

GENESIS
SOFTWARE MANUAL
SEGA ENTERPRISES

Copyright 1989 Sega of Japan

sega

genesis

soft

This manual may not, in whole or part, be copied, photocopied, reproduced, or translated without prior written consent from Sega Enterprises.

sega onisoft

Genesis Technical Overview

February 28, 1990

GENESIS:

68000 @8MHz

- Main CPU
- 1 MByte (8 Mbit) ROM Area
- 64 KByte RAM Area

VDP (Video Display Processor)

- Dedicated video display processor
 - Controls playfield and sprites
 - Capable of DMA
 - Horizontal and vertical interrupts

Z80 @4MHz

- Controls PSG (Programmable Sound Generator) & FM Chips
- 8 KBytes of dedicated Sound RAM

VIDEO:

- NOTE: Playfield and Sprites are character-based
- Display Area (visual)
 - 40 Chars wide x 28 chars high
 - Each char is 8 x 8 pixels
 - Pixel resolution = 320 x 224
 - 3 Planes
 - 2 Scrolling playfields
 - 1 Sprite plane
 - Definable priorities between planes
 - Playfields:
 - 6 Different sizes
 - 1 Playfield can have a "fixed" window
 - Playfield map
 - Each char position takes 2 bytes, that includes:
 - Char name (10 bits); points to char definition
 - Horizontal flip
 - Vertical flip
 - Color palette (2 bits); index into CRAM
 - Priority

- Scrolling:
 - 1 Pixel scrolling resolution
 - Horizontal:
 - Whole playfield as unit
 - Each character line
 - Each scan line
 - Vertical
 - Whole playfield as unit
 - 2 Char wide columns
- Sprites:
 - 1 x 1 Char up to 4 x 4 chars
 - Up to 80 sprites can be defined
 - Up to 20 sprites displayed on a scan line
 - Sprite priorities
- Character Definitions
 - 4 bits/pixel; points to color register
 - 4 bytes/scanline of char
 - 32 bytes for complete char definition
 - Playfield and sprite chars are the same

COLOR:

- Uses CRAM (part of the VDP)
- 64 9-bit wide color registers
 - 64 colors out of 512 possible colors
 - 3 bits of Red
 - 3 bits of Green
 - 3 bits of Blue
- 4 palettes of 16 colors
 - 0th color (of each palette) is always transparent

OTHER:

- DMA
 - Removes the 68000 from the BUS
 - Can move 205 bytes/scanline during VBLANK
 - There are 36 scanlines during VBLANK
 - DMA can move 7380 bytes during VBLANK
 - Horizontal & Vertical Interrupts

SOUND:

- Z80 Controls:
 - PSG (TI 76489 chip)
 - FM chip (Yamaha YM 2612)
 - 6-Channel stereo
- Z80 can access ROM data
- 8 KBytes RAM

HARDWARE:

- 2 Controllers
 - Joypad
 - 3 Buttons
 - Start button
- 1 External port
- 2 Video-outs (RF & RGB)
- Audio jack (stereo)
- Volume control (for audio jack)

sega nintendo

sega onisoft

TABLE OF CONTENTS

I.	MEMORY MAP	1
A.	Mega Drive 16 Bit Mode	1
1.	68K MEMORY MAP	1
2.	Z80 MEMORY MAP	2
3.	68000 ACCESS TO Z80 MEMORY	2
4.	I/O AREA	3
5.	CONTROL AREA	3
6.	VDP AREA	4
II.	VDP 315-5313	5
A.	TERMINOLOGY	6
B.	DISPLAY SPECIFICATION	6
C.	VDP STRUCTURE	8
1.	CTRL	8
2.	VRAM	8
3.	CRAM	8
4.	VSRAM	8
5.	DMA	8
D.	INTERRUPT	9
1.	VERTICAL INTERRUPT (V-INT)	9
2.	HORIZONTAL INTERRUPT (H-INT)	9
3.	EXTERNAL INTERRUPT (EX-INT)	10
E.	VDP PORT	10
1.	\$ C 0 0 0 0 0 (DATA PORT)	10
2.	\$ C 0 0 0 0 4 (CONTROL PORT)	11
3.	\$ C 0 0 0 0 8 (HV COUNTER)	12
F.	VDP REGISTER	13
1.	MODE SET REGISTER NO.1	13
2.	MODE SET REGISTER NO.2	13
3.	PATTERN NAME TABLE BASE ADDRESS FOR SCROLL A	13
4.	PATTERN NAME TABLE BASE ADDRESS FOR WINDOW	13
5.	PATTERN NAME TABLE BASE ADDRESS FOR SCROLL B	14
6.	SPRITE ATTRIBUTE TABLE BASE ADDRESS	14
7.	BACKGROUND COLOR	14
8.	H INTERRUPT REGISTER	14
9.	MODE SET REGISTER NO. 3	15
10.	MODE SET REGISTER NO. 4	15
11.	H SCROLL DATA TABLE BASE ADDRESS	16
12.	AUTO INCREMENT DATA	16
13.	SCROLL SIZE	16
14.	WINDOW H POSITION	16
15.	WINDOW V POSITION	17
16.	DMA LENGTH COUNTER LOW	17
17.	DMA LENGTH COUNTER HIGH	17

18.	DMA SOURCE ADDRESS LOW	17
19.	DMA SOURCE ADDRESS MID	17
20.	DMA SOURCE ADDRESS HIGH	17
G.	ACCESS VDP RAM	18
1.	RAM ADDRESS SETTING	18
2.	VRAM ACCESS	19
3.	CRAM ACCESS	22
4.	VSRAM ACCESS	23
5.	ACCESS TIMING	24
6.	HV COUNTER	25
H.	DMA TRANSFER	26
1.	MEMORY TO VRAM	26
2.	VRAM FILL	28
3.	VRAM COPY	31
4.	DMA TRANSFER CAPACITY	34
I.	SCROLLING SCREEN	35
1.	SCROLLING SCREEN SIZE	37
2.	HORIZONTAL SCROLLING	39
3.	VERTICAL SCROLLING	41
4.	SCROLL PATTERN NAME	42
5.	PATTERN GENERATOR	44
J.	WINDOW	46
1.	DISPLAY POSITION	46
2.	WINDOW PRIORITY	50
3.	WINDOW PATTERN NAME	51
K.	SPRITE	52
1.	DISPLAY POSITION	53
2.	SPRITE ATTRIBUTE	55
3.	SPRITE SIZE	56
4.	SPRITE's DISPLAY CAPACITY	57
5.	PRIORITY BETWEEN SPRITES	58
6.	SPRITE PATTERN GENERATOR	60
L.	PRIORITY	61
M.	COLOR PALETTE	67
N.	INTERLACE MODE	69
III.	BACKWARD COMPATIBILITY MODE	71
A.	MARK III (MS-Japan)	71
B.	MASTER SYSTEM	71
C.	RAM BOARD	71

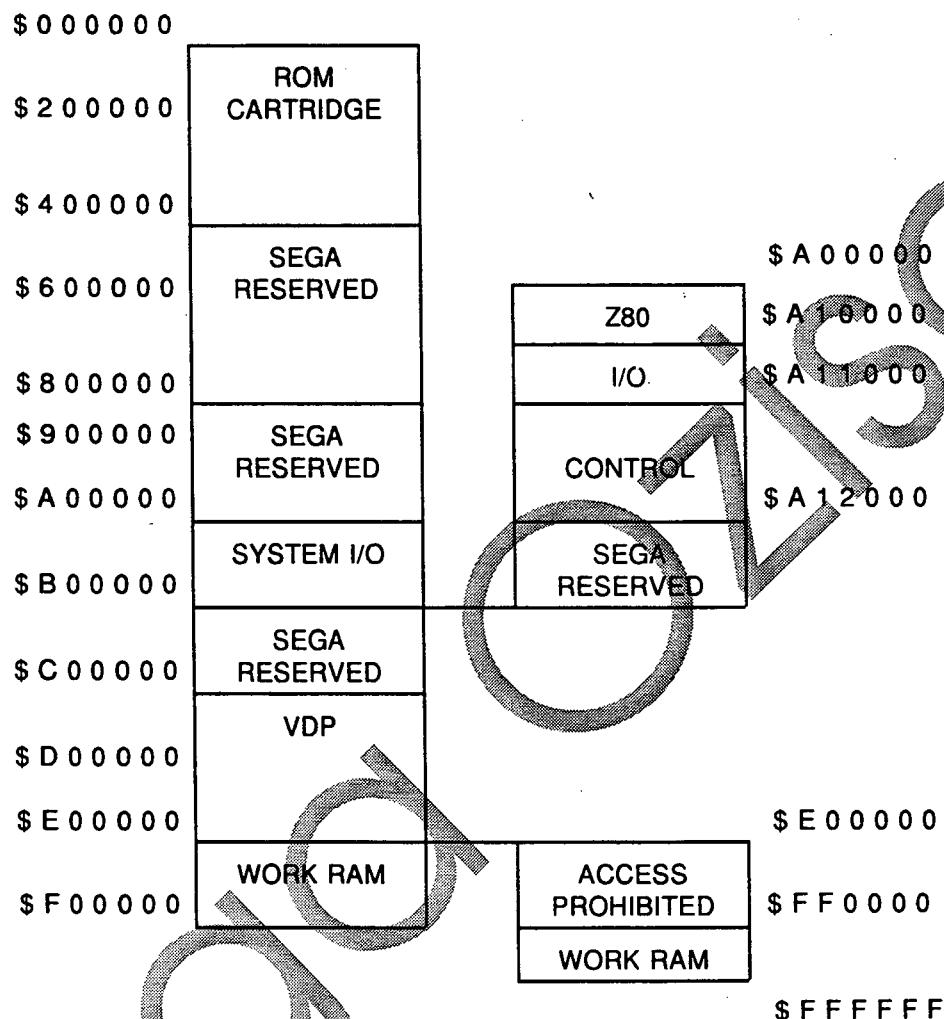
IV.	SYSTEM I/O	72
A.	VERSION NUMBER	72
B.	I/O PORT	72
C.	MEMORY MODE	76
D.	Z80 CONTROL	76
1.	Z80 BusReq	76
2.	Z80 Reset	76
E.	Z80 AREA	77
V.	VRAM MAPPING	79
VI.	SOUND SOFTWARE MANUAL	
A.	280 MAPPING	
B.	68K CONTROL OF Z80	
C.	FM SOUND CONTROL	
D.	PSG CONTROL	
E.	D/A CONTROL	
VII.	APPENDIX	

segag *Onisoft*

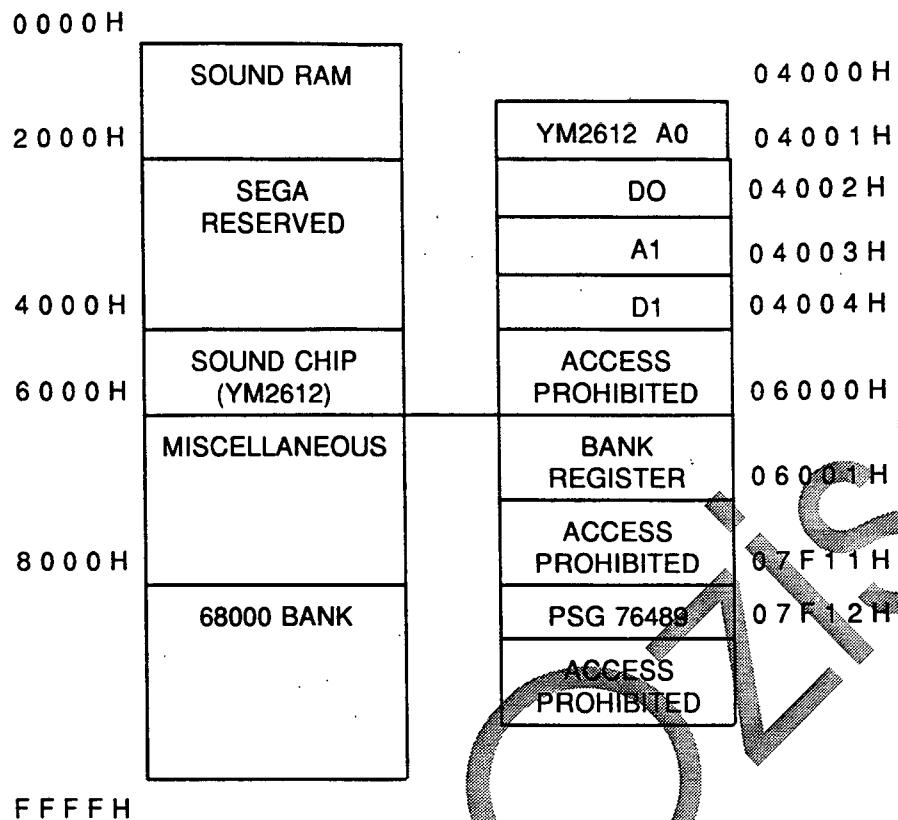
I. MEMORY MAP

A. Mega Drive 16 Bit Mode (as distinct from Master System Compatibility Mode)

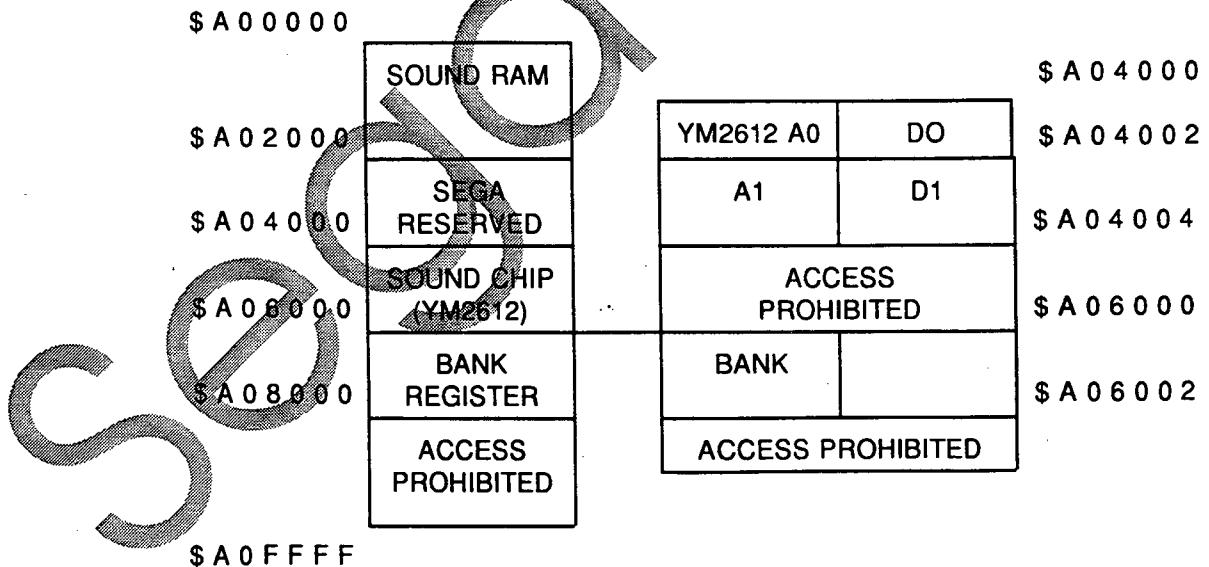
1. 68K MEMORY MAP



2. Z80 MEMORY MAP



3. 68000 ACCESS TO Z80 MEMORY



4. I/O AREA

\$ A 1 0 0 0 0	
\$ A 1 0 0 0 2	
\$ A 1 0 0 0 8	
\$ A 1 0 0 0 E	
\$ A 1 0 0 1 4	T x DATA R x DATA (1) S - MODE
\$ A 1 0 0 1 A	T x DATA R x DATA (2) S - MODE
\$ A 1 0 0 2 0	T x DATA R x DATA (E) S - MODE
\$ A 1 F F F F	ACCESS PROHIBITED

5. CONTROL AREA

\$ A 1 1 0 0 0	
\$ A 1 1 0 0 2	
\$ A 1 1 1 0 0	
\$ A 1 1 1 0 2	
\$ A 1 1 2 0 0	MEMORY MODE
\$ A 1 1 2 0 2	ACCESS PROHIBITED
\$ A 1 F F F F	Z80 BUSREQ
	ACCESS PROHIBITED
	Z80 RESET
	ACCCCESS PROHIBITED

6. VDP AREA

\$ C 0 0 0 0 0

\$ C 0 0 0 0 4

DATA

\$ C 0 0 0 0 8

CONTROL

\$ C 0 0 0 0 A

HV COUNTER

\$ C 0 0 0 1 0

ACCESS
PROHIBITED

\$ C 0 0 0 1 2

ACCESS
PROHIBITED PSG
76489

ACCESS
PROHIBITED

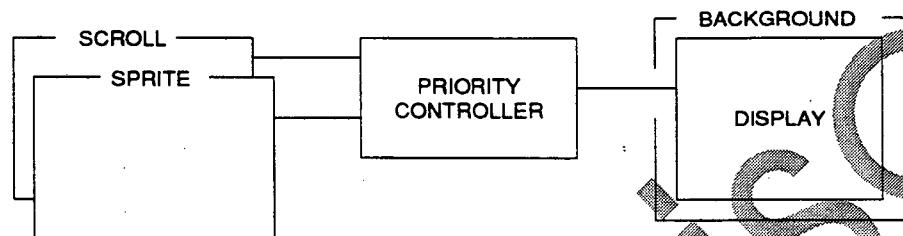
\$ D F F F F F

sega onisoft

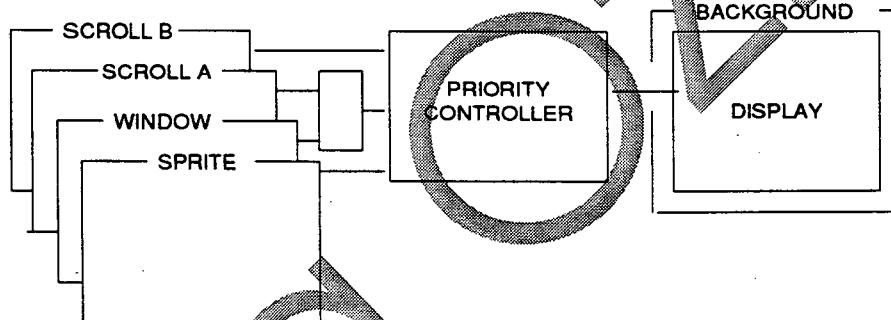
II. VDP 315-5313 (Video Display Processor)

The VDP controls screen display. VDP has graphic Modes IV and V, where Mode IV is for compatibility with the MASTER SYSTEM and V is for the new Mega drive functions. There are no advantages to using Mode IV, so it is assumed that all Mega drive development will use Mode V. In Mode V, the VDP display has 4 planes: SPRITE, SCROLLA/WINDOW, SCROLLB, and BACKGROUND.

