HuC6270

CMOS Video Display Controller

MANUAL

TABLE OF CONTENTS

1. DESCRIPTION

The HuC6270 is a CMOS video display controller (VDC) which contains an externally synchronizable sync signal generator, a 16 bit address unit for an externally connected video RAM (VRAM), a sprite attribute table buffer (SATB) containing attributes of sprites, shift registers used to display background and sprites, and a priority circuit on a single chip. Other distinctive features are the CPU's capability of accessing the VRAM for reading or writing background/sprite information, and the function of block data transfers between SATB and VRAM, or between VRAMs.

1.1 Features

- Monolithic CMOS video display controller
- Screen composition: background + sprite
- Screen configuration...

	Pattern size	Corlor specification
Background	8 × 8 dots*	16 of 256 colors
Sprite	16 × 16 dots	16 of 256 colors

* Charactor cicle

- Pattern size Color specification
- Pattern definition: Patterns defined in externally connected VRAM
- Video data: 9 bits parallel (TTL compatible)
- Supports transfer of data between CPU and VRAM and transfer of video data to display IC
- Single power: 5V
- 80-pin plastic flat package

2. FUNCTIONS

2.1 Internal Registers

2.1.1 Access to Internal Registers

The HuC6270 has various internal registers. The CPU gains read/write accesses to them to implement the powerful HuC6270 capabilities.

- <Registers directly accessible or not from the CPU>

  The address register and the status register can be directly accessed by the CPU when both pins A1 and $\overline{CS}$ are held low.

  Those other than the above two can only be accessed by specifying an appropriate register number in the address register and setting conditions which drive A1 high and $\overline{CS}$ low.

2.1.2 List of Internal Registers

$\overline{CS}$	A1	R/W	REG No.	Sym-bol	Register name	A0													
						1								0					
						D 15	D 14	D 13	D 12	D 11	D 10	D 9	D 8	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0
1	—	—	—	—	—																
0	0	W	—	AR	Address									AR							
0	0	R	—	SR	Status	0								BSY	VD	DV	DS	RR	OR	CR	
0	1	W	R00	MAWR	Memory Address Write	MAWR															
0	1	W	R01	MARR	Memory Address Read	MARR															
0	1	W	R02	VWR	VRAM Data Write	VWR															
0	1	R	R02	VRR	VRAM Data Read	VRR															
0	1	—	R03	—	Reserved																
0	1	—	R04	—	Reserved																
0	1	W	R05	CR	Control							IW	DR	TE	BB	SB	EX	IE			
0	1	W	R06	RCR	Scanning Line Detection							RCR									
0	1	W	R07	BXR	BGX Scroll							BXR									
0	1	W	R08	BYR	BGY Scroll							BYR									
0	1	W	R09	MWR	Memory Access Width									CM	SCREEN		SM		VM		
0	1	W	R0A	HSR	Horizontal Sync				HDS								HSW				
0	1	W	R0B	HDR	Horizontal Display				HDE						HDW						
0	1	W	R0C	VPR	Vertical Sync	VDS											VSW				
0	1	W	R0D	VDW	Vertical Display							VDW									
0	1	W	R0E	VCR	Vertical Display End Position									VDW							
0	1	W	R0F	DCR	Block Transfer Control												DSR	DI/D	SI/D	DVC	DSC
0	1	W	R10	SOUR	Block Transfer Source Address	SOUR															
0	1	W	R11	DESR	Block Transfer Destination Address	DESR															
0	1	W	R12	LENR	Block Transfer Length	LENR															
0	1	W	R13	DVSSR	VRAM-SATB Block Transfer Source	DVSSR															

● A0 is "0" or the high byte data if A0 is "1". This can be summarized

● Shaded area

The shaded area is not available.

2.1.3 Functions of Internal Registers

(1) Address Register (AR)

(A0=0, A1=0, R/W=W)

MSB LSB

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											AR				

The address register is a write only register which addresses an HuC6270 internal register, R00 – R13. Before reading or writing an internal register, its register number must be written into the AR. The target register can be selected when its number is written into the AR with both A1 and $\overline{CS}$ at "L" level.

WARNING:

Do not use AR=04.

If number "04" is set in the AR, the system can not be assured of normal operation.

(2) Status Register (SR)

(A0=0, A1=0, R/W=R)

MSB LSB

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
←…… 0 ……→									BSY	VD	DV	DS	RR	OR	CR

The status register gives the internal status of the HuC6270. If an interrupt enabled by the interrupt enable control bits of the control register (CR) and the block transfer control register (DCR) occurs, the corresponding bit of the status register is set, driving the interrupt request (IRQ) active. The status register is automatically cleared after reading its contents (except for BSY, bit 6).

(a) Collision Detect (CR; bit 0)

This status indicates that sprite #0 has collided with any of sprites #1 – #63.

(b) Over Detect (OR; bit 1)

This status is set in any of the following three cases:

1) An attempt was made to display more than 17 sprites on a scanning line.
2) Some sprite pattern data cannot be fetched into the HuC6270 during a horizontal blanking period.
3) All of the sprite pattern data cannot be fetched into the sprite shift register because the CGX of some sprites are set.

(c) Scanning Line Detect (RR; bit 2)

This status indicates that the scanning line counter has matched up to the setting of the scanning line detection register.

(d) Block Transfer (between VRAM and STAB) End Detect (DS; bit 3)

This status indicates that the block transfer of data between the VRAM and the SATB has been completed.

(e) Block transfer (between VRAMs) End Detect (DV; bit 4)

This status indicates that the blcok transfer of data between VRAMs has been completed. When a vertical display cycle starts, the block transfer between VRAMs will be cancelled. In this case, no End Detect status will be generated.

(f) Vertical Blanking Period Detect (VD; bit 5)

This status indicates that a vertical blanking period has started.

(g) Busy (BSY; bit 6)

This status indicates that the VRAM is being accessed (read or written) in response to the CPU access. Even if BSY is set, no interrupt request ($\overline{IRQ}$) will occur. While BSY is held at "1", do not write data into the IW (bits 11 and 12 of CR R05).

(This bit basically provides $\overline{BUSY}$ pin information, with its logic reversed.)

(3) Memory Address Write Register (MAWR R00)

MSB LSB

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAWR															

The memory address write register sets an address at which the CPU starts to write data into the VRAM.

(4) Memory Address Read Register (MARR R01)

MSB LSB

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MARR															

The memory address read register sets an address at which the CPU starts to read data from the VRAM.

When the high byte of this register is written, the HuC6270 starts to read data from the VRAM for loading it into the VRAM data read register. When this read operation is completed, the register is automatically incremented.

(5) VRAM Data Write Register (VWR R02)

MSB LSB

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VWR															

The VRAM data write register is used when data is transferred from the CPU to the VRAM. When the high byte of this register is written, the HuC6270 starts to write data into the VRAM. When this write operation is completed, the memory address write register is incremented.

(6) VRAM Data Read Register (VRR R02)

MSB LSB

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VRR															

The VRAM data read register is used when data is transferred from the VRAM to the CPU. The data read from the VRAM at the address set in the memory address read register is stored in the VRAM data read register. Thus the contents of the VRAM at that address are read by reading the stored data. Reading the high byte of the VRR register triggers reading the next word of the VRAM.

WARNING

While data is writing into or reading from the VRAM, do not change the division ratio of the frequency of the clock for the VCE (video color encoder). (This means that the clock frequency must not be reduced.)

(7) Control Register (CR R05)

MSB LSB

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			IW		DR	TE		BB	SB	EX		IE			

The control register is used to set an operation mode of the HuC6270. The EX, TE, and DR fields will become valid immediately after they are updated.

(a) Interrupt Request Enable (IE; R05, bits 0 – 3)

This field enables the interrupt requested by each bit.

