TEXAS INSTRUMENTS

# 9900

# TMS9918A/TMS9928A/TMS9929A Video Display Processors

MICROPROCESSOR SERIES™

## 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.

Copyright © 1982
Texas Instruments Incorporated

## TABLE OF CONTENTS


| SECTION | | PAGE |
|---|---|---|
| 1. | INTRODUCTION | 1-1 |
| | 1.1 Description | 1-1 |
| | 1.2 Features | 1-1 |
| | 1.3 Typical Applications | 1-2 |
| | 1.4 Acronyms and Glossary | 1-3 |
| 2. | ARCHITECTURE | 2-1 |
| | 2.1 CPU Interface | 2-1 |
| | 2.1.1 CPU Interface Control Signals | 2-1 |
| | 2.1.2 CPU Write to VDP Register | 2-1 |
| | 2.1.3 CPU Write to VRAM | 2-3 |
| | 2.1.4 CPU Read from VDP Status Register | 2-4 |
| | 2.1.5 CPU Read from VRAM | 2-4 |
| | 2.1.6 VDP Interrupt | 2-5 |
| | 2.1.7 VDP Initialization | 2-5 |
| | 2.2 Write-Only Registers | 2-5 |
| | 2.2.1 Register 0 | 2-5 |
| | 2.2.2 Register 1 | 2-5 |
| | 2.2.3 Register 2 | 2-7 |
| | 2.2.4 Register 3 | 2-7 |
| | 2.2.5 Register 4 | 2-7 |
| | 2.2.6 Register 5 | 2-7 |
| | 2.2.7 Register 6 | 2-7 |
| | 2.2.8 Register 7 | 2-7 |
| | 2.2.9 Setup Values for VDP Registers 2 through 6 | 2-7 |
| | 2.3 Status Register | 2-11 |
| | 2.3.1 Interrupt Flag (F) | 2-11 |
| | 2.3.2 Coincidence Flag (C) | 2-11 |
| | 2.3.3 Fifth Sprite Flag (5S) and Number | 2-11 |
| | 2.4 Video Display Modes | 2-12 |
| | 2.4.1 Graphics I Mode | 2-17 |
| | 2.4.2 Graphics II Mode | 2-19 |
| | 2.4.3 Multicolor Mode | 2-21 |
| | 2.4.4 Text Mode | 2-23 |
| | 2.4.5 Sprites | 2-25 |
| | 2.4.6 A Step-by-Step Approach to Create Patterns and Sprites | 2-29 |
| 3. | VDP INTERFACES AND OPERATION | 3-1 |
| | 3.1 VDP/VRAM Interface | 3-1 |
| | 3.1.1 VRAM Interface Control Signals | 3-1 |
| | 3.1.2 VRAM Memory Types | 3-1 |
| | 3.1.3 VDP to DRAM Address Connections | 3-1 |
| | 3.2 VRAM Memory Address Derivation | 3-3 |
| | 3.3 VRAM Addressing Example | 3-4 |
| | 3.4 Monitor Interfaces | 3-5 |
| | 3.4.1 TMS9918A Monitor Interface | 3-5 |
| | 3.4.2 TMS9928A/9929A Monitor Interface | 3-6 |
| | 3.5 TMS9918A External VDP Operation | 3-7 |
| | 3.6 Oscillator and Clock Generation | 3-8 |
| | 3.6.1 TMS9918A Color Phase Generation | 3-8 |
| | 3.6.2 Video Sync and Control Generation | 3-8 |

## TABLE OF CONTENTS (continued)


| SECTION | | PAGE |
|---|---|---|
| 3.7 | VDP Terminal Assignments | 3-9 |
| 3.7.1 | TMS9918A Terminal Assignments | 3-9 |
| 3.7.2 | TMS9928A/9929A Terminal Assignments | 3-11 |
| 3.7.3 | TMS9918A/9928A/9929A Crystals | 3-12 |
| 4. | DEVICE APPLICATIONS | 4-1 |
| 4.1 | VDP to TMS9900 Interface | 4-1 |
| 4.2 | TMS9918A/9928A/9929A Interface | |
| 4.2.1 | TM990 (TMS9918A/9928A/9929A) Parts List | 4-4 |
| 4.2.2 | Composite Video Output | 4-4 |
| 4.2.3 | Oscillator and Timing | 4-4 |
| 4.2.4 | VRAM Connections | 4-5 |
| 4.3 | VDP Initialization | 4-5 |
| 4.4 | Typical Software Program | 4-6 |
| 4.4.1 | General | 4-6 |
| 4.5 | TMS9900 Software SubRoutines | 4-6 |
| 5. | TMS9918A/9928A/9929A ELECTRICAL SPECIFICATIONS | 5-1 |
| 5.1 | Absolute Maximum Ratings | 5-1 |
| 5.2 | Recommended Operating Conditions | 5-1 |
| 5.3 | Electrical Characteristics | 5-2 |
| 5.4 | Timing Requirements | 5-4 |
| 5.5 | Switching Characteristics | 5-5 |


## LIST OF APPENDICES


| APPENDIX | | PAGE |
|---|---|---|
| A | ASCII Character Set | A-1 |
| B | Choosing VRAM Memory | B-1 |
| C | Pattern and Screen Worksheets | C-1 |

## LIST OF ILLUSTRATIONS


| FIGURE | | PAGE |
|---|---|---|
| 1-1 | System Block Diagram | 1-2 |
| 2-1 | VDP Block Diagram | 2-2 |
| 2-2 | VDP to CPU Interface | 2-3 |
| 2-3 | VDP Registers | 2-6 |
| 2-4 | VDP Display Planes (Definition) | 2-12 |
| 2-5 | VDP Display Planes (First 32 Planes) | 2-13 |
| 2-6 | TMS9928A/9929A Signal Waveforms for Multiple VDP Operation | 2-14 |
| 2-7 | Using Color Difference Signals to Mix External Color Difference Type Source | 2-15 |
| 2-8 | Using Color Difference Signals to Mix External Video Sources | 2-15 |
| 2-9 | Pattern Graphics Name Table Mapping | 2-17 |
| 2-10 | Graphics I Mode Mapping | 2-18 |
| 2-11 | Pattern Display Mapping | 2-18 |
| 2-12 | Graphics II Mode Mapping | 2-20 |
| 2-13 | Pattern Display Mapping | 2-21 |
| 2-14 | Multicolor List Mapping | 2-21 |
| 2-15 | Multicolor Block Display | 2-22 |
| 2-16 | Multicolor Mode Mapping | 2-23 |
| 2-17 | Text Mode Name Table Pattern Positions | 2-23 |
| 2-18 | Mapping of VRAM into the Pattern Plane in Text Mode | 2-24 |
| 2-19 | Sprite Attribute Table Entry | 2-25 |
| 2-20 | Sprite Mapping | 2-26 |
| 2-21 | Size 1 Sprite Mapping | 2-28 |
| 3-1 | VRAM Interface | 3-2 |
| 3-2 | VDP-VRAM Memory Allocation | 3-5 |
| 3-3 | Composite Video Pull-Down Circuit | 3-5 |
| 3-4 | Use of TMS9928A/9929A with Different Monitors | 3-6 |
| 3-5 | Cascading Two TMS9918A VDPs | 3-7 |
| 3-6 | Cascading Two TMS9918A/9929A VDPs | 3-7 |
| 4-1 | Minimum System Interface to TMS9900 | 4-1 |
| 4-2 | TMS9918A/9928A/9929A Interface | 4-2 |
| 4-3 | TM990 (TMS9918A/9928A/9929A) Demo Board | 4-3 |
| 4-4 | RF Modulator Connection | 4-4 |
| 4-5 | External Frequency Source | 4-5 |
| 4-6 | VDP Register Initialization Procedure | 4-5 |
| 5-1 | Load Circuit for COMVID (All Devices) and R-Y, Y, B-Y Switching Characteristics (TMS9928A/9929A) | 5-9 |
| 5-2 | Load Circuits for All Outputs Except COMVID, R-Y, Y, B-Y | 5-9 |
| 5-3 | CPU-VDP Write Cycle (All Devices) | 5-10 |
| 5-4 | CPU-VDP Read Cycle (All Devices) | 5-10 |
| 5-5 | VRAM Write Cycle | 5-11 |
| 5-6 | VRAM Read Cycle | 5-11 |
| 5-7 | External Clock Timing Waveform | 5-12 |
| 5-8 | TMS9918A COMVID Horizontal Timing | 5-12 |
| 5-9 | TMS9918A Vertical Timing | 5-12 |

## LIST OF ILLUSTRATIONS (continued)


| FIGURE | | PAGE |
|---|---|---|
| 5-10 | TMS9928A/9929A Y Horizontal Timing | 5-13 |
| 5-11 | TMS9928A/9929A R-Y Horizontal Timing | 5-13 |
| 5-12 | TMS9928A/9929A B-Y Horizontal Timing | 5-14 |
| 5-13 | TMS9929A Vertical Timing | 5-14 |


## LIST OF TABLES


| TABLE | | PAGE |
|---|---|---|
| 2-1 | CPU/VDP Data Transfers | 2-3 |
| 2-2 | Memory Access Delay Times | 2-4 |
| 2-3 | Color Assignments | 2-17 |
| 2-4 | Graphics I Mode Color Table | 2-19 |
| 2-5 | Sprite Pattern Formats | 2-25 |
| 3-1 | VDP to DRAM Address Connections | 3-1 |
| 3-2 | Pattern Graphics Address Location Tables | 3-3 |
| 3-3 | Screen Display Parameters | 3-8 |
| 4-1 | VDP Port Addresses for Figure 4-1 | 4-1 |

# 1. INTRODUCTION

## 1.1 Description

The TMS9918A/9928A/9929A video display processors (VDP) are N-channel MOS LSI devices used in video systems where data display on a raster-scanned home color television set or color monitor is desired. These devices generate all necessary video, control, and synchronization signals and also control the storage, retrieval, and refresh of display data in the dynamic screen refresh memory. The interfaces to the microprocessor, refresh memory, and the TV require a minimum of additional electronics for the TMS9918A.

In Section 1.4, there is a list of acronyms and a glossary of terms used in this manual.

The TMS9928A/9929A VDPs are functionally identical to the TMS9918A except that the NTSC color encoding circuitry has been removed and replaced with luminance and color difference signals. The TMS9918A is pin-for-pin compatible with the TMS9928A/9929A, except for three pins, the composite video output, the external video input and the CPU clock output. These pins are replaced with the Black/White luminance and composite sync (Y) output and two color difference pins, Blue (B-Y) and Red (R-Y) outputs, respectively. The color difference outputs allow the user to generate Red-Green-Blue (R-G-B) drive for direct color gun control, or composite video for use with NTSC or PAL video color monitor. However, to connect these three outputs to a R-G-B or monitor requires additional R-G-B or encoder circuitry.

The TMS9918A/9928A have a 525-line format for U.S. televisions while the TMS9929A has a 625-line format for use with the European PAL system.

The VDP has four video display modes: Graphics I, Graphics II, Multicolor and Text mode. The Text mode provides twenty-four 40-character rows in two colors and is intended to maximize the capacity of the TV screen to display alphanumeric character. The Multicolor mode provides an unrestricted 64 × 48 color-dot display employing 15 colors plus transparent. The Graphics I mode provides a 256 × 192 pixel display for generating pattern graphics in 15 colors plus transparent. The Graphics II mode is an enhancement of Graphics I mode, allowing it to generate more complex color and pattern displays. The four video display modes are described in detail in Section 2.4.

The video display consists of 35 planes: external VDP, backdrop, pattern plane, and 32 Sprite Planes. The planes are vertically stacked with the external VDP being the bottom or innermost plane. The backdrop plane is the next plane followed by the pattern plane that contains Graphics I and Graphics II patterns with the 32 Sprite Planes as the top planes.

The TMS9918A/9928A/9929A VDPs use either a 4K, 8K, or 16K-type low-cost dynamic memory (TMS4027, TMS4108, TMS4116) for storage of the display parameters.

The TMS9918A, TMS9928A, and TMS9929A interface identically to the host microprocessor making them software compatible. Thus, all references to VDP in this document apply to all three devices, except where noted.

## 1.2 FEATURES

- Single-chip solution for interfacing color TVs (excluding Random-Access Memory (RAM) and Radio Frequency (RF) modulator (TMS9918A only)
- 256 × 192 resolution on TV screen
- 15 unique colors plus transparent
- General 8-bit bidirectional interface to Central Processor Unit (CPU)
- Direct wiring to 4K, 8K, or 16K dynamic RAM memories
- Automatic and transparent refresh of dynamic RAMs
- Multiple VDP systems capability
- External VDP input capability (TMS9918A only)
- Composite video output (TMS9918A only)
- Unique planar representation for 3D simulation
- Standard 40-pin package
- Color difference outputs allow RGB drive — TMS9928A/9929A

## 1.3 TYPICAL APPLICATIONS

- Color computer terminals
- Home computers
- Drafting/design aids
- Teaching aids
- Industrial process monitoring
- Home educational systems
- Animation aids
- European 625-line TV (TMS9929A only)

The following example of a typical application may help introduce the user to the TMS9918A VDP. Figure 1-1 is a block diagram of a typical application. Each of the concepts presented in the example is described more fully in later sections of this manual.

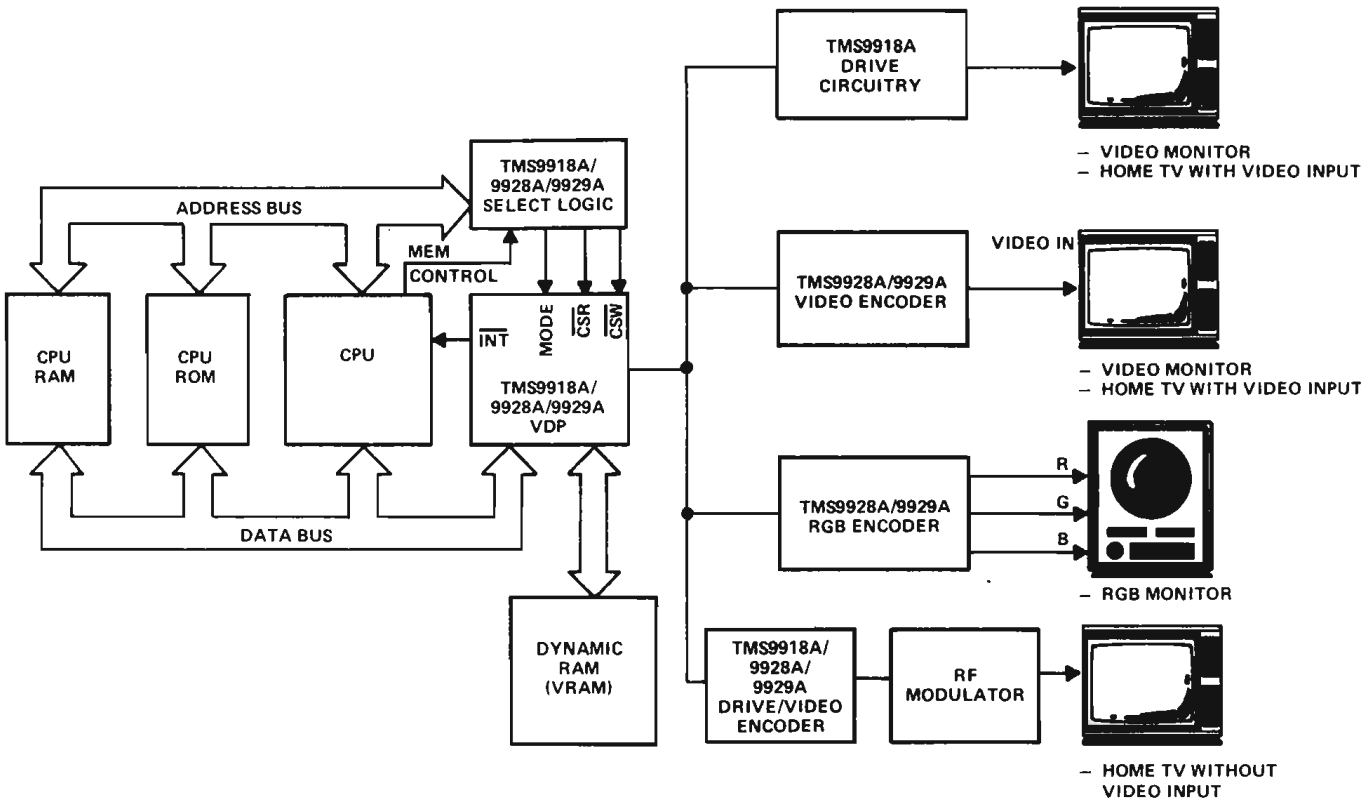



**FIGURE 1-1 – SYSTEM BLOCK DIAGRAM**

The VDP basically has three interfaces: CPU, color monitor, and dynamic refresh RAM (VRAM), the contents of which define the TV image. The TMS9918A also has eight 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 from 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.

The dynamic RAM interface consists of direct wiring of eight 4K × 1, 8K × 1, or 16K × 1 dynamic RAS/CAS-type RAMs to the VDP. The amount of RAM required is dependent upon the features selected for use in the application.

The interface to the monitor can consist of either wiring the TMS9918A's composite video output pin (suitably buffered) to the input of a color or black-and-white monitor, or using an appropriate RF modulator to feed the signal into a TV antenna terminal. The TMS9928A/9929A require additional encoder circuitry to interface to a RGB or to a composite video monitor.

The VDP operates in four modes, and each one can affect the way the VRAM is mapped onto the television screen. In Graphics I and II modes, characters are mapped onto the screen in 8 × 8 pixel blocks, yielding 24 lines of 32 blocks (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 picture elements (pixels), all of one solid color. In addition to these, sprites can be superimposed onto the television image in Graphics I, II, and Multicolor mode. Furthermore, signals entering the TMS9918A through the external VDP input can be used as a background to the TMS9918A.

## ACRONYMS AND GLOSSARY

| | |
|---|---|
| B-Y | Blue color difference output |
| COMVID (Composite Video) | Contains luminance, chrominance and all sync pulse necessary for horizontal and vertical timing |
| CAS | Column Address Strobe |
| CPU | Central Processor Unit |
| CSR | CPU from VDP read select |
| CSW | CPU to VDP write select |
| CPUCLK | XTAL — 3 |
| GROMCLK | XTAL — 24 |
| LSB | Least Significant Bit |
| LSI | Large Scale Integration |
| MOS | Metal Oxide Semiconductor |
| MHz | Megahertz |
| MSB | Most Significant Bit |
| NTSC | National Television Standards Committee which specifies television signal standards for the USA |
| PAL | Phase Alternating Line |
| Pixel | Picture Element — the smallest point on the TV screen that can be independently controlled. |
| RAM | Random-Access Memory |
| RAS | Row-Address Strobe |
| RASTER | The area in which an image is reproduced |
| RF | Radio Frequency |
| R-G-B | Red-Green-Blue |
| ROM | Read-Only Memory |
| R/W | Read/Write |
| R-Y | Red color difference output |
| 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 |
| VDP | Video Display Processor |
| VRAM | Video RAM; refers to the dynamic RAMs that connect to the VDP and whose contents define the TV image |
| Y | Black/white luminance and composite sync |

# 2. ARCHITECTURE

The TMS9918A video display processor (VDP) is designed to provide a simple interface between a microprocessor and a raster-scanned color television. The TMS9928A/9929A VDPs are designed as a simple interface between a microprocessor, and R-G-B monitor or video encoder which produces the video for a video monitor. Figure 2-1 is a block diagram of the major portions of the VDP architecture interfaces to the VDP, CPU, VRAM, and color television.

## 2.1 CPU INTERFACE

The VDP interface to the CPU using an 8-bit bidirectional data bus, three control lines, and an interrupt is shown in Figure 2-2. Through this interface the CPU can conduct four operations:

(1) Write data bytes to VRAM

(2) Read data bytes from VRAM

(3) Write to one of the eight VDP write-only registers

(4) 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 three control lines of the VDP.

**NOTE**

The CPU can communicate with the VDP simultaneously and asynchronously with the VDP's TV screen refresh operations. The VDP performs memory management and allows periodic intervals 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 CSW, CSR, and MODE inputs. CSW is the CPU to VDP write select. When it is active (low), the eight bits on CD0-CD7 are strobed into the VDP. CSR is the CPU from VDP read select. When it is active (low), the VDP outputs eight bits on CD0-CD7 to the CPU. CSW and CSR should never be simultaneously low at the same time. If both are low, the VDP outputs data on CD0-CD7 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 TMS9900).

### 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 control the VDP operation and determine the way in which VRAM is allocated. The status register contains interrupt, sprite coincidence and fifth sprite status flags.

Each of the eight VDP write-only registers can be loaded using two 8-bit data transfers from the CPU. Table 2-1 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 MSB of the second byte must be a 1. The next four bits are 0s, and the lowest three bits make up the destination register number. The MODE input is high for both byte transfers.

To rewrite the data in an internal register after a byte of data has already been loaded, the status register must be read so that internal CPU interface logic is reinitialized and will accept the next byte as data and not as a register destination. This situation may be encountered in interrupt-driven program environments. Whenever the status of VDP write parameters is in question, this procedure should be used.

**NOTE**

The CPU address is destroyed by writing to the VDP register.

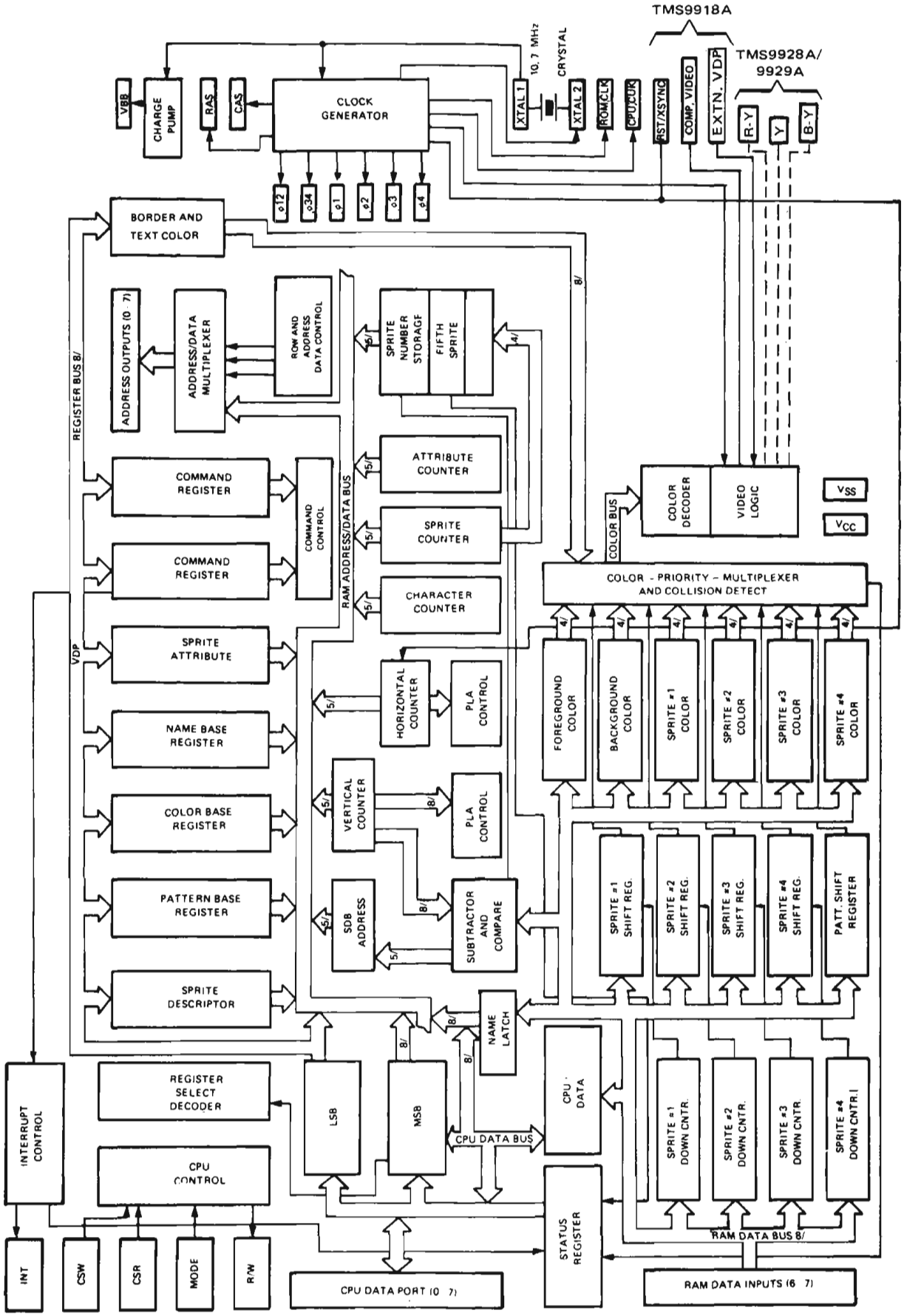



FIGURE 2-1 — VDP BLOCK DIAGRAM

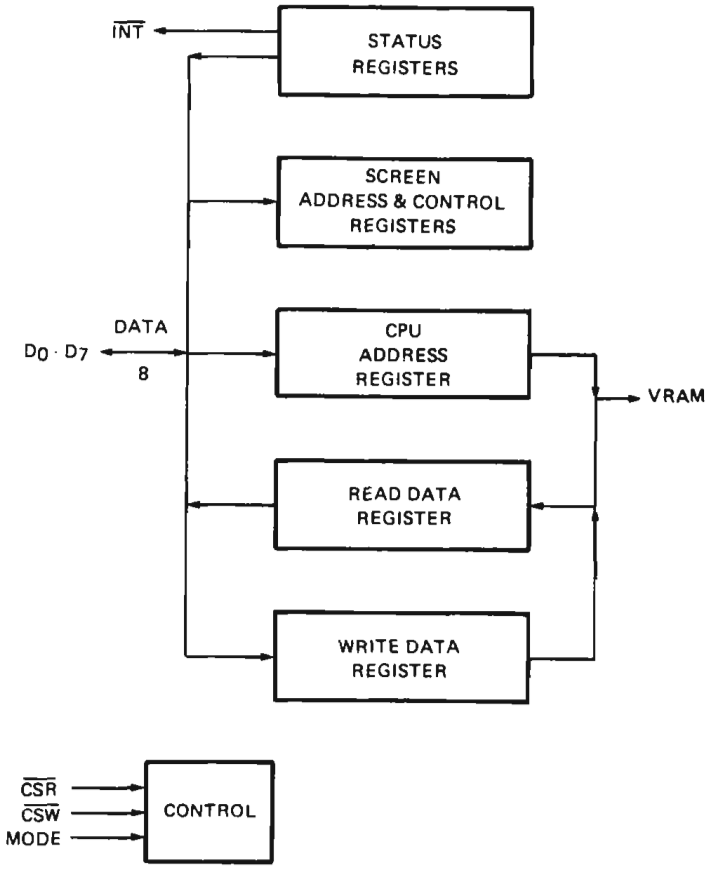



**FIGURE 2-2 — VDP TO CPU INTERFACE**

### 2.1.3 CPU Write to VRAM

The CPU transfers data to the VRAM through the VDP using a 14-bit autoincrementing address register. The address register setup requires 2-byte transfers. A 1-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 1-byte transfers since the address register is already set up. During setup of the address register, the two MSBs of the second address byte must be 0 and 1 respectively. MODE is high for both address transfers and low for the data transfer. CSW is used in all transfers to strobe the 8 bits into the VDP. See Table 2-1.

**TABLE 2-1 — 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 SETUP | $A_6$ | $A_7$ | $A_8$ | $A_9$ | $A_{10}$ | $A_{11}$ | $A_{12}$ | $A_{13}$ | 0 | 1 | 1 |
| BYTE 2 ADDRESS SETUP | 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 SETUP | $A_6$ | $A_7$ | $A_8$ | $A_9$ | $A_{10}$ | $A_{11}$ | $A_{12}$ | $A_{13}$ | 0 | 1 | 1 |
| BYTE 2 ADDRESS SETUP | 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. CSR is used to signal the VDP that a read operation is required.

### 2.1.5 CPU Read from VRAM

The CPU reads from the VRAM through the VDP using the autoincrementing address register. A 1-byte transfer is then required to read the data from the addressed VRAM byte. The address register is then autoincremented. Sequential VRAM data reads require only a 1-byte transfer since the address register is already set up. During setup of the address register, the two MSBs of the second address byte must be 0. By setting up the address this way, a read cycle to VRAM is initiated and read data will be available for the first data transfer to the CPU. (See Table 2-1). 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 the last data transfer and 2 microseconds following address setup.

The CPU interacts with VRAM memory through the VDP. The amount of time necessary for the CPU to transfer a byte of data to or from VRAM memory can vary from 2 to 8 microseconds. Once the VDP has been told to read or write a byte of data to or from VRAM it takes approximately 2 microseconds until the VDP is ready to make the data transfer. In addition to this 2 microsecond delay, the VDP must wait for a CPU access window; i.e., the period of time when the VDP is not occupied with memory refresh or screen display and is available to read or write data.

The worst case time between windows occurs during the Graphics I or Graphics II mode when sprites are being used. During the active display, CPU windows occur once every 16 memory cycles giving a maximum delay of 6 microseconds (a memory cycle takes about 372 nanoseconds). In the Text mode the CPU windows occur at least once out of every three memory cycles or a worst case delay of about 1.1 microseconds. Finally, in the Multicolor mode, CPU windows occur at least once out of every four memory cycles.

If the user needs to access memory in 2 microseconds, two situations occur where the time waiting for an access window is effectively zero. Both of these are independent of the display mode being used.

The first situation occurs when the blank bit of register 1 is 0. With this bit low, the entire screen will show only border color and the VDP does not have to wait for a CPU access window at any time.

The second situation occurs when the VDP is in the vertical refresh mode. The VDP issues an interrupt output at the end of each active area. This signal indicates that the VDP is entering the vertical refresh mode and that for the next 4.3 milliseconds there is no waiting for an access window. If the user wants the CPU to access memory during this interval, it is necessary for the controlling CPU to monitor the interrupt output of the VDP (the CPU can either poll this output or use it as an interrupt input).

The program that monitors the interrupt output must allow for its own delays in responding to the interrupt signal and recognize how much time it has left during the 4300 microsecond refresh period. The CPU must write a 1 to the interrupt enable bit of Register 1 in order to enable the interrupt for each frame, and then read the status register each time an interrupt is issued to clear the interrupt output. A summary of these delay times is presented in Table 2-2.

**TABLE 2-2 — MEMORY ACCESS DELAY TIMES**

| CONDITION | MODE | VDP DELAY | TIME WAITING FOR AN ACCESS WINDOW | TOTAL TIME |
|---|---|---|---|---|
| Active Display Area | Text | 2 μs | 0 - 1.1 μs | 2 - 3.1 μs |
| Active Display Area | Graphics I, II | 2 μs | 0 - 5.95 μs | 2 - 8 μs |
| 4300 μs after Vertical Interrupt Signal | All | 2 μs | 0 μs | 2 μs |
| Register I Blank Bit 0 | All | 2 μs | 0 μs | 2 μs |
| Active Display Area | Multicolor | 2 μs | 0 - 1.5 μs | 2 - 3.5 μs |

### 2.1.6 VDP Interrupt

The VDP INT output pin is used to generate an interrupt at the end of each active-display scan, which is about every 1/60 second for the TMS9918A/9928A and 1/50 second for the TMS9929A. The 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 RESET input is active (low) and must be held low for a minimum of 3 microseconds. The external reset synchronizes all clocks with 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 the VRAM.

## 2.2 WRITE-ONLY REGISTERS

The eight VDP write-only registers are shown in Figure 2-3. 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 that specify starting locations of various sub-blocks of VRAM. The definitions of these sub-blocks are described in Section 2.4. Register 7 is used to define backdrop and text colors.

Each register is described in the following paragraphs.

### 2.2.1 Register 0

Register 0 contains two VDP option control bits. All other bits are reserved for future use and must be 0s.

BIT 6 M3 (mode bit 3) (see Section 2.3.2 for table and description)

BIT 7 External VDP enable/disable

0 disables external VDP input
1 enables external VDP input

**NOTE**

Enabling bit 7 in the TMS9928A/9929A causes A-Y and B-Y to go to the sync level only when all planes in front of the pixel under question are transparent.

### 2.2.2 Register 1 (contains 8 VDP option control bits)

BIT 0 4/16K selection

0 selects 4027 RAM operation
1 selects 4108/4116 RAM operation

BIT 1 BLANK enable/disable

0 causes the active display area to blank
1 enables the active display
Blanking causes the display to show border color only

BIT 2 IE (Interrupt Enable)

0 disables VDP interrupt
1 enables VDP interrupt

BIT 3,4 M1, M2 (mode bits 1 and 2)
M1, M2 and M3 determine the operating mode of the VDP:

| M1 | M2 | M3 | |
|---|---|---|---|
| 0 | 0 | 0 | Graphics I mode |
| 0 | 0 | 1 | Graphics II mode |
| 0 | 1 | 0 | Multicolor Mode |
| 1 | 0 | 0 | Text mode |

BIT 5 Reserved

BIT 6 Size (sprite size select)

0 selects Size 0 sprites (8 × 8 bit)
1 selects Size 1 sprites (16 × 16 bits)

BIT 7 MAG (Magnification option for sprites)

0 selects MAG0 sprites (1X)
1 selects MAG1 sprites (2X)

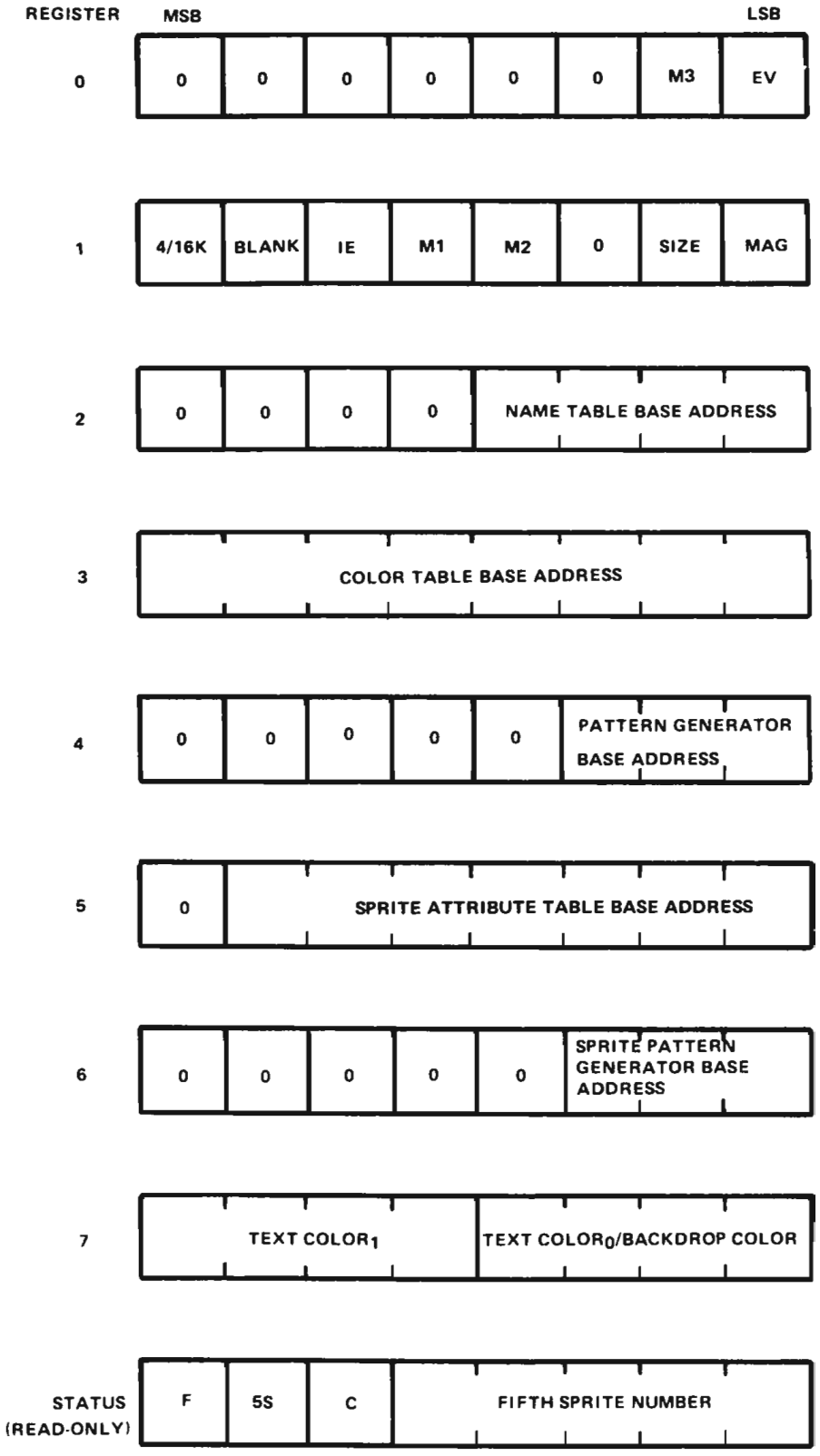



FIGURE 2-3 — VDP REGISTERS

### 2.2.3 Register 2

Register 2 defines the base address of the Name Table sub-block. The range of 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).

### 2.2.4 Register 3

Register 3 defines the base address of the Color Table sub-block. The range of 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).

### 2.2.5 Register 4

Register 4 defines the base address of the Pattern, Text or Multicolor Generator sub-block. The range of 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).

### 2.2.6 Register 5

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

### 2.2.7 Register 6

Register 6 defines the base address of the Sprite Pattern Generator sub-block. The range of 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).

### 2.2.8 Register 7

The upper 4 bits of Register 7 contain the color code of color 1 in the Text mode. The lower 4 bits contain the color code for color 0 in the Text mode and the backdrop color in all modes.

### 2.2.9 Setup Values for VDP Registers 2 through 6.

**VRAM TABLE ADDRESSING**

Register 2 in the VDP contains the starting address for the Name Table sub-block.

$$R2 * 400_{(16)} = \text{START ADDRESS}$$

| R2 | ADDRESS | |
|---|---|---|
| 00 | 0000 | |
| 01 | 0400 | |
| 02 | 0800 | |
| 03 | 0C00 | – MAXIMUM NUMBER FOR 4K RAMS |
| 04 | 1000 | |
| 05 | 1400 | |
| 06 | 1800 | |
| 07 | 1C00 | |
| 08 | 2000 | |
| 09 | 2400 | |
| 0A | 2800 | |
| 0B | 2C00 | |
| 0C | 3000 | |
| 0D | 3400 | |
| 0E | 3800 | |
| 0F | 3C00 | – MAXIMUM NUMBER |

Register 3 in the VDP contains the starting address for the Color Table.