GRAPHIC IV MODE (COMPATIBILITY MODE)



GRAPHIC V MODE (16 BIT MODE)



A. TERMINOLOGY

1. A unit of position on X, Y coordinates is called a DOT.
2. A minimum unit of display is called a PIXEL.
3. CELL means an 8 (pixel) x 8 (pixel) pattern.
4. SCROLL indicates a repositionable screen-spanning play field.
5. CPU usually indicates the 68000.
6. VDP stands for Video Display Processor.
7. CTRL stands for Control.
8. VRAM stands for VDP RAM, the 64K bytes area of RAM accessible only through the VDP.
9. CRAM stands for Color RAM, 64 9 bit words inside the VDP chip.
10. VSRAM stands for Vertical Scroll RAM, 40 10 bit words inside the VDP chip.
11. DMA stands for Direct Memory Access, the process by which the VDP performs high speed fills or memory copies.
12. PSG stands for Programmable Sound Generator, a class of low-capability sound chips. The Mega drive contains a Texas Instruments 76489 PSG chip.
13. FM stands for Frequency Modulation, a class of high-capability sound chip. The Mega drive contains a Yamaha 2612 FM chip.

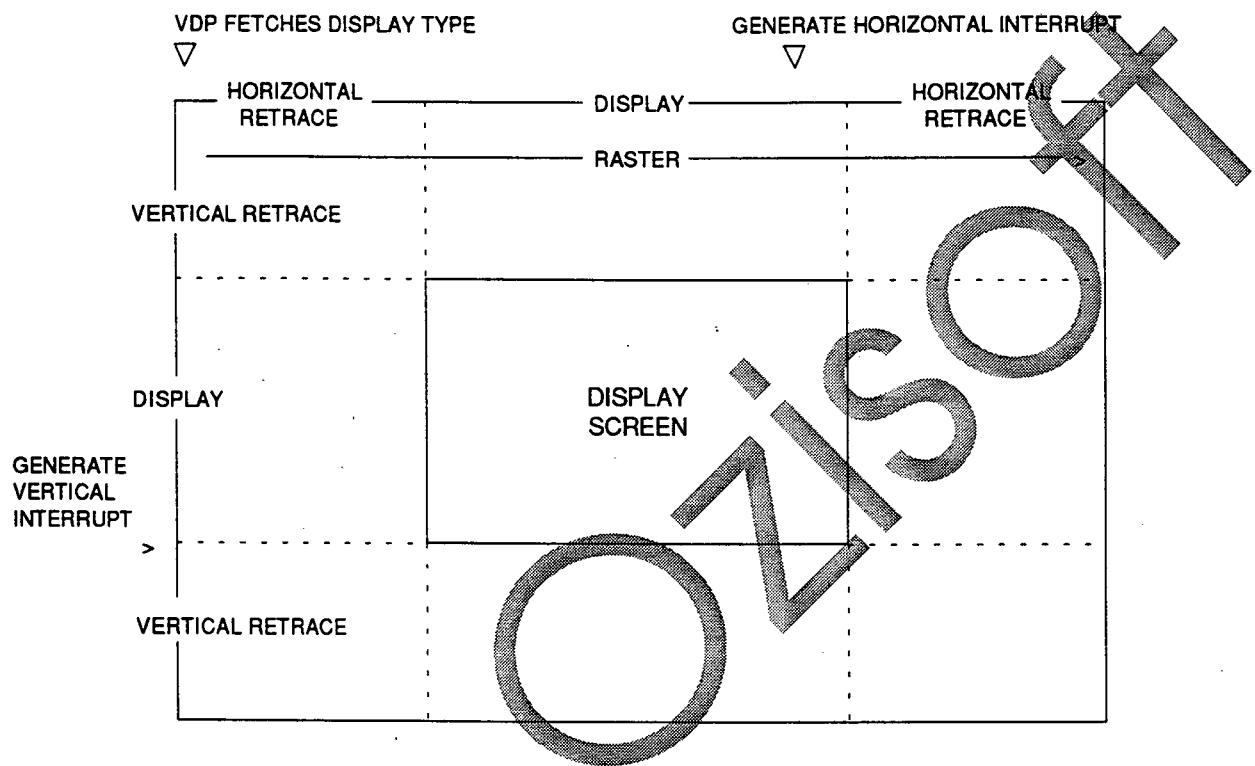
B. DISPLAY SPECIFICATION

DISPLAY SPECIFICATION OUTLINE

DISPLAY SIZE	THERE ARE TWO MODES: 32 * 28 CELL (256 * 224 PIXEL) 40 * 28 CELL (320 * 224 PIXEL)
CHARACTER GENERATOR	8 * 8 CELLS 1300-1800 depending on general system configuration.
SCROLL PLAYFIELDS	Two scrolling play fields, whose size in cells is selectable from: 32 * 32, 32 * 64, 32 * 128 64 * 32, 64 * 64 128 * 32
SPRITE	Sprite size is programmable on a sprite by sprite basis, with the following choices: 8 * 8, 8 * 16, 8 * 24, 8 * 32 16 * 8, 16 * 16, 16 * 24, 16 * 32 24 * 8, 24 * 16, 24 * 24, 24 * 32 32 * 8, 32 * 16, 32 * 24, 32 * 32 There are 64 sprites available when the screen is in 32 cell wide mode, or 80 when the screen is in 40 cell wide mode.
WINDOW	1 window associated with the Scroll A play field.
COLORS	64 colors/512 possibilities.

For PAL (the European television 50 Hz standard), a vertical size of 30 cells (240 dots) is selectable.

The VDP supports both NTSC and PAL television standards. In both cases, the screen is divided into active scan, where the picture is displayed, and vertical retrace (or vertical blanking) where the monitor prepares for the next display.



Numbers of rasters in a screen are as follows:

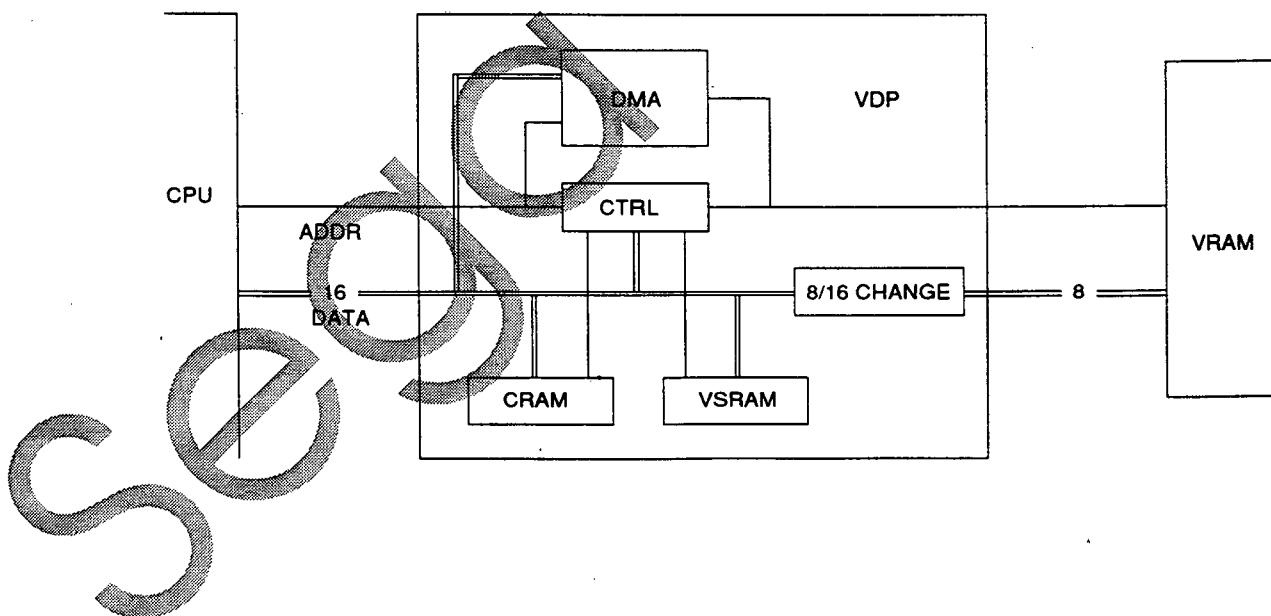
	LINES/SCREEN	VCELL NO.	LINE NO. (DISPLAY)	LINE NO. (RETRACE)
NTSC	262	28	224 RASTER	38 RASTER
PAL	312	28	224 RASTER	98 RASTER
		30	240 RASTER	82 RASTER

C. VDP STRUCTURE

The CPU controls the VDP by special I/O memory locations.

1. CTRL (control)
This controls REGISTER, VRAM, CRAM, VSRAM, DMA DISPLAY, etc.
2. VRAM (VDP RAM)
General purpose storage area for display data.
3. CRAM (COLOR RAM)
64 colors divided into 4 palettes of 16 colors each.
4. VSRAM (Vertical Scroll RAM)
Up to 20 different vertical scroll values each for scrolling play fields A and B.
5. DMA (Direct Memory Access)
The VDP may move data at high speed from CPU memory to VRAM, CRAM, and VSRAM instead of the CPU, by taking the 68000 off the bus and doing DMA itself.

The VDP can also fill the VRAM with a constant, or copy from VRAM to VRAM without disturbing the 68000.



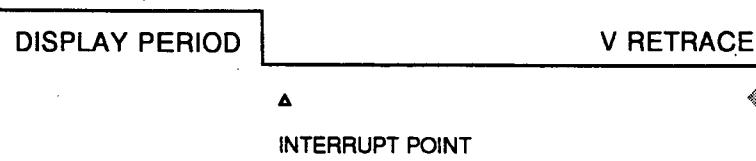
D. INTERRUPT

There are three interrupts, Vertical, Horizontal, and External. You can control each interrupt by the IEO, IE1, and IE2 bits in the VDP registers. The interrupts use the AUTO-VECTOR mode of the 68000 and are at levels 6, 4, and 2 respectively; the level 6 vertical interrupt having the highest priority.

IEO V Interrupt	(LEVEL 6)
IE1 H Interrupt	(LEVEL 4)
IE2 External Interrupt	(LEVEL 2)
1: Enable	
0: Disable	

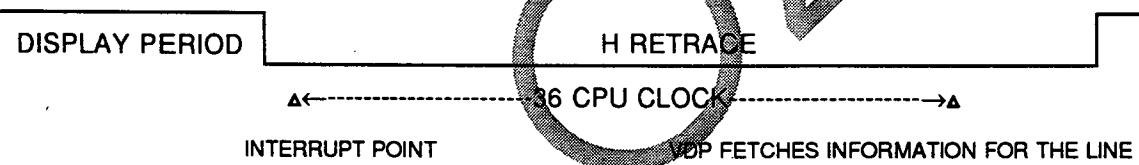
1. VERTICAL INTERRUPT (V-INT)

The vertical interrupt occurs just after V retrace.

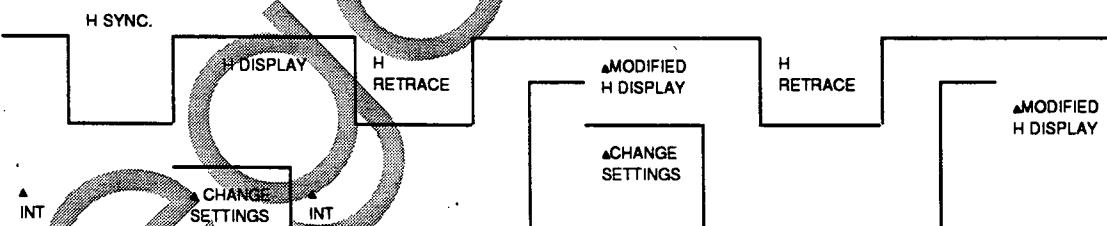


2. HORIZONTAL INTERRUPT (H-INT)

The Horizontal Interrupt occurs just before H retrace.



The VDP loads the required display information, including all required register values, for the line in about 36 clocks; thus the CPU can control the display of the next line but not the line on which the interrupt occurs.



The horizontal interrupt is controlled by a line counter in register #10.

If this line counter is changed at each interrupt, the desired spacing of interrupts may be achieved.

Thus: If register #10 equals 00h, then the interrupt occurs every line.

If register #10 equals 01h, then the interrupt occurs every other line.

If register #10 equals 02h, then the interrupt occurs every third line.

3. EXTERNAL INTERRUPT (EX-INT)

The external interrupt is generated by a peripheral device (gun, modem) and stops the H, V counter for later examination by the CPU.

HL INPUT PIN

INTERRUPT HAPPEN (COUNTER RATCHED)

Please see other sections of this manual for information about the H, V counter and the initialization of the external interrupt.

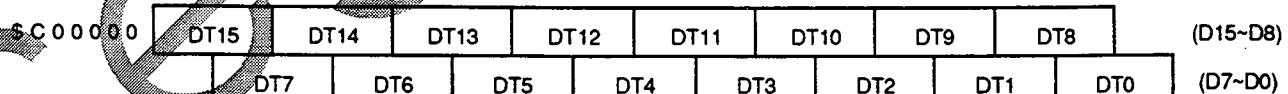
E. VDP PORT

The VDP ports are at location 68000 in the 68000 memory space.

	UPPER BYTE	LOWER BYTE
\$ C 0 0 0 0 0	DATA PORT	
\$ C 0 0 0 0 2	"	
\$ C 0 0 0 0 4	CONTROL PORT	
\$ C 0 0 0 0 6	"	
\$ C 0 0 0 0 8	HV COUNTER	
\$ C 0 0 0 0 A	PROHIBITED	
\$ C 0 0 0 0 C	PROHIBITED	
\$ C 0 0 0 0 E	PROHIBITED	
\$ C 0 0 0 1 0	PROHIBITED	PSG

1. \$ C 0 0 0 0 0 (DATA PORT)

READ/WRITE VRAM, VSRAM, CRAM



\$C00000 and \$C00002 are functionally equivalent.

2. \$C00004 (CONTROL PORT)

READ: STATUS REGISTER

\$C00004	*	*	*	*	*	*	EMPT	FULL	(D15~D8)
	F	SOVR	C	ODD	VB	HB	DMA	PAL	(D7~D0)

* NO USE

EMPT... 1: WRITE FIFO EMPTY

0:

FULL... 1: WRITE FIFO FULL

0:

F... ... 1: V interrupt happened

SOVR... 1: Sprites overflow occurred, too many in one line.

Over 17 in 32 cell mode.

Over 21 in 40 cell mode.

C... ... 1: Collision happened between non-zero pixels in two sprites.

0:

ODD ... 1: Odd frame in interlace mode.

0: Even frame in interlace mode.

VB ... 1: During V blanking

0:

HB ... 1: During H blanking

0:

DMA ... 1: DMA BUSY

0:

PAL ... 1: PAL MODE

0: NTSC MODE

WRITE1: REGISTER SET

\$C00004	1	0	0	RS4	RS3	RS2	RS1	RS0	(D15~D8)
	D7	D6	D5	D4	D3	D2	D1	D0	(D7~D0)

* \$C00004 and \$C00006 are functionally equivalent.

RS4 ~ RS0: Register No.

D7 ~ D0: Data

* You must use word or long word access to the VDP ports when setting the registers.
Long word access is equivalent to two word accesses, with D31 - D16 written first.

WRITE2: ADDRESS SET

1st \$ C 0 0 0 0 4	CD1	CD0	A13	A12	A11	A10	A9	A8	(D15~D8)
	A7	A6	A5	A4	A3	A2	A1	A0	(D7~D0)

2st \$ C 0 0 0 0 4	0	0	0	0	0	0	0	0	(D15~D8)
	CD5	CD4	CD3	CD2	0	0	A15	A14	(D7~D0)

CD5 ~ CD0: ID CODE

A15 ~ A0: DESTINATION RAM ADDRESS

ACCESS MODE	CD5	CD4	CD3	CD2	CD1	CD0
VRAM WRITE	0	0	0	0	0	1
CRAM WRITE	0	0	0	0	1	1
VSRAM WRITE	0	0	0	1	0	1
VRAM READ	0	0	0	0	0	0
CRAM READ	0	0	0	0	0	0
VSRAM READ	0	0	0	1	0	0

* You must use word or long word when performing these operations.