IE field	Symbol	Contents
0	CC	Collision Detect
1	OC	Over Detect
2	RC	Scanning Line Detect
3	VC	Vertical Blanking Period Detect

(b) External Sync (EX; R05, bits 4 and 5)

EX field		Contents
5	4	
0	0	Both $\overline{\text{VSYNC}}$ and $\overline{\text{HSYNC}}$ work as input and are synchronized to external signals.
0	1	$\overline{\text{VSYNC}}$ works as input and is synchronized to external signals. ($\overline{\text{HSYC}}$ works as output.)
1	0	Invalid
1	1	Both $\overline{\text{VSYNC}}$ and $\overline{\text{HSYNC}}$ work as output.

(c) Sprite Blanking (SB; R05, bit 6)

This bit controls whether or not sprites are to be displayed on the screen. Once set, the SB will become valid from the next horizontal display period.

(This excludes clearing of the burst mode.)

0: Clears the sprite.

1: Displays the sprite.

(d) Background Blanking (BB; R05, bit 7)

This bit controls whether or not background is to be displayed on the screen. Once set, the BB will become valid from the next horizontal display period.

(This excludes clearing of the burst mode.)

0: Clears the background.

1: Displays the background.

When SB = BB = 0, the burst mode will start with the next frame. In the burst mode:

1) The HuC6270 stops to access the VRAM to display and enables the CPU to access the VRAM.
2) VRAM-VRAM block transfer becomes always possible.
3) VD0 – VD7 outputs "L" level, while SPBG outputs "H" level.
   Burst mode is cleared in next frame after SB = 1 or BB = 1.

(e) DISP Output Select (TE; R05, bits 8 and 9)

TE field		DISP output	Contents
9	8		
0	0	DISP	Outputs "H" level during display.
0	1	$\overline{\text{BURST}}$	Indicates the position in which Color Burst is inserted. Low active.
1	0	$\overline{\text{INTHSYNC}}$	Internal horizontal SYNC.
1	1	—	Invalid

(f) Dynamic RAM Refresh (DR; R 5, bit 10)
When the VRAM access width of the memory access width register for background is 2 or 4 dots, setting the DR to "1" provides refresh addresses at MA0 – MA15.

(g) Memory Address Read/Write Register Increment Select (IW; R05, bits 11 and 12)
Once set, the memory address read/write register is automatically incremented each time data is transferred to it. The IW field selects the extent of incrementing.
The register is incremented by accessing the high byte.

IW bit		Extent of incrementing
12	11	
0	0	+1
0	1	$+20_{16}$
1	0	$+40_{16}$
1	1	$+80_{16}$

(8) Scanning Line Detection Register (RCR R06)

MSB LSB

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						RCR									

The scanning line detection register sets which line should be used for the invocation of an interrupt in a CRT scanning operation. The interrupt will occur when the value of the internal scanning line counter matches the value set in the scanning line detection register. The counter is set to '64' at the previous scanning line of start timing of display period and is incremented for each scan.

(9) BGX Scroll Register (BXR R07)

MSB LSB

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						BXR									

The BGX scroll register is used to horizontally scroll the background display. If this register is updated, the screen will become valid with the next scanning line.

(10) BGY Scroll Register (BYR R08)

MSB LSB

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							BYR								

The BGY scroll register is used to vertically scroll the background display. If the register is updated during display, the screen will become valid with the next scanning line (BYR + 1).

(11) Memory Access Width Register (MWR R09)

MSB LSB

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								CM	SCREEN			SM		VM	

(a) VRAM Access width Mode (VM; R09, bits 1 and 0)

This field selects how many clocks should be used for VRAM, background attribute table (BAT), or character generator (CG) accesses, or for block transfer.

A value should be selected according to the memory access speed of the VRAM. If the field is updated, it will become valid at the beginning of a vertical blanking period. Writing/reading data to/from the VRAM can be allowed after the next vertical blanking period detect interrupt.

VM bit		Access width	Assignment for one character cycle (8 dots)							
1	0		1	2	3	4	5	6	7	8
0	0	1	CPU	BAT	CPU	—	CPU	CG0	CPU	CG1
0	1	2	BAT		CPU		CG0		CG1	
1	0	2	BAT		CPU		CG0		CG1	
1	1	4	BAT				CG0 or CG1			

In the 4 clocks mode, data is displayed with four of the 256 colors.

(b) Sprite Access Width Mode (SM: R09, bits 2 and 3)

This field selects how many clocks should be used for access to the sprite generator (SP) during a horizontal blanking period. If the field is updated, it will become valid at the beginning of a vertical blanking period.

SM bit		Access width	Assignment for one character cycle							
3	2		1	2	3	4	5	6	7	8
0	0	1	SP0	SP1	SP2	SP3	SP0	SP1	SP2	SP3
1	0	*1 2	SP0 (SP2		SP1 SP3)		SP0 (SP2		SP1 SP3)	
1	0	2	SP0		SP1		SP2		SP3	
1	1	*2 4	SP0 SP2				SP1 SP3			

*1. The least significant bit of a pattern code selects (SP0, SP1) or (SP2, SP3).

*2. SP0-SP3 are fetched during two consecutive character cycles.

In the 4 clocks mode, there may be cases where sprites are not correctly displayed if an odd number of horizontal blanking periods is applied.

(c) Screen (SCREEN; R09, bits 4 6)

This field selects how many characters should be contained in the virtual screen in the X or Y direction. If the field is updated, it will become valid at the beginning of a vertical blanking period.

SCREEN bit			Number of characters	
6	5	4	X direction	Y direction
0	0	0	32	32
0	0	1	64	32
0	1	0	128	32
0	1	1	128	32
1	0	0	32	64
1	0	1	64	64
1	1	0	128	64
1	1	1	128	64

(d) CG MODE (CM; R09, bit 7)

This bit is used to select the character generator blocks for the 4 clocks mode. Once set, the CM will become valid with the next scanning line.

0: (CG0)CH0, CH1

1: (CG1)CH2, CH3

(12) Horizontal Sync Register (HSR R0A)

MSB LSB

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HDS										HSW				

The horizontal sync register specifies the horizontal sync pulse width (HSW) and the horizontal display start position (HDS). Once updated, the HDS and HSW will become valid at the beginning of each period.

(a) Horizontal Sync Pulse Width (HSW; R0A, bits 0-4)

This field contains the "L" level pulse width of a horizontal sync pulse in terms of character cycles.

The data is 5 bits long, and can be any value from 1 to 32 meeting the specifications of the CRT used. If horizontal pulse width "N" is selected, the HSW should be "N-1".

(b) Horizontal Display Start Position (HDS; R0A, bits8-14)

This field contains the interval between the rising edge of a horizontal sync signal to the start of its display in terms of character cycles. The data is 7 bits long, indicating the optimum horizontal position on the CRT. If horizontal display position (horizontal back porch) "N" is selected, the HDS should be "N-1".

(13) Horizontal Display Register (HDR R0B)

MSB LSB

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HDE								HDW						

The horizontal display register specifies the horizontal display width (HDW) and the horizontal display end position (HDE). Once updated, the HDW and HDE will become valid at the beginning of each period.

(a) Horizontal Display Width (HDW; R0B, bits 0-6)
This field contains the display period per line in terms of character cycles. The data is 7 bits long, and is based on the number of horizontal characters appearing on the CRT. If the horizontal display width "N" is selected, the HDW should be "N-1".

(b) Horizontal Display End Position (HDE; R0B, bits 8-14)
This field contains the interval between the end of horizontal display and the falling edge of the horizontal sync pulse in terms of character cycles. The data is 7 bits long, indicating the optimum horizontal display position on the CRT. If horizontal display end position (horizontal back porch) "N" is selected, the HDE should be "N-1".
NOTE: HSW=HDS=HDE=0 is invalid.

(14) Vertical Sync Register (VPR R0C)

MSB LSB

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VDS											VSW				

The vertical sync register specifies the vertical sync pulse width (VSW) and the vertical display start position (VDS). Once updated, the VDS and VSW will become valid at the beginning of each period.

(a) Vertical Sync Pulse Width (VSW; R0C, bits 0-4)
This field contains the "L" level pulse width of a vertical sync pulse in terms of scanning lines. The data is 5 bits long, and can be any value from 1 to 2 meeting the specifications of the CRT used. If vertical sync pulse width "N" is selected, the VSW should be "N-1".