(R3) * $40_{(16)}$ STARTING ADDRESS

| R3 | START ADDRESS | R3 | START ADDRESS | R3 | START ADDRESS |
|---|---|---|---|---|---|
| 00 | 0000 | 28 | 0A00 | 50 | 1400 |
| 01 | 0040 | 29 | 0A40 | 51 | 1440 |
| 02 | 0080 | 2A | 0A80 | 52 | 1480 |
| 03 | 00C0 | 2B | 0AC0 | 53 | 14C0 |
| 04 | 0100 | 2C | 0B00 | 54 | 1500 |
| 05 | 0140 | 2D | 0B40 | 55 | 1540 |
| 06 | 0180 | 2E | 0B80 | 56 | 1580 |
| 07 | 01C0 | 2F | 0BC0 | 57 | 15C0 |
| 08 | 0200 | 30 | 0C00 | 58 | 1600 |
| 09 | 0240 | 31 | 0C40 | 59 | 1640 |
| 0A | 0280 | 32 | 0C80 | 5A | 1680 |
| 0B | 02C0 | 33 | 0CC0 | 5B | 16C0 |
| 0C | 0300 | 34 | 0D00 | 5C | 1700 |
| 0D | 0340 | 35 | 0D40 | 5D | 1740 |
| 0E | 0380 | 36 | 0D80 | 5E | 1780 |
| 0F | 03C0 | 37 | 0DC0 | 5F | 17C0 |
| 10 | 0400 | 38 | 0E00 | 60 | 1800 |
| 11 | 0440 | 39 | 0E40 | 61 | 1840 |
| 12 | 0480 | 3A | 0E80 | 62 | 1880 |
| 13 | 04C0 | 3B | 0EC0 | 63 | 18C0 |
| 14 | 0500 | 3C | 0F00 | 64 | 1900 |
| 15 | 0540 | 3D | 0F40 | 65 | 1940 |
| 16 | 0580 | 3E | 0F80 | 66 | 1980 |
| 17 | 05C0 | 3F | 0FC0* | 67 | 19C0 |
| 18 | 0600 | 40 | 1000 | 68 | 1A00 |
| 19 | 0640 | 41 | 1040 | 69 | 1A40 |
| 1A | 0680 | 42 | 1080 | 6A | 1A80 |
| 1B | 06C0 | 43 | 10C0 | 6B | 1AC0 |
| 1C | 0700 | 44 | 1100 | 6C | 1B00 |
| 1D | 0740 | 45 | 1140 | 6D | 1B40 |
| 1E | 0780 | 46 | 1180 | 6E | 1B80 |
| 1F | 07C0 | 47 | 11C0 | 6F | 1BC0 |
| 20 | 0800 | 48 | 1200 | 70 | 1C00 |
| 21 | 0840 | 49 | 1240 | 71 | 1C40 |
| 22 | 0880 | 4A | 1280 | 72 | 1C80 |
| 23 | 08C0 | 4B | 12C0 | 73 | 1CC0 |
| 24 | 0900 | 4C | 1300 | 74 | 1D00 |
| 25 | 0940 | 4D | 1340 | 75 | 1D40 |
| 26 | 0980 | 4E | 1380 | 76 | 1D80 |
| 27 | 09C0 | 4F | 13C0 | 77 | 1DC0 |

*Maximum number for 4K RAMS

(R3)* $40_{(16)}$ STARTING ADDRESS (Concluded)

| R3 | START ADDRESS | R3 | START ADDRESS | R3 | START ADDRESS |
|---|---|---|---|---|---|
| 78 | 1E00 | A6 | 2980 | D3 | 34C0 |
| 79 | 1E40 | A7 | 29C0 | D4 | 3500 |
| 7A | 1E80 | A8 | 2A00 | D5 | 3540 |
| 7B | 1EC0 | A9 | 2A40 | D6 | 3580 |
| 7C | 1F00 | AA | 2A80 | D7 | 35C0 |
| 7D | 1F40 | AB | 2AC0 | D8 | 3600 |
| 7E | 1F80 | AC | 2B00 | D9 | 3640 |
| 7F | 1FC0 | AD | 2B40 | DA | 3680 |
| 80 | 2000 | AE | 2B80 | DB | 36C0 |
| 81 | 2040 | AF | 2BC0 | DC | 3700 |
| 82 | 2080 | B0 | 2C00 | DD | 3740 |
| 83 | 20C0 | B1 | 2C40 | DE | 3780 |
| 84 | 2100 | B2 | 2C80 | DF | 37C0 |
| 85 | 2140 | B3 | 2CC0 | E0 | 2800 |
| 86 | 2180 | B4 | 2D00 | E1 | 3840 |
| 87 | 21C0 | B5 | 2D40 | E2 | 3880 |
| 88 | 2200 | B6 | 2D80 | E3 | 38C0 |
| 89 | 2240 | B7 | 2DC0 | E4 | 3900 |
| 8A | 2280 | B8 | 2E00 | E5 | 3940 |
| 8B | 22C0 | B9 | 2E40 | E6 | 3980 |
| 8C | 2300 | BA | 2E80 | E7 | 39C0 |
| 8D | 2340 | BB | 2EC0 | E8 | 3A00 |
| 8E | 2380 | BC | 2F00 | E9 | 1A40 |
| 8F | 23C0 | BD | 2F40 | EA | 3A80 |
| 90 | 2400 | BE | 2F80 | EB | 3AC0 |
| 91 | 2440 | BF | 2FC0 | EC | 3B00 |
| 92 | 2480 | C0 | 3000 | ED | 3B40 |
| 93 | 24C0 | C1 | 3040 | EE | 3B80 |
| 94 | 2500 | C2 | 3080 | EF | 3BCD |
| 95 | 2540 | C3 | 30C0 | F0 | 3C00 |
| 96 | 2580 | C4 | 3100 | F1 | 3C40 |
| 97 | 25C0 | C5 | 3140 | F2 | 3C80 |
| 98 | 2600 | C6 | 3180 | F3 | 3CC0 |
| 99 | 2640 | C7 | 31C0 | F4 | 2D00 |
| 9A | 2680 | C8 | 3200 | F5 | 3D40 |
| 9B | 26C0 | C9 | 3240 | F6 | 3D80 |
| 9C | 2700 | CA | 3280 | F7 | 3DC0 |
| 9D | 2740 | CB | 32C0 | F8 | 3E00 |
| 9E | 2780 | CC | 3300 | F9 | 3E40 |
| 9F | 27C0 | CD | 3340 | FA | 3E80 |
| A0 | 2800 | CE | 3380 | FB | 3EC0 |
| A1 | 2840 | CF | 33C0 | FC | 3F00 |
| A2 | 2880 | D0 | 3400 | FD | 3F40 |
| A3 | 28C0 | D1 | 3440 | FE | 3F80 |
| A4 | 2900 | D2 | 3480 | FF | 3FC0 |
| A5 | 2940 | | | | |

Register 4 in the VDP contains the starting address for the Pattern Generator Sub-block.

$$(R4) * 800_{(16)} = \text{START ADDRESS}$$

| R4 | START ADDRESS | |
|---|---|---|
| 00 | 0000 | |
| 01 | 0800 | – Max # for 4K RAMS |
| 02 | 1000 | |
| 03 | 1800 | |
| 04 | 2000 | |
| 05 | 2800 | |
| 06 | 3000 | |
| 07 | 3800 | – Max # for 16K RAMS |

Register 5 in the VDP contains the starting address for the Sprite Attribute Table.

$$(R5) * 80_{(16)} = \text{START ADDRESS}$$

| R5 | START ADDRESS | R5 | START ADDRESS | R5 | START ADDRESS | R5 | START ADDRESS |
|---|---|---|---|---|---|---|---|
| 00 | 0000 | 21 | 1080 | 40 | 2000 | 60 | 3000 |
| 01 | 0080 | 22 | 1100 | 41 | 2080 | 61 | 3080 |
| 02 | 0400 | 23 | 1180 | 42 | 2100 | 62 | 3100 |
| 03 | 0180 | 24 | 1200 | 43 | 2180 | 63 | 3180 |
| 04 | 0200 | 25 | 1280 | 44 | 2200 | 64 | 3200 |
| 05 | 0280 | 26 | 1300 | 45 | 2280 | 65 | 3280 |
| 06 | 0300 | 27 | 1380 | 46 | 2300 | 66 | 3300 |
| 07 | 0380 | 28 | 1400 | 47 | 2380 | 67 | 3380 |
| 08 | 0400 | 29 | 1480 | 48 | 2400 | 68 | 3400 |
| 09 | 0480 | 2A | 1500 | 49 | 2480 | 69 | 3480 |
| 0A | 0500 | 2B | 1580 | 4A | 2500 | 6A | 3500 |
| 0B | 0580 | 2C | 1600 | 4B | 2580 | 6B | 3580 |
| 0C | 0600 | 2D | 1680 | 4C | 2600 | 6C | 3600 |
| 0D | 0680 | 2E | 1700 | 4D | 2680 | 6D | 3680 |
| 0E | 0700 | 2F | 1780 | 4E | 2700 | 6E | 3700 |
| 0F | 0780 | 30 | 1800 | 4F | 2780 | 6F | 3780 |
| 10 | 0800 | 31 | 1880 | 50 | 2800 | 70 | 3800 |
| 11 | 0880 | 32 | 1900 | 51 | 2880 | 71 | 3880 |
| 12 | 0900 | 33 | 1980 | 52 | 2900 | 72 | 3900 |
| 13 | 0980 | 34 | 1A00 | 53 | 2980 | 73 | 3980 |
| 14 | 0A00 | 35 | 1A80 | 54 | 2A00 | 74 | 3A00 |
| 15 | 0A80 | 36 | 1B00 | 55 | 2A80 | 75 | 3A80 |
| 16 | 0B00 | 37 | 1B80 | 56 | 2B00 | 76 | 3B00 |
| 17 | 0B80 | 38 | 1C00 | 57 | 2B80 | 77 | 3B80 |
| 18 | 0C00 | 39 | 1C80 | 58 | 2C00 | 78 | 3C00 |
| 19 | 0C80 | 3A | 1D00 | 59 | 2C80 | 79 | 3C80 |
| 1A | 0D00 | 3B | 1D80 | 5A | 2D00 | 7A | 3D00 |
| 1B | 0D80 | 3C | 1E00 | 5B | 2D80 | 7B | 3D80 |
| 1C | 0E00 | 3D | 1E80 | 5C | 2E00 | 7C | 3E00 |
| 1D | 0E80 | 3E | 1F00 | 5D | 2E80 | 7D | 3E80 |
| 1E | 0F00 | 3F | 1F80 | 5E | 2F00 | 7E | 3F00 |
| 1F | 0F80 * | | | 5F | 2F80 | 7F | 3F80 |
| 20 | 1000 | | | | | | |

*Maximum number for 4K RAMS

Register 6 contains the value for the starting address of the Sprite Pattern Generator sub-block.

STARTING ADDRESS = R6 *<800

| R6 | START ADDRESS | |
|---|---|---|
| 00 | 0000 | |
| 01 | 0800 | – Max # for 4K DRAMS |
| 02 | 1000 | |
| 03 | 1800 | |
| 04 | 2000 | |
| 05 | 2800 | |
| 06 | 3000 | |
| 07 | 3800 | – Max # for 16K RAMS |

## 2.3 STATUS REGISTER

The VDP has a single 8-bit status register that 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 2-3 and is discussed in the following paragraphs.

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. However, asynchronous reads will 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.

### 2.3.1 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 (1), the VDP interrupt output (INT) will be active (low) whenever the F status flag is a 1.

Note that the status register needs to be read frame by frame in order to clear the interrupt and receive the new interrupt of the next frame.

### 2.3.2 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 overlapping pixel. Transparent colored sprites, as well as those that are partially or completely off the screen, are also considered. Sprites beyond the Sprite Attribute Table terminator (D016) are not considered. The C flag is cleared to a 0 after the status register is read or the VDP is externally reset. The status register should be read immediately upon powerup to ensure that the coincidence flag is reset.

The VDP checks each pixel position for coincidence during the generation of the pixel regardless of where it is located on the screen. This occurs every 1/60th of a second for the TMS9918A and TMS9928A and every 1/50th of a second for the TMS9929A. Thus, when moving sprites more than one pixel position during these intervals, it is possible for the sprites to have multiple pixels overlapping or even to have passed completely over one another when the VDP checks for coincidence.

### 2.3.3 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.

## 2.4 VIDEO DISPLAY MODES

The VDP displays an image on the screen that can best be envisioned as a set of display planes sandwiched together. Figure 2-4 shows the definition of each of the planes. Objects on planes closest to the viewer have higher priority. In cases where two entities on two 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 (Figure 2-5) each may contain a single sprite. 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 three sizes: 8 × 8 pixels, 16 × 16 pixels, and 32 × 32 pixels.

Behind the Sprite Planes is the Pattern Plane. The Pattern Plane is used for textual and graphics images generated by the Text, Graphics I, Graphics II, or Multicolor modes. 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 and lowest priority plane is the External VDP Plane. Its image is defined by the external VDP input pin which allows the TMS9918A to mix the external video signal internal to the chip.

This mixing must occur outside of the chip for the TMS9928A and TMS9929A. This is achieved through the color difference outputs swinging to a special level (sync level is shown in Figure 2-6) not used by the color difference signals in normal operation. This occurs when bit 7 of Register 0 is set high. External mixing circuitry is required to detect this change in the level of the color difference signals and then switch from the VDP signals to an external source's signals (see Figures 2-7 and 2-8).

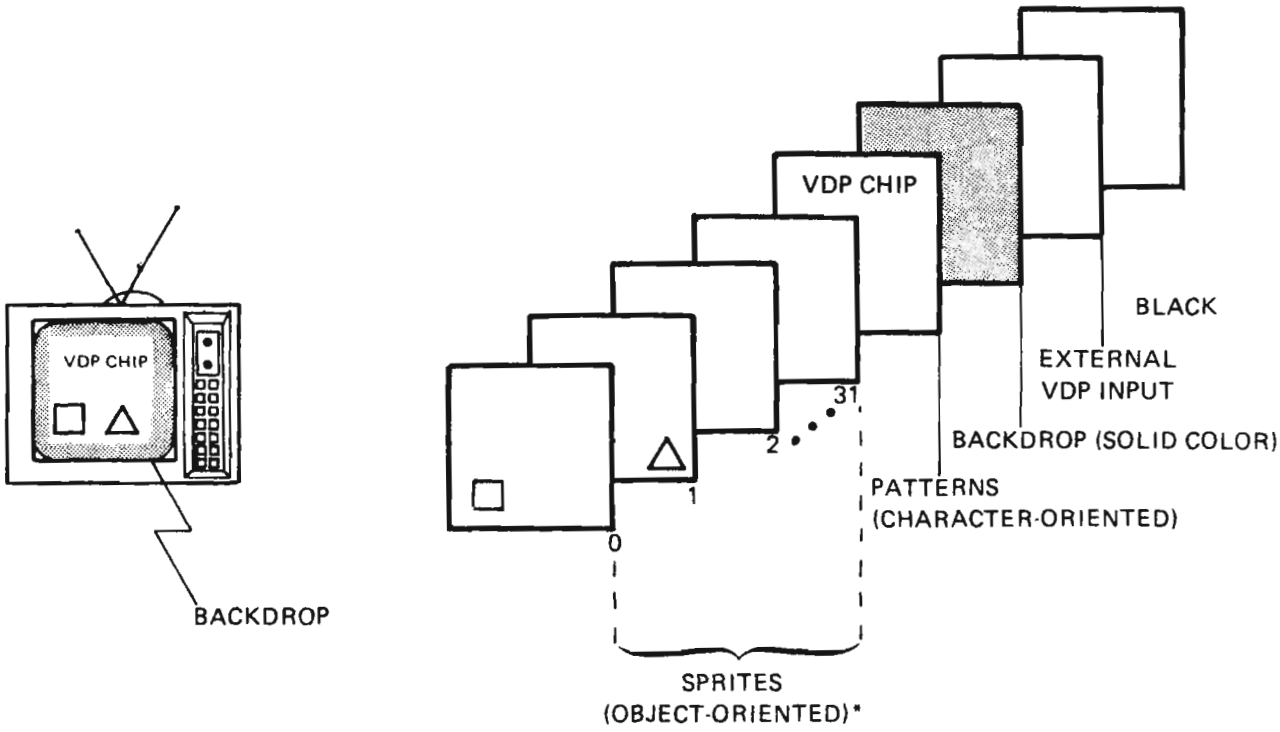



**FIGURE 2-4 — VDP DISPLAY PLANES (DEFINITION)**

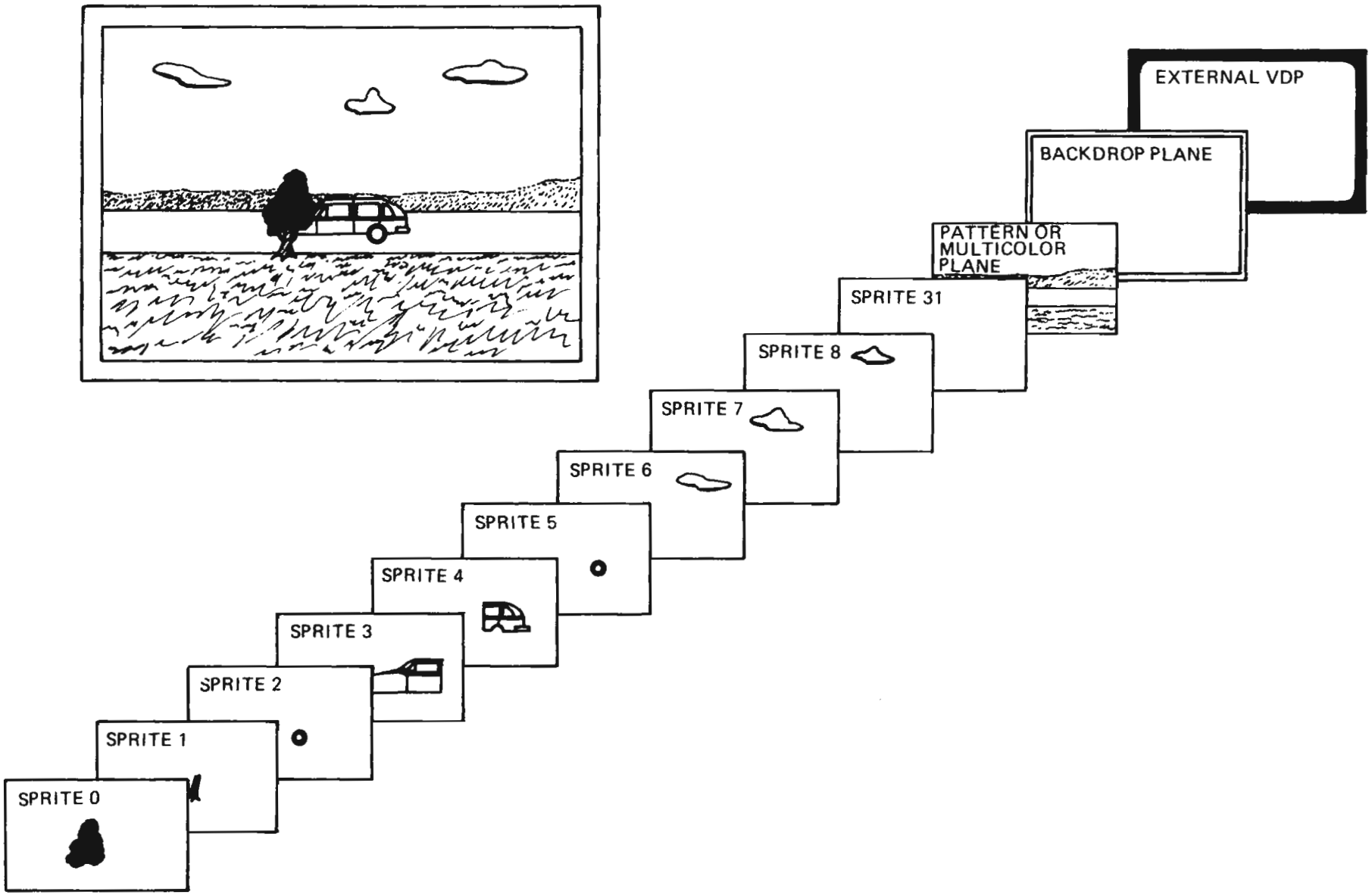



FIGURE 2-5 — VDP DISPLAY PLANES (FIRST 32 PLANES)

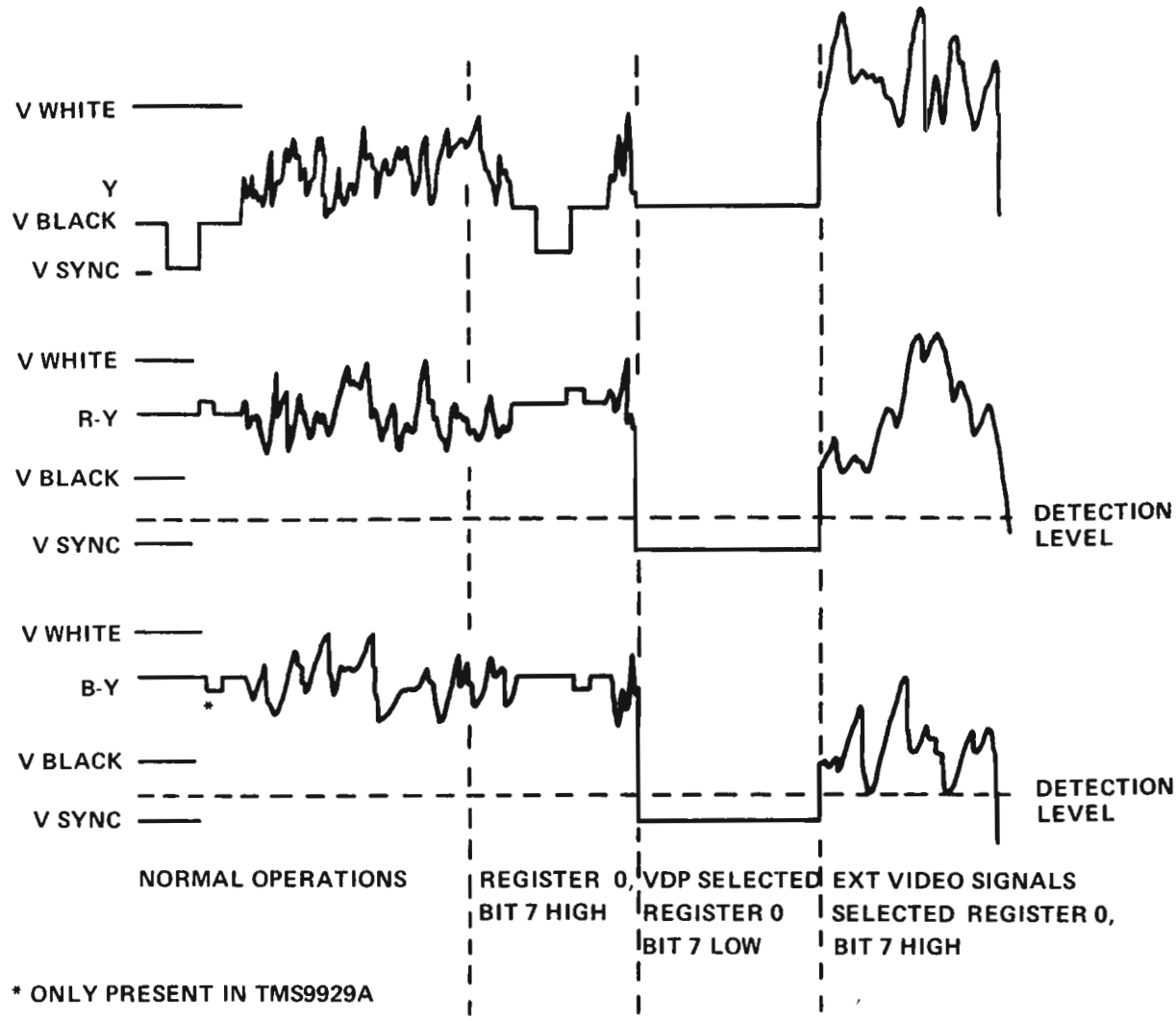



FIGURE 2-6 — TMS9928A/9929A SIGNAL WAVEFORMS FOR MULTIPLE VDP OPERATION

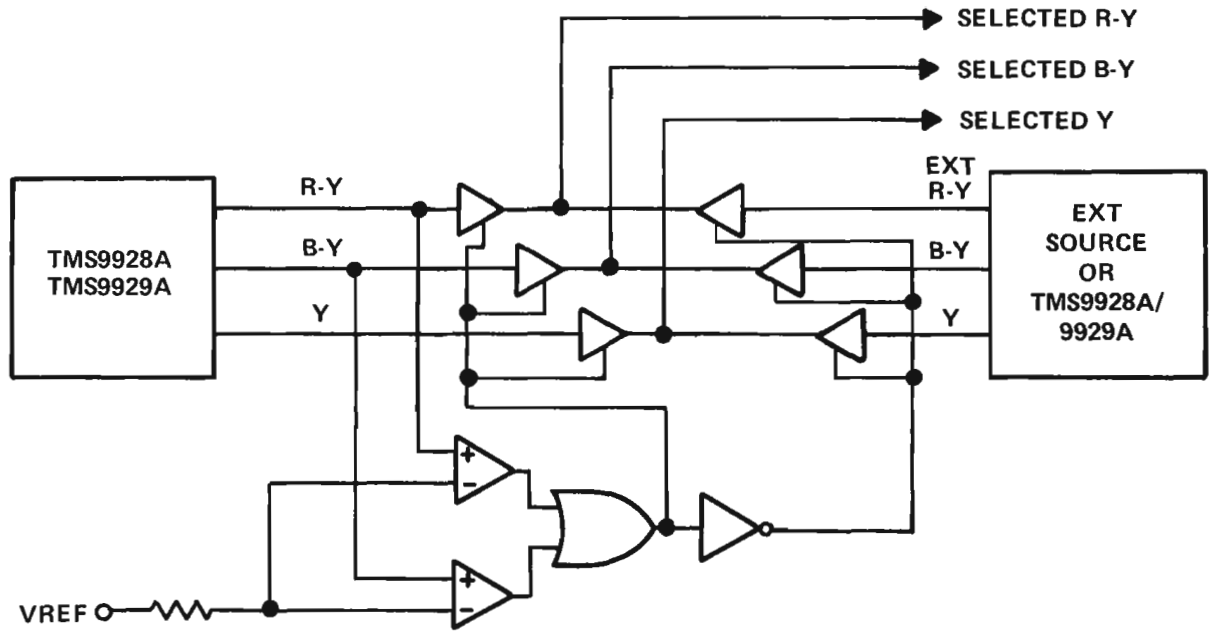



FIGURE 2-7 — USING COLOR DIFFERENCE SIGNALS TO MIX EXTERNAL COLOR DIFFERENCE TYPE SOURCE

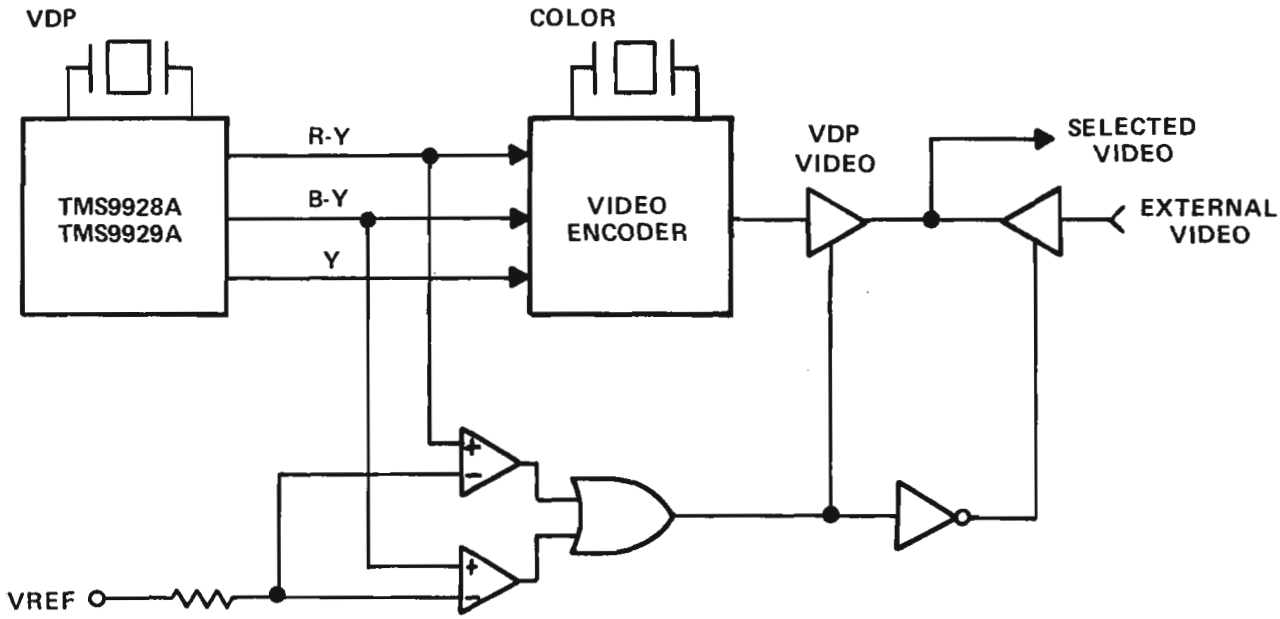



FIGURE 2-8 — USING COLOR DIFFERENCE SIGNALS TO MIX EXTERNAL VIDEO SOURCES

The backdrop consists of a single color 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 if the external VDP mode is not selected.

The 32 Sprite Planes are used for the 32 sprites in the Multicolor and 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 nontransparent, the colors of the lower planes are automatically replaced by the sprite color.

There is also 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 that are active on the display will cause the coincidence flag to set. The VDP status register provides a flag bit and the number of the fifth sprite whenever this occurs. The Pattern Plane is used in the Text, Multicolor, and Graphics modes for display of the graphic patterns of characters. Whenever a pixel on the Pattern Plane is nontransparent, 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 four video color display modes that appear on the Pattern Plane: Graphics I mode, Graphics II mode, Text mode, and Multicolor mode. Graphics I and Graphics II modes cause 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 the Graphics modes.

In Graphics I mode, 256 possible patterns may be defined for the 768 pattern positions with two unique colors allowed for each line of a pattern definition. Thus, all 15 colors plus transparent may be used in a single pattern position.

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 by VDP Register 7.

In Multicolor mode, the screen is broken into a grid of 64 × 48 positions, each of which is a 4 × 4 pixel. 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 Sprite Pattern Generator Table and the Sprite Attribute Table are used to form 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 15 colors shown in Table 2-3. The VDP colors also provide eight different gray levels for displays on monochrome television; the luminance value in the table indicates these levels, 0.00 being black and 1.00 being white. All other values in the table are expressed as percentages of the white/black voltage swing.

**NOTE**

The gray levels differ slightly for the TMS9918A when compared to the TMS9928A/9929A.

Whenever all planes are of the transparent color at a given point, and external video is not selected, the color shown at that point will be black.

TABLE 2-3 — COLOR ASSIGNMENTS

| COLOR HEX | COLOR | TMS9918A | | TMS9928A/9929A | | |
|---|---|---|---|---|---|---|
| | | LUMINANCE (DC) VALUE | CHROMINANCE (AC VALUE) | Y | COLOR DIFFERENCE R-Y | B-Y |
| 0 | TRANSPARENT | 0.00 | - | - | - | - |
| 1 | BLACK | 0.00 | - | 0.00 | .47 | .47 |
| 2 | MEDIUM GREEN | .53 | .53 | .53 | .07 | .20 |
| 3 | LIGHT GREEN | .67 | .40 | .67 | .17 | .27 |
| 4 | DARK BLUE | .40 | .60 | .40 | .4 | 1.00 |
| 5 | LIGHT BLUE | .53 | .53 | .53 | .43 | .93 |
| 6 | DARK RED | .47 | .47 | .47 | .83 | .30 |
| 7 | CYAN | .67 | .60 | .73 | 0.00 | .70 |
| 8 | MEDIUM RED | .53 | .60 | .53 | .93 | .27 |
| 9 | LIGHT RED | .67 | .60 | .67 | .93 | .27 |
| A | DARK YELLOW | .73 | .47 | .73 | .57 | .07 |
| B | LIGHT YELLOW | .80 | .33 | .80 | .57 | .17 |
| C | DARK GREEN | .46 | .47 | .47 | .13 | .23 |
| D | MAGENTA | .53 | .40 | .53 | .73 | .67 |
| E | GRAY | .80 | - | .80 | .47 | .47 |
| F | WHITE | 1.00 | - | 1.00 | .47 | .47 |
| - | BLACK LEVEL | 0.00 | - | 0.00 | .47 | .47 |
| - | COLOR BURST | 0.00 | .40 | 0.00 | 47(28A)<br>73(29A) | .1(28A)<br>.2(29A) |
| - | SYNC LEVEL | -0.40 | - | -.46 | .47 | .47 |
| - | EXTERNAL VIDEO | - | - | 0.00 | .47 | .47 |
| - | LEVEL | - | - | 0.00 | -.46 | -.46 |

### 2.4.1 Graphics I Mode

The VDP is in Graphics I mode when M1, M2, and M3 bits in VDP registers 1 and 0 are zero. When in this mode the Pattern Plane is divided into a grid of 32 columns by 24 rows of pattern positions as shown in Figure 2-9). Each of the pattern positions contains 8 × 8 pixel. The tables in VRAM used to generate the Pattern Plane are the Pattern Generator, Name, and Color Tables which require 2848 VRAM bytes. Figure 2-9 illustrates the mapping of these tables into the Pattern Plane. 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.

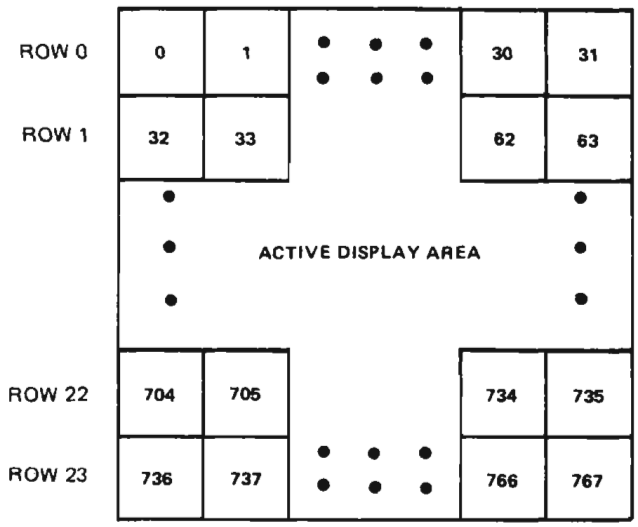



FIGURE 2-9 — PATTERN GRAPHICS NAME TABLE MAPPING

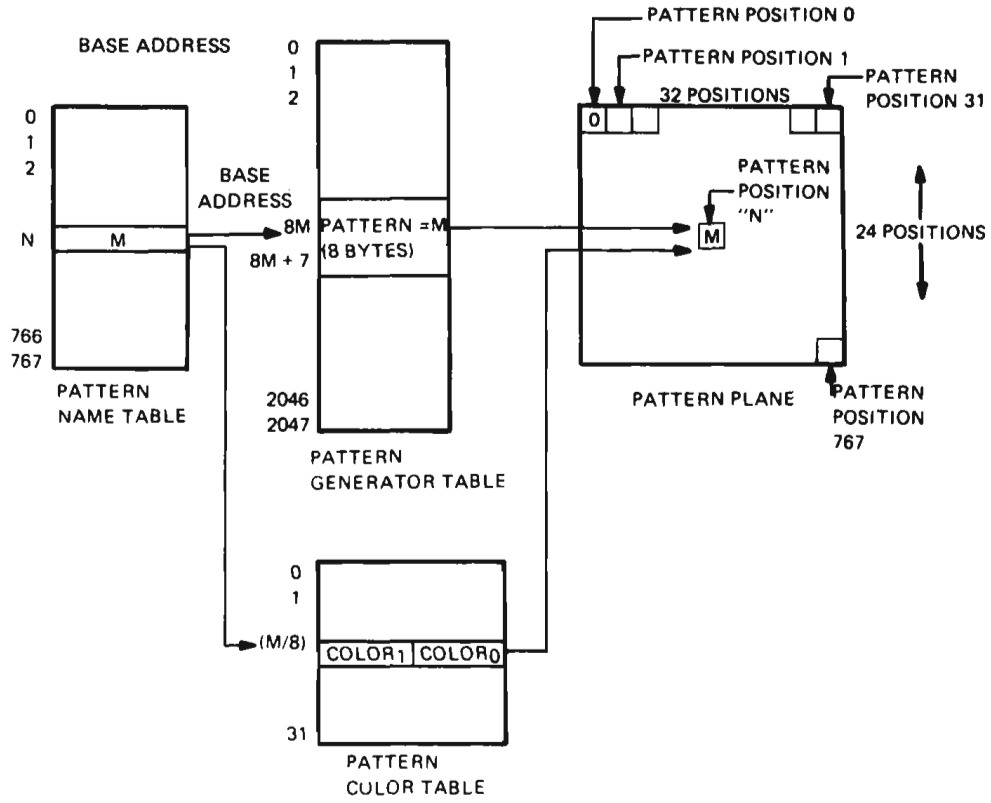



**FIGURE 2-10 — GRAPHICS I MODE MAPPING**

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. All of the 1s in the 8-byte pattern can designate one color (color 1), while all the 0s can designate another color (color 0).

The full 8-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 2-kilobyte boundary. The starting address of the table is determined by the generator base address in VDP Register 4. The base address forms the three MSBs of the 14-bit VRAM address for each Pattern Generator Table entry. The next 8 bits indicate the 8-bit name of the selected pattern definition. The lowest 3 bits of the VRAM address indicate the row number within the pattern definition.

There are 8 bytes 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 bit of each of the eight bytes defines the first column of the pattern. The remaining rows and columns are similarly 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 4 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 2-11.

| ROW/BYTE | COLUMN (PATTERN) | | | | | | BIT (PATTERN DEFINITION) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 0 | | C | C | C | C | C | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | | | | | | C | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 2 | | | | | | C | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 3 | | | C | C | C | C | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 4 | | | | | | C | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 5 | | | | | | C | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 6 | | C | C | C | C | C | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 7 | | | | | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

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

**FIGURE 2-11 — PATTERN DISPLAY MAPPING**

The color of the 1s and 0s is defined by the Pattern Color Table that contains 32 entries, each of which is 1 byte long. Each entry defines two colors: the most significant 4 bits of each entry define the color of the 1s, and the least significant 4 bits define the color of the 0s. 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 2-4 for assignments.) Thus, 32 different pairs of colors may be displayed simultaneously.

The Pattern Name Table is located in a contiguous 768-byte block in VRAM beginning on a 1-kilobyte 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 4 bits of the 14-bit VRAM address. The lower 10 bits of the VRAM address are formed from the row and column counters. An example of pattern name table addressing is given in Section 3.3.

**TABLE 2-4 — GRAPHICS I MODE COLOR TABLE**

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

Each byte entry in the Name Table is either 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 eight patterns in a group; the color codes are stored in the VDP Color Table. The Color Table is located in a 32-byte clock 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 8 bits of the 14-bit Color Table entry VRAM address. The next bit is a 0 and the lowest 5 bits are equal to the upper 5 bits of the corresponding Name Table entries.

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.

When the Pattern Generator Table is loaded with a pattern set, manipulation of the Pattern Name Table contents can change the appearance of the screen. Alternatively, a dynamically changing set of patterns throughout the course of a graphics session is easily accomplished since all tables are in VRAM. A total of 2848 VRAM bytes are required for the Pattern, Name, Color and Generator tables. Less memory is needed 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.

### 2.4.2 Graphics II Mode

The VDP is in the Graphics II mode bits (M1 = 0, M2 = 0 and M3 = 1). The Graphics II mode is similar to Graphics I mode except it allows a larger library of patterns so that a unique pattern generator entry may be made for each of the 768 (32 × 24) pattern positions on the video screen. Additionally, more color information is included in each 8 × 8 graphics pattern. Thus, two unique colors may be specified for each byte of the 8 × 8 pattern. A larger amount of VRAM (12 kilobytes) is required to implement the full usage of the Graphics II mode.

Like Graphics I mode, the Graphics II mode Pattern Name Table contains 768 entries which correspond to the 768 pattern positions on the display screen. Because the Graphics I mode pattern names are only 8 bits in length, a maximum of 256 pattern definitions may be addressed using the addressing scheme discussed in Section 2.4.10. Graphics II mode, however, segments the display screen into three equal parts of 256 pattern positions each and also segments the Pattern

Generator Table into three equal blocks of 2048 bytes each. Pattern definitions in the first third of the display screen correspond to pattern positions in the upper third. Likewise, pattern definitions in the second and third blocks of the Pattern Generator Table correspond to the second and third areas of the Pattern Plane.

The Pattern Name Table is also segmented into three blocks of 256 names each so that names found in the upper third reference pattern definitions are found in the upper 2048 bytes in Pattern Generator Table. Similarly, the second and third blocks reference pattern definitions in the second 2048-byte block and third-2048 byte block, respectively. Thus, if 768 patterns are uniquely specified, an 8-bit pattern name will be used three times, once in each segment of the Pattern Name Table. The Pattern Generator Table falls on 8-kilobyte boundaries and may be located in the upper or lower half of 16K memory based on the MSB of the pattern generator base in VDP Register 4. The LSBs must be set to all 1s.

The Color Table is also 6144 bytes long and is segmented into three equal blocks of 2048 bytes. Each entry in the Pattern Color Table is 8 bytes which provides the capability to uniquely specify color 1 and color 0 for each of the 8 bytes of the corresponding pattern definition. The addressing scheme is exactly like that of the Pattern Generator Table except for the location of the table in VRAM. This is controlled by the loading of the MSB of the color base in VDP Register 3. The LSBs must be set to all 1s.

Figure 2-12 illustrates the Graphics II mode mapping scheme. Note that pattern names, P1, P2, and P3, correspond to pattern generator entries in the three blocks of the Pattern Generator Table. Note also how these three names map to the display screen. Figure 2-13 is an example of a Pattern Generator and Pattern Color Table entry.