3. \$ C 0 0 0 0 8 (HV COUNTER)

NON INTERLACE MODE

\$ C 0 0 0 0 8	VC7	VC6	VC5	VC4	VC3	VC2	VC1	VC0	(D15~D8)
	HC8	HC7	HC6	HC5	HC4	HC3	HC2	HC1	(D7~D0)

INTERLACE MODE

\$ C 0 0 0 0 8	VC7	VC6	VC5	VC4	VC3	VC2	VC1	VC8	(D15~D8)
	HC8	HC7	HC6	HC5	HC4	HC3	HC2	HC1	(D7~D0)

HC8 ~ HC1: H COUNTER

VC8 ~ VC0: V COUNTER

F. VDP REGISTER

VDP has write only register No. 0~No. 23 and read only status register totalling 25 register. There are two modes for register setting; one is mode 4, the other mode 5. We tell you about mode 5 in this section and about mode 4 in MARK III section.

If you change mode in one frame you can get various effects.

1. MODE SET REGISTER NO.1

REG. #0	MSB	0	0	0	IE1	0	1	M3	0	LSB
---------	-----	---	---	---	-----	---	---	----	---	-----

- IE1 ... 1: Enable H interrupt (68000 LEVEL 4)
0: Disable H interrupt (REG #10)
- M3 ... 1: H, V counter stop
0: Enable read, H, V counter

2. MODE SET REGISTER NO.2

REG. #1	MSB	0	DISP	IE0	M1	M2	-1	0	0	LSB
---------	-----	---	------	-----	----	----	----	---	---	-----

- DISP ... 1: Enable Display
0: Disable Display
- IE0 ... 1: Enable V interrupt (68000 LEVEL 6)
0: Disable V interrupt
- M1 ... 1: DMA Enable
0: DMA Disable
- M2 ... 1: V 30 cell mode (PAL mode)
0: V 29 cell mode (PAL mode; always 0 in NTSC mode)

3. PATTERN NAME TABLE BASE ADDRESS FOR SCROLL A

REG. #2	MSB	0	0	SA15	SA14	SA13	0	0	0	LSB
---------	-----	---	---	------	------	------	---	---	---	-----

VRAM ADDR %XXX0_0000_0000_0000

4. PATTERN NAME TABLE BASE ADDRESS FOR WINDOW

REG. #3	MSB	0	0	WD15	WD14	WD13	WD12	WD11	0	LSB
---------	-----	---	---	------	------	------	------	------	---	-----

WD11 should be 0 in H40 cell mode
 VRAM ADDR %XXXX_X000_0000_0000 (H32 cell mode)
 VRAM ADDR %XXXX_0000_0000_0000 (H40 cell mode)

5. PATTERN NAME TABLE BASE ADDRESS FOR SCROLL B

	MSB									LSB
REG. #4	0	0	0	0	0	SB15	SB14	SB13		

VRAM ADDR %XXX0_0000_0000_0000

6. SPRITE ATTRIBUTE TABLE BASE ADDRESS

	MSB									LSB
REG. #5	0	AT15	AT14	AT13	AT12	AT11	AT10	AT9		

AT9 should be 0 in H40 cell mode

VRAM ADDR %XXXX_XXX0_0000_0000 (32 cell)

VRAM ADDR %XXXX_XX00_0000_0000 (40 cell)

	MSB									LSB
REG. #6	0	0	0	0	0	0	0	0		

7. BACKGROUND COLOR

	MSB									LSB
REG. #7	0	0	CPT1	CPT0	COL3	COL2	COL1	COL0		

CPT 1,0: COLOR PALLET

COL 3~0: COLOR CODE

	MSB									LSB
REG. #8	0	0	0	0	0	0	0	0		

	MSB									LSB
REG. #9	0	0	0	0	0	0	0	0		

8. H INTERRUPT REGISTER

	MSB									LSB
REG. #10	HIT7	HIT6	HIT5	HIT4	HIT3	HIT2	HIT1	HIT0		

This register makes H interrupt timing by number of laster.

H interrupt is enabled by IE = 1.

9. MODE SET REGISTER NO. 3

	MSB				LSB			
REG. #11	0	0	0	0	IE2	VSCR	HSCR	LSCR

IE2... ... 1: Enable external interrupt (68000 LEVEL 2)
2: Disable external interrupt

* See INTERRUPT and SYSTEM I/O

VSCR: V scroll mode

VSCR	FUNCTION
0	FULL SCROLL
1	EACH 2CELL SCROLL

HSCR, LSCR: H scroll mode

HSCR	LSCR	FUNCTION
0	0	FULL SCROLL
0	1	PROHIBITED
1	0	EACH 1CELL SCROLL
1	1	EACH 1LINE SCROLL

* BOTH SCROLL A and B

10. MODE SET REGISTER NO. 4

	MSB				LSB			
REG. #12	RS0	0	0	0	S/TE	LSM1	LSM0	RS1

RS0... ... 0: HORIZONTAL 32 CELL MODE
1: HORIZONTAL 40 CELL MODE

RS1... ... 0: HORIZONTAL 32 CELL MODE
1: HORIZONTAL 40 CELL MODE

* You should set same No. in RSO, RS1.

32 cell 0000_XXX0

40 cell 1000_XXX1

S/TE ... 1: Enable SHADOW and HIGHLIGHT
0: Disable SHADOW and HIGHLIGHT

LSM1, LSM0: Interlace mode setting

LSM1	LSM0	FUNCTION
0	0	NO INTERLACE
0	1	INTERLACE
1	0	PROHIBITED
1	1	INTERLACE (DOUBLE RESOLUTION)

11. H SCROLL DATA TABLE BASE ADDRESS

	MSB									LSB
REG. #13	0	0	HS15	HS14	HS13	HS12	HS11	HS10		

VRAM ADDR %XXXX_XX00_0000_0000

	MSB									LSB
REG. #14	0	0	0	0	0	0	0	0		

12. AUTO INCREMENT DATA

This register controls bias number of increment data.

	MSB									LSB
REG. #15	INC7	INC6	INC5	INC4	INC3	INC2	INC1	INC0		

INC7~0: Bias number (0~\$FF)

This number is added automatically after RAM access.

13. SCROLL SIZE

	MSB									LSB
REG. #16	0	0	VSZ1	VSZ2	0	0	HSZ1	HSZ0		

VSZ1, 0: VSIZE

VSZ1	VSZ0	FUNCTION
0	0	V 32 cell
0	1	V 64 cell
1	0	PROHIBITED
1		V 128 cell

HSZ1, 0: HSIZ

HSZ1	HSZ0	FUNCTION
0	0	H 32 cell
0	1	H 64 cell
1	0	PROHIBITED
1	1	H 128 cell

* Both of scroll A and B

14. WINDOW H POSITION

	MSB									LSB
REG. #17	RIGT	0	0	WHP5	WHP4	WHP3	WHP2	WHP1		

RIGT... 0: Window is in left side from base point.

1: Window is in right side from base point.

WHP5~1 Base pointer 0 = Left side

1 = 1 cell right

2 ...

15. WINDOW V POSITION

	MSB							LSB
REG. #18	DOWN	0	0	WVP4	WVP3	WVP2	WVP1	WVP0

DOWN ... 0: Window is in upper side from base point.

1: Window is in lower side from base point.

WPV4~0 Base pointer 0 = Upper side

1 = 1 cell down

2 ...

16. DMA LENGTH COUNTER LOW

	MSB									LSB
REG. #19	LG7	LG6	LG5	LG4	LG3	LG2	LG1	LG0		

17. DMA LENGTH COUNTER HIGH

	MSB									LSB
REG. #20	LG15	LG14	LG13	LG12	LG11	LG10	LG9	LG8		

LG15~0: DMA LENGTH COUNTER

18. DMA SOURCE ADDRESS LOW

	MSB									LSB
REG. #21	SA8	SA7	SA6	SA5	SA4	SA3	SA2	SA1		

19. DMA SOURCE ADDRESS MID.

	MSB									LSB
REG. #22	SA16	SA15	SA14	SA13	SA12	SA11	SA10	SA9		

20. DMA SOURCE ADDRESS HIGH

	MSB									LSB
REG. #23	DMD1	DMD0	SA22	SA21	SA20	SA19	SA18	SA17		

SA22~1: DMA SOURCE ADDRESS

DMD1, 0: DMA MODE

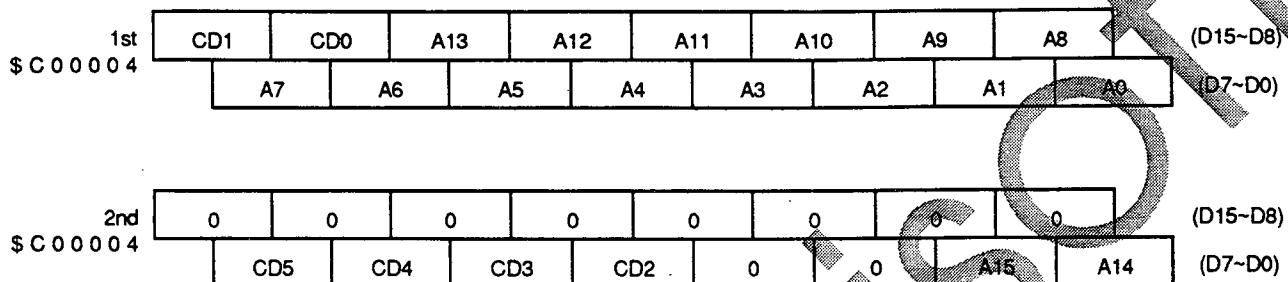
DMD1	DMD0	FUNCTION
0	SA23	MEMORY TO VRAM
1	0	VRAM FILL
1	1	VRAM COPY

G. ACCESS VDP RAM

1. RAM ADDRESS SETTING

You can access VRAM, CRAM and VSRAM after writing 32 bits of control data to \$C00004 or \$C00006.

You have to use word or long word when addressing. If you use long word D31~D16 is 1st, D15~D0 2nd.



CD5~CD0: ID CODE

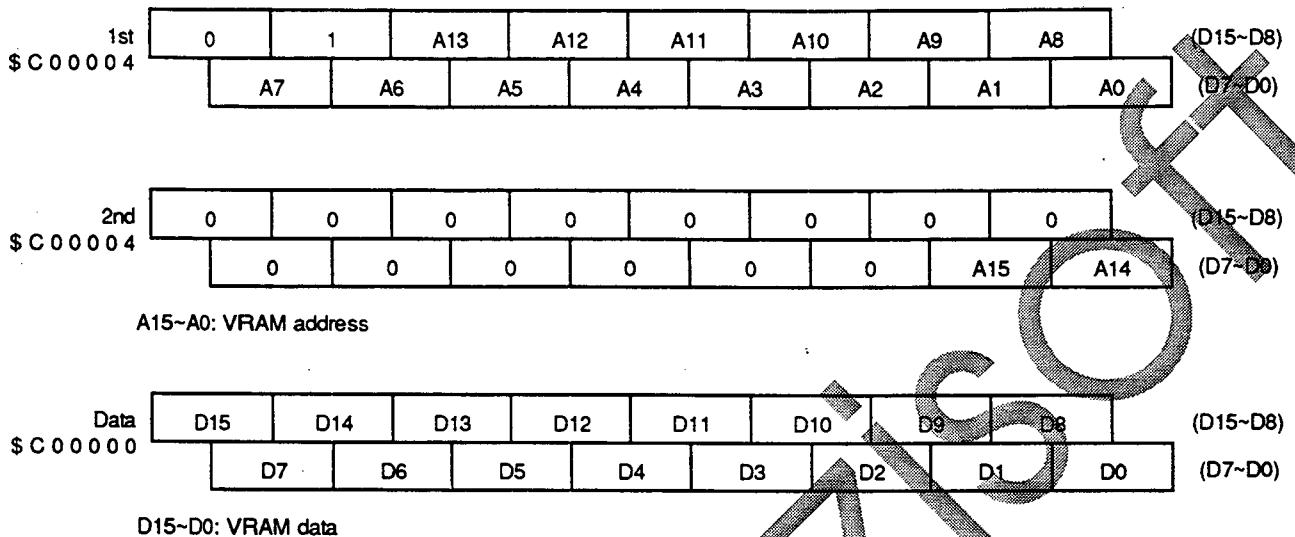
A15~A0: DESTINATION RAM ADDRESS

	CD5	CD4	CD3	CD2	CD1	CD0
VRAM WRITE	0	0	0	0	0	1
CRAM WRITE	0	0	0	0	1	1
VSRAM WRITE	0	0	0	1	0	1
VRAM READ	0	0	1	0	0	0
CRAM READ	0	0	1	0	0	0
VSRAM READ	0	0	0	1	0	0

2. VRAM ACCESS

VRAM address range from 0 to 0FFFFH, 64K bytes total.

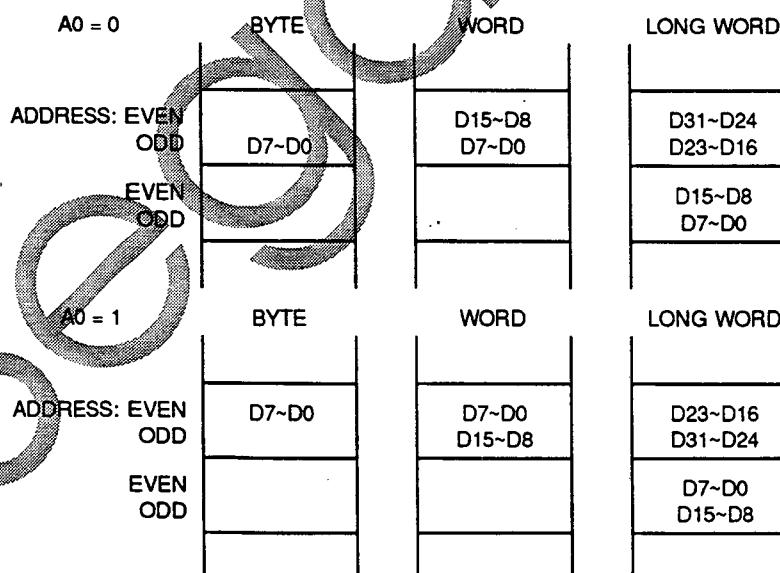
VRAM access addressing is as follows when writing:



When you use long word D31~D16 is 1st, D15~D0 is 2nd. When you do byte writing, data is D7~D0, and may be written to \$C00000 or \$C00001.

VRAM address is increased by the value of REGISTER #15, independent data size. VRAM address A0 is used in the calculation of the address increment, but is ignored during address decoding.

VRAM addressing and decoding are as follows: the VRAM address decode uses A15~A1, and A0 specifies the data write format. Write data cannot cross a word boundary, high and low bytes are exchanged if A0=1.



(EXAMPLE) START ADDRESS: 0 REG. #15=2

ADDRESS:	BYTE		WORD		LONG WORD	
	1st	D7~D0	1st	D15~D8 D7~D0	1st	D31~D24 D23~D16
1	2nd	D7~D0	2nd	D15~D8 D7~D0	2nd	D31~D24 D23~D16
2	3rd	D7~D0	3rd	D15~D8 D7~D0	3rd	D31~D24 D23~D16
3	4th	D7~D0	4th	D15~D8 D7~D0	4th	D31~D24 D23~D16
4	5th	D7~D0	5th	D15~D8 D7~D0	5th	D31~D24 D23~D16
5						
6						
7						
8						
9						

START ADDRESS: 0 REG. #15=1

ADDRESS:	BYTE		WORD		LONG WORD	
	2nd	D7~D0	2nd	D7~D0 D15~D8	1st	D7~D0 D15~D8
0	1st	D7~D0	4th	D7~D0 D15~D8	2nd	D7~D0 D15~D8
1	3rd	D7~D0	6th	D7~D0 D15~D8	3rd	D7~D0 D15~D8
2	4th	D7~D0	8th	D7~D0 D15~D8	4th	D7~D0 D15~D8
3	5th	D7~D0	10th	D7~D0 D15~D8	5th	D7~D0 D15~D8
4	6th	D7~D0				
5	7th	D7~D0				
6	8th	D7~D0				
7	9th	D7~D0				
8	10th	D7~D0				
9						

START ADDRESS: 1 REG. #15=2

	BYTE	WORD	LONG WORD
ADDRESS: 0	1st D7~D0	1st D7~D0 D15~D8	1st D23~D16 D31~D24
1			
2	2nd D7~D0	2nd D7~D0 D15~D8	1st D23~D16 D31~D24
3			
4	3rd D7~D0	3rd D7~D0 D15~D8	2nd D23~D16 D31~D24
5			
6	4th D7~D0	4th D7~D0 D15~D8	2nd D23~D16 D31~D24
7			
8	5th D7~D0	5th D7~D0 D15~D8	3rd D23~D16 D31~D24
9			

START ADDRESS: 1 REG. #15=1

	BYTE	WORD	LONG WORD
ADDRESS: 0	1st D7~D7	1st D7~D0 D15~D8	1st D23~D16 D31~D24
1			
2	3rd D7~D7	3rd D7~D0 D15~D8	2nd D23~D16 D31~D24
3	2nd D7~D7		
4	5th D7~D7	5th D7~D0 D15~D8	3rd D23~D16 D31~D24
5	4th D7~D7		
6	7th D7~D7	7th D7~D0 D15~D8	4th D23~D16 D31~D24
7	6th D7~D7		
8	9th D7~D7	9th D7~D0 D15~D8	5th D23~D16 D31~D24
9	8th D7~D7		

VRAM READ

1st	0	0	A13	A12	A11	A10	A9	A8	(D15~D8)
\$ C 0 0 0 0 4	A7	A6	A5	A4	A3	A2	A1	A0	(D7~D0)

2nd	0	0	0	0	0	0	0	0	(D15~D8)
\$ C 0 0 0 0 4	0	0	0	0	0	0	0	A15	(D7~D0)

A15~A0: VRAM ADDRESS

Data	D15	D14	D13	D12	D11	D10	D9	D8	(D15~D8)
\$ C 0 0 0 0 0	D7	D6	D5	D4	D3	D2	D1	D0	(D7~D0)

D15~D0: VRAM DATA

The data is always read in word units. A0 is ignored during the reading; no swap of bytes occurs if A0=1.