(b) Vertical Display Start Position (VDS; R0C, bits 8-15)
This field contains the interval between the rising edge of a vertical sync pulse to the start of display period in terms of scanning lines. If verticaldisplay start position (vertical back porch) "N" is selected, the VDS should be "N-2".

(15) Vertical Display Register (VDR R0D)

MSB LSB

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							VDW								

The vertical display register specifies the vertical display period in terms of scanning lines. The data is 9 bits long, and is based on the number of scanning lines to appear on the CRT. If the vertical display width "N" is selected, the VDW should be "N-1".
Once updated, the VDW will become valid at the beginning of a vertical display period.

(16) Vertical Display End Position Register (VCR R0E)

MSB LSB

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								VCR							

The vertical display end position register specifies the interval between the end of a vertical display period and the falling edge of the vertical sync pulse in terms of scanning lines. The data is 8 bits long, indicating the optimum vertical display position on the CRT. If vertical display end position (vertical front porch) "N" is selected, the VCR should be "N". Once updated, the VCR becomes valid at the beginning of a vertical front porch.

(17) Block Transfer Control Register (DCR R0F)

MSB LSB

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											DSR	DI/D	SI/D	DVC	DSC

The block transfer control register controls what mode should be used for block transfer of data between the two channels, VRAM-SATB and VRAM-VRAM.

(a) VRAM-SATB Transfer Complete Interrupt Request Enable (DSC; R0F, bit 0)
This bit controls whether or not an interrupt should be enabled when a VRAM-SATB block transfer is completed.
0: Interrupt disabled
1: Interrupt enabled

(b) VRAM-VRAM Transfer Complete Interrupt Request Enable (DVC; R0F, bit 1)
This bit controls whether or not an interrupt should be enabled whena VRAM-VRAM block transfer is completed.
0: Interrupt disabled
1: Interrupt enabled

(c) Source Address INC/DEC (SI/D; R0F, bit 2)
This bit controls whether the source address for VRAM-VRAM block transfers should be automatically incremented or decremented.
0: Incremented
1: Decremented

(d) Destination Address INC/DEC (DI/D; R0F, bit 3)
This bit controls whether the destination address for VRAM-VRAM block transfers should be automatically incremented or decremented.
0: Incremented
1: Decremented

(e) VRAM-SATB Transfer Auto-Repeat (DSR; R0F, bit 4)
This bit controls whether or not a VRAM-SATB block transfer should be repeated for each vertical blanking period.
0: Not repeat
1: Repeat

(18) Block Transfer Source Address Register (SOUR R10)

MSB LSB

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOUR															

The block transfer source address register specifies the source start address for a VRAM-VRAM block transfer. The block transfer will not work normally if this register is updated during the VRAM-VRAM transfer cycle.

(19) Block Transfer Destination Address Register (DESR R11)

MSB LSB

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DESR															

The block transfer destination address register specifies the destination start address for a VRAM-VRAM block transfer. The block transfer will not work normally if this register is updated during the VRAM-VRAM transfer cycle.

(20) Block Transfer Length Register (LENR R12)

MSB LSB

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LENR															

The block transfer length register specifies the length of a block for VRAM-VRAM transfers. The length can be set up to 64K words. If word count "M" is selected, the LENR should be "M-1". The block transfer will not work normally if this register is updated during the VRAM-VRAM transfer cycle.

(21) VRAM-SATB Block Transfer Source Address Register (DVSSR R13)

MSB LSB

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DVSSR															

The VRAM-SATB block transfer source address register specifies the source start address for a VRAM-SATB block transfer. The block transfer will not work normally if this register is updated during the VRAM-VRAM transfer cycle.

NOTE: Block Transfer

(a) A VRAM-VRAM block transfer can be performed during a vertical blanking period or in the burst mode. If the block length specified for particular data exceeds a vertical blanking period, the rest data will not be transferred in the next vertical blanking period. It is triggered by access to the high byte of the block length register (LENR).

(b) For VRAM-SATB block transfers, 256 words are transferred at the beginning of a vertical blanking period It is triggered by access to the high byte of the VRAM-SATB block transfer source address register (DVSSR). If the register is set, a block transfer operation will start at the beginning of the following vertical blanking period.

2.2 Display Functions

2.2.1 Definition of Display Screen

Fig. 2-2-1 shows the setting of each register which defines the display screen. The values on the horizontal axis are given in terms of characters, and those on the vertical axis in terms of scanning lines.

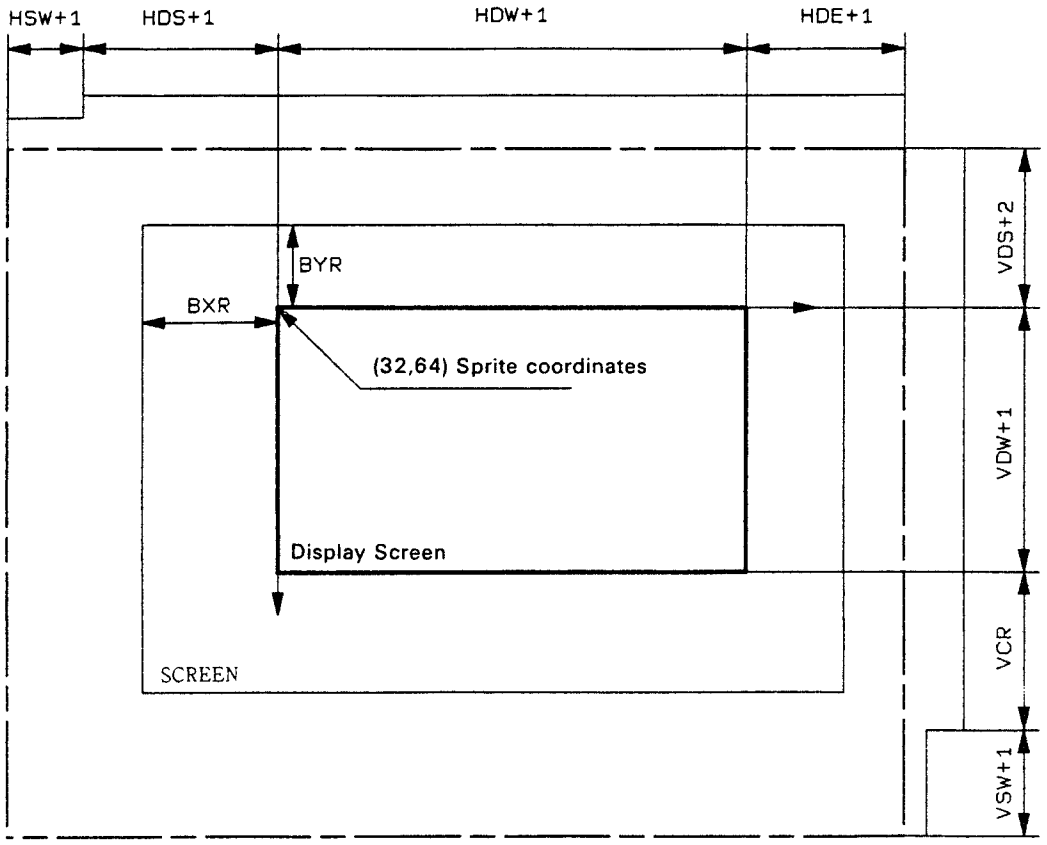


Fig. 2-2-1 Definition of Display Screen

2.2.2 Scroll Control

The HuC6270 allows the background to be smoothly scrolled both vertically and horizontally.

(a) Vertical scroll

The background can be scrolled, line by line, by controlling the BGY scroll register.

(b) Horizontal scroll

The background can be scrolled, dot by dot, by controlling the BGX scroll register.