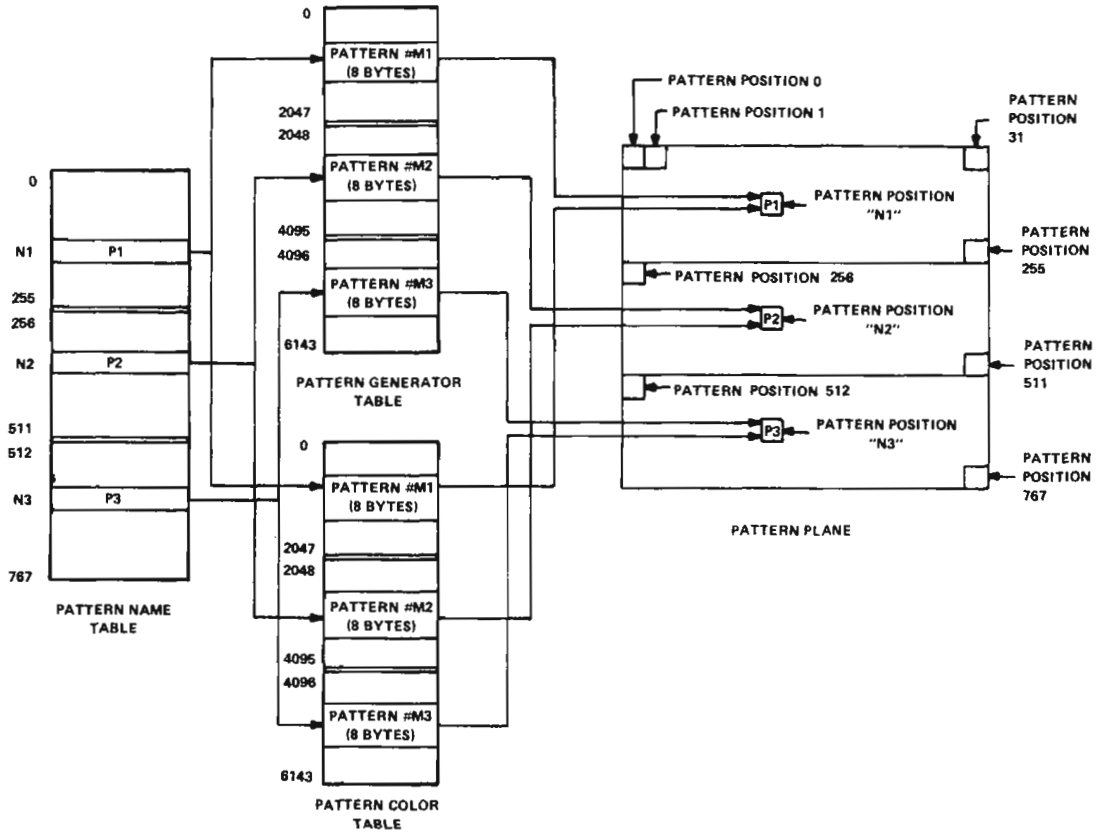



**FIGURE 2-12 — GRAPHICS II MODE MAPPING**

| | PATTERN GENERATOR TABLE ENTRY | | | | | | | | PATTERN | | | | | | | | PATTERN COLOR TABLE ENTRY | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | 0 3 | 4 7 | |
| ROW 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | B | 1 | B | B | B | B | B | 1 | 1 (BLACK) | B (LT. YELLOW) | 0 ROW |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | B | B | 7 | B | B | B | 7 | B | 7 (CYAN) | B (LT. YELLOW) | 1 |
| 2 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | B | B | B | C | B | C | B | B | C (GREEN) | B (LT. YELLOW) | 2 |
| 3 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | B | B | B | B | E | B | B | B | E (GRAY) | B (LT. YELLOW) | 3 |
| 4 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | B | B | B | B | 8 | B | B | B | 8 (MED. RED) | B (LT. YELLOW) | 4 |
| 5 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | B | B | B | B | 5 | B | B | B | 5 (LT. BLUE) | B (LT. YELLOW) | 5 |
| 6 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | B | B | B | B | 6 | B | B | B | 6 (DK. RED) | B (LT. YELLOW) | 6 |
| 7 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | B | B | B | B | D | B | B | B | D (MAGENTA) | B (LT. YELLOW) | 7 |

FIGURE 2-13 — PATTERN DISPLAY MAPPING

### 2.4.3 Multicolor Mode

The VDP is in Multicolor mode when mode bits M1 = 0, M2 = 1, and M3 = 0. Multicolor mode provides an unrestricted 64 × 48 color square display. Each color square contains a 4 × 4 block of pixels. The color of each of the color squares can be any one of the 15 video display colors plus transparent. Consequently, all 15 colors can be used simultaneously in the Multicolor mode. The Backdrop and Sprite Planes are still active in the Multicolor mode.

The Multicolor Name Table is the same as that for the graphics modes, consisting of 768 name entries, although the name no longer points to a color list. Color is now derived from the Pattern Generator Table. The name points to an 8-byte segment of VRAM in the Pattern Generator Table.

Only 2 bytes of the 8-byte segment are used to specify the screen image. These 2 bytes specify four colors, each color occupying a 4 × 4-pixel area. The 4 MSBs of the first byte define the color of the upper left quarter of the multicolor pattern; the LSBs define the color of the upper right quarter. The second byte similarly defines the lower left and right quarters of the multicolor pattern. The 2 bytes thus map into an 8 × 8-pixel multicolor pattern. (See Figure 2-14).

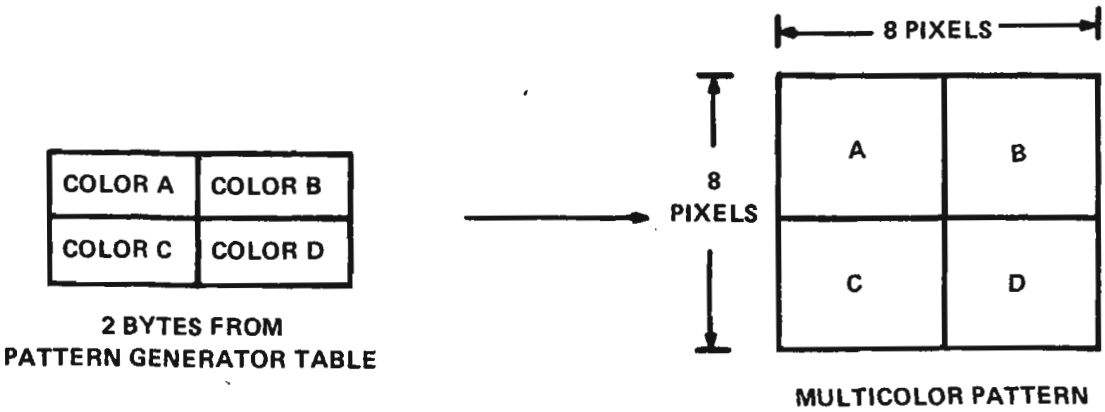



FIGURE 2-14 — MULTICOLOR LIST MAPPING

The location of the 2 bytes within the 8-byte segment pointed to by the name is dependent upon the screen position where the name is mapped. For names in the top row (names 0-31), the 2 bytes are the first two within the groups of 8-byte segments pointed to by the names. The next row of names (32-63) uses the bytes 3 and 4 within the 8-byte segments. The next row of names uses bytes 5 and 6 while the last row of names uses bytes 7 and 8. This series repeats for the remainder of the screen.

For example, referring to Figure 2-15 if Name Table entry 0 (pattern position 0) multicolor block #N (name = N), the multicolor pattern displayed will be an 8 × 8-pixel block consisting of colors A, B, C, and D which comprise the first two bytes of the Multicolor Table. If, however, name #N is located in Name Table entry 33, (Pattern position 33), the colors displayed will be colors E, F, G, and H as specified by bytes 3 and 4 of the multicolor block pointed to by the name.

Likewise, pattern positions which lie in rows 2 and 3 would cause colors I, J, K, L and colors M, N, O, P, respectively, to be displayed. Thus, it can be seen that the color displayed from the multicolor generator block is dependent upon pattern position on the screen. Figure 2-16 illustrates the Multicolor mode mapping scheme.

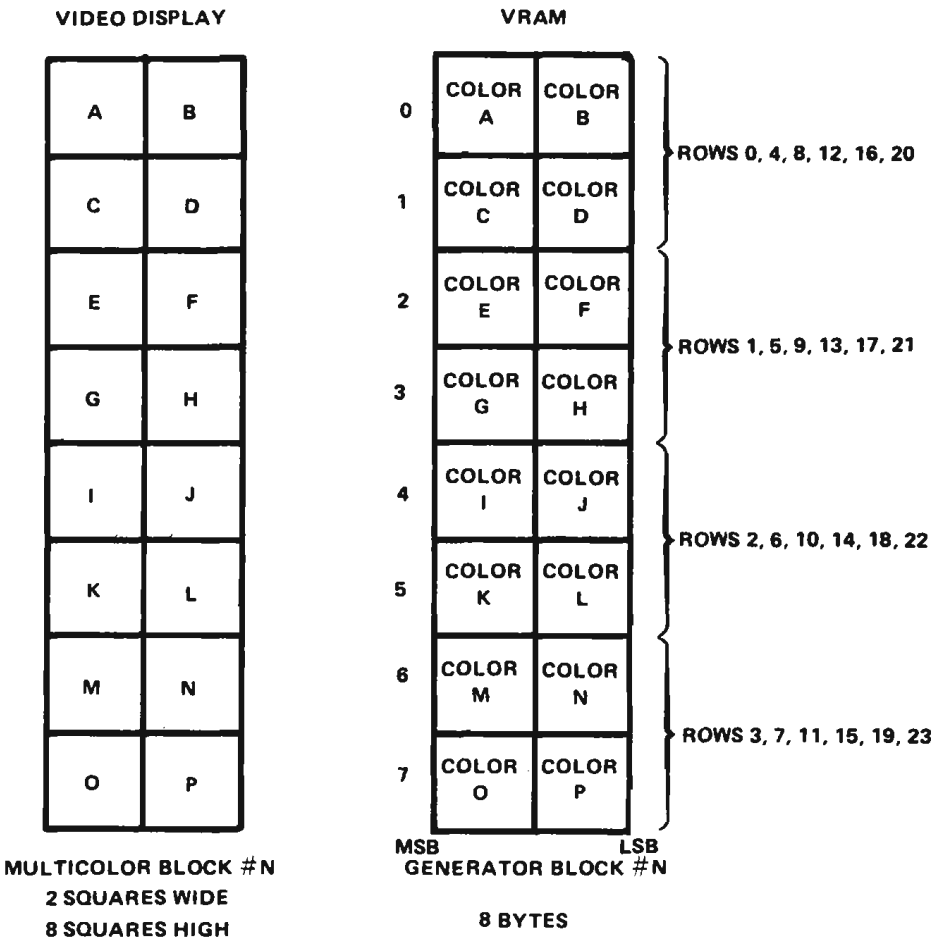



FIGURE 2-15 – MULTICOLOR BLOCK DISPLAY

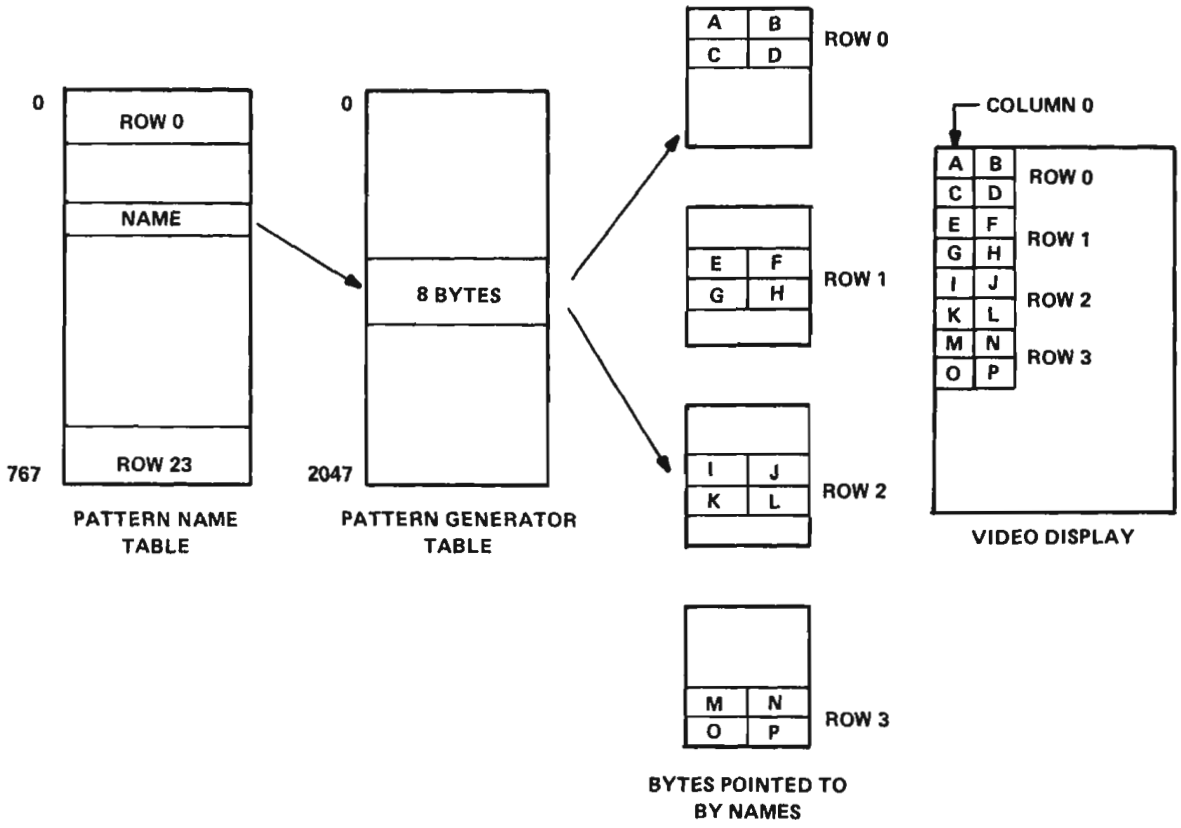



FIGURE 2-16 — MULTICOLOR MODE MAPPING

The mapping of VRAM contents to screen image is simplified by using duplicate names in the Name Table since the series of bytes used within the 8-byte segment specifies a 2 × 8 color square pattern on the screen as a straightforward translation from the 8-byte segment in VRAM pointed to by the common name.

When used in this manner, 768 bytes are still used for the Name Table and 1536 bytes are used for the color information in the Pattern Generator Table (24 rows × 32 columns × 8 bytes/pattern position). Thus, a total of 1728 bytes in VRAM are required. It should be noted that the tables begin on even 1K and 2K boundaries and are therefore not contiguous. An example of multicolor VRAM memory allocation is given in Section 3.3.

### 2.4.4 Text Mode

The VDP is in Text mode when mode bits M1 = 1, M2 = 0, and M3 = 0. In this mode, the screen is divided into a grid of 40 text positions across and 24 down. (See Figure 2-17). 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 definition into each of the 960 pattern cells on the Pattern Plane (Figure 2-18). Sprites are not available in Text mode.

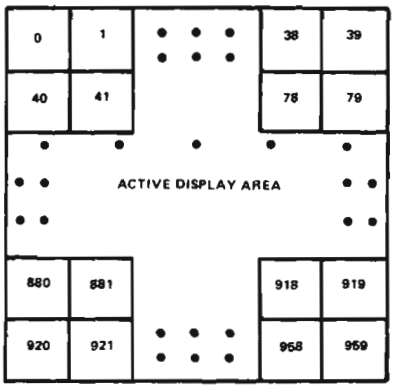



FIGURE 2-17 — TEXT MODE NAME TABLE PATTERN POSITIONS

As with the Graphics modes, 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 8-byte block defines a text pattern in which all the 1s in the text pattern take on one color when displayed on the screen, while all the 0s take on another color. These colors are chosen by loading VDP Register 7 with the color 1 and color 0 in the left and right nibbles. respectively (see Section 2.2).

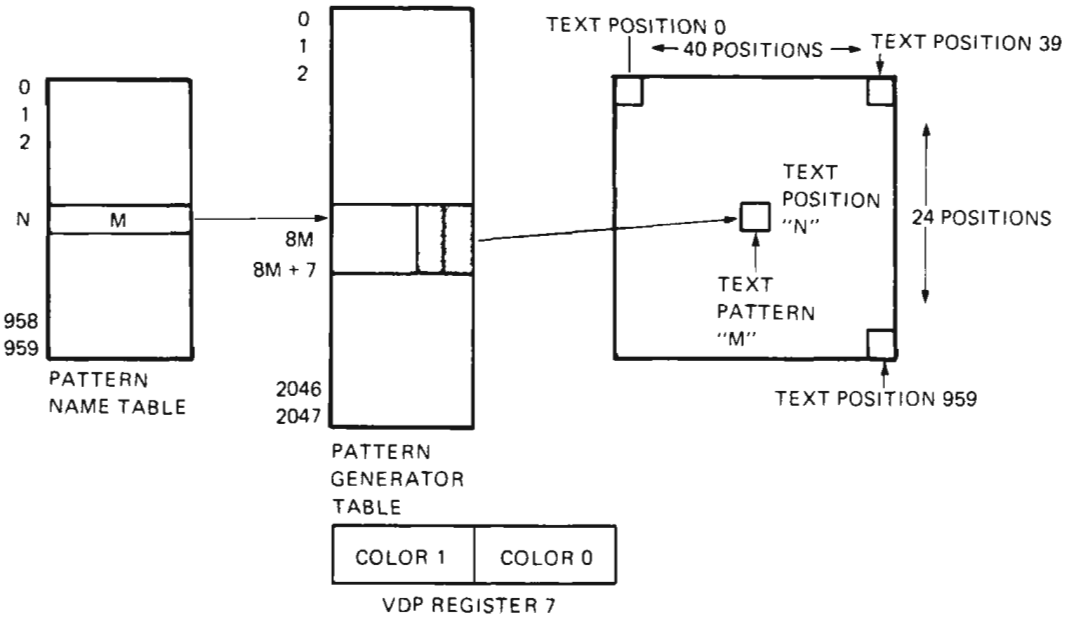



**FIGURE 2-18 — MAPPING OF VRAM INTO THE PATTERN PLANE IN TEXT MODE**

In the Text mode, the Pattern Name Table determines the position of the text pattern on the screen as shown in Figure 2-18. There are 960 entries in the Pattern Name Table, each 1 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 correspond to the top row of text pattern positions on the screen, the next 40 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 1-kilobyte 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 4 bits of the 14-bit VRAM address. The lower 10 bits of the VRAM address point to 1 of 960 pattern cells. The name table is organized by rows. An example of Pattern Name Table addressing is given in Section 4.

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 the name implies, the Text mode is intended mainly for textual applications, especially those in which the 32 patterns-per-line in Graphics modes is insufficient. The advantage is that eight 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 Graphics I mode. This is done by ensuring that the least significant 2 bits of all the character patterns are 0. Thus, a switch from Text mode to Pattern mode 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 Graphics Modes, once a character set has been defined and placed into the Pattern Generator, updating the Pattern Name Table will produce and manipulate textual material on the screen.

The full 8-bit pattern name is used to select 1 of the 256 pattern definitions in the pattern generator table. The table is a 2048-byte block in VRAM, beginning on a 2-kilobyte boundary. The starting address of the table is determined by the generator base address in VDP Register 4. The base address forms the 3 MSBs of the 14-bit VRAM address for each Pattern Generator Table entry. The next 8 bits are equal to the 8-bit name of the selected pattern definition. The lowest 3 bits of the VRAM address are equal to the row number within the pattern definition.

There are 8 bytes 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 least significant 2 bits in each byte are not used. However, it is strongly recommended that these bits be 0s. 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 4 bits of VDP Register 7. A 0 bit selects the other color code (color 0) which is in the least significant 4 bits of the same VDP Register. Figure 2-18 is an example of pattern definition mapping.

A total of 3008 VRAM bytes are required for the Pattern Name 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.

### 2.4.5 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.

The sub-blocks in VRAM that define sprites are the Sprite Attribute Table (see Figure 2-19) and the Sprite Generator Table (see Section 4.4). 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 2-5.

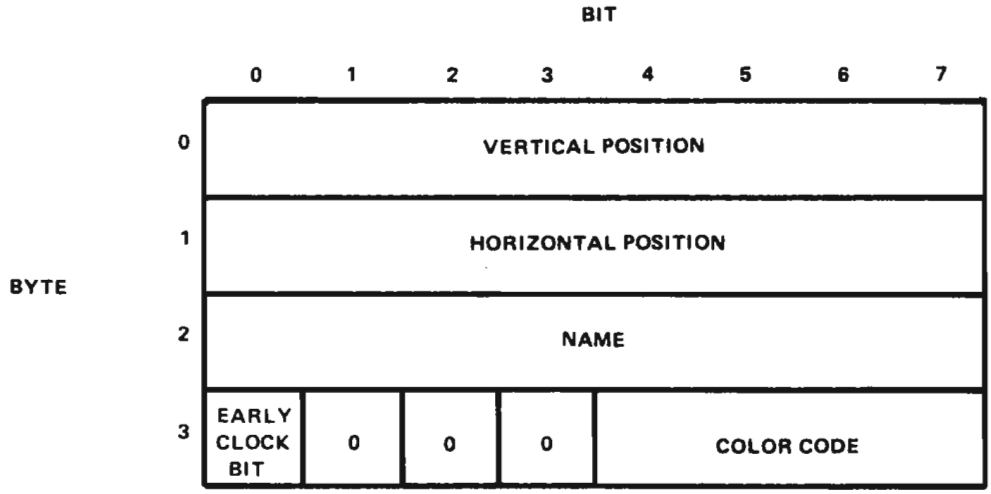



FIGURE 2-19 — SPRITE ATTRIBUTE TABLE ENTRY

TABLE 2-5 — 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 2-20 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 four bytes. The entries are ordered so 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 and is located in a contiguous 128-byte block in VRAM, beginning on a 128-byte boundary.

The starting address of the table is determined by the 7-bit Sprite Attribute Table base address in VDP Register 5. The base address forms the upper 7 bits of the 14-bit VRAM address. The next 5 bits of the VRAM address are equal to the sprite number. The lowest 2 bits select 1 of the 4 bytes in Sprite 2 Attribute Table entry for each sprite. Each table entry contains 4 bytes which specify the sprite position, sprite pattern name, and color, as shown in Figure 2-19.

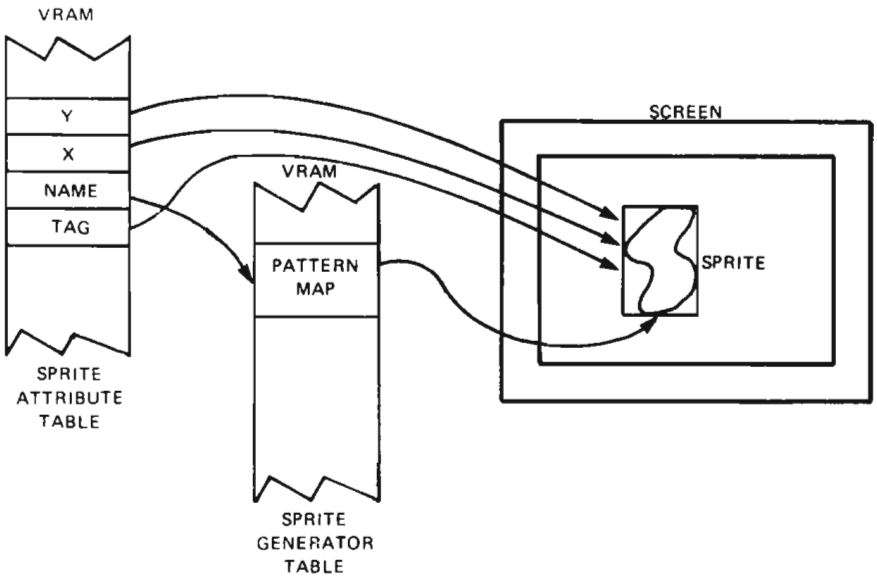



**FIGURE 2-20 — SPRITE MAPPING**

The first two 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 butts the sprite up against the left edge of the backdrop. Note that all measurements are taken from the upper left pixel of the sprite.

When the first two bytes of an entry position a sprite so it overlaps backdrop, the part of the sprite that is within the backdrop is displayed normally. The part of the sprite that overlaps the backdrop is hidden from view by the backdrop. This allows the animator to move a sprite into display from behind the backdrop.

The displacement in the first byte is partially signed, in that values for vertical displacement between −31 and 0 (E116 to 0) allow a sprite to bleed-in from the top 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.

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 8-bit pointer to the sprite patterns definition, the Sprite Generator Table. The sprite name is similar to that in the Graphics Modes.

Byte 4 of the Sprite Attribute Table entry contains the color of the sprite in its lower 4 bits (see Table 2-3 for color assignments). The MSB is the Early Clock (EC) bit. When set to 0, this bit does nothing. When set to 1, 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 the 2-kilobyte 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 a sprite's shape. The 1s in the Sprite Generator cause the sprite to be defined at the point; 0s 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 3 MSB of the 14-bit VRAM address. The next 8 bits of the address are equal to sprite name, and the last 3 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 1, and the number of the violating fifth sprite is loaded into the status register (see Section 2.3).

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 described in the following paragraphs.

| | |
|---|---|
| MAG = 0, SIZE = 0: | No options chosen |
| MAG = 1, SIZE = 0 | The Sprite Generator Table uses 8 bytes 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: | The Sprite Generator Table uses 31 bytes 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 2-21. Mapping is still 1 bit to 1 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 (D016) 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 in the preceding paragraph. 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.

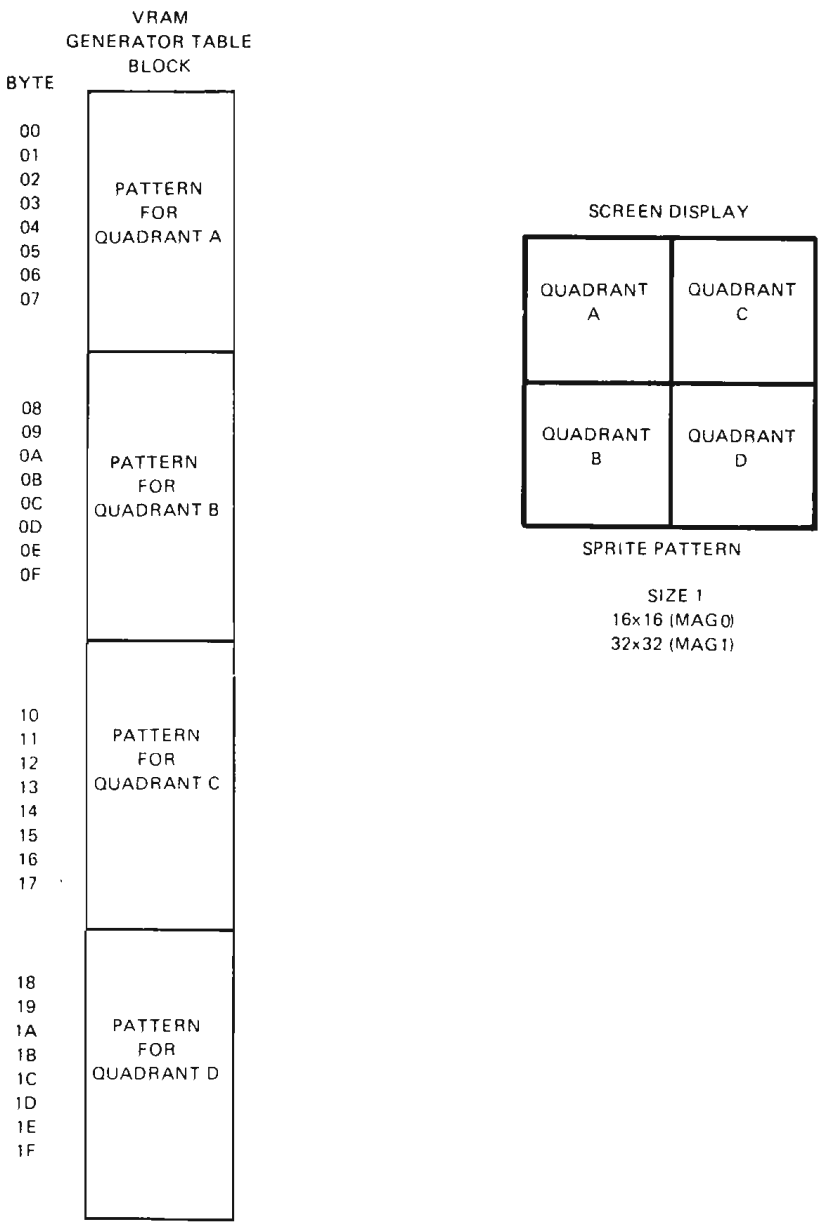



**FIGURE 2-21 — SIZE 1 SPRITE MAPPING**

### 2.4.6 A Step-by-Step Approach to Create Patterns and Sprites

PATTERNS

1. Use an 8 × 8 pattern similar to that in Figure A. Each small square represents one pixel on the screen.

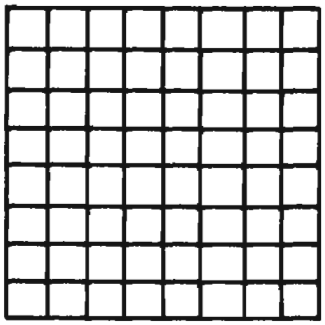


**FIGURE A**

2. Fill in the blocks to create your text character or graphics pattern. Examples of the letter A and an ARROW are shown in Figures B and C.

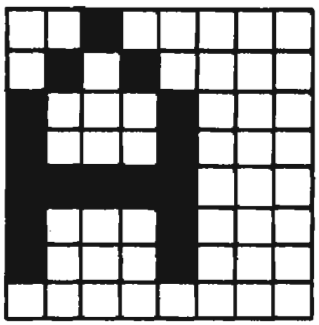


**FIGURE B**

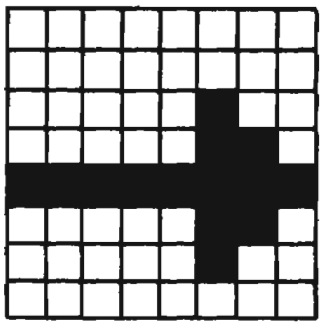
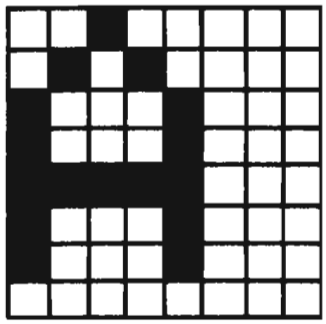


**FIGURE C**

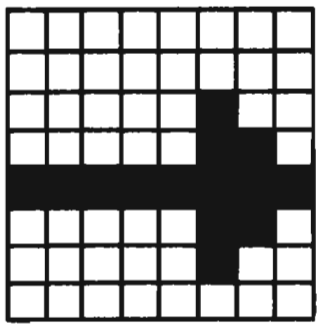
**NOTE**

If these patterns are to be used in the Text mode, (40 patterns per line), the pattern should be inside a left-justified 6 × 8 block like the A shown in Figure B. If all of the Text patterns are inside this 6 × 8 block, they can be used for Text and Graphics 1 and 2 modes.

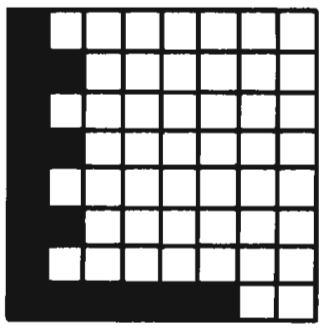
3. Assign 1s to the filled-in areas and 0s to the blanks. Then convert the 1s and 0s to their hexadecimal equivalents, as shown in Figure D.



= 00100000 = $20_{(16)}$
= 01010000 = $50_{(16)}$
= 10001000 = $88_{(16)}$
= 10001000 = $88_{(16)}$
= 11111000 = $F8_{(16)}$
= 10001000 = $88_{(16)}$
= 10001000 = $88_{(16)}$
= 00000000 = $00_{(16)}$



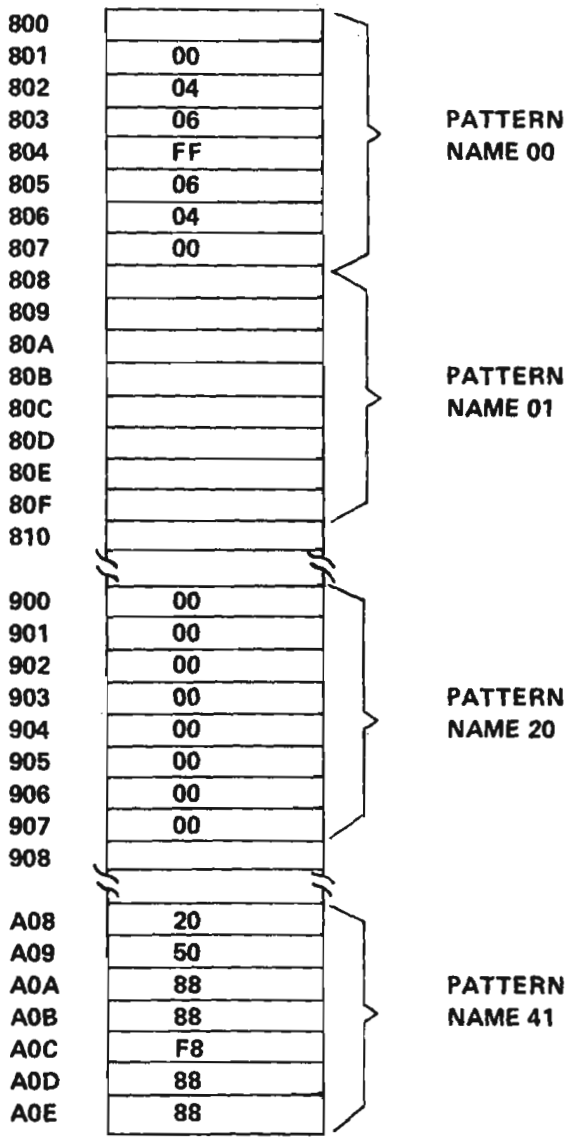
= $00_{(16)}$
= $00_{(16)}$
= $04_{(16)}$
= $06_{(16)}$
= $FF_{(16)}$
= $06_{(16)}$
= $04_{(16)}$
= $00_{(16)}$



= $80_{(16)}$
= $C0_{(16)}$
= $80_{(16)}$
= $C0_{(16)}$
= $80_{(16)}$
= $C0_{(16)}$
= $80_{(16)}$
= $FC_{(16)}$

FIGURE D

4. Now place the eight bytes defining the pattern into the Pattern Generator Table. Assume the Pattern Generator Table sub-block is located at $800_{16}$ and the arrow pattern is to be named $00_{16}$. Then place the eight pattern bytes as follows:




**NOTE**

When using text in your applications, you can place the eight bytes of the text character in its ASCII number location.

Example: ASCII SPACE = $20_{16}$
? = $3F_{16}$
A = $41_{16}$
B = $42_{16}$
C = $43_{16}$
Etc.

This simplifies writing text to the screen. Simply write the ASCII name directly to the Pattern Name Table.

A space character is shown in Pattern Generator Table position 20, and A is shown in pattern name 41.

1. Determine whether to use 8 × 8 or 16 × 16 sprite patterns. Then use the appropriate work pattern, as shown in Figures E and F.

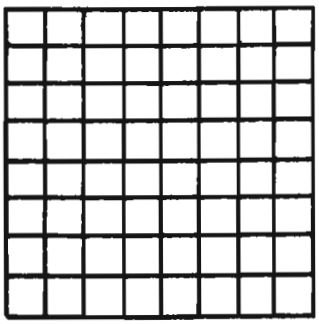


FIGURE E

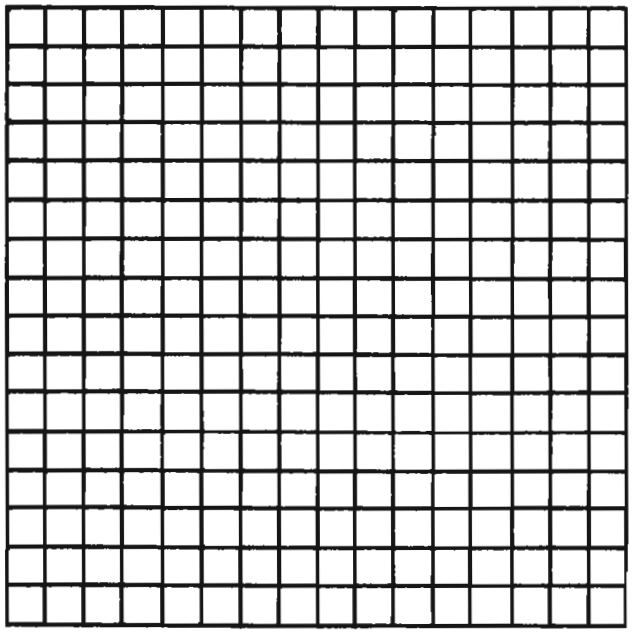


FIGURE F

2. Fill in the blocks to create your sprite pattern. Examples are shown in Figures G and H.

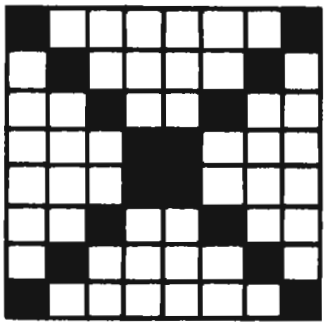


FIGURE G

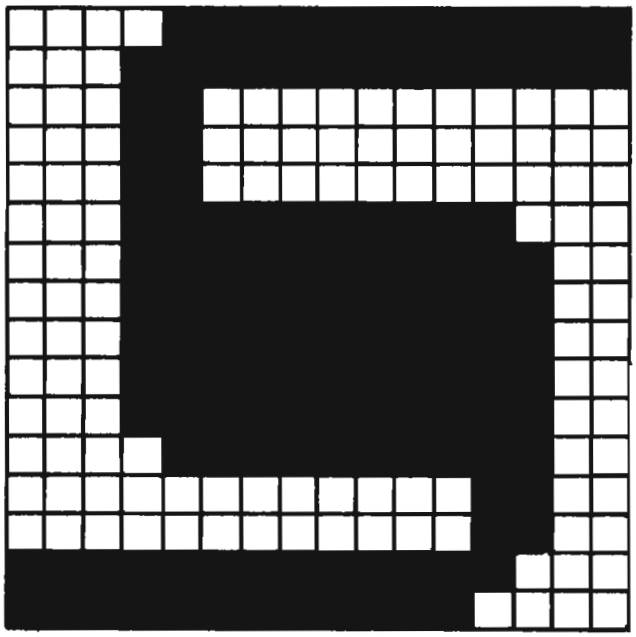


FIGURE H

3. Next encode the sprite patterns as in the Pattern Section. The 8 × 8 sprite encodes exactly as the 8 × 8 pattern, but the 16 x 16 sprite encodes as shown in Figure J.

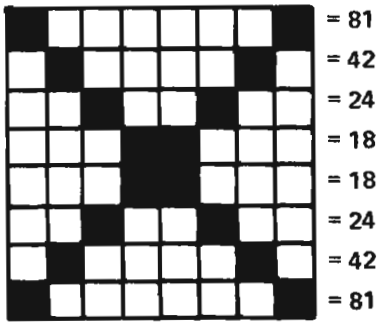



FIGURE I

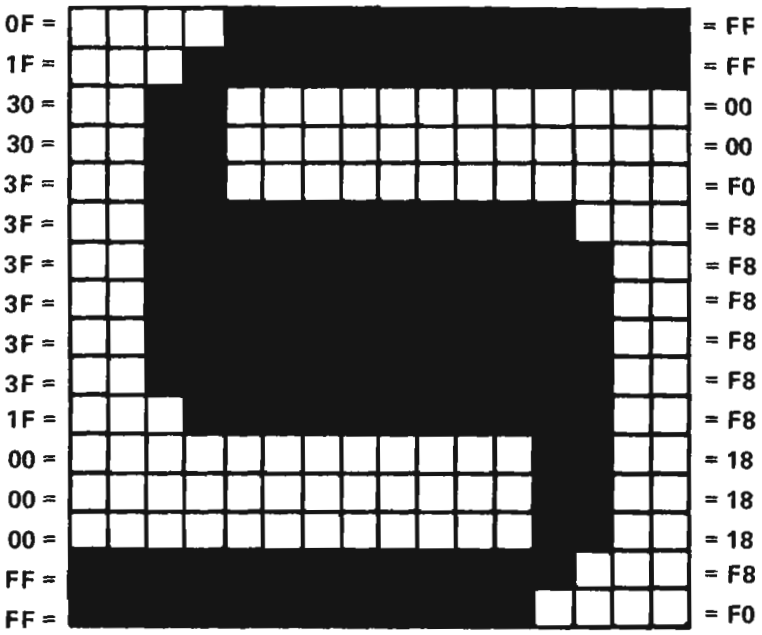



FIGURE J

Break the 16 × 16 block pattern into four 8 × 8 patterns. Next, encode the 8 × 8 patterns starting in the upper left corner, then do the lower left, upper right, and lower right.