Subsequent reads are from address incremented by REGISTER #15. A0 is used in calculation of the next address.

3. CRAM ACCESS

The CRAM contains 128 bytes, addresses 0 to 7FH. For word wide writes to the CRAM, use:

1st	1	1	0	0	0	0	0	0	(D15~D8)
\$ C 0 0 0 0 4	0	A6	A5	A4	A3	A2	A1	A0	(D7~D0)

2nd	0	0	0	0	0	0	0	0	(D15~D8)
\$ C 0 0 0 0 4	0	0	0	0	0	0	0	0	(D7~D0)

A6~A0: CRAM ADDRESS

Data	0	0	0	0	B2	B1	B0	0	(D15~D8)
\$ C 0 0 0 0 0	G2	G1	G0	0	R2	R1	R0	0	(D7~D0)

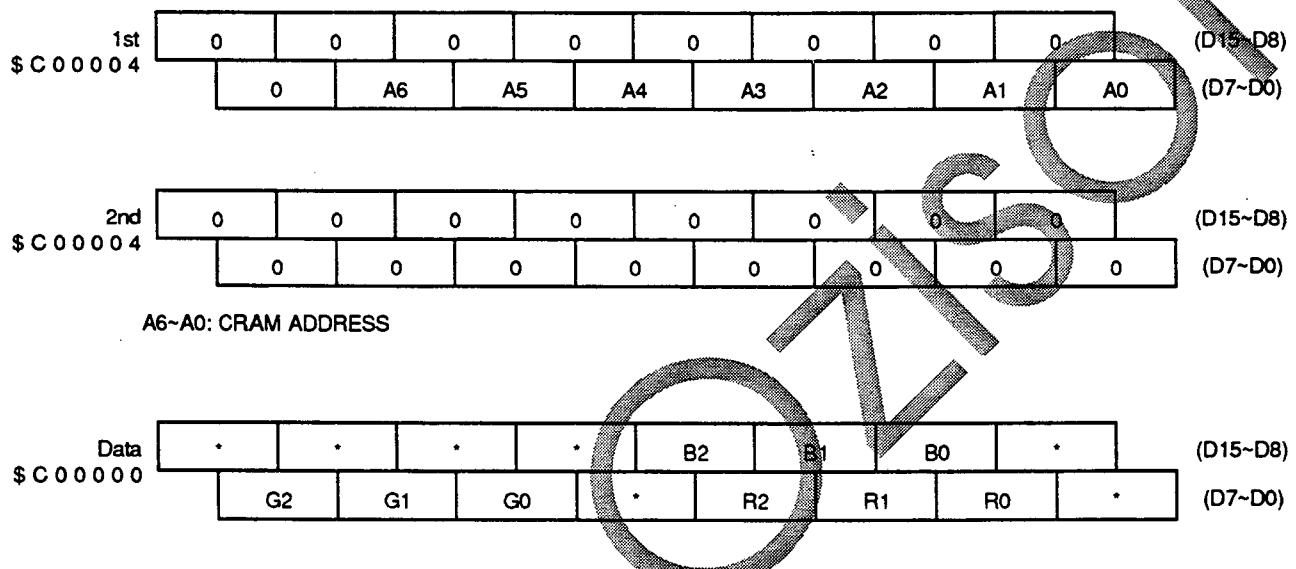
D15~D0 are valid when we use word for data set. If the writes are byte wide, write the high byte to \$C00000 and the low byte to \$C00001.

A long word wide access is equivalent to two sequential word wide accesses. Place the first data in D31 - D16, and the second data in D15 - D0.

The data may be written sequentially; the address is incremented by the value of REGISTER #15 after every write, independent of whether the width is byte or word.

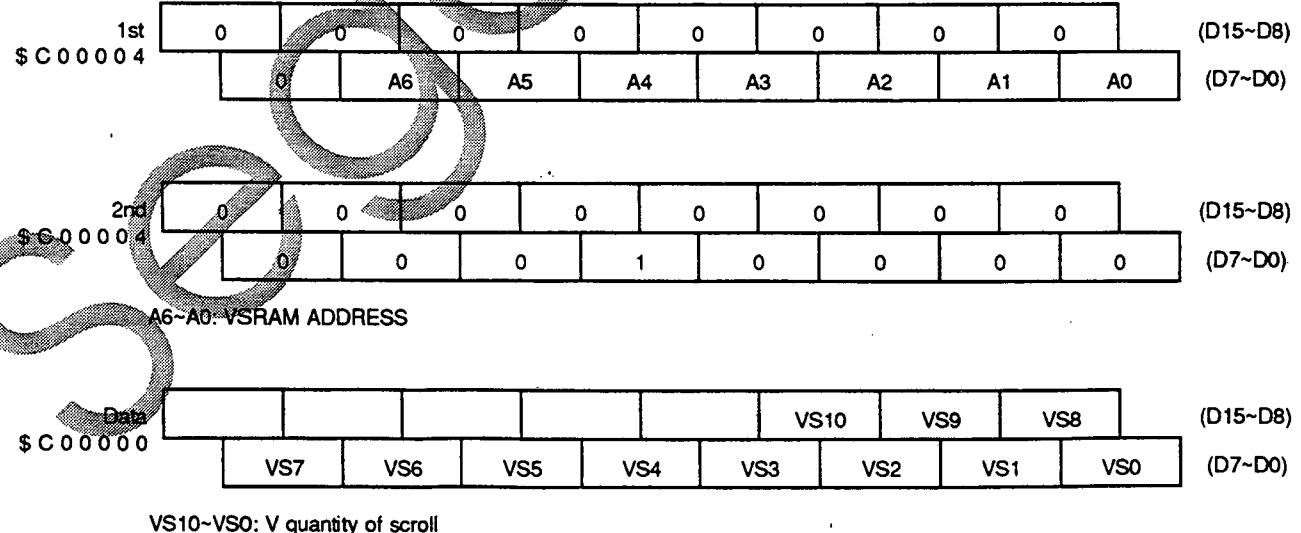
Note that A0 is used in the increment, but not in address decoding, resulting in some interesting side effects if writes are attempted at odd addresses.

For word wide reads from the CRAM, use:



4. VSRAM ACCESS

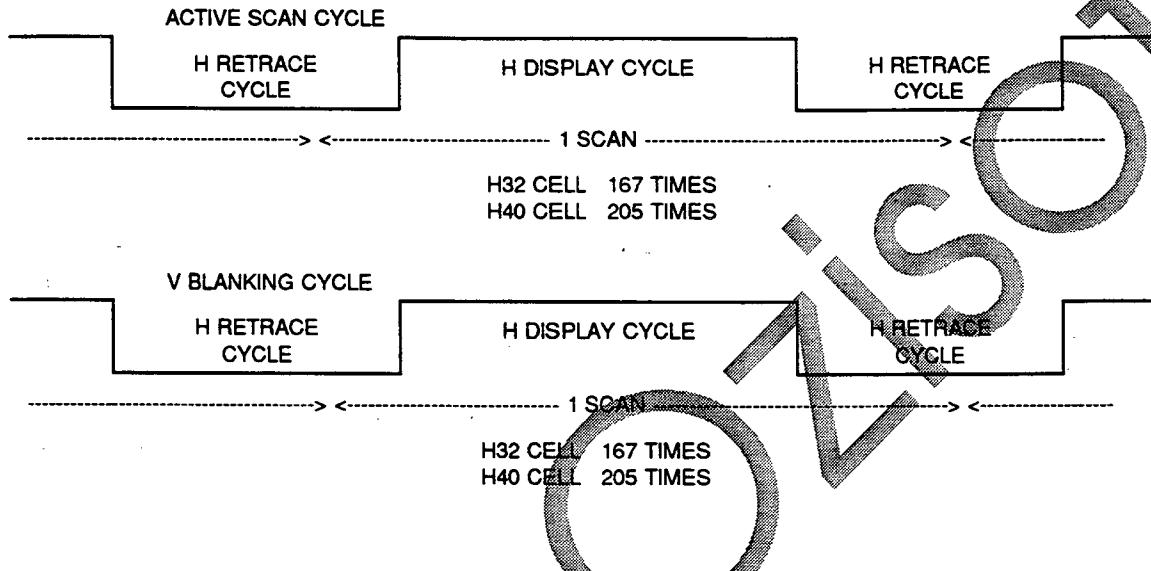
The VSRAM contains 80 bytes, addresses 0 to 4FH. For word wide writes to the VSRAM, use:



5. ACCESS TIMING

The CPU and VDP access VRAM, CRAM and VSRAM using timesharing. Because the VDP is very busy during the active scan, the CPU accesses are limited. However, during vertical blanking, the CPU may access the VDP continuously.

The number of permitted accesses by the CPU additionally depends on whether the screen is in 32 cell mode or 40 cell mode. Additionally, the access size depends on the RAM type; a VRAM access is byte wide, but CRAM and VSRAM are word wide.



For example, in 32 Cell mode, the CPU may access the VRAM 16 times during horizontal scan in a single line. Each access is a byte write, so this amounts to 8 words. However, CRAM and VSRAM, though sharing the 16 time limit, are word accesses so that 16 words may be written in a single line.

Although there is a four-word FIFO, if writes are done in a tight loop during active scan, the FIFO will fill up and the CPU will eventually end up waiting to write.

The maximum wait times are:

Display Mode	Maximum Waiting Time
H32 cell	Approximately 5.96 μ sec
H40 cell	Approximately 4.77 μ sec

As the CPU has unlimited access to the RAMs during vertical blanking, the wait case never arises.

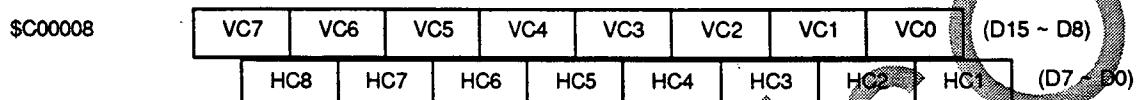
6. HV COUNTER

The HV Counter's function is to give the horizontal and vertical location of the television beam. If the "M3" Bit of Register #0 is set, the HV Counter will then freeze when trigger signal HL goes high, as well as triggering a level 2 interrupt.

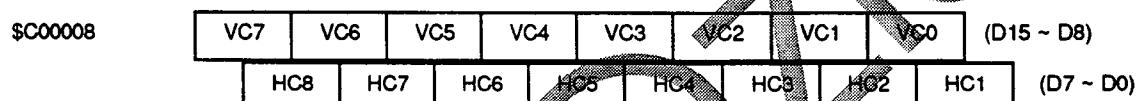
M3	Counter Latch Mode
0	Counter is not latched by trigger signal
1	Counter is latched by trigger signal

M3: Register #0

Non-Interlace Mode



Interlace Mode



V-Counter: VC7 ~ VC0

Display Mode	Counter Data
V 28 Cell	0 ~ DFH
V 30 Cell	0 ~ EFH

Display Mode	Counter Data
H 32 Cell	0 ~ 7FH
H 40 Cell	0 ~ 9FH

The counter only has eight bits each for H and V, so Interlace Mode and 40 Cell (320 dot) modes present some problems. During Interlace Mode, the LSB of the vertical position is replaced by the new MSB. The horizontal resolution problem is solved by always dropping the LSB.

Caution:

As the HV Counter's value is not valid during vertical blanking, check to be sure that it is active scan before using the value.

H. DMA TRANSFER

DMA (Direct Memory Access) is a high-speed technique for memory access to the VRAM, CRAM, and VSRAM. During DMA, VRAM, CRAM and VSRAM occur at the fastest possible rate (refer to the section on Access Timing). There are three modes of DMA access, as can be seen below, all of which may be done to VRAM or CRAM or VSRAM. The 68K is stopped during memory to VRAM/CRAM/VSRAM DMA, but the Z80 continues to run as long as it does not attempt access to the 68K memory space.

The DMA is quite fast during VBLANK (about double the tightest possible 68K loop's speed), but during active scan the speed is the same as a 68K loop.

Please note that after this point, VRAM is used as a generic term for VRAM/CRAM/VSRAM.

DMD1	DMD0	DMA Mode	Size
0	SA23	A. Memory to V-RAM	Word to Byte (H) & (L)
1	0	B. VRAM Fill	Byte to Byte
1	1	C. VRAM Copy	Byte to Byte

DMD1, DMD0: Reg. #23 * DMD0 = SA23

Source address are \$000000~\$3FFFFF (ROM) and \$FF0000~\$FFFFFF (RAM) for memory to VRAM transfers. In the case of ROM to VRAM transfers, a hardware feature causes occasional failure of DMA unless the following two conditions are observed:

- The destination address write (to address \$C00004) must be a word write.
- The final write must use the work RAM. There are two ways to accomplish this: (1) by copying the DMA program into RAM, or (2) by doing a final "move W RAM address, \$C00004"

1. MEMORY TO VRAM

This function transfers data from 68K memory to VRAM, CRAM or VSRAM. During this DMA all 68K processing stops. The source address is \$000000~\$3FFFFF for ROM or \$FF0000~\$FFFFFF for RAM. The DMA reads are word-wide, writes are byte-wide for VRAM and word-wide for CRAM and VSRAM. The destination is specified by:

CD2	CD1	CD0	Memory Type
0	0	1	VRAM
0	1	1	CRAM
1	0	1	VSRAM

Setting of DMA

- A. M1 (Register #1) = 1 : DMA ENABLE.
- B. Increment number set to Register #15 (normally #2).
- C. Transfer word number set to Registers #19, #20.
- D. Source address and DMA mode set into Registers #21, #22 and #23.
- E. Set the destination address.
- F. *VDP gets the CPU bus.
- G. *DMA start.
- H. *VDP releases the CPU bus.
- I. M1 has to be 0 after confirmation of DMA finish : DMA DISENABLE.

DMA starts after Step E.

You must set M1=1 only during DMA; otherwise, we cannot guarantee the operation.
Source addresses were increased with +2 and destination address increased with content of Register #15.

Content of Register. Register #1 has another bit.

Register #15	INC7	INC6	INC5	INC4	INC3	INC2	INC1	INC0
--------------	------	------	------	------	------	------	------	------

INC7 ~ INC0 : Number of increment

Register #1	0	DISP	IEO	M1	M2	1	0	0
-------------	---	------	-----	----	----	---	---	---

Register #19	LG7	LG6	LG5	LG4	LG3	LG2	LG1	LG0
--------------	-----	-----	-----	-----	-----	-----	-----	-----

Register #20	LG15	LG14	LG13	LG12	LG11	LG10	LG9	LG8
--------------	------	------	------	------	------	------	-----	-----

Register #21	SA8	SA7	SA6	SA5	SA4	SA3	SA2	SA1
--------------	-----	-----	-----	-----	-----	-----	-----	-----

Register #22	SA16	SA15	SA14	SA13	SA12	SA11	SA10	SA9
--------------	------	------	------	------	------	------	------	-----

Register #23	0	SA23	SA22	SA21	SA20	SA19	SA18	SA17
--------------	---	------	------	------	------	------	------	------

1st \$C00004	CD1	CD0	DA13	DA12	DA11	DA10	DA9	DA8	(D15 ~ D8)
	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0	(D7 ~ D0)

2nd \$C00004	0	0	0	0	0	0	0	0	(D15 ~ D8)
	1	0	0	CD2	0	0	DA15	DA14	(D7 ~ D0)

LG15 ~ LG0: Number of move word

SA23 ~ SA1: Source address (in 68000)

DA15 ~ DA0: Destination address (in VDP)

CD2 ~ CD0: RAM selection

2. VRAM FILL

FILL mode fills with same data from free even VRAM address. FILL for only VRAM.

How to set FILL (DMA):

- A. M1 (Register #1) = 1 : DMA ENABLE
- B. Increment number set to #15 (normally #1)
- C. Fill size set to #19, #20.
- D. DMA mode set to #23.
- E. Destination address and FILL data set.
- F. *DMA start.
- G. M1=0 after confirmation of finishing : DMA DISENABLE

DMA starts after Step E.

M1 should be 1 in the DMA transfer; otherwise, we cannot guarantee the operation.

Destination address is incremented with Register #15. VDP does not ask but open for CPU but CPU cannot access VDP without PSG, HV counter, and status. You can realize end of DMA by DMA bit in status register.

Register Setting. Register #1 has another bit.

