V9938 MSX-VIDEO
Technical Data Book

ASCII CORPORATION / NIPPON GAKKI CO., LTD.

YAMAHA
NIPPON GAKKI CO., LTD.

PREFACE

The V9938 introduced in this manual is a Very Large-Scale Integrated Circuit (VLSI) that was developed as a Video Display Processor (VDP) for the MSX2. The MSX personal computer standard was introduced in 1983 by ASCII Corporation and Microsoft Incorporated. At present, the MSX is manufactured and marketed worldwide. In 1985, out of the desire to strengthen some of the functions of the original MSX, the MSX2 standard was developed. In addition to being software-compatible with the MSX, the MSX2 supports new media and has video processing capabilities that are not available on conventional 8-bit personal computers.

To make the MSX2 a reality, two requirements for the Video Processor were upward compatiblity with the existing TMS9918A (the VDP for the MSX) software while increasing the number of functions. The V9938 was developed through the joint efforts of ASCII Corporation, Microsoft Incorporated, and YAMAHA.

The following functions are supported on the V9938.

- Full bit-mapped mode
- 80-column text display
- Access using X- and Y-coordinates. The load of the I/O driver has been lightened. The X-Y coordinates are independent of the screen mode.
- Fundamental commands implemented by hardware to decrease the processing time of the I/O driver: AREA MOVE, LINE, SEARCH, RASTER OPERATION, etc.
- Digitize and external synchronization
- Color palette (9 bits x 16 patterns)
- Linear RGB video output
- More sprites per horizonatal line

Because the V9938 has the above functions, it provides for superior video capabilities that make it possible for its use in a variety of applications, including the MSX2. CAPTAIN terminals and NAPLPS terminals using the V9938 have already been developed. We hope that the V9938 will be a standard video processing device on a worldwide basis.

This manual was written so as to explain how to set the parameters of the V9938 and is a reference for developing applications and systems software for it.

We are pleased that you have chosen to develop software for the V9938 and that you have referred to this manual for assistance.

Finally, we would like to express our deep gratitude to the people at NTT as well as the other related manufacturers for their valuable opinions which contributed to the development of the V9938.

August, 1985

ASCII Corporation

CONTENTS

PART 2 MSX-VIDEO DATA PROCESSOR LSI DATA SHEET

APPENDIX

PART 1

MSX-VIDEO DATA PROCESSOR
V9938
USER'S MANUAL

V9938 MSX-VIDEO USER'S MANUAL

BASIC INPUT AND OUTPUT

1. Accessing the Control Registers

There are two ways to set data in the MSX-VIDEO control registers (R#0 to R#46), which we will describe below.

1.1 Direct access

Output the data and the register number in sequence to port #1. Since this order is always used, be careful when you access the MSX-VIDEO for an interrupt routine.

	MSB 7	6	5	4	3	2	1	0 LSB	
Port #1 First byte	D7	D6	D5	D4	D3	D2	D1	D0	DATA
Second byte	1	0	R5	R4	R3	R2	R1	R0	REGISTER #

1.2 Indirect access

Specify the register number in control register R#17 (Control Register Pointer).

First set the register number in R#17 (using direct addressing) by sending data to Port #3. When you set the data in R#17, you can also set its MSB (AII, the autoincrement bit) to control autoincrementing. The data in R#17 cannot be changed by indirect addressing.

If autoincrementing is prohibited, the contents of R#17 will be unchanged, and thus you do not have to reset R#17.

	MSB 7	6	5	4	3	2	1	0 LSB	
Register #17	AII	0	R5	R4	R3	R2	R1	R0	REGISTER #
	1: Auto increment inhibit 0: Auto increment on								
Port #3 First byte	D7	D6	D5	D4	D3	D2	D1	D0	DATA
Port #3 Second byte	D7	D6	D5	D4	D3	D2	D1	D0	DATA
Port #3 nth byte	D7	D6	D5	D4	D3	D2	D1	D0	DATA

2. Accessing the Pallete Registers

To set data in the MSX-VIDEO palette registers (P#0 to P#15/9 bit), you must first set the palette register number in register R#16 (Color palette address pointer) and subsequently output the two bytes of data (in order) through port #2.

	MSB 7	6	5	4	3	2	1	0 LSB	
Register #16	0	0	0	0	C3	C2	C1	C0	Pallete #
Port #2 First byte	0	R2	R1	R0	0	B2	B1	B0	DATA
		Red data				Blue data			
Port #2 Second byte	0	0	0	0	0	G2	G1	G0	DATA
						Green data			

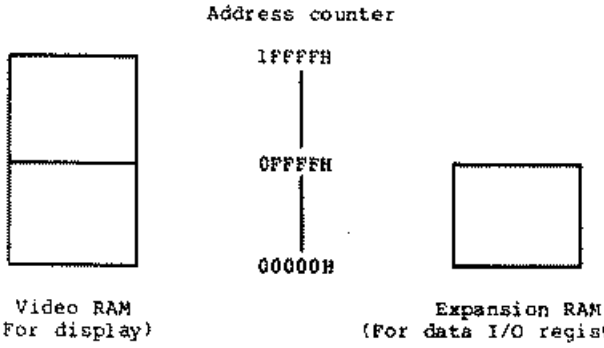
3. Accessing the Status Registers

To read the status registers of the MSX-VIDEO (S#0 to S#9), you must first set the register number in R#15 (Status register pointer) and read the data through Port #1.

	MSB 7	6	5	4	3	2	1	0 LSB	
Register #15	0	0	0	0	S3	S2	S1	S0	Status register #
Port #1 Read data	D7	D6	D5	D4	D3	D2	D1	D0	DATA

4. Accessing the Video RAM

A Video RAM of 128K bytes plus an expansion RAM of 64K bytes can be connected to the MSX-VIDEO. The memory maps for these cases are shown in the map below.



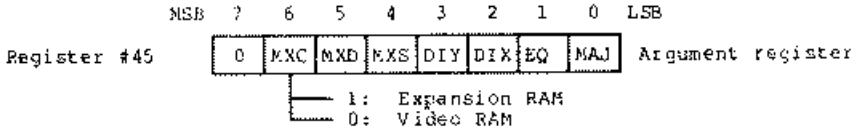
Accessing memory

To access memory, follow the procedure below.

1. Switch banks (VRAM to Expansion RAM)
2. Set the address counter (A16 to A14)
3. Set the address counter (A7 to A0)
4. Set the address counter (A13 to A8), and specify read or write
5. Read or write the data

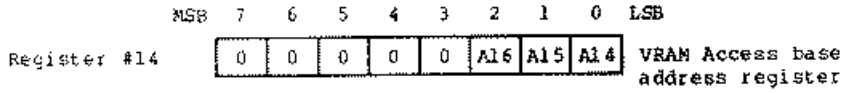
1. Switching banks (VRAM to Expansion RAM)

Since the contents of R#45 (Argument register) do not change each time that memory is accessed, it is not necessary to respecify bit 6 of register R#45 (which specifies banking) every time that you are to do banking.



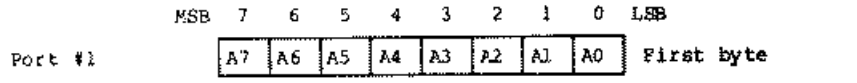
2. Setting the address counter (A16-A14)

Set the high-order three bits (A16 to A14) of the address counter using register R#14 (VRAM Access base address register).



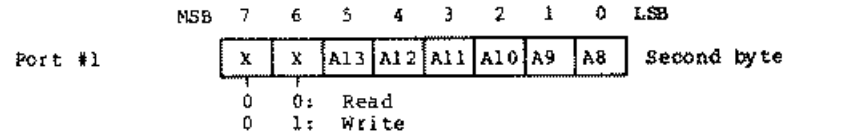
3. Setting the address counter (A7 to A0)

Set the low-order eight bits (A7 to A0) of the address counter by outputting data to Port #1.



4. Setting the address counter (A13 to A8) and specifying read or write

Set the remaining six bits (A13 to A8) of the address counter and specify read or write by outputting data to Port #1.



5. Reading or writing data

Since the address counter is automatically incremented when data is read from or written to Port #0, you may continually access blocks of data.

* To access the VRAM, you can also use commands. These commands will be explained in a later chapter.
* Refer to the data sheet for access timings.

REGISTER FUNCTIONS

1. CONTROL REGISTERS #0 to #23 (Write only)
#32 to #46 (Write only)

1.1 Mode Registers

	MSB 7	6	5	4	3	2	1	0 LSB	
R#0	0	DG	IE2	IE1	M5	M4	M3	0	Mode Register 0
R#1	0	BL	IE0	M1	M2	0	SI	MAG	Mode Register 1
R#8	MS	LP	TP	CB	VR	0	SPD	BW	Mode Register 2
R#9	LN	0	S1	S0	IL	E0	*NT	DC	Mode Register 3

* Indicates negative logic.

R#0
- DG : Sets the color bus to input mode, and inputs data into the VRAM.
- IE2: Enables interrupt from Lightpen by Interrupt Enable 2.
- IE1: Enables interrupt from Horizontal scanning line by Interrupt Enable 1.
- M5 : Used to change the display mode.
- M4 : Used to change the display mode.
- M3 : Used to change the display mode.

R#1
- BL : When 1, screen display enabled. When 0, screen disabled.
- IE0: Enables interrupt from Horizontal scanning line by Interrupt Enable 0.
- M1 : Used to change the display mode.
- M2 : Used to change the display mode.
- SI : When 1, sprite size is 16 x 16. When 0, 8 x 8.
- MA : Sprite expansion; when 1: expanded. When 0, normal.

R#8
- MS : When 1, sets the color bus to input mode and enables mouse.
  When 0, sets the color bus to output mode and disables mouse.
- LP : When 1, enables light pen. When 0, disables light pen.
- TP : Sets the color of code 0 to the color of the palette.
- CB : When 1, sets the color bus to input mode.
  When 0, sets the color bus to output mode.
- VR : Selects the type of Video RAM.
  1 = 64K x 1 bit or 64K x 4 bits.
  0 = 16K x 1 bit or 16K x 4 bits.
- SPD: When 1, disables display of sprite. When 0, displays sprite.
- BW : When 1, sets black and white in 32 tones.
  When 0, sets color (available only with a composite encoder).

R#9
- LN : When 1, sets the horizontal dot count to 212.
  When 0, sets the horizontal dot count to 192.
- S1 : Selects simultaneous mode.
- S0 : Selects simultaneous mode.

IL : When 1, interlace (Complete NTSC timing)
When 0, non-interlace (Incomplete NTSC timing)
EO : When 1, displays two graphic screens interchangably by Even field/Odd field.
When 0, displays the same graphic screen by Even field/Odd field.
*NT : When 1, PAL (313 lines); when 0, NTSC (262 lines). (For RGB output only)
DC : When 1, sets *DLCLK to input mode; when 0, sets *DLCLK to output mode.

1.2 Table Base Address Registers

The table base address registers are a set of registers to declare the addresses of tables in the VRAM to be used by MSX-VIDEO.

Note that when these registers are accessed, the control codes that the screen may receive depends on the display mode. For this purpose, you must mask the unwanted bits.

	MSB 7	6	5	4	3	2	1	0 LSB	
R#2	0	A16	A15	A14	A13	A12	A11	A10	Pattern name table base address register
R#3	A13	A12	A11	A10	A9	A8	A7	A6	Color table base address register low
R#10	0	0	0	0	0	A16	A15	A14	Color table base address register high
R#4	0	0	A16	A15	A14	A13	A12	A11	Pattern generator table base address register
R#5	A14	A13	A12	A11	A10	A9	A8	A7	Sprite attribute table base address register low
R#11	0	0	0	0	0	0	A16	A15	Sprite attribute table base address register high
R#6	0	0	A16	A15	A14	A13	A12	A11	Sprite pattern generator table base address register

1.3 Color Registers

The color registers are used to control the MSX-VIDEO's text and background screen colors as well as blinking, etc.

	MSB 7	6	5	4	3	2	1	0 LSB	
R#7	TC3	TC2	TC1	TC0	BD3	BD2	BD1	BD0	Text color/Back drop color register

TC3 to TC0: Specifies the text color according to TEXT 1 and TEXT 2 modes.
BD3 to BD0: Specifies the back drop color in all display modes.

	MSB 7	6	5	4	3	2	1	0 LSB	
R#12	T23	T22	T21	T20	BC3	BC2	BC1	BC0	Text color/Back color register

In TEXT 2 mode, if the attributes for blinking are set, the color set in this register and set in R#7 are displayed alternately.

T23 to T20: Specifies the color of part 1 of the pattern.
BC3 to BC0: Specifies the color of part 0 of the pattern.

	MSB 7	6	5	4	3	2	1	0 LSB	
R#13	ON3	ON2	ON1	ON0	OF3	OF2	OF1	OF0	Blinking period register

In the bit map modes of GRAPH4 to GRAPH7, the two pages are alternately displayed (blinked). Place data in this register to set the display page to an odd page to begin blinking. This register is also used in the TEXT2 mode.

ON3 to ON0: Display time for even page
OF3 to OF0: Display time for odd page

	MSB 7	6	5	4	3	2	1	0 LSB	
R#20	0	0	0	0	0	0	0	0	Color burst register 1
R#21	0	0	1	1	1	0	1	1	Color burst register 2
R#22	0	0	0	0	0	1	0	1	Color burst register 3

The above values are preset when the power is applied. If all values in the above three registers are set to 0, the color burst signal of the composite video output will be erased.

If the above values are subsequently reset to the preset values, the normal color burst signal will be output.

1.4 Display Registers

The display registers are used to control the display position on the CRT.

	MSB 7	6	5	4	3	2	1	0 LSB	
R#18	V3	V2	V1	V0	H3	H2	H1	H0	Display adjust register

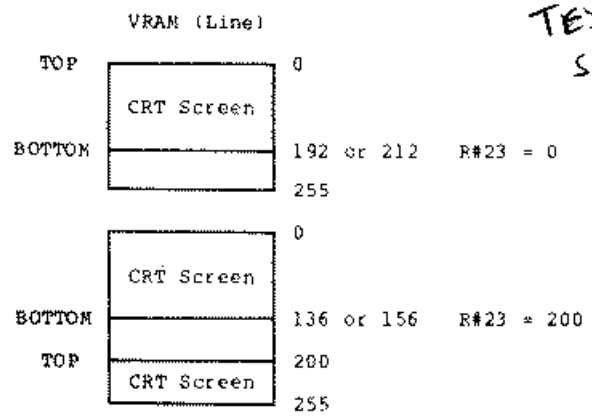
The above register is used to adjust the display position on the CRT.

H = 7 . . . H = 1, H = 0, H = 15 . . . H = 8
(Left) (Center) (Right)

V = 8 . . . V = 15, V = 0, V = 1 . . . V = 7
(Bottom) (Center) (Top)

	MSB 7	6	5	4	3	2	1	0 LSB	
R#23	DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0	Display offset register

The above register sets the location of the line to begin display.



TEXT 1 k2 !
scrol within page.

	MSB 7	6	5	4	3	2	1	0 LSB	
R#19	IL7	IL6	IL5	IL4	IL3	IL2	IL1	IL0	Interrupt line register

You may specify interrupts when the MSX-VIDEO begins to display a specified scanning line. To enable the interrupt, use the above register to set the scanning line.

1.5 ACCESS REGISTERS

The access registers are a set of registers used when accessing the MSX-VIDEO registers or the VRAM.

	MSB 7	6	5	4	3	2	1	0 LSB	
R#14	0	0	0	0	0	A16	A15	A14	VRAM Access base address register

When accessing the MSX-VIDEO and the Video RAM (VRAM), set the high-order three bits of the address in the VRAM access base address register.

When data is set in this register, and the VRAM is accessed, if there is a carry from A13, the data in the register is automatically incremented. In GRAPHIC1, GRAPHIC2, MULTICOLOR, and TEXT1 modes, the data in the register is not automatically incremented.

R#15	0	0	0	0	S3	S2	S1	S0	Status register pointer

When reading the MSX-VIDEO status registers (S#0 to S#9), set the contents of the Status register pointer.

	MSB 7	6	5	4	3	2	1	0 LSB	
R#16	0	0	0	0	C3	C2	C1	C0	Color palette address register

When setting the color palette of the MSX-VIDEO, set the number of the palette in the Color palette address register.

	MSB 7	6	5	4	3	2	1	0 LSB	
R#17	AII	0	RS5	RS4	RS3	RS2	RS1	RS0	Control register pointer

In the MSX-VIDEO, the above control register pointer may be used to access another register. In addition, according to the setting of the AII bit, the data can be automatically incremented.

AII = 1: Auto increment disabled
AII = 0: Auto increment enabled

1.6 COMMAND REGISTERS

The following command registers are used when executing a command on the MSX-VIDEO. Details on the use of these command registers will be presented in a later chapter.

	MSB 7	6	5	4	3	2	1	0 LSB	
R#32	SX7	SX6	SX5	SX4	SX3	SX2	SX1	SX0	Source X low register
R#33	0	0	0	0	0	0	0	SX8	Source X high register
R#34	SY7	SY6	SY5	SY4	SY3	SY2	SY1	SY0	Source Y low register
R#35	0	0	0	0	0	0	SY9	SY8	Source Y high register
R#36	DX7	DX6	DX5	DX4	DX3	DX2	DX1	DX0	Destination X low register
R#37	0	0	0	0	0	0	0	DX8	Destination X high register
R#38	DY7	DY6	DY5	DY4	DY3	DY2	DY1	DY0	Destination Y low register
R#39	0	0	0	0	0	0	DY9	DY8	Destination Y high register
R#40	NX7	NX6	NX5	NX4	NX3	NX2	NX1	NX0	Number of dots X low register
R#41	0	0	0	0	0	0	0	NX8	Number of dots X high register
R#42	NY7	NY6	NY5	NY4	NY3	NY2	NY1	NY0	Number of dots Y low register
R#43	0	0	0	0	0	0	NY9	NY8	Number of dots Y high register
R#44	CH3	CH2	CH1	CH0	CL3	CL2	CL1	CL0	Color register
R#45	0	MXC	MXD	MXS	DIY	DIX	EQ	MAJ	Argument register
R#46	CM3	CM2	CM1	CM0	LO3	LO2	LO1	LO0	Command register

2. STATUS REGISTERS #0 to #9 (Read only)

The following status registers are read-only registers for reporting the status when the MSX-VIDEO is read.

	MSB 7	6	5	4	3	2	1	0 LSB	
S#0	F	5S	C	Fifth sprite number					Status register 0

F: Vertical scanning interrupt flag
When S#0 is read, this flag is reset.

5S: Flag for the fifth sprite
Five sprites are aligned on the first horizontal line (In the G3 to G7 modes, 9 sprites are allowed)

C: Collision flag
Two sprites have collided.

Fifth sprite number:
The number of the fifth (or ninth) sprite.

	MSB 7	6	5	4	3	2	1	0 LSB	
S#1	FL	LPS	Identification #					FH	Status register 1

FL: Lightpen flag (Lightpen flag set)
If the lightpen is to detect light, this bit as well as the IE2 bit must be both set in order for an interrupt to be enabled. When S#1 is read, FL is reset.

Mouse switch 2 (Mouse flag set)
The second switch on the mouse was pressed.
In this case, when S#1 is read, FL is not reset.

LPS: Lightpen switch (Lightpen flag set)
The lightpen switch was pressed.
In this case, when S#1 is read, LPS is not reset.

Mouse switch 1 (Mouse flag set)
The first switch on the mouse was pressed.
In this case, when S#1 is read, LPS is not reset.

Identification number:
The identification number (ID #) of the MSX-VIDEO. 0 - V9938 2 - V9958

FH: Horizontal scanning interrupt flag
Horizontal scanning interrupt (which is specified in R#19) flag. If IE1 is set, an interrupt is enabled. When S#1 is read, FH is reset.

	MSB 7	6	5	4	3	2	1	0 LSB	
S#2	TR	VR	HR	BD	1	1	EO	CE	Status register 2

TR: Transfer ready flag
When the CPU sends commands to the VRAM and other devices, the CPU checks this flag while transferring data. When this flag is set to 1, transfer may be done.

Set during Vblank on Border

VR: Vertical scanning line timing flag
During vertical scanning, this flag is set to 1.

HR: Horizontal scanning line timing flag
During horizontal scanning, this flag is set to 1.

Set during Hblank

BD: Boundary color detect flag
When the search command is executed, this flag detects whether the boundary color was detected or not.

EO: Display field flag
When 0, indicates the first field.
When 1, indicates the second field.

CE: Command execution flag
Indicates that a command is being executed.

	MSB 7	6	5	4	3	2	1	0 LSB	
S#3	X7	X6	X5	X4	X3	X2	X1	X0	Column register low
S#4	1	1	1	1	1	1	1	X8	Column register high
S#5	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	Row register low
S#6	1	1	1	1	1	1	EO	Y8	Row register high

The above registers are set to indicate the collision location of sprites, the location of lightpen detection, and the relative movement of the mouse.

S#7	C7	C6	C5	C4	C3	C2	C1	C0	Color register

The above color register is used when the POINT and VRAM to CPU commands are executed. The VRAM data is set in this register.

S#8	BX7	BX6	BX5	BX4	BX3	BX2	BX1	BX0	Border X register low
S#9	1	1	1	1	1	1	1	BX8	Border X register high

When the search command is executed and the border color has been detected, the X coordinate is set in the above registers.

TEXT 1 MODE

Characteristics

- Pattern size : 6 dots (w) x 8 dots (h)
- Patterns : 256 types
- Screen pattern count : 40 (w) x 24 (h) patterns
- Pattern colors : Two colors out of 512 colors (per screen)
- VRAM area per screen : 4K bytes

Controls

- Pattern font : VRAM pattern generator table
- Screen pattern location : VRAM pattern name table
- Pattern color code 1 : High-order four bits of R#7
- Pattern color code 0 : Low-order four bits of R#7
- Background color code : Low-order four bits of R#7

Initial Settings

1. Mode and Register Settings

	MSB 7	6	5	4	3	2	1	0 LSB	
R#0	0	DG	IE2	IE1	0*	0*	0*	0	Mode register 0
R#1	0	BL	IE0	1*	0*	0	SI	MAG	Mode register 1
R#8	MS	LP	TP	CB	VR	0	SPD	BW	Mode register 2
R#9	LN	0	S1	S0	IL	EO	**NT	DC	Mode register 3

* Examples of settings in TEXT 1 mode
** Indicates negative logic
All other bits are set accordingly

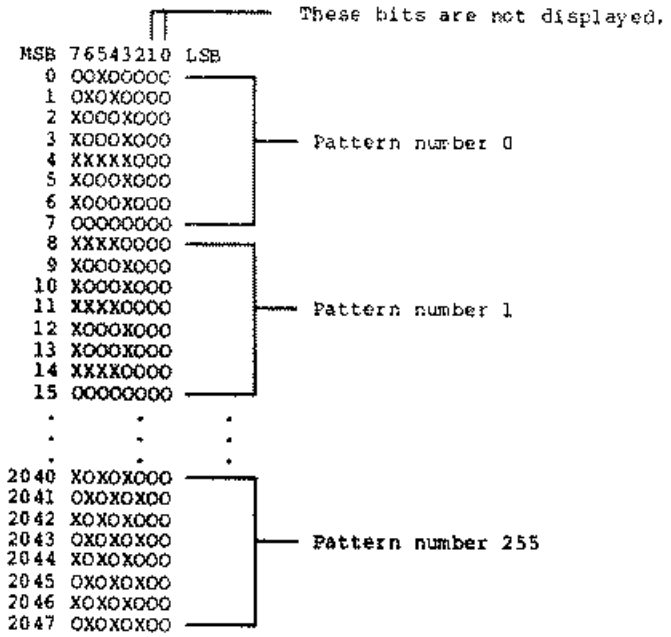
2. Pattern Generator Table Settings

- The pattern generator table is an area that stores the pattern fonts.
- Each pattern has a number from PN0 to PN255.
- The font for each pattern is constructed from 8 bytes, and the lower two bits of each of the eight bytes is not displayed.
- Set the beginning (head) address of the pattern generator table in register R#4.

	MSB 7	6	5	4	3	2	1	0 LSB	
R#4	0	0	A16	A15	A14	A13	A12	A11	Pattern generator table base address register

Pattern generator table

(X=1, O=0)

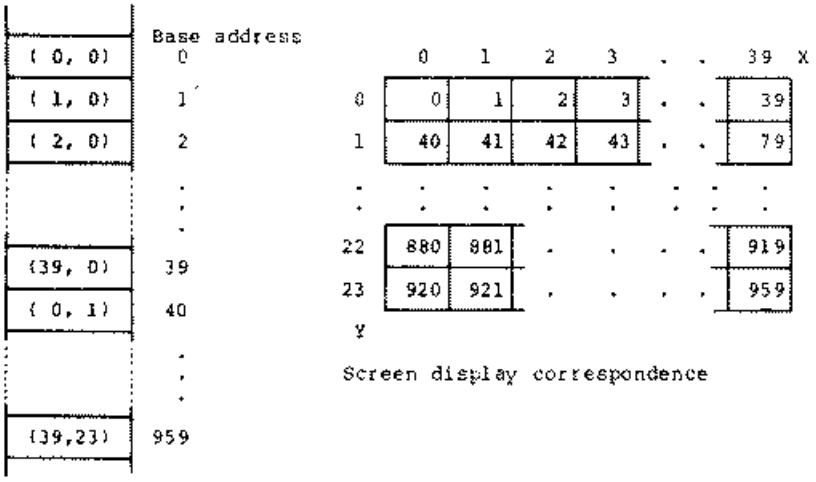


3. Pattern name table settings

- The pattern name table is composed of one byte for each screen pattern. Each byte specifies a unique pattern.
- Set the beginning (head) address of the pattern name table in register R#2.

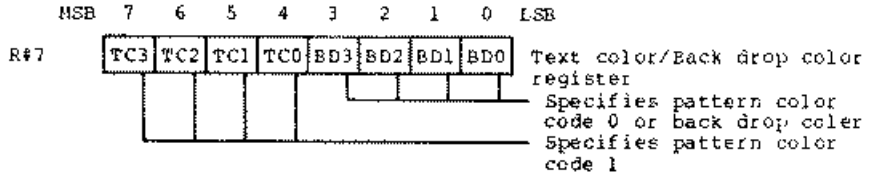
	MSB 7	6	5	4	3	2	1	0 LSB	
R#2	0	A16	A15	A14	A13	A12	A11	A10	Pattern name table base address register

Pattern name table

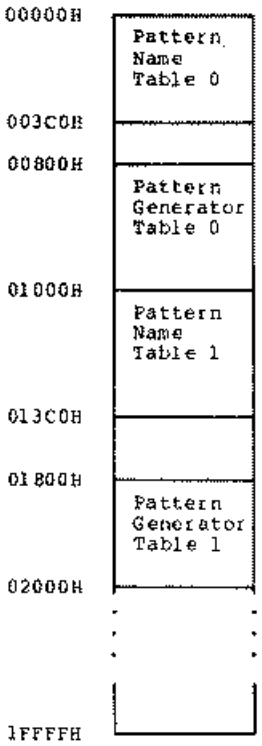


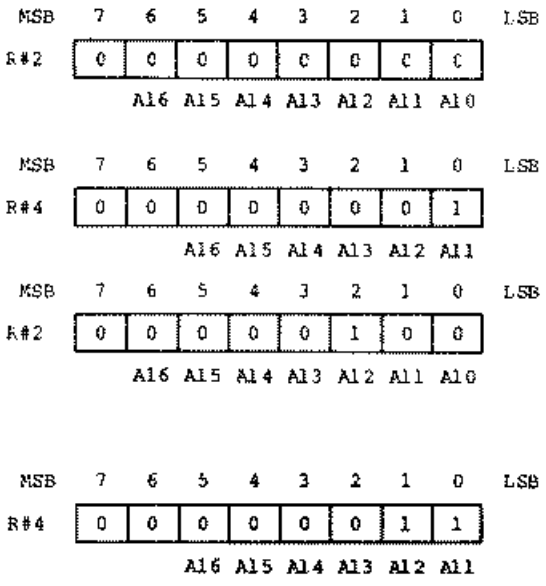
Screen display correspondence

4. Color register settings



Example of VRAM allocation in TEXT 1 mode





A maximum of 32 pages may be allocated in the same manner (using a 128K-byte VRAM).

TEXT 2 MODE

Characteristics

- Pattern size : 6 dots (w) x 8 dots (h)
- Patterns : 256 types
- Screen pattern count : 80 (w) x 24 (h) patterns
  80 (w) x 26.5 (h) patterns
- Pattern blinking : Possible for each character
- Pattern colors : Two colors out of 512 colors (per screen), four if usirg blinking
- VRAM area per screen : 8K bytes

Controls

- Pattern font : VRAM pattern generator table
- Screen pattern location : VRAM pattern name table
- Blink attributes : VRAM color table
- Pattern color code 1 : High-order four bits of R#7
- Pattern color code 0 : Low-order four bits of R#7
- Background color code : Low-order four bits of R#7
- Pattern color code 1 : High-order four bits of R#12 (Used for blinking)
- Pattern color code 0 : Low-order four bits of R#12 (Used for blinking)

Initial Settings

1. Mode and Register Settings

	MSB 7	6	5	4	3	2	1	0 LSB	
R#0	0	DG	IE2	IE1	0*	1*	0*	0	Mode register 0
R#1	0	BL	IE0	1*	0*	0	SI	MAG	Mode register 1
R#8	MS	LP	TP	CB	VR	0	SPD	BW	Mode register 2
R#9	LN	0	S1	S0	IL	EO	**NT	DC	Mode register 3

* Examples of settings in TEXT 2 mode
** Indicates negative logic
In this display mode, if LN is set to 1, 26.5 lines are selected, and if LN is set to 0, 24 lines are selected.
All other bits are set accordingly

2. Pattern Generator Table Settings

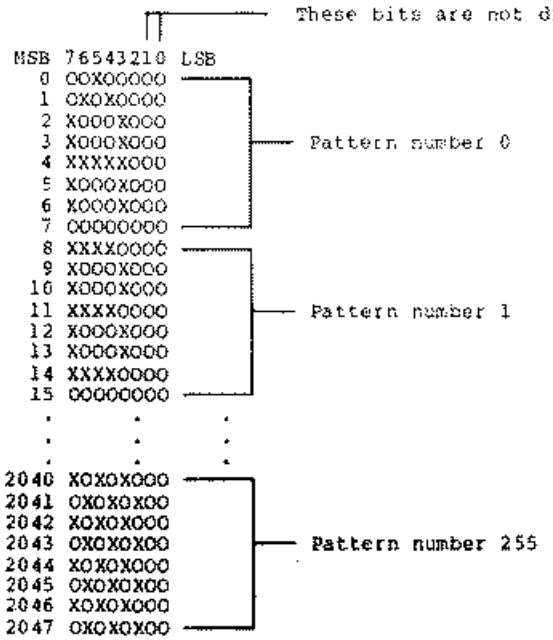
- The pattern generator table is an area that stores the pattern fonts.
- Each pattern has a number from PN0 to PN255.
- Set the beginning (head) address of the pattern generator table in register R#4.

	MSB 7	6	5	4	3	2	1	0 LSB	
R#4	0	0	A16	A15	A14	A13	A12	A11	Pattern generator table base address register

- The font for each pattern is constructed from 8 bytes, and the lower two bits of each of the eight bytes is not displayed.

Pattern generator table

(X=1, O=0)

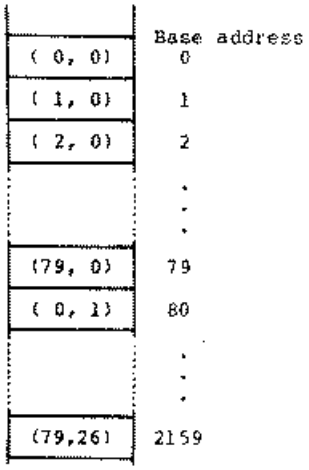


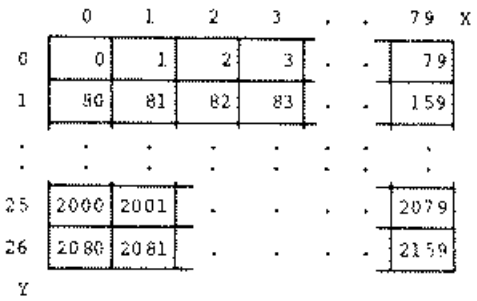
3. Pattern name table settings

- The pattern name table is composed of one byte for each screen pattern. Each byte specifies a unique pattern.
- If LN is set to 0, the screen display pattern is 80 (w) x 24 (h); and if LN is set to 1, the screen display pattern is 80 (w) x 26.5 (h). The upper half of the 27th pattern (h) is displayed.
- Set the beginning (head) address of the pattern name table in register R#2.

	MSB 7	6	5	4	3	2	1	0 LSB	
R#2	0	A16	A15	A14	A13	A12	1	1	Pattern name table base address register

Pattern name table





Screen display correspondence

4. Color table settings

- In TEXT 2 mode, each pattern has a separate bit for the attribute area, and if this bit is set to 1, the pattern blink attribute will be set.
- Set the beginning (head) address of the color table in registers R#3 and R#10.

	MSB 7	6	5	4	3	2	1	0 LSB	
R#3	A13	A12	A11	A10	A9	1	1	1	Color table
R#10	0	0	0	0	0	A16	A15	A14	base address registers

COLOR TABLE

MSB	7	6	5	4	3	2	1	0 LSB	
0	(0, 0)	(1, 0)	(2, 0)	(3, 0)	(4, 0)	(5, 0)	(6, 0)	(7, 0)	Base address
1	(8, 0)	(9, 0)	(10, 0)	(11, 0)	(12, 0)	(13, 0)	(14, 0)	(15, 0)	
269	(72,26)	.	.	.	.	.	.	(79,26)	

5. Color register settings

- Set the color for pattern 1 in the high-order bits of register R#7.
- Set the color for pattern 0 in the low-order bits of register R#7.

	MSB 7	6	5	4	3	2	1	0	LSB	
R#7	TC3	TC2	TC1	TC0	BD3	BD2	BD1	BD0		Text color/Back drop color register

- Set the blink attribute for the corresponding pattern by setting an alternate color code in register R#12. The pattern will be blinked by using the color codes set in registers R#7 and R#12.