4. Place the 8 bytes for 8 × 8 sprites or 32 bytes for 16 × 16 sprites in the Sprite Generator Table. Assuming the sprite generator table is located at location 0000, Figures K and L show how the tables should look for 8 × 8 and 16 × 16 sprites.

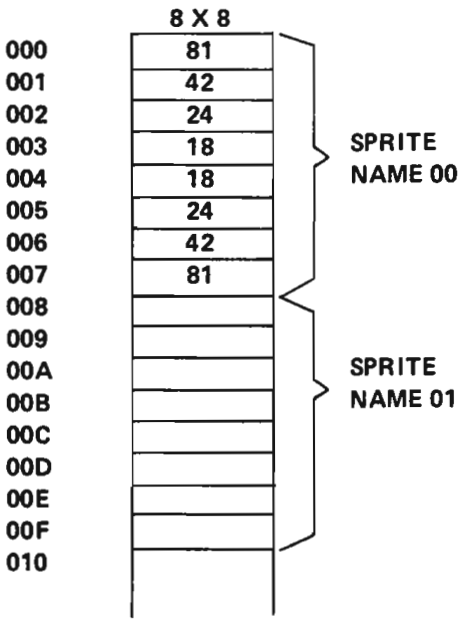



FIGURE K

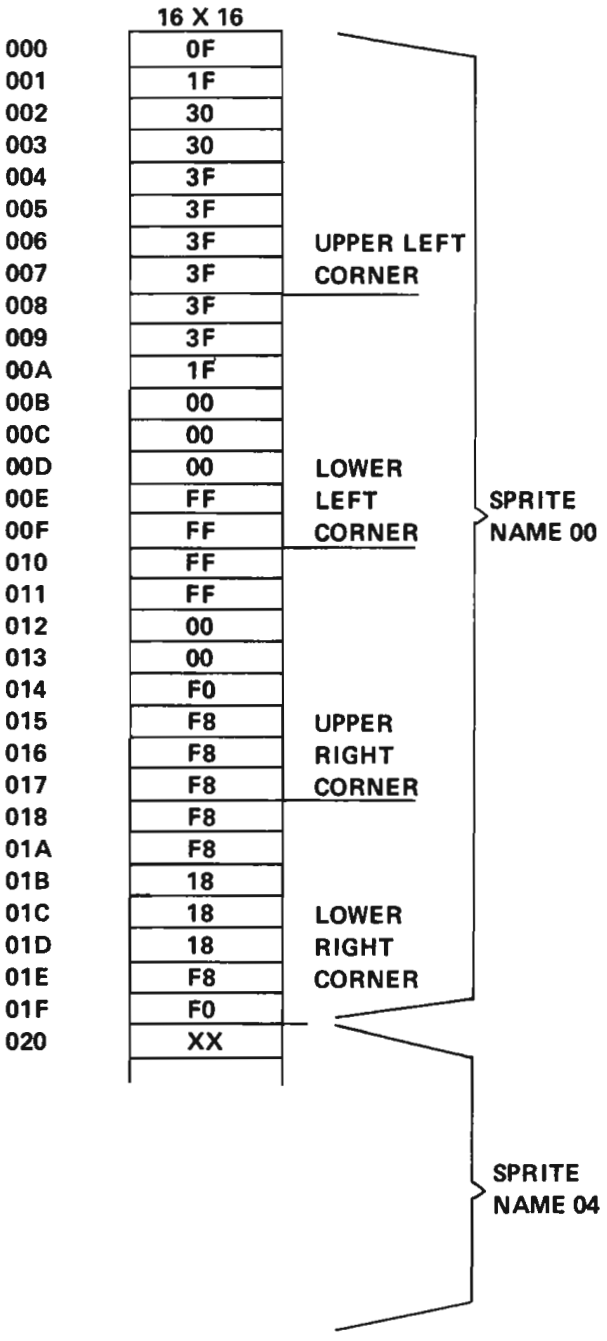



FIGURE L

16 × 16 sprite patterns start in the table with the byte from the upper left-hand corner. Then start with the upper right, going toward the lower right.

# 3. VDP INTERFACES AND OPERATION

## 3.1 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 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.

### 3.1.1 VRAM Interface Control Signals

The VDP-VRAM interface consists of two 8-bit data buses (RD0-RD7 unidirectional, AD0-AD7 bidirectional) and three control lines, as shown in Figure 3-1. 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). The column address is output when $\overline{CAS}$ is active (low). Data is output to the VRAM when R/$\overline{W}$ is active (low).

### 3.1.2 VRAM Memory Types

The VDP can use 4027-type 4K, 4108-type 8K, 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 4108- and 4116-type RAMs. There is a minor difference between the way 4027s and 4108s/4116s are wired to the VDP. In the 4027, all $\overline{CE}$ pins are tied to ground. In the 4108/4116 the A6 lines on the 4116 and 4108 (the same pin as $\overline{CE}$ on 4027's) are all tied to AD1 on the TMS9918A. A jumper can be used to select the VRAM type.

### 3.1.3 VDP to DRAM Address Connections

The VDP can be easily connected to either the 4027 or 4116 DRAMs. However, due to different pin numbering standards, it is possible to connect the VDP to the DRAMs incorrectly. Table 3-1 shows the recommended way to connect a VDP to either DRAM. Other DRAMs, such as the single +5 V supply type, can also be used by following the 4K or 16K columns in Table 3-1.

**TABLE 3-1 — VDP TO DRAM ADDRESS CONNECTIONS**

| VDP | 4116 or 16K | 4027 or 4K |
|---|---|---|
| AD0 | DATA ONLY | DATA ONLY |
| AD1 | A6 | DATA ONLY |
| AD2 | A5 | A5 |
| AD3 | A4 | A4 |
| AD4 | A3 | A3 |
| AD5 | A2 | A2 |
| AD6 | A1 | A1 |
| AD7 | A0 | A0 |

When connecting the data ports together, ensure that corresponding RAMs (assuming 8 by 1 DRAMs) are properly connected to the corresponding input or output of the VDP. For example, AD0 of the corresponding input or output D input of the RAM, and RD0 of the VDP should connect to the Q output of the same RAM. The same is true for all AD and RD corresponding pins for each of the eight DRAMs.

**NOTE**

CD0 is the MSB of the CD bus; CD7 is the LSB.
AD0 is the MSB of the AD bus; AD7 is the LSB.
RD0 is the MSB of the RD bus; RD7 is the LSB.
RAMs have the reverse convention.
AD7 is the MSB of the AD bus, and AD0 is the LSB.

Therefore, AD7 of the VDP connects to AD of the 4116, and AD1 connects to A6. Data coming into the VDP on CD0 goes to VRAM on AD0 and returns to the VDP on RD0.

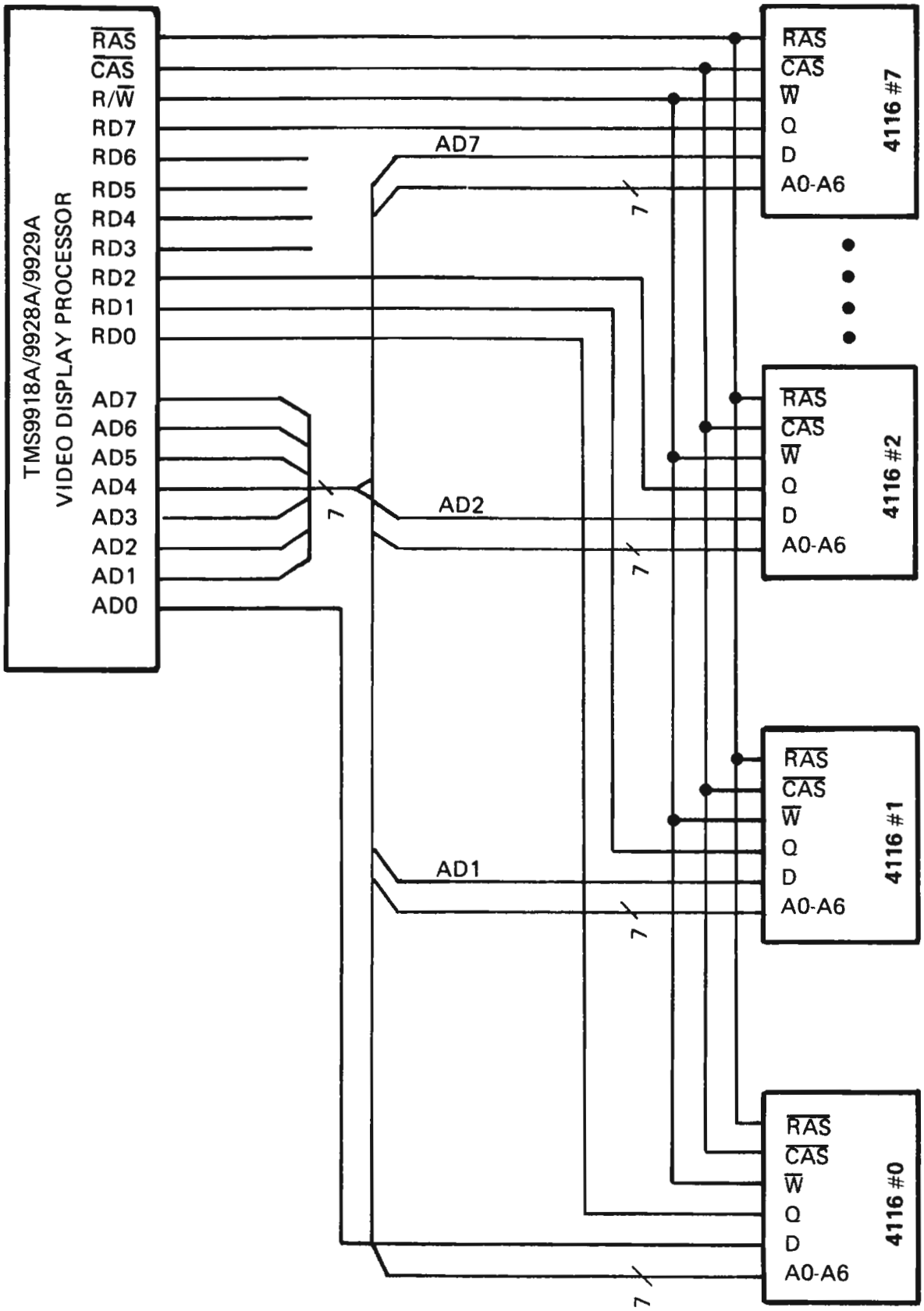



FIGURE 3-1 — VRAM INTERFACE

## 3.2 VRAM MEMORY ADDRESS DERIVATION

Table 3-2 summarizes the VRAM address derivation for all VDP modes of operation. Section 4 of this manual contains examples of how typical VRAM addresses are computed by the VDP.

**TABLE 3-2 – PATTERN GRAPHICS ADDRESS LOCATION TABLES**

**GRAPHICS I MODE ADDRESS LOCATION**

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

**GRAPHICS II MODE ADDRESS LOCATION**

| ADDRESS TYPE | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | COMMENTS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1) PATTERN NAME ADDRESS | NTB | | | | | | | | | | | | | | PATTERN NAME TABLE BASE (VDP REG2) |
| | | | | | ROW | | | | | | | | | | PATTERN POSITION ROW |
| | | | | | | | | | | | COLUMN | | | | PATTERN POSITION COLUMN |
| 2) PATTERN COLOR ADDRESS | | XX | | | | | | | | | | | | | PATTERN COLOR TABLE BASE MSB (VDP REG3) |
| | | | | | NAME | | | | | | | | | | TWO MSB FROM VERTICAL COUNTER |
| | | | | | | | | | | | | | | | ALL 8 BITS OF NAME |
| | | | | | | | | | | | | XXX | | | COLOR TABLE BYTE/LINE |
| 3) PATTERN GENERATOR ADDRESS | | XX | | | | | | | | | | | | | PATTERN NAME TABLE BASE MSB (VDP REG4) |
| | | | | | NAME | | | | | | | | | | TWO MSB FROM VERTICAL COUNTER |
| | | | | | | | | | | | | | | | ALL 8 BITS OF NAME |
| | | | | | | | | | | | | XXX | | | PATTERN GENERATOR BYTE/LINE NUMBER |

**TEXT MODE ADDRESS LOCATION**

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

TABLE 3-2 — PATTERN GRAPHICS ADDRESS LOCATION TABLES (CONTINUED)

SPRITE ADDRESS LOCATION

| 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)<br>SPRITE NUMBER<br>ATTRIBUTE NUMBER:<br>00 FOR VERTICAL POSITION<br>01 FOR HORIZONTAL POSITION<br>10 FOR NAME<br>11 FOR TAG (EARLY CLOCK AND COLOR) |
| SIZE = 0 SPRITE PATTERN GENERATOR | SPGB | | | | | NAME | | | | | | | XXX | | SPRITE PATTERN GENERATOR BASE (VDP REG4)<br>NAME ATTRIBUTE OF SPRITE<br>THREE LSB'S GIVE BYTE/LINE NUMBER |
| SIZE = 1 SPRITE PATTERN GENERATOR | SPGB | | | | | | NAME (0-5) | | | | | XXXXX | | | SPRITE PATTERN GENERATOR BASE (VDP REG4)<br>SIX MSB OF NAME<br>SIZE = 1 SPRITE BYTE NUMBER (SEE FIGURE 4-4) |

MULTICOLOR ADDRESS LOCATION

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

The TMS9918A/9928A operates at 262 lines per frame and approximately 60 frames per second in a noninterlaced mode of operation. The TMS9929A operates at 313 lines per frame and approximately 50 frames per second in a noninterlaced mode of operation.

## 3.3 VRAM ADDRESSING EXAMPLE

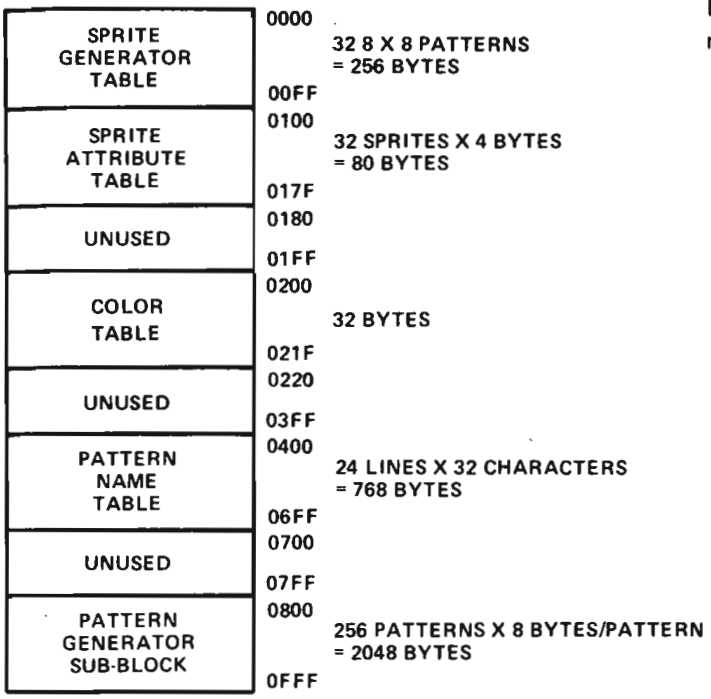
A typical application might require up to 256 unique 8 × 8 patterns with no more than 2 colors per pattern and up to 32 8 × 8 sprites.

These conditions dictate in which mode the VDP is to be used. The sprite requirement and the 8 × 8 pattern blocks eliminate the text and multicolor modes, respectively. This leaves only the Graphics I and Graphics II modes, and since two colors per block are all that are necessary, Graphics I is employed due to its ease of use.

Figure 3-2 shows a memory map that allows these functions to fit into a 4K memory area.

Register values for Figure 3-2 are as follows:

Register 0 = 00 External VDP disabled, M3 = 0
Register 1 = C0 16K DRAM selected, Blank = 1, Graphics 1 mode selected, SIZE = 0, MAG = 0
Register 2 = 01 Name Table Start Address @ >400
Register 3 = 08 Color Table Start Address @ >0200
Register 4 = 01 Pattern Generator Start Address @ >800
Register 5 = 02 Sprite Attribute Table Start Address >100
Register 6 = 00 Sprite Pattern Generator Start Address @>0000
Register 7 = XX Determined by user.




If the same application required 16 x 16 bit sprites, then the memory map could be modified as follows:

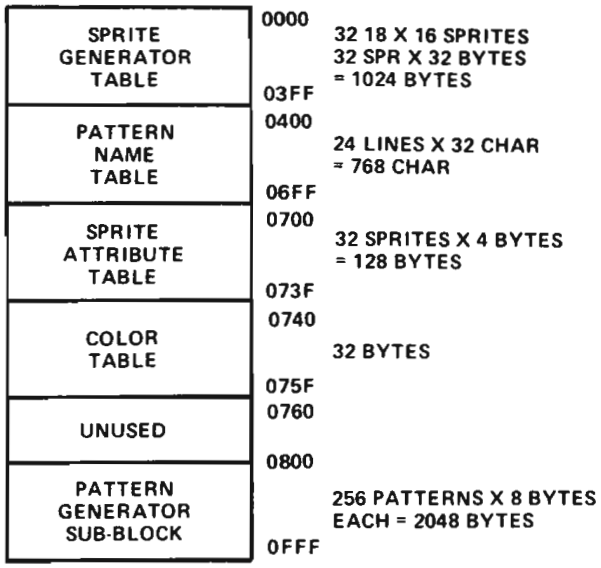



FIGURE 3-2 — VDP-VRAM MEMORY ALLOCATION

## 3.4 MONITOR INTERFACES

### 3.4.1 TMS9918A Monitor Interface

The composite video output signal from the TMS9918A drives a color 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 to determine if the VDP can be connected directly to it. The internal output buffer device on the composite video pin is a source-follower MOS transistor that requires an external pull-down resistor to $V_{SS}$ as shown in Figure 3-3. Typically a 330-ohm resistor is recommended to provide a 1.9-volt synchronization level. The load resistor (RL) defines the sharpness of the edges on the video signals. A lower resistor value gives faster fall times and a sharper picture.

In some cases, it may be necessary to provide a simple interface circuit to match the VDP output voltages with the monitor specifications. To drive a standard television that is not outfitted with a composite video input, the signal can be run into the television antenna terminals by using an appropriate RF modulator on the VDP output. Take care to ensure a proper match between VDP, RF modulator, and TV.

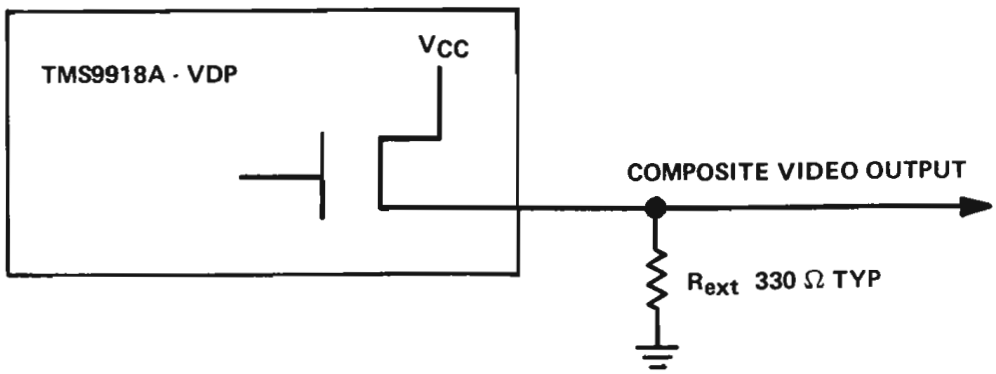
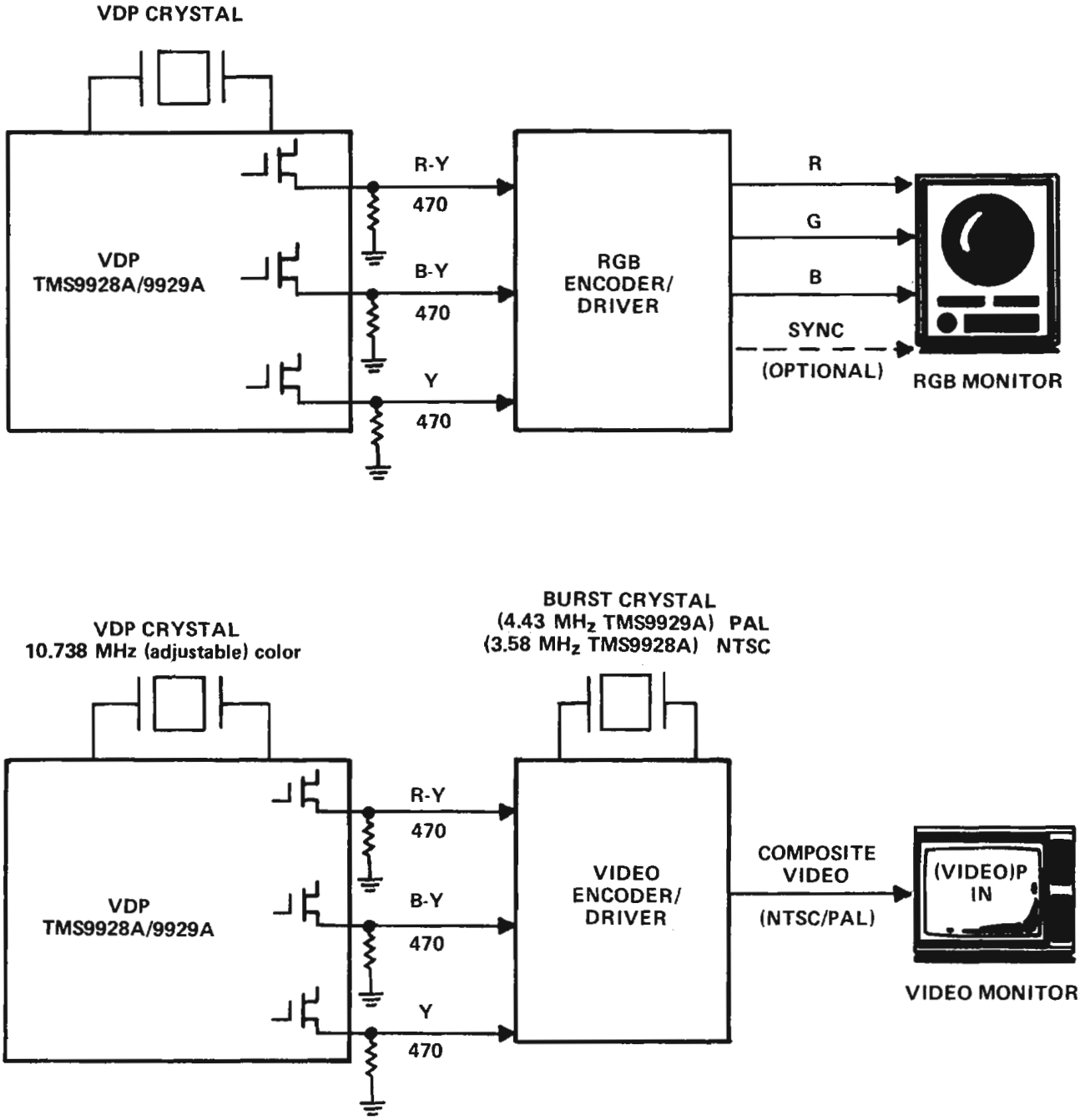



FIGURE 3-3 — COMPOSITE VIDEO PULL-DOWN CIRCUIT

### 3.4.2 TMS9928A/9929A Monitor Interface

The Y, R-Y and B-Y output signals require external encoder circuitry to drive a video color monitor; an R-G-B matrix circuitry is required to drive R-G-B color monitors. The Y output signal contains all necessary horizontal and vertical synchronization signals as well as luminance while the R-Y and B-Y signals contain the unmodulated chrominance information and are used in the NTSC and PAL systems to modulate two carriers in quadrature. The internal output buffer devices on these pins are source-follower MOS transistors that require an external pull-down resistor to $V_{SS}$, as shown in Figure 3-4. A 330 ohm resistor is recommended.




NOTE: The LM1889 is typically used in the video encoder circuitry.

FIGURE 3-4 — USE OF TMS9928A/9929A WITH DIFFERENT MONITORS

## 3.5 TMS9918A EXTERNAL VDP OPERATION

The external VDP interface allows cascading multiple VDPs. Figures 3-5 and 3-6 illustrate cascading two VDPs. Note that the VDPs must be reset by a common reset source to assure synchronization on an open loop basis. This reset source should have fast edges so that rise and fall times are less than 30 ns. Occasionally, synchronization is not obtained after reset, in which case, reset should be reapplied.

The video matching circuit ensures that the video signal of external VDP is biased correctly and of the proper amplitude. This ensures the luminance levels of the external and VDP colors are matched and external VDP video does not bleed through into the composite video output of the first VDP. The internal circuit assures that a perfect match results if the external video is of the same amplitude as the composite video of the VDP and its dc level is increased by a MOS threshold voltage (typically 0.7 volts). This adjustment can be varied to change the relative luminance levels of the two video signals and thus modify the picture appearance.

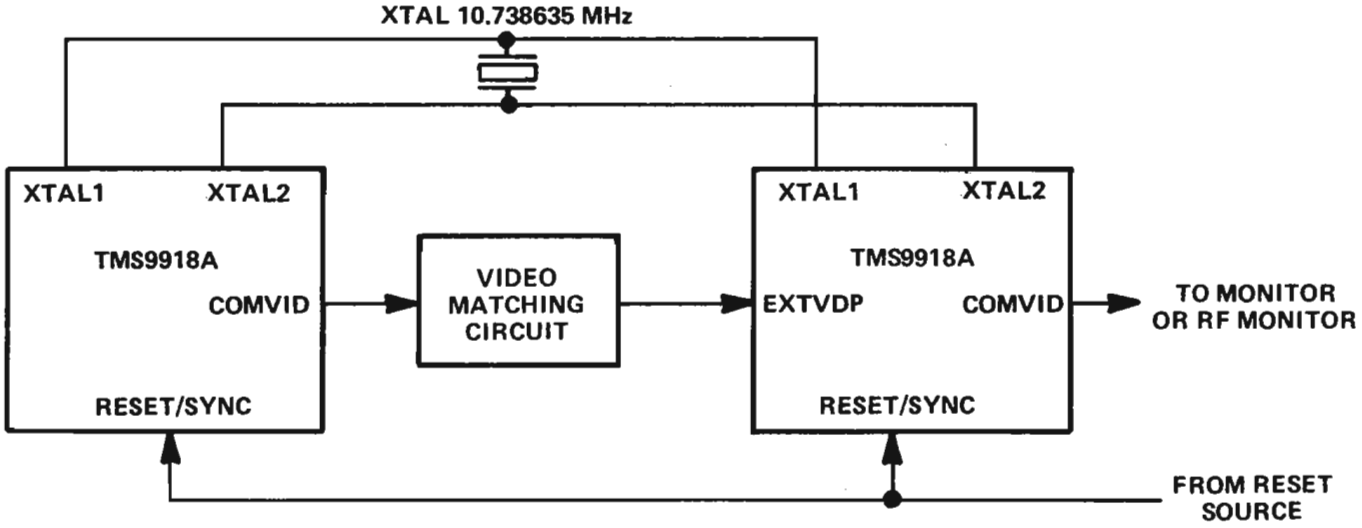



FIGURE 3-5 — CASCADING TWO TMS9918A VDPs

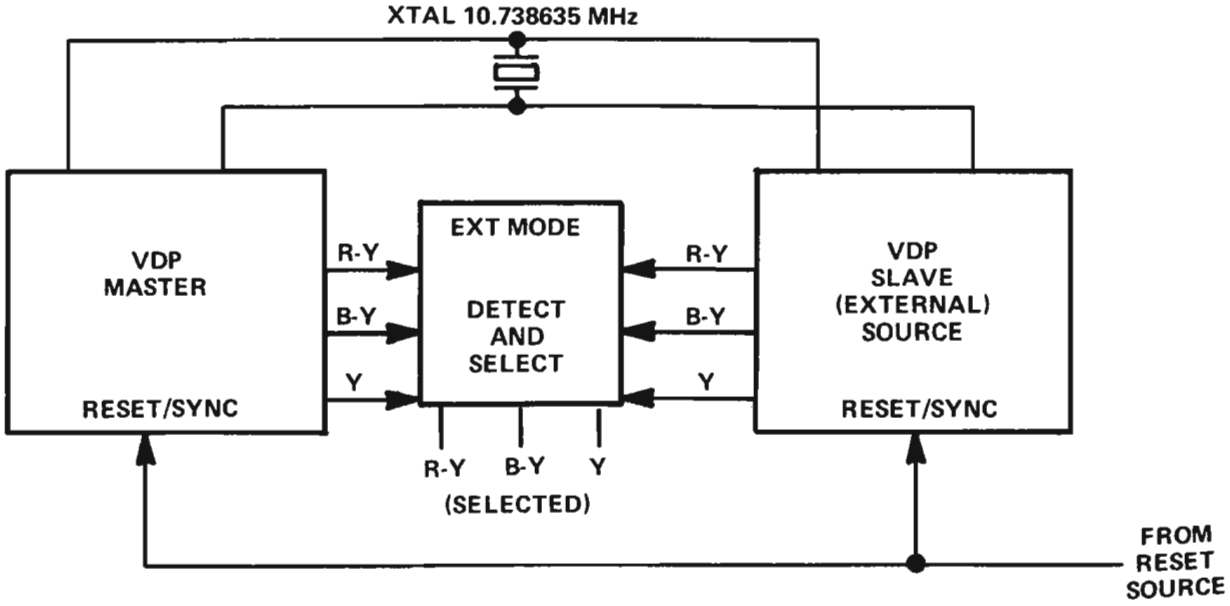



FIGURE 3-6 — CASCADING TWO TMS9918A/9929A VDPs

For the External VDP input plane to be visible, the External VDP Enable bit in VDP Register 0 (EXVID) should be set to a 1. The backdrop color (VDP Register 7, lower 4 bits) should be set to transparent (0). For the external VDP 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 VDP feature can be used in either Graphics I, Graphics II, Text, or Multicolor mode.

## 3.6 OSCILLATOR AND CLOCK GENERATION

The VDP is designed to operate with a 10.738635 ($\pm$ 0.005) MHz crystal input to generate the required internal clock signals. A fundamental-frequency parallel-mode crystal is 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 for TMS9918A only). The GROMCLK is developed from the master clock frequency divided by 24 (3.58 MHz for TMS9928A only).

### 3.6.1 TMS9918A Color Phase Generation

The 10.7 + MHz master clock and its complement generate an internal six-phase 3.579545 MHz ($\pm$ 10 Hz) clock to provide the video color signals and the color burst reference used 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 with the color and tint controls of the target color monitor.

### 3.6.2 Video Sync and Control Generation

Decoding the outputs of the horizontal and vertical counters generates the horizontal and vertical control signals. The pixel clock drives the horizontal counter which in turn increments the vertical counter.

Table 3-3 gives the relative count values of the screen display parameters. Within the active display area during Graphics I mode, the three LSBs of the horizontal counter address the individual picture element of each pattern displayed. Also, during the vertical active display period, the three LSBs of the vertical counter address each individual line in the 8 $\times$ 8 patterns. The Graphics II, Multicolor and Text modes use the counters similarly.

The TMS9918A/9929A operates at 262 lines per frame and approximately 60 frames per second in a noninterlaced mode of operation. The TMS9929A operates at 313 lines per frame and approximately 50 frames per second in a noninterlaced mode of operation.

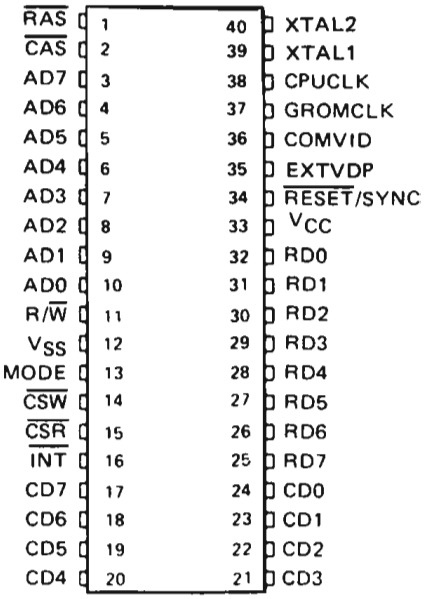
**TABLE 3-3 — SCREEN DISPLAY PARAMETERS**

| PARAMETER | PIXEL CLOCK CYCLES | |
|---|---|---|
| **HORIZONTAL** | **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 |
| | 342 | 342 |
| **VERTICAL** | **LINE** | |
| VERTICAL ACTIVE DISPLAY | 192 | |
| BOTTOM BORDER | 24 | |
| BOTTOM BLANKING | 3 | |
| VERTICAL SYNC | 3 | |
| TOP BLANKING | 13 | |
| TOP BORDER | 27 | |
| | 262 | |

## 3.7 VDP TERMINAL ASSIGNMENTS

### 3.7.1 TMS9918A Terminal Assignments

| SIGNATURE | TERMINAL | I/O | DESCRIPTION |
|---|---|---|---|
| XTAL1, XTAL2 | 40,39 | I | 10.7 + MHz crystal inputs* |
| CPUCLK | 38 | O | VDP color burst frequency clock. Typically not used on the TMS9918A, this is the color burst frequency clock. |
| GROMCLK | 37 | O | VDP output clock = XTAL/24. Typically not used. |
| COMVID | 36 | O | Composite video output for the TMS9918A. |
| EXTVDP | 35 | I/O | On the TMS9918A, this is the external VDP input. |
| $\overline{\text{RESET}}$/ SYNC | 34 | I | The $\overline{\text{RESET}}$ pin is a trilevel input pin. When it is below 0.8 volts, RESET initializes the VDP. When it is above 9 volts, RESET is the synchronizing input for external video. |
| $V_{CC}$ | 33 | I | + 5 volt supply |
| RD0 MSB | 32 | I | VRAM read data bus |
| RD1 | 31 | I | |
| RD2 | 30 | I | |
| RD3 | 29 | I | |
| RD4 | 28 | I | |
| RD5 | 27 | I | |
| RD6 | 26 | I | |
| RD7 | 25 | I | |
| CD0 MSB | 24 | I/O | CPU data bus; (CD0) is the most significant bit |
| CD1 | 23 | I/O | |
| CD2 | 22 | I/O | |
| CD3 | 21 | I/O | |
| CD4 | 20 | I/O | |
| CD5 | 19 | I/O | |
| CD6 | 18 | I/O | |
| CD7 LSB | 17 | I/O | |
| $\overline{\text{INT}}$ | 16 | O | CPU interrupt output. |
| $\overline{\text{CSR}}$ | 15 | I | CPU-VDP read strobe |
| $\overline{\text{CSW}}$ | 14 | I | CPU-VDP write strobe |
| MODE | 13 | I | CPU interface mode select; usually a processor address line |




* When driven externally, both inputs must be driven.

** The least significant address bit (AD7) is wired to A0 of the dynamic RAMs. Likewise, AD6 is wired to A1 of the RAMs. Care must be exercised in assuring proper orientation of the TMS 9918A address outputs to the dynamic RAM address inputs.

**TMS9918A Terminal Assignments (continued)**

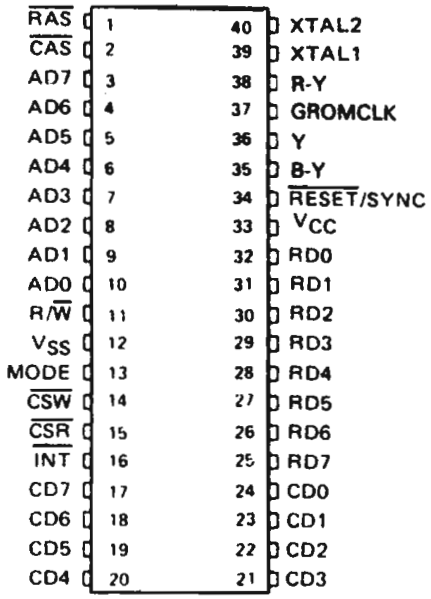
| SIGNATURE | TERMINAL | I/O | DESCRIPTION |
|---|---|---|---|
| $V_{SS}$ | 12 | I | Ground References |
| R/$\overline{W}$ | 11 | O | VRAM write strobe |
| AD0 MSB | 10 | O | VRAM address/data bus (multiplexed high and low order VRAM address and output data bytes) |
| AD1 | 9 | O | AD0 is the most significant 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 | |
| $\overline{CAS}$ | 2 | O | VRAM column address strobe |
| $\overline{RAS}$ | 1 | O | VRAM row address strobe |

* When driven externally, both inputs must be driven.

** The least significant address bit (AD7) is wired to A0 of the dynamic RAMs. Likewise, AD6 is wired to A1 of the RAMs. Care must be exercised in assuring proper orientation of the TMS 9918A address outputs to the dynamic RAM address inputs.

### 3.7.2 TMS9928A/9929A Terminal Assignments

| SIGNATURE | TERMINAL | I/O | DESCRIPTION |
|---|---|---|---|
| XTAL1, XTAL2 | 40,39 | I | 10.7 + MHz crystal inputs* |
| R-Y | 38 | O | VDP color burst frequency clock. On the TMS9928A/9929A, this is the R-Y color difference output. |
| GROMCLK | 37 | O | VDP output clock = XTAL/24. Typically not used. |
| Y | 36 | O | Composite video output. On the TMS9928A/9929A, this is the Y (black/white luminance and composite sync) output. |
| B-Y | 35 | I/O | External VDP input. On the TMS9928A/9929A, this is the B-Y color difference output. |
| $\overline{\text{RESET}}$/ SYNC | 34 | I | The $\overline{\text{RESET}}$ pin is a trilevel input pin. When it is below 0.8 volts, RESET initializes the VDP. When it is above 9 volts, RESET is the synchronizing input for external video. |
| $V_{CC}$ | 33 | I | +5 volt supply |
| RD0 MSB | 32 | I | VRAM read data bus |
| RD1 | 31 | I | |
| RD2 | 30 | I | |
| RD3 | 29 | I | |
| RD4 | 28 | I | |
| RD5 | 27 | I | |
| RD6 | 26 | I | |
| RD7 | 25 | I | |
| CD0 MSB | 24 | I/O | CPU data bus; (CD0) is the most significant bit |
| CD1 | 23 | I/O | |
| CD2 | 22 | I/O | |
| CD3 | 21 | I/O | |
| CD4 | 20 | I/O | |
| CD5 | 19 | I/O | |
| CD6 | 18 | I/O | |
| CD7 LSB | 17 | I/O | |
| $\overline{\text{INT}}$ | 16 | O | CPU interrupt output. |
| $\overline{\text{CSR}}$ | 15 | I | CPU-VDP read strobe |
| $\overline{\text{CSW}}$ | 14 | I | CPU-VDP write strobe |
| MODE | 13 | I | CPU interface mode select; usually a processor address line |




* When driven externally, both inputs must be driven.

** The least significant address bit (AD7) is wired to A0 of the dynamic RAMs. Likewise, AD6 is wired to A1 of the RAMs.

| SIGNATURE | TERMINAL | I/O | DESCRIPTION |
|---|---|---|---|
| $V_{SS}$ | 12 | I | Ground References |
| R/$\overline{W}$ | 11 | O | VRAM write strobe |
| AD0 MSB | 10 | O | VRAM address/data bus (multiplexed high and low order VRAM address and output data 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 | |
| $\overline{CAS}$ | 2 | O | VRAM row address strobe |
| $\overline{RAS}$ | 1 | O | VRam row address strobe |

* When driven externally, both inputs must be driven.
** The least significant address bit (AD7) is wired to A0 of the dynamic RAMs. Likewise, AD6 is wired to A1 of the RAMs.

### 3.7.3 TMS9918A/9928A/9929A Crystals

Crystals for the TMS9918A/9928A/9929A can be purchased from the following:

NDK
10080 North Wolfe Rd
Suite 220
Cuppertino, CA 95014
Telephone:
(408)255-0831
Telex: 352057

CTS Knights, Inc.
400 Reiman Ave
Sandwich, Ill 60548
Telephone:
(815)786-8411

# 4. DEVICE APPLICATIONS

This section describes the hardware and software interface between a TMS9918A/9928A/9929A VDP and a TMS9900 microprocessor. Some considerations in the use of the VDP for text and graphics applications are also described.