Register #15

INC7	INC6	INC5	INC4	INC3	INC2	INC1	INC0
------	------	------	------	------	------	------	------

INC7 ~ INC0: Increment number

Status

*	*	*	*	*	*	EMPT	FULL
F	SOVR	C	ODD	VB	HB	DMA	PAL

DMA: 1: DMA Busy

*: Not care

Register #1

0	DISP	IEO	M1	M2	1	0	0
---	------	-----	----	----	---	---	---

Register #19

LG7	LG6	LG5	LG4	LG3	LG2	LG1	LG0
-----	-----	-----	-----	-----	-----	-----	-----

Register #20

LG15	LG14	LG13	LG12	LG11	LG10	LG9	LG8
------	------	------	------	------	------	-----	-----

Register #23

1	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

1st \$C00004

0	1	DA13	DA12	DA11	DA10	DA9	DA8	(D15 ~ D8)
DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0	(D7 ~ D0)

2nd \$C00004

0	0	0	0	0	0	0	0	(D15 ~ D8)	
1	0	0	0	0	0	0	DA15	DA14	(D7 ~ D0)

\$C00000

FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	(D15 ~ D8)
FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0	(D7 ~ D0)

LG15 ~ LG0: Fill byte number

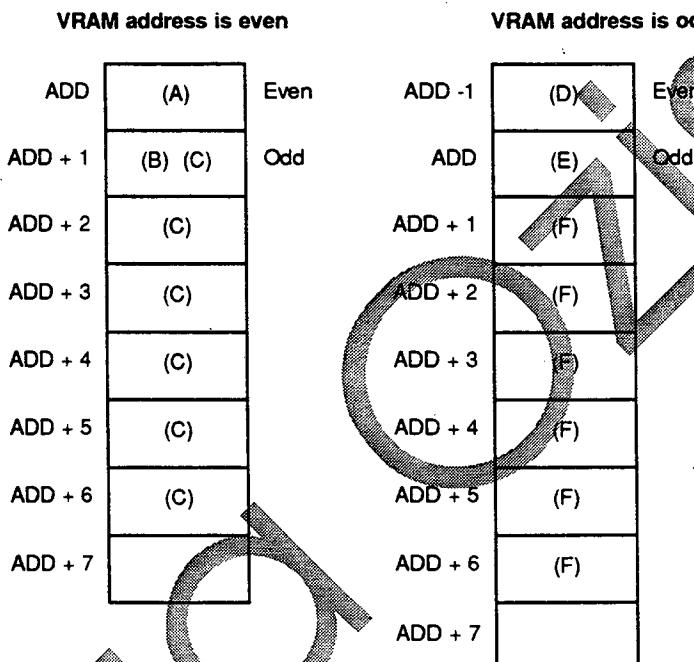
DA15 ~ DA0: Destination address

FD15 ~ FD0: FILL data

When setting first and second by long word, first will be D31-D16 and second will be D15-D0.

Example 1 Fill data are word. Register #15 = 1.

1. V-RAM address is even.
 - A. First, low side of Fill data are written in V-RAM address.
 - B. Second, upper side of Fill data are written in V-RAM+1.
 - C. V-RAM address is added register #15, written upper side Fill data in V-RAM at next each step.
2. V-RAM address is odd.
 - D. First, upper side of FILL data are written in V-RAM address -1.
 - E. Second, low side of FILL data are written in V-RAM.
 - F. Same as (C.)



You must rewrite data (C) into ADD + 1 after write data (B).

Example 2 Fill data are word. Register #15 = 2.

VRAM address = even

ADD	(A) lower
ADD + 1	(B) upper
ADD + 2	(C) lower
ADD + 3	upper
ADD + 4	(C) lower
ADD + 5	upper
ADD + 6	(C) lower
ADD + 7	upper

Even

Odd

VRAM address = odd

ADD - 1	(D) upper
ADD	(E) lower
ADD + 1	
ADD + 2	(F) upper
ADD + 3	lower
ADD + 4	(F) upper
ADD + 5	lower
ADD + 6	(F) upper
ADD + 7	lower

Even

Odd

Example 3 Fill data are byte.

1. V-RAM address is even
(A) = (B) = (C) = BYTE DATA

2. V-RAM address is odd
(D) = (E) = (F) = BYTE DATA

3. VRAM COPY

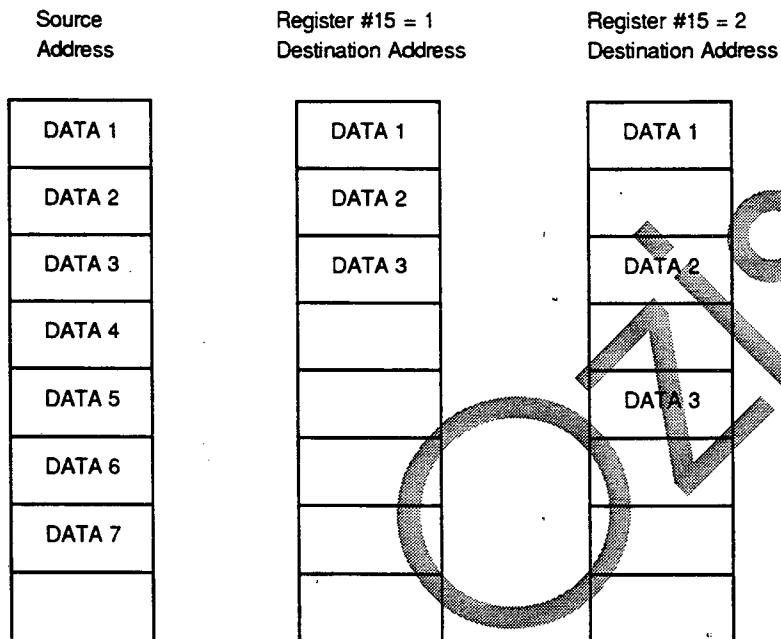
This function copies from source address to destination address by number of Copy byte.

DMA Setting

- A. M1 (Register #1) = 1 : DMA ENABLE.
- B. Number of copy bytes in #19, #20
- C. Source address and DMA mode in #23.
- D. Destination address set.
- E. *DMA transfer.
- F. After confirming DMA finish: M1 = 0 : DMA DISENABLE

DMA starts when (D) above is finished. Apply M1=1 only during DMA transfer. In other cases, if M1 = 1 is set, there is no guarantee that it will function correctly. At the time of DMA transfer, the destination address is incremented by the set value of Register #15. During DMA transfer, although the VDP does not require CPU to make a bus available, no access is possible from CPU to VDP except for PSG, HV counter, Status Read. DMA transfer finish can be recognized by referring to the Status Register's DMA bit.

Example With Transfer byte = 3 at the time of VRAM Copy



Caution

In the case of VRAM Copy, "read from VRAM" and "write to VRAM" are repeated per byte. Therefore, when the Source Area and Transfer Area are overlapped, the transfer may not be performed correctly.

Registers are as follows. Register #1 includes bits set for purposes other than DMA. Therefore, pay careful attention in this regard.

Register #15

INC7	INC6	INC5	INC4	INC3	INC2	INC1	INC0
------	------	------	------	------	------	------	------

INC7 ~ INC0: Increment number

Status

*	*	*	*	*	*	*	EMPT	FULL
F	SOVR	C	ODD	VB	HB	DMA	PAL	

DMA: 1: DMA Busy

Register #1:

0	DISP	IEO	M1	M2	1	0	0
---	------	-----	----	----	---	---	---

Register #19

LG7	LG6	LG5	LG4	LG3	LG2	LG1	LG0
-----	-----	-----	-----	-----	-----	-----	-----

Register #20

LG15	LG14	LG13	LG12	LG11	LG10	LG9	LG8
------	------	------	------	------	------	-----	-----

Register #21

SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0
-----	-----	-----	-----	-----	-----	-----	-----

Register #22

SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8
------	------	------	------	------	------	-----	-----

Register #23

1	1	0	0	0	0	0	0
---	---	---	---	---	---	---	---

1st \$C00004

0	0	DA13	DA12	DA11	DA10	DA9	DA8
---	---	------	------	------	------	-----	-----

(D15 ~ D8)

DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
-----	-----	-----	-----	-----	-----	-----	-----

(D7 ~ D0)

2nd \$C00004

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

(D15 ~ D8)

(D7 ~ D0)

1	1	0	0	0	0	0	DA15	DA14
---	---	---	---	---	---	---	------	------

LG15 ~ LG0: Number of copy byte

SA15 ~ SA0: Source address

DA15 ~ DA0: Destination address

When setting first and second by long word, first will be D31~D16 and second will be D15~D0.

4. DMA TRANSFER CAPACITY

Transfer quantity varies, depending on the Display Mode as follows:

DMA Mode	Display Mode	Screen Scanning	Transfer bytes per line
MEMORY TO VRAM	H32 Cell	During effective screen During V Blank	16 bytes 167 bytes
	H40 Cell	During effective screen During V Blank	18 bytes 205 bytes
VRAM FILL	H32 Cell	During effective screen During V Blank	15 bytes 166 bytes
	H40 Cell	During effective screen During V Blank	17 bytes 204 bytes
VRAM COPY	H32 Cell	During effective screen During V Blank	8 bytes 83 bytes
	H40 Cell	During effective screen During V Blank	9 bytes 102 bytes

In the Memory to VRAM, in the case where CRAM and VSRAM are the destinations, number of words (not bytes) should apply. One line during V Blank allows for data transfer to all the addresses of CRAM and VSRAM.

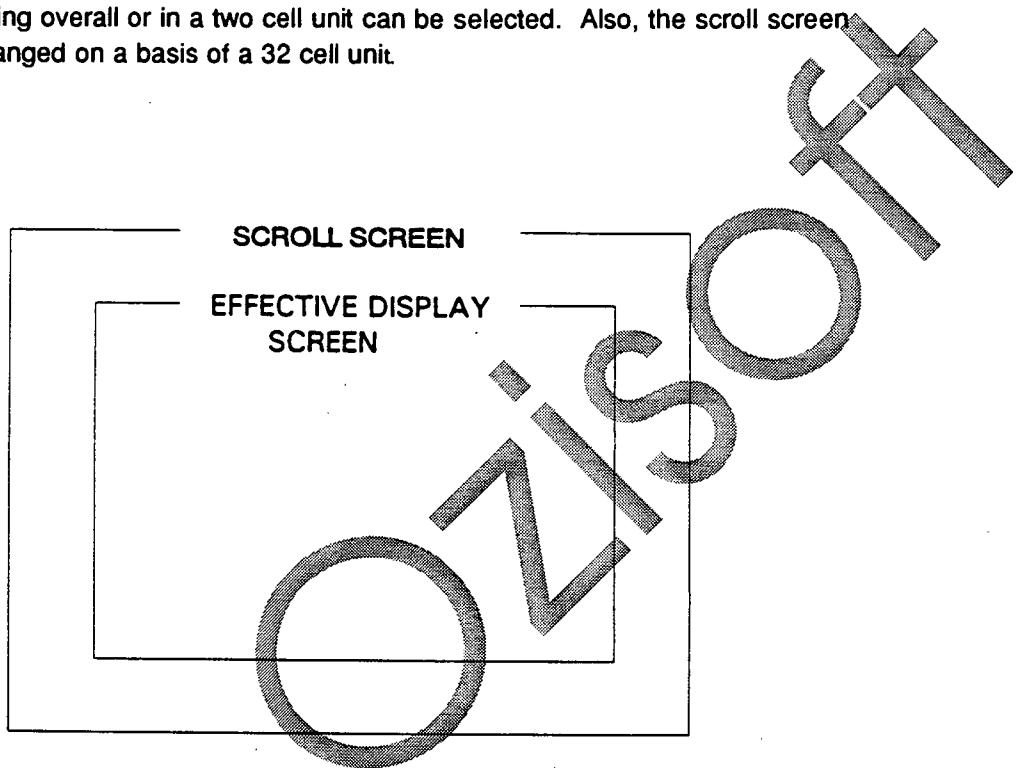
Note that when calculating, the transfer quantity in one screen (1/60 sec) varies depending on the number of lines during V Blank (refer to *Display Mode*) in the case of NTSC (video signal) and PAL systems.

Display Mode	No. of Horizontal Lines
V28 Cell (NTSC)	36
V28 Cell (PAL)	87
V30 Cell (PAL)	71

Where Register #1 DISP=0, i.e., when on-screen display is not made, the Transfer quantity is the same as Transfer Bytes Per Line during Blanking.

I. SCROLLING SCREEN

There are two different scroll screens, A and B, which separately can scroll vertically and horizontally on a basis of a one-dot unit. In the horizontal direction, scrolling overall or based on a one cell unit or one line unit can be selected, and in the vertical direction, scrolling overall or in a two cell unit can be selected. Also, the scroll screen size can be changed on a basis of a 32 cell unit.



sega

For the scrolling screen display, the following register setting and VRAM are required:

Scroll "A" Pattern Name Table Base Address

Register #2	0	0	SA15	SA14	SA13	0	0	0
-------------	---	---	------	------	------	---	---	---

Scroll "B" Pattern Name Table Base Address

Register #4	0	0	0	0	0	SB15	SB14	SB13
-------------	---	---	---	---	---	------	------	------

Mode Set Register #3

Register #11	0	0	0	0	IE2	VSCR	HSCR	LSCR
--------------	---	---	---	---	-----	------	------	------

Mode Set Register #4

Register #12	RS0	0	0	0	S/TE	LSM1	LSM0	RS1
--------------	-----	---	---	---	------	------	------	-----

H Scroll Data Table Base Address

Register #13	0	0	HS15	HS14	HS13	HS12	HS11	HS10
--------------	---	---	------	------	------	------	------	------

Scroll Size

Register #16	0	0	VSZ1	VSZ0	0	0	HSZ1	HSZ0
--------------	---	---	------	------	---	---	------	------

VRAM

Scroll "A" Pattern Name Table Max 8 Kbyte
Scroll "B" Pattern Name Table Max 8 Kbyte
H Scroll Data Table Max 960 byte

VSRAM

V-Scroll Data Table Max 80 byte

1. SCROLLING SCREEN SIZE

The screen size can be set by VSZ1, VSZ0, HSZ1, and HSZ0 (Register #16). The following six kinds can be set for both Scroll Screens A and B.

32*32/32*64/32*128

64*32/64*64

128*32

VSZ1	VSZ0	Function
0	0	V 32 cell
0	1	V 64 cell
1	0	Prohibited
1	1	V 128 cell

HSZ1	HSZ0	Function
0	0	H 32 cell
0	1	H 64 cell
1	0	Prohibited
1	1	H 128 cell

Scroll Screen's Pattern Name Table Address exists in the VRAM and is designated by Registers #2 and #4. Depending VRAM and Scroll Screen correspond to each other differently.

Example

Register #16 = 00H: 32*32 cell

0		1		32 CELL		30		31	
0	0000	0002				003c	003e		
1	0040	0042				007c	007e		
32 cell									
30	0780	0782				07bc	07be		
31	07c0	07c2				07fc	07fe		

Example

Register #16 = 11H: 64*64 cell

	0	1	64 CELL	62	63
0	0000	0002	~	007c	007e
1	0080	0082		00fc	00fe
64 cell					
62	1f00	1f02	~	1f7c	1f7e
63	1fc0	1fc2	~	1ffc	1ffe

Example

Register #16 = 03H: 32*128 cell

	0	1	128 CELL	126	127
0	0000	0002	~	00fc	00fe
1	0100	0102		01fc	01fe
32 cell					
30	1e00	1e02	~	1efc	1efe
31	1f00	1f02	~	1ffc	1ffe

A value shown in a frame indicates an offset from the Pattern Name Table Base Address.

2. HORIZONTAL SCROLLING

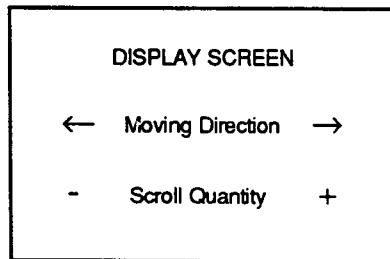
The Display Screen allows for scrolling overall or based on one cell unit, or on a dot by dot basis in one line unit. Either one of the above scrolling can be selected by HSCR and LSCR (Register #11). A setting applies to both Scroll Screens A and B.

HSCR	LSCR	FUNCTION
0	0	Overall Scrolling
0	1	Prohibited
1	0	Scroll in one cell unit
1	1	Scroll in one line unit

HSCR, LCSR: Register #11

The effective scroll quantity is equivalent to 10 bits (000H ~ 3FFFH).

Taking the Display Screen as standard, the scroll direction will be as follows:



Horizontally scrolling quantity setting area:

H Scroll Data Table is in VRAM. From the base address which was set by Register #13, set the scrolling quantity of Screens A and B alternately. Also, the scrolling quantity data setting position varies depending on the following mode (overall, 1 cell or 1 line).

Mode	Setting Position
Overall	Line 0
1 cell	Every 8th line starting from line 0
1 line	All lines

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

00	A - Scrolling Quantity of Screen A	Overall, Cell, Line
02	B - Scrolling Quantity of Screen B	Overall, Cell, Line
04	A - Scrolling Quantity of Screen A	Line
06	B - Scrolling Quantity of Screen B	Line
08	A - Scrolling Quantity of Screen A	Line
0A	B - Scrolling Quantity of Screen B	Line
1C	A - Scrolling Quantity of Screen A	Line
1E	B - Scrolling Quantity of Screen B	Line
20	A - Scrolling Quantity of Screen A	Cell, Line
22	B - Scrolling Quantity of Screen B	Cell, Line
3FC	A - Scrolling Quantity of Screen A	Line
3FE	B - Scrolling Quantity of Screen B	Line

D15~D10 can be freely utilized for program software.

3. VERTICAL SCROLLING

The Display Screen allows for scrolling overall or every 2 cells in a dot unit. The setting can be done by VSCR (Register #11). A setting applies to both Scroll Screens A and B.

VSCR	Function
0	Overall Scroll
1	2-Cell Unit Scroll

VSCR: Register #11

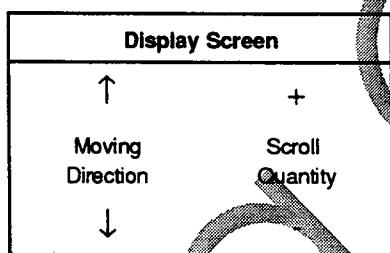
The scrolling quantity is equivalent to 11 bits (000H ~ 7FFH). However, it will be as shown below in the Interlace Mode.