	MSB 7	6	5	4	3	2	1	0	LSB	
R#12	T23	T22	T21	T20	BC3	BC2	BC1	BC0		Text color/Back color register

6. Blink register settings

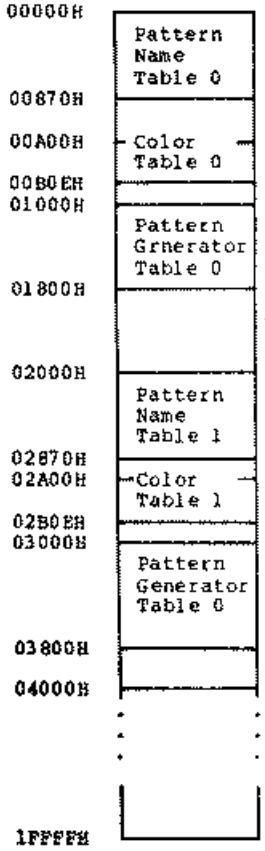
- The color codes set in registers R#7 and R#12 will be alternately displayed for blinking; however, the blinking period attribute (time on and time off) can also be set in register R#13.

	MSB 7	6	5	4	3	2	1	0	LSB	
R#13	ON3	ON2	ON1	ON0	OF3	OF2	OF1	OF0		Blinking period register
	ON time				OFF time					

- The data for the ON and OFF times are shown below.(NTSC)

DATA (Binary)	TIME (ms)
0 0 0 0	0
0 0 0 1	166.9
0 0 1 0	333.8
0 0 1 1	500.6
0 1 0 0	667.5
0 1 0 1	834.4
0 1 1 0	1001.3
0 1 1 1	1168.2
1 0 0 0	1335.1
1 0 0 1	1501.9
1 0 1 0	1668.8
1 0 1 1	1835.7
1 1 0 0	2002.6
1 1 0 1	2169.5
1 1 1 0	2336.3
1 1 1 1	2503.2

Example of VRAM allocation in TEXT 2 mode



Pattern name table base address

MSB	7	6	5	4	3	2	1	0	LSB
R#2	0	0	0	0	0	0	1	1	

A16 A15 A14 A13 A12

Color table base address

MSB	7	6	5	4	3	2	1	0	LSB
R#3	0	0	1	0	1	1	1	1	

A13 A12 A11 A10 A9

R#10	0	0	0	0	0	0	0	0	

A16 A15 A14

Pattern generator table base address

MSB	7	6	5	4	3	2	1	0	LSB
R#4	0	0	0	0	0	0	1	0	

A16 A15 A14 A13 A12 A11

A maximum of 16 pages may be allocated in the same manner (using a 128K-byte VRAM).

MULTICOLOR MODE

Characteristics

- Screen composition : 64 (w) x 48 (h) color blocks
- Color blocks : Sixteen colors out of 512 colors
- Sprite mode : Sprite mode 1
- VRAM area per screen : 4K bytes

Controls

- Color block color code : VRAM pattern generator table
- Color block location : VRAM pattern name table
- Background color code : Low-order four bits of R#7
- Sprites : VRAM sprite attribute table
  VRAM sprite pattern table

Initial Settings

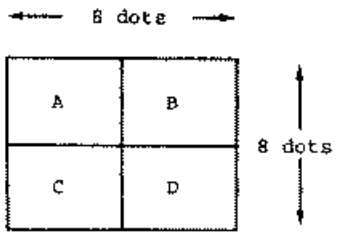
1. Mode and Register Settings

	MSB 7	6	5	4	3	2	1	0 LSB	
R#0	0	DG	IE2	IE1	0*	0*	0*	0	Mode register 0
R#1	0	BL	IE0	0*	1*	0	SI	MAG	Mode register 1
R#8	MS	LP	TP	CB	VR	0	SPD	BW	Mode register 2
R#9	LN	0	S1	S0	IL	EO	**NT	DC	Mode register 3

* Examples of settings the display mode to the MULTICOLOR mode
** Indicates negative logic

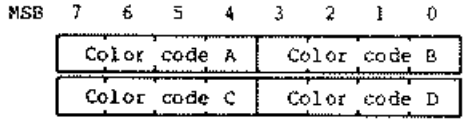
2. Pattern Generator Table Settings

- The pattern generator table is an area that stores the colors of the color blocks.
- Each pattern is made up of four color blocks. These patterns are approximately 8 x 8 when the dots available for the screen display area is 256 x 192 dots.

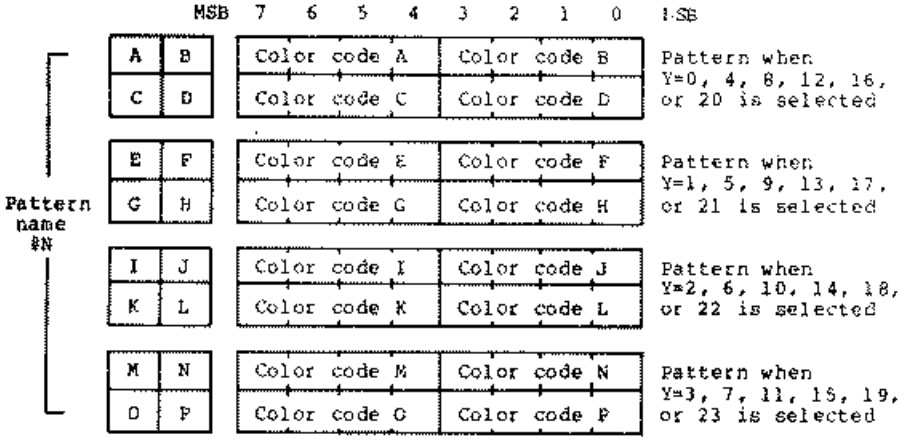


For each block A, B, C, and D, sixteen colors may be specified.

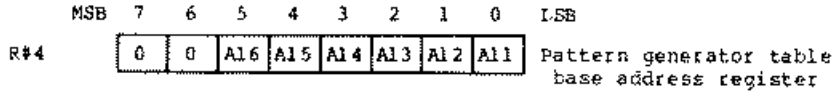
- In the MULTICOLOR mode, two bytes are used for each pattern, and each pattern includes four color blocks.



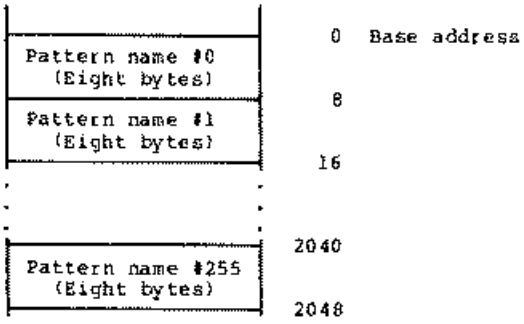
- In the MULTICOLOR mode, for each pattern name, there are four corresponding color blocks, and according to the y-coordinate, the pattern names are automatically set.



- Set the beginning (head) address of the pattern generator table in register R#4.



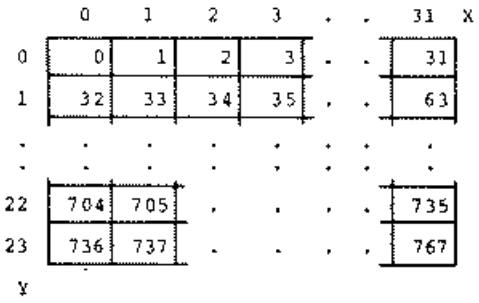
Pattern generator table



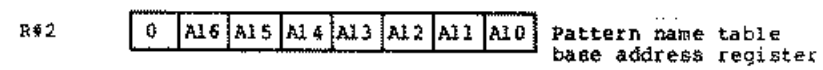
3. Pattern name table settings

- The pattern name table is composed of one byte for each screen pattern. Each byte specifies a unique pattern number.

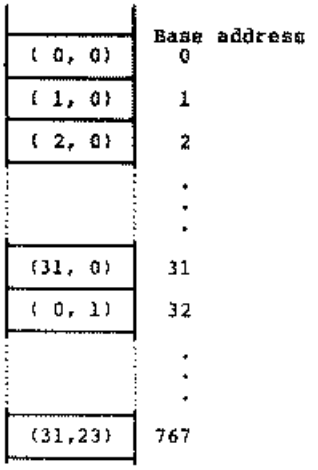
Pattern name table



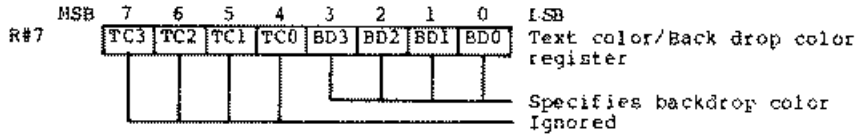
- Set the beginning (head) address of the pattern name table in register R#2.



Pattern name table

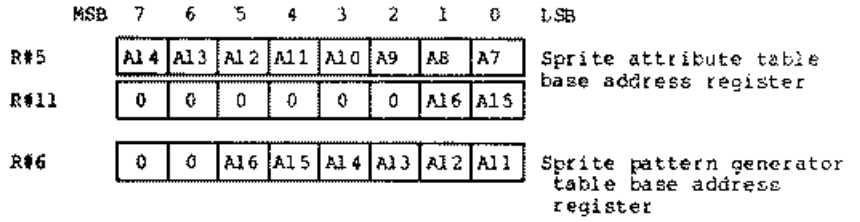


4. Color register settings

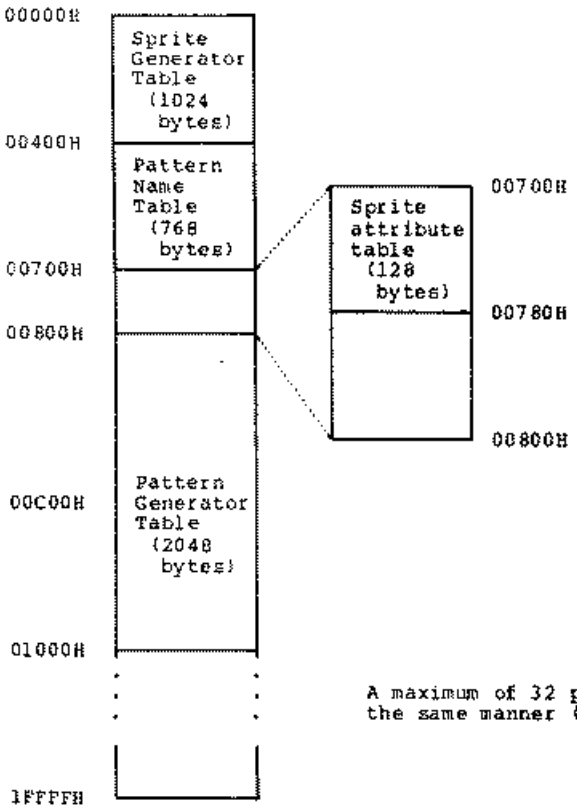


5. Sprite settings

- Set the beginning (head) address of the sprite attribute table in registers R#5 and R#11; and set the beginning (head) address of the sprite pattern generator table in register R#6. For details about sprites, see the section on SPRITE MODE 1.



Example of VRAM allocation in MULTICOLOR mode



A maximum of 32 pages may be allocated in the same manner (using a 128K-byte VRAM).

GRAPHIC 1 MODE

Characteristics

- Pattern size : 8 dots (w) x 8 dots (h)
- Patterns : 256 types
- Screen pattern count : 32 (w) x 24 (h) patterns
- Pattern colors : 16 colors out of 512 colors (per screen)
- Sprite mode : Sprite mode 1
- VRAM area per screen : 4K bytes

Controls

- Pattern font : VRAM pattern generator table
- Screen pattern location : VRAM pattern name table
- Pattern color codes 1 & 0 : Can be specified as a group for each 8-pattern set, in the VRAM color table
- Background color code : Low-order four bits of R#7
- Sprites : VRAM sprite attribute table, VRAM sprite pattern table

Initial Settings

1. Mode and Register Settings

	MSB 7	6	5	4	3	2	1	0 LSB	
R#0	0	DG	IE2	IE1	0*	0*	0*	0	Mode register 0
R#1	0	BL	IE0	0*	0*	0	SI	MAG	Mode register 1
R#8	MS	LP	TP	CB	VR	0	SPD	BW	Mode register 2
R#9	LN	0	S1	S0	IL	EO	**NT	DC	Mode register 3

* Examples of settings in GRAPHIC 1 mode
** Indicates negative logic

2. Pattern Generator Table Settings

- The pattern generator table is an area that stores the pattern fonts.
- Each pattern has a number from PN0 to PN255.
- The font for each pattern is constructed from 8 bytes.
- Set the beginning (head) address of the pattern generator table in register R#4.

	MSB 7	6	5	4	3	2	1	0 LSB	
R#4	0	0	A16	A15	A14	A13	A12	A11	Pattern generator table base address register

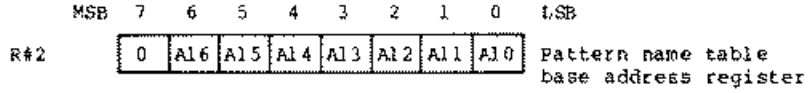
Pattern generator table

(X=1, O=0)

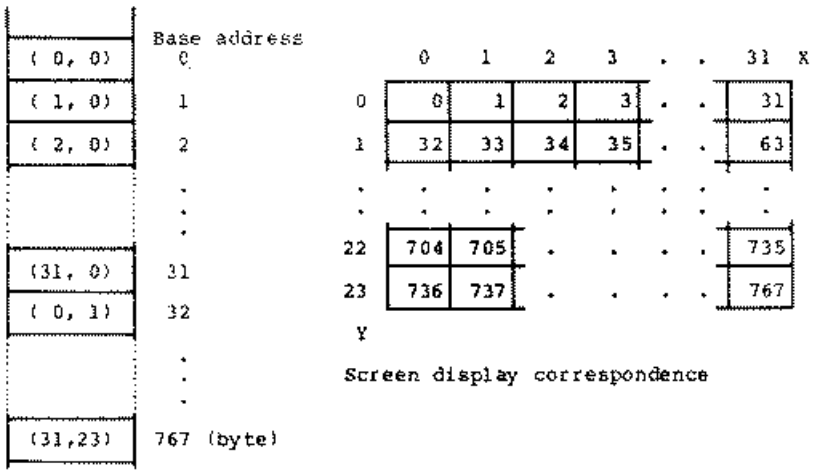
```
      MSB 76543210 LSB
        0 OOXXXOOO ------- Base address
        1 OXOOOXOO
        2 XOOOOOXO
        3 XOOOOOXO ------- Pattern number 0
        4 XXXXXXXO
        5 XOOOOOXO
        6 XOOOOOXO
        7 OOOOOOOO -------
        8 XXXXXXOO -------
        9 XOOOOOXO
       10 XOOOOOXO
       11 XXXXXXOO ------- Pattern number 1
       12 XOOOOOXO
       13 XOOOOOXO
       14 XXXXXXOO
       15 OOOOOOOO -------
        .     .     .
        .     .     .
        .     .     .
     2040 XOXOXOXO -------
     2041 OXOXOXOX
     2042 XOXOXOXO
     2043 OXOXOXOX ------- Pattern number 255
     2044 XOXOXOXO
     2045 OXOXOXOX
     2046 XOXOXOXO
     2047 OXOXOXOX -------
```

3. Pattern name table settings

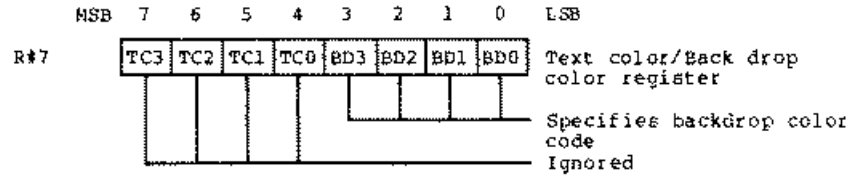
- The pattern name table is composed of one byte for each screen pattern. Each byte specifies a unique pattern.
- Set the beginning (head) address of the pattern name table in register R#2.



Pattern name table



4. Color register settings

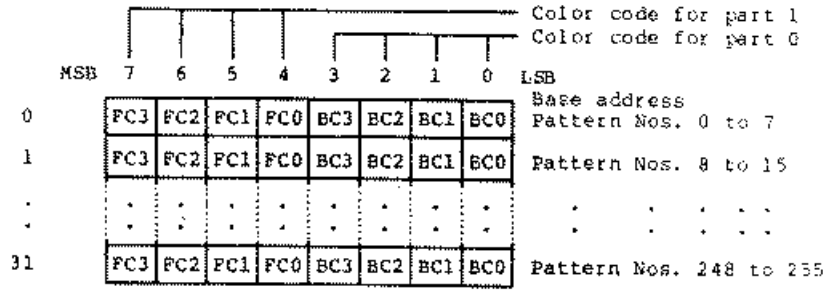


5. Color table settings

- The colors for pattern color 1 and pattern color 0 are set in groups of eight patterns.
- Set the beginning (head) address of the color table in registers R#3 and R#10.

	MSB 7	6	5	4	3	2	1	0 LSB	
R#3	A13	A12	A11	A10	A9	A8	A7	A6	Color table base address
R#10	0	0	0	0	0	A16	A15	A14	register

Color table

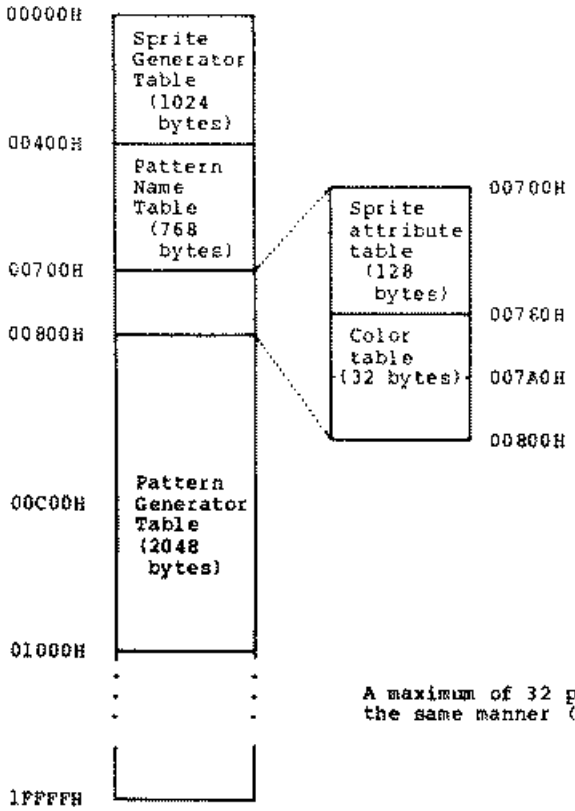


6. Sprite settings

- Set the beginning (head) address of the sprite attribute table in registers R#5 and R#11; and set the beginning (head) address of the sprite pattern generator table in register R#6. For details on sprites, see the section on SPRITE MODE 1.

	MSB 7	6	5	4	3	2	1	0 LSB	
R#5	A14	A13	A12	A11	A10	A9	A8	A7	Sprite attribute table
R#11	0	0	0	0	0	0	A16	A15	base address register
R#6	0	0	A16	A15	A14	A13	A12	A11	Sprite pattern generator table base address register

Example of VRAM allocation in GRAPHIC 1 mode



A maximum of 32 pages may be allocated in the same manner (using a 128K-byte VRAM).

GRAPHIC 2 AND GRAPHIC 3 MODES

Characteristics

- Pattern size : 8 dots (w) x 8 dots (h)
- Patterns : 768 types
- Screen pattern count : 32 (w) x 24 (h) patterns
- Pattern colors : 16 colors out of 512 colors (per screen)
- Sprite modes : Sprite mode 1 (GRAPHIC 2)
  Sprite mode 2 (GRAPHIC 3)
- VRAM area per screen : 16K bytes

* The GRAPHIC 2 and GRAPHIC 3 modes are identical except for the sprite modes.

Controls

- Pattern font : VRAM pattern generator table
- Screen pattern location : VRAM pattern name table
- Pattern color codes 1 & 0 : Can be specified as a group for each raster, in the VRAM color table
- Background color code : Low-order four bits of R#7
- Sprites : VRAM sprite attribute table, VRAM sprite pattern table

Initial Settings

1. Mode and Register Settings

	MSB 7	6	5	4	3	2	1	0 LSB	
R#0	0	EXG	IE2	IE1	0*	M*	M*	0	Mode register 0
R#1	0	BL	IE0	0*	0*	0	SI	MAG	Mode register 1
R#8	MS	LP	TP	CB	VR	0	SPD	BW	Mode register 2
R#9	LN	0	S1	S0	IL	EO	**NT	DC	Mode register 3

0, 1 for GRAPHIC 2 mode
1, 0 for GRAPHIC 3 mode

* Examples of settings in GRAPHIC 2 mode or GRAPHIC 3 mode
** Indicates negative logic
All other bits are set accordingly.

2. Pattern Generator Table Settings

- The pattern generator table is an area that stores the pattern fonts.
- Each pattern group has a number from PN0 to PN255; and since each group may have three members, 768 patterns may be specified.
- The font for each pattern is constructed from 8 bytes.
- Set the beginning (head) address of the pattern generator table in register R#4.

	MSB 7	6	5	4	3	2	1	0 LSB	
R#4	0	0	A16	A15	A14	A13	1	1	Pattern generator table base address register

3. Color table settings

- The colors for pattern color 1 and pattern color 0 are set as a group of one raster.
- The color table corresponds to the pattern generator table on a one-to-one basis.
- Set the beginning (head) address of the color table in registers R#3 and R#10.

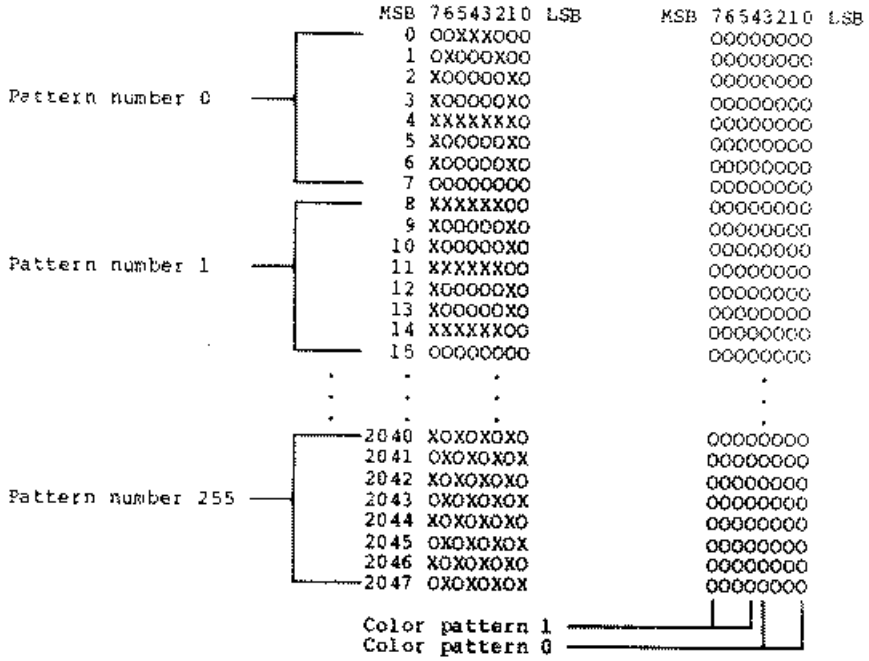
	MSB 7	6	5	4	3	2	1	0 LSB	
R#3	A13	1	1	1	1	1	1	1	Color table base address
R#10	0	0	0	0	0	A16	A15	A14	register

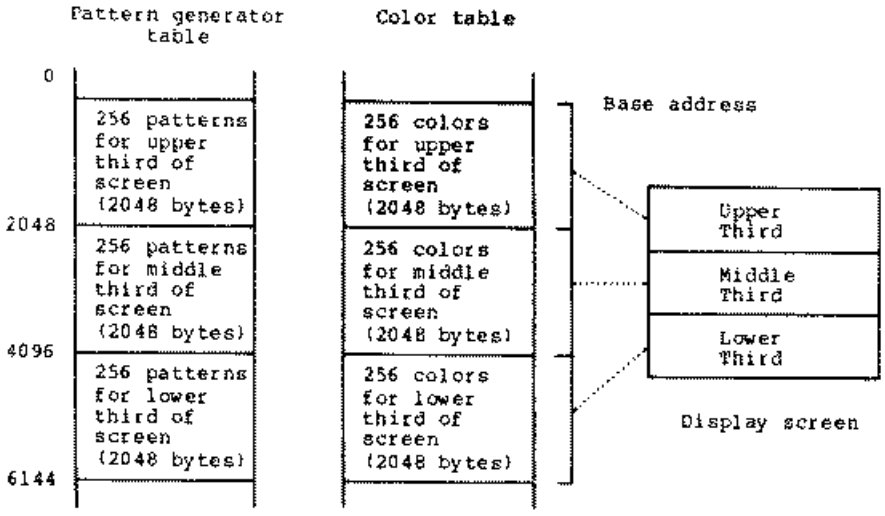
4. Color register settings

	MSB 7	6	5	4	3	2	1	0 LSB	
R#7	TC3	TC2	TC1	TC0	BD3	BD2	BD1	BD0	Text color/Back drop color register
									Specifies backdrop color code
									Ignored

Pattern generator table

(X=1, O=0)





3. Pattern name table settings

- The pattern name table is composed of one byte for each screen pattern. Each byte specifies a unique pattern.
- The upper, middle, and lower parts of the screen can be used as three different parts, for a total of 768 patterns.

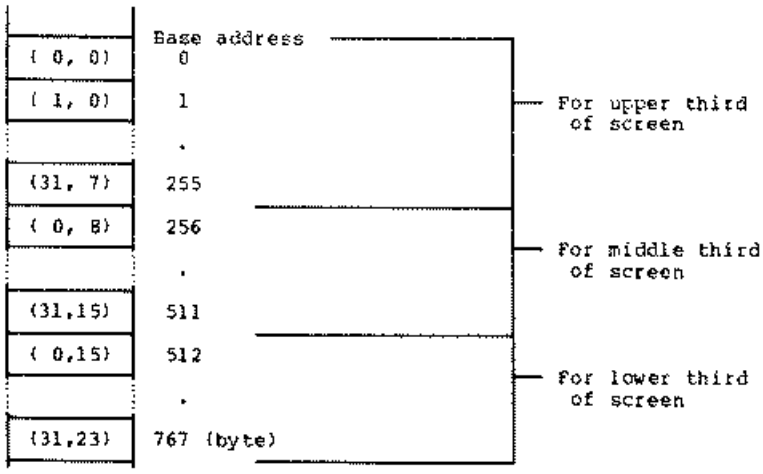
Pattern name table

(0, 0)	Pattern display area for upper third of screen (256 bytes)	(31, 0)
(0, 7)		(31, 7)
(0, 8)	Pattern display area for middle third of screen (256 bytes)	(31, 8)
(0,15)		(31,15)
(0,16)	Pattern display area for lower third of screen (256 bytes)	(31,16)
(0,23)		(31,23)

- Set the beginning (head) address of the pattern name table in register R#2.

	MSB 7	6	5	4	3	2	1	0 LSB	
R#2	0	A16	A15	A14	A13	A12	A11	A10	Pattern name table base address register

Pattern name table

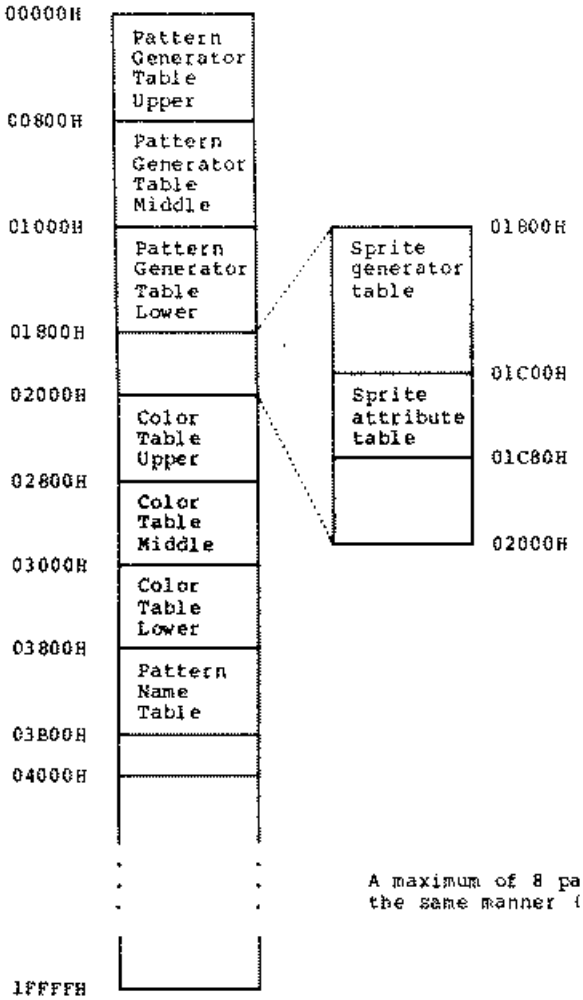


4. Sprite settings

- Set the beginning (head) address of the sprite attribute table in registers R#5 and R#11; and set the beginning (head) address of the sprite pattern generator table in register R#6. For details about sprites pertaining to GRAPHIC 2 mode, see the section on SPRITE MODE 1, and for details about sprites pertaining to GRAPHIC 3 mode, see the section on SPRITE MODE 2.

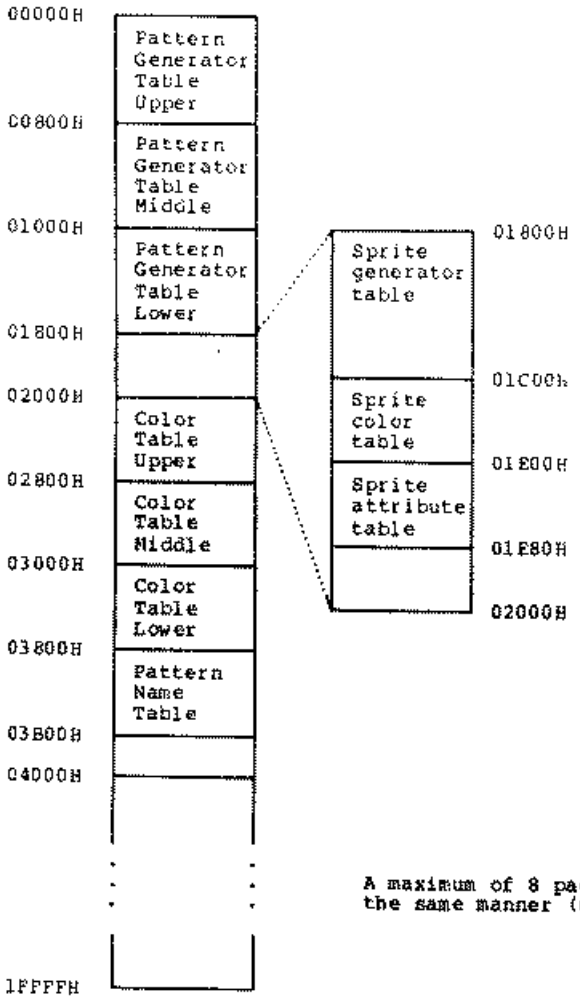
	MSB 7	6	5	4	3	2	1	0 LSB	
R#5	A14	A13	A12	A11	A10	A9	A8	A7	Sprite attribute table
R#11	0	0	0	0	0	0	A16	A15	base address register
R#6	0	0	A16	A15	A14	A13	A12	A11	Sprite pattern generator table base address register

Example of VRAM allocation in GRAPHIC 2 mode



A maximum of 8 pages may be allocated in the same manner (using a 128K-byte VRAM).

Example of VRAM allocation in GRAPHIC 3 mode



A maximum of 8 pages may be allocated in the same manner (using a 128K-byte VRAM).

GRAPHIC 4 MODE

Characteristics

- Bit-mapped Graphics Mode
- Screen size : 256 (w) x 212 (h) dots
  256 (w) x 192 (h) dots
- Screen colors : 16 colors out of 512 colors (per screen)
- Sprite mode : Sprite mode 2
- VRAM area per screen : 32K bytes

Controls

- Graphics : VRAM pattern name table
- Background color code : Low-order four bits of R#7
- Sprites : VRAM sprite attribute table, VRAM sprite pattern table

Initial Settings

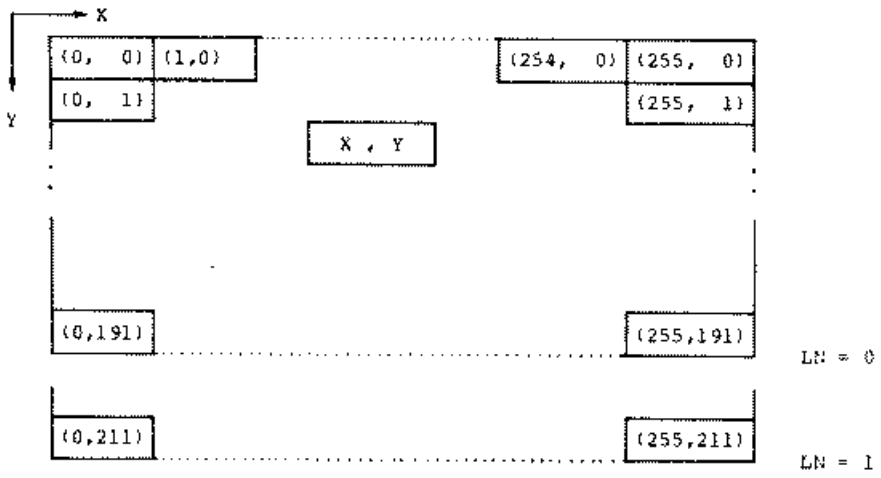
1. Mode and Register Settings

	MSB 7	6	5	4	3	2	1	0 LSB	
R#0	0	DG	IE2	IE1	0*	1*	1*	0	Mode register 0
R#1	0	BL	IE0	0*	0*	0	SI	MAG	Mode register 1
R#8	MS	LP	TP	CB	VR	0	SPD	BW	Mode register 2
R#9	LN	0	S1	S0	IL	EO	**NT	DC	Mode register 3

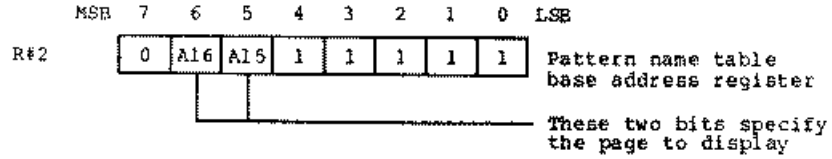
* Examples of settings in GRAPHIC 4 mode
** Indicates negative logic
In GRAPHIC 4 mode, if LN is set to 1, the screen height is 212 dots, and if LN is set to 0, the screen height is 192 dots.
All other bits are set accordingly.