2.3 Background Display Functions

2.3.1 Background Display

(a) Character size: 8 x 8 dots

(b) Number of characters that can be defined in the character generator (CG): 4,096 (maximum)

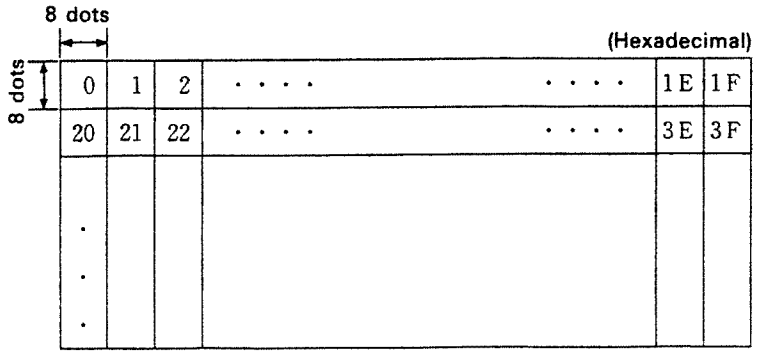
(c) Colors displayed: 16 of 256 colors for each character

(4 of 256 if 4-dot mode is selected by the memory access width register)

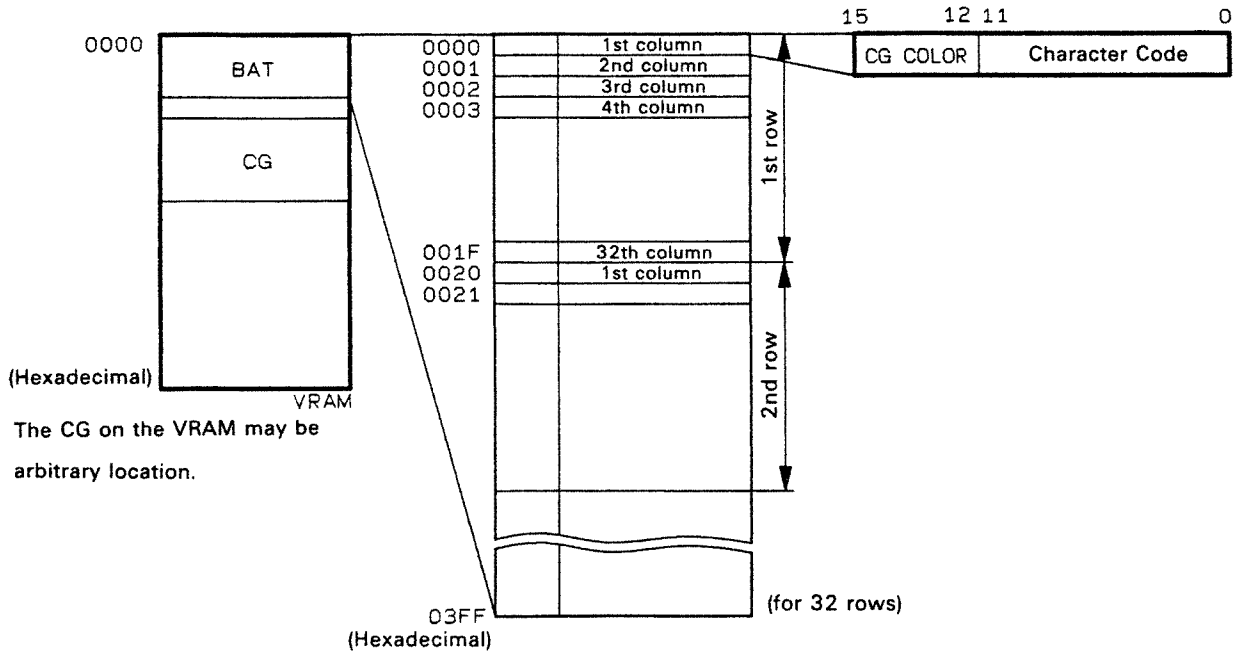
2.3.2 Background Attribute Table (BAT)

The background attribute table resides at the addresses beginning with 0 on the VRAM, and controls what character will be displayed in each position on the virtual screen in what color. The top left character on the screen is at address 0, followed by another character at address 1.

(1) BAT Addresses of Characters Appearing on Virtual Screen
(for 32 x 32 character configuration)



(2) Location of BAT on VRAM and its Contents



(3) Background Attribute Table Configuration

MSB LSB

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CG COLOR				Character code											

(a) Character code
Specifies the character pattern defined by the character generator (CG) within the VRAM.

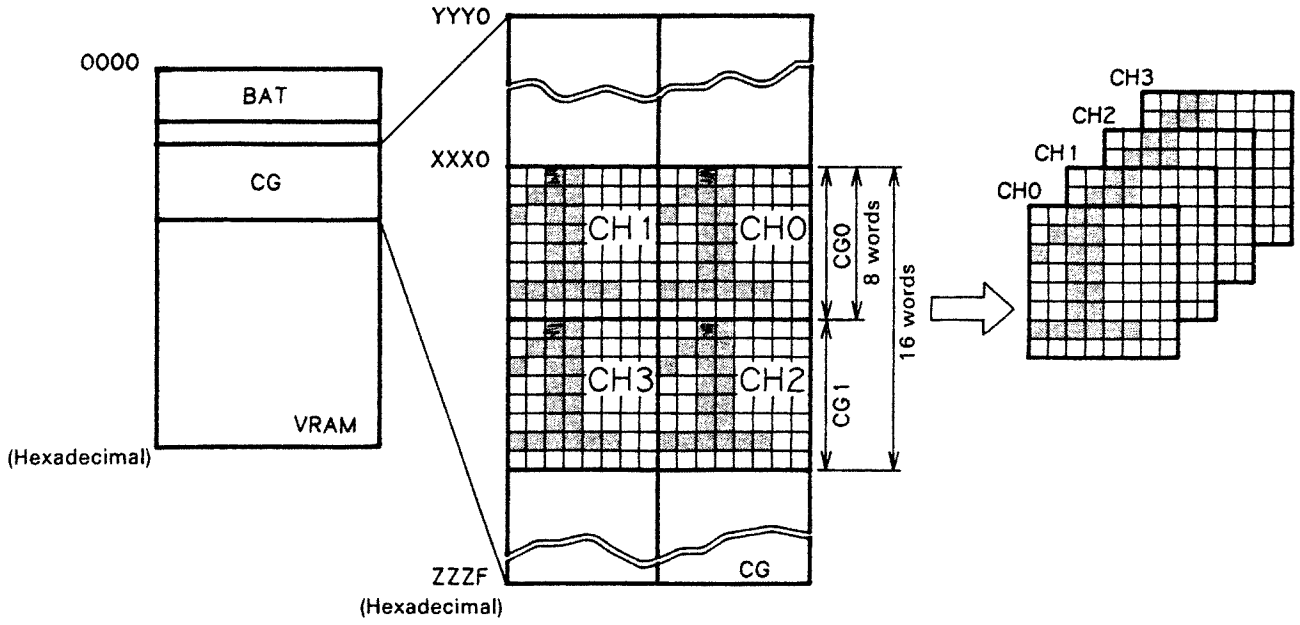
(b) CG COLOR
Specifies a 4-bit area color code.

2.3.3 Character Generator (CG)

The character generator occupies a space of the VRAM in which character patterns are defined. One character on the screen consists of 8 x 8 dots. In order to define each dot as a 4-bit color code, four 8 x 8 dots areas in the VRAM are used. They are named CH0, CH1, CH2, and CH3; a pattern is defined using a total of 16 words 8 words (CG0) of CH0 and CH1, and 8 words (CG1) of CH2 and CH3.

