 14-bit VRAM address. The next eight bits of the address are equal to sprite name, and the last three bits are equal to the row number within the sprite pattern. The address formation is slightly modified for $SIZE_1$ sprites.

There is a maximum limit of four sprites that can be displayed on one horizontal line. If this rule is violated, the four highest-priority sprites on the line are displayed normally. The fifth and subsequent sprites are not displayed on that line. Furthermore, the fifth-sprite bit in the VDP status register is set to a one, and the number of the violating fifth sprite is loaded into the status register (see Section 2.5).

Larger sprites than 8 × 8 pixels can be used if desired. The MAG and SIZE bits in VDP register 1 are used to select the various options. The options are described here:

MAG = 0, SIZE = 0: No options chosen;

MAG = 1, SIZE = 0: 8 bytes are still used in the Sprite Generator Table to describe the sprite; however, each bit in the Sprite Generator maps into 2 × 2 pixels on the TV screen, effectively doubling the size of the sprite to 16 × 16.

MAG = 0, SIZE = 1: 32 bytes are used in the Sprite Generator Table to define the sprite shape; the result is a 16 × 16 pixel sprite. The mapping of the 32 bytes into the sprite image is as shown in Figure 17. Mapping is still one bit-to-one pixel.

MAG = 1, SIZE = 1: Same as MAG = 0,SIZE = 1 except each bit now maps into a 2 × 2 pixel area, yielding a 32 × 32 sprite.

The VDP provides sprite coincidence checking. The coincidence status flag in the VDP status register is set to a '1' whenever two active sprites have '1' bits at the same screen location.

Sprite processing is terminated if the VDP finds a value of 208 ($D0_{16}$) in the vertical position field of any entry in the Sprite Attribute Table. This permits the Sprite Attribute Table to be shortened to the minimum size required; it also permits the user to blank out part or all of the sprites by simply changing one byte in VRAM.

A total of 2176 VRAM bytes are required for the Sprite Name and Pattern Generator Tables. Significantly less memory is required if all 256 possible sprite pattern definitions are not required. The Sprite Attribute Table can also be shortened as described above. The tables can be overlapped to reduce the amount of VRAM required for sprite generation. Examples of VRAM memory allocation are provided in Section 3.3.

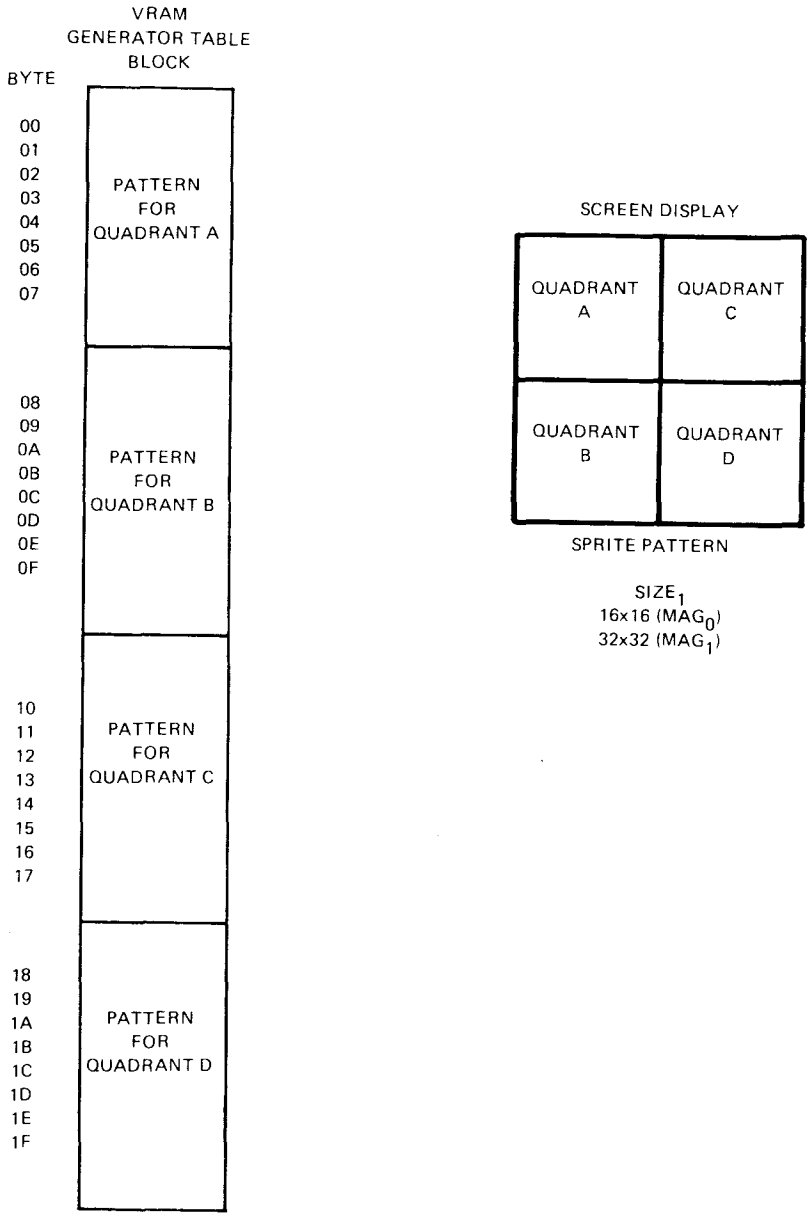


FIGURE 17. SIZE 1 SPRITE MAPPING

2.8 EXTERNAL VIDEO

As shown in Figure 6, Composite Video signals that are input to the 9918 through the External Video Input pin appear on the television screen in the plane behind the backdrop. Thus VDP-generated images on the Pattern plane and the sprite planes can be superimposed upon an incoming signal. The source of the signal may be a standard NTSC broadcast signal, the output from a NTSC compatible video-tape recorder, another VDP chip's output, or any other NTSC-compatible signal.

2.8.1 Hardware Design for External Video

The signal that is desired to appear on the external video plane should be brought in to the VDP through the External Video Input pin, while the horizontal and vertical synchronization pulses must be encoded onto the RESET pin, which is tri-level.

The RESET input has 0, 5 and 12 Volt levels. The 5 Volt level is the normal state of the RESET pin. When the pin's voltage is pulled to ground, the horizontal and vertical counters within the VDP are reset momentarily; the Composite Video output then puts out a signal corresponding to an all-black screen. When a 12-volt pulse is applied to the RESET pin (from its normal 5-Volt level), the pulse is interpreted in one of two ways: if it is a short-duration pulse, it is interpreted as a horizontal sync pulse, and the VDP's horizontal counter is reset. If it is a long-duration pulse, it is interpreted as a vertical sync signal, and both horizontal and vertical counters in the VDP are reset to 0. See the Electrical Specifications for values and tolerances on these pulse durations, as well as tolerances on the voltage levels.

2.8.2 Software Requirements

In order for the External Video Input plane to be visible, the External Video Enable bit in VDP Register 0 (EXVID) should be set to a one. The backdrop color (VDP Register 7, lower 4 bits) should be set to transparent (0). For the External Video plane to show through at a given spot on the screen, the Pattern color at that spot should be transparent, and all sprites should not be in the way (alternatively, a sprite that was in the way could be made transparent in color). Note that the External Video feature can be used in Pattern, Text or Multicolor mode.

2.9 The following figure summarizes how to generate addresses.

PATTERN GRAPHICS ADDRESS LOCATION TABLE

ADDRESS TYPE	0	1	2	3	4	5	6	7	8	9	10	11	12	13	COMMENTS
1) PATTERN NAME ADDRESS	NTB										C-ROW			C-COLUMN	PATTERN NAME TABLE BASE (VDP REG2) PATTERN CELL ROW PATTERN CELL COLUMN
2) PATTERN COLOR ADDRESS	COLB							0				NAME (0-4)			PATTERN COLOR TABLE BASE (VDP REG3) ALWAYS "0" IN BIT 8 FIVE MOST SIGNIFICANT BITS OF NAME
3) PATTERN GENERATOR ADDRESS	PGB										NAME			XXX	PATTERN GENERATOR BASE (VDP REG4) ALL 8 BITS OF NAME THREE LSB'S FORM BYTE/LINE NUMBER

TEXT MODE ADDRESS LOCATION TABLE

ADDRESS TYPE	0	1	2	3	4	5	6	7	8	9	10	11	12	13	COMMENTS
TEXT MODE NAME ADDRESS	NTB										TEXT POSITION				PATTERN NAME TABLE BASE (VDP REG2) EQUALS (TEXT CELL ROW # TIMES 40) PLUS (TEXT CELL COLUMN NUMBER)
TEXT MODE PATTERN ADDRESS	PGB										NAME			XXX	PATTERN GENERATOR BASE (VDP) REG4) NAME BYTE/LINE NUMBER