2. Pattern name table settings

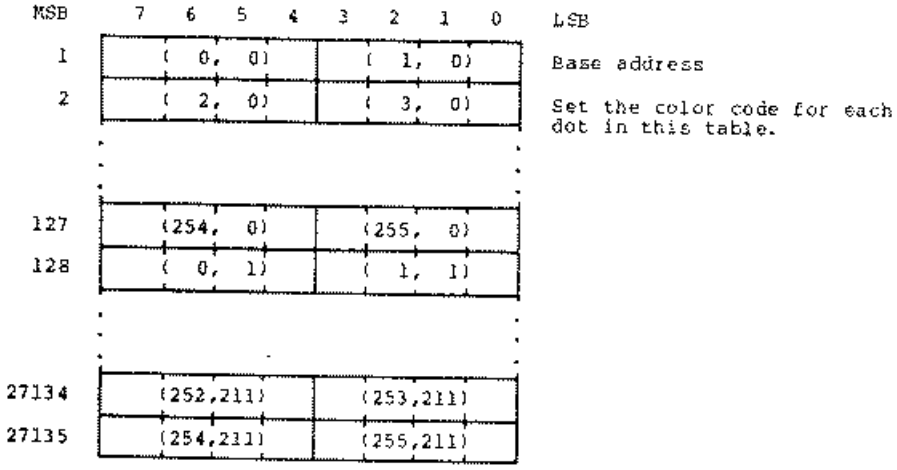
- The pattern name table is composed of one byte for every two dots on the screen. A color can be assigned for each dot from a selection of 16 colors out of 512 colors.



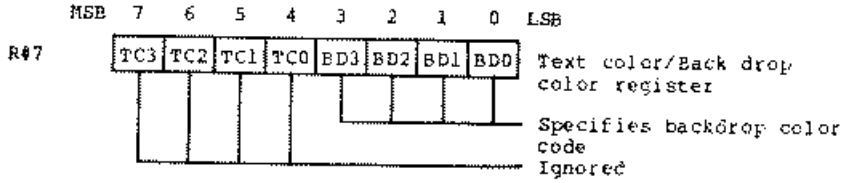
- Set the beginning (head) address of the pattern name table in register R#2.



Pattern name table



3. Color register settings

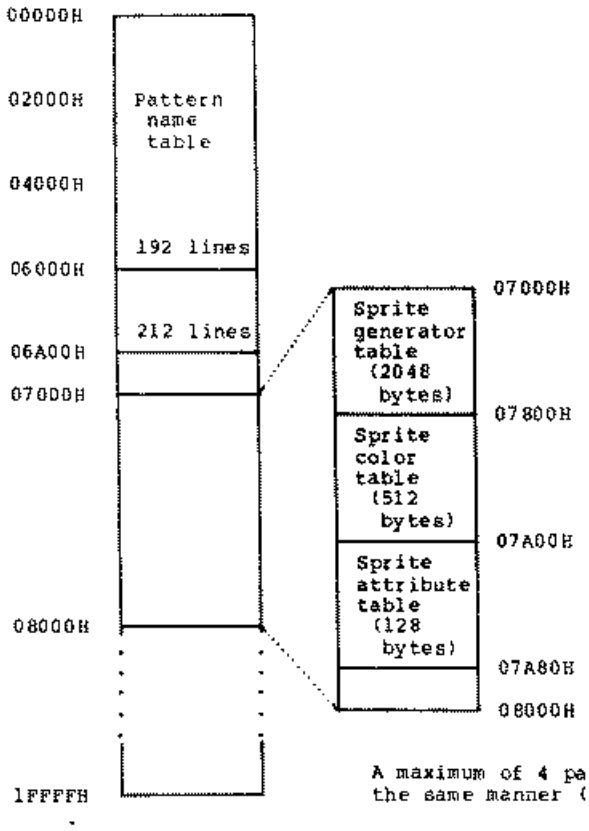


4. Sprite settings

- Set the beginning (head) address of the sprite attribute table in registers R#5 and R#11; and set the beginning (head) address of the sprite pattern generator table in register R#6. For details about sprites, see the section on SPRITE MODE 2.

	MSB 7	6	5	4	3	2	1	0 LSB	
R#5	A14	A13	A12	A11	A10	1	1	1	Sprite attribute table
R#11	0	0	0	0	0	0	A16	A15	base address register
R#6	0	0	A16	A15	A14	A13	A12	A11	Sprite pattern generator table base address register

Example of VRAM allocation in GRAPHIC 4 mode



A maximum of 4 pages may be allocated in the same manner (using a 128K-byte VRAM).

GRAPHIC 5 MODE

Characteristics

- Bit-mapped Graphics Mode
- Screen size : 512 (w) x 212 (h) dots
  512 (w) x 192 (h) dots
- Screen colors : 4 colors out of 512 colors (per screen)
- Sprite mode : Sprite mode 2
- VRAM area per screen : 32K bytes

Controls

- Graphics : VRAM pattern name table
- Background color code : Low-order four bits of R#7
- Sprites : VRAM sprite attribute table, VRAM sprite pattern table

Initial Settings

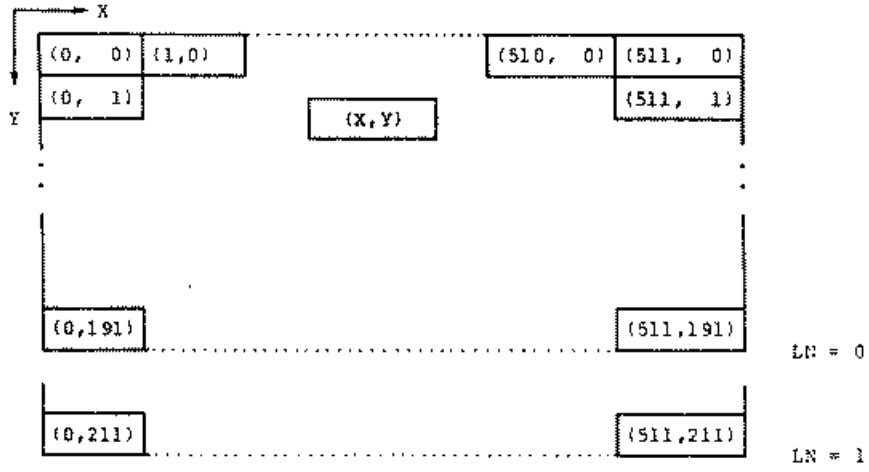
1. Mode and Register Settings

	MSB 7	6	5	4	3	2	1	0 LSB	
R#0	0	DG	IE2	IE1	1*	0*	0*	0	Mode register 0
R#1	0	BL	IE0	0*	0*	0	SI	MAG	Mode register 1
R#8	MS	LP	TP	CB	VR	0	SPD	BW	Mode register 2
R#9	LN	0	S1	S0	IL	EO	**NT	DC	Mode register 3

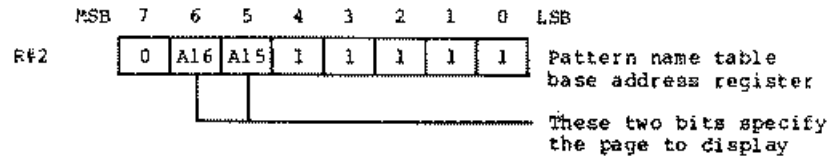
* Examples of settings in GRAPHIC 5 mode
** Indicates negative logic
In GRAPHIC 5 mode, if LN is set to 1, the screen height is 212 dots, and if LN is set to 0, the screen height is 192 dots.
All other bits are set accordingly.

2. Pattern name table settings

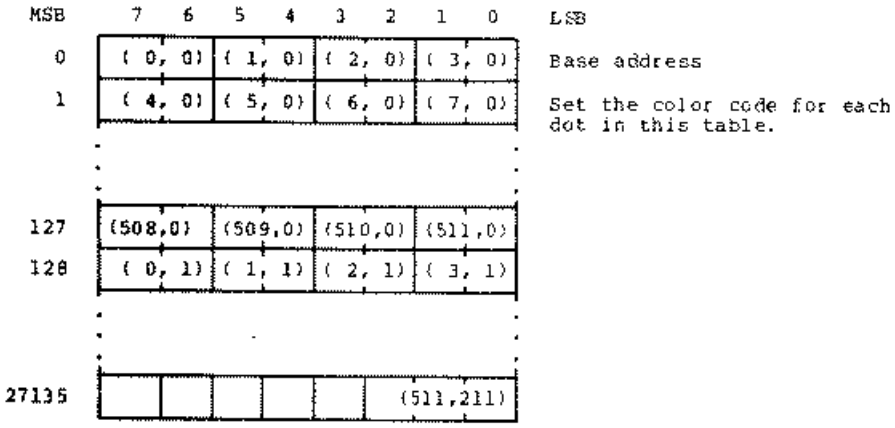
- The pattern name table is composed of one byte for every four dots on the screen. A color can be assigned for each dot from a selection of 4 colors out of 512 colors.



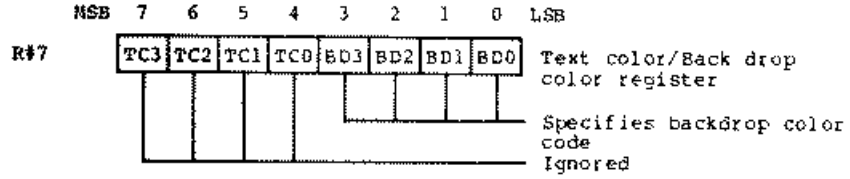
- Set the beginning (head) address of the pattern name table in register R#2.



Pattern name table

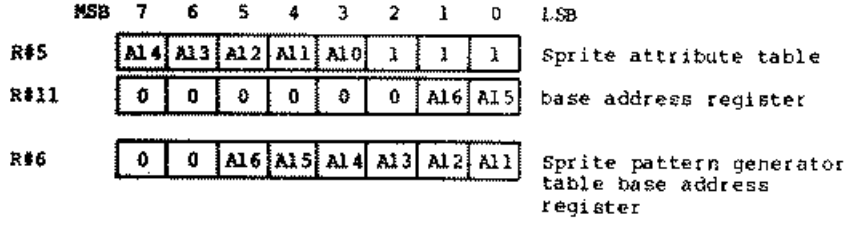


3. Color register settings



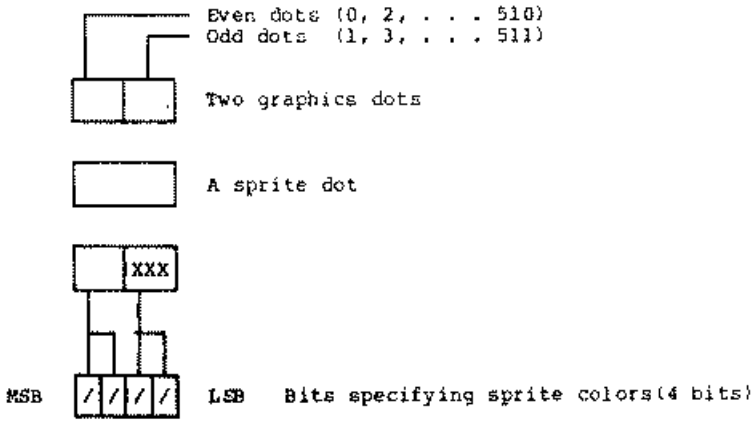
4. Sprite settings

- **Set the beginning** (head) address of the sprite attribute table in **registers R#5 and R#11**; and set the beginning (head) address of the **sprite pattern generator table** in register R#6. For details about **sprites, see the** section on SPRITE MODE 2.

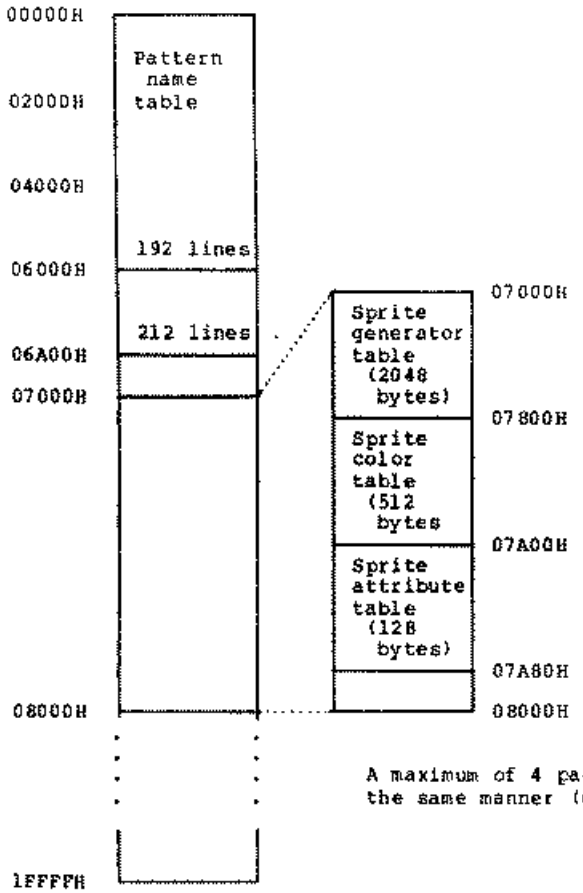


5. Hardware tiling function

- In GRAPHIC 5 mode, a hardware tiling function processes the sprite and background colors. For these colors, you can specify four bits; however, of these four bits, the higher-order two bits specify the color code of the even dots, and the lower-order two bits specify the color code of the odd dots of the x-coordinate (0 to 511).
- In GRAPHIC 5 mode, the size of one dot of a sprite is approximately twice that of a graphics dot; however, when this tiling function is used, one dot of a sprite may be displayed in two colors.
- The even and odd dots of the background color may also be specified in the same manner.



Example of VRAM allocation in GRAPHIC 5 mode



A maximum of 4 pages may be allocated in the same manner (using a 128K-byte VRAM).

GRAPHIC 6 MODE

Characteristics

- Bit-mapped Graphics Mode	
- Screen size	: 512 (w) x 212 (h) dots 512 (w) x 192 (h) dots
- Screen colors	: 16 colors out of 512 colors (per screen)
- Sprite mode	: Sprite mode 2
- VRAM area per screen	: 128K bytes (Two screens)

* To use this mode, the VRAM must have 128K bytes.

Controls

- Graphics	: VRAM pattern name table
- Background color code	: Low-order four bits of R#7
- Sprites	: VRAM sprite attribute table, VRAM sprite pattern table

Initial Settings

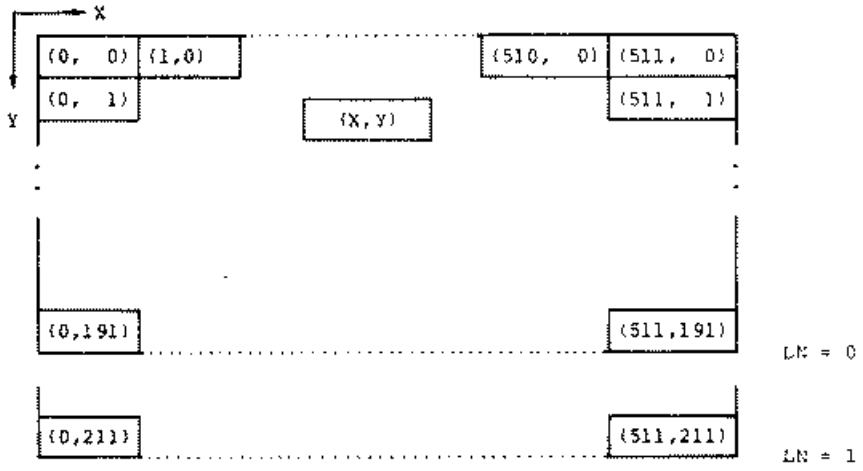
1. Mode and Register Settings

	MSB 7	6	5	4	3	2	1	0 LSB	
R#0	0	DG	IE2	IE1	1*	0*	1*	0	Mode register 0
R#1	0	BL	IE0	0*	0*	0	SI	MAG	Mode register 1
R#8	MS	LP	TP	CB	VR	0	SPD	BW	Mode register 2
R#9	LN	0	S1	S0	IL	EO	**NT	DC	Mode register 3

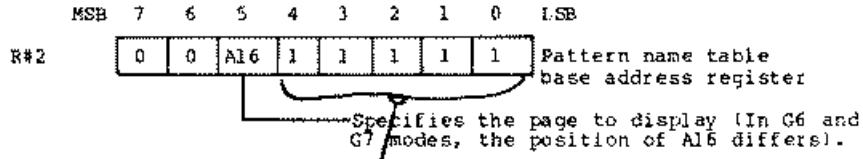
* Examples of settings in GRAPHIC 6 mode
** Indicates negative logic
In GRAPHIC 6 mode, if LN is set to 1, the screen height is 212 dots, and if LN is set to 0, the screen height is 192 dots.
All other bits are set accordingly.

2. Pattern name table settings

- The pattern name table is composed of one byte for every two dots on the screen. A color can be assigned for each dot from a selection of 16 colors out of 512 colors.

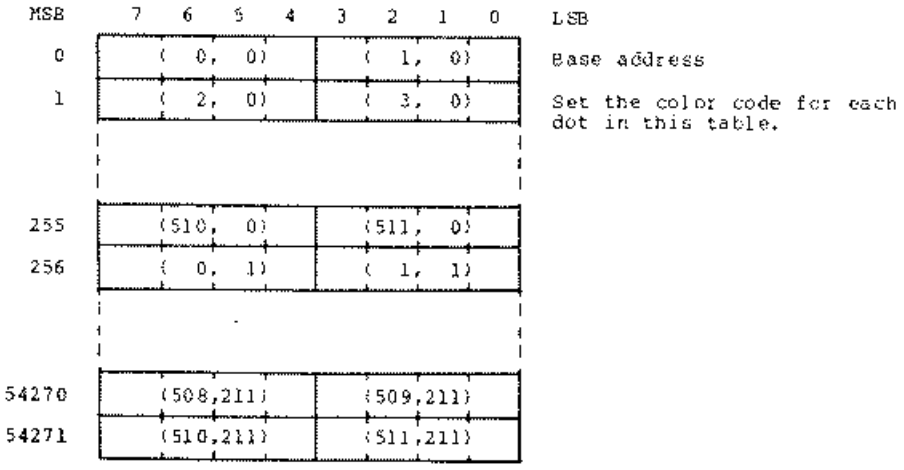


- Set the beginning (head) address of the pattern name table in register R#2.

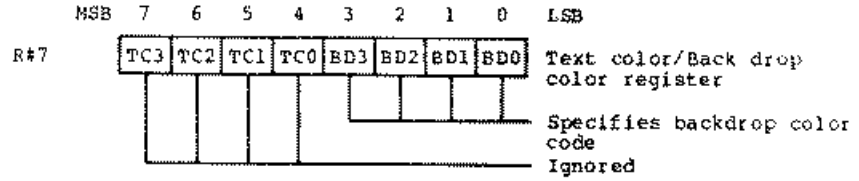


A16 mask over bits 7 - 5 of vertical line number

Pattern name table

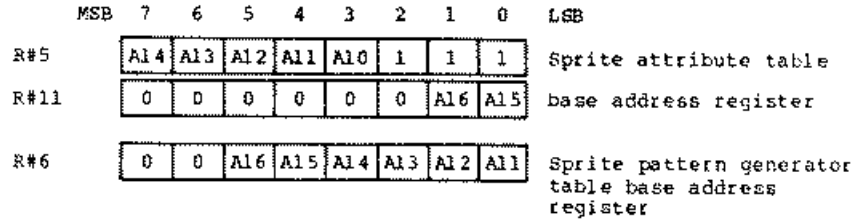


3. Color register settings

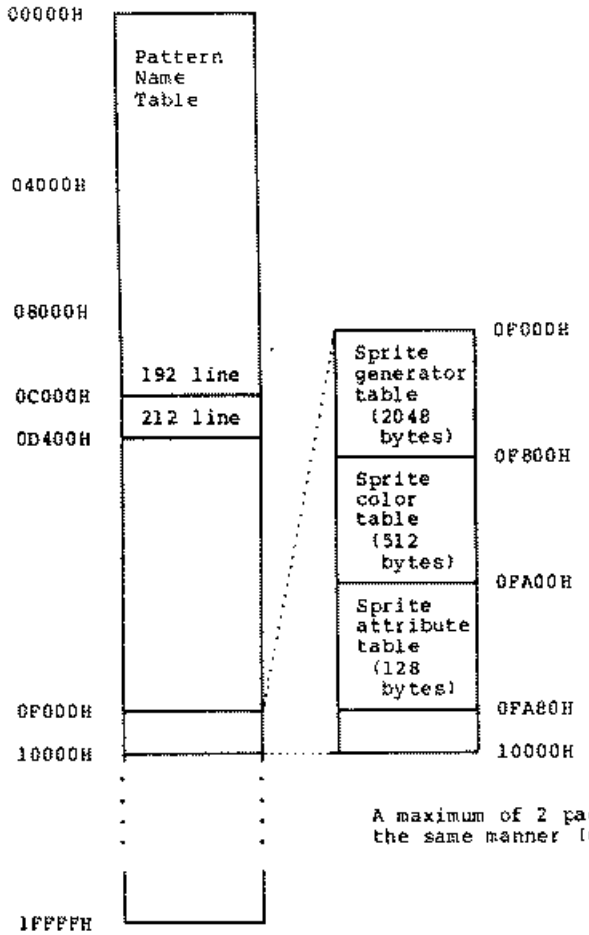


4. Sprite settings

- Set the beginning (head) address of the sprite attribute table in registers R#5 and R#11; and set the beginning (head) address of the sprite pattern generator table in register R#6. For details about sprites, see the section on SPRITE MODE 2.



Example of VRAM allocation in GRAPHIC 6 mode



A maximum of 2 pages may be allocated in the same manner (using a 128K-byte VRAM).

GRAPHIC 7 MODE

Characteristics

- Bit-mapped Graphics Mode
- Screen size : 256 (w) x 212 (h) dots
  256 (w) x 192 (h) dots
- Screen colors : 256 colors (per screen)
- Sprite mode : Sprite mode 2
- VRAM area per screen : 128K bytes (Two screens)

* To use this mode, the VRAM must have 128K bytes.

Controls

- Graphics : VRAM pattern name table
- Background color code : Low-order four bits of R#7
- Sprites : VRAM sprite attribute table, VRAM sprite pattern table

Initial Settings

1. Mode and Register Settings

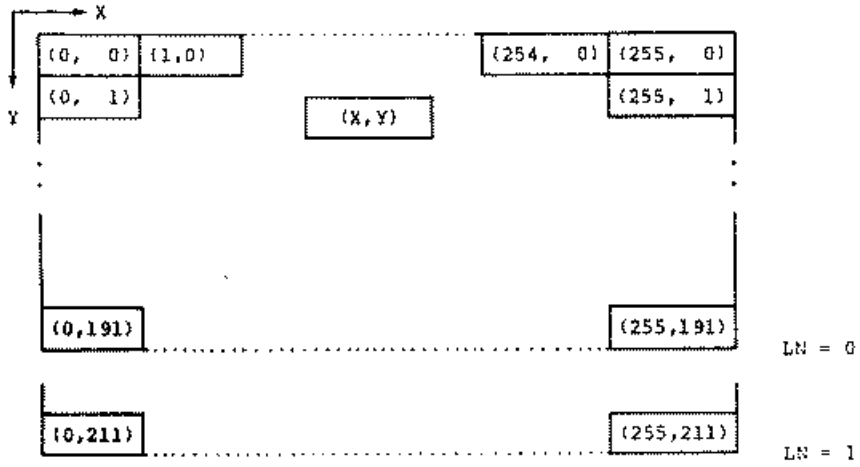
	MSB 7	6	5	4	3	2	1	0 LSB	
R#0	0	DG	IE2	IE1	1*	1*	1*	0	Mode register 0
R#1	0	BL	IE0	0*	0*	0	SI	MAG	Mode register 1
R#8	MS	LP	TP	CB	VR	0	SPD	BW	Mode register 2
R#9	LN	0	S1	S0	IL	EO	**NT	DC	Mode register 3

* Examples of settings in GRAPHIC 7 mode
** Indicates negative logic

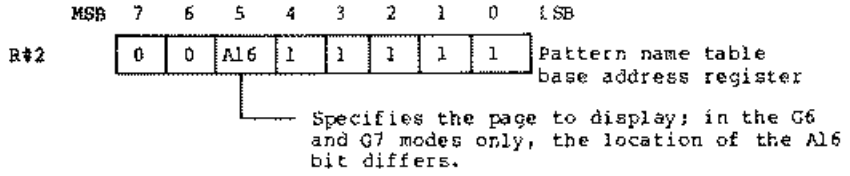
In GRAPHIC 7 mode, if LN is set to 1, the screen height is 212 dots, and if LN is set to 0, the screen height is 192 dots.
All other bits are set accordingly.

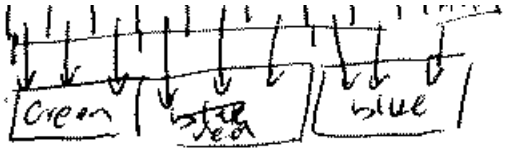
2. Pattern name table settings

- The pattern name table is composed of one byte for every dot on the screen. A color can be assigned for each dot from a selection of 256 colors.

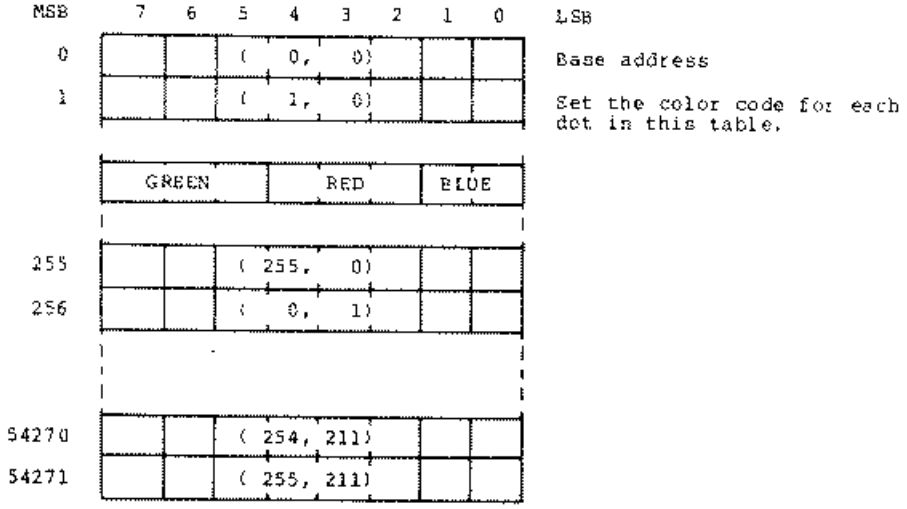


- Set the beginning (head) address of the pattern name table in register R#2.

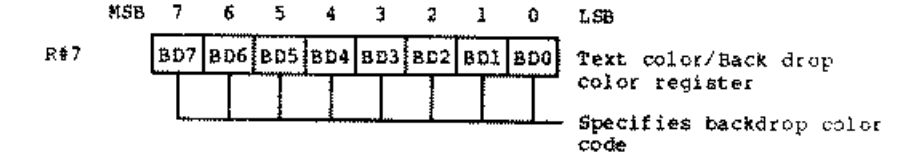




Pattern name table

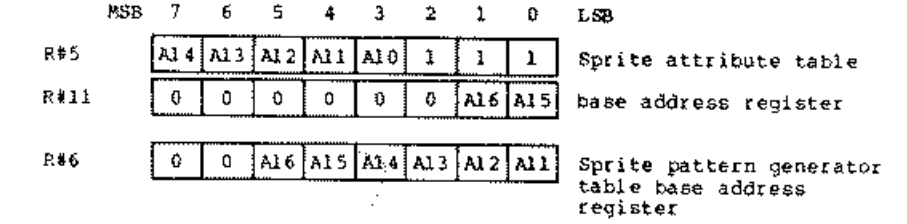


3. Color register settings

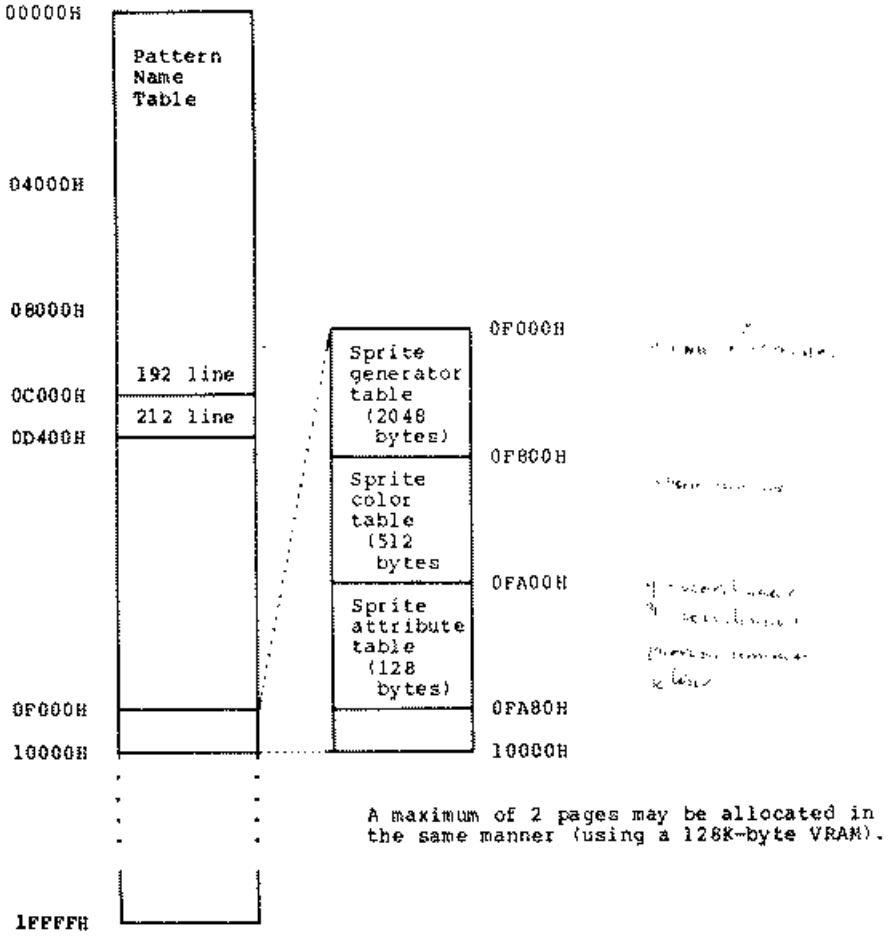


4. Sprite settings

- Set the beginning (head) address of the sprite attribute table in registers R#5 and R#11; and set the beginning (head) address of the sprite pattern generator table in register R#6. For details about sprites, see the section on SPRITE MODE 2.



Example of VRAM allocation in GRAPHIC 7 mode



A maximum of 2 pages may be allocated in the same manner (using a 128K-byte VRAM).

COMMANDS

1. Types of Commands

It is very easy to use MSX-VIDEO commands to perform functions such as LINE and PSET for use with graphics, and for transferring parts of the screen.

Summary of Commands

Command Name	Destination	Source	Late	Mnemonic	CM3	CM2	CM1	CM0
High-speed	VRAM	CPU	Byte	HMMC	1	1	1	1
move	VRAM	VRAM	Byte	YMMM	1	1	1	0
	VRAM	VRAM	Byte	HMMM	1	1	0	1
	VRAM	VDP	Byte	HMMV	1	1	0	0
Logical move	VRAM	CPU	Dot	LMMC	1	0	1	1
	CPU	VRAM	Dot	LMCM	1	0	1	0
	VRAM	VRAM	Dot	LMMM	1	0	0	1
	VRAM	VDP	Dot	LMMV	1	0	0	0
Line	VRAM	VDP	Dot	LINE	0	1	1	1
Search	VRAM	VDP	Dot	SRCH	0	1	1	0
Pset	VRAM	VDP	Dot	PSET	0	1	0	1
Point	VDP	VRAM	Dot	POINT	0	1	0	0
Invalid					0	0	1	1
Invalid					0	0	1	0
Invalid					0	0	0	1
Stop					0	0	0	0

- Commands are executed in the MSX-VIDEO by writing the data into R#46 (the Command Register, hereafter abbreviated CMR), and setting bit 0 of status register S#2 (CE/Command Execute) to 1. Before this can be done, the necessary parameters must be first have been set in registers R#32 to R#45.
- When the command execution is complete, CE is set to 0.
- To abort a command while it is being executed, execute a STOP.
- The results of command execution are only guaranteed during bit map mode (Graph4 to Graph7).

2. Page Concept

The parameters used for the MSX-VIDEO are all x-y coordinates. In other words, the internal command processor of the MSX-VIDEO accesses the entire VRAM area as x-y coordinates of the display mode.

When a screen is to be displayed, 212 lines of the same page are displayed (selected by R#23). To select the page to be displayed, use R#2.

When a command is being executed, the contents of the display screen are ignored.

The display modes and their relationships to the coordinates are shown in the table below.