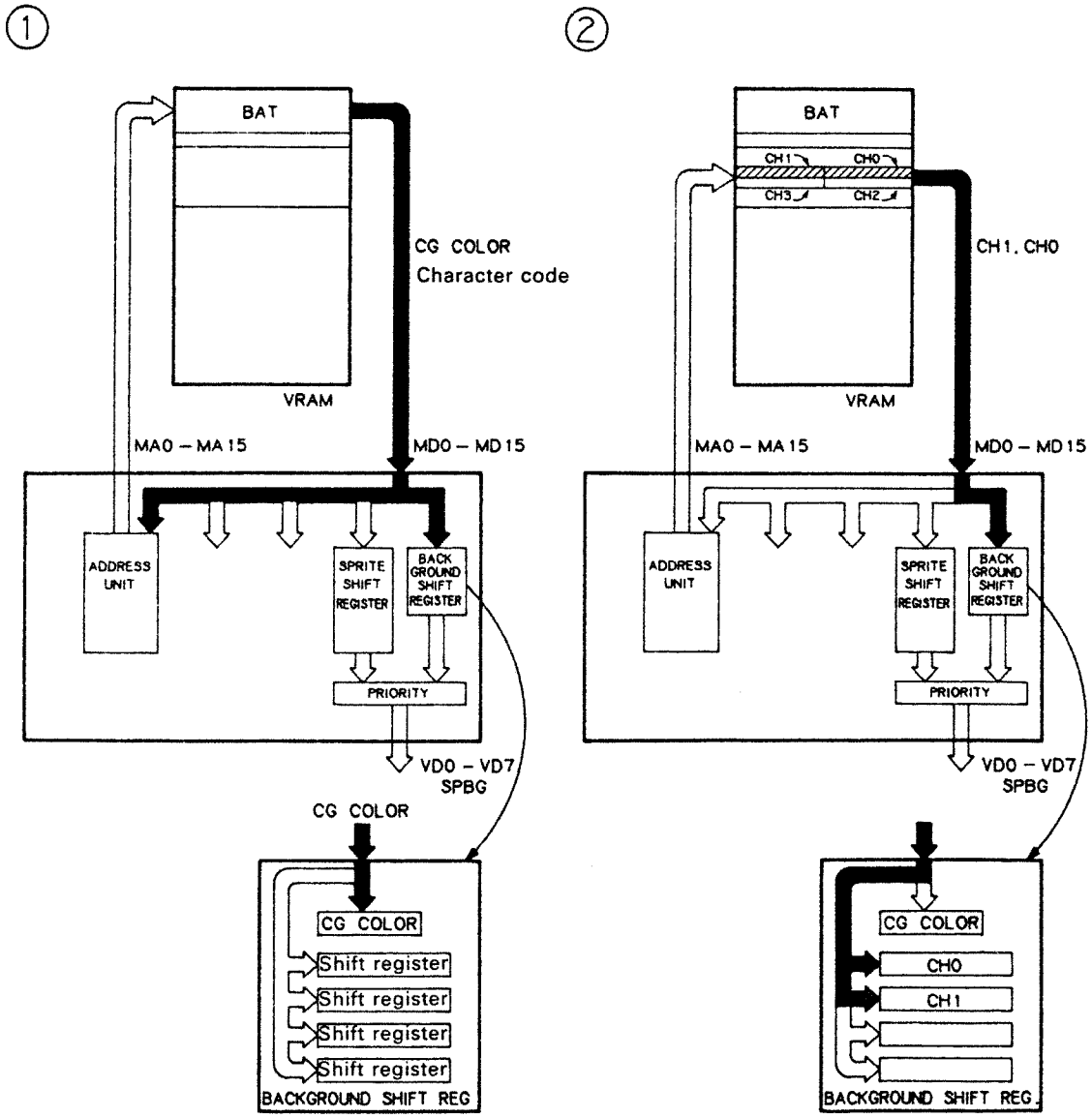
Area CH0 should start at address "××××××××××××0000" <binary>.

The first 12 bits ("××××××××××××" or "×××$_{16}$" <hexadecimal>) constitutes a character code.

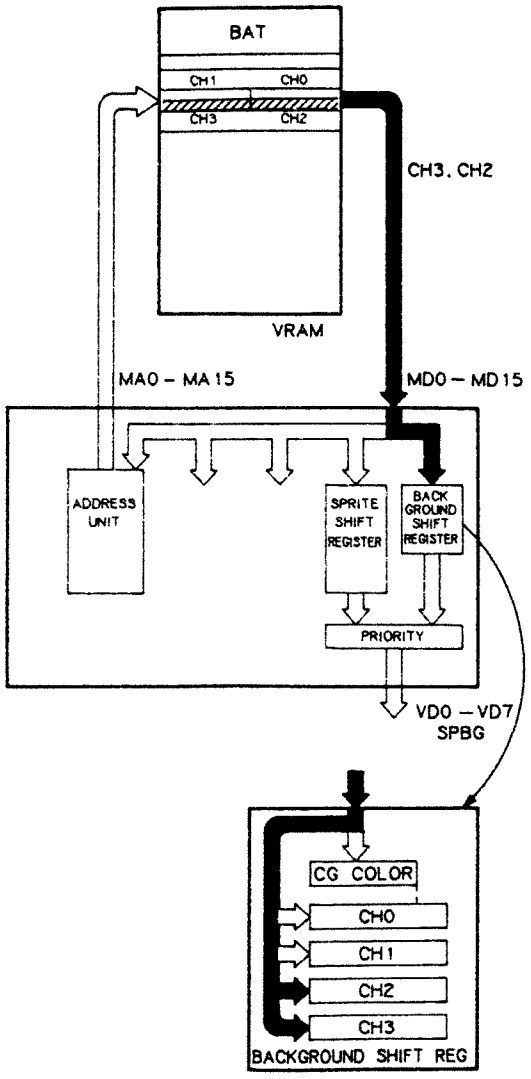


2.3.4 Background Display Control

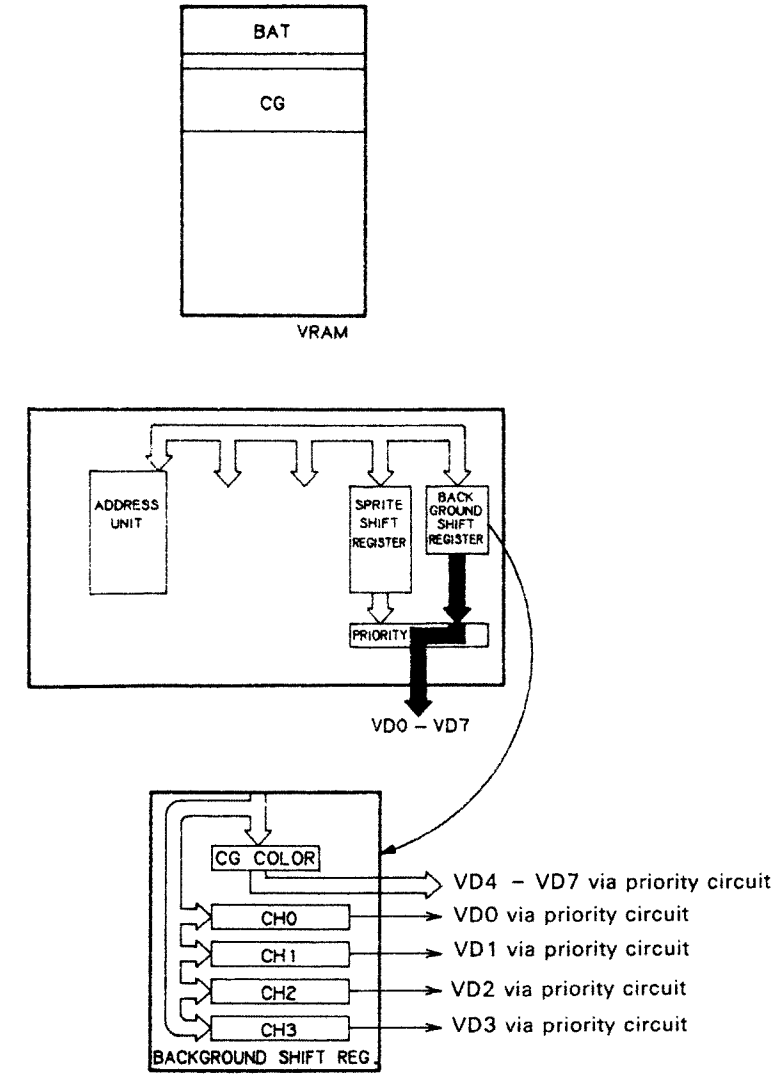
The background is controlled during a horizontal display period. Figs. ① through ④ below depict the flow of data for display of one character. ① An address is generated from the beam position and the background attribute table is read to obtain a character code and a CG COLOR. ②③ The address of the character generator is generated from the character code obtained, then the pattern data is read and taken into the background shift register within the HuC6270. ④ The pattern, with the CG COLOR, is output from the background shift register.
Two words (CH0-CH1 and CH2-CH3) are read from the character generator for 16-color display, and one word (CH0-CH1 or CH2-CH3) for 4-color display.