SPRITE ADDRESS LOCATION TABLE

ADDRESS TYPE	0	1	2	3	4	5	6	7	8	9	10	11	12	13	COMMENTS
SPRITE ATTRIBUTE ADDRESS	SAB										SPRITE #			XX	SPRITE ATTRIBUTE TABLE BASE (VDP REG5) SPRITE NUMBER ATTRIBUTE NUMBER: 00 FOR VERTICAL POSITION 01 FOR HORIZONTAL POSITION 10 FOR NAME 11 FOR TAG (EARLY CLOCK AND COLOR)
SIZE = 0 SPRITE PATTERN GENERATOR	SPGB										NAME			XXX	SPRITE PATTERN GENERATOR BASE (VDP REG4) NAME ATTRIBUTE OF SPRITE THREE LSB'S GIVE BYTE/LINE NUMBER
SIZE = 1 SPRITE PATTERN GENERATOR	SPGB										NAME (0-5)			XXXXX	SPRITE PATTERN GENERATOR BASE (VDP REG4) SIX MSP OF NAME SIZE = 1 SPRITE BYTE NUMBER (SEE FIGURE 19)

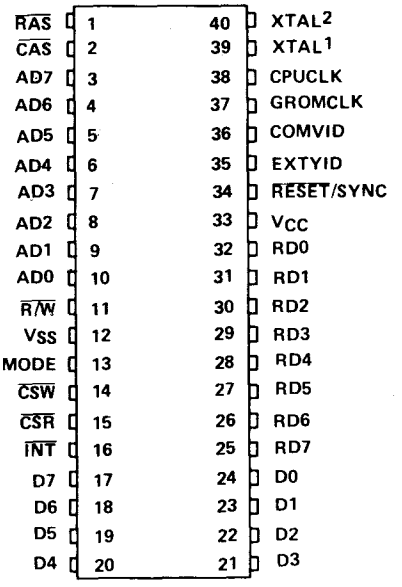
MULTICOLOR ADDRESS LOCATION TABLE

ADDRESS TYPE	0	1	2	3	4	5	6	7	8	9	10	11	12	13	COMMENTS
4) MULTICOLOR NAME ADDRESS	NTD										C-ROW			C-COLUMN	NAME TABLE PHASE (VDP REG2) PATTERN CELL ROW PATTERN CELL COLUMN
5) MULTICOLOR COLOR GENERATOR ADDRESS	PGB										NAME			XXX	PATTERN GENERATOR BASE (VDP REG4) NAME FROM NAME FETCH THREE LSB'S FORM BYTE/SQUARE ROW

FIGURE 18. PATTERN GRAPHICS ADDRESS LOCATION TABLE

2.10 VDP TERMINAL ASSIGNMENTS

SIGNATURE	TERMINAL	I/O	DESCRIPTION
D0 MSB	24	I/O	Processor data bus (D0 is the most significant bit)
D1	23	I/O	
D2	22	I/O	
D3	21	I/O	
D4	20	I/O	
D5	19	I/O	
D6	18	I/O	
D7	17	I/O	
MODE	13	I	CPU interface mode select; usually a processor address line
$\overline{CSR}$	15	I	CPU-VDP read strobe
$\overline{CSW}$	14	I	CPU-VDP write strobe
V_{CC}	33	I	+5V Supply
V_{SS}	12	I	Ground Reference
RD0 MSB	32	I	VRAM read data bus (RD0 is the most significant bit)
RD1	31	I	
RD2	30	I	
RD3	29	I	
RD4	28	I	
RD5	27	I	
RD6	26	I	
RD7	25	I	
AD0 MSB	10	O	VRAM address/data bus (multiplexed high and low order VRAM address and output dat bytes)
AD1	9	O	AD0 is the most significant bit and is used only for data and not for addressing.
AD2	8	O	
AD3	7	O	
AD4	6	O	
AD5	5	O	
AD6	4	O	
AD7	3	O	
$\overline{RAS}$	1	O	VRAM row address strobe
$\overline{CAS}$	2	O	VRAM column address strobe
R/$\overline{W}$	11	O	VRAM write strobe
$XTAL^1$, $XTAL^2$	40, 39	I	10.7+ MHz Crystal Inputs
GROMCLK	37	O	VDP output clock = XTAL/24. Typically not used.
$\overline{RESET}$/SYNC	34	I	RESET — This pin is a trilevel input pin. When it is below 0.8V, RESET initializes the VDP. Whe it is above 9.0V, RESET is the synchronizing input for external video.
EXTVID	35	I	External video input
CPUCLK	38	O	NTSC color burst frequency clock. Typically not used.
$\overline{INT}$	16	O	CPU interrupt output.
COMVID	36	O	NTSC composite video output



3.0 DEVICE APPLICATION

The following section defines recommended software techniques for controlling the TMS 9918 Video Display Processor. Also shown are examples of microprocessor code to communicate with the VDP. In these examples, the microprocessor used is a TMS 9900.

3.1 DEVICE INITIALIZATION

Section 2.1.7 outlines the function of the $\overline{\text{RESET}}$ pin on the TMS 9918. When this signal is applied, zero's are written to registers 0, 1 and 7. The effect is that the TV image goes black. In order to arrive at an active state, the following sequence of operations might be done:

1) Determine the size of VRAM, if necessary. This can be done by first assuming that the VRAM is 16K bytes, and setting the 4K/16K bit in VDP register 1 to one. An attempt to write to a high-end location, followed by a read back, will indicate whether the VRAM is 16K or only 4K.

2) Load default values into all the VDP registers; this initializes the backdrop color, the base addresses of all the tables in VRAM, the mode, etc. The value written to Register 1 should keep the screen turned off.

3) Load up the Pattern Color Table and the Pattern Generator Table; if a default character set exists, load this into the Pattern Generator. Initialize the Pattern Name Table to reflect the desired screen appearance upon screen turn-on.

4) Set up the Sprite Generator Table and the Sprite Attribute Table.

5) Turn on screen by writing to VDP register 1.

3.2 RECOMMENDED DSR'S

The simplest method of CPU communication with the VDP is to set up a root set of Device Service Routines (DSR'S). These are subroutines which, when called with the proper parameters passed to them, perform the desired I/O operation on the TMS 9918 VDP.

The circuit shown in Figure 19 illustrates a very simple interface between a 9900 CPU and the 9918 VDP. The end result is that the VDP is communicated with via the 9900 MOVB instruction (MOVE BYTE). In the circuit, the VDP's 8-bit data bus is connected to the 8 most significant bits of the 9900's 16-bit data bus. The VDP Mode pin is connected directly to the CPU's A14 address line; to generate the $\overline{\text{CSR}}$ and $\overline{\text{CSW}}$ signals, the following logic equations are used:

$$\overline{\text{CSR}} = \text{A0} * \text{A13} * \text{DBIN}$$

$$\overline{\text{CSW}} = \text{A0} * \text{WE}$$

DBIN and $\overline{\text{WE}}$ are signals from the 9900 which indicate direction on the data bus. DBIN is true when the CPU is attempting to do a read data operation, while $\overline{\text{WE}}$ is true in order to strobe out data put out on the data bus by the CPU.

A0, the most significant address line, is used as a VDP select signal. Thus, whenever the CPU is reading or writing data in the upper half of its address space (>8000 and above), the VDP is activated. All addresses above >8000, then, become VDP port addresses. Of course, in a more sophisticated design more decoding of the address lines would be used in order to avoid wasting address space.