GRAPH 4	Address	GRAPH 5
(0,0) (255,0) Page 0 (0,255) (255,255)	00000H	(0,0) (511,0) Page 0 (0,255) (511,255)
(0,256) (255,256) Page 1 (0,511) (255,511)	08000H	(0,256) (511,256) Page 1 (0,511) (511,511)
(0,512) (255,512) Page 2 (0,767) (255,767)	10000H	(0,512) (511,512) Page 2 (0,767) (511,767)
(0,768) (255,768) Page 3 (0,1023) (255,1023)	18000H 1FFFFH	(0,768) (511,768) Page 3 (0,1023) (511,1023)

GRAPH 7	Address	GRAPH 6
(0,0) (255,0) Page 0 (0,255) (255,255)	00000H	(0,0) (511,0) Page 0 (0,255) (511,255)
(0,256) (255,256) Page 1 (0,511) (255,511)	10000H 1FFFFH	(0,256) (511,256) Page 1 (0,511) (511,511)

3. Logical Operations

When the LINE, PSET, and LOGICAL MOVE commands are executed on the MSX-VIDEO, the operations may be performed on the color or the screen. To do logical operations on the MSX-VIDEO, write the lower four bits of R#46 (Command register) simultaneously when you specify the command.

Summary of Logical Operations

Name	Operation	LO3	LO2	LO1	LO0
IMP	DC=SC	0	0	0	0
AND	DC=SC*DC	0	0	0	1
OR	DC=$\overline{SC}$+DC	0	0	1	0
EOR	DC=$\overline{SC}$*DC+SC*$\overline{DC}$	0	0	1	1
NOT	DC=$\overline{SC}$	0	1	0	0
---		0	1	0	1
---		0	1	1	0
---		0	1	1	1
TIMP	if SC=0 then DC=DC else DC=SC	1	0	0	0
TAND	if SC=0 then DC=DC else DC=SC*DC	1	0	0	1
TOR	if SC=0 then DC=DC else DC=$\overline{SC}$+DC	1	0	1	0
TEOR	if SC=0 then DC=DC else DC=$\overline{SC}$*DC+SC*$\overline{DC}$	1	0	1	1
TNOT	if SC=0 then DC=DC else DC=$\overline{SC}$	1	1	0	0
---		1	1	0	1
---		1	1	1	0
---		1	1	1	1

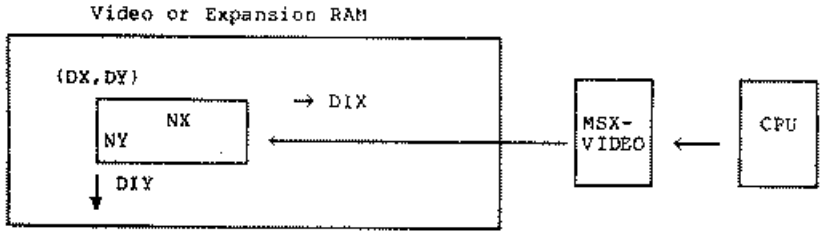
* SC = Source Color code
* DC = Destination Color code
* EOR = Exclusive OR

4. Explanations of Commands

4.1 HMMC (High-speed move CPU to VRAM)

The HMMC command transfers data from the CPU to the Video or expansion RAM in a specified rectangular area (in x-y coordinates) via the MSX-VIDEO.

Since the data to be transferred is done in units of one byte, there is a limitation, according to the display mode, on the value for x.



4.1.1 HMMC Execution Order

1. First, set the necessary parameters in the command register of the MSX-VIDEO.

   MXD: Select destination memory
   0: Video RAM
   1: Expansion RAM

   DX: Basic x-coordinate of destination (0 to 511) *1
   DY: Basic y-coordinate of destination (0 to 1023)

   NX: Dots to move in x-direction (0 to 511) *1
   NY: Dots to move in y-direction (0 to 1023)

   *1 Note that in the G4 and G6 modes, the lower one bit, and in the G5 mode, the lower two bits, are lost.

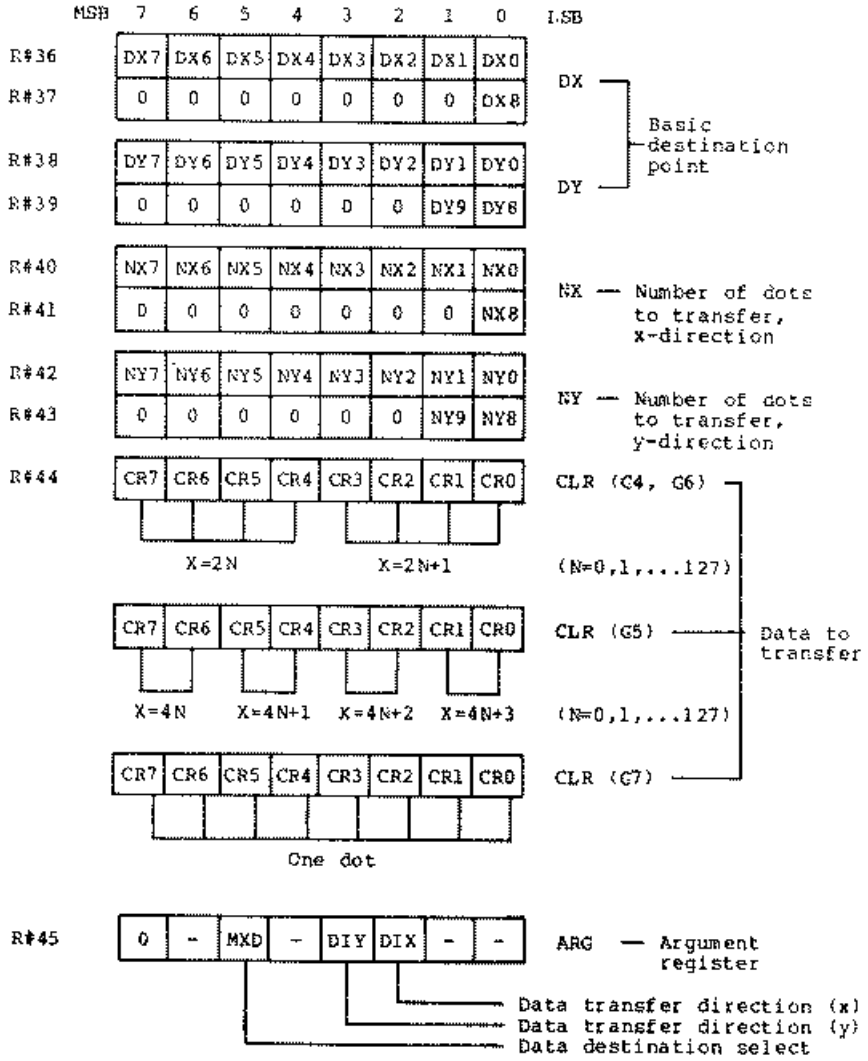
   DIX: Direction for NX from x-coordinate of destination
   0: Right
   1: Left
   DIY: Direction for NY from y-coordinate of destination
   0: Down
   1: Up

   CLR (R#44:Color register):
   First byte of data to be transferred

2. After you specify the above data, execute the command by writing 1 1 1 1 0 0 0 0 B into the CMR(R#46:Command register).

3. While checking TR and CE in Status Register S#2, send the second byte and all bytes following into the CLR register.

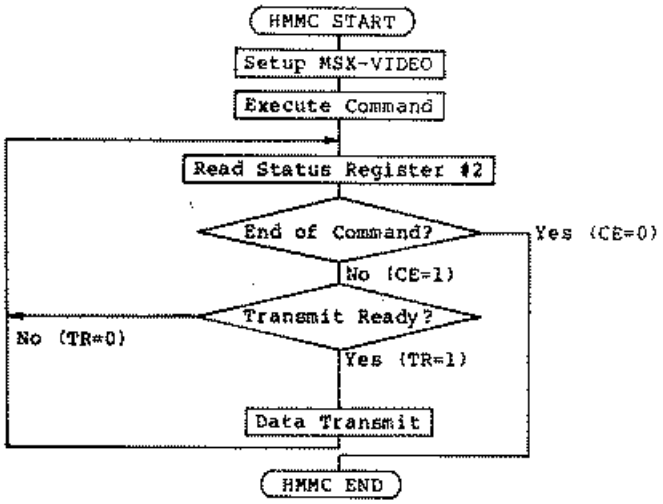
4.1.2 Setting up the HMMC register



4.1.3 Execution of HMMC commands

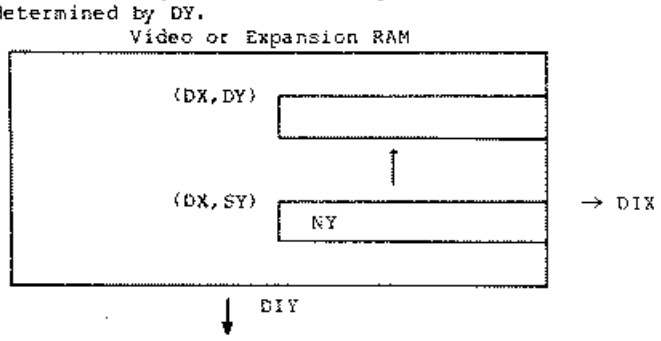
	MSB 7	6	5	4	3	2	1	0	LSB
R#46	1	1	1	1	-	-	-	-	CMR

4.1.4 Flowchart of HMMC execution



4.2 YMMM (High-speed move VRAM to VRAM, y only)

The YMMM command transfers data from the area specified by DX, SY, NY, DIX, DIY and the right (or left) edge of the Video RAM, in the y-direction determined by DY.



4.2.1 YMMM Execution Order

1. First, set the necessary parameters in the command register of the MSX-VIDEO.

   MXD: Select destination memory
   0: Video RAM
   1: Expansion RAM

   DY: Basic y-coordinate of destination (0 to 1023)

   DX: Basic x-coordinate of source point (0 to 511) *1
   SX: Basic y-coordinate of source point (0 to 1023)

   NY: Dots to move in y-direction (0 to 1023)

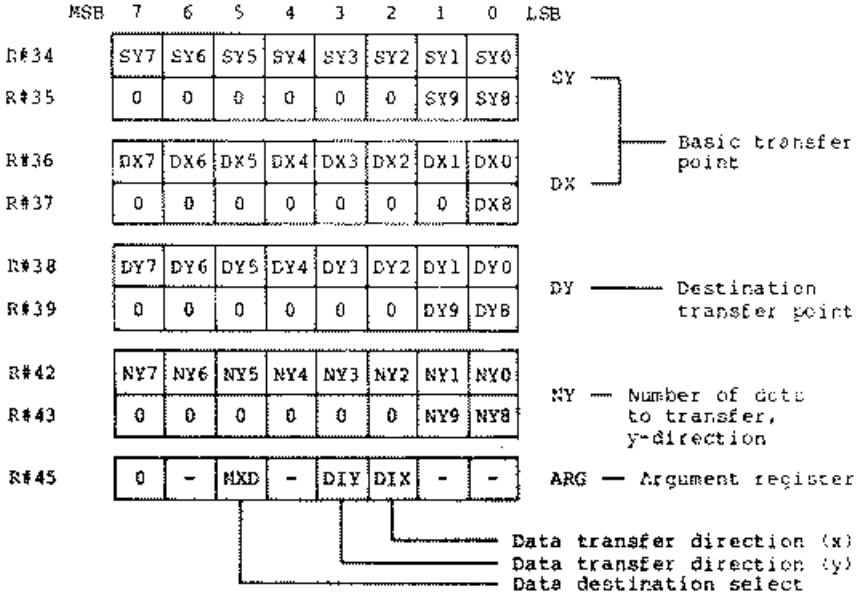
   *1 Note that in the G4 and G6 modes, the lower one bit, and in the G5 mode, the lower two bits, are lost.

   DIX: Direction for x-coordinate of source point to the right or left end of screen
   0: Right
   1: Left
   DIY: Direction of source point from NY
   0: Down
   1: Up

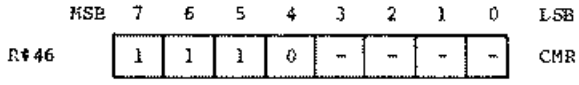
2. After you specify the above data, execute the command by writing 1 1 1 1 0 0 0 0 B into the CMR(R#46:Command register).

3. The above procedure will execute the YMMM command in the MSX-VIDEO. While executing the YMMM command, the CE bit of the status register (S#2) will be set to 1, and when the command is complete, it will be reset to 0.

4.2.2 Setting up the YMMM register



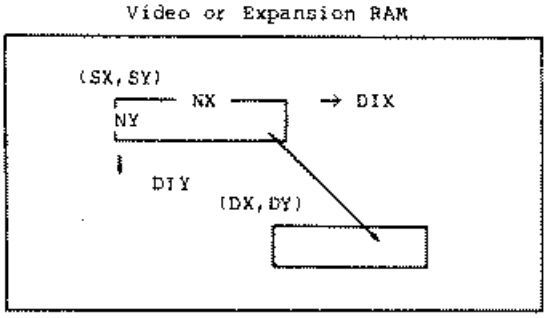
4.2.3 Execution of YMMM Commands



4.3 HMMM (High-speed move VRAM to VRAM)

The HMMM command transfers data in a specified rectangular area from the VRAM or the expansion RAM to the VRAM or the expansion RAM.

Since the data to be transferred is done in units of one byte, there is a limitation, according to the display mode, on the value for x.



4.3.1 HMMM Execution Order

1. **First, set the necessary parameters in the command register of the MSX-VIDEO.**

   MXS: Select source memory
   0: Video RAM
   1: Expansion RAM
   MXD: Select destination memory
   0: Video RAM
   1: Expansion RAM

   SX: Source point x-coordinate (0 to 511) *1
   SY: Source point y-coordinate (0 to 1023)

   NX: Dots to move in x-direction (0 to 511) *1
   NY: Dots to move in y-direction (0 to 1023)

   DIX: Direction for NX from source point
   0: Right
   1: Left
   DIY: Direction for NY from source point
   0: Down
   1: Up

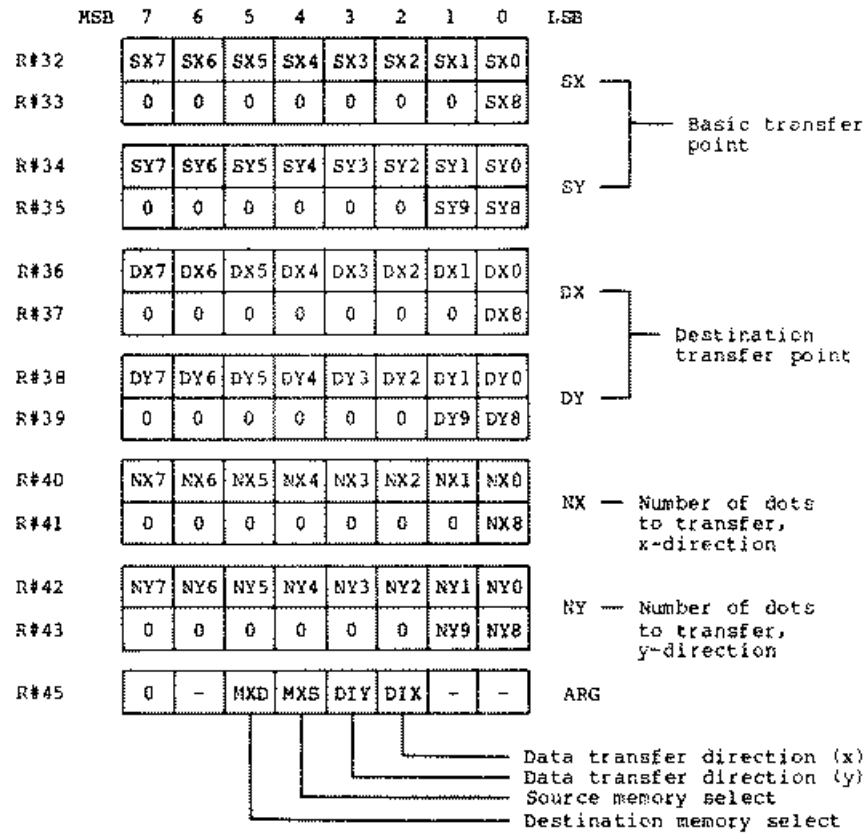
   DX: Basic x-coordinate of destination (0 to 511) *1
   DY: Basic y-coordinate of destination (0 to 1023)

   *1 Note that in the G4 and G6 modes, the lower one bit, and in the G5 mode, the lower two bits of SX, DX, and NX, are lost.

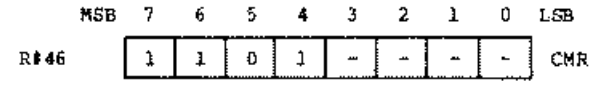
2. After you specify the above data, execute the command by writing 1 1 0 1 0 0 0 0 B into the CMR(R#46:Command register).

3. The above procedure will execute the HMMM command in the MSX-VIDEO. While executing the HMMM command, the CE bit of the status register (S#2) will be set to 1, and when the command is complete, it will be reset to 0.

4.3.2 Setting up the HMMM register



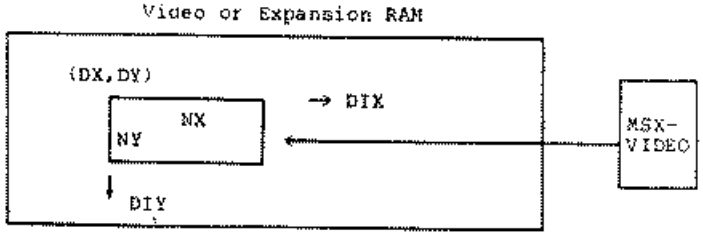
4.3.3 Executing the HMMM command



4.4 HMMV (High-speed move VDP to VRAM)

The HMMV command is used to paint in a specified rectangular area of the VRAM or the expansion RAM.

Since the data to be transferred is done in units of one byte, there is a limitation, according to the display mode, on the value for x.



4.4.1 HMMV Execution Order

1. First, set the necessary parameters in the command register of the MSX-VIDEO.

   MXD: Select destination memory
   0: Video RAM
   1: Expansion RAM

   NX: Dots to move in x-direction (0 to 511) *1
   NY: Dots to move in y-direction (0 to 1023)

   DIX: Direction for NX from source point
   0: Right
   1: Left
   DIY: Direction for NY from source point
   0: Down
   1: Up

   DX: Basic x-coordinate of destination (0 to 511) *1
   DY: Basic y-coordinate of destination (0 to 1023)

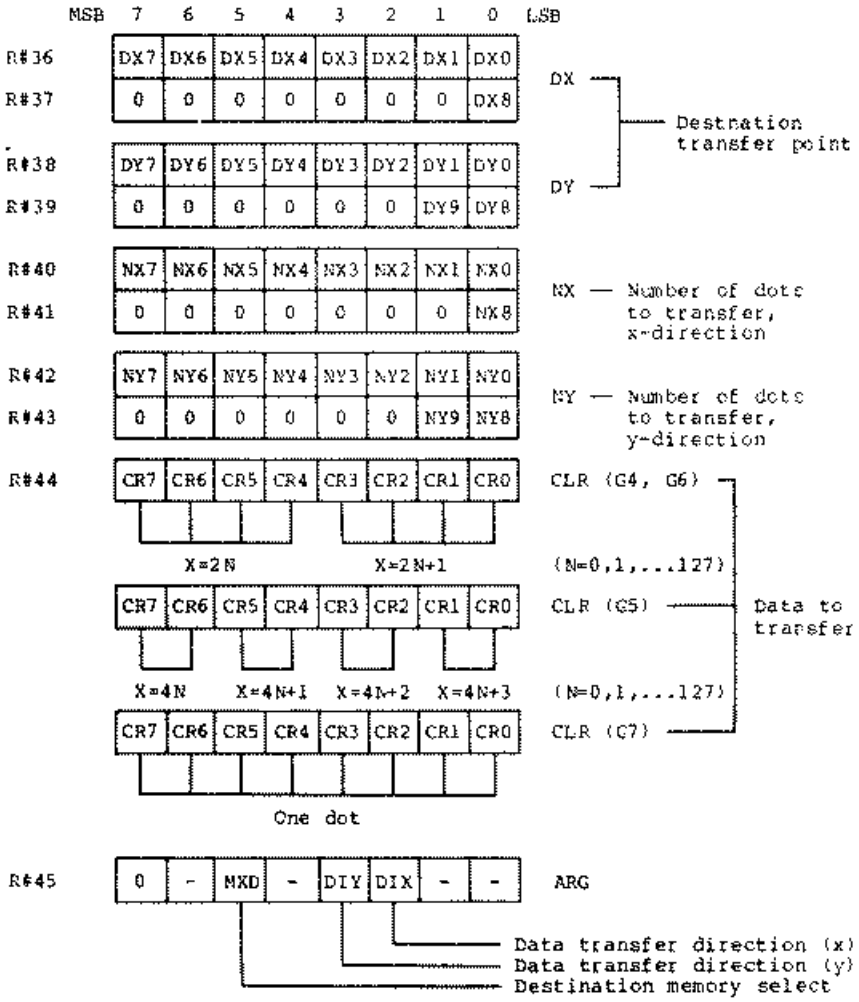
   *1 Note that in the G4 and G6 modes, the lower one bit, and in the G5 mode, the lower two bits of DX and NX, are lost.

   CLR: Color code data

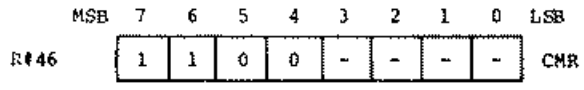
2. After you specify the above data, execute the command by writing 1 1 0 0 0 0 0 0 into the CMR.

3. The above procedure will execute the HMMV command in the MSX-VIDEO. While executing the HMMV command, the CE bit of the status register (S#2) will be set to 1, and when the command is complete, it will be reset to 0.

4.4.2 Setting up the HMMV register



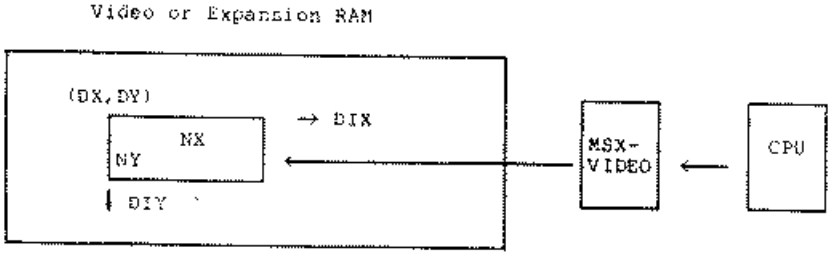
4.4.3 Executing the HMMV command



4.5 LMMC (Logical move CPU to VRAM)

The LMCC command transfers data from the CPU to the Video or expansion RAM in a specified rectangular area (in x-y coordinates) via the MSX-VIDEO.

Since the data to be transferred is done in units of dots, logical operations may be done on the destination points.



4.5.1 LMMC Execution Order

1. First, set the necessary parameters in the command register of the MSX-VIDEO.

   MXD: Select destination memory
   0: Video RAM
   1: Expansion RAM

   DX: Basic x-coordinate of destination (0 to 511)
   DY: Basic y-coordinate of destination (0 to 1023)

   NX: Dots to move in x-direction (0 to 511)
   NY: Dots to move in y-direction (0 to 1023)

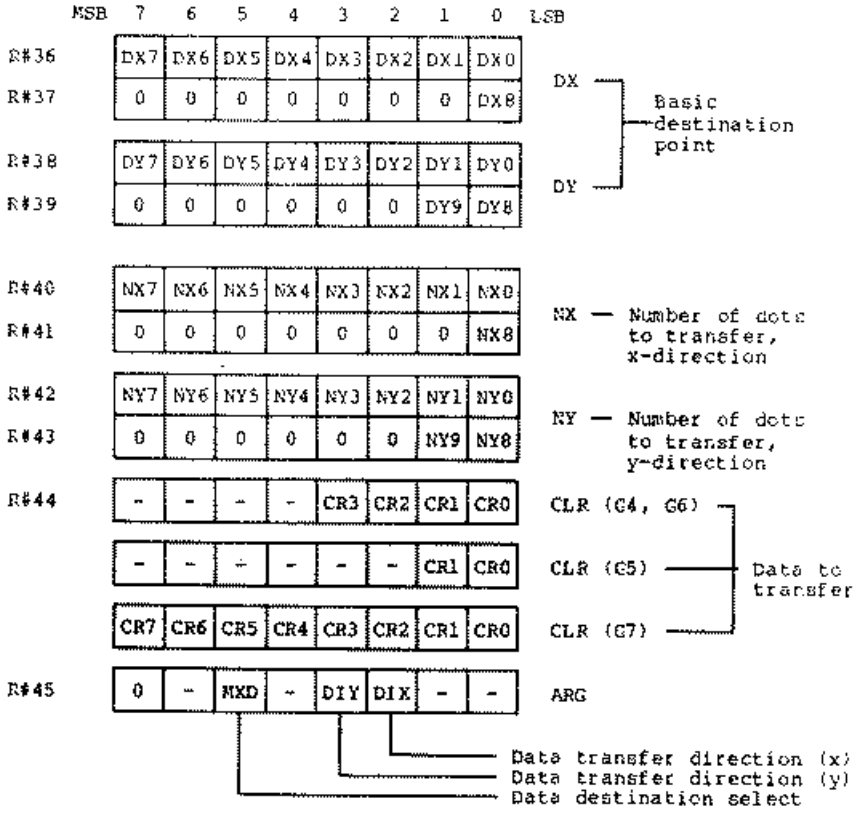
   DIX: Direction for NX from x-coordinate of destination
   0: Right
   1: Left
   DIY: Direction for NY from y-coordinate of destination
   0: Down
   1: Up

   CLR: First byte of data to be transferred

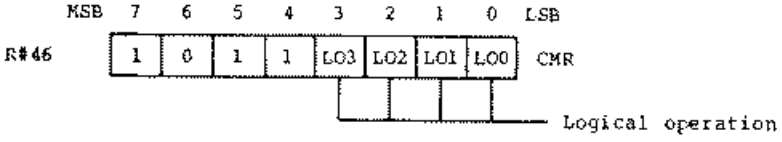
2. After you specify the above data, execute the command. Write 1 0 1 1 B into the higher four bits of the command register (CMR), and place the logical operation code in the lower four bits of CMR.

3. While checking TR and CE in Status Register S#2, send the second byte and all bytes following into the CLR register.

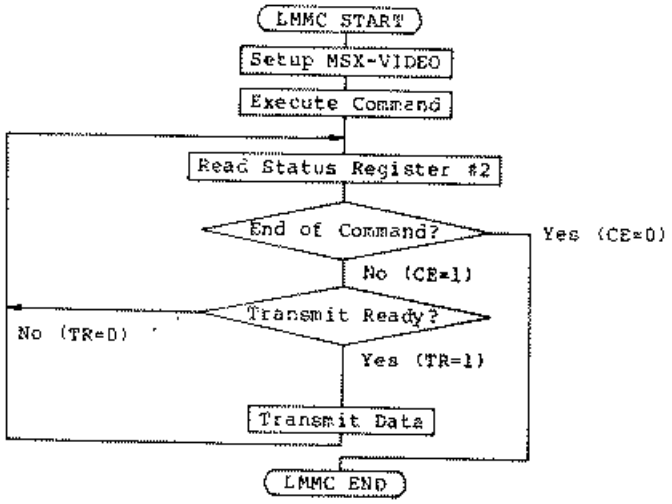
4.5.2 Setting up the LMMC register



4.5.3 Execution of LMMC commands



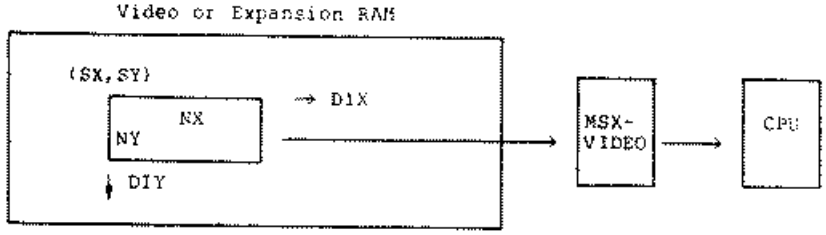
4.5.4 Flowchart of LMMC execution



4.6 LMCM (Logical move VRAM to CPU)

The LMCM command transfers data from the Video or expansion RAM to the CPU, in a specified rectangular area (in x-y coordinates) via the MSX-VIDEO.

The data is transferred in units of dots.



4.6.1 LMCM Execution Order

1. First, set the necessary parameters in the command register of the MSX-VIDEO.

   MXS: Select source memory
   0: Video RAM
   1: Expansion RAM

   SX: Basic x-coordinate of source point (0 to 511)
   SY: Basic y-coordinate of source point (0 to 1023)

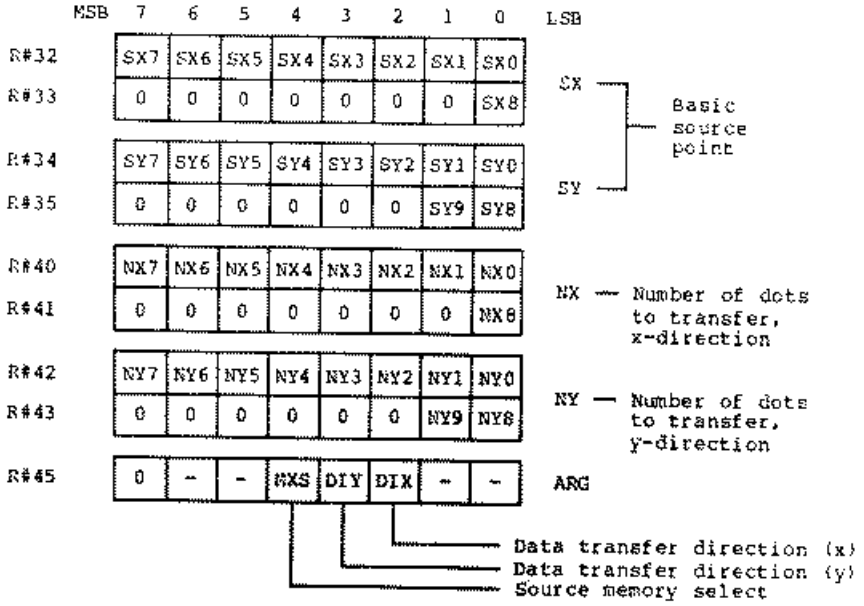
   NX: Dots to move from source point in x-direction (0 to 511)
   NY: Dots to move from source point in y-direction (0 to 1023)

   DIX: Direction for NX from x-coordinate of source point
   0: Right
   1: Left
   DIY: Direction for NY from y-coordinate of source point
   0: Down
   1: Up

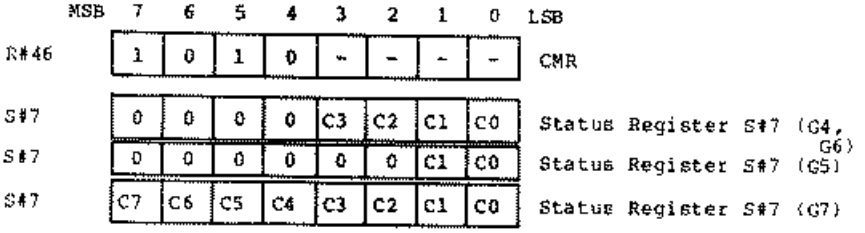
2. After you specify the above data, execute the command by writing 1 0 1 0 0 0 0 0 B into the CMR.

3. While checking TR and CE in Status Register S#2, read Status Register S#7.

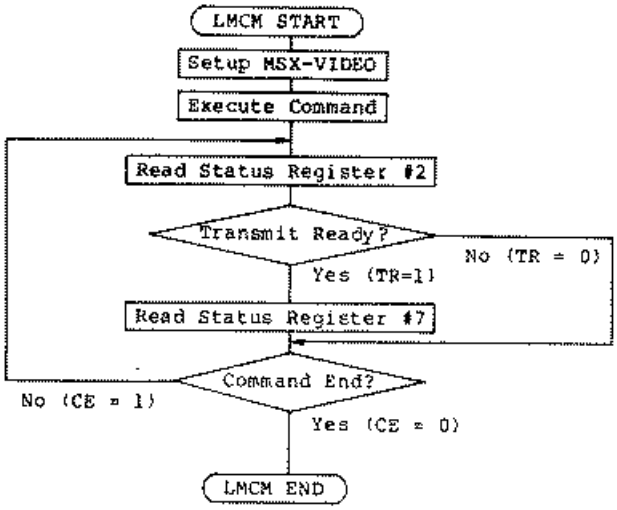
4.6.2 Setting up the LMCM register



4.6.3 Execution of LMCM commands



4.6.4 Flowchart of LMCM execution

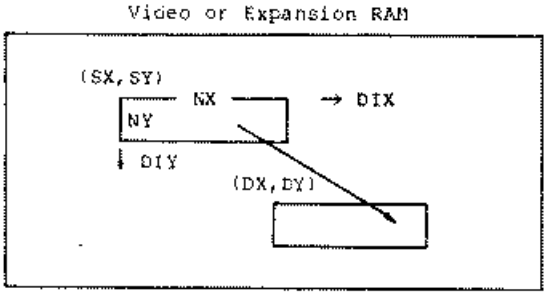


Notes 1. TR must be reset before the command is executed. Read Status Register #7 when you set up the MSX-VIDEO.
2. Even though the data is set in Status Register #7 and TR=1, the command will be completed within the MSX-VIDEO and CE is set to 0.

4.7 LMMM (Logical move VRAM to VRAM)

The LMMM command transfers data in a specified rectangular area from the VRAM or the expansion RAM to the VRAM or the expansion RAM.

Since the data to be transferred is done in units of dots, logical operations may be done on the destination data.