Non-Interlace: The effective scrolling quantity is equivalent to 10 bits.

Interlace 1: Same as above.

Interlace 2: The effective scrolling quantity is equivalent to 11 bits.

Taking the Display Screen as standard, the scrolling direction will be as follows:



Set the V Scroll quantity by VSRAM.

Alternately set the scroll quantity of Screens A and B. Depending on the Scroll Mode, the Data setting positions differ.

Mode	Setting Position
Overall	Only at the beginning
2 cell	Set to all

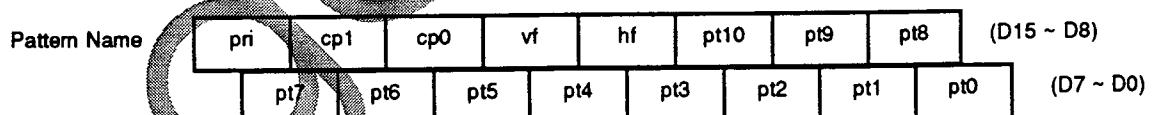
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

00	A - Scrolling Quantity of Screen A	0, 1 Cell, Overall
02	B - Scrolling Quantity of Screen B	0, 1 Cell, Overall
04	A - Scrolling Quantity of Screen A	2, 3 Cell
06	B - Scrolling Quantity of Screen B	2, 3 Cell
08	A - Scrolling Quantity of Screen A	4, 5 Cell
0A	B - Scrolling Quantity of Screen B	4, 5 Cell
0C	A - Scrolling Quantity of Screen A	6, 7 Cell
0E	B - Scrolling Quantity of Screen B	6, 7 Cell
4C	A - Scrolling Quantity of Screen A	38, 39 Cell
4E	B - Scrolling Quantity of Screen B	38, 39 Cell

D15~D11 is indefinite.

4. SCROLL PATTERN NAME

The Scroll Screen's name table is in VRAM and set by Registers #2 and #4. The Pattern Name requires 2 bytes (1 word) per cell the Scroll Screen. Depending on the Scroll Screen's size, VRAM and Scroll Screen correspond with each other differently. Refer to Scroll Screen Size.



pri: Refer to Priority

cp1: Color palette selection bit (see Color Palette)

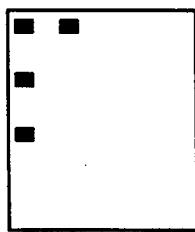
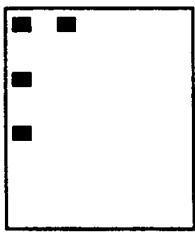
cp0: Color palette selection bit (see Color Palette)

vf: V Reverse Bit 1: Reverse

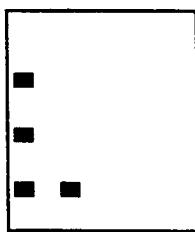
hf: H Reverse Bit 1: Reverse

pt10 ~ pt0: Pattern Generator Number

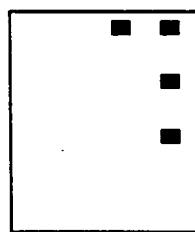
Reverse bits vf and hf allow for H and V reverse on cell unit basis.



vf = 0
hf = 0



vf = 1
hf = 0



vf = 0
hf = 1



vf = 1
hf = 1

segd

5. PATTERN GENERATOR

Pattern Generator has VRAM 0000H as base address, and a pattern is expressed on an 8 x 8 dot basis. To define a pattern, 32 bytes are required. Starting from 0000H, it proceeds in the sequence of Pattern Generator 0, 1, 2, etc . The relationship between the display pattern and memory is as follows:

1	2	3	4	5	6	7	8
a	□	□	□	□	□	□	□
b	□	□	□	□	□	□	□
c	□	□	□	□	□	□	□
d	□	□	□	□	□	□	□
e	□	□	□	□	□	□	□
f	□	□	□	□	□	□	□
g	□	□	□	□	□	□	□
h	□	□	□	□	□	□	□

	0	1	2	3				
	D7	D0	D7	D0	D7	D0	D7	D0
00	a1	a2	a3	a4	a5	a6	a7	a8
04	b1	b2	b3	b4	b5	b6	b7	b8
08	c1	c2	c3	c4	c5	c6	c7	c8
0C	d1	d2	d3	d4	d5	d6	d7	d8
10	e1	e2	e3	e4	e5	e6	e7	e8
14	f1	f2	f3	f4	f5	f6	f7	f8
18	g1	g2	g3	g4	g5	g6	g7	g8
1C	h1	h2	h3	h4	h5	h6	h7	h8

The display colors and memory relationship is as follows:

D7	D6	D5	D4	D3	D2	D1	D0
COL3	COL2	COL1	COL0	COL3	COL2	COL1	COL0

In Interlace Mode 2, one cell consists of 8 x 16 dots and therefore 64 bytes (16 long words) are required.

	1	2	3	4	5	6	7	8
a	□	□	□	□	□	□	□	□
b	□	□	□	□	□	□	□	□
c	□	□	□	□	□	□	□	□
d	□	□	□	□	□	□	□	□
e	□	□	□	□	□	□	□	□
f	□	□	□	□	□	□	□	□
g	□	□	□	□	□	□	□	□
h	□	□	□	□	□	□	□	□
i	□	□	□	□	□	□	□	□
j	□	□	□	□	□	□	□	□
k	□	□	□	□	□	□	□	□
l	□	□	□	□	□	□	□	□
m	□	□	□	□	□	□	□	□
n	□	□	□	□	□	□	□	□
o	□	□	□	□	□	□	□	□
p	□	□	□	□	□	□	□	□

	0	1	2	3	4	5	6	7
D7	D0	D7	D0	D7	D0	D7	D0	D0
00	a1	a2	a3	a4	a5	a6	a7	a8
04	b1	b2	b3	b4	b5	b6	b7	b8
08	c1	c2	c3	c4	c5	c6	c7	c8
0C	d1	d2	d3	d4	d5	d6	d7	d8
10	e1	e2	e3	e4	e5	e6	e7	e8
14	f1	f2	f3	f4	f5	f6	f7	f8
18	g1	g2	g3	g4	g5	g6	g7	g8
1C	h1	h2	h3	h4	h5	h6	h7	h8
20	i1	i2	i3	i4	i5	i6	i7	i8
24	j1	j2	j3	j4	j5	j6	j7	j8
28	k1	k2	k3	k4	k5	k6	k7	k8
2C	l1	l2	l3	l4	l5	l6	l7	l8
30	m1	m2	m3	m4	m5	m6	m7	m8
34	n1	n2	n3	n4	n5	n6	n7	n8
38	o1	o2	o3	o4	o5	o6	o7	o8
3C	p1	p2	p3	p4	p5	p6	p7	p8

J. WINDOW

For Window display, the following register setting and VRAM areas are required.

Window Pattern Name Table and Base Address

Register #3	0	0	WD15	WD14	WD13	WD12	WD11	0
-------------	---	---	------	------	------	------	------	---

Mode Set Register Number 4

Register #12	RS0	0	0	0	S/TE	LSM1	LSM0	RS1
--------------	-----	---	---	---	------	------	------	-----

Window H Position

Register #17	RIGT	0	0	WHP5	WHP4	WHP3	WHP2	WHP1
--------------	------	---	---	------	------	------	------	------

Window V Position

Register #18	DOWN	0	0	WVP4	WVP3	WVP2	WVP1	WVPO
--------------	------	---	---	------	------	------	------	------

VRAM: Window Pattern Name Table Maximum 4 Kbytes.

1. DISPLAY POSITION

The Window Display Position is designated by Registers #17 and #18. Screen display can be divided on a unit basis of H2 cells and V1 cell. The dividing position varies depending on resolution.

	0	1	2	3	4	5	34	35	36	37	38	39
0												
1												
2												
25												
26												
27												

H 40 cells / V 28 cells mode

Register #17

RIGT	0	0	WHP5	WHP4	WHP3	WHP2	WHP1
------	---	---	------	------	------	------	------

Register #18

DOWN	0	0	WVP4	WVP3	WVP2	WVP1	WVP0
------	---	---	------	------	------	------	------

RIGT: 0 Displays Window from the left to H dividing position
 1 Displays Window from the H dividing position to the right end
 DOWN: 0 Displays Window from the top end to the V dividing position.
 1 Displays Window from the V dividing position to the bottom end.

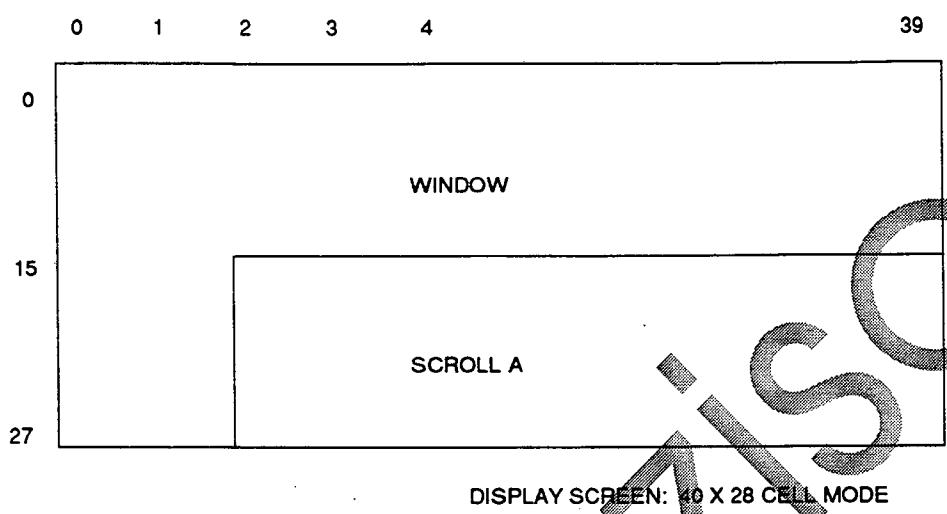
WHP5 ~ WHP1: H dividing position
 WVP4 ~ WVP0: V dividing position

H Resolution	Dividing Position (WHP)
32 cell	0 ~ 16 (0 ~ 32 cell)
40 cell	0 ~ 20 (0 ~ 40 cell)

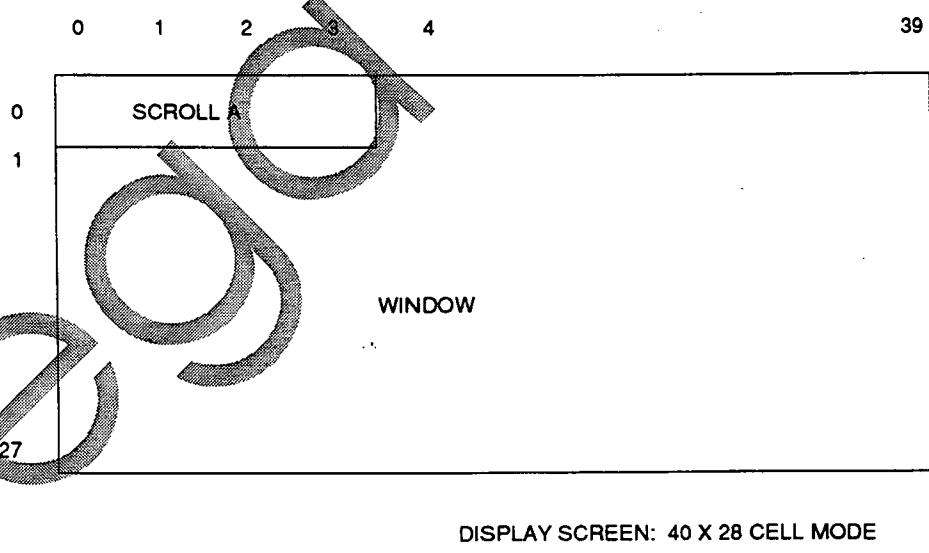
V Resolution	Dividing Position (WVP)
28 cell	0 ~ 28
30 cell	0 ~ 30

Setting Example

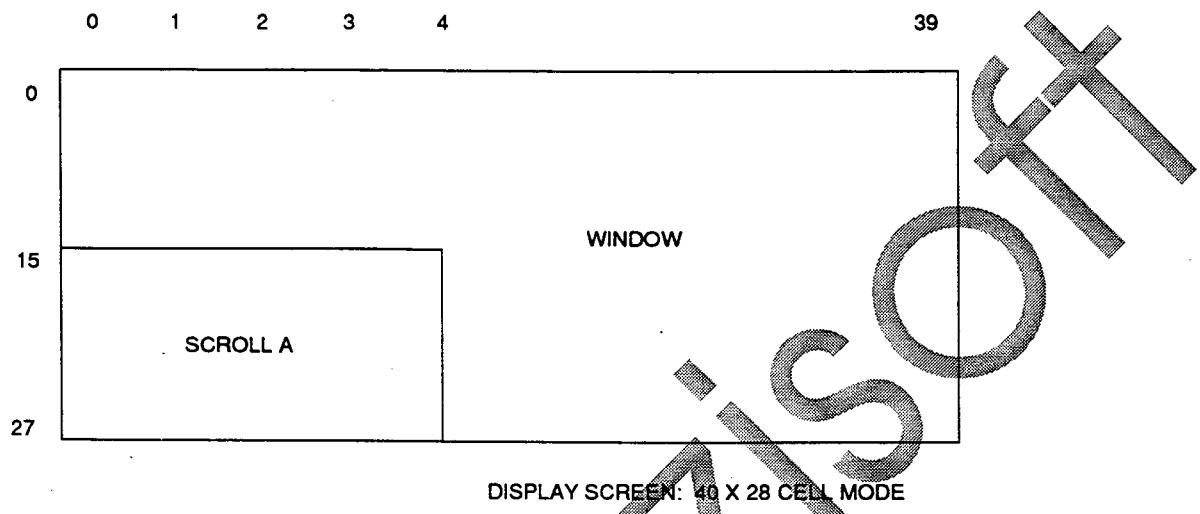
Register #17: 000H+01H Window from the left end to the second cell
Register #18: 000H+10H Window from the top end to the 16th cell



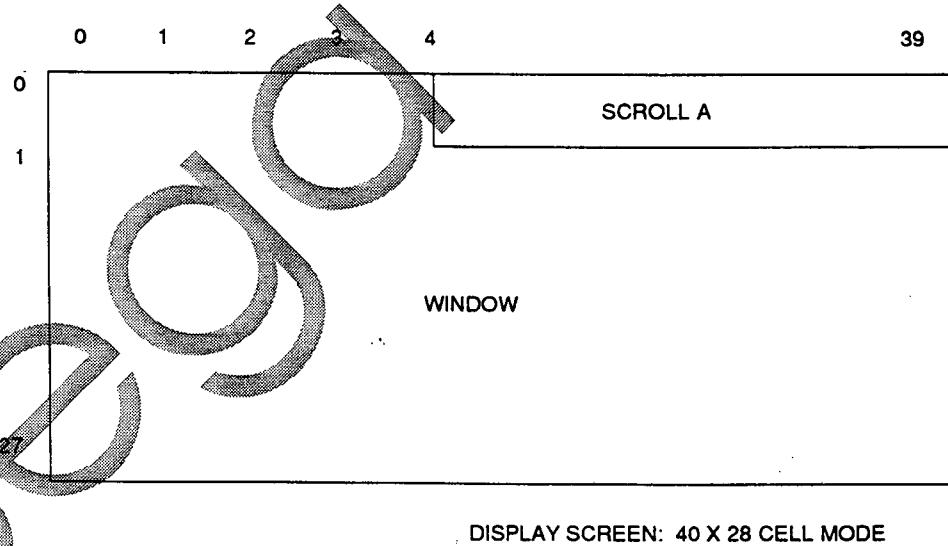
Register #17: 80H+02H Window from the left end 4th cell to the right end
Register #18: 80H+01H Window from the 2nd cell to the bottom end



Register #17: 80H+01H Window from the 4th cell to the right end
Register #18: 00H+10H Window from the top end to the 16th cell



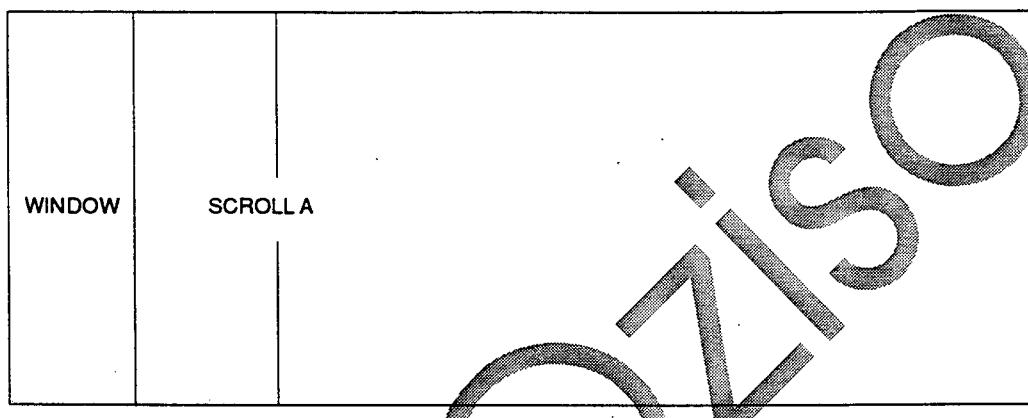
Register #17: 00H+02H Window to the 4th cell from the left
Register #18: 80H+01H Window from the 2nd cell to the bottom end



2. WINDOW PRIORITY

Window Priority is handled in the same way as in Scroll A. Scroll A is not displayed in the area where Window is displayed. Also, only when Window is set to the left and Scroll A is moved in H direction, the character corresponding to two cells on the right side of the boundary between Window and Scroll A will be disfigured.