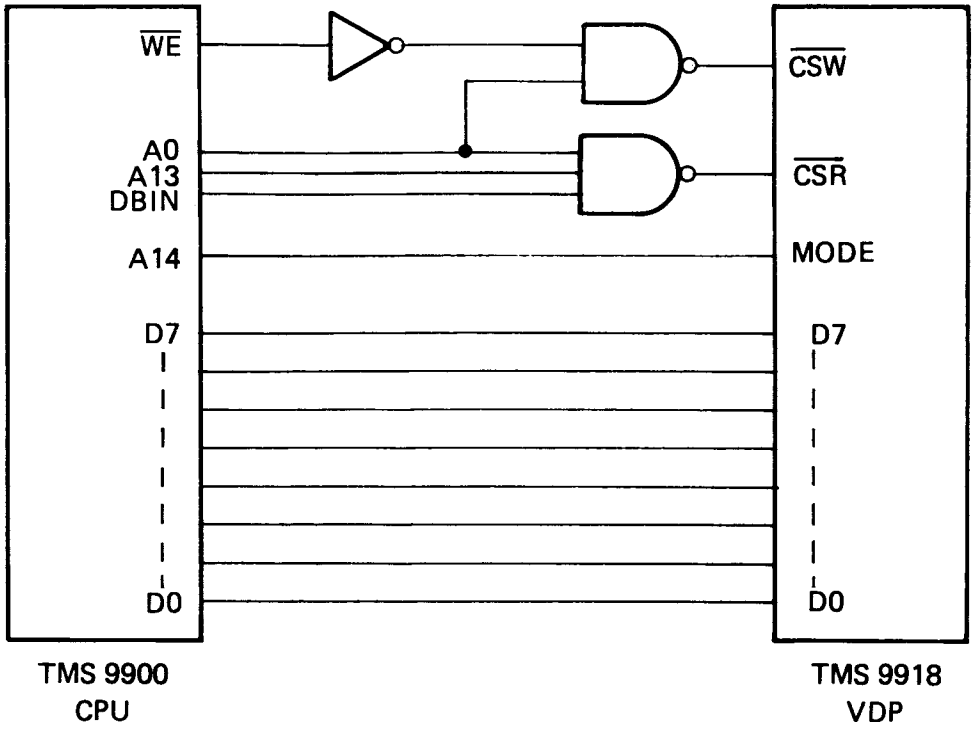


FIGURE 19. MINIMAL 9900/9918 INTERFACE

A13 is used in order to block out the read-data operaton that the 9900 does before any write-data operation. Without the A13 line being present, a spurious pulse on the $\overline{CSR}$ input to the VDP would occur before any desired pulsing of the $\overline{CSW}$ input. The programmer needs to be careful that during a write operation to the VDP, the port address must have A13 = 0 in order to block the read-before-write. When doing a read operation, A13 must be 1 in order to allow to pulse.

Referring to Table 1 and the example interface circuit in Figure 19, the following port addresses can be defined:

Operation	Symbol	$\overline{CSW}$	$\overline{CSR}$	MODE	PORT
Write data to VRAM	VDPWD	0	1	0	>8000
Write address to VRAM or Write to VDP register	VDPADR	0	1	1	>8002
Read data from VRAM	VDPRD	1	0	0	>8004
Read VDP status	VDPST	1	0	1	>8006

Now that the port addresses are defined, code can be written to do the four root operations on the VDP: Write data to VRAM, read data from VRAM, write to a VDP register, read the VDP status register, as described in Table 1.

1) Write data to VRAM:

```
MOVB <lower byte of address>,@VDPADR
MOVB <upper byte of address + 040>,@VDPADR
MOVB <data>,@VDPWD
```

2) Read data from VRAM:

```
MOVB <lower byte of address>,@VDPADR
MOVB <upper byte of address>,@VDPADR
MOVB @VDPRD,<destination>
```

3) Write data to VDP register:

```
MOVB <data>,@VDPADR
MOVB <register number + 080>,@VDPADR
```

4) Read VDP status register:

```
MOVB @VDPST,<destination>
```

Note that in the case of writing and reading data from VRAM, the last MOVB instruction in the sequences can be repeated in order to access successive locations in VRAM. Thus, in addition to the four root-level routines for VDP communication, one might add two "block-move" routines which allow fast loading and reading from tables in VRAM.

3.3 EXAMPLE VRAM ARRANGEMENTS

The following are some examples of sets of values to use in the VDP registers in order to achieve a satisfactory arrangement of tables in VRAM.

3.3.1 Pattern Mode

The following register values allow near-full usage of 4K bytes of VRAM in Pattern mode.

Register 2: >00 . . puts Pattern Name Table at 0 to >2FF;
Register 3: >0E . . puts Pattern Color Table at >380 to >39F;
Register 4: >01 . . puts Pattern Generator at >800 to >FFF;
Register 5: >06 . . puts Sprite Attribute Table at >300 to >37F;
Register 6: >00 . . puts Sprite Generator Table at 0 to >800;
only half of this is used, making the range >400 to >7FF;
Sprite names range between >80 and >FF.

Figure 20 illustrates the allocation of VRAM graphically.

3.3.2 Text Mode

Since the Text mode uses only the Pattern Name Table and the Pattern Generator Table, allocation of VRAM is trivial. Assuming 256 unique characters are required, the following allocation prevents segmentation of available VRAM:

Register 2: >02 . . puts Pattern Name Table at >800 to >BBF;
Register 4: >00 . . puts Pattern Generator Table at 0 to >7C0;

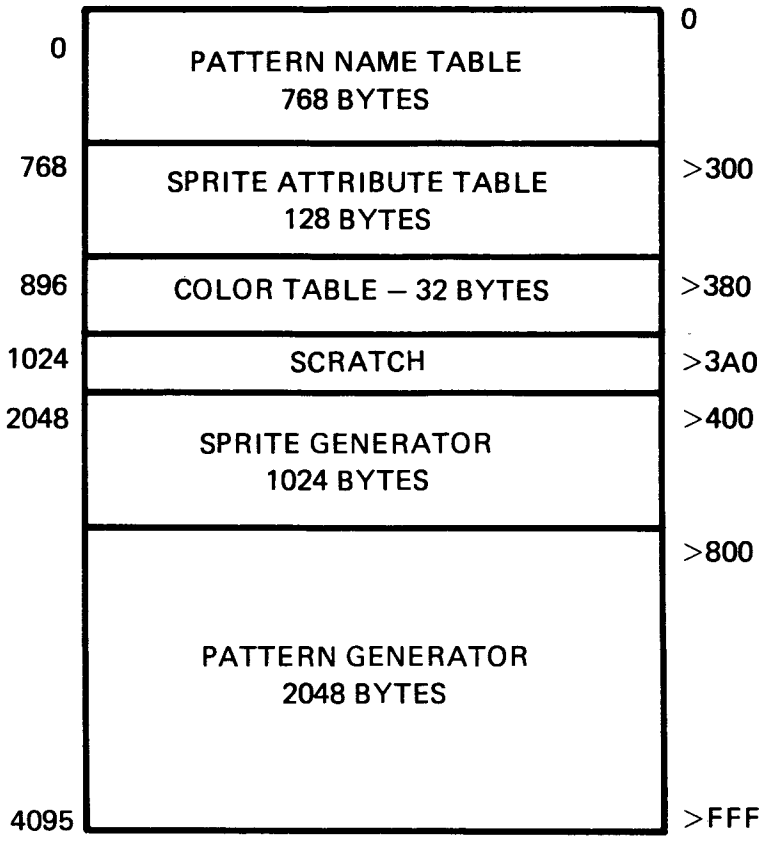


FIGURE 20. TYPICAL VRAM ALLOCATION

If only about half of the available 256 text patterns are available for use, the following allocation is suitable:

Register 2: >00 . . puts Pattern Name Table at 0 to >3BF;
Register 4: >00 . . puts Pattern Generator at 0 to >7FF;
only text patterns 120 through 255 are used, the first text pattern definition starting at >3C0.

3.3.3 Multicolor Mode

The same allocation as is used for Pattern mode (Section 3.3.1) can be used for Multicolor mode. The differences are that the Pattern Color Table is meaningless in Multicolor mode, and the Pattern Name Table is only 192 bytes long.

3.4 SPRITE MANIPULATION

Animation of objects is made easy through the use of sprites. By having the sprite's vertical and horizontal displacement values updated in real time, accurate positioning and smooth motion of the sprites is possible. For example, the following Pascal algorithm causes sprite 0 to move across the screen from left to right:

```
FOR I = 0 TO 255 DO
    BEGIN
    <write I to (Sprite Attribute Table + 1) >
    WAIT (.1 seconds)
    END;
```

In some cases it is necessary to "bleed in" a sprite from the left edge of the screen, move it across the screen, then have it "bleed off" the right edge of the screen. The following does it:

```
FOR I = 0 TO 64 DO                    . . 64 was arbitrary;
    BEGIN
    <write I to (Sprite Attribute Table + 1) >
    WAIT (.1 seconds)
    END;
(Sprite Name Table +3) = (Sprite Attribute Table + 3) AND >F
                          . . turn off Early Clock bit;

FOR I = 32 to 255 DO
    BEGIN
    <write I to (Sprite Attribute Table + 1) >
    WAIT (.1 seconds)
    END;
```

A change in the shape of the sprite can be achieved in two ways. The first method involves merely modifying the sprite generator bits in order to effect the change. The second method is to change the Sprite Name pointer, the third byte of the entry. For example, in order to simulate the rotation of a wheel on the screen, several "snapshots" of a rotating wheel could be placed into the Sprite Generator simultaneously. Then, in order to cause the wheel to rotate, the Sprite Name pointer is cycled in real time to point to the various snapshots.