4.7.1 LMMM Execution Order

1. First, set the necessary parameters in the command register of the MSX-VIDEO.

   MXS: Select source memory
   0: Video RAM
   1: Expansion RAM

   MXD: Select destination memory
   0: Video RAM
   1: Expansion RAM

   SX: Source point x-coordinate (0 to 511)
   SY: Source point y-coordinate (0 to 1023)

   NX: Dots to move in x-direction (0 to 511)
   NY: Dots to move in y-direction (0 to 1023)

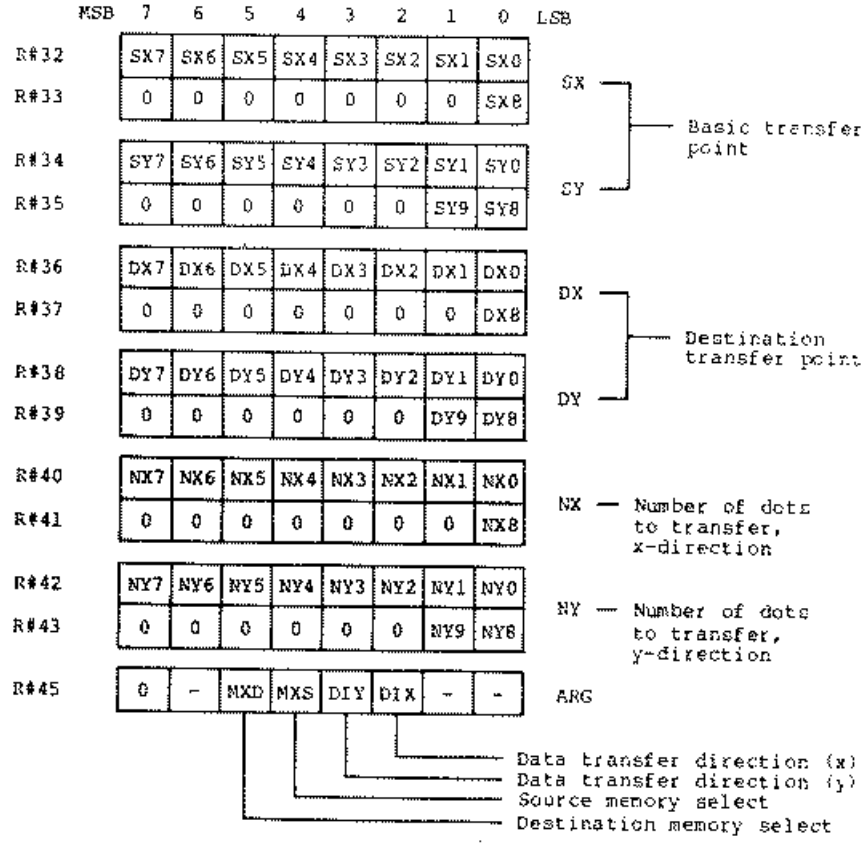
   DIX: Direction for NX from source point
   0: Right
   1: Left
   DIY: Direction for NY from source point
   0: Down
   1: Up

   DX: Basic x-coordinate of destination (0 to 511)
   DY: Basic y-coordinate of destination (0 to 1023)

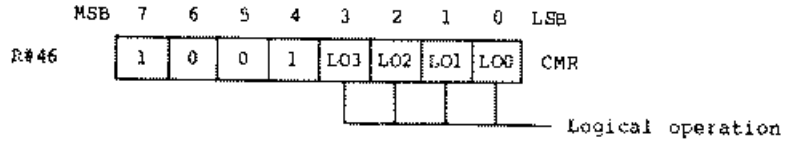
2. After you specify the above data, execute the command. Write 1 0 0 1 B into the upper four bits of the command register (CMR) and the logical operation into the lower four bits.

3. The above procedure will execute the LMMM command in the MSX-VIDEO. While executing the LMMM command, the CE bit of the status register (S#2) will be set to 1, and when the command is complete, it will be reset to 0.

4.7.2 Setting up the LMMM register



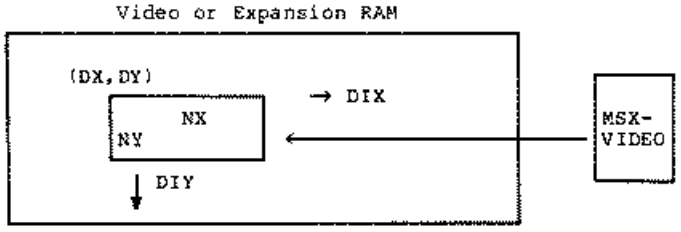
4.7.3 Executing the LMMM command



4.8 LMMV (Logical move VDP to VRAM)

The LMMV command paints in a specified rectangular area of the Video or Expansion RAM according to a specified color code.

The data is transferred in units of one dot, and a logical operation may be done on the destination data.



4.8.1 LMMV Execution Order

1. First, set the necessary parameters in the command register of the MSX-VIDEO.

   MXD: Select destination memory
   0: Video RAM
   1: Expansion RAM

   NX: Dots to move in x-direction (0 to 511)
   NY: Dots to move in y-direction (0 to 1023)

   DIX: Direction for NX from source point
   0: Right
   1: Left
   DIY: Direction for NY from source point
   0: Down
   1: Up

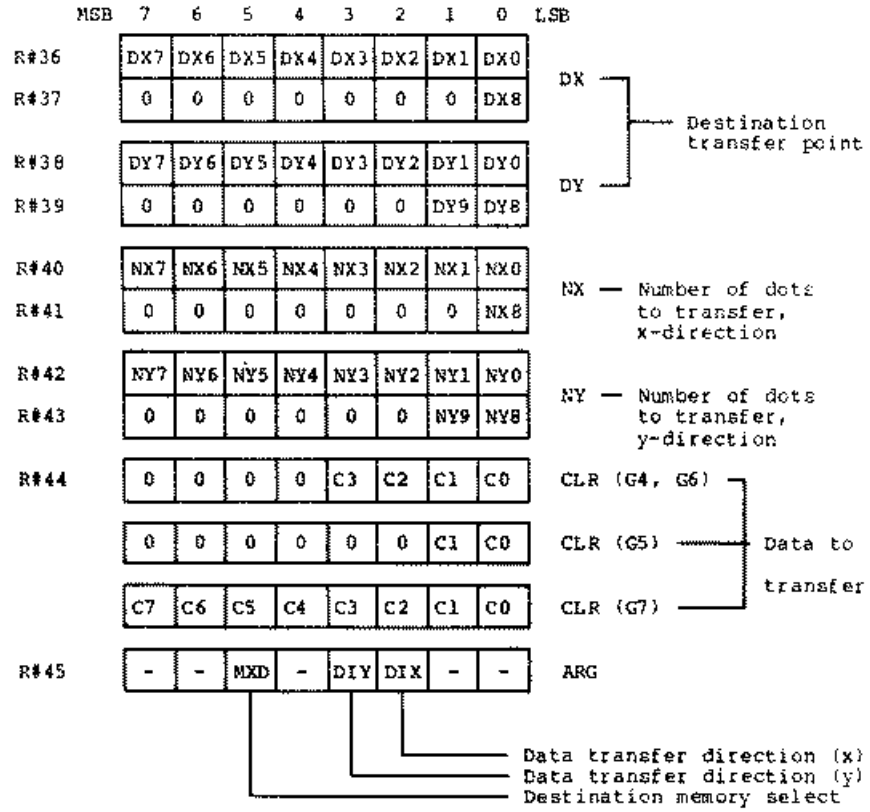
   DX: Basic x-coordinate of destination (0 to 511)
   DY: Basic y-coordinate of destination (0 to 1023)

   CLR: Color code data

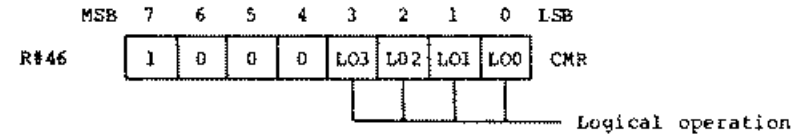
2. After you specify the above data, execute the command. Write 1 0 0 0 B into the upper four bits of the command register (CMR), and the logical operation into the lower four bits of the CMR.

3. The above procedure will execute the LMMV command in the MSX-VIDEO. While executing the LMMV command, the CE bit of the status register (S#2) will be set to 1, and when the command is complete, it will be reset to 0.

4.8.2 Setting up the LMMV register



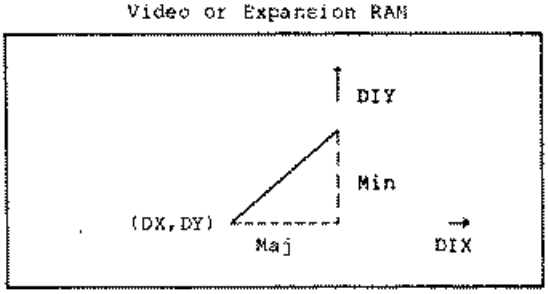
4.8.3 Executing the LMMV command



4.9 LINE

The LINE command draws a straight line in the Video or Expansion RAM.

The line drawn is the hypoteruse that results after the long and short sides of a triangle are defined. The two sides are defined as distances from a single point.



4.9.1 LINE Execution Order

1. First, set the necessary parameters in the command register of the MSX-VIDEO.

   MXD: Select destination memory
   0: Video RAM
   1: Expansion RAM

   Maj: Number of dots in long side (0 to 1023)
   Min: Number of dots in short side (0 to 511)

   MAJ: Direction for long side
   0: Long side is in x-axis
   1: Long side is in y-axis (or x-axis = y-axis)

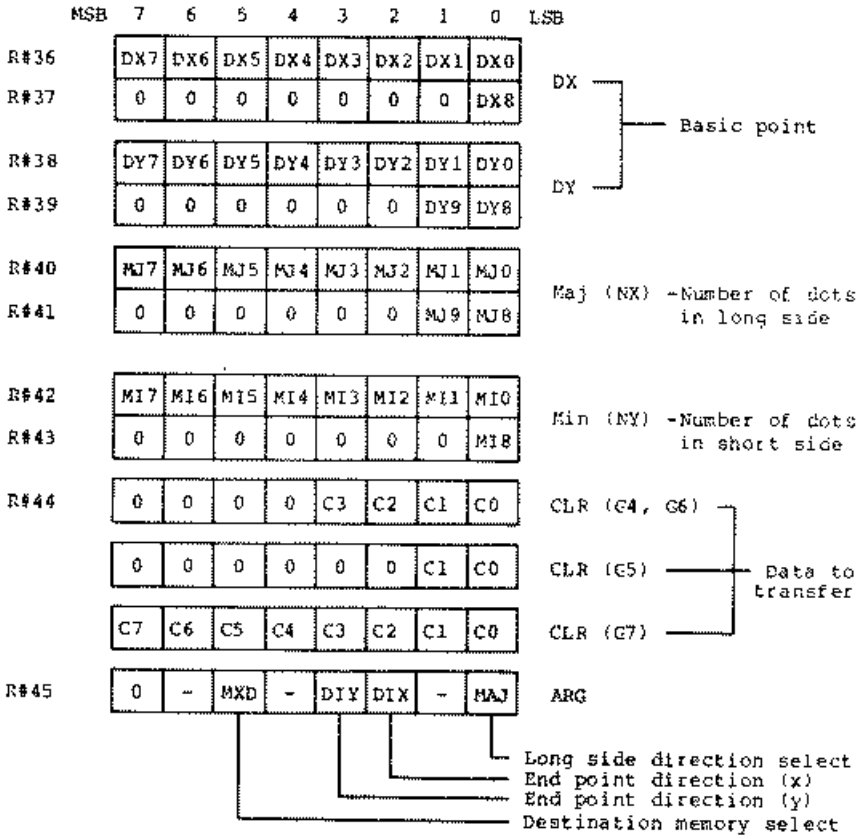
   DIX: Direction from source point to end point
   0: Right
   1: Left
   DIY: Direction from source point to end point
   0: Down
   1: Up

   DX: Basic x-coordinate (0 to 511)
   DY: Basic y-coordinate (0 to 1023)

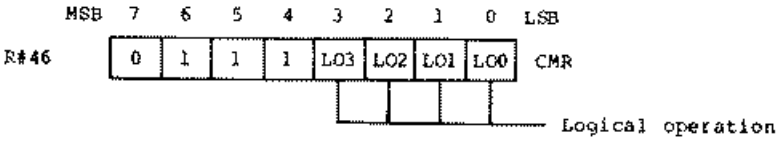
2. After you specify the above data, execute the command. Write 0 1 1 1 B into the upper four bits of the command register (CMR), and the logical operation into the lower four bits of the CMR.

3. The above procedure will execute the LINE command in the MSX-VIDEO. While executing the LINE command, the CE bit of the status register (S#2) will be set to 1, and when the command is complete, it will be reset to 0.

4.9.2 Setting up the LINE register

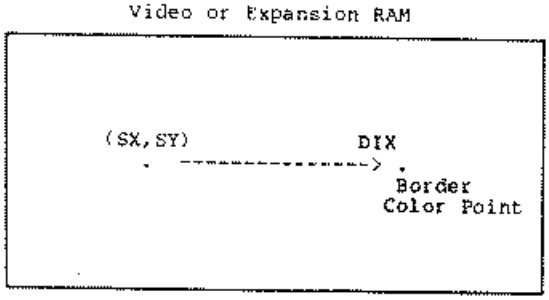


4.9.3 Executing the LINE command



4.10 SRCH

The SRCH command searches for a border color in the Video or Expansion RAM to the right or left of a basic point.



4.10.1 SRCH Execution Order

1. First, set the necessary parameters in the command register of the MSX-VIDEO.

   MXD: Select search memory
   0: Video RAM
   1: Expansion RAM

   DIX: Direction from source point to search
   0: Right
   1: Left

   EQ: When 1, ends execution when border color is found.
   When 0, ends execution when a color other than the border color is found.

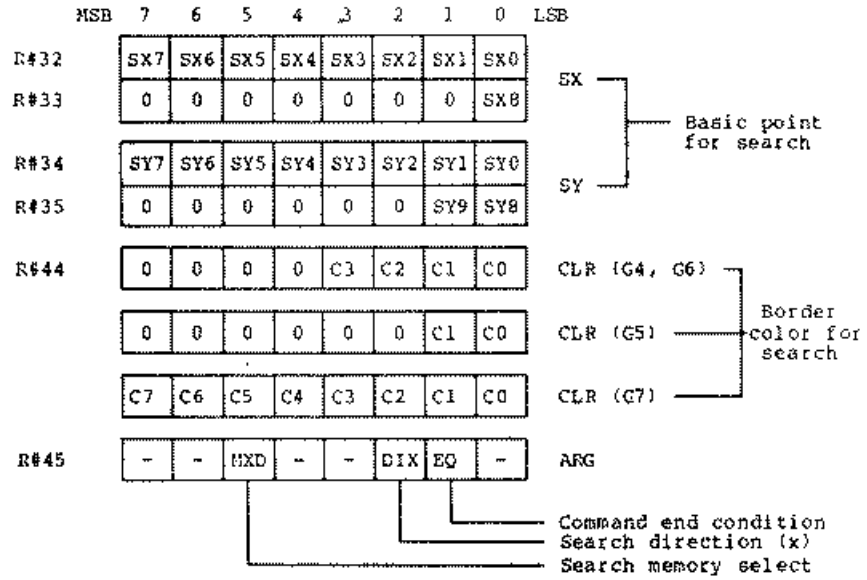
   SX: Basic x-coordinate for search (0 to 511)
   SY: Basic y-coordinate for search (0 to 1023)

   CLR: Color code data

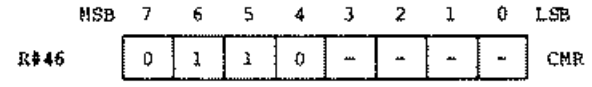
2. After you specify the above data, execute the command by writing 0 1 1 0 0 0 0 0 B into the CMR.

3. The above procedure will execute the SRCH command in the MSX-VIDEO. While executing the SRCH command, the CE bit of the status register (S#2) will be set to 1, and when the command is complete, it will be reset to 0. If the color is found the BD bit is set to 1.

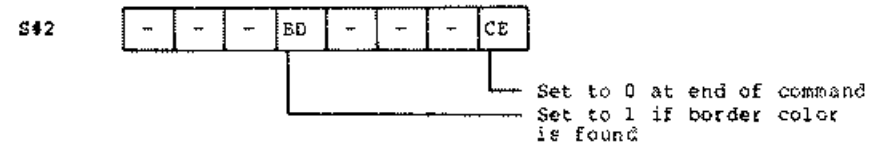
4.10.2 Setting up the SRCH register

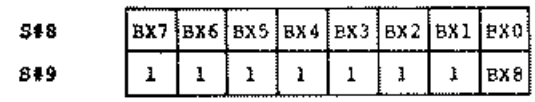


4.10.3 Executing the SRCH command

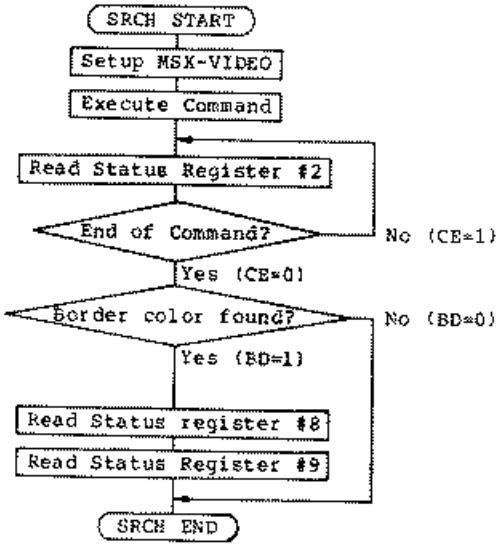


Status registers





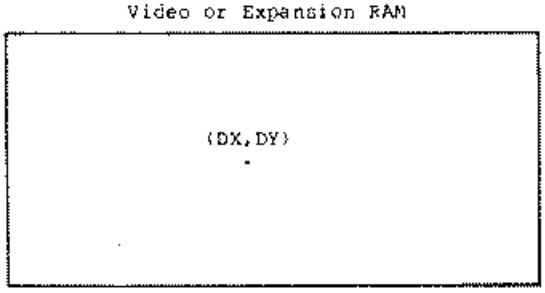
4.10.4 Flowchart of SRCH execution



4.11 PSET

The PSET command draws a dot in the Video or Expansion RAM.

A logical operation is done on the data of a dot that is already displayed.



4.11.1 PSET Execution Order

1. First, set the necessary parameters in the command register of the MSX-VIDEO.

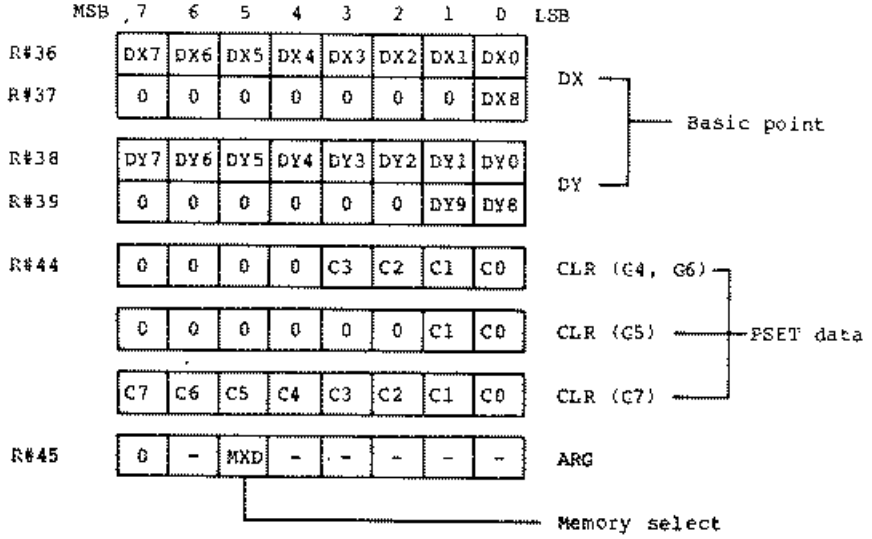
   MXD: Select memory
   0: Video RAM
   1: Expansion RAM

   DX: x-coordinate of dot (0 to 511)
   DY: y-coordinate of dot (0 to 1023)

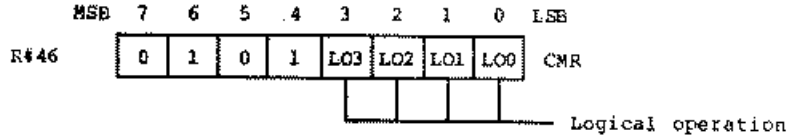
2. After you specify the above data, execute the command. Write 0 1 0 1 B into the high-order four bits of the command register (CMR) and the logical operation into the low-order four bits of CMR.

3. The above procedure will execute the PSET command in the MSX-VIDEO. While executing the PSET command, the CE bit of the status register (S#2) will be set to 1, and when the command is complete, it will be reset to 0.

4.11.2 Setting up the PSET register

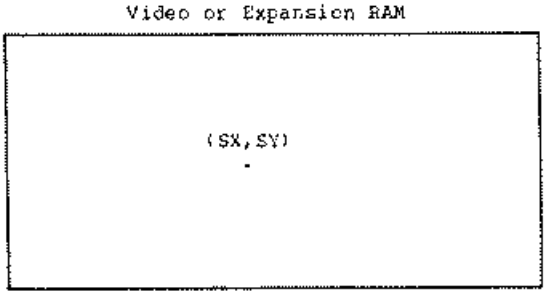


4.11.3 Executing the PSET command



4.12 POINT

The POINT command draws a dot of the color specified in the Video or Expansion RAM.



4.12.1 POINT Execution Order

1. First, set the necessary parameters in the command register of the MSX-VIDEO.

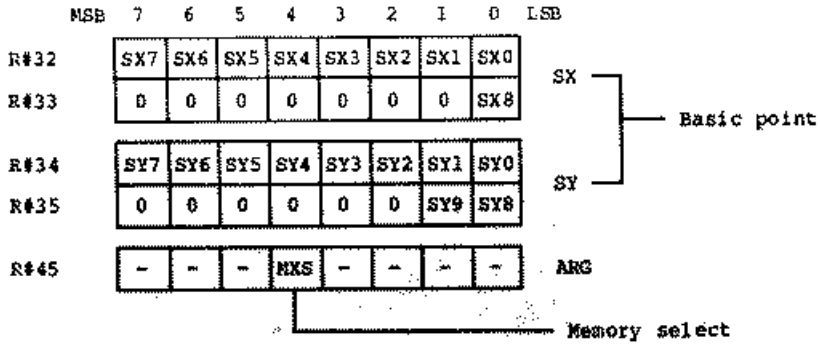
   MXS: Select memory
   0: Video RAM
   1: Expansion RAM

   SX: x-coordinate of dot (0 to 511)
   SY: y-coordinate of dot (0 to 1023)

2. After you specify the above data, execute the command by writing 0 1 0 0 0 0 0 0 B into the CMR.

3. The above procedure will execute the POINT command in the MSX-VIDEO. While executing the POINT command, the CE bit of the status register (S#2) will be set to 1, and when the command is complete, it will be reset to 0. The color code data is set to status register (S#7).

4.12.2 Setting up the POINT register



4.12.3 Executing the POINT command

	MSB 7	6	5	4	3	2	1	0 LSB	
R#46	0	1	0	0	-	-	-	-	CMR

Status registers

S#2	-	-	-	-	-	-	-	CE	

Set to 0 at end of command

S#7	0	0	0	0	C3	C2	C1	C0	Color code (G4, G6)
S#7	0	0	0	0	0	0	C1	C0	Color code (G5)
S#7	C7	C6	C5	C4	C3	C2	C1	C0	Color code (G7)

5. Speeding up the processing of commands

The processing of commands can be speeded up by the following two methods.

5.1 Inhibit the display of sprites

If bit 1 of Register R#8 (SPD) is set to 1, processing for sprites can be used for commands instead; and therefore the command execution is faster.

5.2 Inhibit the screen display

If bit 6 of Register R#1 (BL) is set to 0, processing for the screen can be used for commands instead; and therefore the command execution is faster.

6. Conditions of registers after command execution

After commands on the MSX-VIDEO are executed, the conditions of the registers will be as listed in the following table.

	SX	SY	DX	DY	NX	NY	CLR	CMR H	CLR L	ARG
HMMC	-	-	-	*	-	#	-	0	-	-
YHMM	-	*	-	*	-	#	-	0	-	-
HMMM	-	*	-	*	-	#	-	0	-	-
HMMV	-	-	-	*	-	#	-	0	-	-
LMMC	-	-	-	*	-	#	-	0	-	-
LMCM	-	*	-	-	-	#	*	0	-	-
LMMM	-	*	-	*	-	#	-	0	-	-
LMMV	-	-	-	*	-	#	-	0	-	-
LINE	-	-	-	*	-	-	-	0	-	-
SRCH	-	-	-	-	-	-	-	0	-	-
PSET	-	-	-	-	-	-	-	0	-	-
POINT	-	-	-	-	-	-	*	0	-	-

- \- Unchanged
- * Coordinate at command end (SY*, DY*) or color code
- # The count (NYB) when the end of the screen was detected

Note: The values for SY*, DY* and NYB are the dots, as substituted for N in the equations below.

SY* = SY + N DY* = DY + N (DIY = 0)
SY* = SY - N DY* = DY - N (DIY = 1)

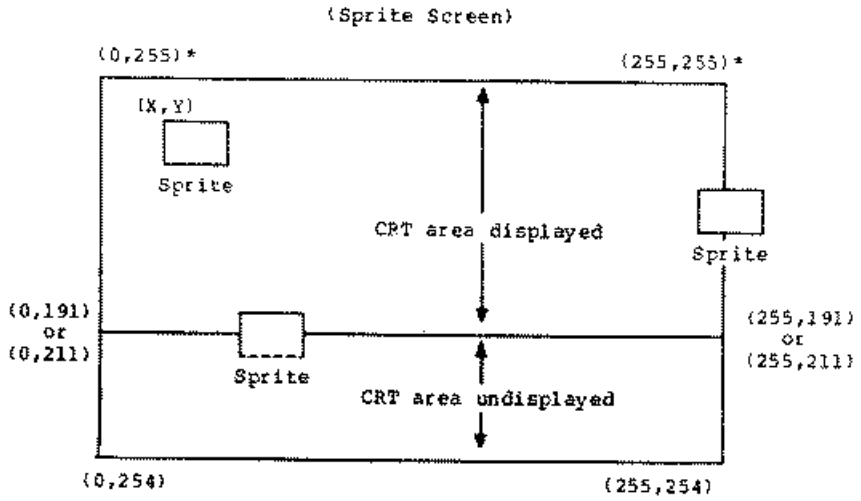
NYB = NY - N

("LINE"command ==> if MAJ=0 then N=N-1)

SPRITES

The MSX-VIDEO can be used to display 32 sprites. The size of the sprites are 8 x 8 dots or 16 x 16 dots. The size in the horizontal direction of a sprite is 1/256 of the screen. The sprites may be placed anywhere on the screen.

Since the sprites are handled in a conceptually independent screen, they do not affect the data within other screens.



*The y-coordinate of the upper edge of a sprite is 255.

The MSX-VIDEO has two sprite display modes. The sprite display mode is automatically selected according to the screen display mode.

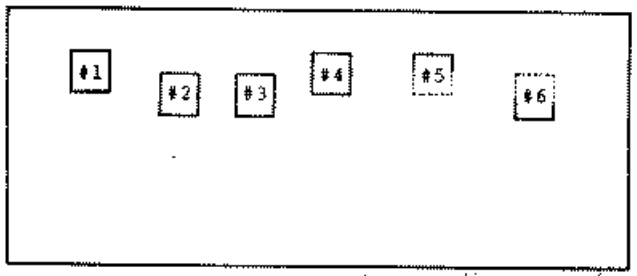
SPRITE MODE 1: GRAPHIC 1, GRAPHIC 2, MULTICOLOR

SPRITE MODE 2: GRAPHIC 3, GRAPHIC 4, GRAPHIC 5,
GRAPHIC 6, GRAPHIC 7

1. SPRITE MODE 1 (G1, G2, MC)

1.1 Characteristics of SPRITE MODE 1

In SPRITE MODE 1, there are 32 sprites, numbered #0 to #31. The sprites assigned the lower numbers have a higher priority. On a single CRT horizontal line, up to 4 sprites with the highest priority are displayed, and the overlapping portions of sprites with lower priorities are not displayed.



When two sprites collide (their pattern color 1 portions have overlapped), this condition may be detected since bit 5 of status register S#0 is set to 1.

In addition, if there are five or more sprites on one horizontal line, bit 6 of status register S#0 will be set to 1, and the lower-order five bits will be set to the number of the fifth sprite.

1.2 SPRITE MODE 1 display (G1, G2, MC)

To display sprites, use the following controls.

- Sprite size — R#1 bit 1
  SI = 1: 16 x 16 dots
  SI = 0: 8 x 8 dots

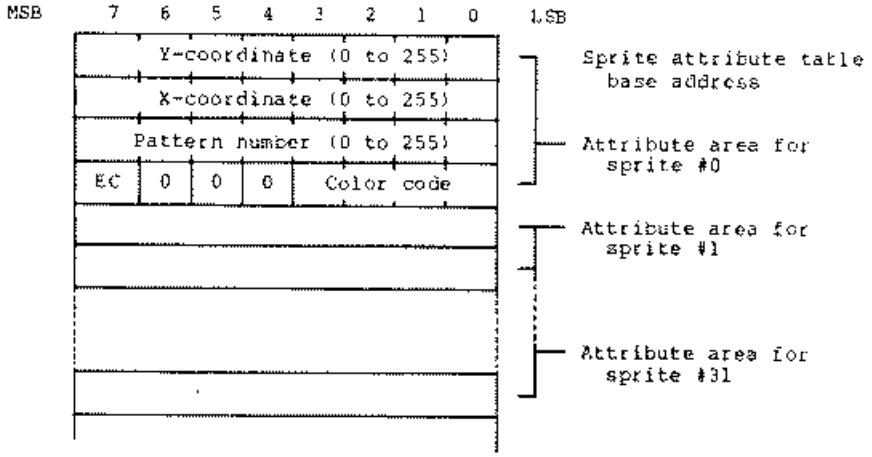
- Sprite magnification — R#1 bit 0
  MAG = 1: Double-size
  MAG = 0: Normal

- Setting the sprite pattern generator table
  Set the sprite's pattern in the Sprite Pattern Generator Table of the VRAM (#0 to #255).

- Setting the sprite attribute table
  Set the sprite's attributes (its coordinates, pattern number, and colors) in the Sprite Attribute Table of the VRAM (#0 to #31).

1.3 Sprite Attribute Table (G1, G2, MC)

The Sprite Attribute Table is an area in the VRAM that contains the display (x- and y-) coordinates, colors, and pattern numbers of the 32 sprites. Each sprite has four bytes of attribute data.



- Y-coordinate (0 to 255)
  Specify the y-coordinate of the sprite.
  If the value for the sprite's y-coordinate is set to 208, all sprites with lower priority will not be displayed. For example, if sprite #10's y-coordinate is set to 208, sprites #10 to #31 will not be displayed.

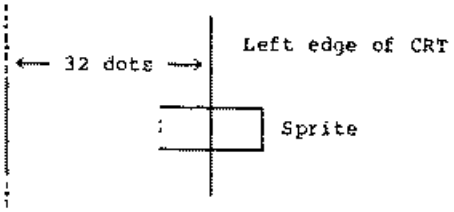
- X-coordinate (0 to 255)
  Specify the x-coordinate of the sprite.

- Pattern number (0 to 255)
  Specify the sprite pattern number in the sprite pattern generator table. If the size of the sprite is 16 x 16, there will be four sprite pattern numbers corresponding to one sprite. In this case, you may specify any one of the four sprite pattern numbers.

  In the above manner, if all sprites are 8 x 8, there will be 256 possible patterns; however, if all sprites are 16 x 16, there will be 64 possible patterns.

- Color code (0 to 15)
  Specify the color code for pattern color 1. The color code for pattern color 0 will be transparent.

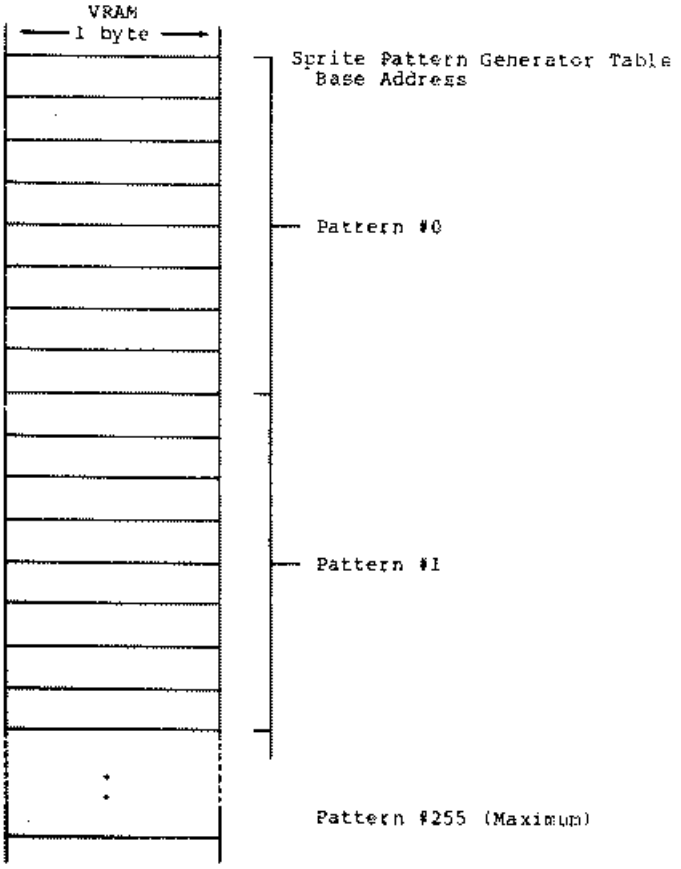
- EC (Early clock)
  When this bit is set to 1, the 32 dots of the sprite are shifted to the left. In other words, when this function is used, the sprite is moved to the left, one dot at a time, from the left edge of the screen.



1.4 Sprite Pattern Generator Table (G1, G2, MC)

The Sprite Pattern Generator Table is an area in the VRAM to specify the sprite patterns (its appearance). The beginning (head) address of this area must be specified in register R#6 (Sprite Pattern Generator Table Base Address Register).

The pattern for each sprite must be written in this area. Eight bytes are used for each pattern, for a total of 256 patterns. Each of the 256 patterns are assigned a sprite pattern number from #0 to #255; if the sprite size is 8 x 8 dots, each sprite has one pattern, and if the sprite size is 16 x 16 dots, each sprite has four patterns.