There will be no malfunctioning when Window is set to the left side and Scroll A is moved only in the V direction; also when Window is set to the right side.



Display of this portion will be disfigured. Therefore, mask Scroll A by using high priority.

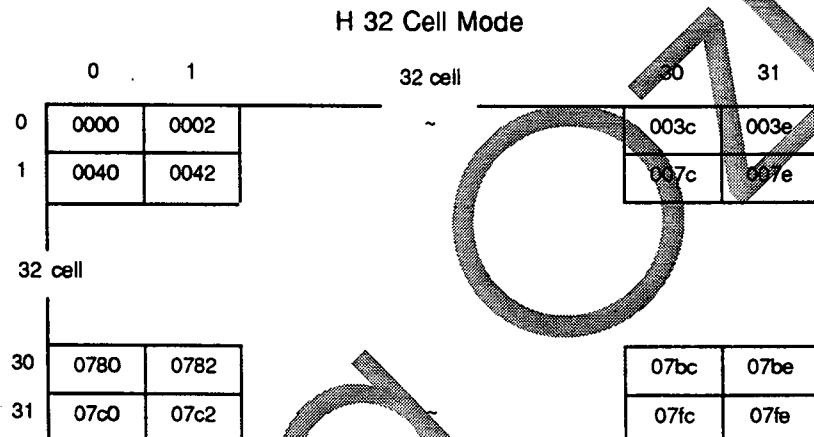
second

3. WINDOW PATTERN NAME

Window Pattern Name Table is on VRAM, and the base address is designated by Register #13. The Pattern Name, the same as in Scroll Screen, requires 2 bytes (1 word) per cell.

Pattern Name	pri	cp1	cp0	vf	hf	pt10	pt9	pt8	(D15 ~ D8)
	pt7	pt6	pt5	pt4	pt3	pt2	pt1	pt0	(D7 ~ D0)
pri:	Refer to Priority								
cp1:	Color palette selection bit								
cp0:	Color palette selection bit								
vf:	V reverse Bit 1: Reverse								
hf:	H Reverse Bit 1: Reverse								
pt10~pt0:	Pattern Generator Number								

Pattern Name and VRAM relation varies depending on H 32 cell/40 cell mode. Pay careful attention to this point.



H 40 Cell Mode

0	1		39	∇	40		62	63			
0	0000	0002	~			004e	0050	~		007c	007e
1	0080	0082				00dc	00e0			00fc	00fe
									32 cell		
30	0f00	0f02	~			0f4e	0f50	~		0f7c	0f7e
31	0fc0	0fc2	~			0fde	0fe0	~		0ffc	0ffe

40~63 are not displayed

Values shown are offset from the Base Address

In the H40 Cell Mode, there exists the area for H64 cells. However, there will be no display from the 41st cell in the H direction.

In the V28 Cell Mode, there will be no display from the V29th cell, and in the 30th Cell Mode, there will be no display from the 31st cell.

K. SPRITE

For Sprite Display, the following register setting and VRAM area are required.

Sprite Attribute Table and Base Address

Register #5

0	AT15	AT14	AT13	AT12	AT11	AT10	AT9
---	------	------	------	------	------	------	-----

Mode Setting Register #4

Register #12

RS0	0	0	0	S/TE	LSM1	LSM0	RS1
-----	---	---	---	------	------	------	-----

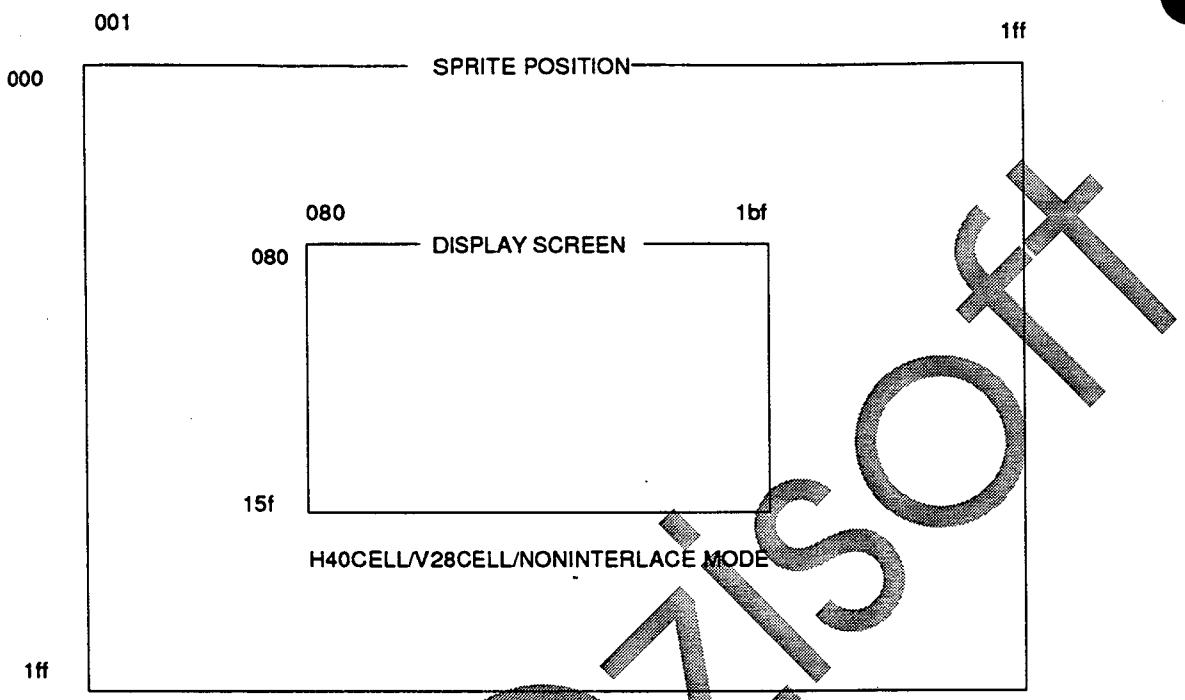
VRAM: Sprite Attribute Table Maximum 640 Bytes

1. DISPLAY POSITION

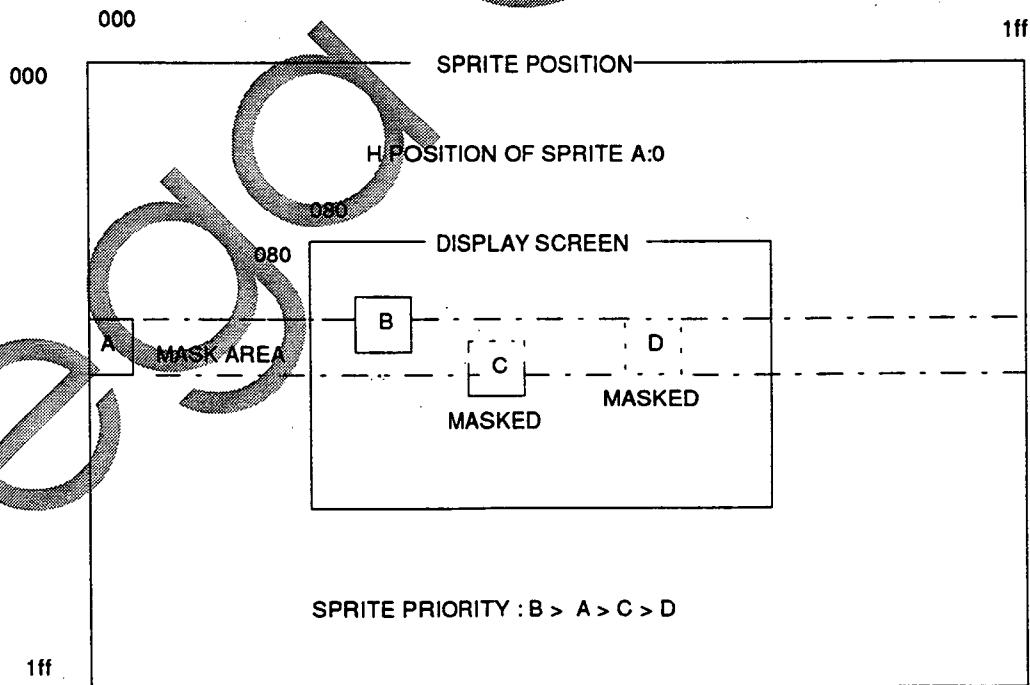
Sprite Position and Display Screen are as follows: When Sprite H position is 0, this is in a special mode. Therefore, pay careful attention to this point.

Resolution	H Position	Display Area
H32 Cell	001~1FFH	080 ~ 17FH
H40 Cell		080 ~ 1BFH

Resolution		V Position	Display Area
V28 Cell	Non-interlace	000 ~ 1FFH	080 ~ 15FH
	Interface 1		
	Interface 2	000 ~ 3FFH	100 ~ 2BFH
V30 Cell	Non-interlace	000 ~ 1FFH	080 ~ 16FH
	Interface 1		
	Interface 2	000 ~ 3FFH	100 ~ 2DFH



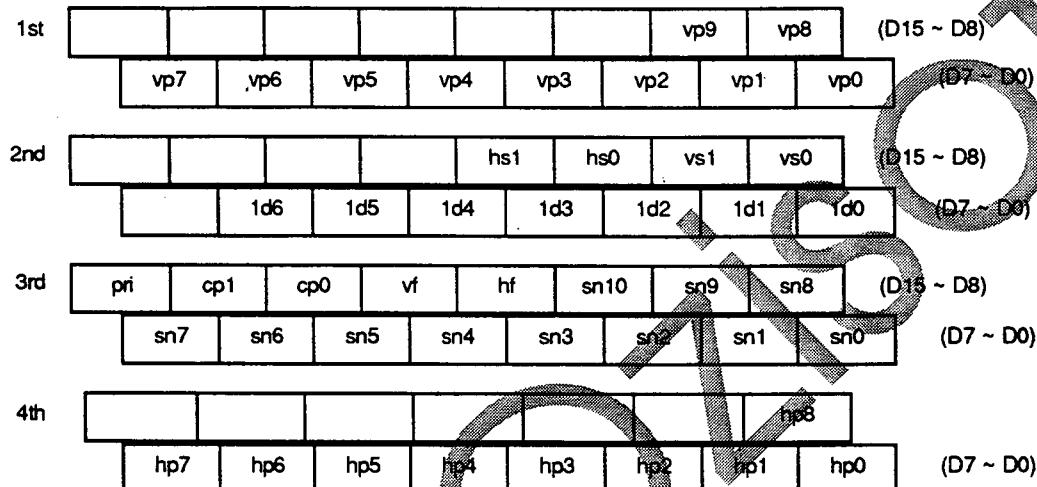
When 0 is set to the SPRITE H POSITION, a low priority sprite on the same line will not be displayed.



2. SPRITE ATTRIBUTE

Sprite Attribute Table is on VRAM and the Base Address is designated by Register #5. Attribute requires 8 bytes (4 words) per Sprite, and indicates Display Position, Priority, Sprite Generator Number, and Attribute.

Starting from the beginning of the Attribute Table, numbers are given in the sequence of Sprite 0, Sprite 1, Sprite 2, Sprite 3, etc. Priority between Sprites is not determined by the sequence of Sprite number, but by each Sprite's Link Data, and thus becomes programmable.



Blank portions can be utilized freely for software.

vp9 ~ vp0: V position

hp8 ~ hp0: H position

hs1, hs0: Sprite's H Size

vs1, vs0: Sprite's V Size

1d6 ~ 1d0: Link Data

pri: Priority Bit (see *Priority*)

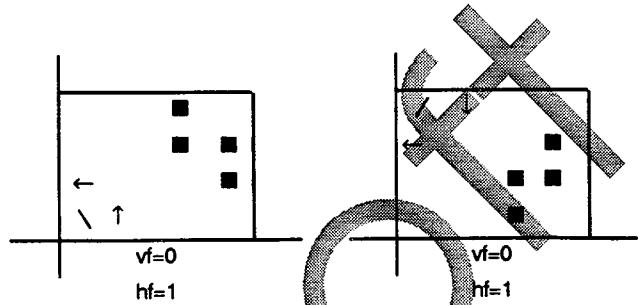
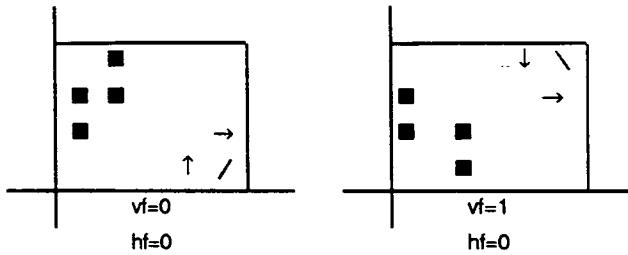
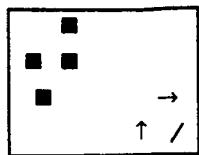
cp1, cp0: Color palette selection bit (see *Color Palette*)

vf: V Reverse Bit 1: Reverse

hf: H Reverse Bit 1: Reverse

sn10 ~ sn0: Sprite Pattern Generator Number

By using Reverse Bit vf, hf, V and H Reverse per Sprite is possible.



3. SPRITE SIZE

Per Sprite dot number can be set on a cell unit basis, by using vs1, vs0, hs1, and hs0.

V		
vs1	vs0	Number of cell
0	0	1 (8 dots)
0	1	2 (16 dots)
1	0	3 (24 dots)
1	1	4 (32 dots)

H		
hs1	hs0	Number of cell
0	0	1 (8 dots)
0	1	2 (16 dots)
1	0	3 (24 dots)
1	1	4 (32 dots)

However, in Interlace Mode 2, one cell is comprised of 8 x 16 dots, therefore, the number of V dots is two times (as compared to Interlace Mode 1).

4. SPRITE's DISPLAY CAPACITY

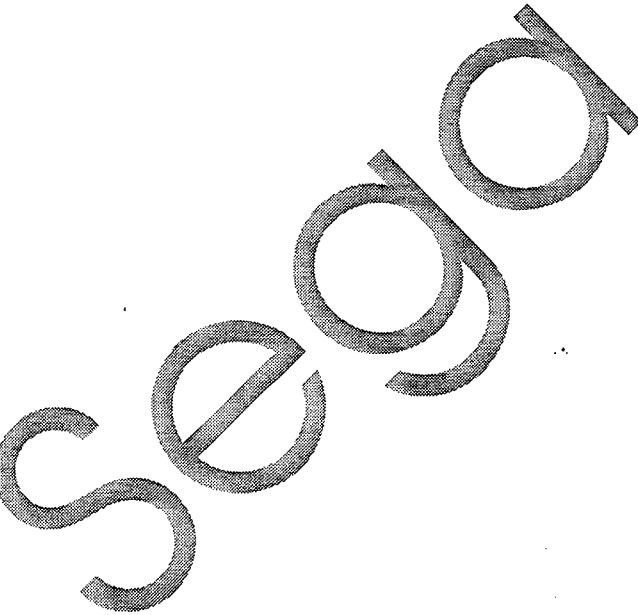
The number of Sprite's maximum display varies depending on H resolution setting.

Resolution	No. of Display	No. of Display per Line	Display Dot per Line
H32 cell	Max. 64 Sprites	Max. 16 Sprites	Max. 256 dots
H40 cell	Max. 80 Sprites	Max. 20 Sprites	Max. 320 dots

Sprite is displayed in the sequential order of Priority.

Example

- With H size 1 cell, when 30 Sprites are intended to be displayed on the same line, up to 16 Sprites counting from the one having highest priority (in the H32 cell mode) and 20 Sprites in the H40 cell mode can be displayed, due to the limitation of display per line.
- With H size 4 cells, when 16 Sprites are intended to be displayed on the same line, up to 8 Sprites, counting from the one having the highest priority (in the H32 cell mode) and 10 Sprites in the H40 cell mode can be displayed, due to the limitation of Display dots.
- With H size 3 cells, when 16 Sprites are intended to be displayed in the same line, 11 Sprites, counting from the one having the highest priority (as for 11th one, however, only for 16 dots from the left end) in the H40 cell mode can be displayed, due to the limitation of the display dots.



5. PRIORITY BETWEEN SPRITES

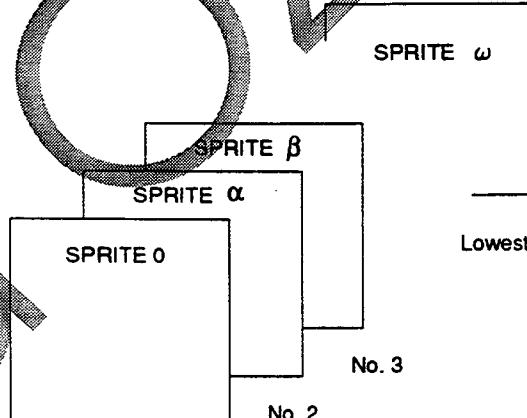
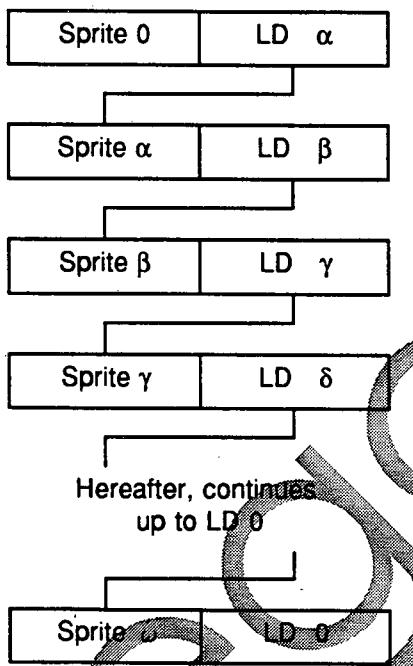
Priority between Sprites is designated by each Sprite's Link Data.