Note that the MAG0, SIZE1 option and the MAG1, SIZE0 option both produce 16 × 16 pixel sprites. The difference is that the MAG bit conserves memory, while the SIZE bit yields better resolution. When 16 × 16 sprites are required, the user needs to make the decision.

Larger objects can be made up from a group of sprites. This is also the technique used in order to achieve multiple-colored objects. When motion is to occur, all the sprite horizontal and vertical counters should be incremented or decremented by the same amount.

3.5 EXTERNAL VIDEO INPUT

The following is a rough example of the use of External Video in Pattern Mode, to put subtitles along the bottom of a broadcast signal:

VDP REGISTER 0: >01 Turn on EX. VID.
VDP REGISTER 1: >40 Turn on screen, Pattern mode.
VDP REGISTER 7: >00 set backdrop to transparent.

Pattern Generator: Load with ASCII character set (see Section 2.5.1)

Pattern Color Table: Select desired colors for patterns.

Pattern Name Table: Load bytes 736 to 767 with desired ASCII values.

4.0 TMS 9918NH ELECTRICAL SPECIFICATIONS, PRELIMINARY

4.1 ABSOLUTE MAXIMUM RATING OVER OPERATING FREE-AIR TEMPERATURE RANGE (unless otherwise noted)*

Supply Voltage, V_{CC} (See Note 1) ..... −0.3 V to 20 V
All Input and Output Voltages ..... −0.3 V to 20 V
Continuous Power Dissipation ..... 1.8 W
Operating Free-Air Temperature Range ..... 0 °C to 55 °C
Storage Temperature Range ..... −55 °C to 150 °C

*Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the "Recommended Operating Conditions" section of this specification not implied. Exposure to absolute-maximum-conditions for extended periods may affect device reliability.

Note 1: Unless otherwise noted, all voltages are with respect to V_{SS}.

Note: Some r-f modulators, when used in conjunction with the TMS 9918NH, may require additional r-f filtering to eliminate video hum distortion on television screens.

4.2 RECOMMENDED OPERATION CONDITIONS

PARAMETERS	MIN	NOM	MAX	UNITS
Supply Voltage, V_{CC}	5.0		5.5	V
Supply Voltage, V_{SS}		0		V
High Level Input Voltage, V_{IH} All Pins Except Reset, XTAL	2.2			V
Low Level Input Voltage, V_{IL} All Pins			0.8	V
High Level Input Voltage Reset, V_{IHR} Reset,	3.0			V
Sync Level Input Voltage, Reset, V_{IHS}	10.0			V
Operating Free-Air Temperature	0		55	°C
High Level Input Voltage, XTAL, V_{IHX}	2.75			
Inactive Input Voltage Reset/Sync, V_{INA}	3.0		6.0	

4.3 ELECTRICAL CHARACTERISTICS OVER FULL RANGES OF RECOMMENDED OPERATING CONDITIONS

PARAMETERS		TEST CONDITIONS	MIN	TYP*	MAX	UNITS
I_i	Input Leakage Current	V_i = 0V to V_{CC} All other pins = 0V			±10	μA
I_o	Tristate Leakage Current D0—D7	V_i0V to V_{CC}			±100	μA
V_{OH}	High Level Output Voltage	I_{OH} = −400 μA	2.4			V
V_{OL}	Low Level Output Voltage for CPU Data	I_{OL} = 1.2 mA			0.6	V
V_{OLM}	Low Level Output Voltage for DRAM I/F	I_{OL} = 800 μA			0.6	V
V_{OHS}	High Level Output Voltage for RAS, CAS WRITE		2.7			V
V_{VW}	Video Voltage Level of White	V_{CC} = 5V		3.2		V
V_{VB}	Video Voltage Level of Black (blank)	V_{CC} = 5V		2.3		V
V_{VS}	Video Voltage Level of Sync	V_{CC} = 5V		1.9		V
V_{VBA}	Video Voltage Peak to Peak of Burst	V_{CC} = 5V		.5		V
V_{VR}	Video Range (White-Black)		.5			V
I_{CC}	Average Supply Current From V_{cc}	T_A = 25 °C		200	250	mA
C_i	Input Capacitance	F = 1MHz unmeasured pins at V_{SS}			10	pf
D_0—D_7	Data I/O Capacitance	F = 1MHz unmeasured pins at V_{SS}			20	pf
C_o	Output Capacitance	F = 1MHz unmeasured pins at V_{SS}			20	pf

*All typical values are at T_a = 25 °C and nominal voltages.

4.4 TIMING REQUIREMENTS OVER RECOMMENDED SUPPLY VOLTAGE RANGE OPERATING FREE-AIR TEMPERATURE RANGE

CPU — VDP Interface

PARAMETERS		MIN	TYP	MAX	UNITS
t_{ASRL}	Address set up time before CSR low		0		NS
t_{ASWL}	Address set up time before CSW low		30		NS
t_{AHWL}	Address hold time after CSW low		30		NS
t_{CSWW}	CSW pulse width, low		200		NS
t_{DSWH}	Data set up time before CSW high		100		NS
t_{CACM}	Chip select high requesting memory access until next chip select low		8		μS
t_{DHWH}	Data hold time after CSW high		30		NS
t_{CACV}	Chip select high not request memory access until next chip select low		3		μS

VDP = VRAM Interface

PARAMETERS		MIN	TYP	MAX	UNITS
t_{CT}	Memory read or write cycle time	372			NS
t_{su}(RDCL)	Input data set up time before CAS low	160			NS
t_h(RDCL)	Input data hold time after CAS low	0			NS

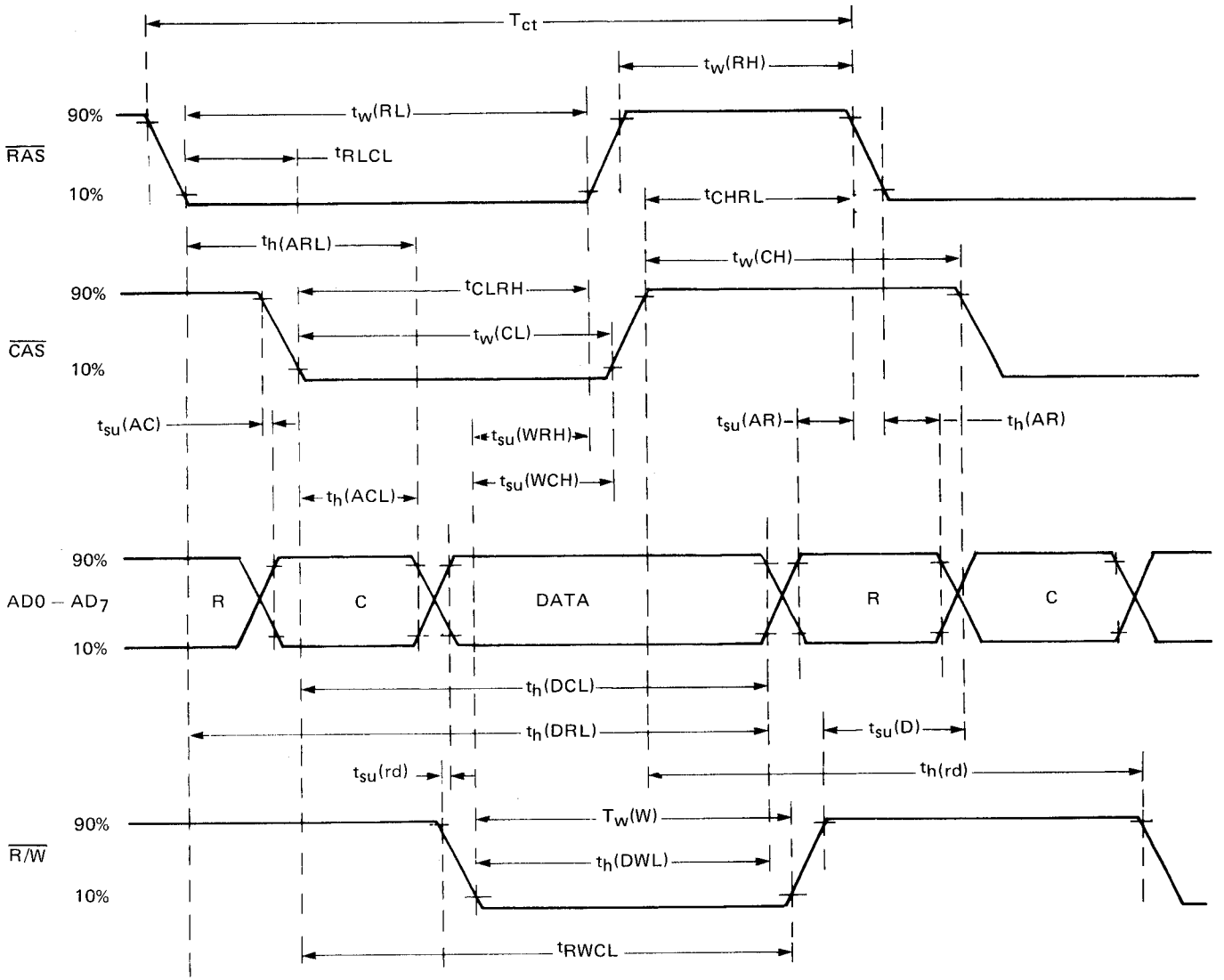
4.5 SWITCHING CHARACTERISTICS OVER FULL RANGE OF RECOMMENDED OPERATIN(CONDITIONS