1.5 Example of Data Setting for Sprite Pattern Generator Table (G1, G2, MC)

(X=1, O=0) N=0, 1, . . . , 255
(Pattern Name Table Base Address for N=0)

```
MSB  LSB
76543210  Address                          Example of 16 x 16 Sprite
OOOOOOOX  8N
OOOOOXXX  8N+1                                 OOOOOOOXXOOOOOOO
OOOXXXXX   .                                   OOOOOXXXXXXOOOOO
OOOOXXXO   .   Pattern number 0                COOXXXXXXXXXXOOO
OOOOXXOX   .                                   OOOOXXXOOXXXOOOO
OOOOXXOX   .                                   OOOOXXOXXOXXOOOO
OOOOXXXO   .                                   OOOOXXOXXOXXOOOO
OOOXXXXX  8N+7                                 OOOOXXXOOXXXOOOO
                                               OOOXXXXXXXXXXOOO
OOXXXXXX                                       OOXXXXXXXXXXXXOO
OXXXXXXX                                       OXXXXXXXXXXXXXXO
XXXXXXXX                                       XXXXXXXXXXXXXXXX
XXXXXXXX      Pattern number 1                 XXXXXXXXXXXXXXXX
OOXXXXOO                                       OOXXXXOOOOXXXXOO
OOOXXOOO                                       OOOXXOOOOOOXXOOO
OOOOOOOO                                       OOOOOOOOOOOOOOOO
OOOOOOOO                                       OOOOOOOOOOOOOOOO

XOOOOOOO
XXXOOOOO
XXXXXOOO
OXXXOOOO      Pattern number 2
XOXXOOOO
XOXXOOOO
OXXXOOOO
XXXXXOOO

XXXXXXOO
XXXXXXXO
XXXXXXXX
XXXXXXXX      Pattern number 3
OOXXXXOO
OOOXXOOO
OOOOOOOO
OOOOOOOO
```

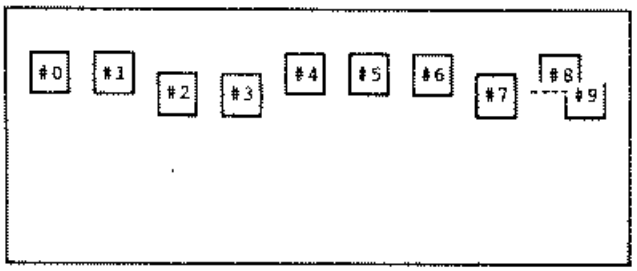
#0	#2
#1	#3

- If the sprite size is 16 x 16, the pattern number specified in the sprite attribute table can be any of the numbers from #0 to #3.

2. SPRITE MODE 2 (G3, G4, G5, G6, G7)

2.1 Characteristics of SPRITE MODE 2

In SPRITE MODE 2, there are 32 sprites, numbered #0 to #31. The sprites assigned the lower numbers have a higher priority. On a single CRT horizontal line, up to 8 sprites with the highest priority are displayed, and the overlapping portions of sprites with lower priorities are not displayed.



When two sprites collide (their solid portions have overlapped), this condition may be detected since bit 5 of status register S#0 is set to 1. When this occurs, the coordinates of the collision will be set in status registers S#3 to S#5.

In addition, if there are nine or more sprites on one horizontal line, bit 6 of status register S#0 will be set to 1, and the lower-order five bits will be set to the number of the ninth sprite.

The colors of the sprite may be specified for each horizontal line.

The sprite priorities may be cancelled by setting the CC bit of the attribute table, and if sprites overlap, a logical OR may be done on the colors of the sprite. In other words, in SPRITE MODE 1, while only two colors may be displayed, in SPRITE MODE 2, four colors may be displayed.

2.2 SPRITE MODE 2 display (G3, G4, G5, G6, G7)

To display sprites, use the following controls.

- **Sprite size** R#1 bit 1
  SI = 1: 16 x 16 dots
  SI = 0: 8 x 8 dots

- Sprite magnification R#1 bit 0
  MAG = 1: Double-size
  MAG = 0: Normal

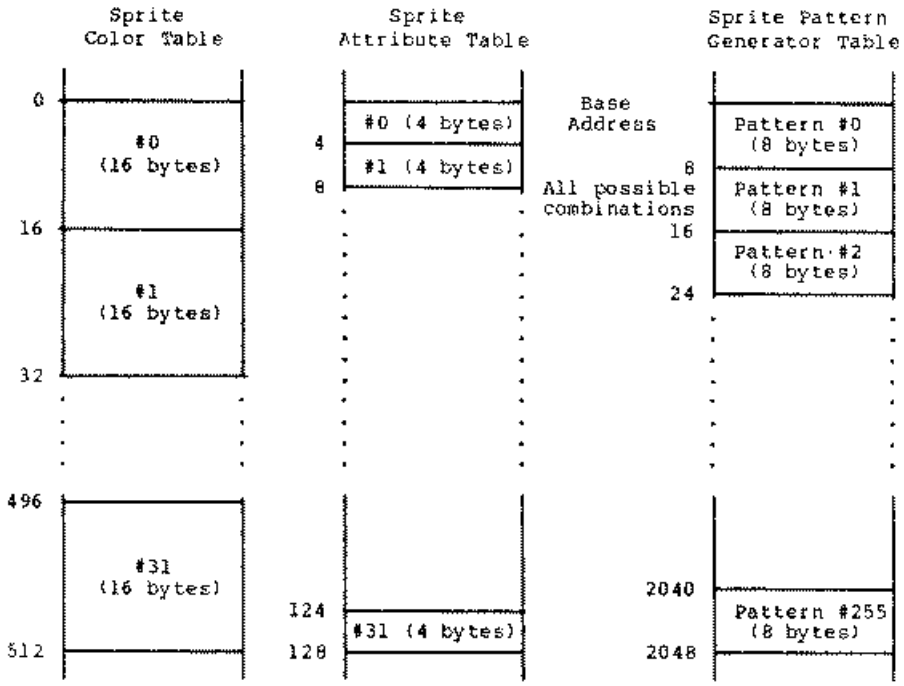
- Sprite display R#8 bit 1
  SPD = 1: Disable
  SPD = 0: Enable

- Setting the sprite pattern generator table
  Set the sprite's pattern in the Sprite Pattern Generator Table of the VRAM (#0 to #255).

- Setting the sprite color table
  Set the sprites' color, EC, CC, and IC in separate lines of the Sprite Color Table of the VRAM.

- Setting the sprite attribute table
  Set the sprite's attributes (its coordinates, pattern number, and colors) in the Sprite Attribute Table of the VRAM (#0 to #31).

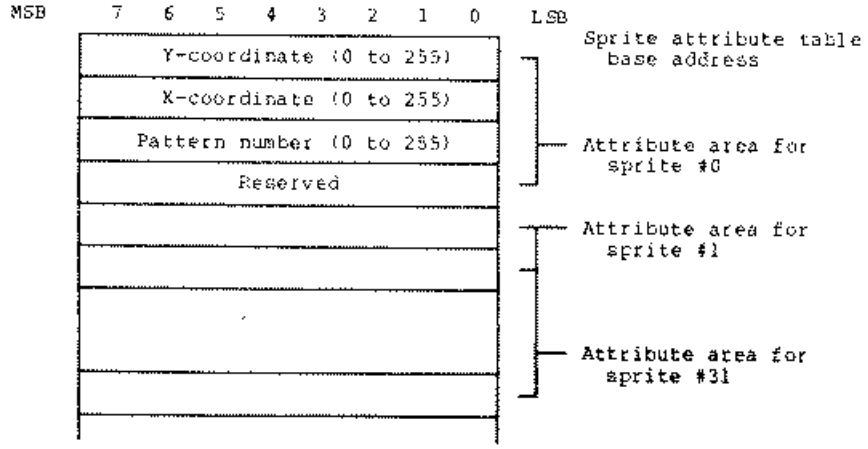
2.3 Relationships between the VRAM Tables (G3,G4,G5,G6,G7)



2.4 Sprite Attribute Table (G3, G4, G5, G6, G7)

The Sprite Attribute Table is an area in the VRAM that contains the

display (x- and y-) coordinates, and pattern numbers of the 32 sprites. Each sprite has four bytes of attribute data.

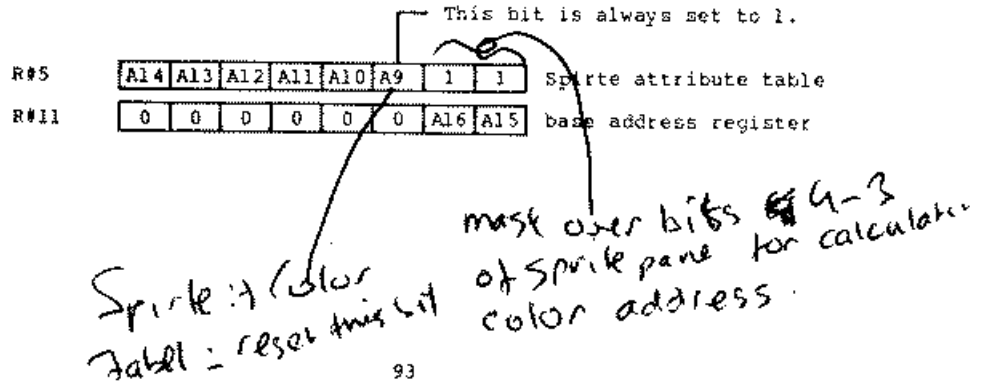


- Y-coordinate (0 to 255)
  Specify the y-coordinate of the sprite.
  If the value for the sprite's y-coordinate is set to 216, all sprites with lower priority will not be displayed. For example, if sprite #10's y-coordinate is set to 216, sprites #10 to #31 will not be displayed.

- X-coordinate (0 to 255)
  Specify the x-coordinate of the sprite.

- Pattern number (0 to 255)
  Specify the sprite pattern number in the sprite pattern generator table. If the size of the sprite is 16 x 16, there will be four sprite pattern numbers corresponding to one sprite. In this case, you may specify any one of the four sprite pattern numbers.

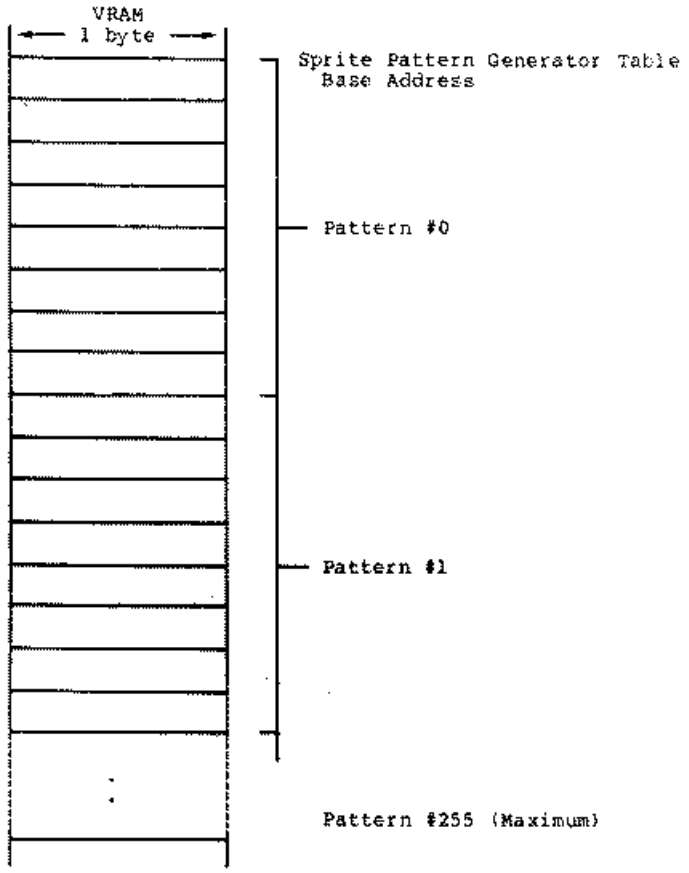
  In the above manner, if all sprites are 8 x 8, there will be 256 possible patterns; however, if all sprites are 16 x 16, there will be 64 possible patterns.



2.5 Sprite Pattern Generator Table (G3,G4,G5,G6,G7)

The Sprite Pattern Generator Table is an area in the VRAM to specify the sprite patterns (its appearance). The beginning (head) address of this area must be specified in register R#6 (Sprite Pattern Generator Table Base Address Register).

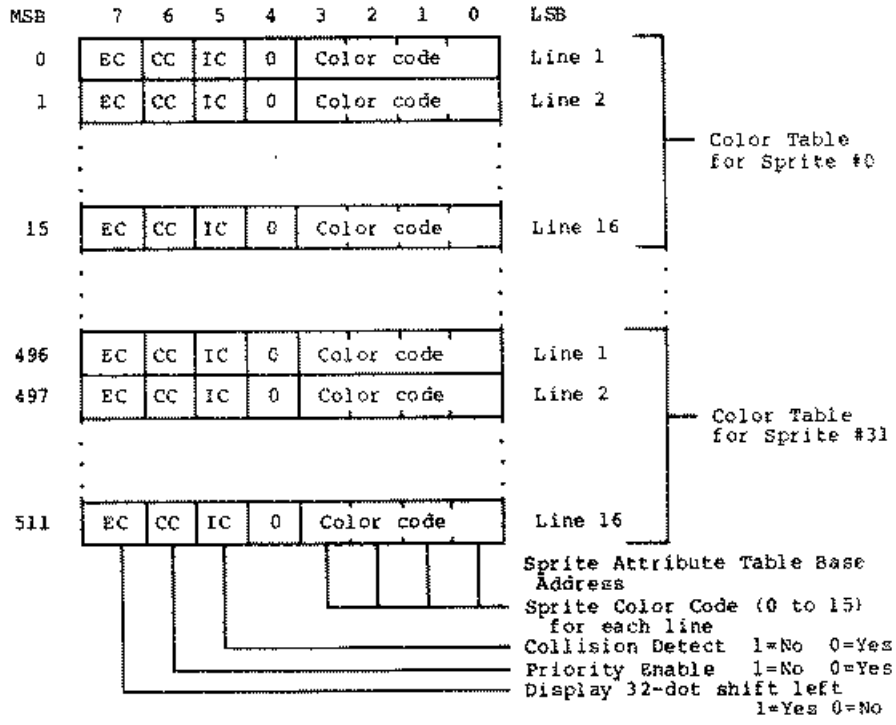
The pattern for each sprite must be written in this area. Eight bytes are used for each pattern, for a total of 256 patterns. Each of the 256 patterns are assigned a sprite pattern number from #0 to #255; if the sprite size is 8 x 8 dots, each sprite has one pattern, and if the sprite size is 16 x 16 dots, each sprite has four patterns.



2.6 Sprite Color Table (G3, G4, G5, G6, G7)

- In SPRITE MODE 2, the color for sprite color pattern 1 may be specified on each line (sprite color pattern 0 will always be transparent). In addition, the sprite priority, collision detection, and EC (Early Clock) may be cancelled or enabled.

- The base address of the Sprite Color Table will always be automatically calculated by subtracting 512 (decimal) from the base address of the Sprite Attribute Table. done by resetting bit 2 R#

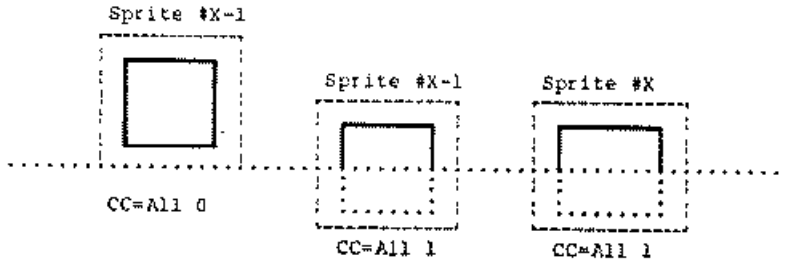


2.7 About the Sprite Priority Order

In SPRITE MODE 2, if the CC bit of the Color Table is set to 1, the sprite priority order is cancelled.

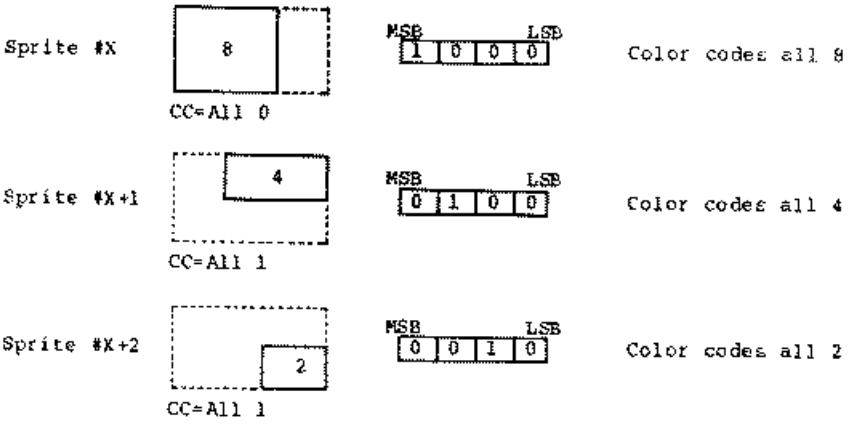
In the above manner, the portions where CC is set to 1 (for each line) will be displayed only on horizontal lines where sprites with a lower number exist. This is diagrammed in the figure below.

Note that in this case, if there are more than 8 sprites on the same line, the ninth sprite and above will not be displayed as explained earlier.

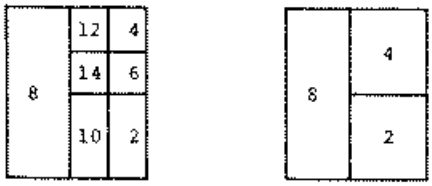


For the portions of the sprite in which CC is set to 1, even if a sprite having the number which is closest to that sprite has CC=0 overlaps the sprite, the collision is not detected. For overlapping sprites, a logical OR is done on the color codes for display.

Example of three sprites being displayed in seven colors



Example of overlapping the above sprites



In SPRITE MODE 2
-No collision detection

In SPRITE MODE 1
-Collision detection

2.8 Sprite Collision

If CC is set to 0 and the sprite with color portion non-0 overlaps, a sprite collision is detected. If sprite collision is detected, bit 5 of status register S#0 is set to 1. This bit is reset to 0 when S#0 is read.

	MSB 7	6	5	4	3	2	1	0 LSB
S#0			C					

If a sprite collision occurs and neither the mouse flag (MO) nor the light pen flag (LP) of register R#8 are set, status registers S#3 to S#6 will be set to the coordinates of the collision.

	MSB 7	6	5	4	3	2	1	0 LSB	
S#3	X7	X6	X5	X4	X3	X2	X1	X0	Collision x-coordinate
S#4	1	1	1	1	1	1	1	X8	
S#5	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	Collision y-coordinate
S#6	1	1	1	1	1	1	Y9	Y8	

If status register S#5 is read, the contents of status registers S#3 to S#6 are reset.

The values that are contained in status registers S#3 to S#5 will contain offsets according to the following formulas.

X (S#4,S#3) Y (S#6,S#5)

Collision coordinates

X=XC+12 Y=YC+8

3. Setting the Sprite Colors

In all GRAPHICS modes other than GRAPHICS 7 mode, the color codes for sprites are common to all modes. The sprite display color is determined by the value in the palette register.

In GRAPHICS 7 mode, the sprite colors are fixed and are not affected by the palette register. The sprite colors in GRAPHICS 7 mode are shown in the following table.

Color code				Green			Red			Blue		
C3	C2	C1	C0	G2	G1	G0	R2	R1	R0	B2	B1	B0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	1	1	0	0	0
0	0	1	1	0	0	0	0	1	1	0	1	0
0	1	0	0	0	1	1	0	0	0	0	0	0
0	1	0	1	0	1	1	0	0	0	0	1	0
0	1	1	0	0	1	1	0	1	1	0	0	0
0	1	1	1	0	1	1	0	1	1	0	1	0
1	0	0	0	1	0	0	1	1	1	0	1	0
1	0	0	1	0	0	0	0	0	0	1	1	1
1	0	1	0	0	0	0	1	1	1	0	0	0
1	0	1	1	0	0	0	1	1	1	1	1	1
1	1	0	0	1	1	1	0	0	0	0	0	0
1	1	0	1	1	1	1	0	0	0	1	1	1
1	1	1	0	1	1	1	1	1	1	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1

TP and Sprites

By controlling TP (Bit 5 of register R#8), color code 0 will be specified in the following manner. The color will affect the sprites.

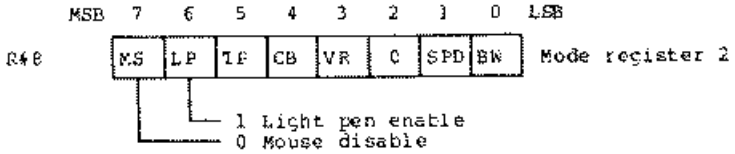
TP=0 Color code 0 will be treated as invisible.
Sprite color part 0 will not be displayed, and if sprites overlap, a sprite collision will not be detected.

TP=1 Color code 0 will be the color that is specified in the palette register. (In GRAPHICS 7 mode only, R=0, G=0, and B=0 will be always set).
If the sprite color part 0 overlaps, a sprite collision will be detected.

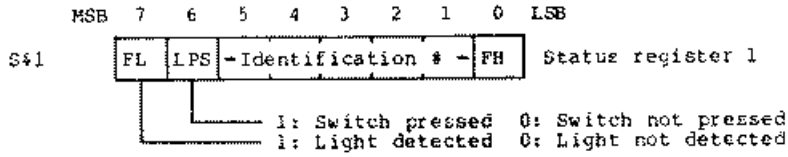
POINTING DEVICES

1. Light pen

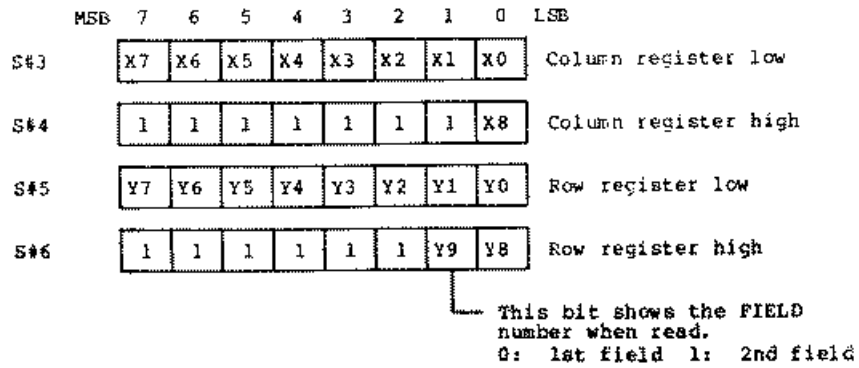
We will explain the light pen function of the MSX-VIDEO below. To use the light pen, set bit 7 of register R#8 to 0 and bit 6 to 1.



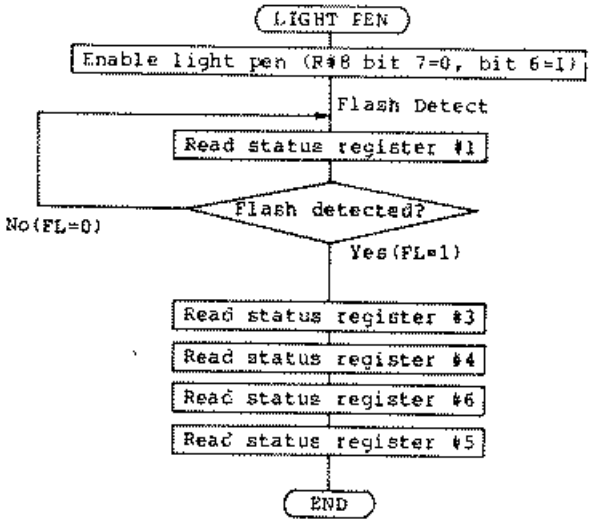
In addition, if you want to enable an interrupt when the light pen has detected light, set bit 5 of register R#0 to 1. This interrupt is reset if status register S#1 is read.



The coordinates at which the light pen detected light will be set in status registers S#3 to S#6. The data set in these registers are correct as long as status register S#5 is not read.

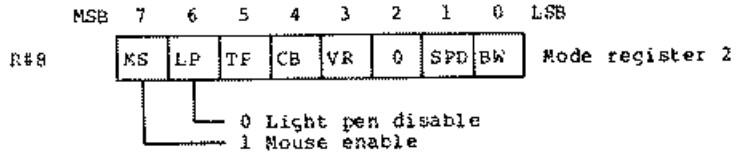


Flowchart of using light pen to detect coordinates



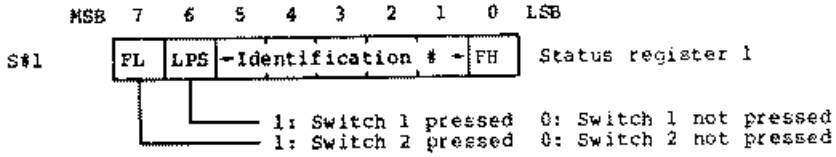
2. Mouse

We will explain the mouse function of the MSX-VIDEO below. Because the mouse uses the Color Bus of the MSX-VIDEO, when you use the mouse, you cannot use the MSX-VIDEO Color Bus for any other purpose. To use the mouse, set bit 7 of register R#8 to 1 and bit 6 to 0.

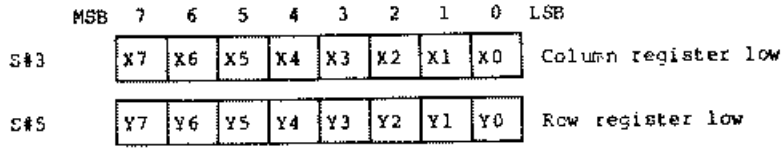


When bit 7 of register R#8 is set to 1, the direction of the Color Bus is automatically changed to input.

You may know if the mouse button has been switched ON and OFF by reading status register S#1. These register will be reset to 0 after being read.



The relative coordinates of the mouse's movement are set as two's complement data in status registers S#3 and S#5.



When 3 or 5 is set in register R#15, the mouse count is not done.

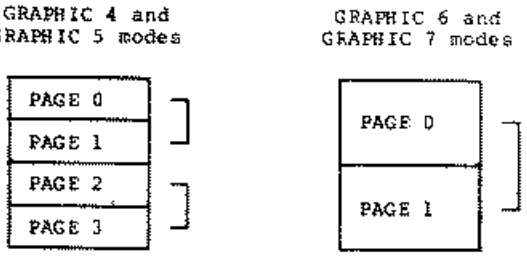
When status registers S#3 and S#5 are read or when counting is to begin, the value in register R#15 must be changed.

SPECIAL FUNCTIONS

1. Alternate display of two graphics screen pages

Two graphics screen pages may be alternately displayed (in GRAPHIC 4 to GRAPHIC 7 modes) automatically.

- The pages that will be alternately displayed are as follows.



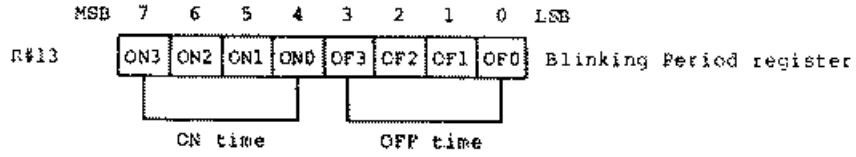
1.1 How to use register R#13

A display time period of between 166 ms and 2053 ms may be specified for each page.

- Specify the odd page in the pattern name table base address (register R#2).

- Specify the ON time (the interval in which the even page is displayed) and the OFF time (the interval in which the odd page is displayed) in register R#13.

* For the values used in the timings, refer to the chart in the section on TEXT 2 MODE.

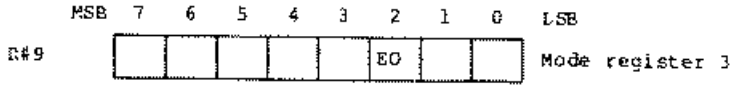


1.2 How to use the EO bit

The EO bit, bit 2 of register R#9, is used to alternately display the two graphics screen pages at 60 Hz.

- Set the odd page in the Pattern Name Table Base Address (register R#2).

- Set bit 2 of register R#9 to 1.



2. Interlace display

The MSX-VIDEO has an interlace display function.

2.1 Displaying the first and second fields on the same page

- Set the IL bit, bit 3 of register R#9, to 1.

2.2 Displaying the even page in the first field and the odd page in the second field

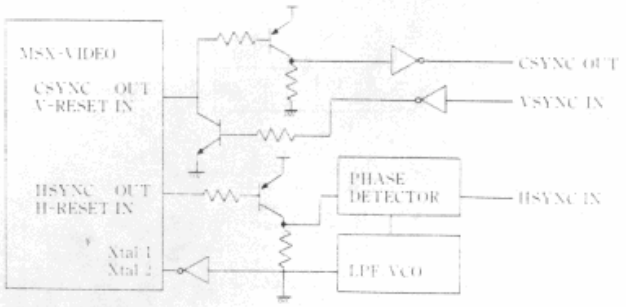
- Set the IL bit, bit 3 of register R#9, to 1.
  Set the EO bit, bit 2 of register R#9, to 1.

- Set the odd page in the pattern name table base address (register R#2).

3. External Synchronization

3.1 GENLOCK method

The GENLOCK method detects the phase differnce between the HSYNC output signal of the MSX-VIDEO and an external HSYNC signal and feeds it back to the system clock.



3.2 Choosing the synchronization mode

The S1 and S0 bits, bits 4 and 5 of register R#9, are used to set the synchronization mode of the MSX-VIDEO.

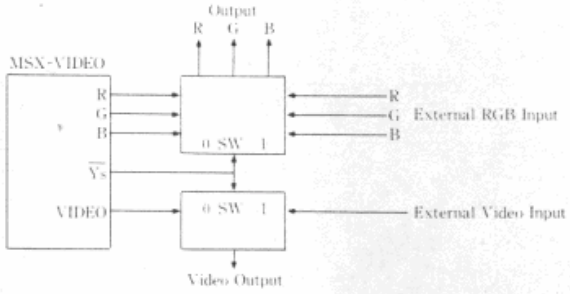
	MSB 7	6	5	4	3	2	1	0 LSB	
R#9			S1	S0					Mode register 3

S1	S0	Sync mode	*Ys	Purpose
0	0	PC SYNC	Selects Normal MSX-VIDEO (0)	Display the MSX-VIDEO screen
0	1	STD SYNC	Appears for transparent parts	Superimpose, digitize, etc.
1	0	STD SYNC	Selects external signal (1)	Display external screen
1	1	--	--	--

4. Superimpose

The MSX-VIDEO can originate a switch signal (*Ys) for superimposing an external video signal and the output of the MSX-VIDEO.

- The synchronization of the MSX-VIDEO must be adjusted to the signal to be superimposed.
- To input an interlaced video signal, the MSX-VIDEO may also be set up for an interlace display.



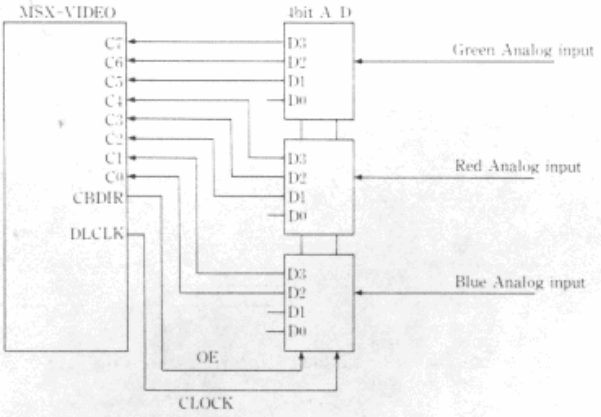
*Ys selects the external video signal when scanning the transparent portion of the MSX-VIDEO screen.

5. Digitize function

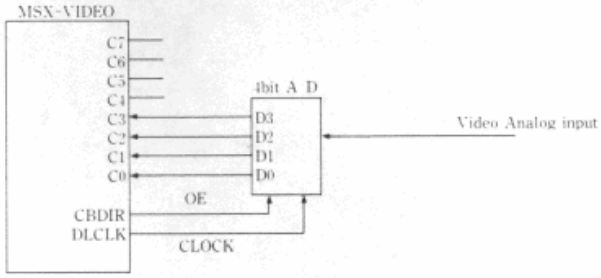
The MSX-VIDEO has a function to read external data over the color bus into the VRAM.

- The digitize function is available in GRAPHIC 4 to GRAPHIC 7 modes only.
- When using a digitizer, the MSX-VIDEO must be synchronized with the external signal as necessary.