## 4.1 VDP TO TMS9900 INTERFACE

The circuit shown in Figure 4-1 illustrates a very simple interface between a TMS9900 microprocessor and a TMS9918A/9928A/9929A. In this circuit, the VDP 8-bit data bus is connected to the 8 MSBs of the TMS9900 16-bit data bus. For mode selection, A14 of the TMS9900 is connected to the mode input pin. Read and write signals to the VDP are as follows:

**CSR =A0*A13*DBIN**
**CSW =A0*WE**

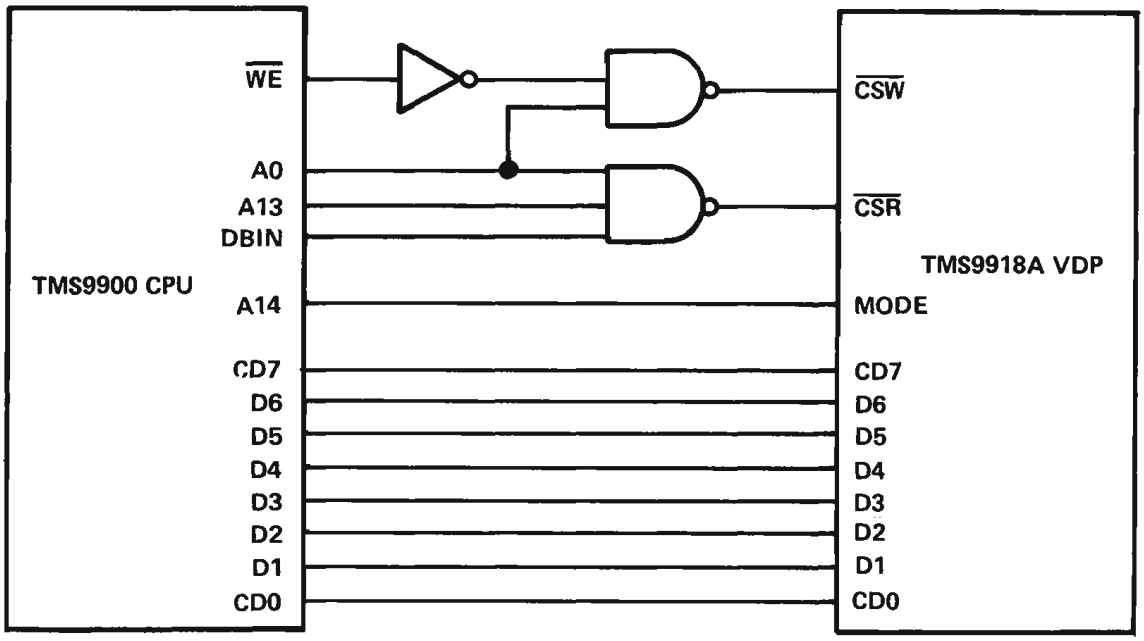



**FIGURE 4-1 – MINIMUM SYSTEM INTERFACE TO TMS9900**

DBIN and $\overline{WE}$ are signals from the TMS9900 which indicate direction flow on the data bus. DBIN is high when the CPU is attempting to do a read data operation, while $\overline{WE}$ is low when the CPU is outputting data onto the data bus.

A0 is used as a VDP select signal. Thus, the VDP is activated whenever the CPU is reading or writing data in the upper half of its address space (>8000 and above). All addresses above >8000 then become VDP port addresses. However, in a more sophisticated design, more decoding of the address lines would be done to select only those unique addresses required by the VDP. The purpose of A13 and decoding logic is to generate unique addresses for read and write operations and to block out the read data operation that occurs on the TMS9900 before a write data operation. Without this blockout logic, a pulse on the CSR input would occur before any desired pulsing of the CSW input, thus causing unwanted operation of the VDP. Referring to Table 4-1 and Figure 4-1, the following port addresses can be defined.

**TABLE 4-1 – VDP PORT ADDRESSES FOR FIGURE 4-1**

| OPERATION | $\overline{CSW}$ | $\overline{CSR}$ | MODE | PORT |
|---|---|---|---|---|
| Write data to VRAM | 0 | 1 | 0 | >8000 |
| Write address to VRAM or Write to VDP register | 0 | 1 | 1 | >8002 |
| Read data from VRAM | 1 | 0 | 0 | >8004 |
| Read VDP status | 1 | 0 | 1 | >8006 |

## 4.2 TMS9918A/9928A/9929A INTERFACE

Figures 4-2 and 4-3 show the hardware components necessary to make the VDP operate with a typical TM990 16-bit bus application. The CPU can be connected as shown to any general-purpose 8-bit data bus and control signals that work with most microprocessors. The VDP interface timing is similar to that of static memories and occupies eight unique memory address locations within the CPU memory address space.

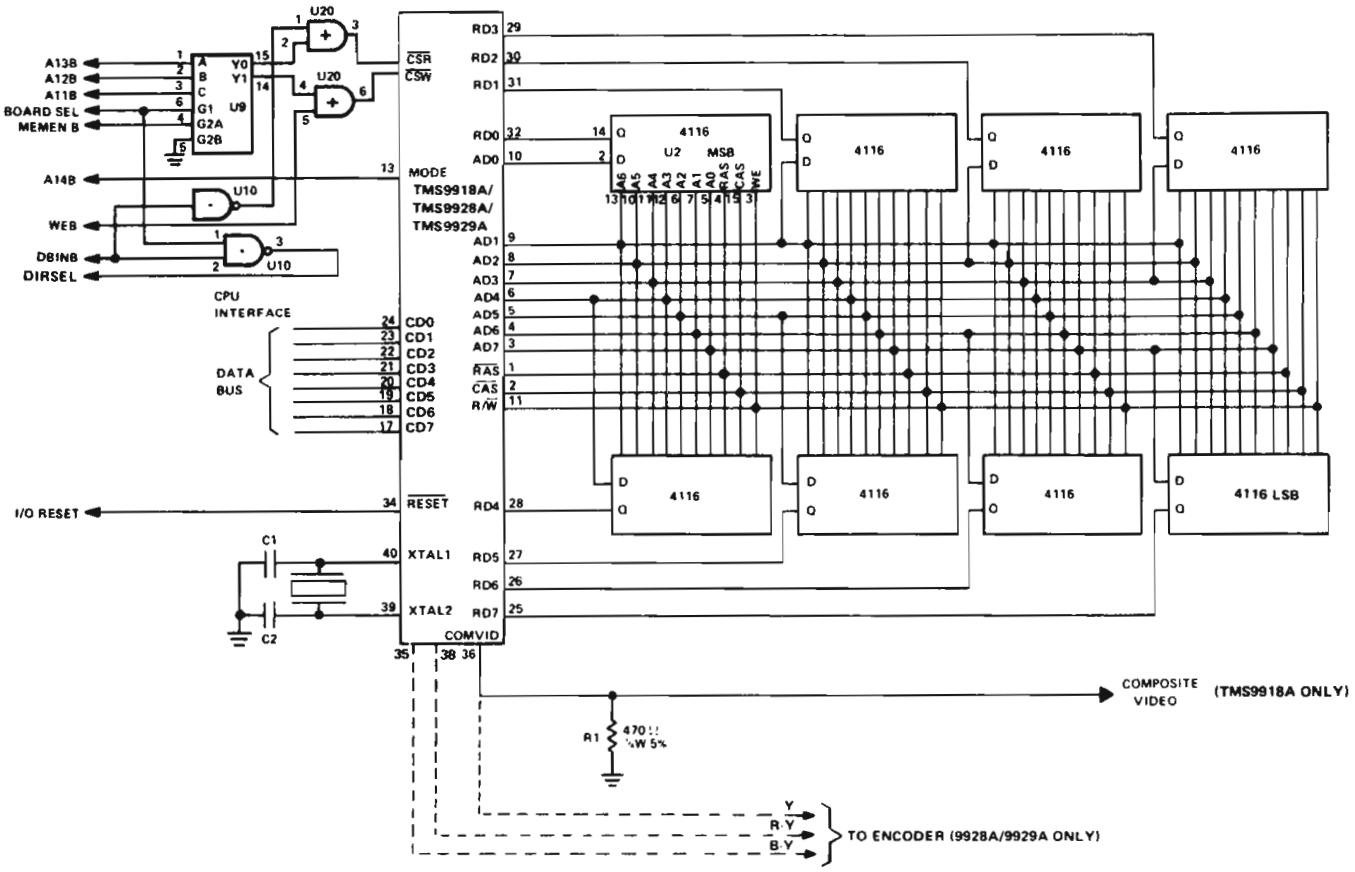



FIGURE 4-2 – TMS9918A/9928A/9929A INTERFACE

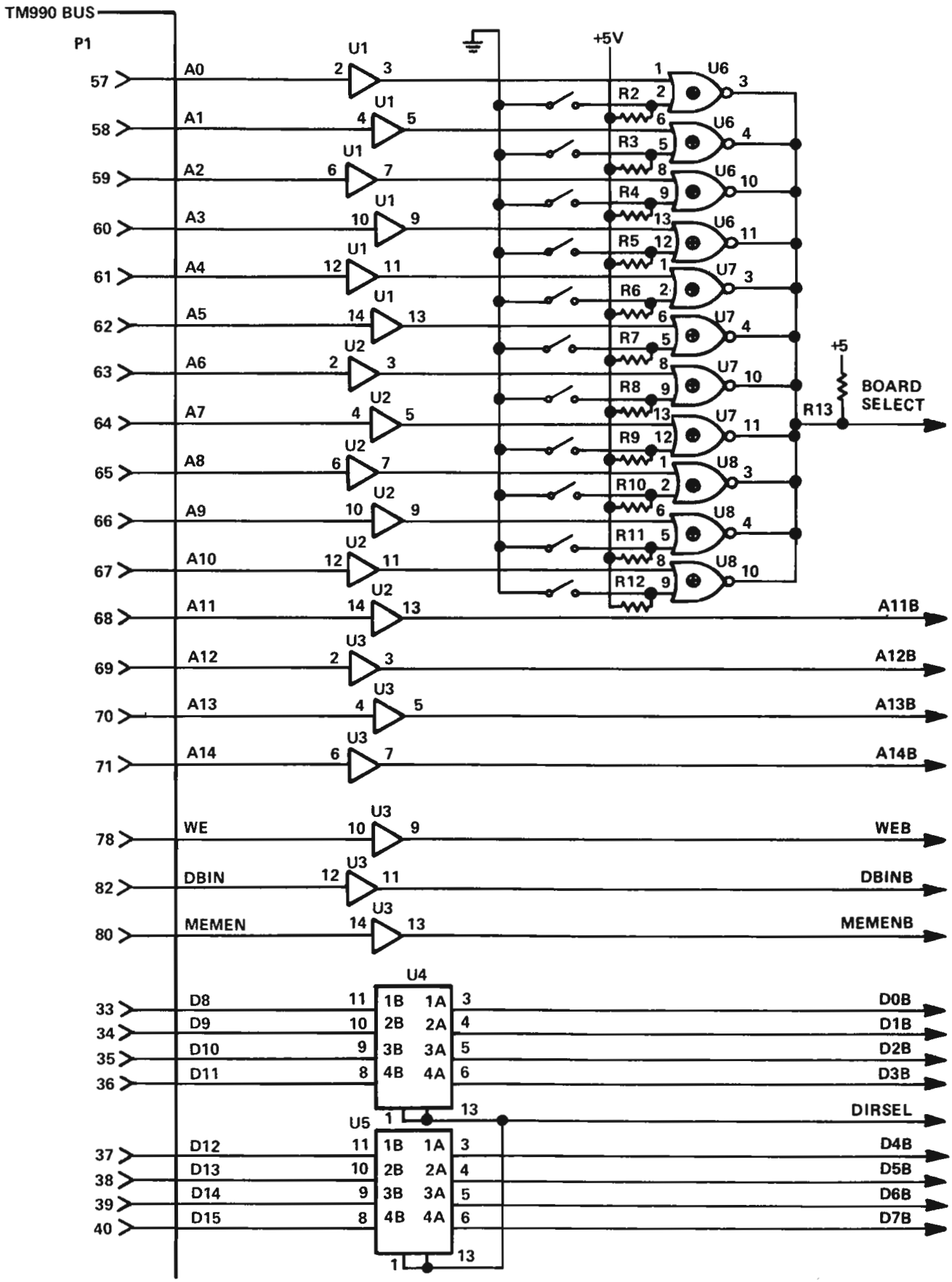



FIGURE 4-3 — TM990 (TMS9918A/9928A/9929A) DEMO BOARD

### 4.2.1 TM990 (TMS9918A/9928A/9929A) Parts List

| | |
|---|---|
| U1,2,3 | 74LS367 |
| U4,5 | 74LS243 |
| U6,7,8 | 74LS266 |
| U9 | 74LS138 |
| U10 | 74LS00 |
| U11 | TMS9918A/9928A/9929A |
| U12-19 | TMS4116 |
| C1,2 | 33 pF |
| Y1 | 10.738635 MHz Crystal |
| SW1-3 | 4-position DIP Switches |
| R1 | 470 Ω 5% 1/4 W |
| R2-R13 | Bourns XXXX or equivalent |

NOTE: All power supply pins of each IC should be bypassed with a .1μF capacitor.

### 4.2.2 Composite Video Output

The TMS9918A composite video output pin (36), is driven by a source-follower MOS transistor that requires an external pull-down resistor to $V_{SS}$. A 470-ohm resistor is typically used to provide a 1.9 volt peak-to-peak signal on the output. This output will drive most color directly, although in some cases it may be necessary to provide a simple interface circuit to match the monitor's input requirements. If a color video monitor is not available, an RF modulator can be used to drive the antenna terminals of a standard color television, as shown in Figure 4-4.

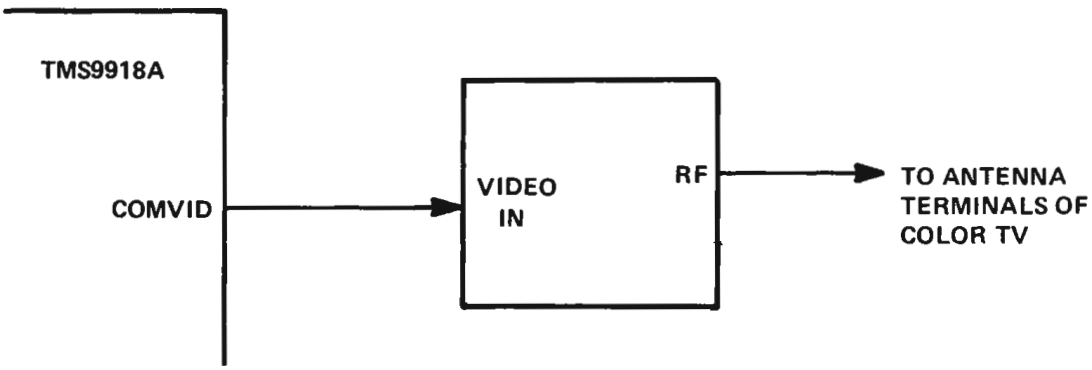



**FIGURE 4-4 – RF MODULATOR CONNECTION**

### 4.2.3 Oscillator and Timing

The TMS9918A/9928A/9929A internal timing generation is controlled by a self-contained oscillator and timing circuits. A 10.738635 (± 0.005%) MHz fundamental-frequency parallel-mode crystal is used to drive the basic oscillator frequency.

C1 and C2 are load capacitors for the parallel-resonant crystal. C1 and C2 values may be varied slightly to obtain more accuracy in timing and color generation and also to compensate for stray capacitance on the PC board. Typical values for C1 and C2 range between 15 pF and 39 pF. A trimmer capacitor with a value of 5 pF to 50 pF may also be used instead of C1 and adjusted to provide proper colors to the video monitor.

The VDP may also be operated with an external oscillator source. The VDP connections for this external source are shown in Figure 4-5.

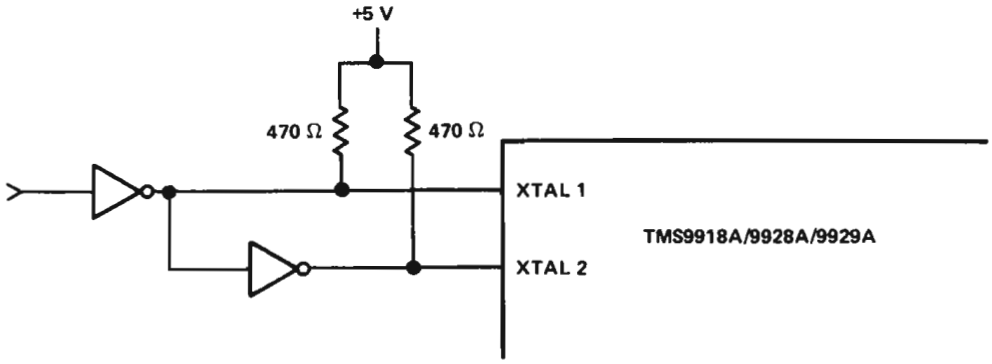



FIGURE 4-5 — EXTERNAL FREQUENCY SOURCE

There may be a slight color shift or a complete color loss in applications of RF modulators if there are mismatches in voltages levels or impedances between the VDP and the RF modulator. See Figure 3-4 for the TMS9928A/9929A interface.

### 4.2.4 VRAM Connections

The VRAM used in Figure 4-2 are 4116-type dynamic RAMs that meet the specifications in Section 5.

Addressing of the VRAM is done through the address bus and the memory control lines, AD1-AD7 and RAS, CAS, and WR, respectively.

Data written to the VRAM is also sent over the address bus. AD0 is a MSB, and AD7 is the LSB. Data written from the VRAM is brought into the VDP via the read data bus, RD0-RD7. The TMS9918A automatically refreshes the VRAM with no interaction necessary from the host CPU.

Note that address 0 (AD0) and data 0 (D0) are the MSBs for the TMS9918A and all other TMS9900 family members. The VRAM pin designations (A0 and D0) referenced in the data manual are shown as being the LSBs to be consistent with 4116-type dynamic RAM data sheets.

## 4.3 VDP INITIALIZATION

After powerup and proper reset timing, the VRAM allocation backdrop color and type of dynamic RAM need to be loaded into the VDP registers.

The values to be loaded can be calculated by using the examples and tables shown in Appendix A. The following flowchart (Figure 4-6) shows a procedure for loading all eight VDP registers. Setting 4.4 contains a typical TMS9900 software program designed to work on the demo board, shown in Figure 4-3.

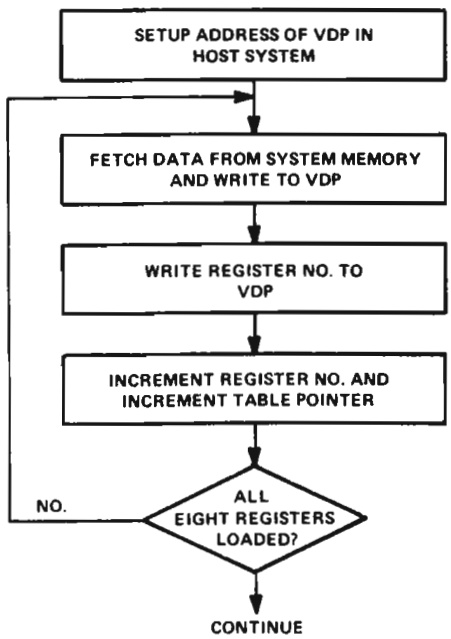



FIGURE 4-6 — VDP REGISTER INITIALIZATION PROCEDURE

## 4.4 TYPICAL SOFTWARE PROGRAM

### 4.4.1 General

This program initializes the TMS9918A and loads the Pattern Generator with the upper case character set. It then loads the color table, clears the screen and prints a sign-on message. After initialization, a user program address can be inserted at location 00A4.

```
DEMO9918   SDSMAC 3.4.0 81.117   15:45:22 MONDAY, SEP 27, 1982.
                                                          PAGE 0002

0001                    IDT 'DEMO9918'
0002 0000               AORG >0000
0003       9000  VRAMW  EQU  >9000 ADDRESS TO WRITE DATA TO VRAM
0004       9002  VDPW   EQU  >9002 ADDRESS TO WRITE DATA TO VDP
0005       9004  VRAMR  EQU  >9004 ADDRESS TO READ DATA FROM VRAM
0006       9006  VDPR   EQU  >9006 ADDRESS TO READ VDP STATUS REGISTER
0007             *
0008             ***************************************************************
0009             *          INITIALIZE THE 9918 WITH THE FOLLOWING:
0010             *
0011             *          REG 0 = 00  EXT VID OFF, GRAPH 2 OFF
0012             *
0013             *          REG 1 = 02  4116, INT DIS, VID ON, GRAPH 1
0014             *                      SIZE 1, MAG OFF
0015             *
0016             *          REG 2 = 01  NAME TABLE SUB BLOCK      @>400
0017             *
0018             *          REG 3 = 08  COLOR TABLE SUB BLOCK     @>200
0019             *
0020             *          REG 4 = 01  PATTERN GEN SUB BLOCK     @>800
0021             *
0022             *          REG 5 = 06  SPRITE NAME TAB SUB BLK   @>300
0023             *
0024             *          REG 6 = 00  SPRITE PATT GEN SUB BLK   @>000
0025             *
0026             *          REG 7 = 07  BACKDROP COLOR IS CYAN
0027             *
0028             ***************************************************************
0029             *
0030             *
0031             *                              NOTE
0032             *          THIS SOFTWARE ASSUMES THAT THE DATA BUS OF THE
0033             *          TMS9918A IS CONNECTED TO THE LEAST SIGNIFICANT
0034             *          BYTE OF THE TMS9900, WITH D7 AS THE MOST
0035             *          SIGNIFICANT BIT AND D15 AS THE LEAST SIGNIFICANT
0036             *          BIT
0037             *
0038             *
0039 0000 0201  INIT   LI   R1,VDPW          VDP WRITE ADDRESS
     0002 9002
0040 0004 0202         LI   R2,SUTA          "SET UP TABLE" ADDRESS
     0006 00B8
0041 0008 0203         LI   R3,>80           ADDRESS OF FIRST VDP REGISTER
     000A 0080
0042 000C C472  LP01   MOV  *R2+,*R1         GET DATA FROM MEM, SEND TO 9918
0043 000E C443         MOV  R3,*R1           SEND REG# TO 9918
0044 0010 0583         INC  R3               INCREMENT REGISTER COUNT
0045 0012 0283         CI   R3,>88           ALL REGS LOADED?
     0014 0088
0046 0016 16FA         JNE  LP01             NO,GO AGAIN
0047             *
0048             *
0049             *          LOAD PROGRAM LOADS THE TEXT PATTERNS FROM
0050             *          A TABLE IN MEMORY TO THE PATTERN GENERATOR
0051             *          SUB-BLOCK IN VRAM.
0052             *          ASCII >20 TO >5F ARE INCLUDED  IN THIS TABLE.
0053             *
0054             *
0055 0018 0201  LPG1   LI   R1,VRAMW         ADDRESS TO WRITE DATA TO VRAM
     001A 9000
```

```
0056 001C 0202          LI   R2,VDPW          ADDRESS TO WRITE TO VDP
     001E 9002
0057 0020 0203          LI   R3,PATT          MEM ADDR OF PATTERNS
     0022 00C0
0058 0024 0204          LI   R4,512           64 CHAR X 8 BYTES - 512 BYTES
     0026 0200
0059 0028 0205          LI   R5,>4900         ADDRESS TO LOAD PATS IN VRAM
     002A 4900
0060 002C C485          MOV  R5,*R2           SEND LSB OF VRAM ADDRESS TO VDP
0061 002E 06C5          SWPB R5               REVERSE BYTES
0062 0030 C445          MOV  R5,*R1           SEND DATA TO VRAM
0063 0032 D173  LPG2    MOVB *R3+,R5          GET BYTE FROM MEM
0064 0034 06C5          SWPB R5               REVERSE BYTES
0065 0036 C445          MOV  R5,*R1           SEND DATA TO VRAM
0066 0038 0604          DEC  R4               ALL DONE YET?
0067 003A 16FB          JNE  LPG2             NO, GO AGAIN
0068                    *
0069                    ********************************************************
0070                    *                    LOAD COLOR TABLE
0071                    *
0072                    *           THIS ROUTINE LOADS THE COLOR TABLE FOR THE
0073                    *           TEXT PATTERNS JUST ENTERED.
0074                    *
0075                    *
0076 003C 0201          LI   R1,VRAMW         ADDRESS TO WRITE DATA TO VRAM
     003E 9000
0077 0040 0202          LI   R2,VDPW          ADDRESS TO WRITE TO VDP
     0042 9002
0078 0044 0203          LI   R3,>4204         START ADDRESS OF TEXT COLOR TABL
     0046 4204
0079 0048 0204          LI   R4,>5F           CHARACTERS WILL BE BLUE ON WHITE
     004A 005F
0080 004C C483          MOV  R3,*R2           SEND LSB OF VRAM ADDRESS TO VDP
0081 004E 0205          LI   R5,8             LOAD COUNT VALUE, 64CHAR/8 = 8
     0050 0008
0082 0052 C444  LCTL    MOV  R4,*R1           SEND COLOR INFO TO VRAM
0083 0054 0605          DEC  R5               TABLE LAODED YET?
0084 0056 16FD          JNE  LCTL             NO, GO AGAIN
0085                    *
0086                    ********************************************************
0087                    *                              CLEAR SCREEN
0088                    *
0089                    *           THIS ROUTINE CLEARS THE SCREEN BY WRITING A SPACE
0090                    *           CHARACTER (ASCII >20) TO ALL LOCATIONS IN THE
0091                    *           NAME TABLE.
0092                    *
0093                    *
0094 0058 0201          LI   R1,VRAMW         ADDRESS TO WRITE DATA TO VRAM
     005A 9000
0095 005C 0202          LI   R2,VDPW          ADDRESS TO WRITE TO VDP
     005E 9002
0096 0060 0203          LI   R3,>4400         START ADDRESS IN NAME TABLE
     0062 4400
0097 0064 C483          MOV  R3,*R2           SEND MSB OF VRAM ADDRESS TO VDP
0098 0066 0202          LI   R2,768           #OF POSITIONS ON SCREEN
     0068 0300
0099 006A 0203          LI   R3,>20           ASCII SPACE CHAR
     006C 0020
0100 006E C443  CSL1    MOV  R3,*R1           SEND SPACE TO SCREEN
0101 0070 0602          DEC  R2               ARE ALL LOCATIONS CLEAR?
```

```
0102 0072 16FD          JNE  CSL1               NO, GO AGAIN
0103                *
0104                ************************************************************
0105                *                      PRINT SIGN ON MESSAGE
0106                *                  AND BRANCH TO USERS PROGRAM
0107                *
0108                *
0109 0074 0201          LI   R1,VRAMW           ADDRESS TO WRITE DATA TO VRAM
     0076 9000
0110 0078 0202          LI   R2,VDPW            ADDRESS TO WRITE TO VDP
     007A 9002
0111 007C 0203          LI   R3,>4400           POSITION OF MESSAGE ON SCREEN
     007E 4400
0112 0080 C483          MOV  R3,*R2             SEND MSB OF VRAM ADDRESS TO VDP
0113 0082 06C3          SWPB R3                 REVERSE BYTES
0114 0084 C483          MOV  R3,*R2             SEND MSB OF VRAM ADDRESS TO VDP
0115 0086 0203          LI   R3,MSG0            ADDRESS OF SIGN ON MESSAGE
     0088 009E
0116 008A 04C4  PRNT    CLR  R4                 CLEAR RECEPTION REGISTER
0117 008C D113          MOVB *R3,R4             GET A BYTE OF TEXT
0118 008E 0284          CI   R4,>FF00           IS IT THE EOM CHARACTER?
     0090 FF00
0119 0092 1303          JEQ  DONE               YES, GOTO NEXT PROGRAM SEGMENT
0120 0094 06C4          SWPB R4                 REVERSE  BYTES
0121 0096 C444          MOV  R4,*R1             SEND CHAR TO VRAM
0122 0098 10F8          JMP  PRNT               GET NEXT CHARACTER
0123 009A 0460  DONE    B    @DONE              INSERT BRANCH TO USERS PROGRAM
     009C 009A
0124                *                           AT THIS POINT
0125 009E   54  MSG0    TEXT 'TEXAS INSTRUMENTS TMS9918'
     009F   45
     00A0   58
     00A1   41
     00A2   53
     00A3   20
     00A4   49
     00A5   4E
     00A6   53
     00A7   54
     00A8   52
     00A9   55
     00AA   4D
     00AB   45
     00AC   4E
     00AD   54
     00AE   53
     00AF   20
     00B0   54
     00B1   4D
     00B2   53
     00B3   39
     00B4   39
     00B5   31
     00B6   38
0126 00B7   FF          BYTE >FF
0127 00B8               EVEN
0128                *
0129                ************************************************************
0130                *           THIS TABLE CONTAINS THE VALUES FOR
0131                *         INITIALIZING THE REGISTERS IN THE 9918A
```

```
0132                    *
0133 00B8   00  SUTA    BYTE >00
0134 00B9   02          BYTE >02
0135 00BA   01          BYTE >01
0136 00BB   08          BYTE >08
0137 00BC   01          BYTE >01
0138 00BD   06          BYTE >06
0139 00BE   00          BYTE >00
0140 00BF   07          BYTE >07
0141                    *
0142                    ***************************************************************
0143                    *                   9918A TEXT PATTERNS
0144                    *
0145                    *           THESE PATTERNS FROM A 5X7 CHARACTER IN THE
0146                    *           8X8 PATTERN BLOCK THAT IS UPPER AND LEFT
0147                    *           JUSTIFIED
0148                    *
0149 00C0 0000  PATT    DATA >0000          CHARACTER SPACE ASCII 20
0150 00C2 0000          DATA >0000
0151 00C4 0000          DATA >0000
0152 00C6 0000          DATA >0000
0153 00C8 2020          DATA >2020          CHARACTER !     ASCII 21
0154 00CA 2020          DATA >2020
0155 00CC 2000          DATA >2000
0156 00CE 2000          DATA >2000
0157 00D0 5050          DATA >5050          CHARACTER "     ASCII 22
0158 00D2 5000          DATA >5000
0159 00D4 0000          DATA >0000
0160 00D6 0000          DATA >0000
0161 00D8 5050          DATA >5050          CHARACTER #     ASCII 23
0162 00DA F850          DATA >F850
0163 00DC F850          DATA >F850
0164 00DE 5000          DATA >5000
0165 00E0 2078          DATA >2078          CHARACTER $     ASCII 24
0166 00E2 A070          DATA >A070
0167 00E4 28F0          DATA >28F0
0168 00E6 2000          DATA >2000
0169 00E8 C0C8          DATA >C0C8          CHARACTER %     ASCII 25
0170 00EA 1020          DATA >1020
0171 00EC 4098          DATA >4098
0172 00EE 1800          DATA >1800
0173 00F0 40A0          DATA >40A0          CHARACTER &     ASCII 26
0174 00F2 A040          DATA >A040
0175 00F4 A890          DATA >A890
0176 00F6 6800          DATA >6800
0177 00F8 2020          DATA >2020          CHARACTER '     ASCII 27
0178 00FA 2000          DATA >2000
0179 00FC 0000          DATA >0000
0180 00FE 0000          DATA >0000
0181 0100 2040          DATA >2040          CHARACTER (     ASCII 28
0182 0102 8080          DATA >8080
0183 0104 8040          DATA >8040
0184 0106 2000          DATA >2000
0185 0108 2010          DATA >2010          CHARACTER )     ASCII 29
0186 010A 0808          DATA >0808
0187 010C 0810          DATA >0810
0188 010E 2000          DATA >2000
0189 0110 20A8          DATA >20A8          CHARACTER *     ASCII 2A
0190 0112 7020          DATA >7020
0191 0114 70A8          DATA >70A8
```

```
0192 0116 2000          DATA >2000
0193 0118 0020          DATA >0020          CHARACTER +     ASCII 2B
0194 011A 20F8          DATA >20F8
0195 011C 2020          DATA >2020
0196 011E 0000          DATA >0000
0197 0120 0000          DATA >0000          CHARACTER ,     ASCII 2C
0198 0122 0000          DATA >0000
0199 0124 2020          DATA >2020
0200 0126 4000          DATA >4000
0201 0128 0000          DATA >0000          CHARACTER -     ASCII 2D
0202 012A 00F8          DATA >00F8
0203 012C 0000          DATA >0000
0204 012E 0000          DATA >0000
0205 0130 0000          DATA >0000          CHARACTER .     ASCII 2E
0206 0132 0000          DATA >0000
0207 0134 0000          DATA >0000
0208 0136 2000          DATA >2000
0209 0138 0008          DATA >0008          CHARACTER /     ASCII 2F
0210 013A 1020          DATA >1020
0211 013C 4080          DATA >4080
0212 013E 0000          DATA >0000
0213 0140 7088          DATA >7088          CHARACTER 0     ASCII 30
0214 0142 98A8          DATA >98A8
0215 0144 C888          DATA >C888
0216 0146 7000          DATA >7000
0217 0148 2060          DATA >2060          CHARACTER 1     ASCII 31
0218 014A 2020          DATA >2020
0219 014C 2020          DATA >2020
0220 014E 7000          DATA >7000
0221 0150 7088          DATA >7088          CHARACTER 2     ASCII 32
0222 0152 0830          DATA >0830
0223 0154 4080          DATA >4080
0224 0156 F800          DATA >F800
0225 0158 F808          DATA >F808          CHARACTER 3     ASCII 33
0226 015A 1030          DATA >1030
0227 015C 0888          DATA >0888
0228 015E 7000          DATA >7000
0229 0160 1030          DATA >1030          CHARACTER 4     ASCII 34
0230 0162 5090          DATA >5090
0231 0164 F810          DATA >F810
0232 0166 1000          DATA >1000
0233 0168 F880          DATA >F880          CHARACTER 5     ASCII 35
0234 016A F008          DATA >F008
0235 016C 0888          DATA >0888
0236 016E 7000          DATA >7000
0237 0170 3840          DATA >3840          CHARACTER 6     ASCII 36
0238 0172 80F0          DATA >80F0
0239 0174 8888          DATA >8888
0240 0176 7000          DATA >7000
0241 0178 F808          DATA >F808          CHARACTER 7     ASCII 37
0242 017A 1020          DATA >1020
0243 017C 4040          DATA >4040
0244 017E 4000          DATA >4000
0245 0180 7088          DATA >7088          CHARACTER 8     ASCII 38
0246 0182 8870          DATA >8870
0247 0184 8888          DATA >8888
0248 0186 7000          DATA >7000
0249 0188 7088          DATA >7088          CHARACTER 9     ASCII 39
0250 018A 8878          DATA >8878
0251 018C 0810          DATA >0810
```

```
0252 018E E000          DATA >E000
0253 0190 0000          DATA >0000          CHARACTER :    ASCII 3A
0254 0192 2000          DATA >2000
0255 0194 2000          DATA >2000
0256 0196 0000          DATA >0000
0257 0198 0000          DATA >0000          CHARACTER ;    ASCII 3B
0258 019A 2000          DATA >2000
0259 019C 2020          DATA >2020
0260 019E 4000          DATA >4000
0261 01A0 1020          DATA >1020          CHARACTER <    ASCII 3C
0262 01A2 4080          DATA >4080
0263 01A4 4020          DATA >4020
0264 01A6 1000          DATA >1000
0265 01A8 0000          DATA >0000          CHARACTER =    ASCII 3D
0266 01AA F800          DATA >F800
0267 01AC F800          DATA >F800
0268 01AE 0000          DATA >0000
0269 01B0 4020          DATA >4020          CHARACTER >    ASCII 3E
0270 01B2 1008          DATA >1008
0271 01B4 1020          DATA >1020
0272 01B6 4000          DATA >4000
0273 01B8 7088          DATA >7088          CHARACTER ?    ASCII 3F
0274 01BA 1020          DATA >1020
0275 01BC 2000          DATA >2000
0276 01BE 2000          DATA >2000
0277 01C0 7088          DATA >7088          CHARACTER @    ASCII 40
0278 01C2 A8B8          DATA >A8B8
0279 01C4 B080          DATA >B080
0280 01C6 7800          DATA >7800
0281 01C8 2050          DATA >2050          CHARACTER A    ASCII 41
0282 01CA 8888          DATA >8888
0283 01CC F888          DATA >F888
0284 01CE 8800          DATA >8800
0285 01D0 F088          DATA >F088          CHARACTER B    ASCII 42
0286 01D2 88F0          DATA >88F0
0287 01D4 8888          DATA >8888
0288 01D6 F000          DATA >F000
0289 01D8 7088          DATA >7088          CHARACTER C    ASCII 43
0290 01DA 8080          DATA >8080
0291 01DC 8088          DATA >8088
0292 01DE 7000          DATA >7000
0293 01E0 F088          DATA >F088          CHARACTER D    ASCII 44
0294 01E2 8888          DATA >8888
0295 01E4 8888          DATA >8888
0296 01E6 F000          DATA >F000
0297 01E8 F880          DATA >F880          CHARACTER E    ASCII 45
0298 01EA 80F0          DATA >80F0
0299 01EC 8080          DATA >8080
0300 01EE F800          DATA >F800
0301 01F0 F880          DATA >F880          CHARACTER F    ASCII 46
0302 01F2 80F0          DATA >80F0
0303 01F4 8080          DATA >8080
0304 01F6 8000          DATA >8000
0305 01F8 7880          DATA >7880          CHARACTER G    ASCII 47
0306 01FA 8080          DATA >8080
0307 01FC 9888          DATA >9888
0308 01FE 7800          DATA >7800
0309 0200 8888          DATA >8888          CHARACTER H    ASCII 48
0310 0202 88F8          DATA >88F8
0311 0204 8888          DATA >8888
```

```
0312 0206 8800          DATA >8800
0313 0208 7020          DATA >7020          CHARACTER I    ASCII 49
0314 020A 2020          DATA >2020
0315 020C 2020          DATA >2020
0316 020E 7000          DATA >7000
0317 0210 0808          DATA >0808          CHARACTER J    ASCII 4A
0318 0212 0808          DATA >0808
0319 0214 0888          DATA >0888
0320 0216 7000          DATA >7000
0321 0218 8890          DATA >8890          CHARACTER K    ASCII 4B
0322 021A A0C0          DATA >A0C0
0323 021C A090          DATA >A090
0324 021E 8800          DATA >8800
0325 0220 8080          DATA >8080          CHARACTER L    ASCII 4C
0326 0222 8080          DATA >8080
0327 0224 8080          DATA >8080
0328 0226 F800          DATA >F800
0329 0228 88D8          DATA >88D8          CHARACTER M    ASCII 4D
0330 022A A8A8          DATA >A8A8
0331 022C 8888          DATA >8888
0332 022E 8800          DATA >8800
0333 0230 8888          DATA >8888          CHARACTER N    ASCII 4E
0334 0232 C8A8          DATA >C8A8
0335 0234 9888          DATA >9888
0336 0236 8800          DATA >8800
0337 0238 7088          DATA >7088          CHARACTER O    ASCII 4F
0338 023A 8888          DATA >8888
0339 023C 8888          DATA >8888
0340 023E 7000          DATA >7000
0341 0240 F088          DATA >F088          CHARACTER P    ASCII 50
0342 0242 88F0          DATA >88F0
0343 0244 8080          DATA >8080
0344 0246 8000          DATA >8000
0345 0248 7088          DATA >7088          CHARACTER Q    ASCII 51
0346 024A 8888          DATA >8888
0347 024C A890          DATA >A890
0348 024E 6800          DATA >6800
0349 0250 F088          DATA >F088          CHARACTER R    ASCII 52
0350 0252 88F0          DATA >88F0
0351 0254 A090          DATA >A090
0352 0256 8800          DATA >8800
0353 0258 7088          DATA >7088          CHARACTER S    ASCII 53
0354 025A 8070          DATA >8070
0355 025C 0888          DATA >0888
0356 025E 7000          DATA >7000
0357 0260 F820          DATA >F820          CHARACTER T    ASCII 54
0358 0262 2020          DATA >2020
0359 0264 2020          DATA >2020
0360 0266 2000          DATA >2000
0361 0268 8888          DATA >8888          CHARACTER U    ASCII 55
0362 026A 8888          DATA >8888
0363 026C 8888          DATA >8888
0364 026E 7000          DATA >7000
0365 0270 8888          DATA >8888          CHARACTER V    ASCII 56
0366 0272 8888          DATA >8888
0367 0274 8850          DATA >8850
0368 0276 2000          DATA >2000
0369 0278 8888          DATA >8888          CHARACTER W    ASCII 57
0370 027A 88A8          DATA >88A8
0371 027C A8D8          DATA >A8D8
```

```
0372 027E 8800          DATA >8800
0373 0280 8888          DATA >8888          CHARACTER X     ASCII 58
0374 0282 5020          DATA >5020
0375 0284 5088          DATA >5088
0376 0286 8800          DATA >8800
0377 0288 8888          DATA >8888          CHARACTER Y     ASCII 59
0378 028A 5020          DATA >5020
0379 028C 2020          DATA >2020
0380 028E 2000          DATA >2000
0381 0290 F808          DATA >F808          CHARACTER Z     ASCII 5A
0382 0292 1020          DATA >1020
0383 0294 4080          DATA >4080
0384 0296 F800          DATA >F800
0385 0298 F8C0          DATA >F8C0          CHARACTER [     ASCII 5B
0386 029A C0C0          DATA >C0C0
0387 029C C0C0          DATA >C0C0
0388 029E F800          DATA >F800
0389 02A0 0080          DATA >0080          CHARACTER       ASCII 5C
0390 02A2 4020          DATA >4020
0391 02A4 1008          DATA >1008
0392 02A6 0000          DATA >0000
0393 02A8 F818          DATA >F818          CHARACTER ]     ASCII 5D
0394 02AA 1818          DATA >1818
0395 02AC 1818          DATA >1818
0396 02AE F800          DATA >F800
0397 02B0 0000          DATA >0000          CHARACTER       ASCII 5E
0398 02B2 2050          DATA >2050
0399 02B4 8800          DATA >8800
0400 02B6 0000          DATA >0000
0401 02B8 0000          DATA >0000          CHARACTER _     ASCII 5F
0402 02BA 0000          DATA >0000
0403 02BC 0000          DATA >0000
0404 02BE F800          DATA >F800
0405 02C0 4020          DATA >4020          CHARACTER       ASCII 60
0406 02C2 1000          DATA >1000
0407 02C4 0000          DATA >0000
0408 02C6 0000          DATA >0000
0409 02C8 0000          DATA >0000          CHARACTER a     ASCII 61
0410 02CA 7088          DATA >7088
0411 02CC F888          DATA >F888
0412 02CE 8800          DATA >8800
0413 02D0 0000          DATA >0000          CHARACTER b     ASCII 62
0414 02D2 F048          DATA >F048
0415 02D4 7048          DATA >7048
0416 02D6 F000          DATA >F000
0417 02D8 0000          DATA >0000          CHARACTER c     ASCII 63
0418 02DA 7880          DATA >7880
0419 02DC 8080          DATA >8080
0420 02DE 7800          DATA >7800
0421 02E0 0000          DATA >0000          CHARACTER d     ASCII 64
0422 02E2 F048          DATA >F048
0423 02E4 4848          DATA >4848
0424 02E6 F000          DATA >F000
0425 02E8 0000          DATA >0000          CHARACTER e     ASCII 65
0426 02EA F080          DATA >F080
0427 02EC E080          DATA >E080
0428 02EE F000          DATA >F000
0429 02F0 0000          DATA >0000          CHARACTER f     ASCII 66
0430 02F2 F080          DATA >F080
0431 02F4 E080          DATA >E080
```