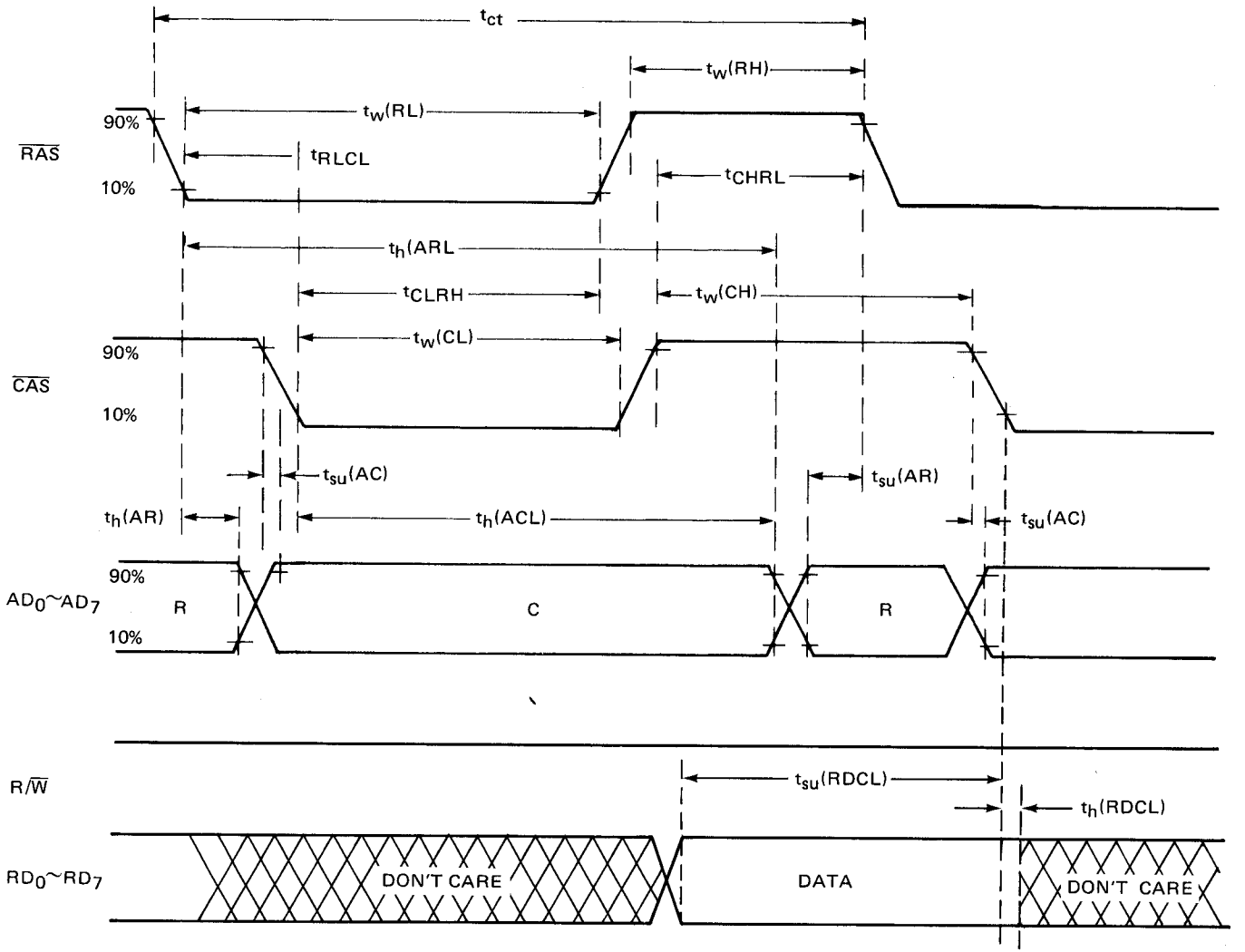
CPU — VDP Interface

PARAMETERS		TEST CONDITIONS	MIN	TYP	MAX	UNITS
t_{DARL}	Data access time from CSR	C_L = 300 pf		300		NS
t_{DDRH}	Data disable time after CSR high			200		NS
t_{DIAC}	Data invalid time after address changes			0		NS
CPU-Clock	−3 Output Clock			3.58		MHz
GROM-Clock	−24 Output Clock			447.5		KHz

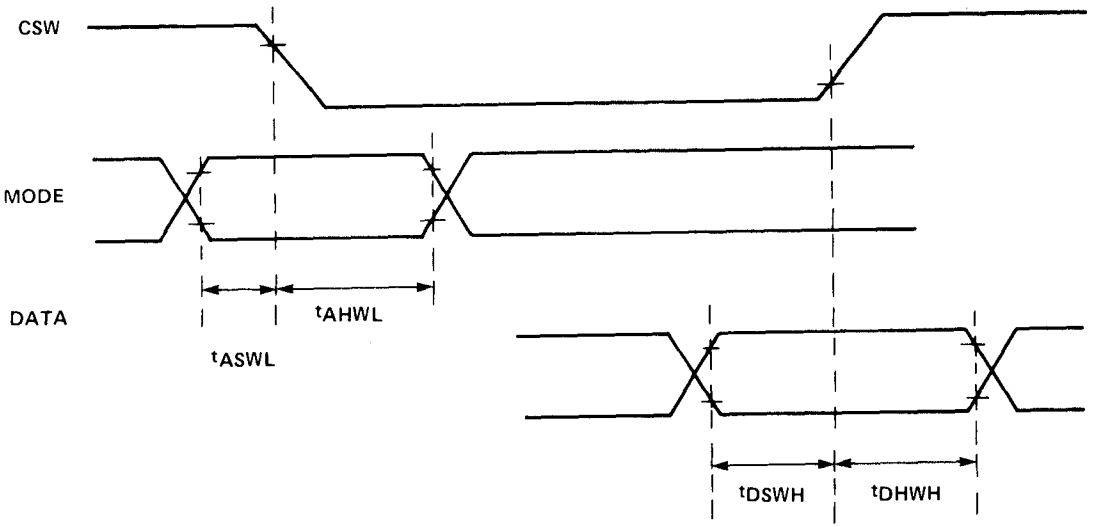
VDP — VRAM Interface

PARAMETERS		MIN	TYP	MAX	UNITS
t_w(CH)	Pulse width, column address strobe high	150			NS
t_w(CL)	Pulse width, column address strobe low	150			NS
t_w(RH)	Pulse width, row address strobe high	110			NS
t_w(RL)	Pulse width, row address strobe low	200			NS
t_w(W)	Write pulse width low	150			NS
t_{su}(AC)	Column address set up time	−25			NS
t_{su}(AR)	Row address set up time	10			NS
t_{su}(D)	Data set up time	−20			NS
t_{su}(RD)	Read command set up time	50			NS
t_{su}(WCH)	Write command set up time before CAS high	75			NS
t_{su}(WRH)	Write command set up time before RAS high	75			NS
t_h(ACL)	Column address hold time after CAS low	45			NS
t_h(AR)	Row address hold time	20			NS
t_h(ARL)	Column address hold time after RAS low	95			NS
t_h(DCL)	Data hold time after RAS low	240			NS
t_h(DRL)	Data hold time after RAS low	280			NS
t_h(DWL)	Data hold time after w low	145			NS
t_h(rd)	Read command hold time	25			NS
t_h(WCL)	Write command hold time after CAS low	245			NS
t_{CHRL}	Delay time, column address strobe high to row address strobe low	100			NS
t_{CLRH}	Delay time, column address strobe low to row address strobe high	150			NS
t_{RLCL}	Delay time, row address strobe low to column address strobe low	35			NS