③


BAT
CH1
CH0
CH3
CH2
VRAM
CH3. CH2
MA0 – MA15
MD0 – MD15
ADDRESS UNIT
SPRITE SHIFT REGISTER
BACK GROUND SHIFT REGISTER
PRIORITY
VD0 – VD7
SPBG
CG COLOR
CH0
CH1
CH2
CH3
BACKGROUND SHIFT REG

④


BAT
CG
VRAM
ADDRESS UNIT
SPRITE SHIFT REGISTER
BACK GROUND SHIFT REGISTER
PRIORITY
VD0 – VD7
CG COLOR
CH0
CH1
CH2
CH3
BACKGROUND SHIFT REG
VD4 – VD7 via priority circuit
VD0 via priority circuit
VD1 via priority circuit
VD2 via priority circuit
VD3 via priority circuit

2.3.5 Background Display Video Output

Background data is output with the following video data:

- SPBG ..... "0" (SPBG=VD8)
- VD7 – VD4 ..... CG COLOR appears.
- VD3 – VD0 ..... Each bit of CH3 – CH0 appears.

VD8	VD7	VD6	VD5	VD4	VD3	VD2	VD1	VD0
0	CG COLOR				CH3	CH2	CH1	CH0
					CG1		CG0	

If VD3=VD2=VD1=VD0=0, then VD7=VD6=VD5=VD4=0 (independently of the CG COLOR).

In the 4 clocks mode, the following video outputs are provided:

If CG0 is selected ..... VD3=VD2=0
VD1=CH1 bit
VD0=CH0 bit

If CG1 is selected ..... VD3=CH3 bit
VD2=CH2 bit
VD1=VD0=0

2.3.6 Video Data during Blanking Period (Border Color)

The video data during a blanking period is; SPBG=1, VD7 – VD0=0

VD8	VD7	VD6	VD5	VD4	VD3	VD2	VD1	VD0
1	0	0	0	0	0	0	0	0

NOTE: VD8=SPBG

2.4 Sprite Functions

2.4.1 Sprite Display

(1) Each sprite can move to pointed coordinates on the display screen. Redefining a pattern is not required.

(2) The 64 sprites are given priority. If two or more sprites overlap with each other, only the highest priority sprite is visible.

(3) Sprite size
16×16 dots

(4) Number of sprites that can be defined in the sprite generator (SG)
1,024 (maximum): If 64K words VRAM are used for defining sprites. (One word consisting of 16 bits)

(5) Number of sprites that can be registered in the sprite attribute table (SAT)
64

(6) Restrictions on sprite display
Up to 16 sprites can exist on one horizontal scanning line. If an attempt is made to display 17 or more sprites on one horizontal scanning line, the 16 sprites having higher priority are displayed and the rest not.
The number of sprites that can be displayed is limited as shown in the table below depending on the values set in the horizontal sync register and the horizontal display register.

Sprite width access mode (SM)	Number of sprites in SATB that can be searched	Number of sprites that can be taken from SG
1 (0, 0)	When d ≧ 32, 64 (all) When d ≦ 31, 2d + 1	2 (e − 2), Max16
2 (0, 1)	Same as above	Same as above
2 (1, 0)	When d ≧ 32, 64 (all) When d ≦ 31, 2d	e − 2, Max16
4 (1, 1)	Same as above	1/2 (e-2), Max16

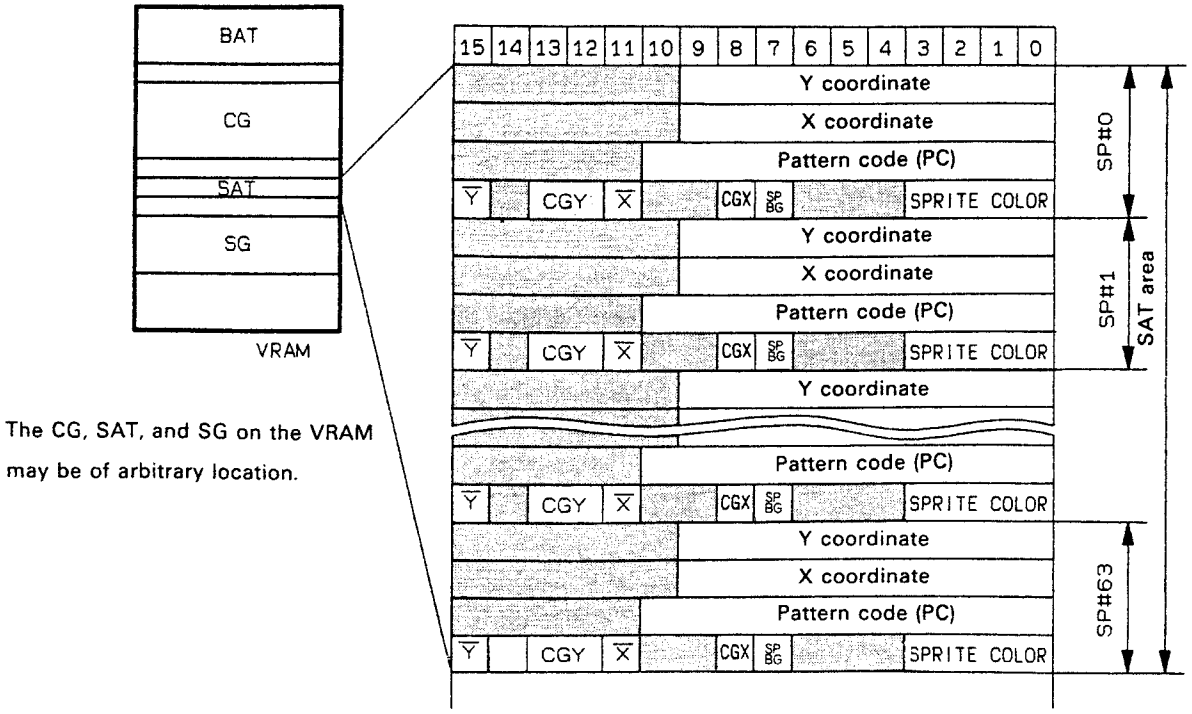
NOTE:
In the above table, the horizontal display width
(d) is HDW + 1 and the horizontal blanking period
(e) is HDE + HSW + HDS + 3.

(7) Color display
16 of the 256 colors are specified for each sprite. (4 of the 256 colors are specified if the 2-dot 4-color sprite display mode is selected.)

2.4.2 Sprite Attribute Table Buffer (SATB)

The sprite attribute table buffer (SATB) is a memory of the HuC6270 which is used for registration of sprite display positions (coordinates X and Y), colors, pattern codes, etc. Direct writing to the SATB from the CPU is impossible. Data can only be written to the SATB by VRAM-SATB block transfer from a sprite attribute table (SAT) area within the VRAM.

(1) Example of placing SAT in VRAM



In the SAT area, four words are used to define one sprite (256 words to define a total of 64 sprites). The priority of sprites follows that of addresses.