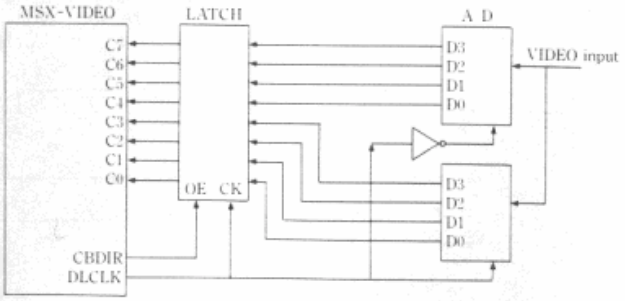
Digitize Block Diagram (GRAPHIC 7 mode)



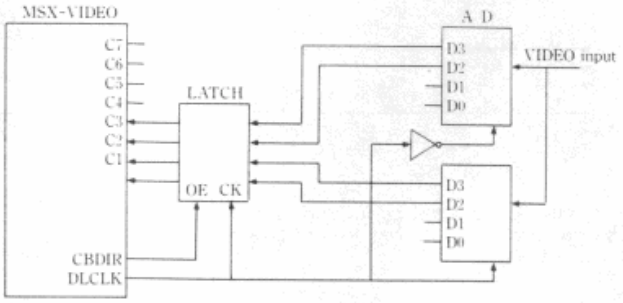
Digitize Block Diagram (GRAPHIC 4 mode)



Digitize Block Diagram (GRAPHIC 6 mode)

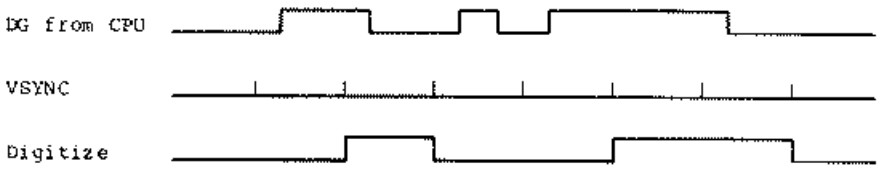


Digitize Block Diagram (GRAPHIC 5 mode)

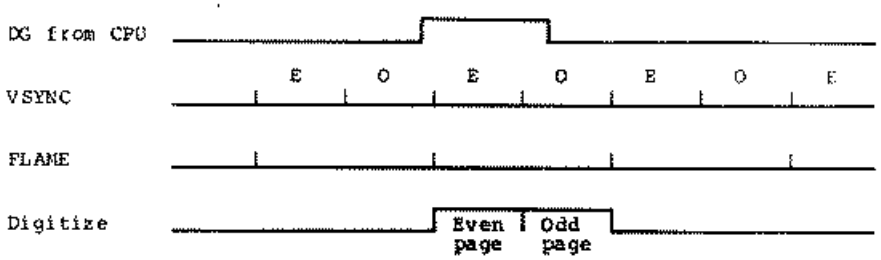


- Digitize control

When the DG bit (bit 6 of register R#0) is set to 1, the MSX-VIDEO initiates input of data from the color bus. This action is automatically performed internally by synchronization with VSYNC.



- Successive reads into the first and second fields



- Bit mask of the color bus

The color bus may be a masked using a register R#7. When a bit of R#7 is set to zero, the input value of the corresponding bit on the color bus is set to zero.

	MSB 7	6	5	4	3	2	1	0 LSB
R#7	M7	M6	M5	M4	M3	M2	M1	M0

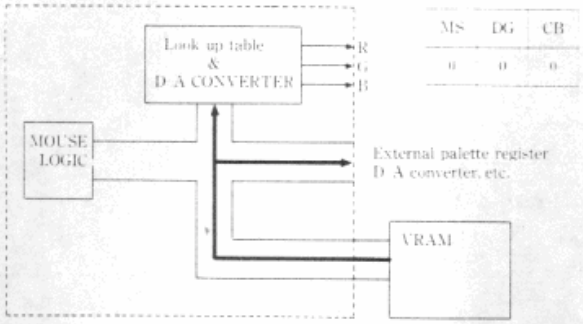
The relationships of the bit vary according to the mode as follows.

	color bus							
	C7	C6	C5	C4	C3	C2	C1	C0
GRAPHIC 4 MODE	-	-	-	-	M3	M2	M1	M0
GRAPHIC 5 MODE	-	-	-	-	M3	M2	M1	M0
GRAPHIC 6 MODE	M3	M2	M1	M0	M3	M2	M1	M0
GRAPHIC 7 MODE	M7	M6	M5	M4	M3	M2	M1	M0

6. Color bus

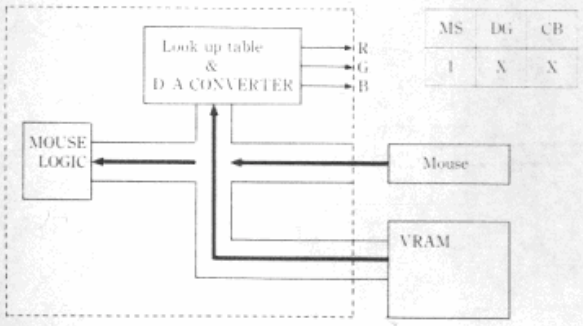
The color bus of the MSX-VIDEO is controlled by the MS, DG, and CB bits. The bits and the data flow will be illustrated below.

6.1 Normal display



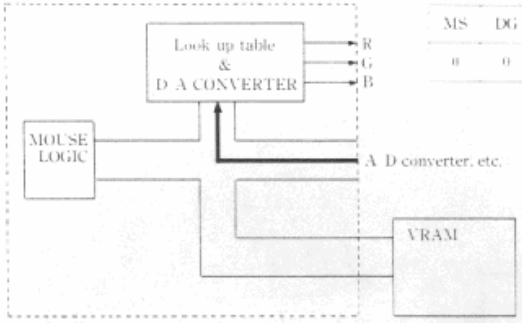
MS	DG	CB
0	0	0

6.2 Mouse



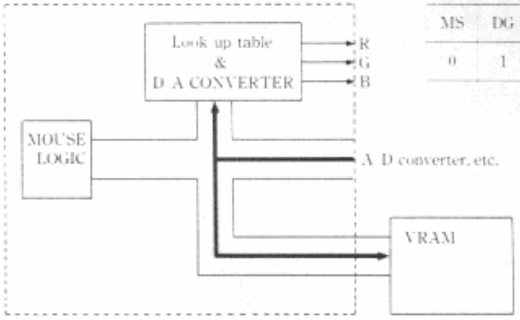
MS	DG	CB
1	X	X

6.3 Displaying external signals



MS	DG	CB
0	0	1

6.4 Digitize



MS	DG	CB
0	1	X

PART 2

MSX-VIDEO DATA PROCESSOR
LSI DATA SHEET

V9938 MSX-VIDEO Data Sheet

1. MSX-VIDEO

1-1 Overview

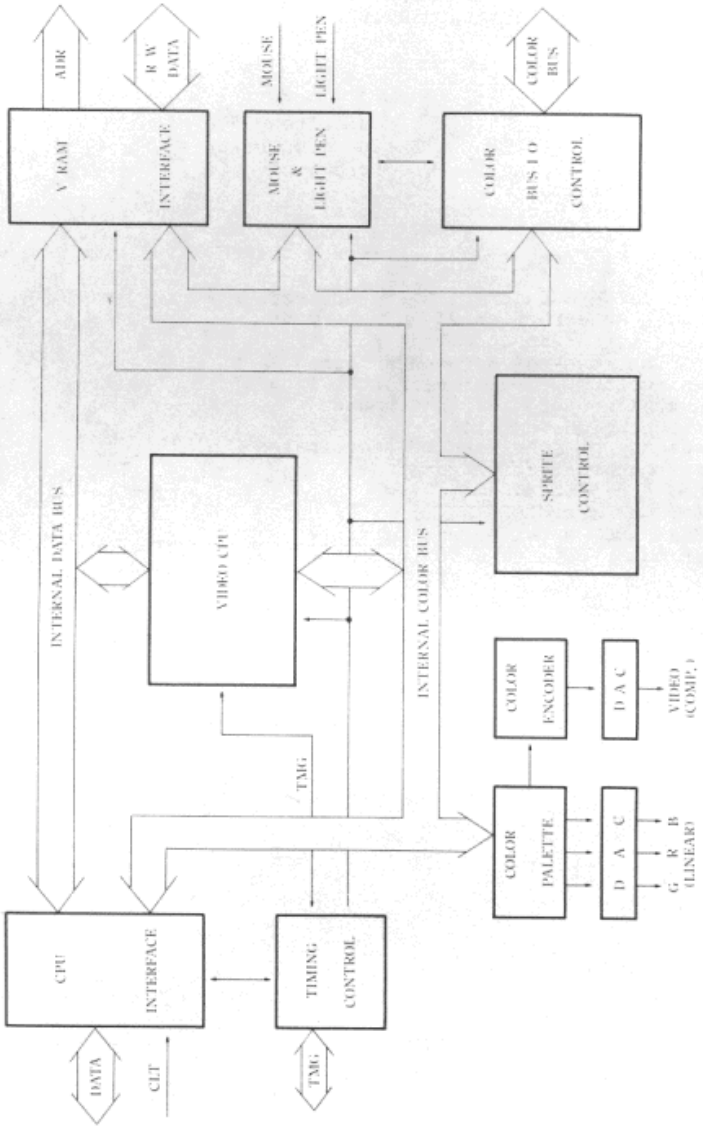
V9938 (MSX-VIDEO) is a N-channel Silicon Gate MOS, 64-pin shrink DIL plastic package Video Display Processor (VDP). The V9938 is software-compatible with the TMS9918A.

1-2 Features

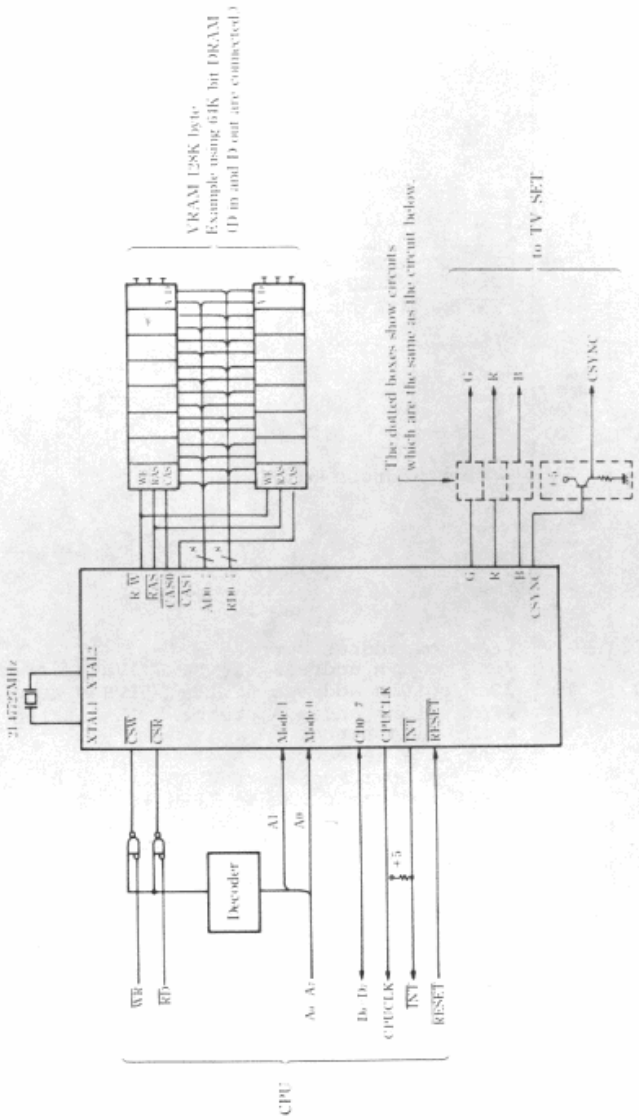
- 5V single power
- Output of both Linear RGB and Composite Video signals
- 512 colors possible with built-in color palette
- Maximum 512 x 424 pixels, 16 colors
- Bit-mapped graphics
- 256 simultaneously-displayed colors
- Supports 16K- to 128K-byte video memory
- May be used with 16K x 1 bit, 16K x 4 bit, 64K x 1 bit, and 64K x 4 bit DRAMs
- DRAM auto refresh function: 256 addresses, 4 ms
- May support an extension video memory
- Interfaces for mouse and light pen
- Maximum 8 sprites per horizontal line
- Different color for each horizontal line of the sprite
- Area move, line, and search commands
- Logical operation function
- Addresses may be specified by coordinates
- External synchronization possible
- Superimpose possible
- Digitization possible
- Multi-MSX-VIDEO architecture possible
- External color palette by using color bus

Note: In this manual, negative logic will be indicated by an asterisk preceding the signal names.

MSX-VIDEO Block diagram

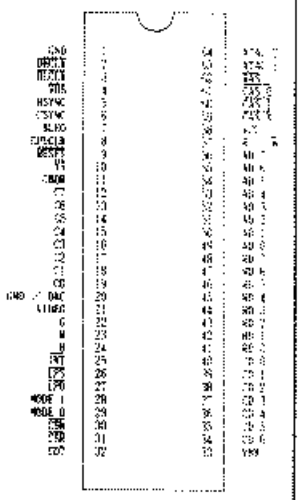


MSX-VIDEO circuit example



2. Pin assignments and functions

Pin name	Pin no.	I/O	Function
CD0 LSB	40	I/O	- CPU data bus
CD1	39	I/O	
CD2	38	I/O	
CD3	37	I/O	
CD4	36	I/O	
CD5	35	I/O	
CD6	34	I/O	
CD7 MSB	32	I/O	-
MODE 0	29	I	- CPU interface mode
MODE 1	28	I	- select
*CSR	31	I	CPU-MSX-VIDEO read strobe
*CSW	30	I	CPU-MSX-VIDEO write strobe
RD0 LSB	41	I/O	- VRAM data bus
RD1	42	I/O	
RD2	43	I/O	
RD3	44	I/O	
RD4	45	I/O	
RD5	46	I/O	
RD6	47	I/O	
RD7 MSB	48	I/O	-
AD0 LSB	49	O	- VRAM address bus
AD1	50	O	
AD2	51	O	
AD3	52	O	
AD4	53	O	
AD5	54	O	
AD6	55	O	
AD7 MSB	56	O	-
*RAS	62	O	VRAM row address strobe
*CAS0	61	O	VRAM column address strobe 0 (VRAM first half)
*CAS1	60	O	VRAM column address strobe 1 (VRAM second half)
*CASX	59	O	VRAM column address strobe X (Expansion VRAM)
R/*W	57	O	VRAM write strobe
*VDS	4	O	VRAM data select *VDS = Low: Access to VRAM is for display data *VDS = High:Access to VRAM is for other than the above
VIDEO	21	O	Composite video signal output
G	22	O	- Linear RGB signal output
R	23	O	
B	24	O	-
*YS	10	O	Signal to switch between MSX-VIDEO RGB output and external video signal (When superimposing) *YS = High: MSX-VIDEO output transparent *YS = Low: MSX-VIDEO output opaque
BLEO	7	O	Tri-level output (Open drain); primary/secondary field and blanking interval High: Secondary field, active Middle: First field, active Low: Blanking interval



HSYNC	5	I/O	Tri-level logic; upper levels (high to middle) for output, lower levels (middle to low) for input High: Non-HSYNC, color burst period Middle: HSYNC or non-color burst period Low: HSYNC input
CSYNC	6	I/O	Tri-level logic; High: Composite SYNC output Low: VSYNC input
CBDR	11	O	Color bus direction High: Input Low: Output
C0 LSB	19	I/O	- Color bus Usually the color code is output; during digitize, this bus is used as the input port. The high-order bits are for mouse input when using the mouse. C4 = XA C5 = XB - C6 = YA C7 = YB
C1	18	I/O	
C2	17	I/O	
C3	16	I/O	
C4	15	I/O	
C5	14	I/O	
C6	13	I/O	
C7 MSB	12	I/O	
*LPS	26	I	Light pen or mouse SW input Low: SW on High: SW off
*LPD	27	I	Light pen detection input or mouse SW input Low: Light detected or SW on High: Any other condition
*DHCLK	2	O	Dot clock output for high resolution; 10.74 MHz open drain
*DLCLK	3	I/O	Dot clock output for low resolution; 5.37 MHz open drain Input may also be done the mode register. This is used for a multi-MSX-VIDEO system.
XTAL 1	63	I	Connect to XTAL or use for connection of an external oscillator
XTAL 2	64	I	
CPUCLK	8	O	Outputs 1/6 of XTAL frequency
*INT	25	O	CPU interrupt output; open drain output Low: interrupt
*RESET	9	I	Initialize all MSX-VIDEO circuits
VCC	58	I	5V power supply
GND	1	I	Ground 0V
GND DAC	20	I	Ground 0V
VBB	33	O	Back bias (No connection)

3. Electrical characteristics and timing chart

3-1 Absolute maximum ratings

Symbol	Parameter	Rating	Unit
VCC	Supply voltage	-0.5 to +7.0	V
Vin	Input voltage	-0.5 to +7.0	V
Ts	Storage temperature	-50 to +125	℃
To	Operating temperature	0 to + 70	℃

3-2 Recommended operating conditions

Symbol	Parameter	Min.	Typ.	Max.	Unit
VCC	Supply voltage	4.75	5.00	5.25	V
Vss	Supply voltage		0		V
Ta	Ambient temperature	0		70	℃
VIL 1	Low-level input voltage (group 1)	-0.3		0.8	V
VIL 2	Low-level input voltage (group 2)	-0.3		0.8	V
VIL 3	External clock low-level input voltage (group 3)	-0.3		0.8	V
VIH 1	High-level input voltage (group 1)	2.2		VCC	V
VIH 2	High-level input voltage (group 2)	2.2		VCC	V
VIH 3	External clock high-level input voltage (group 3)	3.5		VCC	V

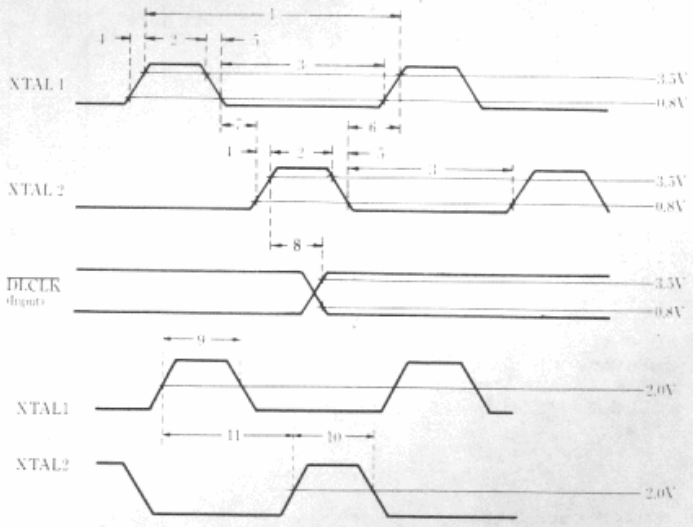
Notes:
Group 1: *CSR, RD0 to RD7, C0 to C7, *LPS, *LPD, *RESET, *DLCLK
Group 2: CD0 to CD7, MODE 0, MODE 1, *CSW
Group 3: XTAL 1, XTAL 2
The video signals are listed separately
Versions A and B of the MSX-VIDEO differ in the following items:

Symbol	Parameter	Min.	Typ.	Max.	Unit
VIL 2	Low-level input voltage (group 2)	-0.3		0.6	V
VIL 3	External clock low-level input voltage (group 3)	-0.3		0.3	V

3-3 Electrical characteristics under recommended operating conditions

External input clock timing

No.	Symbol	Parameter	Min.	Typ.	Max.	Unit
1	fXTAL	XTAL clock frequency	20.26	21.48	22.55	MHz
2	TXWH	XTAL clock high-level pulse width	5			ns
3	TXWL	XTAL clock low-level pulse width	5			ns
4	TXR	XTAL clock rise time			10	ns
5	TXF	XTAL clock fall time			10	ns
6	TXD21	XTAL clock delay time 2 ⟶ 1	0			ns
7	TXD12	XTAL clock delay time 1 ⟶ 2	0			ns
8	TLIXD	*DLCLK (input) - XTAL clock delay time	20		50	ns
9	TW1	XTAL1 pulse width	12			ns
10	TW2	XTAL2 pulse width	20			ns
11	TPD	XTAL1-XTAL2 relative delay time	15		24	ns



*RESET Input timing

No.	Symbol	Parameter	Min.	Typ.	Max.	Unit
1	TRESET	*RESET low-level pulse width	10			ms

DC characteristics

Symbol	Parameter	Condition	Min.	Max.	Unit
VOL4	Low-level output voltage (group 4)	IOL = 1.6 mA		0.4	V
VOL5	Low-level output voltage (group 5)	IOL = 1.6 mA		0.4	V
VOL6	Low-level output voltage (group 6)	IOL = 10 mA		0.4	V
VOL7	Low-level output voltage (group 7)	IOL = 1.6 mA		0.4	V
VOH4	High-level output voltage (group 4)	IOH = 100 uA	2.4		V
VOH5	High-level output voltage (group 5)	IOH = 60 uA	2.7		V
ILI	Input leak current			10	uA
ILO	Output leak current (floating)			25	uA
ICC	Current consumption			230	mA

Notes:
Group 4: CD0 to CD7, RD0 to RD7, AD0 to AD7, *VDS, CBDR, CPUCLK, C0 to C7
Group 5: *RAS, *CAS0, *CAS1, *CASX, R/*W
Group 6: *DLCLK, *DHCLK
Group 7: *INT

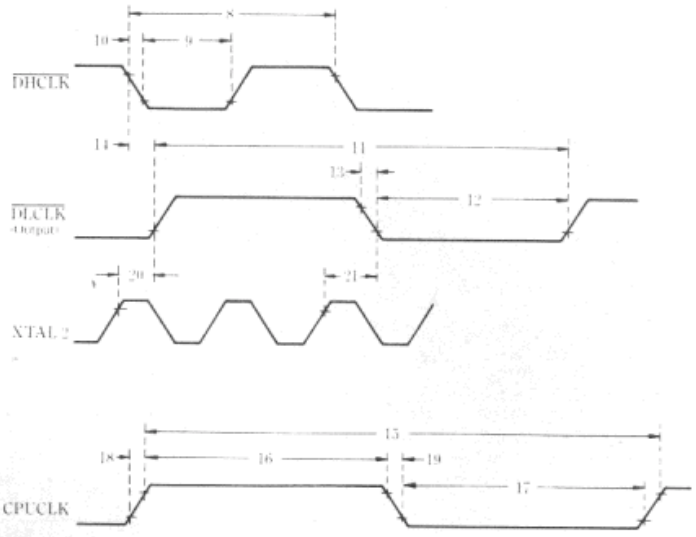
Input/output power capacities

Symbol	Parameter	Condition	Min.	Max.	Unit
CIN	Input power capacity	VIN = 0 V		10	pF
COUT	Output power capacity	VOUT = 0 V		10	pF

External output clock timing

No.	Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
8	fDHCLK	*DHCLK frequency		10.13	10.74	11.28	MHz
9	THWL	*DHCLK low-level pulse width		20			ns
10	THF	*DHCLK fall time				25	ns
11	fDLCLK	*DLCLK frequency	CL = 50 pF	5.06	5.37	5.64	MHz
12	TLOWL	*DLCLK (output) low-level pulse width		60			ns
13	TLOF	*DLCLK (output) fall time				15	ns
14	THLOD	*DHCLK-*DLCLK (output) delay time		-15		15	ns
15	fCPUCLK	CPUCLK frequency		3.37	3.58	3.76	MHz
16	TCWH	CPUCLK high-level pulse width	CL = 100 pF	110			ns
17	TCWL	CPUCLK low-level pulse width		110			ns
18	TCR	CPUCLK rise time				25	ns
19	TCF	CPUCLK fall time				25	ns
20	TLOHXD	*DLCLK (output) high-XTAL delay time	CL = 50 pF	20		50	ns
21	TLOLXD	*DLCLK (output) low-XTAL delay time		20		50	ns

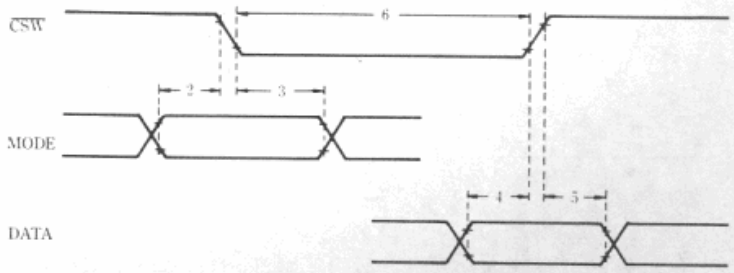
Note: **The values shown for *DHCLK and *DLCLK assume that RL = 1 k ohm.**



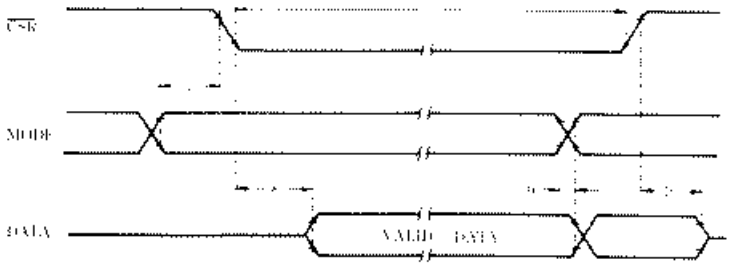
External output clock timing

CPU-MSX-Video Interface

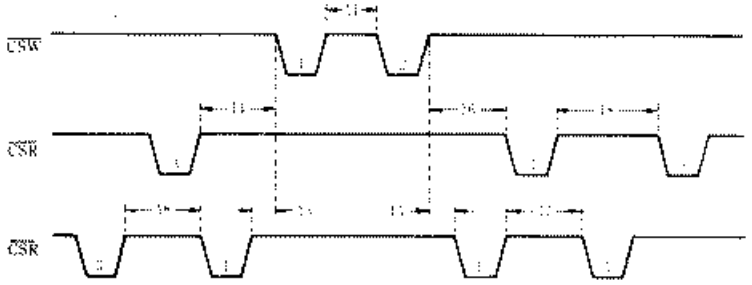
No.	Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
1	TASR	Address setup time (related to *CSR)		0			ns
2	TASW	Address setup time (related to *CSW)		30			ns
3	TAHW	Address hold time		50			ns
4	TDSW	Data setup time		30			ns
5	TDHW	Data hold time		30			ns
6	TCSW	*CSW pulse width		186	700	2000	ns
7	TCSR	*CSR pulse width		186	700	2000	ns
8	TRAC	Data access time	CL =		100	150	ns
9	TPVX,A	Data invalid time	300 pF	0			ns
10	TPVX	Data disable time			65	100	ns
11	TW1W	*CSW pulse width high, 2nd-1st, 1st-2nd byte		2			us
12	TW2W	*CSW pulse width high, 2nd-3rd, 3rd-3rd, 3rd-1st byte		8			us
13	TS1RW	*CSR-*CSW setup time, 1st-1st byte		2			us
14	TS2RW	*CSR-*CSW setup time, 3rd-1st byte		8			us
15	TS1WR	*CSW-*CSR setup time, 2nd-1st byte		2			us
16	TS2WR	*CSW-*CSR setup time, 2nd-3rd byte		8			us
17	TW1R	*CSR pulse width high, 1st-1st byte		2			us
18	TW2R	*CSR pulse width high, 3rd-1st, 3rd-3rd byte		8			us



CPU-MSX-VIDEO write cycle interface

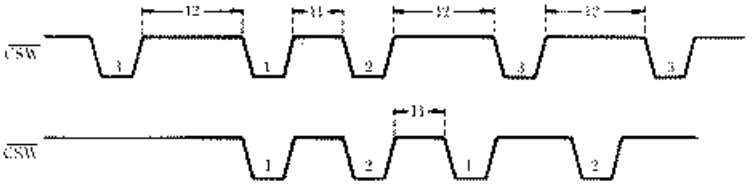


CPU-MSX-VIDEO read cycle interface



Note: The numbers n (where n = 1, 2, 3) marked in the pulses show the byte order (1st, 2nd, 3rd) sent from the CPU.

MSX-VIDEO register read timing



Note: The numbers n (where n = 1, 2, 3) marked in the pulses show the byte order (1st, 2nd, 3rd) sent from the CPU.

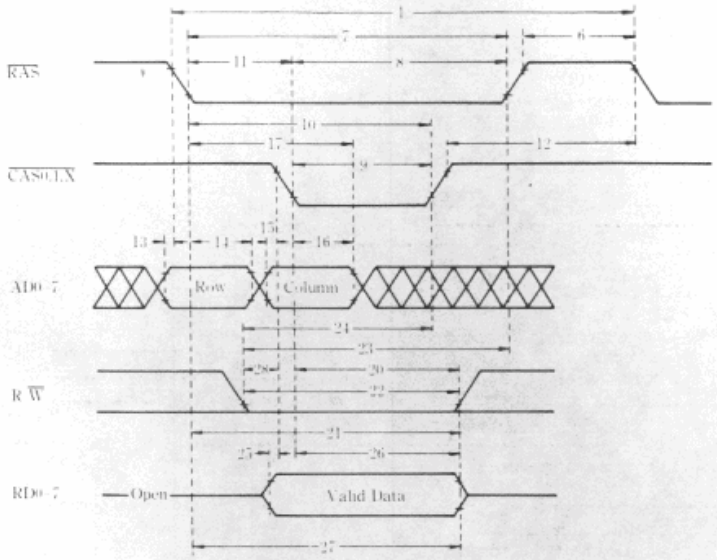
MSX-VIDEO register write timing

MSX-VIDEO-VRAM interface

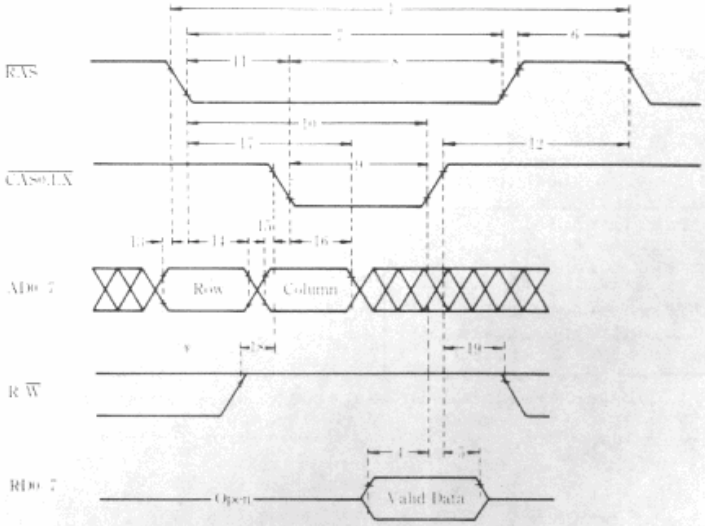
No.	Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
1	TRC	Memory read/write cycle time		266	279		
2	TPC	Page mode cycle time		177	186		
4	TDSC	Read data setup time		20			
5	TDHC	Read data hold time		0			
6	TRP	*RAS precharge time		90			
7	TRAS	*RAS pulse width		130			
8	TRSH	*RAS hold time		60			
9	TCAS	*CAS pulse width		85			
10	TCSH	*CAS hold time		140			
11	TRCD	*RAS-*CAS delay time		40			
12	TCRP	*CAS-*RAS precharge time		90			
13	TRARD	Row address-*RAS delay time		50			
14	TRAH	Row address hold time		12			
15	TCACD	Column address-*CAS delay time		0			
16	TCAH	Column address hold time	CL = 150 pF	100			ns
17	TCAR	Column address hold time (for *RAS)		130			
18	TRCD	Read command-*CAS delay time		30			
19	TRCH	Read command hold time		30			
20	TWCH	Write command hold time		70			
21	TWRH	Write command hold time (for *RAS)		150			
22	TWP	Write command pulse width		120			
23	TRWL	Write command-*RAS read width		150			
24	TCWL	Write command-*CAS read width		120			
25	TDCD	Write data-*CAS delay time		0			
26	TDH	Write data hold time		50			
27	TDHR	Write data hold time (for *RAS)		110			
28	TWCD	Write command-*CAS delay time		30			
29	TCP	*CAS precharge time (page mode cycle)		70			

Note: The above specifications apply to version C of the V9938. The following items apply to Versions A and B. To distinguish Versions A, B, and C, refer to page 134 of this data sheet.

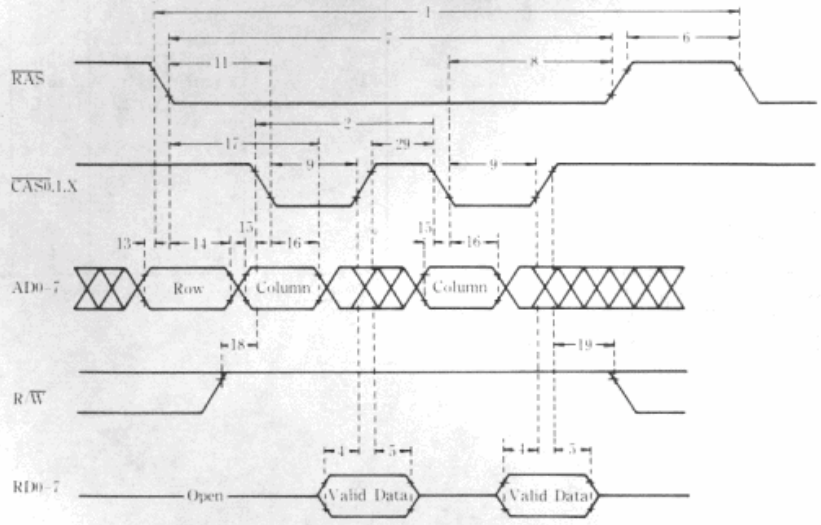
No.	Symbol	Parameter	Version	
			A	B
14	TRAH	Row address hold time	10	10
15	TCACD	Column address-*CAS delay time	-5	0
26	TDH	Write data hold time	30	30
27	TDHR	Write data hold time (for *RAS)	70	70
—	TRSH*	*RAS hold time during read only	15	No rating
—	TRAS*	*RAS width during *RAS refresh only	60	No rating



VRAM write cycle (early write)



VRAM read cycle



VRAM page mode cycle

Composite video signal output level

Versions A and B

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
VWHITE	White level output voltage		2.20	2.55	2.90	V
VWHITE (B/W)	White level output voltage (using Black & white)		2.30	2.65	3.00	V
VBLACK	Black level output voltage		2.00	2.30	2.70	V
VSYNC	Sync level output voltage	RL = 470 ohms	1.90	2.20	2.60	V
VCB	Color burst pulse width		0.14	0.19	0.24	V
VP-P	Electric potential difference between white and sync levels		0.24	0.35	0.44	V
VP-P (B/W)	Electric potential difference between white and sync levels (using black & white)		0.34	0.45	0.54	V

Note: The typical values listed in the above table assume that VCC = 5.00 V and TA = 25 °C.

Version C

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
VWHITE	White level output voltage		2.20	2.60	3.00	V
VWHITE (B/W)	White level output voltage (using Black & white)		2.50	2.80	3.20	V
VBLACK	Black level output voltage		1.80	2.20	2.50	V
VSYNC	Sync level output voltage	RL = 470 ohms	1.60	2.00	2.30	V
VCB	Color burst pulse width		0.16	0.22	0.28	V
VP-P	Electric potential difference between white and sync levels		0.40	0.60	0.75	V
VP-P (B/W)	Electric potential difference between white and sync levels (using black & white)		0.60	0.80	0.95	V

Note: The typical values listed in the above table assume that VCC = 5.00 V and TA = 25 °C.