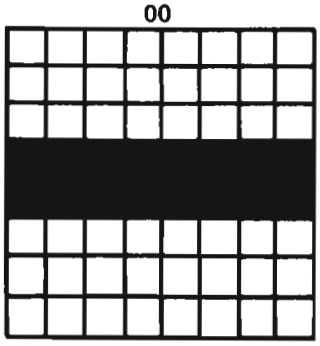
```
0432 02F6 8000          DATA >8000
0433 02F8 0000          DATA >0000          CHARACTER g     ASCII 67
0434 02FA 7880          DATA >7880
0435 02FC B888          DATA >B888
0436 02FE 7000          DATA >7000
0437 0300 0000          DATA >0000          CHARACTER h     ASCII 68
0438 0302 8888          DATA >8888
0439 0304 F888          DATA >F888
0440 0306 8800          DATA >8800
0441 0308 0000          DATA >0000          CHARACTER i     ASCII 69
0442 030A F820          DATA >F820
0443 030C 2020          DATA >2020
0444 030E F800          DATA >F800
0445 0310 0000          DATA >0000          CHARACTER j     ASCII 6A
0446 0312 7020          DATA >7020
0447 0314 20A0          DATA >20A0
0448 0316 E000          DATA >E000
0449 0318 0000          DATA >0000          CHARACTER k     ASCII 6B
0450 031A 90A0          DATA >90A0
0451 031C A0C0          DATA >A0C0
0452 031E 9000          DATA >9000
0453 0320 0000          DATA >0000          CHARACTER l     ASCII 6C
0454 0322 8080          DATA >8080
0455 0324 8080          DATA >8080
0456 0326 F800          DATA >F800
0457 0328 0000          DATA >0000          CHARACTER m     ASCII 6D
0458 032A 88D8          DATA >88D8
0459 032C A888          DATA >A888
0460 032E 8800          DATA >8800
0461 0330 0000          DATA >0000          CHARACTER n     ASCII 6E
0462 0332 88C8          DATA >88C8
0463 0334 A898          DATA >A898
0464 0336 8800          DATA >8800
0465 0338 0000          DATA >0000          CHARACTER o     ASCII 6F
0466 033A F888          DATA >F888
0467 033C 8888          DATA >8888
0468 033E F800          DATA >F800
0469 0340 0000          DATA >0000          CHARACTER p     ASCII 70
0470 0342 F088          DATA >F088
0471 0344 F080          DATA >F080
0472 0346 8000          DATA >8000
0473 0348 0000          DATA >0000          CHARACTER q     ASCII 71
0474 034A F888          DATA >F888
0475 034C A890          DATA >A890
0476 034E E000          DATA >E000
0477 0350 0000          DATA >0000          CHARACTER r     ASCII 72
0478 0352 F888          DATA >F888
0479 0354 F8A0          DATA >F8A0
0480 0356 9000          DATA >9000
0481 0358 0000          DATA >0000          CHARACTER s     ASCII 73
0482 035A 7880          DATA >7880
0483 035C 7008          DATA >7008
0484 035E F000          DATA >F000
0485 0360 0000          DATA >0000          CHARACTER t     ASCII 74
0486 0362 F820          DATA >F820
0487 0364 2020          DATA >2020
0488 0366 2000          DATA >2000
0489 0368 0000          DATA >0000          CHARACTER u     ASCII 75
0490 036A 8888          DATA >8888
0491 036C 8888          DATA >8888
```

```
 0492 036E 7000          DATA >7000
 0493 0370 0000          DATA >0000          CHARACTER v     ASCII 76
 0494 0372 8888          DATA >8888
 0495 0374 90A0          DATA >90A0
 0496 0376 4000          DATA >4000
 0497 0378 0000          DATA >0000          CHARACTER w     ASCII 77
 0498 037A 8888          DATA >8888
 0499 037C A8D8          DATA >A8D8
 0500 037E 8800          DATA >8800
 0501 0380 0000          DATA >0000          CHARACTER x     ASCII 78
 0502 0382 8860          DATA >8860
 0503 0384 2060          DATA >2060
 0504 0386 8800          DATA >8800
 0505 0388 0000          DATA >0000          CHARACTER y     ASCII 79
 0506 038A 8850          DATA >8850
 0507 038C 2020          DATA >2020
 0508 038E 2000          DATA >2000
 0509 0390 0000          DATA >0000          CHARACTER z     ASCII 7A
 0510 0392 F810          DATA >F810
 0511 0394 2040          DATA >2040
 0512 0396 F800          DATA >F800
 0513 0398 3840          DATA >3840          CHARACTER       ASCII 7B
 0514 039A 20C0          DATA >20C0
 0515 039C 2040          DATA >2040
 0516 039E 3800          DATA >3800
 0517 03A0 4020          DATA >4020          CHARACTER       ASCII 7C
 0518 03A2 1008          DATA >1008
 0519 03A4 1020          DATA >1020
 0520 03A6 4000          DATA >4000
 0521 03A8 E010          DATA >E010          CHARACTER       ASCII 7D
 0522 03AA 2018          DATA >2018
 0523 03AC 2010          DATA >2010
 0524 03AE E000          DATA >E000
 0525 03B0 40A8          DATA >40A8          CHARACTER       ASCII 7E
 0526 03B2 1000          DATA >1000
 0527 03B4 0000          DATA >0000
 0528 03B6 0000          DATA >0000
 0529 03B8 A850          DATA >A850          CHARACTER       ASCII 7F
 0530 03BA A850          DATA >A850
 0531 03BC A850          DATA >A850
 0532 03BE A800          DATA >A800
 0533                    END
NO ERRORS,     NO WARNINGS
```

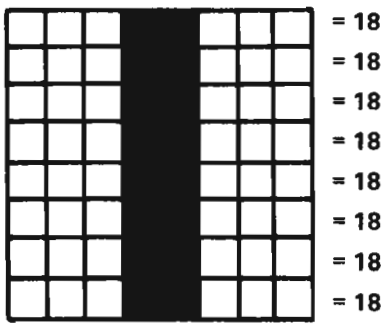
## 4.5 TMS9900 SOFTWARE SUBROUTINES

NOTE: Before using any of the line drawing subroutines, the "Load Line Drawing Patterns" subroutine must be executed.

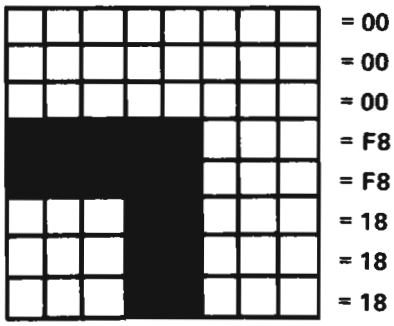
PATTERN 00



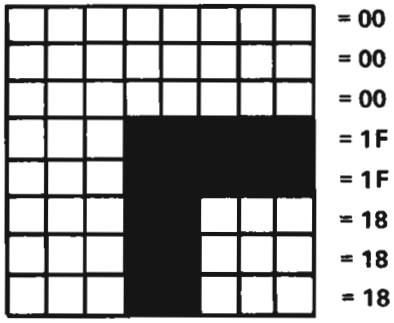

PATTERN 01



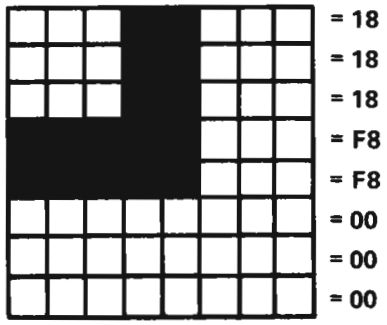

PATTERN 02



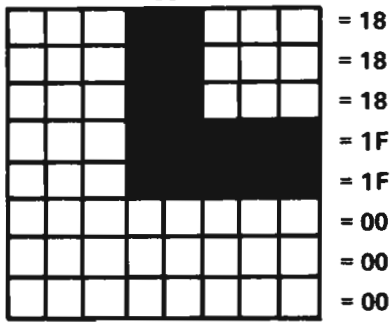

PATTERN 03



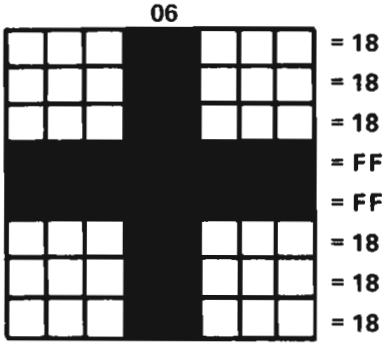

PATTERN 04




PATTERN 05




PATTERN 06



```
0001                    IDT 'SEGMENTS'
0002                ************************************************************
0003                *                                                          *
0004                *                    TMS9918A SUBROUTINES                  *
0005                *                                                          *
0006                ************************************************************
0007        9000    VRAMW   EQU  >9000 ADDRESS TO WRITE DATA TO VRAM
0008        9002    VDPW    EQU  >9002 ADDRESS TO WRITE DATA TO VDP
0009        9004    VRAMR   EQU  >9004 ADDRESS TO READ DATA FROM VRAM
0010        9006    VDPR    EQU  >9006 ADDRESS TO READ STATUS FROM VDP
0011                ************************************************************
0012                *                                                          *
0013                *                 LOAD LINE DRAWING PATTERNS               *
0014                *                                                          *
0015                *          REGISTERS USED:                                 *
0016                *                                                          *
0017                *          REG 1 = RESERVED                                *
0018                *          REG 2 = RESERVED                                *
0019                *          REG 3 = ADDRESS OF PATTERN GENERATOR            *
0020                *          REG 4 = PATTERN LOCATION IN MEMORY              *
0021                *          REG 5 = COLORS OF DRAWING PATTERNS              *
0022                *                          (USER DEFINED)                  *
0023                ************************************************************
0024                *
0025                *
0026 0000 0201          LI   R1,VRAMW          ADDRESS TO WRITE DATA TO VRAM
     0002 9000
0027 0004 0202          LI   R2,VDPW           ADDRESS TO WRITE TO VDP
     0006 9002
0028 0008 0203          LI   R3,>4800          ADDRESS IN PATT GEN FOR PATT 0
     000A 4800
0029 000C 0204          LI   R4,PATD           DRAWING PATTERNS LOCATED IN MEM
     000E 002E'
0030 0010 C483          MOV  R3,*R2            SEND MSB OF VRAM ADDRESS TO VDP
0031 0012 06C3          SWPB R3                REVERSE BYTES
0032 0014 C483          MOV  R3,*R2            SEND LSB OF VRAM ADDRESS TO VDP
0033 0016 0203          LI   R3,56             7 PATTERNS  X 8 BYTES EACH
     0018 0038
0034 001A D474  LLD1    MOVB *R4+,*R1          SEND BYTE TO VRAM
0035 001C 0603          DEC  R3                DECREMENT BYTE COUNT
0036 001E 16FD          JNE  LLD1              IF NOT DONE, GET NEXT BYTE
0037 0020 0203          LI   R3,>4200          ADDRESS OF COLOR TABLE
     0022 4200
0038 0024 C483          MOV  R3,*R2            SEND LSB OF VRAM ADDRESS TO VDP
0039 0026 06C3          SWPB R3                REVERSE BYTES
0040 0028 C483          MOV  R3,*R2            SEND MSB OF VRAM ADDRESS TO VDP
0041 002A C445          MOV  R5,*R1            SEND COLOR BYTE TO VRAM
0042 002C 045B          B    *R11              RETURN TO CALLING PROGRAM
0043                *
0044                *
0045                *     PATTERNS FOR LINE DRAWING
0046                *
0047                *
0048 002E 0000  PATD    DATA >0000        PATTERN  00
0049 0030 00FF          DATA >00FF
0050 0032 FF00          DATA >FF00
0051 0034 0000          DATA >0000
0052 0036 1818          DATA >1818        PATTERN  01
0053 0038 1818          DATA >1818
0054 003A 1818          DATA >1818
```

```
0055 003C 1818          DATA >1818
0056 003E 0000          DATA >0000        PATTERN  02
0057 0040 00F8          DATA >00F8
0058 0042 F818          DATA >F818
0059 0044 1818          DATA >1818
0060 0046 0000          DATA >0000        PATTERN  03
0061 0048 001F          DATA >001F
0062 004A 1F18          DATA >1F18
0063 004C 1818          DATA >1818
0064 004E 1818          DATA >1818        PATTERN  04
0065 0050 18F8          DATA >18F8
0066 0052 F800          DATA >F800
0067 0054 0000          DATA >00C0
0068 0056 1818          DATA >1818        PATTERN  05
0069 0058 181F          DATA >181F
0070 005A 1F00          DATA >1F00
0071 005C 0000          DATA >0000
0072 005E 1818          DATA >1818        PATTERN  06
0073 0060 18FF          DATA >18FF
0074 0062 FF18          DATA >FF18
0075 0064 1818          DATA >1818
0076              *
0077              *
0078              ************************************************************
0079              *                                                          *
0080              *                LOAD SPRITES SUBROUTINE                   *
0081              *                                                          *
0082              *      REGISTERS USED:                                     *
0083              *                                                          *
0084              *      R1 = RESERVED                                       *
0085              *      R2 = RESERVED                                       *
0086              *      R3 = ADDRESS OF SPRITE TABLE IN VRAM                *
0087              *      R4 = MEMORY ADDRESS OF SPRITE TABLE (USER DEF)      *
0088              *      R5 = NUMBER OF BYTES TO TRANSFER (USER DEF)         *
0089              *                                                          *
0090              ************************************************************
0091              *
0092 0066 0201          LI   R1,VRAMW         ADDRESS TO WRITE DATA TO VRAM
     0068 8C00
0093 006A 0202          LI   R2,VDPW          ADDRESS TO WRITE TO VDP
     006C 8C02
0094 006E 0203          LI   R3,>4000         ADDRESS OF SPRITE TABLE IN VRAM
     0070 4000
0095 0072 C483          MOV  R3,*R2           SEND LSB OF VRAM ADDRESS TO VDP
0096 0074 06C3          SWPB R3               REVERSE BYTES
0097 0076 C483          MOV  R3,*R2           SEND MSB OF VRAM ADDRESS TO VDP
0098 0078 D0F4  LDPL    MOVB *R4+,R3          GET BYTE OF DATA FROM MEM
0099 007A 06C3          SWPB R3               REVERSE BYTES
0100 007C C443          MOV  R3,*R1           SEND DATA TO VRAM
0101 007E 0605          DEC  R5               ARE WE DONE YET?
0102 0080 16FB          JNE  LDPL             NO, GO AGAIN
0103 0082 045B          B    *R11             YES, RETURN TO CALLING PROGRAM
```

```
0105                ************************************************************
0106                *                                                          *
0107                *                CLEAR SCREEN SUBROUTINE                   *
0108                *                                                          *
0109                *     REGISTERS USED:                                      *
0110                *                                                          *
0111                *     R1 = RESERVED                                        *
0112                *     R2 = RESERVED                                        *
0113                *     R3 = START ADDRESS ON SCREEN                         *
0114                *                                                          *
0115                ************************************************************
0116                *
0117                *
0118 0084 0201     CLSC   LI   R1,VRAMW         ADDRESS TO WRITE DATA TO VRAM
     0086 9000
0119 0088 0202            LI   R2,VDPW          ADDRESS TO WRITE TO VDP
     008A 9002
0120 008C 0203            LI   R3,>4400         START LOCATION OF THE NAME TABLE
     008E 4400
0121 0090 C483            MOV  R3,*R2           SEND LSB OF VRAM ADDRESS TO VDP
0122 0092 06C3            SWPB R3               REVERSE BYTES
0123 0094 C483            MOV  R3,*R2           SEND MSB OF VRAM ADDRESS TO VDP
0124 0096 0202            LI   R2,768           #OF POSITIONS ON SCREEN
     0098 0300
0125 009A 0203            LI   R3,>20           ASCII SPACE CHAR
     009C 0020
0126 009E C443     CSL1   MOV  R3,*R1           SEND SPACE CHAR TO VRAM
0127 00A0 0602            DEC  R2               ARE ALL LOCATIONS CLEAR?
0128 00A2 16FD            JNE  CSL1             NO, GO AGAIN
0129 00A4 045B            B    *R11             YES, RETURN TO CALLING PROGRAM
0130                *
0131                *
0132                ************************************************************
0133                *                                                          *
0134                *                 PRINT MESSAGE SUBROUTINE                 *
0135                *                AND BRANCH TO USERS PROGRAM               *
0136                *                                                          *
0137                *          REGISTERS USED:                                 *
0138                *                                                          *
0139                *          R1 = RESERVED                                   *
0140                *          R2 = RESERVED                                   *
0141                *          R3 = STARTING ADDRESS OF MESSAGE IN NAME        *
0142                *               TABLE (USER DEFINED)                       *
0143                *          R4 = MEMORY ADDRESS  OF MESSAGE (USER DEFINED)  *
0144                *                                                          *
0145                *          NOTE:  END MESSAGE STRING WITH A BYTE 00        *
0146                *                                                          *
0147                ************************************************************
0148                *
0149                *
0150 00A6 0201     PRNT   LI   R1,VRAMW         ADDRESS TO WRITE DATA TO VRAM
     00A8 9000
0151 00AA 0202            LI   R2,VDPW          ADDRESS TO WRITE TO VDP
     00AC 9002
0152 00AE C483            MOV  R3,*R2           SEND LSB OF VRAM ADDRESS TO VDP
0153 00B0 06C3            SWPB R3               REVERSE BYTES
0154 00B2 C483            MOV  R3,*R2           SEND MSB OF VRAM ADDRESS TO VDP
0155 00B4 D0B4     PRL1   MOVB *R4+,R2          GET BYTE OF TEXT FROM MEM
0156 00B6 1303            JEQ  PRL2             IF ZERO, THEN END OF MESS
0157 00B8 06C2            SWPB R2               INDEX BYTE INTO POSITION
```

```
0158 00BA C442          MOV  R2,*R1          SEND CHAR TO VRAM
0159 00BC 10FB          JMP  PRL1            GET NEXT CHAR
0160 00BE 045B  PRL2    B    *R11            RETURN TO CALLING PROGRAM
0161                    *
0162                    *
0163                    ************************************************************
0164                    *                                                          *
0165                    *            ERASE TO END OF SCREEN SUBROUTINE             *
0166                    *                                                          *
0167                    *      REGISTERS USED:                                     *
0168                    *                                                          *
0169                    *      R1 = RESERVED                                       *
0170                    *      R2 = RESERVED                                       *
0171                    *      R3 = ADDRESS IN NAME TABLE TO START ERASURE         *
0172                    *           (USER DEFINED), R3 MUST BE EQUAL  TO           *
0173                    *           OR GREATER THAN >4400, AND MUST BE LESS        *
0174                    *           THAN OR EQUAL TO >46FF                         *
0175                    *                                                          *
0176                    ************************************************************
0177                    *
0178 00C0 0201  EEOS    LI   R1,VRAMW        ADDRESS TO WRITE DATA TO VRAM
     00C2 9000
0179 00C4 0202          LI   R2,VDPW         ADDRESS TO WRITE TO VDP
     00C6 9002
0180 00C8 C483          MOV  R3,*R2          SEND LSB OF VRAM ADDRESS TO VDP
0181 00CA 06C3          SWPB R3              REVERSE BYTES
0182 00CC C483          MOV  R3,*R2          SEND MSB OF VRAM ADDRESS TO VDP
0183 00CE 0202          LI   R2,>20          LOAD R2 WITH 'SPACE' CHAR
     00D0 0020
0184 00D2 C442  EES1    MOV  R2,*R1          SEND 'SPACE' TO SCREEN
0185 00D4 0583          INC  R3              INCREMENT CHAR COUNT
0186 00D6 0283          CI   R3,>4700        ARE WE AT THE END OF SCREEN
     00D8 4700
0187 00DA 1AFB          JL   EES1            IF NOT GO AGAIN
0188 00DC 045B          B    *R11            YES, RETURN TO CALLING PROGRAM
0189                    *
0190                    *
0191                    ************************************************************
0192                    *                                                          *
0193                    *                    ERASE LINE SUB                        *
0194                    *                                                          *
0195                    *      REGISTERS USED:                                     *
0196                    *                                                          *
0197                    *      R1 = RESERVED                                       *
0198                    *      R2 = RESERVED                                       *
0199                    *      R3 = STARTING ADDRESS IN LINE IN NAME               *
0200                    *           TABLE TO BE ERASED (USER DEFINED)              *
0201                    ************************************************************
0202                    *
0203                    *
0204 00DE 0201  ERLN    LI   R1,VRAMW        ADDRESS TO WRITE DATA TO VRAM
     00E0 9000
0205 00E2 0202          LI   R2,VDPW         ADDRESS TO WRITE TO VDP
     00E4 9002
0206 00E6 C483          MOV  R3,*R2          SEND LSB OF VRAM ADDRESS TO VDP
0207 00E8 06C3          SWPB R3              REVERSE BYTES
0208 00EA C483          MOV  R3,*R2          SEND MSB OF VRAM ADDRESS TO VDP
0209 00EC 0202          LI   R2,>20          LOAD R2 WITH 'SPACE' CHAR
     00EE 0020
0210 00F0 0203          LI   R3,32           LOAD R3 WITH # OF POSTIONS
```

```
     00F2 0020
0211 00F4 C442  ERL1   MOV  R2,*R1          SEND 'SPACE' CAHR TO NAME TABLE
0212 00F6 0603         DEC  R3              DECREMENT CHAR COUNT
0213 00F8 16FD         JNE  ERL1            IF NOT DONE, GO AGAIN
0214 00FA 045B         B    *R11            DONE, RETURN TO CALLING PROG
0215              *
0216              *
0217              ************************************************************
0218              *                                                          *
0219              *                    DRAW A HORIZ LINE                     *
0220              *                                                          *
0221              *      REGISTERS USED:                                     *
0222              *                                                          *
0223              *      R3 = ADDRESS OF UPPER LEFT CORNER                   *
0224              *      R4 = # OF HORIZ POSITIONS                           *
0225              *      R5 = # OF VERT POSITIONS                            *
0226              *      R9 = PATTERN # OFFSET                               *
0227              *                                                          *
0228              ************************************************************
0229              *
0230              *
0231 00FC 0201  DBOX   LI   R1,>9000        ADDRESS OF DATA TO 9918
     00FE 9000
0232 0100 0202         LI   R2,>9002        ADDRESS OF ADDRESSES TO 9918
     0102 9002
0233 0104 C483         MOV  R3,*R2          SEND LSB OF ADDRESS TO 9918
0234 0106 06C3         SWPB R3              REVERSE BYTES
0235 0108 C483         MOV  R3,*R2          SEND MSB OF ADDRESS TO 9918
0236 010A 06C3         SWPB R3              REVERSE BYTES
0237 010C C189         MOV  R9,R6           GET OFFSET
0238 010E 0226         AI   R6,>01          POINT TO UPPER LEFT CORNER PATTE
     0110 0001
0239 0112 C446         MOV  R6,*R1          SEND IT TO THE 9918
0240 0114 C1C4         MOV  R4,R7           STORE HORIZ COUNT IN TEMP REG
0241 0116 0647         DECT R7              DETERMINE (LENGTH - CORNERS)
0242 0118 1304         JEQ  DBL2            NO LENGTH OTHER THAN CORNERS
0243 011A C189         MOV  R9,R6           PATTERN 00 + OFFSET
0244 011C C446  DBL1   MOV  R6,*R1          SEND LINE SEGMENT TO 9918
0245 011E 0607         DEC  R7              DEC LINE COUNT
0246 0120 16FD         JNE  DBL1            NOT DONE YET, GO AGAIN
0247 0122 C189  DBL2   MOV  R9,R6           GET OFFSET
0248 0124 0226         AI   R6,>05          POINT TO UPPER RIGHT CORNER
     0126 0005
0249 0128 C446         MOV  R6,*R1          SEND IT TO 9918
0250 012A C203         MOV  R3,R8           SEND ADDRESS OF UPPER LEFT TO TE
0251 012C C1C5         MOV  R5,R7           SEND VERT COUNT TO TEMP
0252 012E 0647         DECT R7              DETERMINE (HEIGTH - CORNERS)
0253 0130 130C         JEQ  DBL4            NO HEIGTH OTHER THAN CORNERS
0254 0132 C189         MOV  R9,R6           GET OFFSET
0255 0134 0226         AI   R6,>03          POINT TO VERT LINE PATTERN
     0136 0003
0256 0138 0228  DBL3   AI   R8,>20          INC VERT POSITION BY 1 CHAR
     013A 0020
0257 013C C488         MOV  R8,*R2          SEND ADDRESS TO 9918
0258 013E 06C8         SWPB R8              REVERSE BYTES
0259 0140 C488         MOV  R8,*R2          SEND ADDRESS TO 9918
0260 0142 06C8         SWPB R8              REVERSE BYTES
0261 0144 C046         MOV  R6,R1           SEND VERT LINE SEGMENT TO 9918
0262 0146 0607         DEC  R7              DECREMENT VERT COUNT
0263 0148 16F7         JNE  DBL3            NOT DONE YET, GO AGAIN
```

```
0264 014A 0228  DBL4   AI   R8,>20          INC VERT POSITION BY 1 CHAR
     014C 0020
0265 014E C488         MOV  R8,*R2          SEND ADDRESS TO 9918A
0266 0150 06C8         SWPB R8              REVERSE BYTES
0267 0152 C088         MOV  R8,R2           SEND ADDRESS TO 9918
0268 0154 06C8         SWPB R8              REVERSE BYTES
0269 0156 C189         MOV  R9,R6           GET OFFSET
0270 0158 0226         AI   R6,>02          POINT TO LOWER LEFT CORNER PATT
     015A 0002
0271 015C C446         MOV  R6,*R1          SEND IT TO 9918
0272 015E C1C4         MOV  R4,R7           SEND HORIZ COUNT TO TEMP
0273 0160 0647         DECT R7              DETERMINE (HORIZ - CORNERS)
0274 0162 1304         JEQ  DBL6            NO HORIZ OTHER THAN CORNERS
0275 0164 C189         MOV  R9,R6           GET OFFSET
0276 0166 C446  DBL5   MOV  R6,*R1          SEND HORIZ PATTERN TO 9918
0277 0168 0607         DEC  R7              DECREMENT HORIZ COUNT
0278 016A 16FD         JNE  DBL5            IF NOT DONE, GO AGAIN
0279 016C C189  DBL6   MOV  R9,R6           GET OFFSET
0280 016E 0226         AI   R6,>04          POINT TO LOWER RIGHT PATTERN
     0170 0004
0281 0172 C446         MOV  R6,*R1          SEND PATTERN TO 9918
0282 0174 C1C5         MOV  R5,R7           STORE VERT COUNT IN TEMP
0283 0176 0647         DECT R7              DECREMENT VERT COUNT
0284 0178 1312         JEQ  DBL8            IF NO VERT SEGMENTS, DONE
0285 017A C1C4         MOV  R4,R7           STORE HORIZ COUNT IN TEMP
0286 017C 0607         DEC  R7              DECREMENT HORIZ COUNT
0287 017E C203         MOV  R3,R8           STORE ADDRESS OF UPPER LEFT COR
0288 0180 A207         A    R7,R8           FIND UPPER RIGHT CORNER LOCATION
0289 0182 C189         MOV  R9,R6           GET OFFSET
0290 0184 0226         AI   R6,>03          POINT TO VERTICAL LINE PATTERN
     0186 0003
0291 0188 C1C5         MOV  R5,R7           STORE VERT COUNT IN TEMP
0292 018A 0647         DECT R7              DETERMINE (HEIGHT - CORNERS)
0293 018C 0228  DBL7   AI   R8,>20          INCREMENT VERT POSITION BY 1 CHA
     018E 0020
0294 0190 C488         MOV  R8,*R2          SEND ADDRESS TO 9918
0295 0192 06C8         SWPB R8              REVERSE BYTES
0296 0194 C488         MOV  R8,*R2          SEND ADDRESS TO 9918
0297 0196 06C8         SWPB R8              REVERSE BYTES
0298 0198 C446         MOV  R6,*R1          SEND PATTERN TO 9918
0299 019A 0607         DEC  R7              DECREMENT VERT COUNT
0300 019C 16F7         JNE  DBL7            IF NOT DONE, GO AGAIN
0301 019E 045B  DBL8   B    *R11            SUB DONE RETURN TO CALLING PROG
0302                   *
0303                   *
0304                   ************************************************************
0305                   *                                                          *
0306                   *              LOAD TEXT COLORS SUBROUTINE                 *
0307                   *                                                          *
0308                   *       REGISTERS USED:                                    *
0309                   *                                                          *
0310                   *       R4 = COLORS FOR TEXT CHARACTERS                    *
0311                   *                                                          *
0312                   ************************************************************
0313                   *
0314 01A0 0201  LDTC   LI   R1,>9000
     01A2 9000
0315 01A4 0202         LI   R2,>9002
     01A6 9002
0316 01A8 0203         LI   R3,>4204        ADDRESS FOR TEXT COLORS IN 9918
```

```
      01AA 4204
 0317 01AC C483          MOV  R3,*R2          SEND ADDRESS TO 9918
 0318 01AE 06C3          SWPB R3              REVERSE BYTES
 0319 01B0 C483          MOV  R3,*R2          8 COLOR CHAR X 8 TEXT/CHAR = 64
 0320 01B2 C444  LCLI    MOV  R4,*R1          SEND WORD TO 9918
 0321 01B4 0602          DEC  R2              DECREMENT COUNT
 0322 01B6 16FD          JNE  LCLI            IF NOT DONE, GO AGAIN
 0323 01B8 045B          B    *R11            DONE, RETURN TO CALLING PROG
 0324              *
 0325              *
 0326              ************************************************************
 0327              *                                                          *
 0328              *                 DRAW A VERTICAL LINE SUB                 *
 0329              *                                                          *
 0330              *      REGISTERS USED:                                     *
 0331              *                                                          *
 0332              *      R3 = ADDRESS ON SCREEN                              *
 0333              *      R4 = # OF POSITIONS                                 *
 0334              *      R9 = PATTERN OFFSET                                 *
 0335              *                                                          *
 0336              ************************************************************
 0337              *
 0338 01BA C483  DVLN    MOV  R3,*R2          SEND ADDRESS TO 9918
 0339 01BC 06C3          SWPB R3              REVERSE BYTES
 0340 01BE C483          MOV  R3,*R2          SEND ADDRESS TO 9918
 0341 01C0 06C3          SWPB R3              REVERSE BYTES
 0342 01C2 C445          MOV  R5,*R1          SEND PATTERN TO 9918
 0343                    END
NO ERRORS,     NO WARNINGS
```

# 5. TMS9918A/9928A/9929A ELECTRICAL SPECIFICATIONS

## 5.1 ABSOLUTE MAXIMUM RATINGS OVER OPERATING FREE-AIR TEMPERATURE RANGE (unless otherwise noted)*

Supply voltage, $V_{CC}$ ..... −0.3 to 20 V
All input voltages ..... −0.3 to 20 V
Output voltage ..... −2 to 7 V
Continuous power dissipation ..... 1.3 W
Operating free-air temperature range ..... 0°C to 70°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 is not implied. Exposure to absolute maximum rated conditions for extended periods may affect device reliability.

## 5.2 RECOMMENDED OPERATING CONDITIONS*

| PARAMETER | | MIN | NOM | MAX | UNIT |
|---|---|---|---|---|---|
| Supply voltage, $V_{CC}$ | | 4.75 | | 5.25 | V |
| Supply voltage, $V_{SS}$ | | | 0 | | V |
| Input Voltage, $V_I$, $\overline{RESET}$/SYNC pin | SYNC active | 10 | | 12 | V |
| | RESET active | | | 0.6 | V |
| | SYNC and RESET inactive | 3 | | 6 | V |
| High-level input, $V_{IH}$ | XTAL1, XTAL2 | 2.75 | | | V |
| | All other inputs | 2.2 | | | V |
| Input voltage, $V_I$, EXT VDP pin (TMS9918A only) | SYNC level | | 2.6 | | V |
| | White level | | 3.7 | | V |
| | Black level | | 3 | | V |
| Low-level input voltage, $V_{IL}$ | | | | 0.8 | V |
| Operating free-air temperature, $T_A$ | | 0 | | 70 | °C |

* All voltage values are with respect to $V_{SS}$.

## 5.3 ELECTRICAL CHARACTERISTICS OVER FULL RANGES OF RECOMMENDED OPERATING CONDITIONS (unless otherwise noted)

**TMS9918A/9928A/9929A**

| PARAMETER | | | TEST CONDITIONS | MIN | TYP† | MAX | UNIT |
|---|---|---|---|---|---|---|---|
| $V_{OH}$ | High-level output voltage | $\overline{RAS}$, $\overline{CAS}$, R/$\overline{W}$ | $I_{OH}$ = 400 μA | 2.7 | 3.4 | | V |
| | | All other outputs | | 2.4 | 3.2 | | |
| $V_{OL}$ | Low-level output voltage | CPU data | $I_{OL}$ = 1.2 mA | | 0.3 | 0.6 | V |
| | | DRAM interface | $I_{OL}$ = 800 μA | | | 0.6 | |
| $I_{OZH}$ | Off-state output current high-level voltage applied, D0-D7 outputs | | $V_O$ = 5.25 V | | 1 | 100 | μA |
| $I_{OZL}$ | Off-state output current high-level voltage applied, D0-D7 outputs | | $V_O$ = 0.4 V | | 1 | -100 | μA |
| $I_{IH}$ | High-level input current | | $V_I$ = 5.25 V, all other pins at 0 V | | | 10 | μA |
| $I_{IL}$ | Low-level input current | | $V_I$ = 0 V, All other pins at 0 V | | | -10 | μA |

**TMS9918A Only (Figure 5-1)**

| PARAMETER | | TEST CONDITIONS | MIN | NOM | MAX | UNIT |
|---|---|---|---|---|---|---|
| $V_{white}$ | Video voltage level of white, COMVID | $R_L$ = 470 Ω | 2.8 | 3.0 | 3.2 V | |
| $V_{black}$ | Video voltage level of black (blank), COMVID | | 2.1 | 2.3 | 2.5 V | |
| $V_{sync}$ | Video voltage level of sync, COMVID | | 1.85 | 2.0 | 2.1 V | |

† All typical values are at $V_{CC}$ = 5.25 V, $T_A$ = 25°C.

## 5.3 ELECTRICAL CHARACTERISTICS OVER FULL RANGES OF RECOMMENDED OPERATING CONDITIONS (unless otherwise noted) (Continued)

**TMS9928A/9929A Only (Figure 5-1)**

| PARAMETER | | TEST CONDITIONS | MIN | NOM | MAX | UNIT |
|---|---|---|---|---|---|---|
| $V_{white}$ | Video voltage level of white, Y, R-Y, B-Y outputs | $R_L$ = 470 Ω | 2.5 | 3 | 3.6 | V |
| $V_{black}$ | Video voltage level of black (blank), Y, R-Y, B-Y outputs | | 1.6 | 2.3 | 2.5 | V |
| $V_{sync}$ | Video voltage level of sync, Y output | | 1.2 | 1.8 | 2 | V |

**TMS9929A Only**

| PARAMETER | | TEST CONDITIONS | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|
| $V_{PS}$ | Color burst video voltage level with respect to V no color | R-Y output | | 0.25 | | V |
| $V_{neg}$ | Color burst video voltage level with respect to V no color | B-Y output | | -0.25 | | V |

**TMS9918A/9928A/9929A (Figure 5-2)**

| PARAMETER | | | TEST CONDITIONS | MIN | NOM | MAX | UNIT |
|---|---|---|---|---|---|---|---|
| | Video voltage difference, white-black, Y, R-Y, B-Y outputs | | | 0.7 | 1.0 | | V |
| $I_{CC}$ | Average supply current from $V_{CC}$ | | $T_A$ = 25°C | | 200 | 250 | mA |
| $C_i$ | Input capacitance | D0-D7 | unmeasured f = 11 MHz, pins at 0 V | | | 20 | pF |
| | | All other inputs | | | | 10 | |
| | | | | | | 10 | |
| $C_o$ | Output capacitance | | unmeasured f = 11 MHz, pins at 0 V | | | 20 | pF |

† All typical values are at $V_{CC}$ = 5.25 V, $T_A$ = 25°C.