(2) SAT configuration

The SAT is configured as shown:

MSB LSB

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						Y-coordinate									
						X-coordinate									
						Pattern code (PC)									
$\overline{Y}$		CGY		$\overline{X}$			CGX	SPBG				SPRITE COLOR			

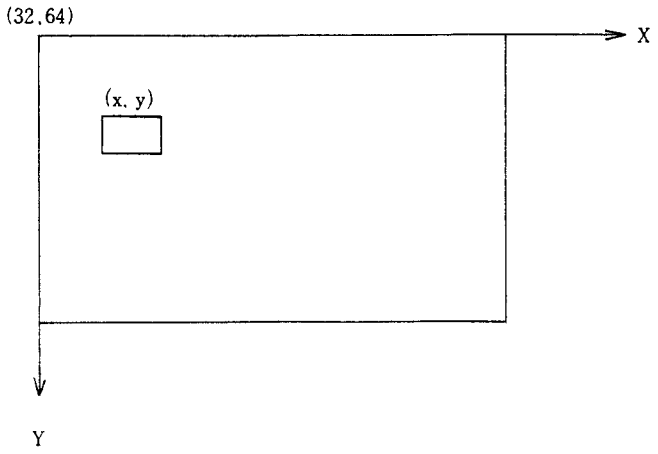
(a) Y-coordinate

The Y-coordinate is the field where the vertical position of a sprite is specified. (0 to 1, 0 2 3)

(b) X-coordinate

The X-coordinate is the field where the horizontal position of a sprite is specified. (0 to 1, 0 2 3)

The coordinates and the coordinate of the left top corner (point) on the screen are as shown below:



(c) Pattern code

This code specifies a sprite pattern which is defined in the sprite generator (SG) of the VRAM. The high-order 10 bits of a pattern code come to equal the high-order 10 bits of a VRAM address. The pattern data defined in this address area is the sprite pattern specified by the pattern code.

The least significant bit of a pattern number controls which is to be selected, (SG0-SG1) or (SG2-SG3), if the 2-dot 4-color sprite display mode is selected.

(d) SPRITE COLOR

This field specifies the 4-bit area color code for a sprite.

(e) SPBG

This bit controls which is given high display priority, background or sprite.

0: Background display

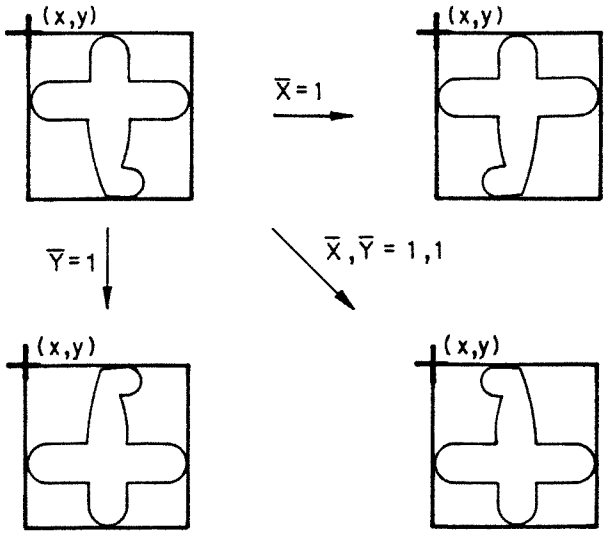
1: Sprite display

(f) $\overline{X}$

Setting X causes a sprite to be right-left inverted when displayed.

(g) $\overline{Y}$

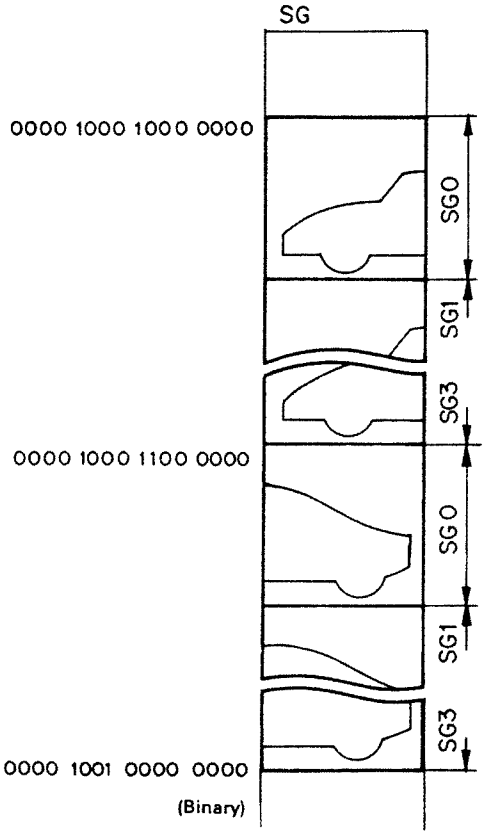
Setting Y causes a sprite to be up-down inverted when displayed.



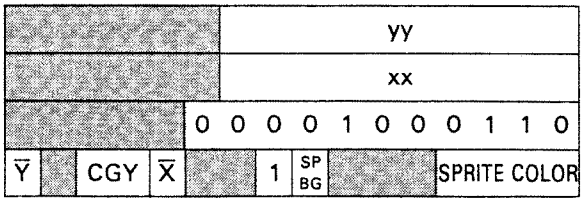
(h) CGX

With CGX set, two specified sprites appear together horizontally in such an order from left to right that one sprite whose pattern code bit 1 (PC1) is "0" precedes the other whose PC1 is "1". If $\overline{X}$ has been set, these two sprites are right-left inverted when they appear.

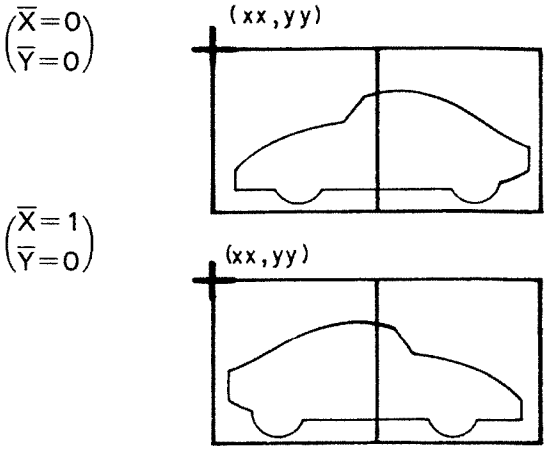
● Example of CGX (address map)



● SAT



● Sprites displayed



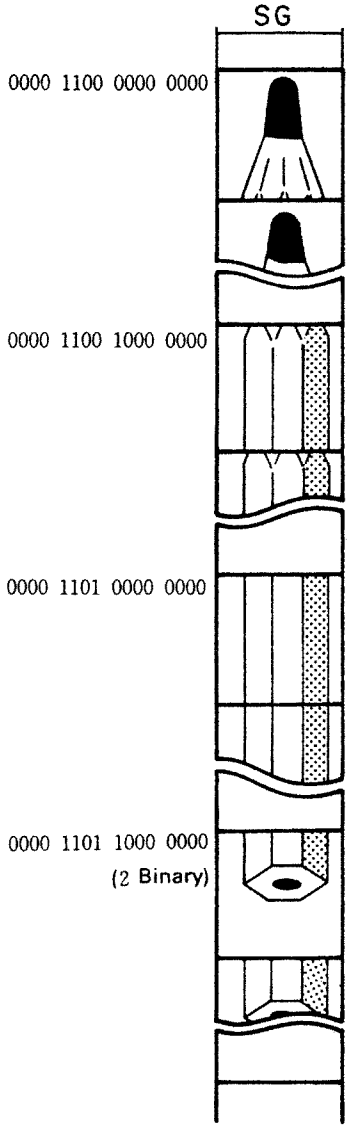
(i) CGY

This field allows a vertically combined display of two specified sprites whose pattern code bits 3 (PC3) and bit2 (PC2) are (0,0) and (0,1), or (1,0) and (1,1) if 2CGY is selected, or four specified sprites whose PC3 and PC2 are (0,0), (0,1), (1,0) and (1,1) if 4CGY is selected. If $\overline{Y}$ has been set, these two or four sprites are up-down inverted when they appear.