RGB output level

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
VRGB 7	RGB maximum output Voltage	RL = 470 ohms	2.5	2.8	3.2	V
VRGB 0	RGB minimum output voltage (Black level)	RL = 470 ohms	1.7	2.0	2.4	V
VP-P	RGB electrical potential difference VRGB7-VRGB0	RL = 470 ohms	0.65	0.8	1.00	V
DRGB	RGB electrical potential difference P-P	RL = 470 ohms			5.0	%

Note: The typical values listed in the above table assume that VCC = 5.00 V and TA = 25 °C.

Synchronize signal output level

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
VTLVH 1	Tri-level output high-level BLEO	RL = 1 k ohm	4.5		VCC	V
VTLVM 1	Tri-level output intermediate level BLEO	RL = 1 k ohm	2.5		3.5	V
VTLVL 1	Tri-level output low-level BLEO	RL = 1 k ohm			0.4	V
VTLVH 2	Tri-level output high-level HSYNC, CSYNC	No load	4.5		VCC	V
VTLVM 2	Tri-level output intermediate level HSYNC, CSYNC	No load	2.7		3.7	V
VTLVL 2	Tri-level output low-level HSYNC, CSYNC	No load			0.8	V
VYH	*Ys output high level	IOH = 100 uA	2.4			V
VYL	*Ys output low level	IOH = 1.6 mA			0.4	V
ITLVH	High-level input current HSYNC,CSYNC	VI = 0.4 V			-4.0	mA
ITLVL	Intermediate level input current HSYNC,CSYNC	VI = 0.4 V			-2.0	mA

Composite video signal

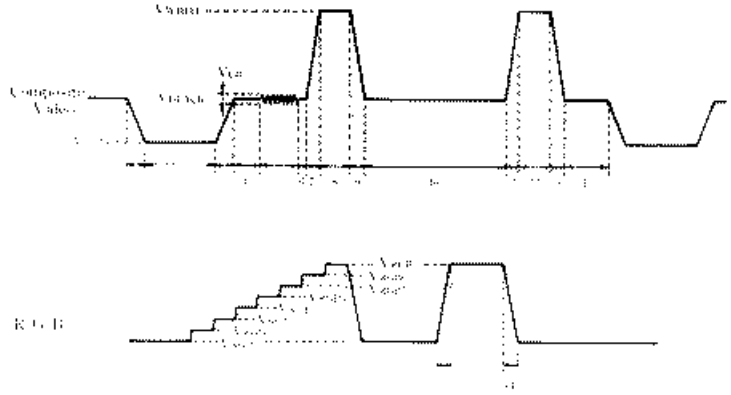
No.	Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
1	TfCV 1	HSYNC fall time				110	ns
2	TWHS	HSYNC pulse width		4.50		4.70	us
3	TrCV 1	HSYNC rise time				90	ns
4	THS-CB	HSYNC color burst delay time		0.40		0.60	us
5	TWCB	Color burst width	RL =	2.60		3.30	us
6	TCB-LB	Color burst-left border delay time	470 ohms	1.10		1.50	us
7	TrCV 2	VBLACK-VWHITE rise time				90	ns
8	TwLB	Left border width	CL =	2.4		2.7	us
9	TfCV 2	VWHITE-VBLACK fall time	150 pF			100	ns
10	TwAD	Active display area		47.00	47.68	48.00	us
11	TwRB	Right border width		2.50		2.80	us
12	TRB-HS	Right border-horizontal synchronous delay time		1.20		1.50	us

Note: Items 8 and 11 are when Display Adjust is 0.

RGB signal

No.	Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
13	TrRGB	RGB signal rise time (VRGB0 ⟶ VRGB7)	RL = 470 ohms			60	ns
14	TfRGB	RGB signal fall time (VRGB7 ⟶ VRGB0)	CL = 150 pF			60	ns

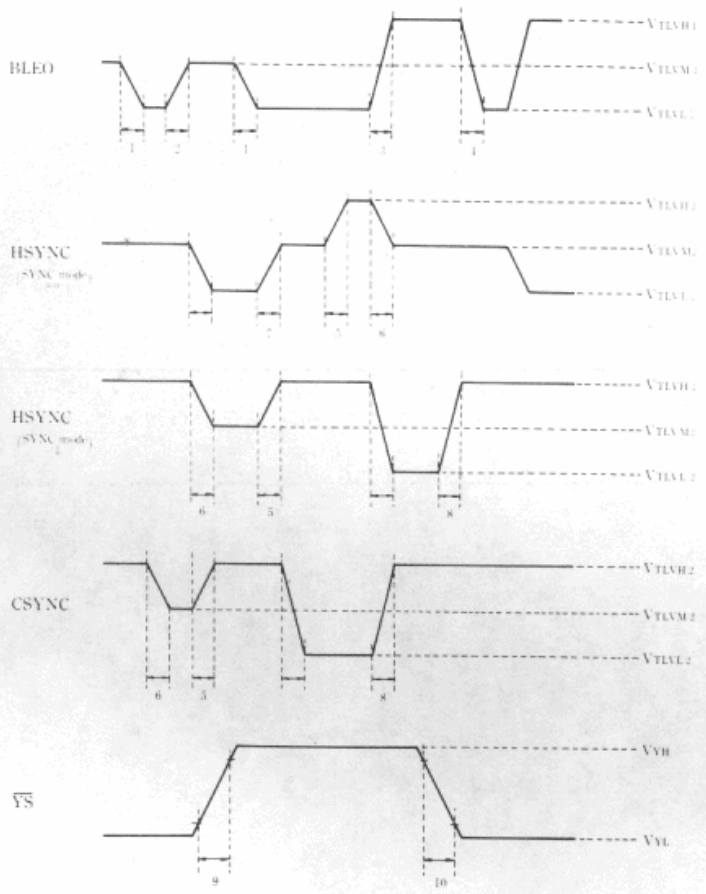
Note: Measurements are 10% to 90%.

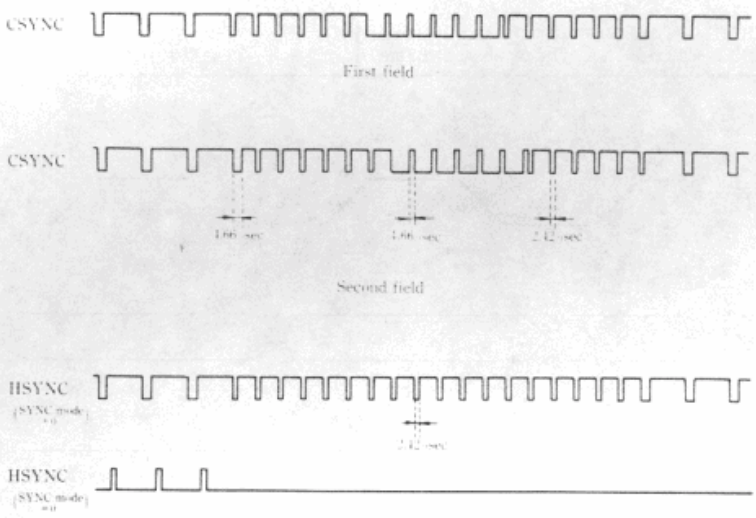


Synchronize signals

No.	Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
1	TfSY 1	BLEO intermediate level-low level fall time				100	ns
2	TrSY 1	BLEO low level-intermediate level rise time				140	ns
3	TrSY 2	BLEO low level-high level rise time				220	ns
4	TfSY 2	BLEO high level-low level fall time	CL = 50 pF			110	ns
5	TrSY 3	SYNC intermediate level-high level rise time				300	ns
6	TfSY 3	SYNC high level-intermediate level fall time				1100	ns
7	TrST 4	SYNC low level-intermediate level rise time				200	ns
8	TrSY 5	SYNC low level-high level rise time				400	ns
9	TrSY 6	*YS low level-high level rise time				25	ns
10	TfSY 6	*YS high level-low level fall time				25	ns

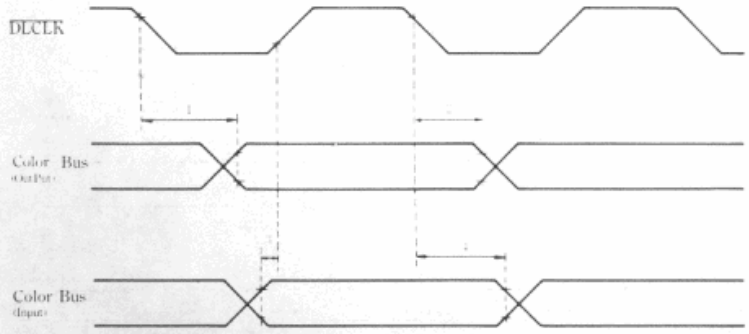
Note: BLEO is the value when RL = 1 k ohm.
Times shown are 10 to 90%.


BLEO
HSYNC
HSYNC
CSYNC
$\overline{YS}$


CSYNC
First field
CSYNC
Second field
HSYNC
HSYNC

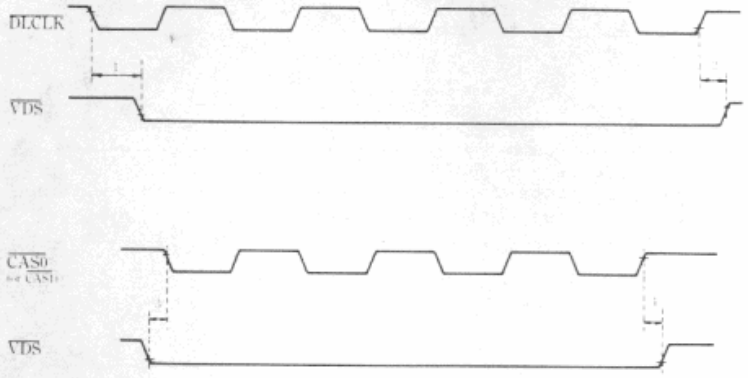
Color bus

No.	Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
1	TDCB0	*DLCLK-color bus output delay time	CL = 50 pF			100	ns
2	THCBO	*DLCLK-color bus output hold time	CL = 50 pF	20			ns
3	TSCBI	Color bus input setup time	-	0			ns
4	THCBI	Color bus input hold time	-	20			ns

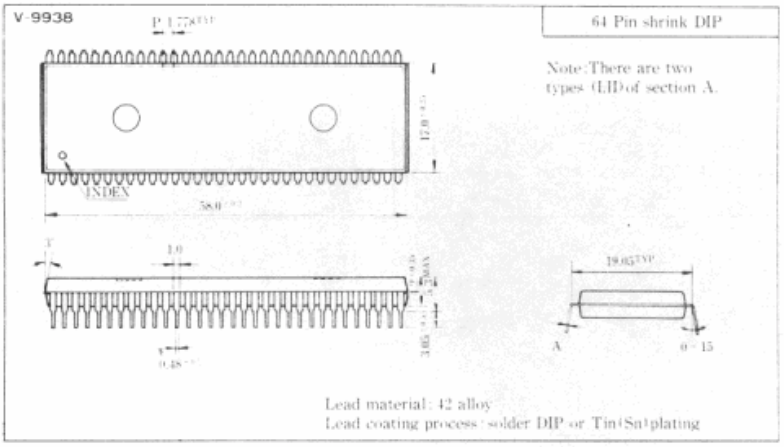


VDS

No.	Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
1	TDVDSL	*DLCLK-*VDS low level delay time	CL = 50 pF	50		100	ns
2	TDVDSH	*DLCLK-*VDS high level delay time		50		100	ns
3	TSVDS	*VDS setup time (for *CAS0 and *CAS1)		20			ns
4	THVDS	*VDS hold time (for *CAS0 and *CAS1)		0			ns

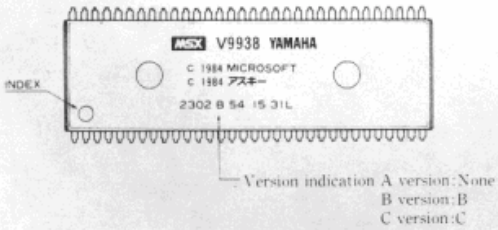


4. External measurements of package



Note: The specifications for this product are subject to change without notice as improvements are made.

5. Version identification



APPENDIX

V9938 MSX-VIDEO Appendix

1. Refresh

The V9938 has a Dynamic RAM (DRAM) auto-refresh function which repeats cycles every 4 ms for 256 addresses.

Special consideration must be given when using a 128 address-2 ms cycle, as the usage depends on whether the DRAMs used are 16K or 64K.

1. 16K DRAMs (VR = 0)

- When the row address and column addresses are seven bits, connect pins AD1 to AD7 on the V9938 to DRAM address pins A0 to A6.

- When the row address is eight bits and the column address is six bits, connect pins AD1 to AD6 on the V9938 to DRAM address pins A1 to A6. Also connect pin AD0 to pin A7. When this configuration is used, only addresses A0 to A6 of the DRAM are refreshed.

2. 64K DRAMs (VR = 1)

- Connect pins AD0 to AD5 and AD7 of the V9938 to pins A0 to A6 of the DRAM. Also connect pin AD6 to pin A7. hen this configuration is used, only addresses A0 to A6 of the DRAM are refreshed.

1-1 V9938 Refresh Address Output

The method of outputting refresh addresses on the V9938 differs depending on the mode being VR or display.

Considering the refresh address as (MSB) R7 R6 R5 R4 R3 R2 R1 R0, 256 addresses are refreshed within 4 ms.

Refresh address output method

V9938 adderss	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
VR = '0'	R5	R4	R3	R2	R1	R0	R7	R6
VR = '0', G6, G7 modes	R0	R7	R6	R5	R4	R3	R2	R1
VR = '0', non-G6, G7 modes	R7	R6	R5	R4	R3	R2	R1	R0

Maximum Refresh Cycle

1. 256 cycle refresh 4 ms
2. 128 cycle refresh
   - Using addresses other than R7 1.25 ms
   - Using addresses other than R6 1.9 ms
   - Using addresses other than R5 2.2 ms
   - Using addresses other than R4 2.5 ms (worst case)

1-2 Example of V9938-DRAM address pin connection and maximum refresh cycle

1. VR = '0'

- DRAM construction with 7-bit row and column addresses

DRAM addresses	A6	A5	A4	A3	A2	A1	A0	Maximum refresh cycle
V9938 addresses (AD)	7	6	5	4	3	2	1	1.9 ms

- DRAM construction with 8-bit row address and 6-bit column address

DRAM addresses	A7	A6	A5	A4	A3	A2	A1	A0	Maximum refresh cycle
V9938 addresses (AD)	7	6	5	4	3	2	1	0	2.2 ms
(AD)	0	6	5	4	3	2	1	7	1.9 ms

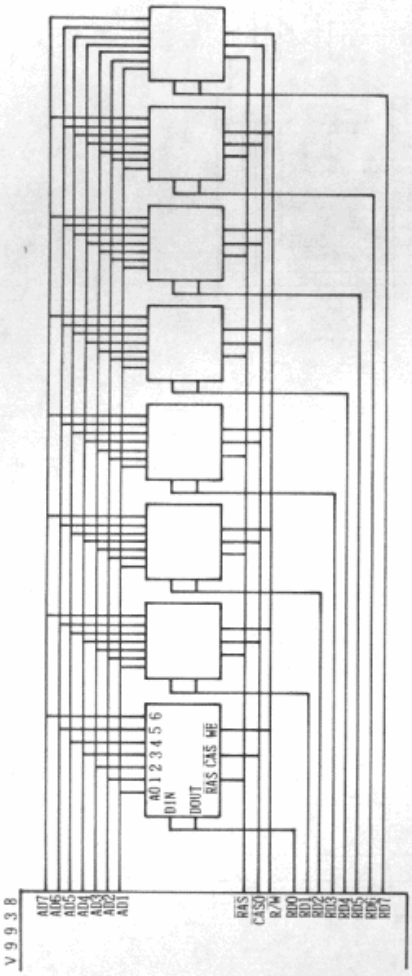
2. VR = '1'

DRAM addresses	A7	A6	A5	A4	A3	A2	A1	A0	Maximum refresh cycle G6/G7 Modes	Maximum refresh cycle Non-G6/G7 Modes
V9938 addresses (AD)	7	6	5	4	3	2	1	0	2.5 ms	1.25 ms
(AD)	6	7	5	4	3	2	1	0	1.25 ms	1.9 ms
(AD)	5	7	6	4	3	2	1	0	1.9 ms	2.2 ms

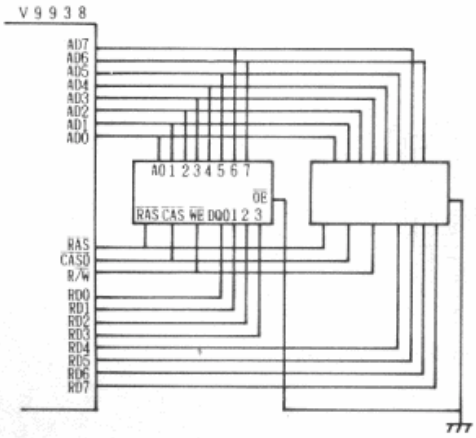
Note: **The maximum refresh cycle is the value when the DRAM refresh addresses are A6 to A0.**

2. Examples of VRAM Interface

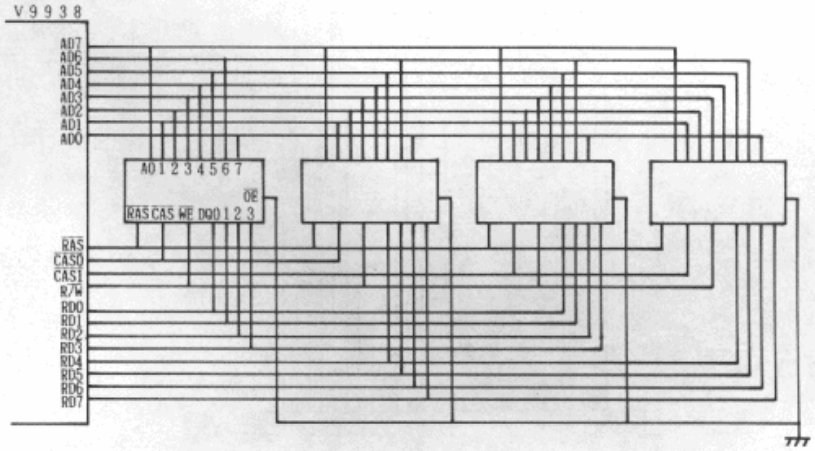
2-1 16K bytes
- (16K x 1) x 8



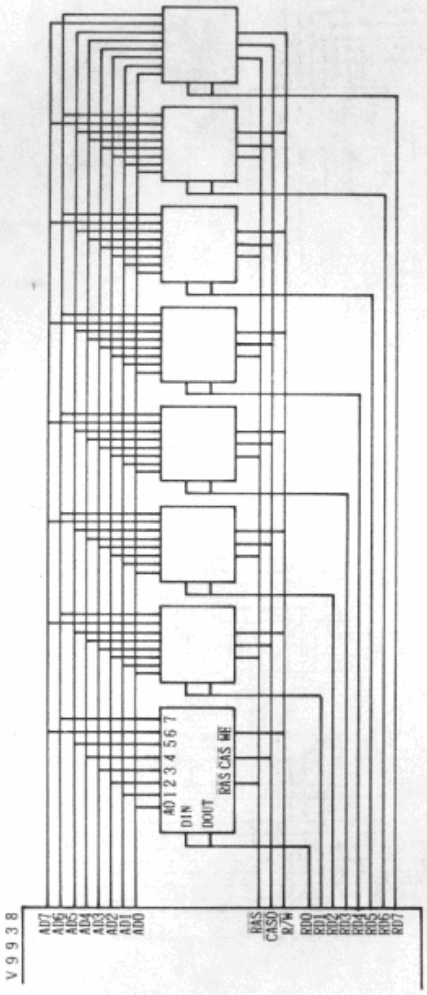
- (16K x 4) x 2



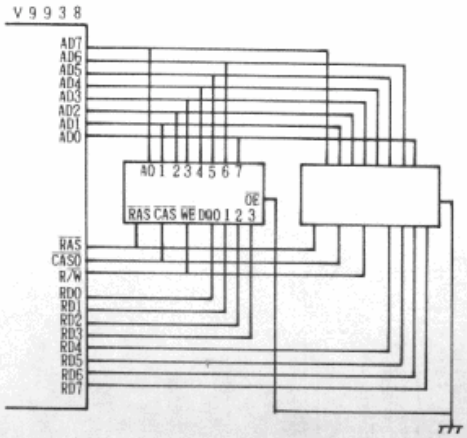
2-2 32K bytes
- (16K x 4) x 4



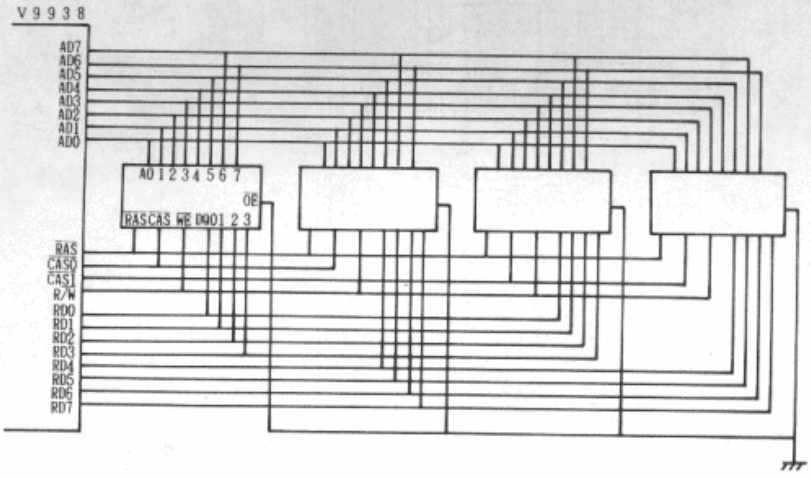
2-3 64K bytes
- (64K x 1) x 8



- (64K x 4) x 2

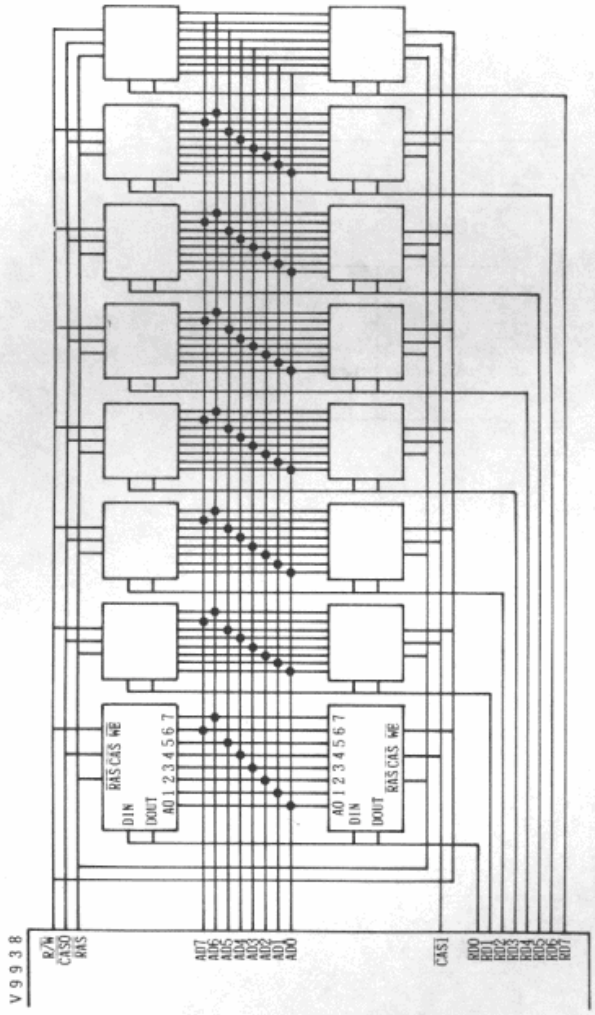


2-4 128K bytes
- (64K x 4) x 4



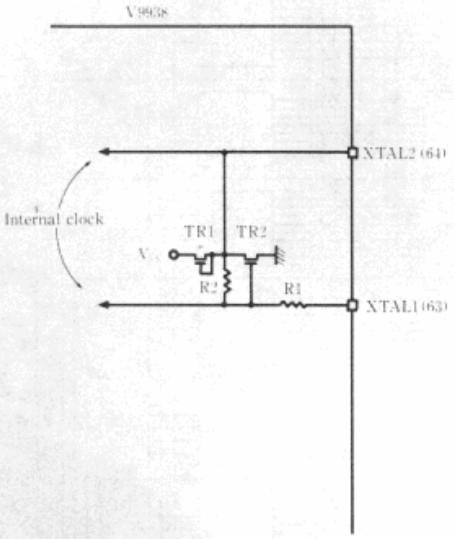
V9938 MSX-VIDEO Appendix

- (64K x 1) x 16



3. Clock oscillation internal circuitry

R1	Approximately 1 k ohms
R2	Approximately 500 k ohms
TR1	Depletion transistor
TR2	Enhancement transistor



4. Usage of unused pins

- Output pins

  All unused output pins may be left open.

- Input or input/output pins

*DLCLK	(3)	Pull up with a resistor
HSYNC	(5)	Open
CSYNC	(6)	Open
C7 to C0	(12) to (19)	Open
*LPS	(26)	Connect to VCC
*LPD	(27)	Connect to VCC
VBB	(33)	Open or connect to GND (1) through a capacitor

5. Cycle mode

The V9938 has three cycle modes, which may be specified according to the following settings of bits S0 and S1 in register #9.

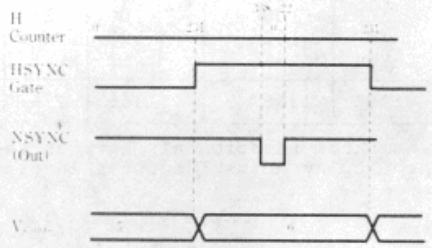
S1	S0	Cycle mode	Screen mode	HSYNC		*YS	fH
				High level	Low level		
0	0	0	PC only	Burst flag	H counter reset input	Low	fXTAL/1368
0	1	1	Mixed	HSYNC	H counter reset input	High/ Low	fXTAL/1365
1	0	2	Ext. Video	HSYNC	H counter reset input	High	fXTAL/1365

6. Cycle input

HSYNC input

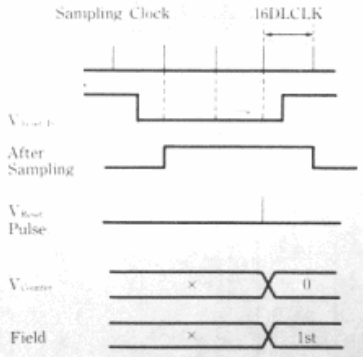
When the V counter is set to 6, input from HSYNC is received, and the H counter is reset on the edge of the transition from High --> Low.

When the HSYNC Gate signal is Low, in other words, the V counter is set to a number other than 6, input signals to the HSYNC input are ignored.



V Reset input (CSYNC)

The V Reset signal is a signal that is internal to the V9938 that cycles at 2.98 us. When three consecutive Lows are received, the V counter is reset. Simultaneously, the first field is selected.



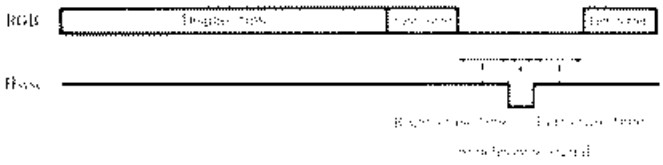
7. Display parameters

7-1 Horizontal display parameters

Unit: XTAL Cycles

	Multicolor mode G1 to G7 modes		Text I mode Text II mode	
S1, S0 (R#9)	1, 2	0	1, 2	0
Display cycle	1024	1024	960	960
Right border	57	59	85	87
Right erase time	26	27	26	27
Synchronize signal	100	100	100	100
Left erase time	102	102	102	102
Left border	56	56	92	92
Total	1365	1368	1365	1368

Note: The above table shows the relationship between the RGB signal and HSYNC when the Display Adjust Register (Register Number 18) is set to 0.



7-2 Vertical display parameters (NTSC)

Unit: Lines

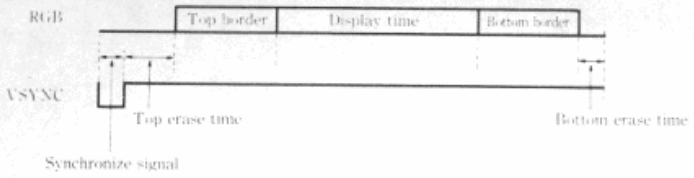
Lines	192 lines LN = 0			212 lines LN = 1		
	Non-Interlace	Interlace Field		Non-Interlace	Interlace Field	
		1st	2nd		1st	2nd
Synchronize signal	3	3	3	3	3	3
Top erase time	13	13	13.5	13	13	13.5
Top border	26	26	26	16	16	16
Display time	192	192	192	212	212	212
Bottom border	25	25.5	25	15	15.5	15
Bottom erase time	3	3	3	3	3	3
Total	262	262.5	262.5	262	262.5	262.5

7-3 Vertical display parameters (PAL)

Unit: Lines

Lines	192 lines LN = 0			212 lines LN = 1		
	Non-Interlace	Interlace Field 1st	Interlace Field 2nd	Non-Interlace	Interlace Field 1st	Interlace Field 2nd
Synchronize signal	3	3	3	3	3	3
Top erase time	13	13	13.5	13	13	13.5
Top border	53	53	53	43	43	43
Display time	192	192	192	212	212	212
Bottom border	49	48.5	48	39	38.5	38
Bottom erase time	3	3	3	3	3	3
Total	313	312.5	312.5	313	312.5	312.5

Note: The above table shows the relationship between RGB and VSYNC when the Display Adjust Register (Register Number 18) is set to 0.



8. Color palette

		G	R	B
Color code	0	0	0	0
	1	0	0	0
	2	6	1	1
	3	7	3	3
	4	1	1	7
	5	3	2	7
	6	1	5	1
	7	6	2	7
	8	1	7	1
	9	3	7	3
	A	6	6	1
	B	6	6	4
	C	4	1	1
	D	2	6	5
	E	5	5	5
	F	7	7	7

The color palette is set as shown in the table when a RESET is done. This table has no meaning in G7 mode since the color palette is not used.

In addition, since there are only four colors available in G5 mode, values from 4 to F have no meaning.

In this table, 0 means that the GRB intensity is 0, and 7 means that the GRB intensity is set to maximum.

9. Composite video color burst

The phase of the color brust is set according to the contents of register numbers 20, 21, and 22. These registers are initialized on the transition between Low --> High of the *Reset signal.

Register number	Value upon initialization
20	&H 00
21	&H 3B
22	&H 05

In addition, when the values of all of the above registers are set to 00, the color burst effect may be removed.

Note: Do not set values other than those listed above in registers 20 to 22.

10. Color bus

Mode	Color bus							
	7	6	5	4	3	2	1	0
Text I, II Multicolor G1 to G4	X	X	X	X	CC3	CC2	CC1	CC0
					←Even →		← Odd →	
G5	X	X	X	X	CC1	CC0	CC1	CC0
	←——— Even ———→				←——— Odd ———→			
G6	CC3	CC2	CC1	CC0	CC3	CC2	CC1	CC0
G7	CC7	CC6	CC5	CC4	CC3	CC2	CC1	CC0

The color bus is used as listed above whether doing input or output. For example, in G4 mode, the lower four bits (CC0 to CC3) output the color code. These bits must be set before the signals are placed on the color bus. In this case, the upper four bits are ignored. In addition, during input, since the lower four bits have the color code, that signal is displayed, and if DG = "1", the information is also stored in the VRAM.

When using high resolution modes such as G5 and G6 modes, the even and odd dots are handled together as in 0-1, 2-3, 4-5, etc. (This is the same during input and output).

11. Sprites in G5 mode

In the G5 mode, the static screen has a resolution of 512 dots in the horizontal direction. In addition, each pixel has two bits for the color code.

Sprites are displayed using half the resolution of a static screen. As a result, the sprite pattern has a resolution of 256 dots in the horizontal direction, and the coordinates are x = 0 to 255.

In order to display a sprite pattern as shown in Fig. 1, you must create a sprite pattern generator table as shown in Fig. 2.

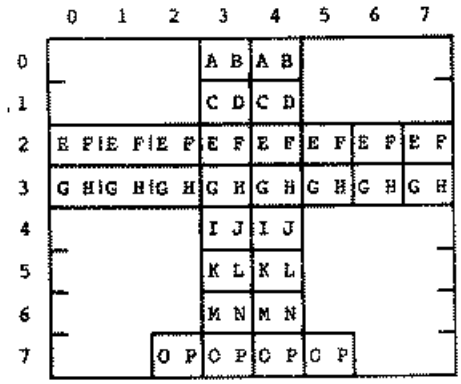


Fig. 1 Sprite pattern

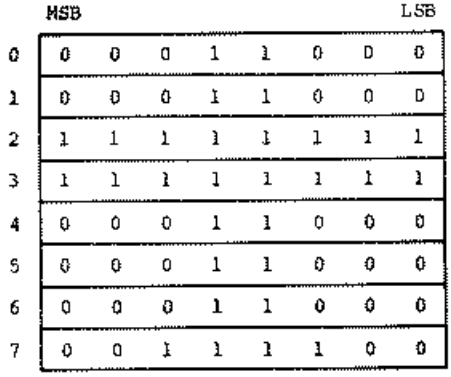


Fig. 2 Pattern generator table

The colors of the sprites depend on the settings of the sprite color table. The upper four bits of the color table use controls such as the Early Clock. The lower four bits are divided into two groups of two bits. The upper two-bit group is for the color of the even pixels, and the lower two-bit group is for the color of odd pixels. If the bits are set individually for each two-bit group and the two groups have different colors, the resolution for sprite colors can be considered as being 512 dots.

If the color table is written as shown in Fig. 3, the sprite pattern will be as shown in Fig. 1.

	MSB		LSB
0		A	B
1		C	C
2		E	F
3		G	H
4		I	J
5		K	L
6		M	N
7		O	P

Fig. 3 Color table