## 5.4 TIMING REQUIREMENTS OVER FULL RANGES OF RECOMMENDED OPERATING CONDITIONS (TMS9918A/9928A/9929A)

**CPU - VDP Interface (Figures 5-3 and 5-4)**

| PARAMETER | | MIN | NOM | MAX | UNIT |
|---|---|---|---|---|---|
| $t_{su(A-RL)}$ | Address setup time before CSR low | | 0 | | ns |
| $t_{su(A-WL)}$ | Address setup time before CSW low | | 30 | | ns |
| $t_{h(WL-A)}$ | Address hold time after CSW low | | 30 | | ns |
| $t_{su(D-WH)}$ | Data setup time before CSW high | | 100 | | ns |
| $t_{h(WH-D)}$ | Data hold time after CSW high | | 30 | | ns |
| $t_{w(WL)}$ | Pulse width, CSW low | | 200 | | ns |
| $t_{w(CS-H1)}$ | Pulse width, chip select high (requesting memory access) | | 8 | | μs |
| $t_{w(CS-H2)}$ | Pulse width, chip select high (not requesting memory access) | | 2 | | μs |

**VDP-VRAM Interface (Figure 5-5 and 5-6)**

| PARAMETER | | MIN | NOM | MAX | UNIT |
|---|---|---|---|---|---|
| $t_c$ | Memory read or write cycle time | 372 | | | ns |
| $t_{su(D-CH)}$ | Input data setup time before CAS high | 60 | | | ns |
| $t_{h(CH-D)}$ | Input data hold time after CAS high | 0 | | | ns |

**External Clock Source (Figure 5-7)**

| PARAMETER | | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|
| $f_{ext}$ | External source frequency | 10.738098 | 10.738635 | 10.739172 | MHz |
| $t_r/t_f$ | External source rise/fall time | | 10 | 15 | ns |
| $t_{wH}$ | External source high-level pulse width | 42 | 47 | 52 | ns |
| $t_{wL}$ | External source low-level pulse width | 42 | 47 | 52 | ns |
| $t_{pD}$ | External source phase delay from XTAL1 falling edge to XTAL2 falling edge | 42 | 47 | 52 | ns |

## 5.5 SWITCHING CHARACTERISTICS OVER FULL RANGE OF RECOMMENDED OPERATING CONDITIONS (TMS9918A/9928A/9929A)

### CPU-VDP Interface

| PARAMETER | | TEST CONDITIONS | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|
| $T_{A(CSR)}$ | Data access time from CSR low | $C_L$ = 300 pF | | 100 | 150 | ns |
| $t_{PVX}$ | Data disable time after CSR high | | | 65 | 100 | ns |
| $t_{PVX,A}$ | Data invalid time from address changes | | | 0 | | ns |
| $f_{CPUCLK}$ | CPU clock output clock frequency ($f_{ext}$ ÷ 3) | | 3.4 | 3.58 | 3.76 | MHz |
| $f_{GROMCLK}$ | GROM clock output clock frequency ($f_{ext}$ ÷ 24) | | 425.12 | 447.5 | 469.88 | kHz |

### VDP-VRAM Interface (Figures 5-5 and 5-6)

| PARAMETER | | TEST CONDITIONS | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|
| $t_w$ | Pulse width, $\overline{CAS}$ high | $C_L$ = 50 pF | 80 | 100 | 120 | ns |
| $t_{w(CL)}$ | Pulse width, $\overline{CAS}$ low | | 220 | 230 | 250 | ns |
| $t_{w(RH)}$ | Pulse width, $\overline{RAS}$ high | | 100 | 125 | 150 | ns |
| $t_{w(RL)}$ | Pulse width, $\overline{RAS}$ low | | 190 | 210 | 230 | ns |
| $t_{w(W)}$ | Pulse width, write pulse | | 170 | 190 | 210 | ns |
| $t_{CA-CL}$ | Delay time, column address to $\overline{CAS}$ low | | -10 | -2 | | ns |
| $t_{RA-RL}$ | Delay time, row address to $\overline{RAS}$ low | | 25 | 45 | 65 | ns |
| $t_{d-WL}$ | Delay time, data to R/$\overline{W}$ low | | 0 | 6 | 20 | ns |
| $t_{WH-CL}$ | Delay time, R/$\overline{W}$ high to $\overline{CAS}$ low | | 25 | 50 | 75 | ns |
| $t_{W-CH}$ | Delay time, R/$\overline{W}$ low to $\overline{CAS}$ high | | 120 | 140 | 160 | ns |
| $t_{W-RH}$ | Delay time, R/$\overline{W}$ low to $\overline{RAS}$ high | | 60 | 75 | 90 | ns |

## 5.5 SWITCHING CHARACTERISTICS OVER FULL RANGE OF RECOMMENDED OPERATING CONDITIONS (TMS9918A/9928A/9929A) (Continued)

**TMS9918A Composite video output (Figures 5-8 and 5-9)**

| PARAMETER | | TEST CONDITIONS | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|
| $t_{CL-CA}$ | Column address valid after $\overline{CAS}$ low | $C_L$ = 50 pF | 45 | 65 | 85 | ns |
| $t_{RL-RA}$ | Row address valid after $\overline{RAS}$ low | | 20 | 25 | 30 | ns |
| $t_{RL-CA}$ | Column address valid after $\overline{RAS}$ low | | 95 | 110 | 130 | ns |
| $t_{CL-D}$ | Data valid after $\overline{CAS}$ low | | 240 | 260 | 280 | ns |
| $t_{RL-D}$ | Data valid after $\overline{RAS}$ low | | 95 | 110 | 125 | ns |
| $t_{WL-D}$ | Data valid after R/$\overline{W}$ low | | 135 | 165 | 195 | ns |
| $t_{CH-WL}$ | Read command valid after $\overline{CAS}$ high | | 0 | | | ns |
| $t_{CL-W}$ | Write command valid after $\overline{CAS}$ low | | 270 | 290 | 310 | ns |
| $t_{CH-RL}$ | Delay time, $\overline{CAS}$ high to $\overline{RAS}$ low | | 45 | 65 | | ns |
| $t_{CL-RH}$ | Delay time, $\overline{CAS}$ low to $\overline{RAS}$ high | | 150 | 170 | 190 | ns |
| $t_{RL-CL}$ | Delay time, $\overline{RAS}$ low to $\overline{CAS}$ low | | 30 | 40 | 50 | ns |

## 5.5 SWITCHING CHARACTERISTICS OVER FULL RANGE OF RECOMMENDED OPERATING CONDITIONS (TMS9918A/9928A/9929A) (Continued)

TMS9918A Composite video output (Figures 5-8 and 5-9)

| PARAMETER | | TEST CONDITIONS | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|
| $t_{fl}$ | Fall time, $V_{black}$ to $V_{sync}$ | $R_L$ = 470 Ω<br>$C_L$ = 150 pF | | 10 | | ns |
| $t_{w(HS)}$ | Pulse width, horizontal sync | | | 4.84 | | µs |
| $t_{rl}$ | Rise time, $V_{sync}$ to $V_{black}$ | | | 20 | | ns |
| $t_{HS-CD}$ | Delay time, sync to color burst | | | 372 | | ns |
| $t_{w(CB)}$ | Width, color burst | | | 261 | | µs |
| $t_{CB-LB}$ | Delay time, color burst to left border | | | 1.49 | | µs |
| $t_{r2}$ | Rise time, $V_{black}$ to $V_{white}$ | | | 60 | | ns |
| $t_{w(LB)}$ | Left border video width | | | 2.42 | | µs |
| $t_{f2}$ | Fall time, $V_{white}$ to $V_{black}$ | | | 110 | | ns |
| $t_{w(AD)}$ | Width of active display area | | | 47.68 | | µs |
| $t_{w(RB)}$ | Right border video width | | | 2.79 | | µs |
| $t_{RB-HS}$ | Delay time, right border to horizontal sync | | | 1.49 | | µs<br>µs |
| $t_{VFB}$ | Vertical front blanking | | | 191.1 | | µs |
| $t_{VS}$ | Vertical sync | | | 191.1 | | µs |
| $V_{VBB}$ | Vertical back blanking | | | 828 | | µs |
| $t_{ABA}$ | Active plus border area time | | | 18.8 | | ms |

NOTE: Fall times depend on external pull-down resistor

## 5.5 SWITCHING CHARACTERISTICS OVER FULL RANGE OF RECOMMENDED OPERATING CONDITIONS (TMS9918A/9928A/9929A) (Continued)

**TMS9928A/9929A Y, R-Y, B-Y outputs (Figures 5-10 through 5-13)**

| PARAMETER | | TEST CONDITIONS | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|---|
| $t_{f3}$ | Fall time, $V_{black}$ to $V_{sync}$ | $R_L$ = 470 Ω<br>$C_L$ = 15 pF | | 100 | | ns |
| $t_{w(HSI)}$ | Pulse width, horizontal sync | | | 4.84 | | µs |
| $t_{r3}$ | Rise time, $V_{sync}$ to $V_{black}$ | | | 150 | | ns |
| $t_{w(BP)}$ | Width, back porch | | | 4.47 | | µs |
| $t_{w(LBI)}$ | Width, left border | | | 2.8 | | µs |
| $t_{w(P)}$ | Pulse width, pixel | | | 186.24 | | ns |
| $t_{w(horz)}$ | Width, horizontal line | | | 63.695 | | µs |
| $t_{w(ADI)}$ | Width, active display area | | | 47.67 | | µs |
| $t_{r4}$ | Rise time, $V_{black}$ to $V_{white}$ | | | 75 | | ns |
| $t_{r4}$ | Fall time, $V_{white}$ to $V_{black}$ | | | 50 | | ns |
| $t_{w(RBI)}$ | Width, right border | | | 2.42 | | µs |
| $t_{w(FP)}$ | Width, front porch | | | 1.49 | | µs |
| $t_{r5}$ | Rise time, V no color to V pos CB | | | 150 | | ns |
| $t_{w(CB1)}$ | Pulse width, pos color burst | | | 2.6 | | µs |
| $t_{f5}$ | Fall time, V pos CB to V no color | | | 100 | | ns |
| $t_{w(CB-LBI)}$ | Delay time, pos CB to left border | | | 1.49 | | µs |
| $t_{f6}$ | Fall time, V no color to V neg CB | | | 100 | | ns |
| $t_{r6}$ | Rise time, V neg CB to V no color | | | 150 | | ns |
| $t_{w(VSI)}$ | Pulse width, vertical sync | | | 465 | | ns |
| $t_{VFBI}$ | Vertical front blanking | | | 191.09 | | µs |
| $t_{VSI}$ | Vertical sync | | | 191.09 | | µs |
| $t_{VBBI}$ | Vertical back blanking | | | 828.04 | | µs |
| $t_{ABAI}$ | Active area plus border area total | | | 18.70 | | mS |
| | Vertical time | | | 19.91 | | mS |

NOTE: Fall times depend on external pull-down resistor.

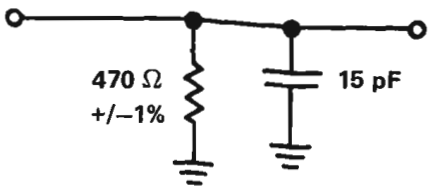



FIGURE 5-1 – LOAD CIRCUIT FOR COMVID (ALL DEVICES) AND R-Y, Y, B-Y SWITCHING CHARACTERISTICS (TMS9928A/9929A)

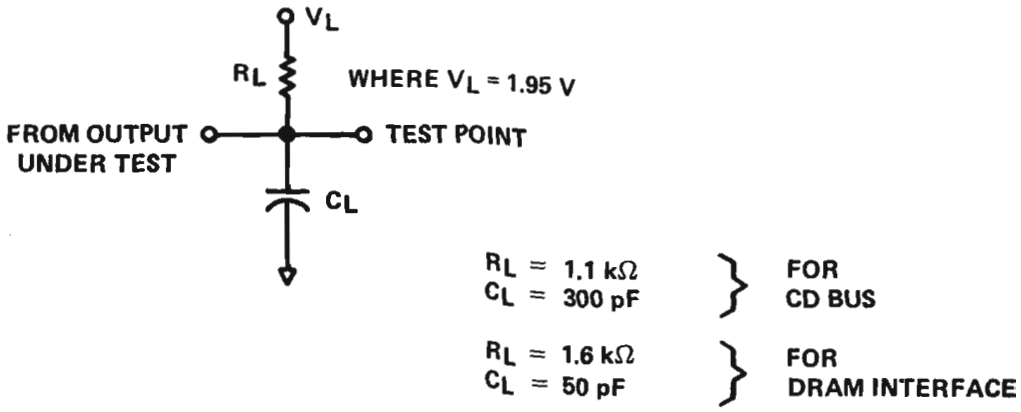



FIGURE 5-2 – LOAD CIRCUITS FOR ALL OUTPUTS EXCEPT COMVID, R-Y, Y, B-Y

WRITE CYCLE

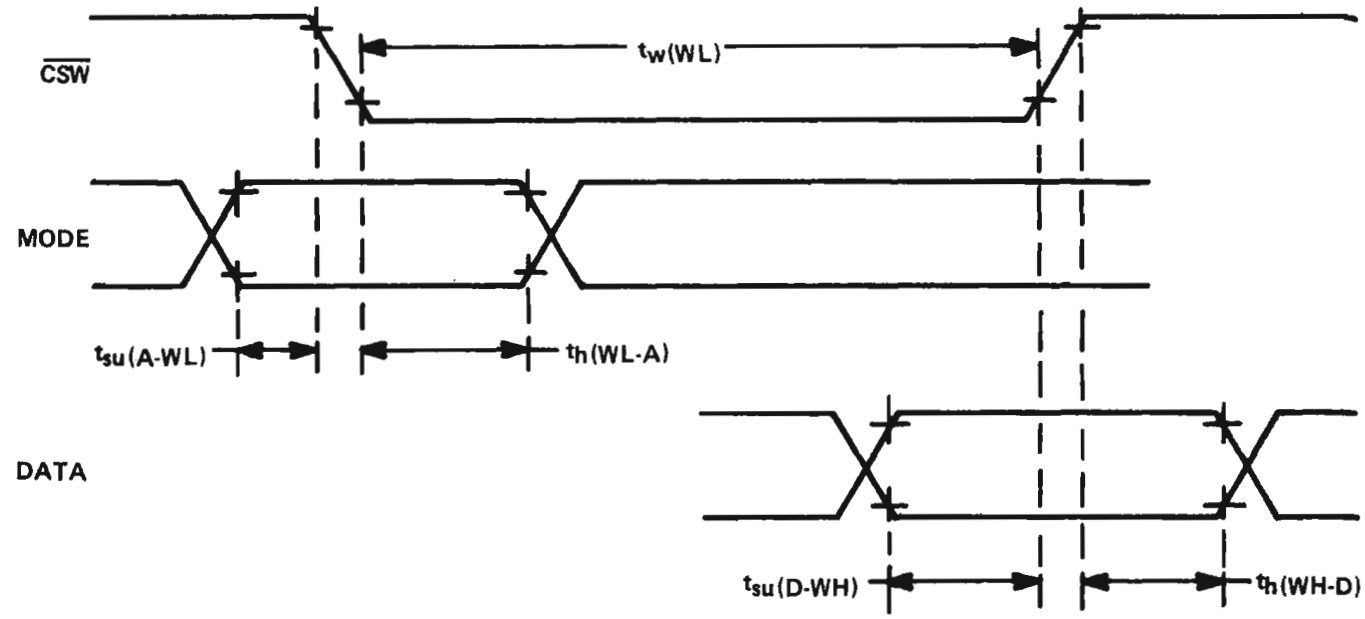
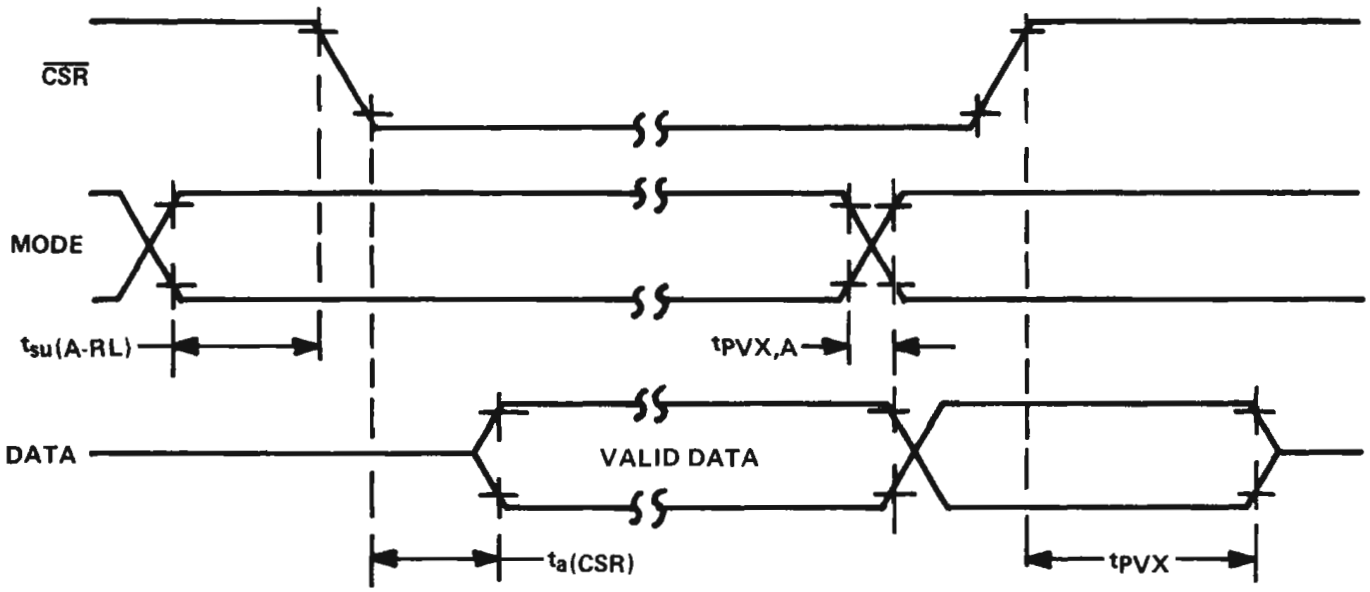



FIGURE 5-3 – CPU-VDP WRITE CYCLE FOR TMS9918A/9928A/9929A

READ CYCLE




NOTE: All measurements are made at 10% and 90% points.

FIGURE 5-4 – CPU-VDP READ CYCLE FOR TMS9918A/9928A/9929A

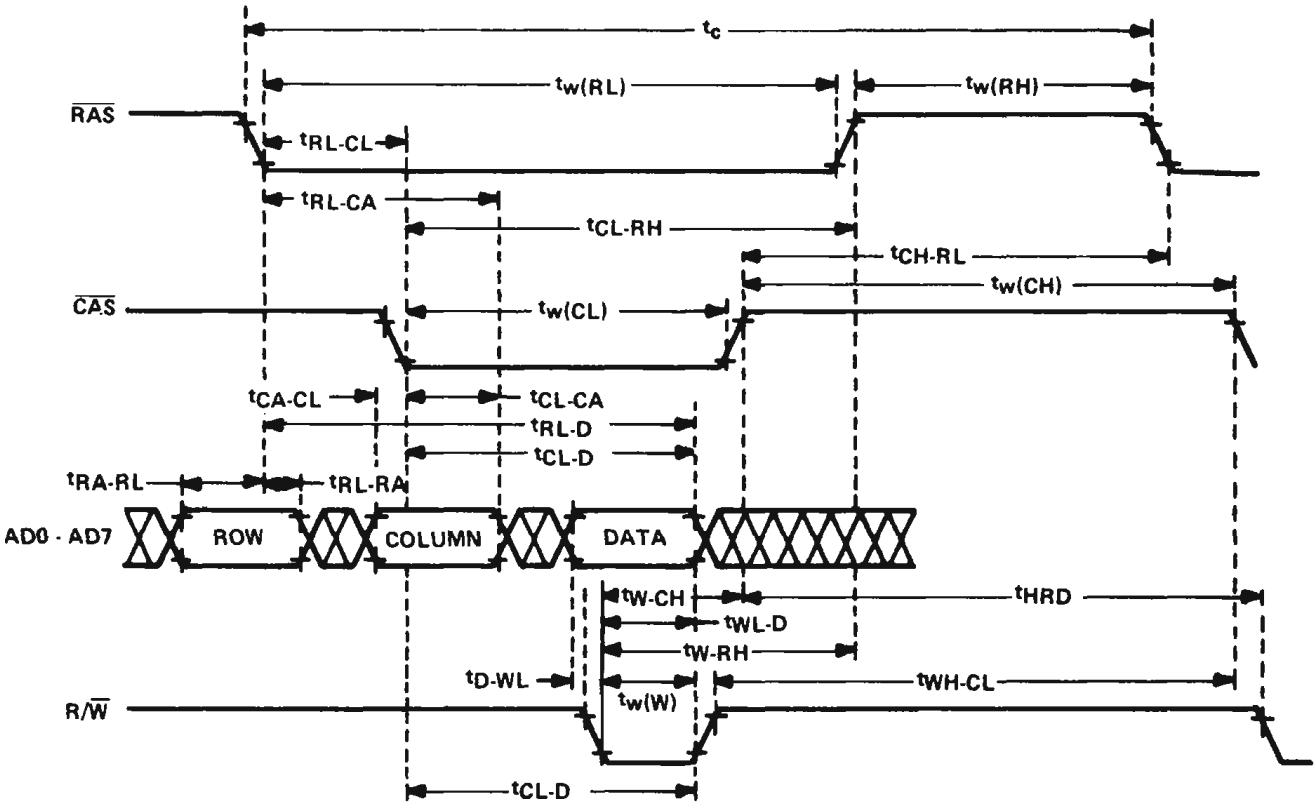
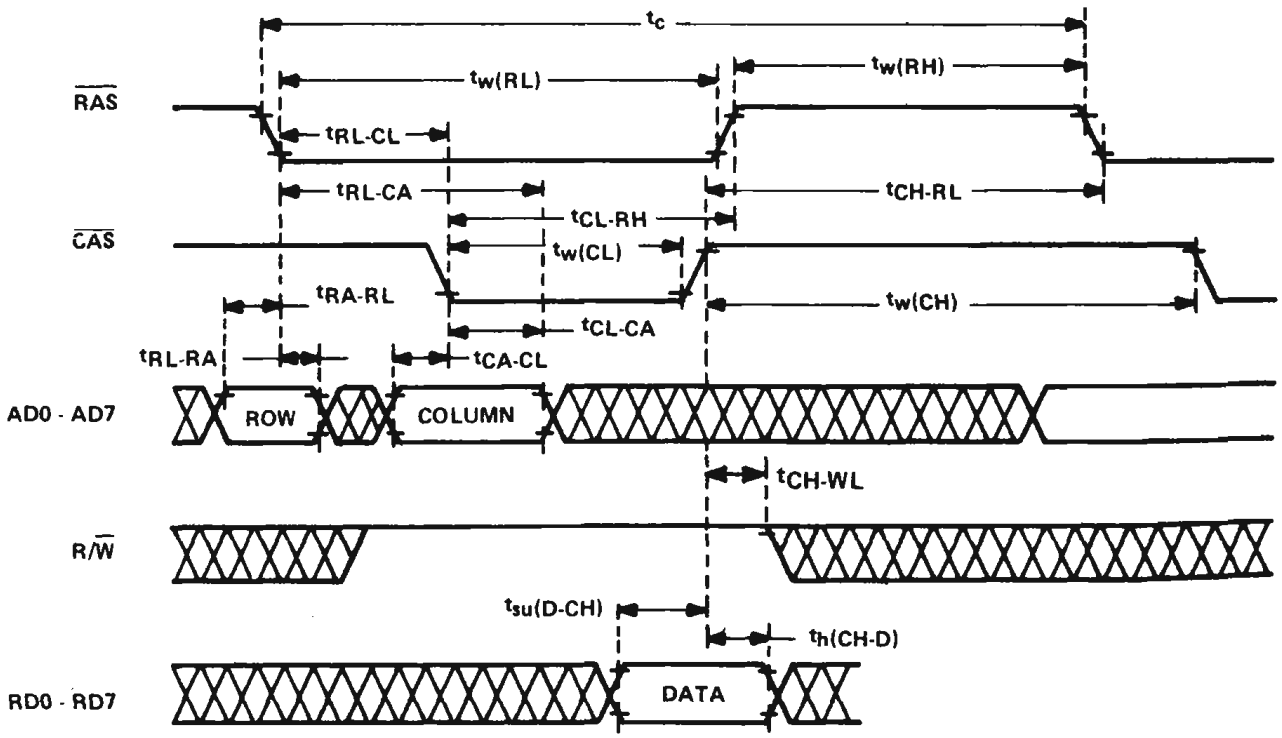
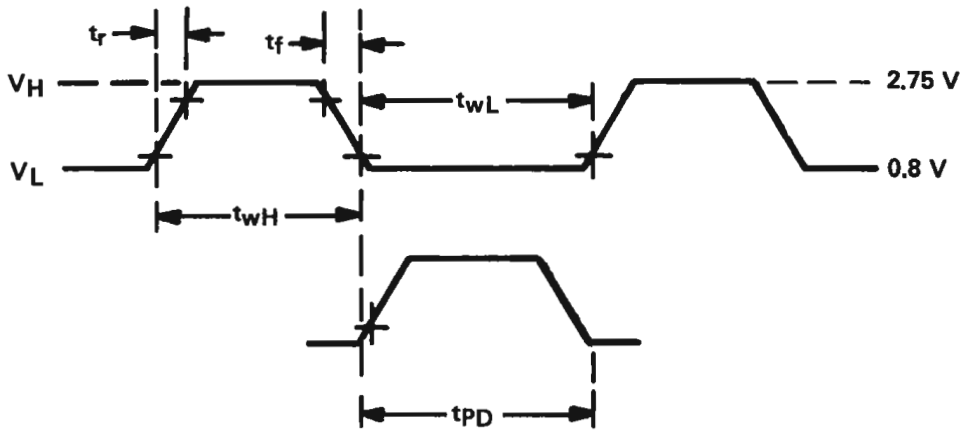



FIGURE 5-5 – VRAM WRITE CYCLE




NOTE: All measurements are made at 10% and 90% points.

FIGURE 5-6 – VRAM READ CYCLE




NOTE: All measurements are made at 10% and 90% points.

FIGURE 5-7 — EXTERNAL CLOCK TIMING WAVEFORM

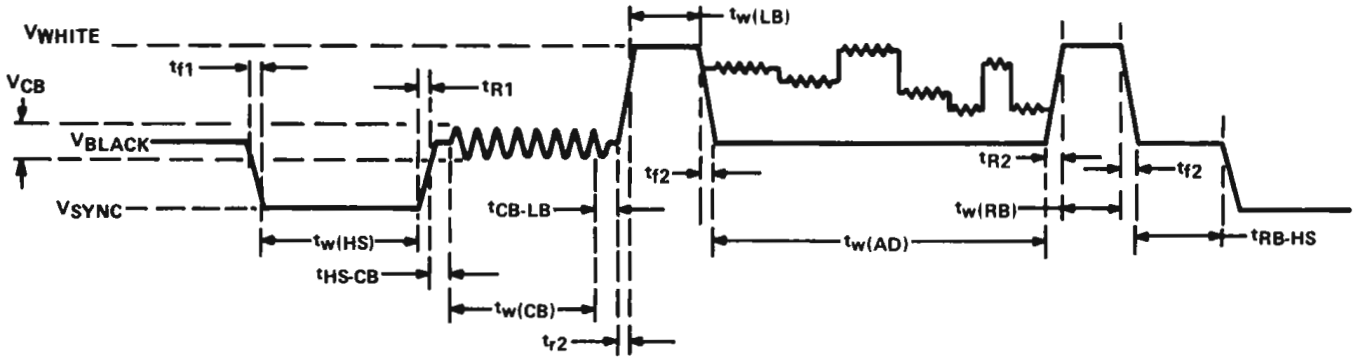
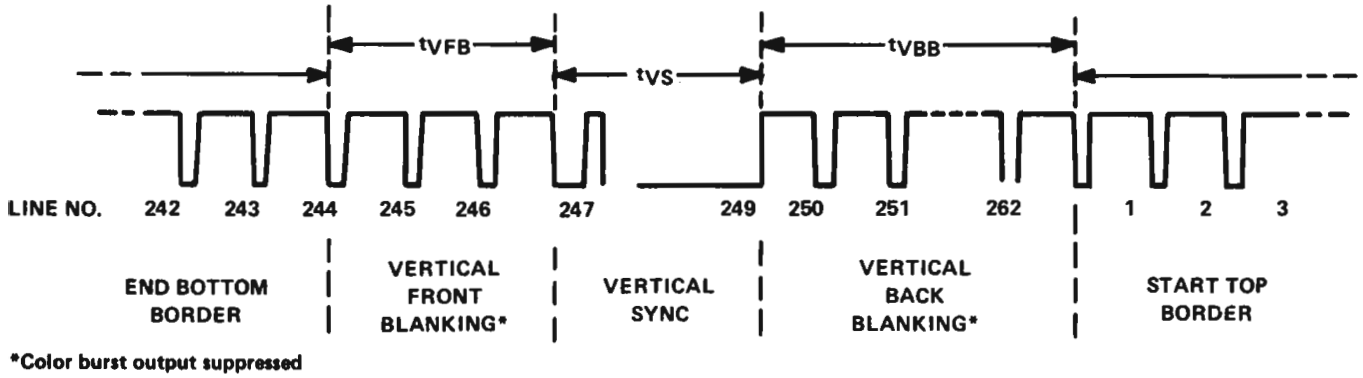



FIGURE 5-8 — TMS9918A COMVID HORIZONTAL TIMING




*Color burst output suppressed

FIGURE 5-9 — TMS9918A VERTICAL TIMING

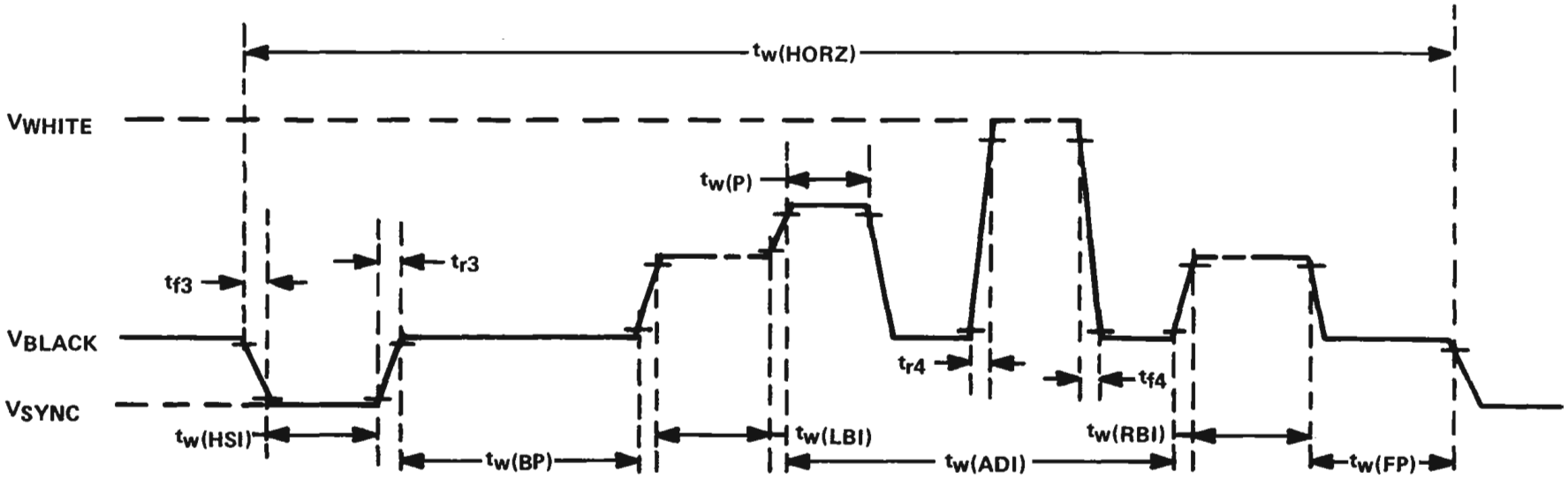
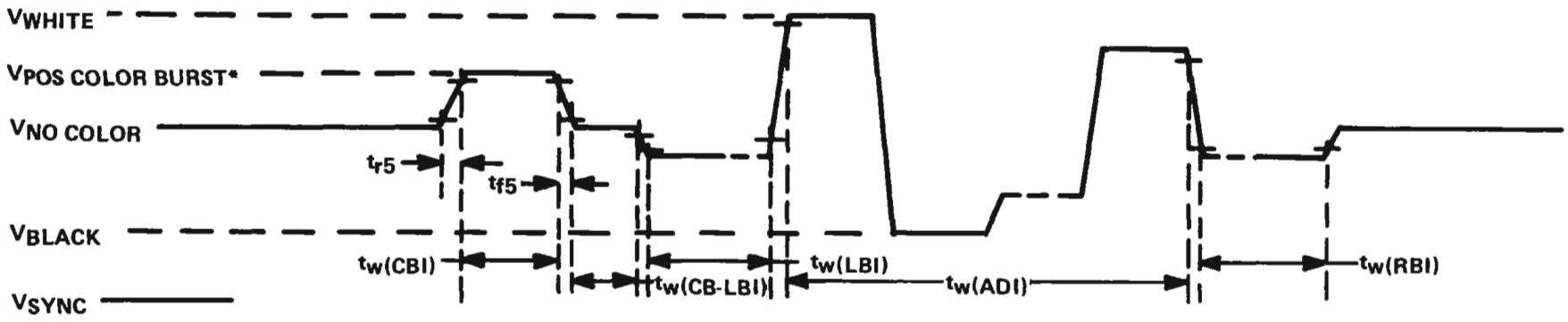


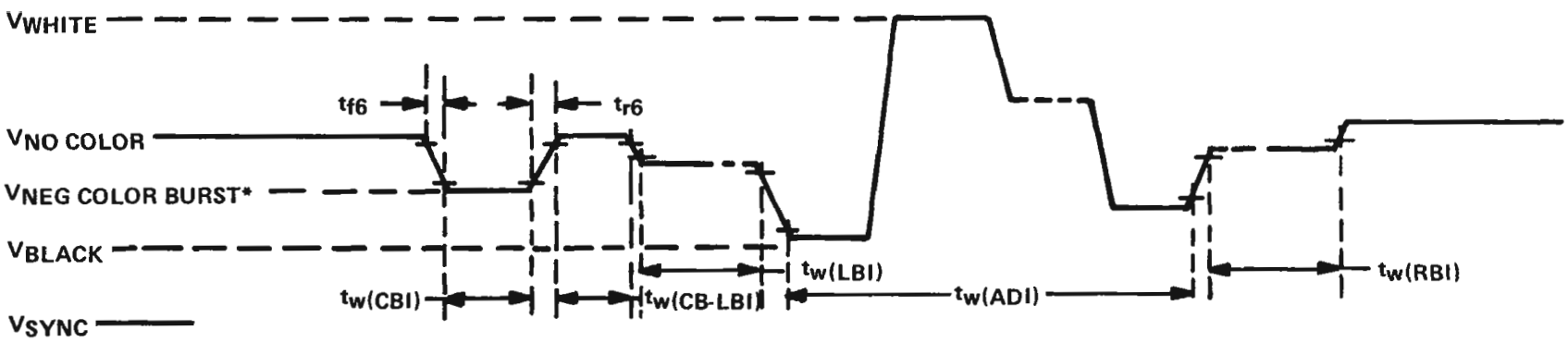

FIGURE 5-10 — TMS9928A/9929A Y HORIZONTAL TIMING






*Absent for the TMS9928A

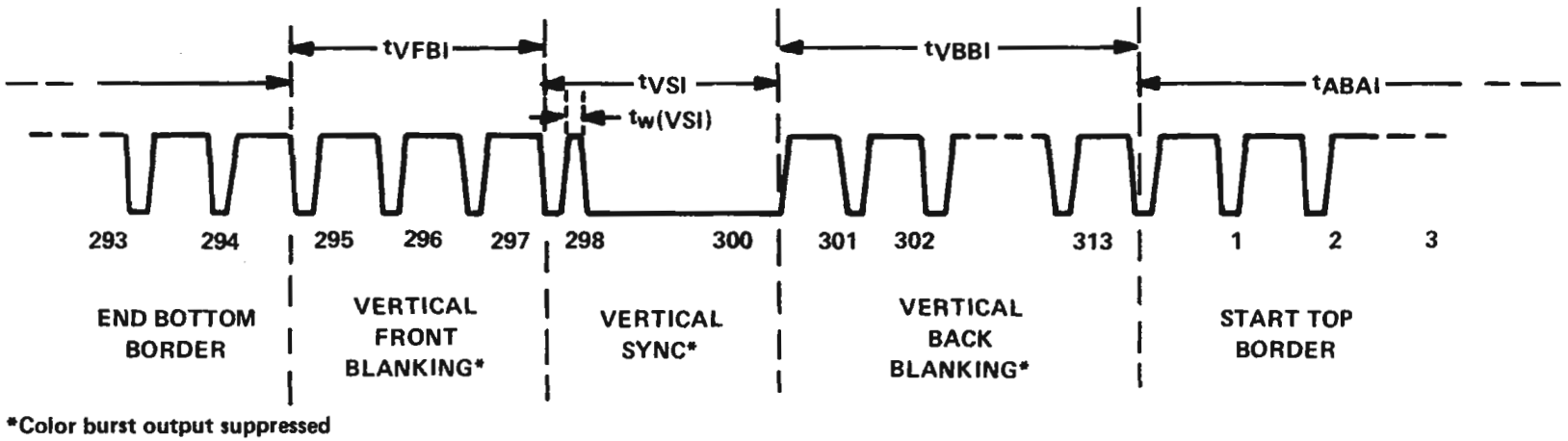
FIGURE 5-11 — TMS9928A/9929A R-Y HORIZONTAL TIMING




*Absent for the TMS9928A

FIGURE 5-12 — TMS9928A/9929A B-Y HORIZONTAL TIMING



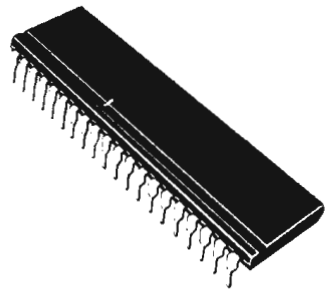


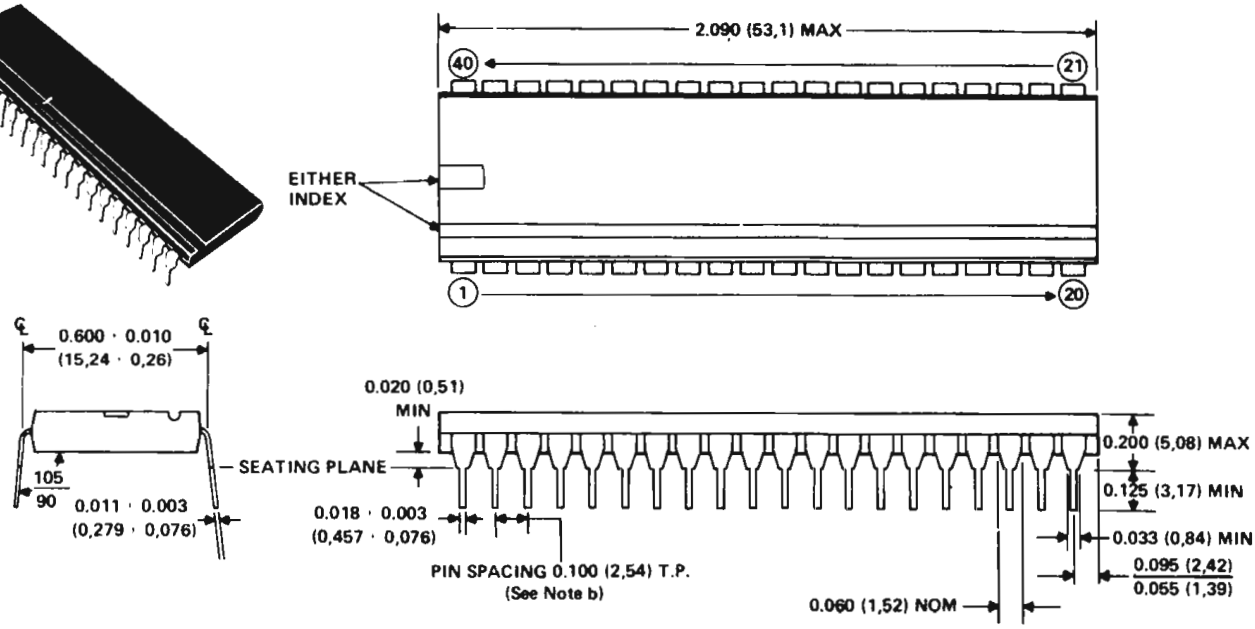

*Color burst output suppressed

FIGURE 5-13 — TMS9929A VERTICAL TIMING

# 6. MECHANICAL DATA

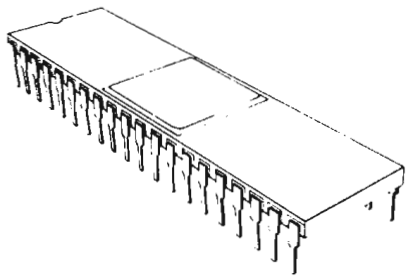
## 6.1 TMS9918 40-PIN PLASTIC DUAL-IN-LINE PACKAGE

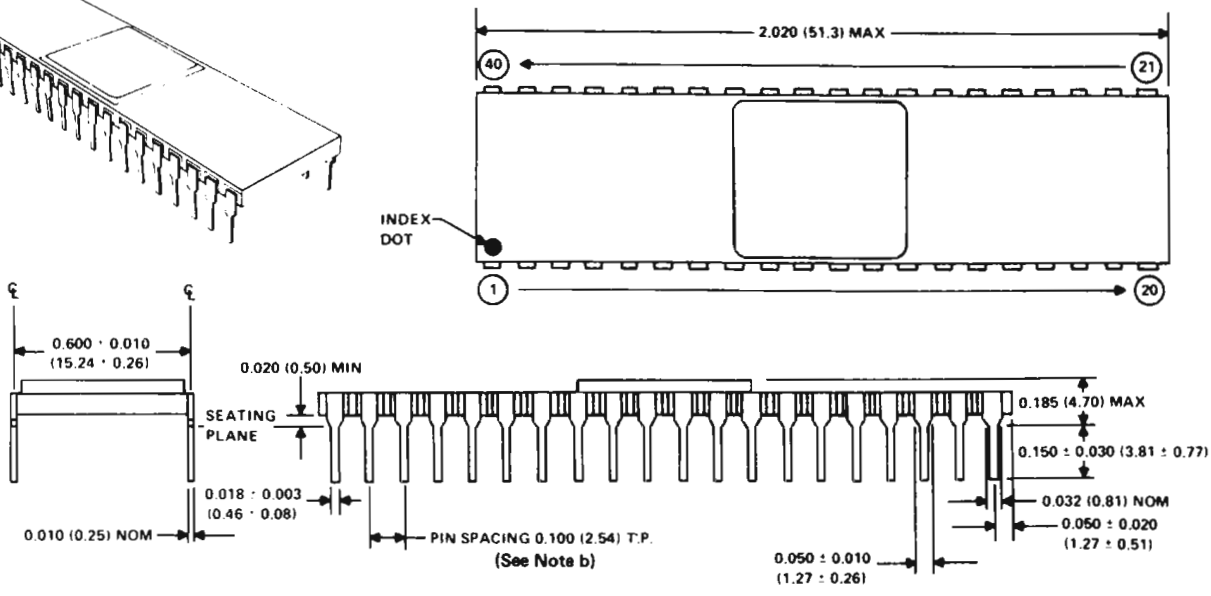





NOTES: a. All linear dimensions are in inches and parenthetically in millimeters. Inch dimensions goven.
b. Each pin centerline is located within 0.010 (0.25) of its true longitudinal position.