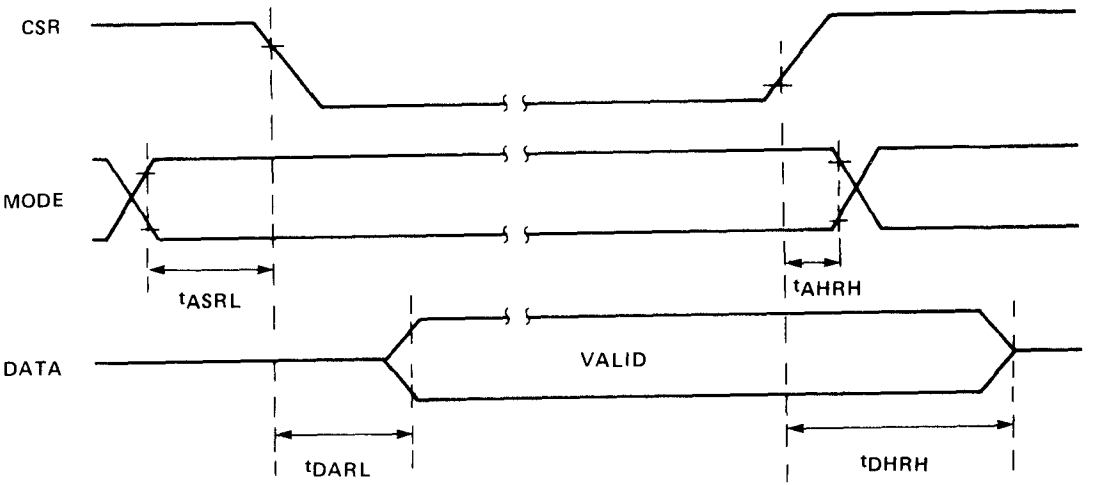

T_{ct}
$t_w(RH)$
$t_w(RL)$
t_{RLCL}
t_{CHRL}
$t_h(ARL)$
$t_w(CH)$
t_{CLRH}
$t_w(CL)$
$t_{su}(AC)$
$t_{su}(WRH)$
$t_{su}(AR)$
$t_h(AR)$
$t_{su}(WCH)$
$t_h(ACL)$
$t_h(DCL)$
$t_h(DRL)$
$t_{su}(D)$
$t_{su}(rd)$
$t_h(rd)$
$T_w(W)$
$t_h(DWL)$
t_{RWCL}
90%
10%
$\overline{RAS}$
$\overline{CAS}$
AD0 – AD_7
$\overline{R/W}$
R
C
DATA
R
C


t_{ct}
$t_{w(RH)}$
$t_{w(RL)}$
90%
$\overline{RAS}$
t_{RLCL}
10%
t_{CHRL}
$t_{h(ARL}$
t_{CLRH}
$t_{w(CH)}$
$t_{w(CL)}$
90%
$\overline{CAS}$
10%
$t_{su(AC)}$
$t_{su(AR)}$
$t_{h(AR)}$
$t_{h(ACL)}$
$t_{su(AC)}$
90%
$AD_0 \sim AD_7$
R
C
R
10%
R/$\overline{W}$
$t_{su(RDCL)}$
$t_{h(RDCL)}$
$RD_0 \sim RD_7$
DON'T CARE
DATA
DON'T CARE

WRITE CYCLE

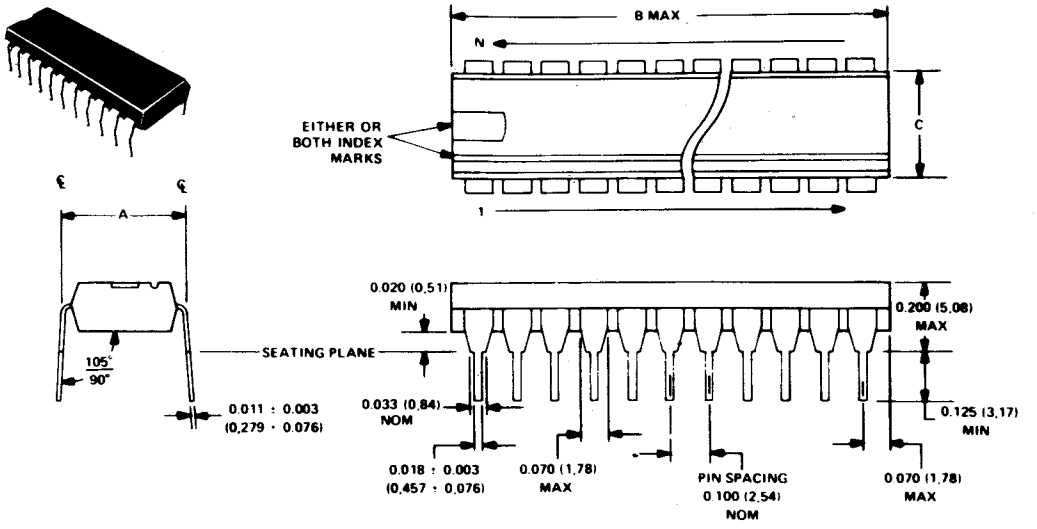


READ CYCLE



5.1 TMS 9918 — 40-PIN PLASTIC PACKAGE

plastic packages

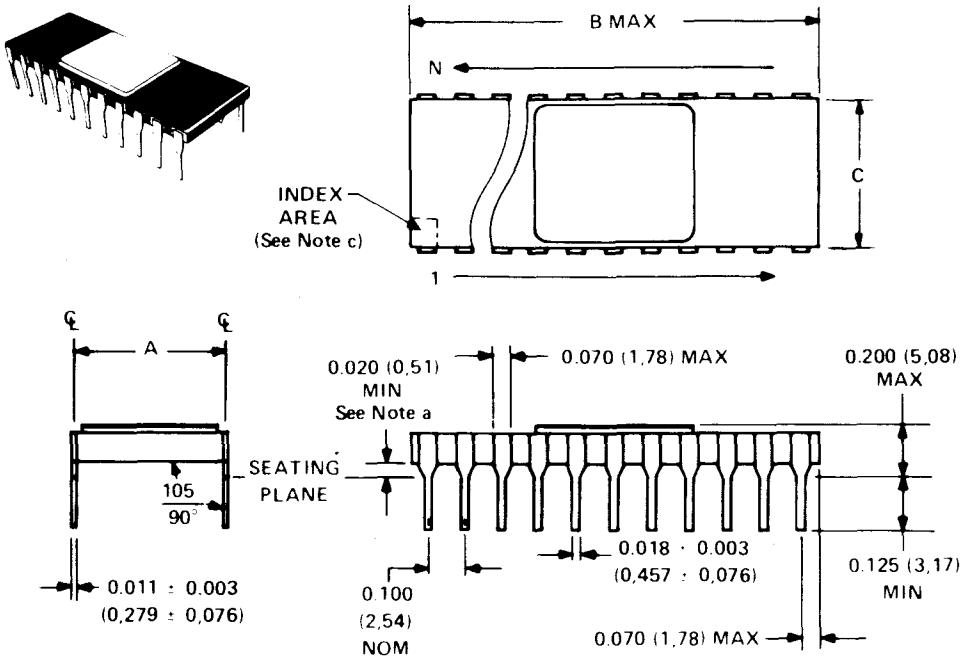


ALL LINEAR DIMENSIONS ARE IN INCHES AND PARENTHETICALLY IN MILLIMETERS. INCH DIMENSIONS GOVERN.

DIM \ PINS	8	16	18	20	22	24	28	40
A ± 0.010 (0,26)	0.300 (7,62)	0.300 (7,62)	0.300 (7,62)	0.300 (7,62)	0.400 (10,16)	0.600 (15,24)	0.600 (15,24)	0.600 (10,24)
B MAX	0.390 (9,9)	0.870 (22,1)	0.920 (23,4)	1.070 (27,2)	1.100 (28,0)	1.290 (32,8)	1.440 (36,6)	2.090 (53,1)
C NOM	0.250 (6,4)	0.250 (6,4)	0.250 (6,4)	0.265 (6,7)	0.350 (8,9)	0.550 (14,0)	0.550 (14,0)	0.550 (14,0)

5.2 TMS 9918 — 40-PIN CERAMIC PACKAGE

ceramic packages with side-brazed leads and metal or epoxy or glass lid seal



NOTES: a. This minimum spacing is valid for printed circuit board mounting with 0.033 (0,84) diameter holes for the leads.
b. All linear dimensions are in inches and parenthetically in millimeters. Inch dimensions govern.
c. The index is placed in this area to identify pin 1 and to provide other information as follows:

1	Pin 1 connected to chip-mounting pad.
△XX	Pin XX connected to chip-mounting pad.
•	No connection to chip-mounting pad.

Other symbols may indicate any combination of up to 4 pins connected to the chip-mounting pad.