● Specification of CGY

Specification of CGY		CGY
13	12	
0	0	NORMAL
0	1	2CGY
1	0	Not used
1	1	4CGY

● Example of 4CGY display

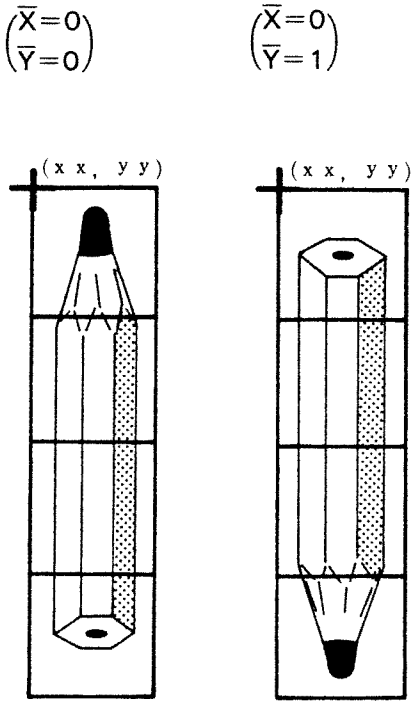


● SAT

MSG LSB

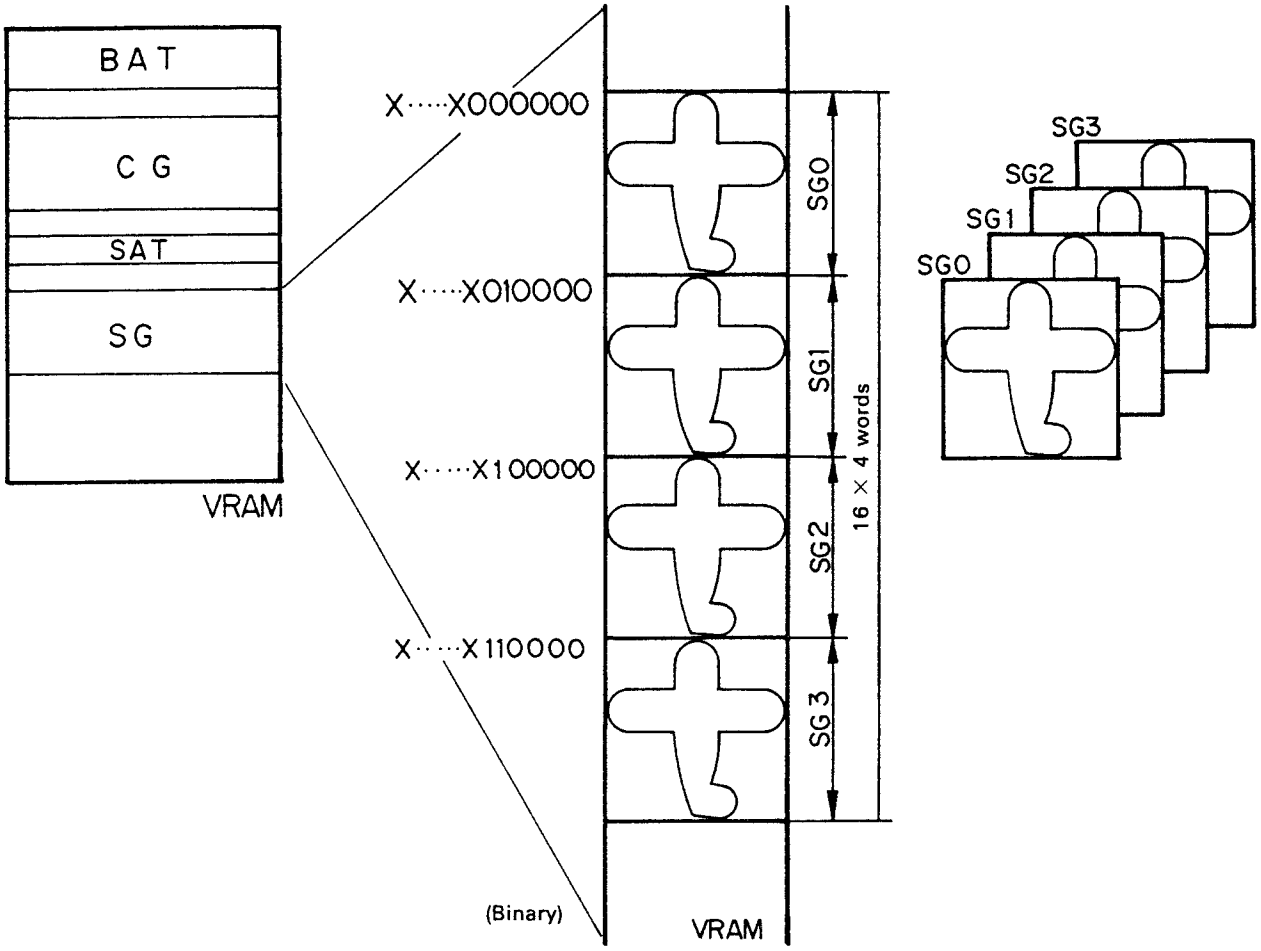
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						yy									
						xx									
					0	0	0	0	1	1	0	0	1	0	0
$\overline{Y}$		1	1			0	SP BG				SRITE COLOR				

● Sprites displayed



2.4.3 Sprite Generator (SG)

The sprite generator (SG) is a memory space of the VRAM that stores sprite patterns. One sprite consists of 4 areas, each containing 16 x 16 dots, numbered SG0, SG1, SG2, and SG3. Each area uses 16 words of memory (a total of consecutive 64 words is required for all the four areas). SG0 should align at address "××××××××××000000" (binary).
The first "××××××××××" comes to equal the high-order 10 bits of a pattern code.



2.4.4 Video Data during Sprite Display

Sprites are output with the following video outputs:

- SPBG.... "1" (SPBG=VD8)
- VD7-VD4 .... SP COLOR
- VD3-VD0 .... Each bit of SG3-SG0 is output.

VD8	VD7	VD6	VD5	VD4	VD3	VD2	VD1	VD0
1	SP COLOR				SG3	SG2	SG1	SG0

When VD3=VD2=VD1=VD0=0, then VD7=VD6=VD5=VD4=0
(independently of the SP COLOR).
In the 2-dot 4-corlor mode, the (SG0, SG1) or (SG2, SG3) selected by the least significant bit (PC0) of a parrern code are output to VD0 and VD1, with VD2 and VD3 being "0".

2.5 Interface with CPU (Access to VRAM)

(a) Writing data into registers
Writing data into HuC6270 registers for a selection display mode or any other function and for setting values, take the following steps:

Step 1: Set the target register number in the address register (AR). (A0=0, A1=0)
Step 2: Write the low byte data. (A0=0, A1=1)
Step 3: Write the high byte data. (A0=0, A1=1)

(b) Reading status

Step 1: Read the status (low byte data). (For the high byte data, "00_{16}" is read.)

(c) Writing data into VRAM
In order to the CPU to write data into the VRAM via the Huc6270, take the following steps:

Step 1: Set the memory address write register (MAWR) number "00_{16}" in the address register (AR).
Step 2: Write the low byte of an appropriate VRAM address.
Step 3: Write the high byte of an appropriate VRAM address.
Step 4: Set the VRAM data write register (VWR) number "02_{16}" in the address register (AR).
Step 5: Write the low byte of data into VRAM data write regiser (VWR).
Step 6: Write the high byte of data into VRAM data write regiser (VWR). (MAWR is incremented.)
Repeat step 5 and 6 asmany times as necessary.

(d) Reading data from VRAM
In order to the CPU to read data into the VRAM via the Huc6270, take the following steps:

Step 1: Set the memory address read register (MARR) number "01_{16}" in the address register (AR).
Step 2: Write the low byte of an appropriate VRAM address.
Step 3: Write the high byte of an appropriate VRAM address.
Step 4: Set the VRAM data read register (VRR) number "02_{16}" in the address register (AR).
Step 5: Read the low byte of data into VRAM data read regiser (VRR).
Step 6: Read the high byte of data into VRAM data read regiser (VRR). (MARR is incremented.)
Repeat step 5 and 6 asmany times as necessary.