## 6.2 TMS9918 40-PIN CERAMIC DUAL-IN-LINE PACKAGE






NOTES: a. All linear dimensions are in inches and parenthetically in millimeters. Inch dimensions goven.
b. Each pin centerline is located within 0.010 (0.25) of its true longitudinal position.

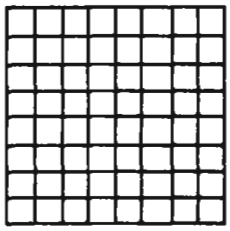
# APPENDIX A

# ASCII CHARACTER SET

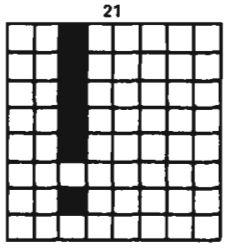
Software programs apply to all three VDPs (TMS9918A/9928A/9929A).

This appendix contains the diagrams and software listing of an upper and lower case ASCII character set. The character matrix is 5 X 7 in the 8 X 8 pixel block. These characters are left-justified so they can be used in the text (6 X 8 pixels) mode.

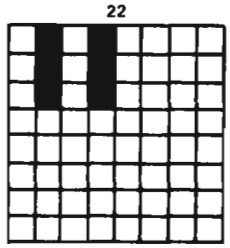
PATTERN 20



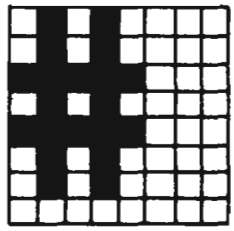

PATTERN 21



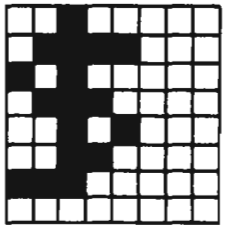

PATTERN 22



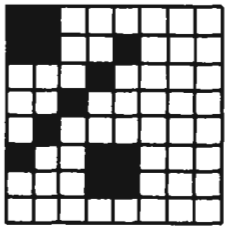

PATTERN 23



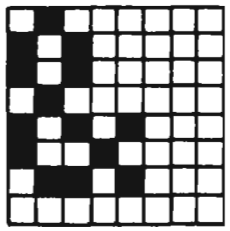

PATTERN 24



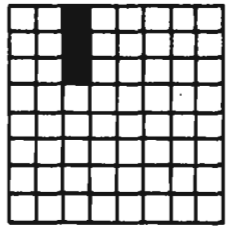

PATTERN 25



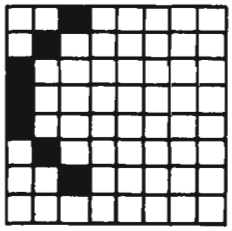

PATTERN 26



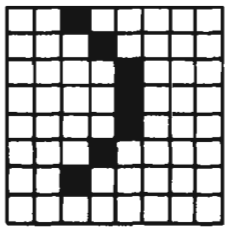

PATTERN 27



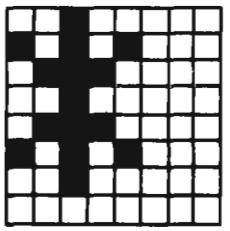

PATTERN 28



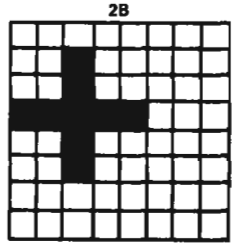

PATTERN 29



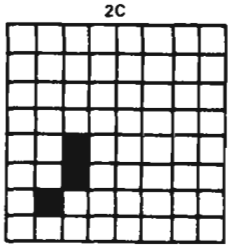

PATTERN 2A



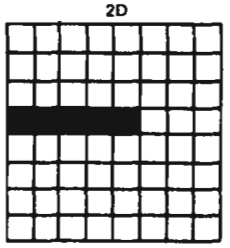

PATTERN 2B



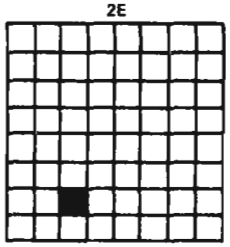

PATTERN 2C



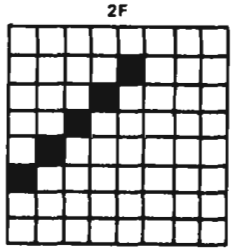

PATTERN 2D



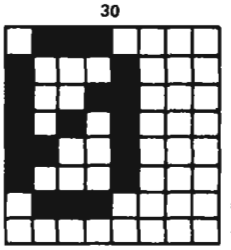

PATTERN 2E



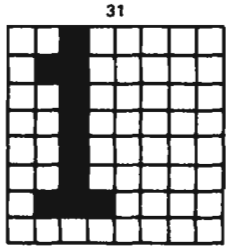

PATTERN 2F



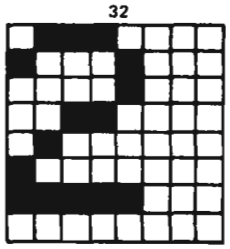

PATTERN 30



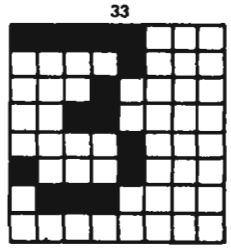

PATTERN 31



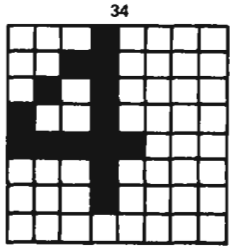

PATTERN 32



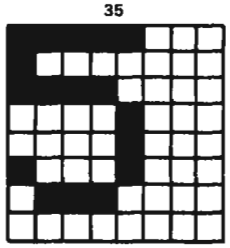

PATTERN 33



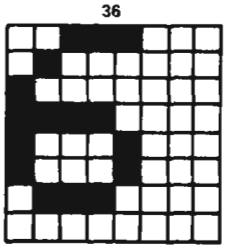

PATTERN 34



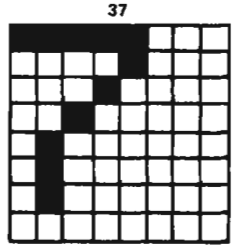

PATTERN 35



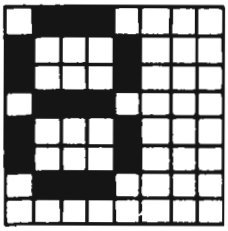

PATTERN 36



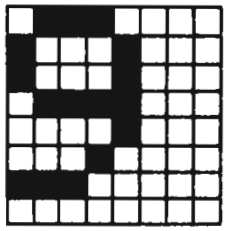

PATTERN 37



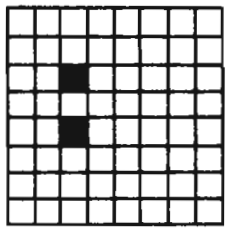
PATTERN 38



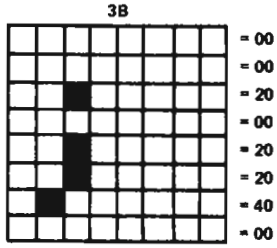

PATTERN 39



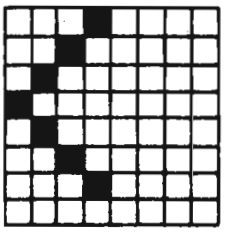

PATTERN 3A



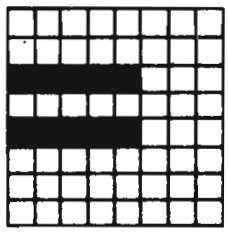

PATTERN 3B



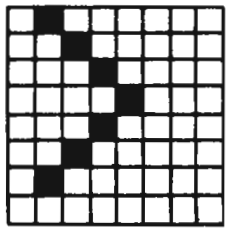

PATTERN 3C



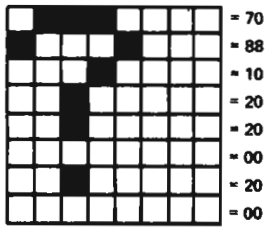

PATTERN 3D



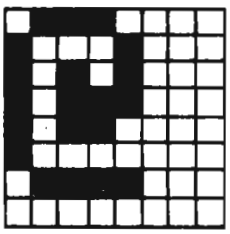

PATTERN 3E



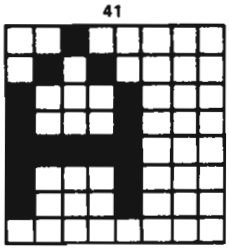

PATTERN 3F



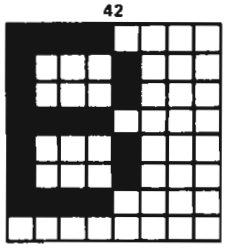

PATTERN 40



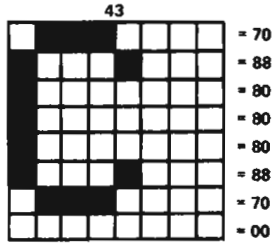

PATTERN 41



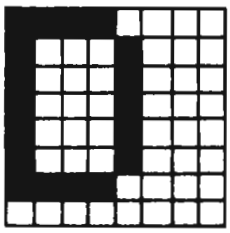

PATTERN 42



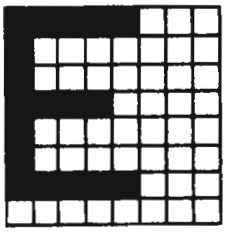

PATTERN 43



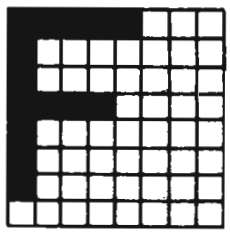

PATTERN 44



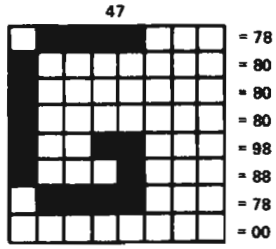

PATTERN 45



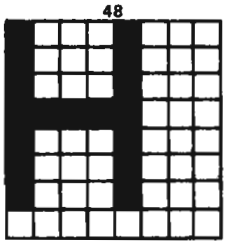

PATTERN 46



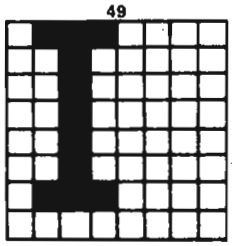

PATTERN 47



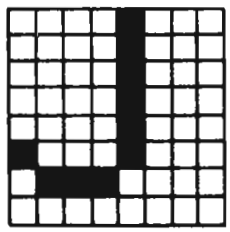

PATTERN 48



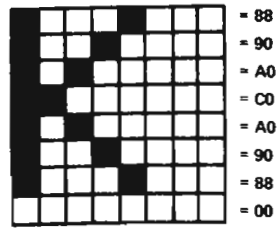

PATTERN 49



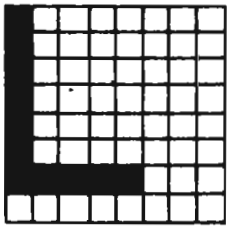

PATTERN 4A



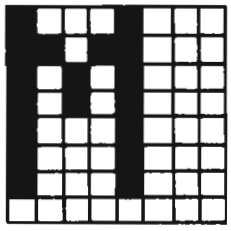

PATTERN 4B



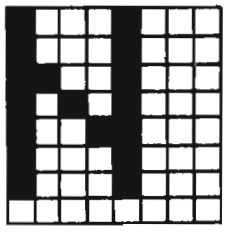

PATTERN 4C



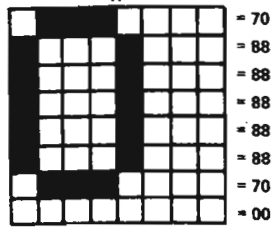

PATTERN 4D



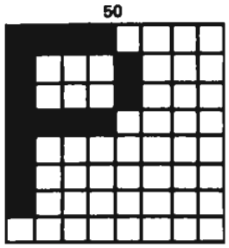

PATTERN 4E




PATTERN 4F

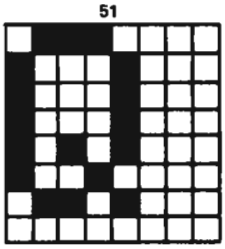


PATTERN 50



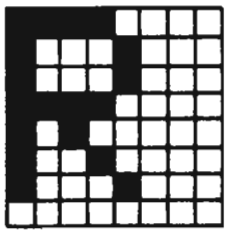
= F0
= 88
= 88
= F0
= 80
= 80
= 80
= 00

PATTERN 51



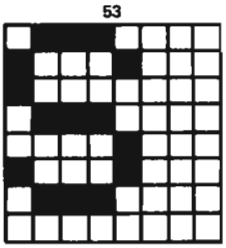
= 70
= 88
= 88
= 88
= A8
= 90
= 68
= 00

PATTERN 52



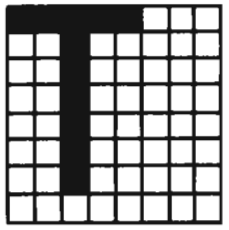
= F0
= 88
= 88
= F0
= A0
= 90
= 88
= 00

PATTERN 53



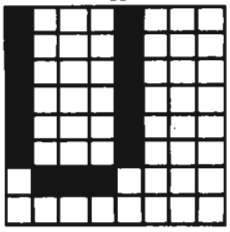
= 70
= 88
= 80
= 70
= 08
= 88
= 70
= 00

PATTERN 54



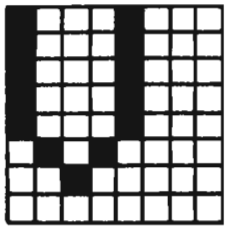
= F8
= 20
= 20
= 20
= 20
= 20
= 20
= 00

PATTERN 55



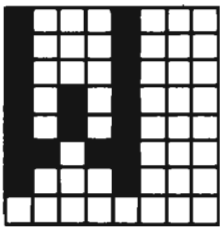
= 88
= 88
= 88
= 88
= 88
= 88
= 70
= 00

PATTERN 56



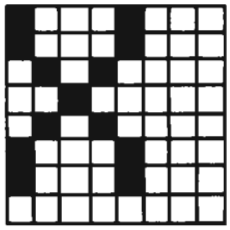
= 88
= 88
= 88
= 88
= 88
= 50
= 20
= 00

PATTERN 57



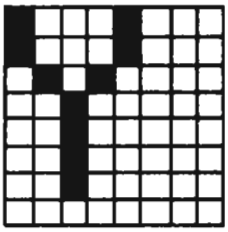
= 88
= 88
= 88
= A8
= A8
= D8
= 88
= 00

PATTERN 58



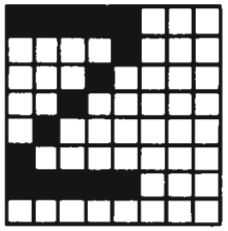
= 88
= 88
= 50
= 20
= 50
= 88
= 88
= 00

PATTERN 59



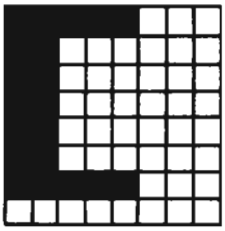
= 88
= 88
= 50
= 20
= 20
= 20
= 20
= 00

PATTERN 5A



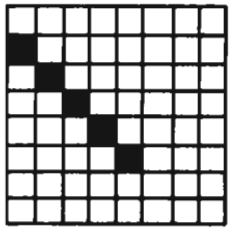
= F8
= 08
= 10
= 20
= 40
= 80
= F8
= 00

PATTERN 5B



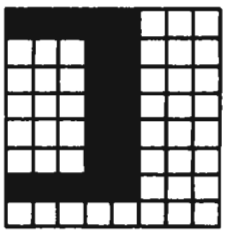
= F8
= C0
= C0
= C0
= C0
= C0
= F8
= 00

PATTERN 5C



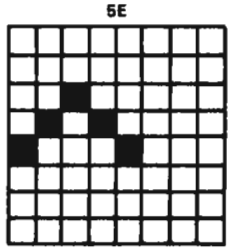
= 00
= 80
= 40
= 20
= 10
= 08
= 00
= 00

PATTERN 5D



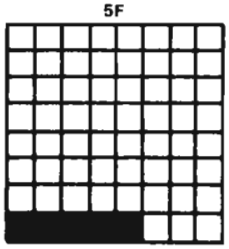
= F8
= 18
= 18
= 18
= 18
= F8
= 00

PATTERN 5E



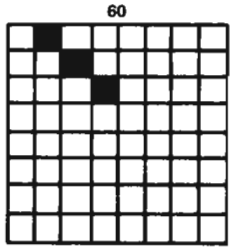
= 00
= 00
= 20
= 50
= 88
= 00
= 00
= 00

PATTERN 5F



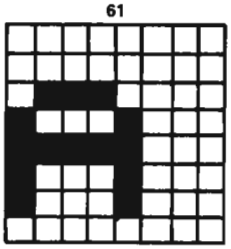
= 00
= 00
= 00
= 00
= 00
= 00
= 00
= F8

PATTERN 60



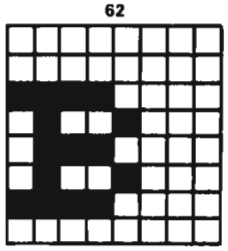
= 40
= 20
= 10
= 00
= 00
= 00
= 00
= 00

PATTERN 61



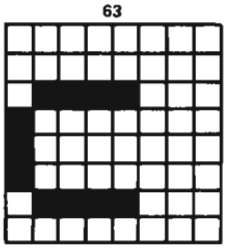
= 00
= 00
= 70
= 88
= F8
= 88
= 88
= 00

PATTERN 62



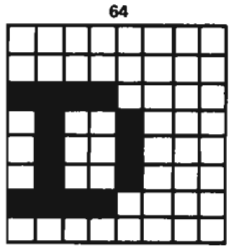
= 00
= 00
= F0
= 48
= 70
= 48
= F0
= 00

PATTERN 63

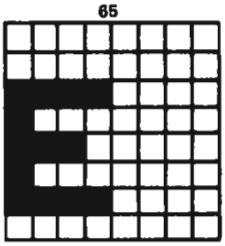


= 00
= 00
= 78
= 80
= 80
= 80
= 78
= 00

PATTERN 64

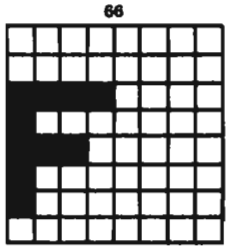


PATTERN 65

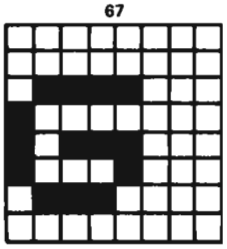


= 00
= 00
= F0
= 80
= E0
= 80
= F0
= 00

PATTERN 66

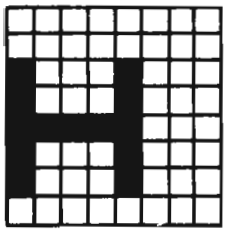


PATTERN 67

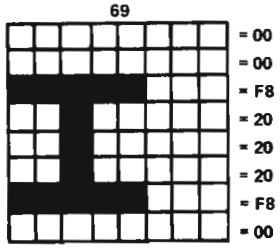


= 00
= 00
= 78
= 80
= B8
= 88
= 70
= 00

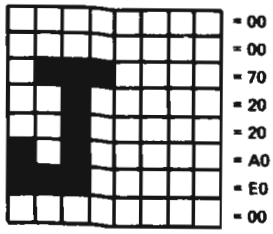
PATTERN 68



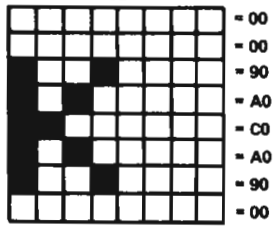

PATTERN 69



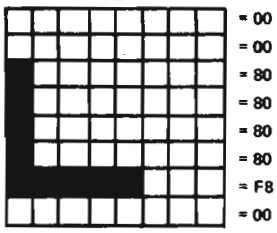

PATTERN 6A



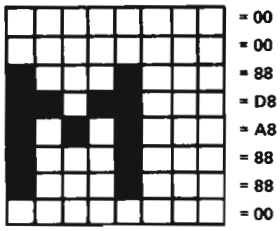

PATTERN 6B



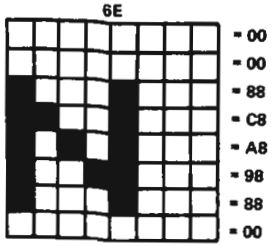

PATTERN 6C



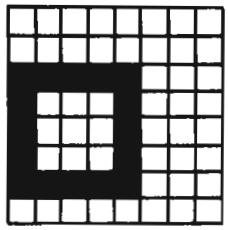

PATTERN 6D



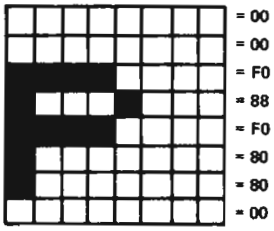

PATTERN 6E



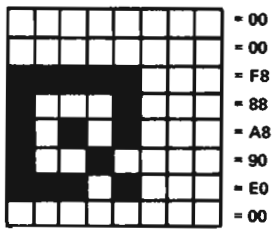

PATTERN 6F



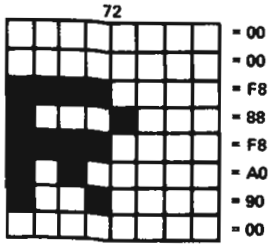
PATTERN 70



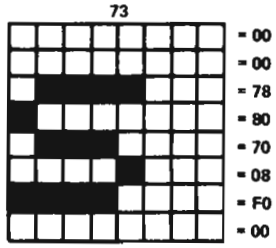

PATTERN 71



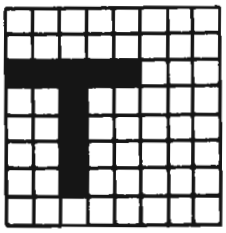

PATTERN 72



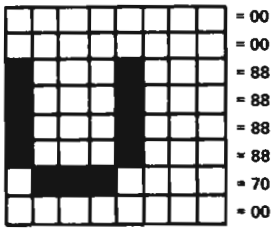

PATTERN 73



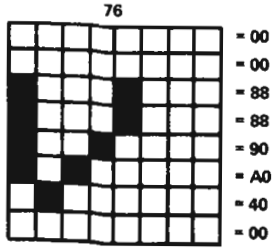

PATTERN 74



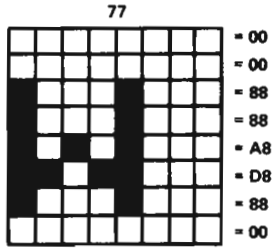
PATTERN 75



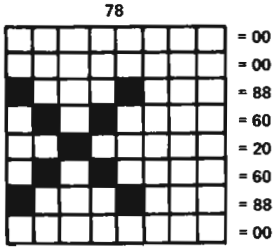

PATTERN 76



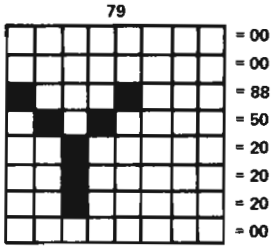

PATTERN 77



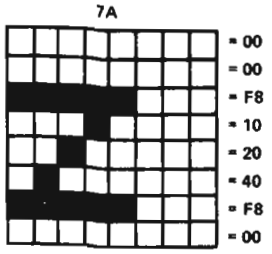

PATTERN 78



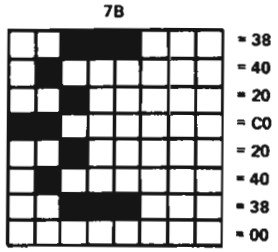

PATTERN 79



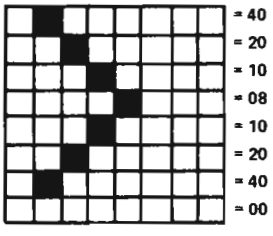

PATTERN 7A



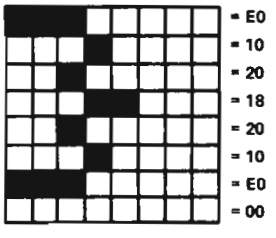

PATTERN 7B



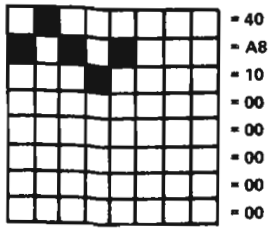

PATTERN 7C



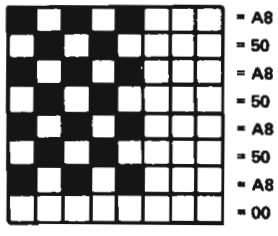

PATTERN 7D




PATTERN 7E




PATTERN 7F



## APPENDIX B

## CHOOSING VRAM MEMORY

When choosing the VRAM memory, the user must take into consideration the propagation delay times of the system in addition to the access time of the memory and data setup time of the VDP.

After the VDP outputs a low level signal on $\overline{RAS}$, there is a delay time ($t_{d(RAS)}$) for this low level to reach the VRAM memory; there is a similar delay ($t_{d(CAS)}$) for a signal output on the $\overline{CAS}$ pin to reach the VRAM memory. Finally, there is a delay ($t_{d(data)}$) for data output by the memory to reach the VDP. These delays (shown in Figure B1) depend on the length of the wires between VDP and memory, and on the capacitive load being driven.

Valid data appearing on RD0-RD7 is strobed into the VDP when $\overline{CAS}$ is brought high. Therefore, the memory chosen must have fast enough access times, $t_{a(R)}$ and $t_{a(C)}$, so that valid data is present on RD0-RD7 when a positive transition occurs on $\overline{CAS}$.

For 16-K memories from Texas Instruments (TMS4116-XX), the times, $t_{RL-CL}$ and $t_{a(C)}$, can vary, but their sum is equal to $t_{a(R)}$ ($t_{RL-CL} + t_{a(C)} = t_{a(R)}$). Thus, when $t_{d(RAS)} \geq t_{d(CAS)}$, the limiting access time is $t_{a(R)}$.

After the memory receives a negative transition on the $\overline{RAS}$ input, the memory access time, $t_{a(R)}$, must be fast enough so that valid data is present on RD0-RD7 when $\overline{CAS}$ goes high (see Figure B2). The equation for this is:

$$t_{RL-CL} + t_{wCL} \geq t_{d(R)} + t_{d(data)} + t_{su(D-CH)}$$

Under worst case conditions, this equation can be used to find out how much time is allowed for system delays using different memories.

**TABLE B-1 — WORST CASE TIMING FOR VDP.**

| MEMORIES | SYSTEM DELAYS |
|---|---|
| $t_{wCL}$ | 230 ns MIN |
| $t_{RL-CL}$ | 40 ns MIN |
| $t_{su(D-CH)}$ | 60 ns MAX |

If the values from Table B1 are placed in the equation, we find ($t_{RL-CL} + t_{wCL}$) VDP MIN $\geq$ [($t_{d(RAS)} + t_{d(data)}$) SYS + $t_{a(R)}$ MEM + $t_{suD-CH}$ VDP MAX]

210 ns - $t_{a(R)}$ MEM MAX $\geq$ [ $t_{d(RAS)} + t_{d(data)}$] SYS MAX

**TABLE B-2 — DRAM SYSTEM DELAYS**

| PART NO. | $t_{a(R)}$ | SYSTEM DELAYS |
|---|---|---|
| 4116-15 | 150 ns | 60 ns MAX |
| 4116-20 | 200 ns | 10 ns MAX |
| 4116-25 | 250 ns | -40 ns MAX |

From the data given here, the VDP will work with both -15 and -20 TMS4116 dynamic RAMs provided the system delays are small enough. The VDP does not meet the $t_{a(R)}$ specifications for the -25 TMS4116 and is unable to use the -25 under worst case conditions. The VDP has been verified to work with both -15 and -20 TMS4116s in a system application. Note that in addition to the equation derived above, that all memory timing requirements must be met as specified in a memory data book.

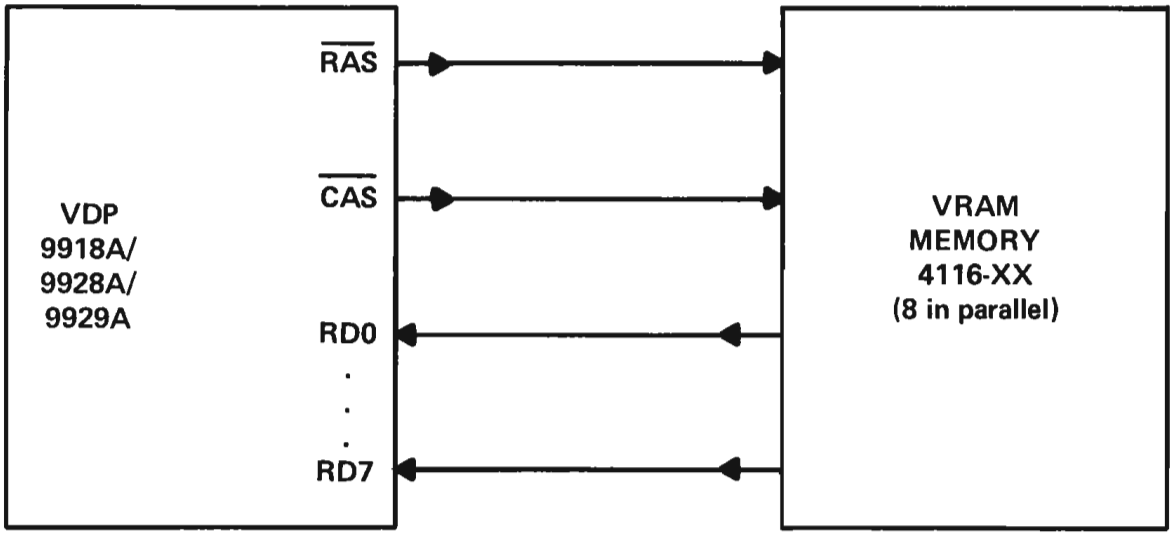



FIGURE B-1 – MEMORY CONFIGURATION SHOWING DELAY TIMES

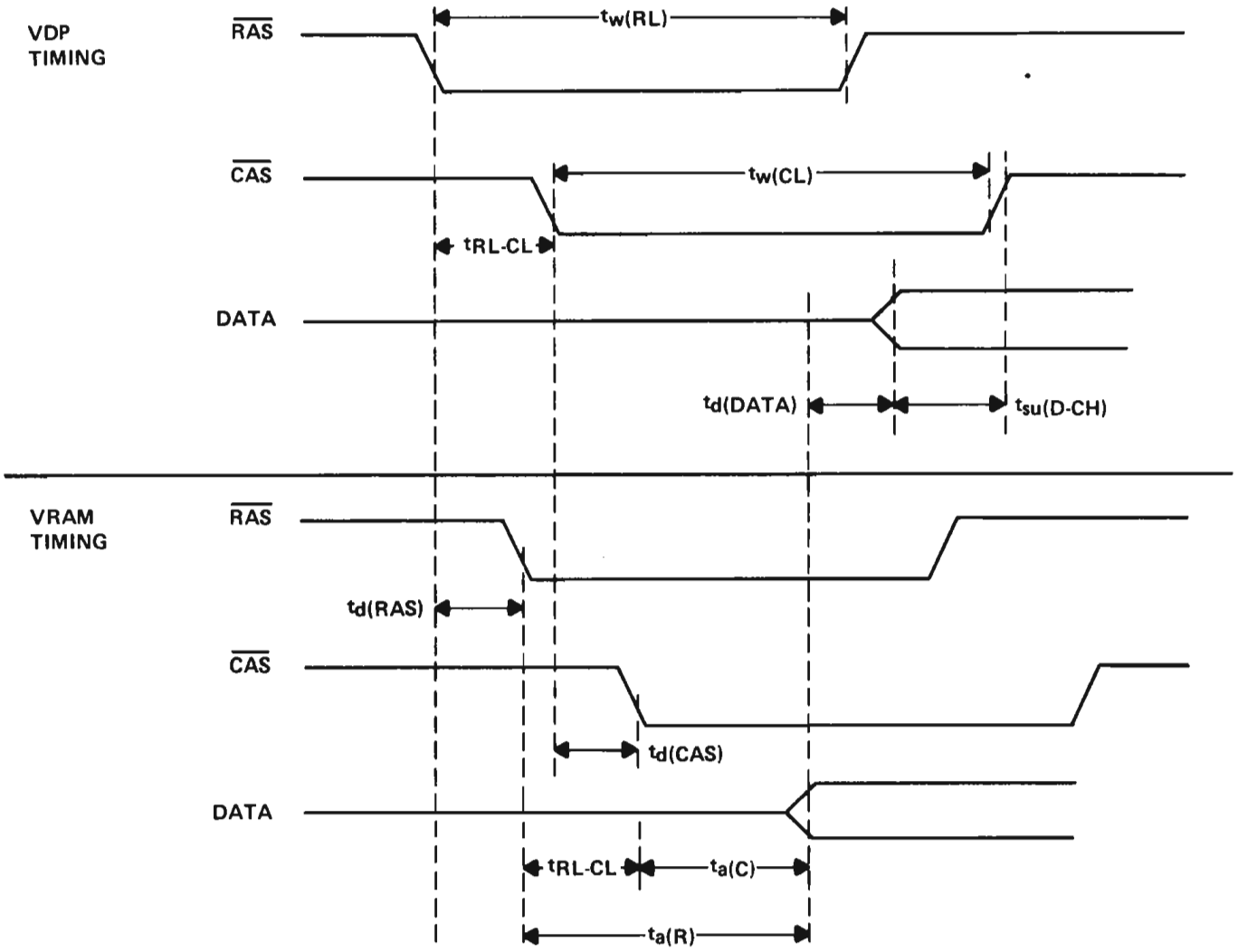
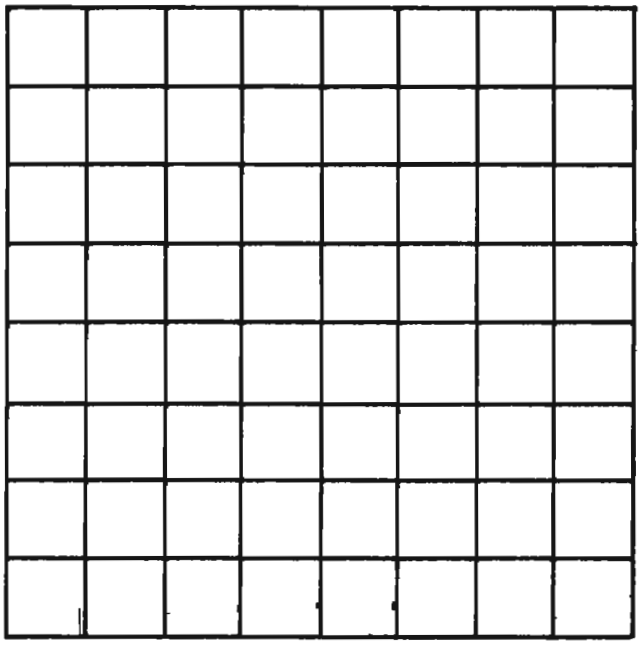


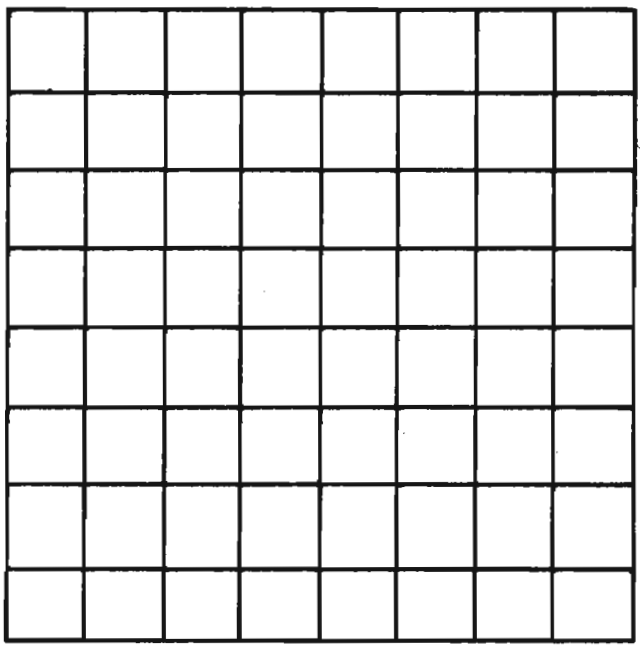

FIGURE B-2 – RELATIVE TIMING OF VRAM TO VDP

# APPENDIX C

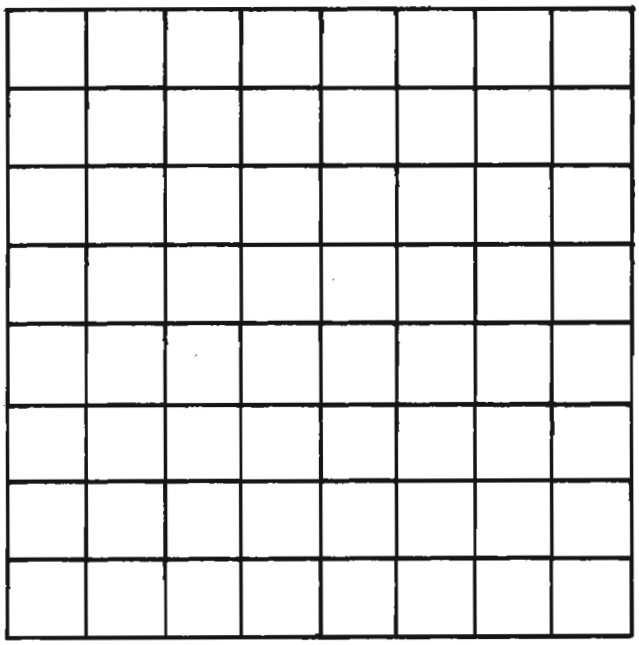
## PATTERN AND SCREEN WORKSHEETS

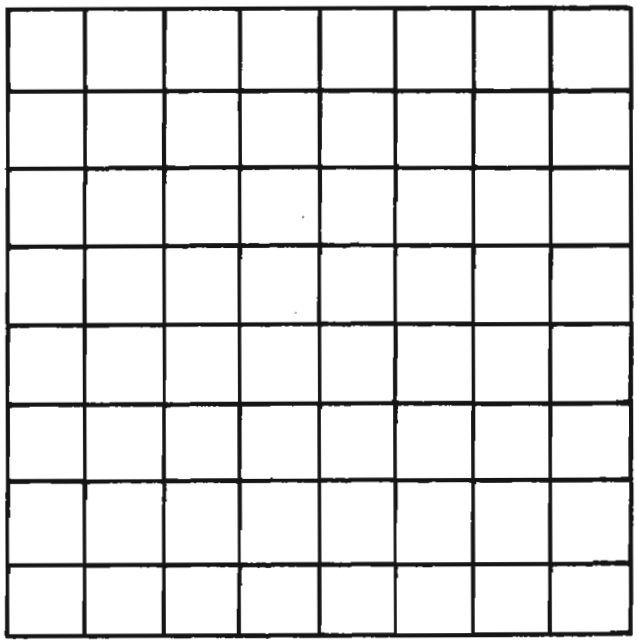
PATTERN
NAME





PATTERN
NAME





TEXAS INSTRUMENTS

November 1982
MP010A

Post Office Box 1443 • Houston, Texas 77001
Semiconductor Group

Printed in U.S.A.