DIM \ PINS	16	18	20	22	24	28	40
A ± 0.010 (0,26)	0.300 (7,62)	0.300 (7,62)	0.300 (7,62)	0.400 (10,16)	0.600 (15,24)	0.600 (15,24)	0.600 (15,24)
B MAX	0.840 (21,4)	0.910 (23,1)	1.020 (25,9)	1.100 (28,0)	1.290 (32,8)	1.415 (36,0)	2.020 (51,3)
C NOM	0.290 (7,4)	0.290 (7,4)	0.290 (7,4)	0.390 (9,9)	0.590 (15,0)	0.590 (15,0)	0.590 (15,0)

APPENDIX A
VRAM TABLE ADDRESSING EXAMPLES

The following three examples demonstrate how typical VRAM addresses are computed by the VDP. In each example, a scaling bit is represented by 'X'; all scaling bits are equivalent to zeroes. Since only part of a term might be used in the address calculation, the bits which are used are underscored. Bits which are not underscored are not used in the calculation and are also equivalent to zeroes.

A.1 PATTERN GRAPHICS MODE

This example calculates the addresses of the name table, color table, and generator table entries for the pattern cell in pattern row 2 and column 29 (pattern cell 92). The addresses of the Sprite Attribute Table and Generator Table for sprite 15 are also calculated.

Note: @ (xxx) implies the contents of location xxx.

The VDP register contents are:

register	value (base 16)
0	00
1	E0
2	01
3	1C
4	01
5	0F
6	00
7	0C

PATTERN NAME TABLE

ADDRESS =	0	0	0	1	X	X	X	X	X	X	X	X	X	X	base address
					0	0	0	1	0	X	X	X	X	X	row number
										1	1	1	0	1	column number
	0	0	0	1	0	0	0	1	0	1	1	1	0	1	= $45D_{16}$

COLOR TABLE

@ ($45D_{16}$) = $F5_{16}$ = pattern name

ADDRESS =	0	0	0	1	1	1	0	0	X	X	X	X	X	X			base address
										1	1	1	1	0	1	0	1	pattern name
	0	0	0	1	1	1	0	0	0	1	1	1	1	0	= $71E_{16}$		

PATTERN GENERATOR TABLE

ADDRESS =	0	0	1	X	X	X	X	X	X	X	X	X	X	X	base address
				1	1	1	1	0	1	0	1	X	X	X	pattern name
	0	0	1	1	1	1	1	0	1	0	1	0	0	0	= $FA8_{16}$

SPRITE ATTRIBUTE TABLE

```
ADDRESS = 0  0   0  1  1  1   1  X  X  X   X  X  X  X      base address
                             0  1  1   1  1  X  X      sprite number
          ------------------------------------------
          0  0   0  1  1  1   1  0  1  1   1  1  0  0  = 7BC16
```

SPRITE PATTERN GENERATOR TABLE

@ ($7BE_{16}$) = 13_{16} = sprite name ($SIZE_0$ must be on an 8 byte boundary)

```
ADDRESS = 0  0   0  X  X  X   X  X  X  X   X  X  X  X      base address
                 0  0  0   1  0  0  1   1  X  X  X      sprite name
          ------------------------------------------
          0  0   0  0  0  0   1  0  0  1   1  0  0  0  = 9816
```

A.2 TEXT GRAPHICS MODE

This example calculates the addresses for the Text Name Table and the Text Pattern Generator Table entries for the pattern cell in row 20 and column 35 (pattern cell 834).

The VDP register contents are:

register	value (base 16)
0	00
1	F0
2	02
3	00
4	00
5	00
6	00
7	96

TEXT NAME TABLE

```
ADDRESS = 0   0   1  0  X  X   X  X  X  X   X  X  X  X      base address
                     1  1   0  1  0  0   0  0  1  0      pattern cell
          ------------------------------------------
          0   0   1  0  1  1   0  1  0  0   0  0  1  0  = B4216   number
```

TEXT PATTERN GENERATOR TABLE

@ ($B42_{16}$) = 79_{16} = text pattern name

```
ADDRESS = 0   0   0  X  X  X   X  X  X  X   X  X  X  X      base address
                 0  1  1   1  1  0  0   1  X  X  X      pattern name
          ------------------------------------------
          0   0   0  0  1  1   1  1  0  0   1  0  0  0  = 3C816
```

This example calculates the addresses of the name table and color generator table entries for the displaying the multi-color block 65 in Figure 11, assuming the name table is organized as suggested in Section 2.7.2.

The VDP register contents are:

register	value (base 16)
0	01
1	0B
2	02
3	00
4	00
5	0C
6	01
7	0B

COLOR GENERATOR TABLE

41_{16} = multicolor name

ADDRESS =	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	base address
				0	1	0	0	0	0	0	1	X	X	X		multicolor name
												X	X	X		square row
	0	0	0	0	1	0	0	0	0	0	1	0	0	0	$= 41_{16}$	

SPRITE ATTRIBUTE TABLE

ADDRESS =	0	0	0	1	1	0	0	X	X	X	X	X	X	X		base address
								1	1	0	1	1	X	X		sprite number
	0	0	0	1	1	0	0	1	1	0	1	1	0	0	$= 66C_{16}$	

SPRITE PATTERN GENERATOR TABLE

$(66C_{16}) = E4_{16}$ = sprite name ($SIZE_1$ must be on a 32 byte boundary)

ADDRESS =	0	0	1	X	X	X	X	X	X	X	X	X	X	X	X	base address
				1	0	1	1	0	1	0	0	X	X	X		sprite name
	0	0	1	1	0	1	1	0	1	0	0	0	0	0	$= DA0_{16}$	

APPENDIX B
VRAM MEMORY ALLOCATION EXAMPLES

The following three examples demonstrate how VRAM can be allocated. In each example, it is assumed that the video display generation tables are in the lower 4K bytes of VRAM. The tables may, nevertheless, be in higher memory if desired.

B.1 PATTERN GRAPHICS MODE

The VDP register contents are:

register	value (base 16)
0	00
1	E0
2	01
3	1C
4	01
5	0F
6	00
7	0C

ADDRESS RANGE	FUNCTION
0000 - 03FF	sprite pattern generator table
0400 - 06FF	pattern name table
0700 - 071F	pattern color table
0720 - 077F	not allocated — CPU scratchpad
0780 - 07FF	sprite attribute table
0800 - 0FFF	pattern generator table
1000 - 3FFF	not allocated — CPU scratchpad (if populated)

B.2 TEXT GRAPHICS MODE

The VDP register contents are:

register	value (base 16)
0	00
1	F0
2	02
3	00
4	00
5	00
6	00
7	96

ADDRESS RANGE	FUNCTION
0000 - 07FF	text pattern generator table
0800 - 0BBF	text name table
0BC0 - 0FFF	not allocated — CPU scratchpad
1000 - 3FFF	not allocated — CPU scratchpad (if populated)

B.3 MULTICOLOR GRAPHICS MODE

The VDP register contents are:

register	value (base 16)
0	01
1	0B
2	02
3	00
4	00
5	0C
6	01
7	0B

ADDRESS RANGE	FUNCTION
0000 - 05FF	multicolor color generator table
0600 - 067F	sprite attribute table
0680 - 07FF	not allocated — CPU scratchpad
0800 - 08BF	multicolor name table
08C0 - 0FFF	sprite pattern generator table
1000 - 3FFF	not allocated — CPU scratchpad (if populated)

APPENDIX C

Clock Characteristics

The TMS 9918 has internal 6 phase clock systems. This is driven by either an external crystal circuit or by a TTL compatible inputs.

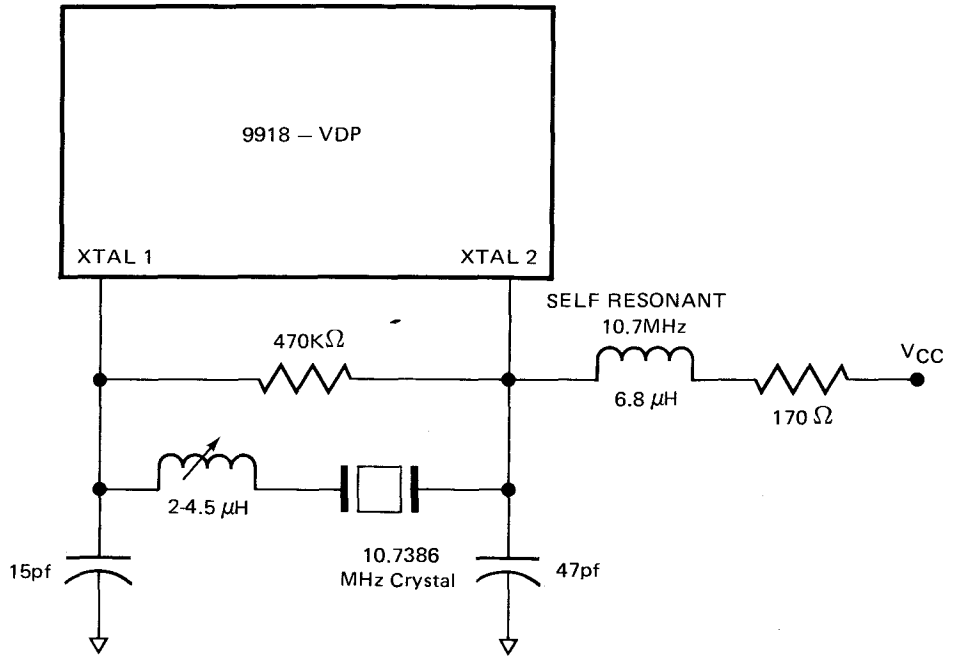


Figure 6.1 Crystal Oscillator Circuit

Internal Crystal Oscillator

The External circuit to interface to the internal oscillator is illustrated in Figure 6.1. This circuit generates maximum stability of duty cycle and phase. The crystal should be of the parallel resonance type.

External Clock

The external drive waveforms to the VDP are shown in Figure 6.2. Duty cycle should be centered around 50%.

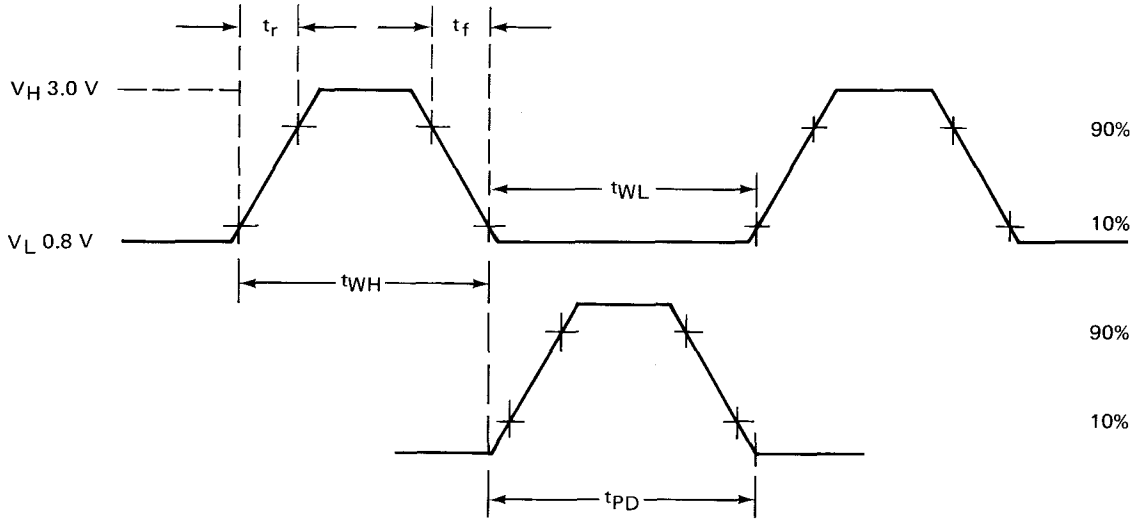


Figure 6.2. External Clock Waveform

PARAMETERS		MIN	TYP	MAX	UNIT
fext	External Source Frequency		10.738		MHz
VH	External Source High Level	3.0			V
VL	External Source Low Level			0.8	V
tr/Tf	External Source Rise/Fall Time		10		ns
tWH	External Source High Level Pulse Width	42	47	52	ns
tWL	External Source Low Level Pulse Width	42	47	52	ns
tPD	External Source Phase Delay from Xtal 1 Falling Edge to Xtal 2 Falling Edge	42	47	52	ns

External Delay Requirements

The external delay should be introduced on the R/W and CAS pins in order to properly interface standard dynamic RAMS (4027 type 4K or 4116 type 16K RAM). The requirements are 15nsec and 10 nsec in the R/W line and CAS line respectively between VDP and VRAMS, e.g., as shown in figure 6.3. 74LS04 is recommended for the delay.

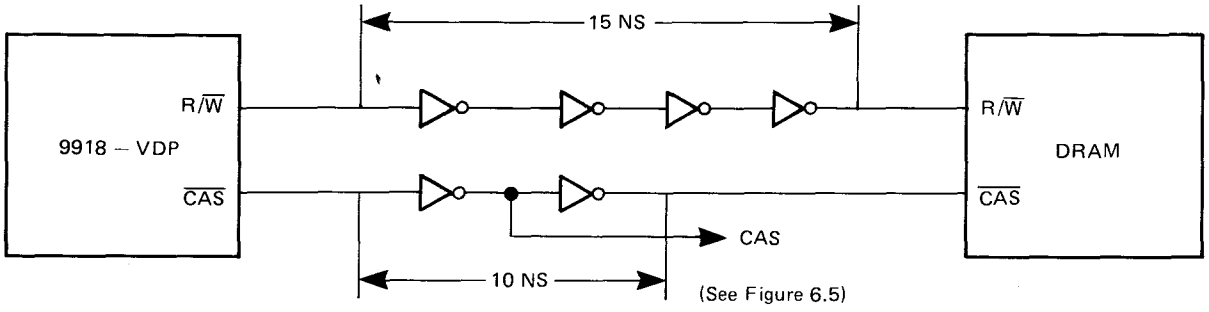


Figure 6.3. External Delay Circuit

Composite Video Pull-Down Circuit

The internal output buffer device on composite video pin is a source follower MOS transistor which requires an external pulldown resistor to V_{SS} as shown in figure 6.4. Typically 1K resistor is recommended to provide 1.9V of sync level.

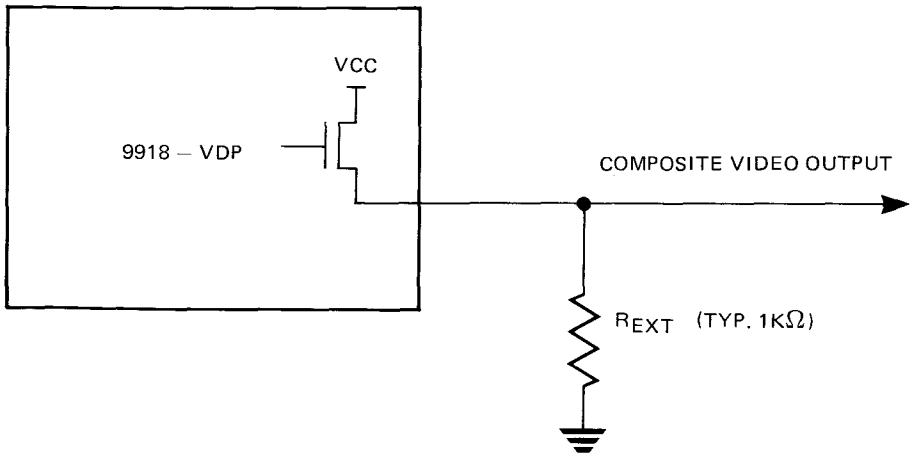


Figure 6.4. Composite Video Pull-Down Circuit

4116 To VDP Data Interface Requirement

If 4116 type RAMs are used, it is required to add latones gated by CAS between 4116 and VDP data lines as shown in figure 6.5. 74LS373 or 74LS363 is recommended.

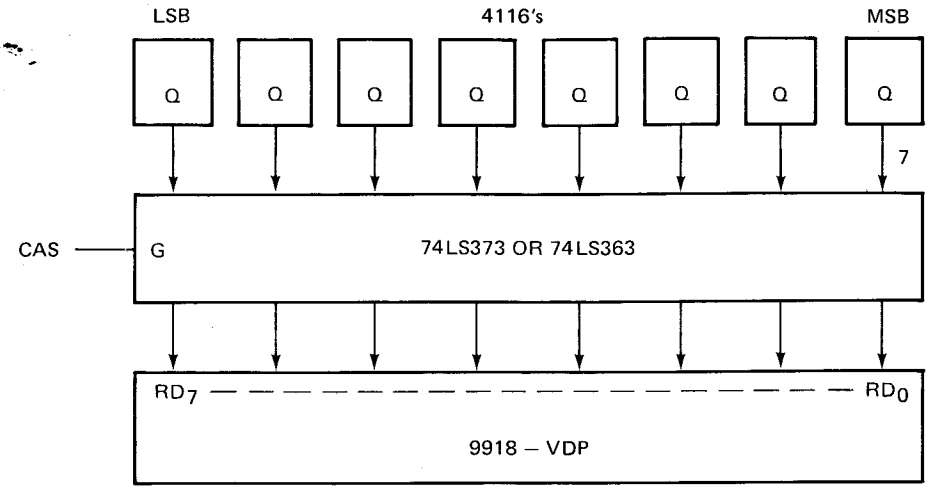


Figure 6.5. 4116 – VDP Data Bus Interface