With Sprite 0 being Priority No. 1, the Sprite No. 2 written in the Link Data there will be Priority No. 2. Priority No. 2 Sprite's Link Data shows Priority No. 3 Sprite. Priority No. 3 Sprite's Link Data shows Priority No. 4 Sprite.

In this way, Priority is sequentially designated by each Sprite's Link Data and thus it is in List form. The value that can be set in the Link Data is 0~ (number of maximum Display on one screen minus one). Be sure to set 0 to the lowest priority Sprite's Link Data.

When 0 is given to the Sprite Link Data, List ends at that Sprite and the Priority will become the lowest. Even in the case that the number of Sprites linked to List is less than the maximum display quantity (64 or 80), the remaining Sprites not linked to Sprite will not be displayed.

When value other than those specified is set to Link Data, or 0 is not set to the lowest Priority Sprite Link Data, ordinary functioning is not guaranteed.



Setting Example

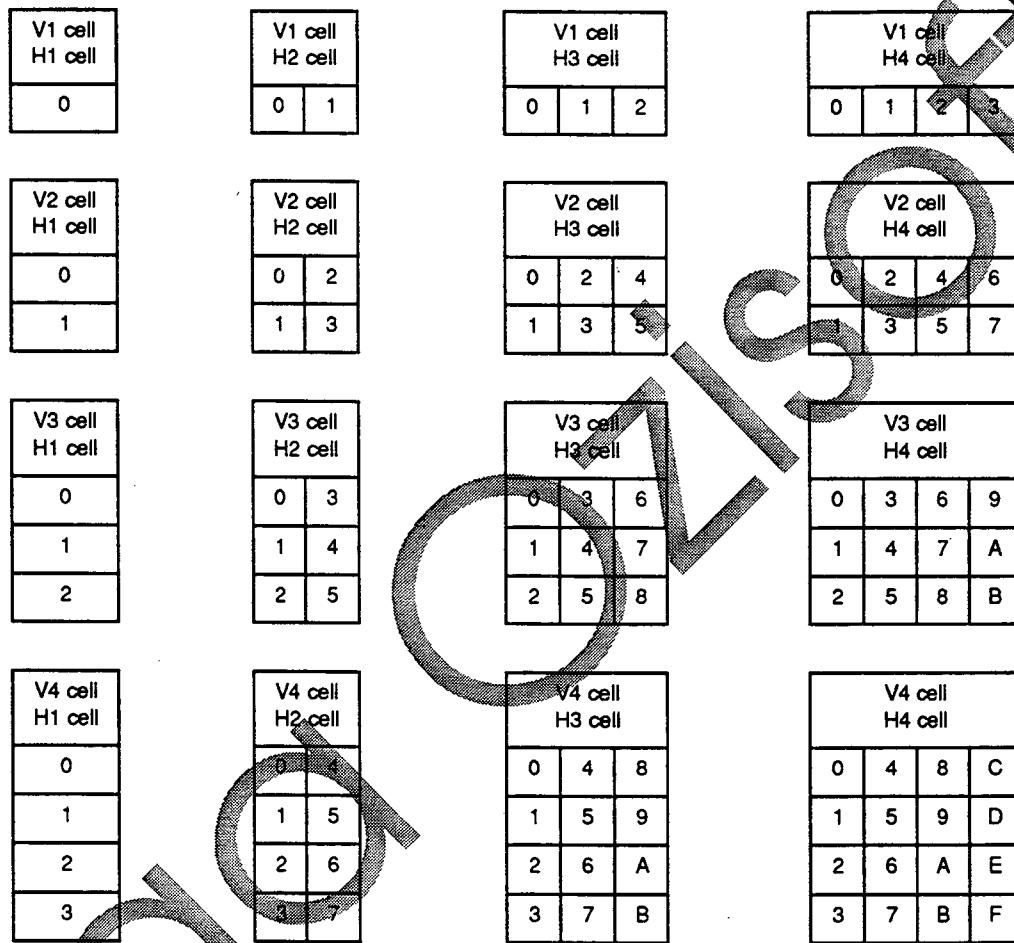
	Link Data
Sprite 0	2
Sprite 1	10
Sprite 2	1
Sprite 3	4
Sprite 4	5
Sprite 5	15
Sprite 6	—
Sprite 7	0
Sprite 8	—
Sprite 9	—
Sprite 10	11
Sprite 11	13
Sprite 12	—
Sprite 13	3
Sprite 14	—
Sprite 15	7
Sprite 16	—

Sprite 0
Sprite 2
Sprite 1
Sprite 10
Sprite 11
Sprite 13
Sprite 3
Sprite 4
Sprite 5
Sprite 15
Sprite 7

The 11 Sprites shown in the Display Priority are displayed on the screen. Sprites No. 6, 8, 9, 12, 14, and 16 onward are not displayed because they are not linked with Link Data List.

6. SPRITE PATTERN GENERATOR

The Sprite Pattern Generator with VRAM 0000H as Base Address expresses one pattern on a basis of 8x8 dots. 32 bytes are required to define one pattern. Every 32 bytes, one pattern is expressed in the sequence of Pattern Generator 0, 1, 2, etc. The relationship of Display Pattern and Memory is the same as in Pattern Generator. Also, Sprite Size and Pattern Generator relationship is as follows:



L. PRIORITY

Priority between Sprite, Scroll A, and Scroll B can be designated.

Priority can be designated by each Pattern Name and Attribute Priority bit. It will be set for the Scroll Screen on a cell unit basis and for each Sprite. By combining each priority bit, Priority will be as follows, however, the Background Priority is always the lowest.

S pri	A pri	B pri	Priority
0	0	0	S>A>B>G
1	0	0	S>A>B>G
0	1	0	A>S>B>G
1	1	0	S>A>B>G
0	0	1	B>S>A>G
1	0	1	S>B>A>G
0	1	1	A>B>S>G
1	1	1	S>A>B>G

S: Sprite

A: Scroll A

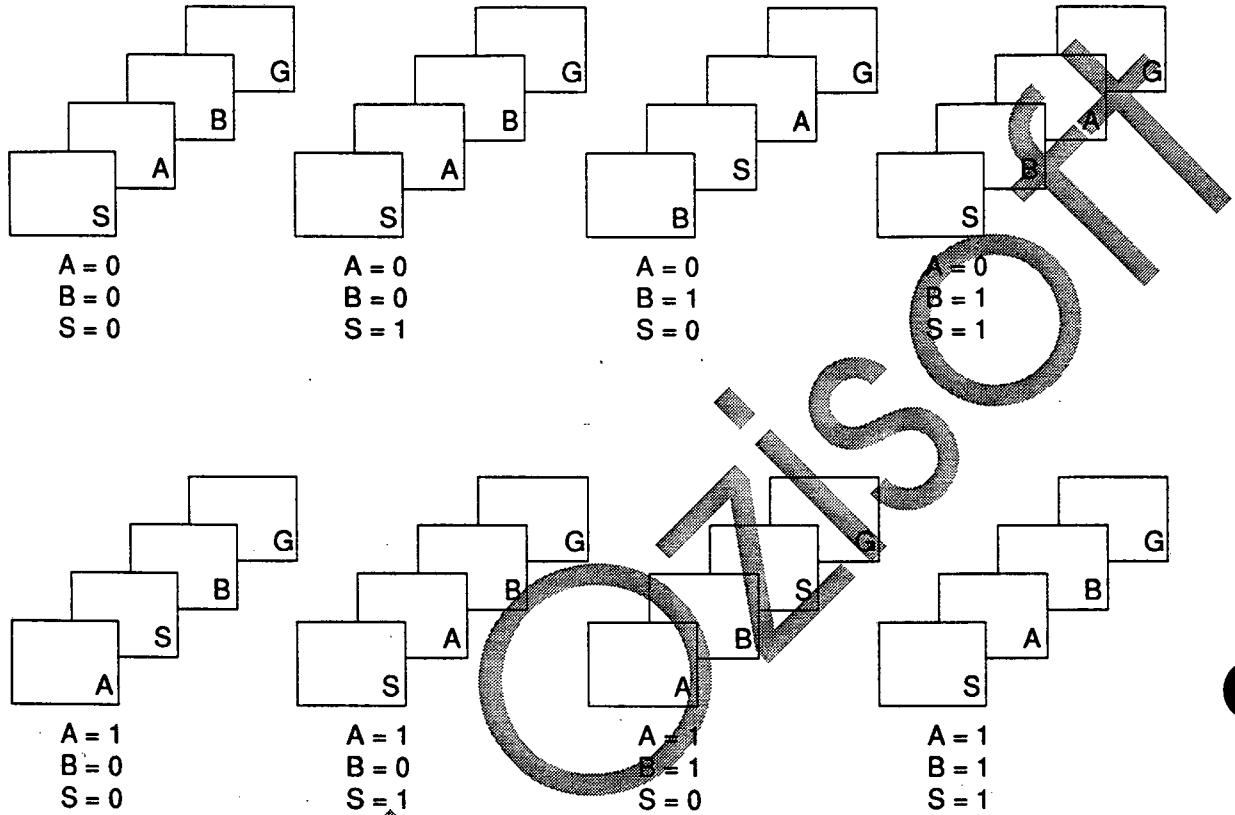
B: Scroll B

G: Background

Also, by combining S/Ten (Register #12) and the above priority, Shadow-Highlight effect function can be utilized.

SEGA

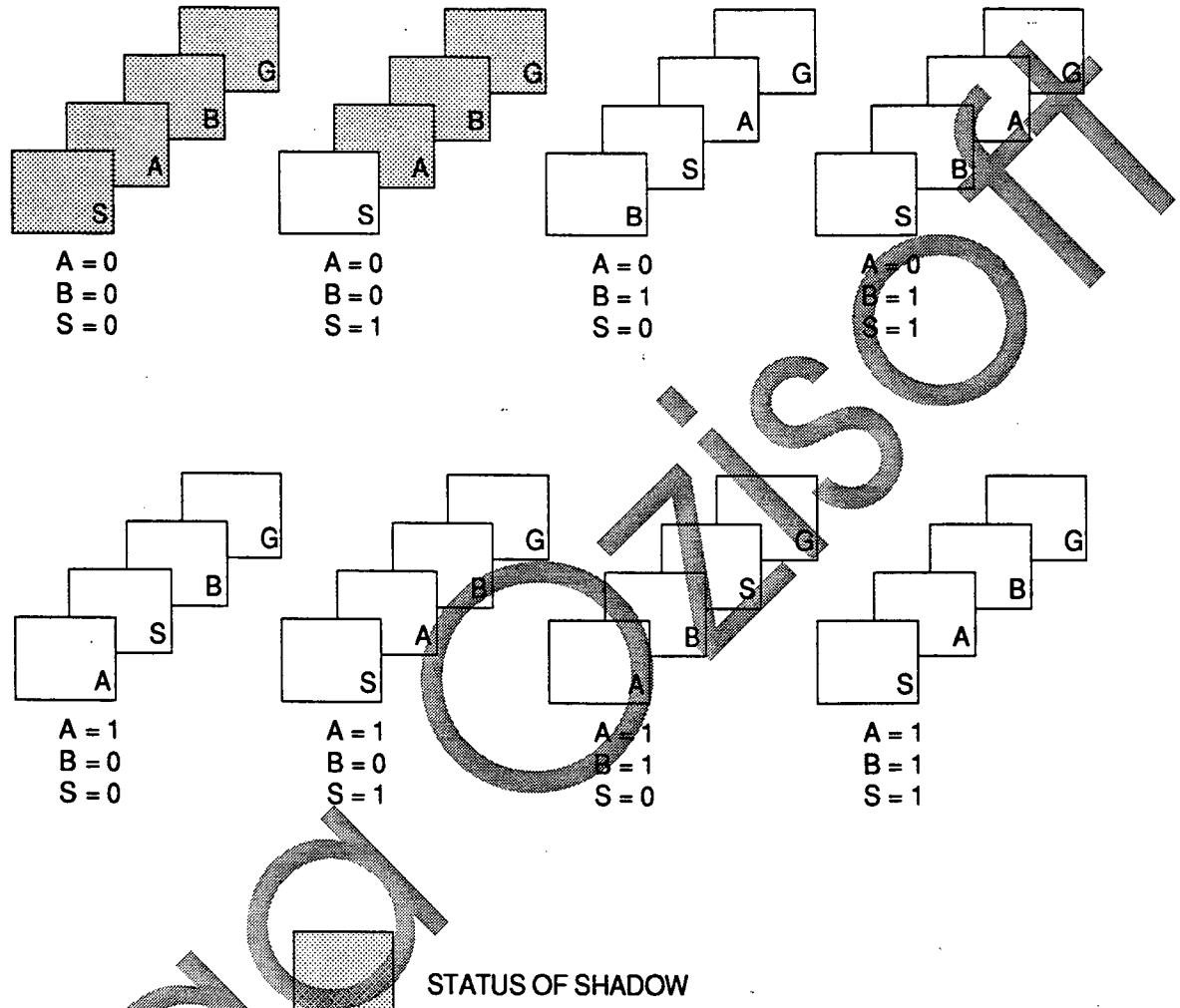
S/TEN = 0



The above shows Priority situation of Sprite, Scroll A, Scroll B and Background. The dot to which Color Code 0 is designated is transparent. Therefore, either one of Scroll Screen A, Scroll Screen B, or Background (the priority of which is one step lower than the transparent one) will appear.

S/TEN = 1

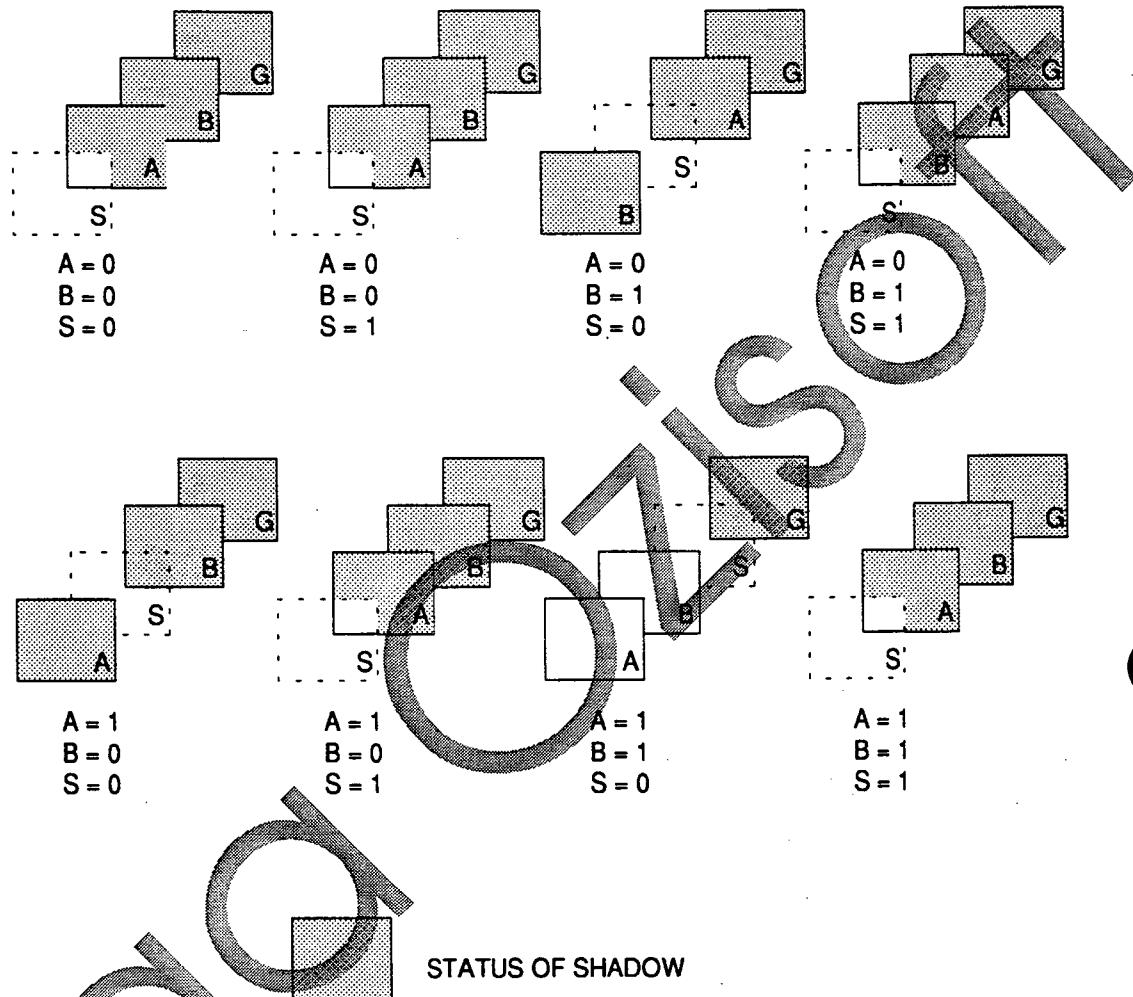
Sprite Color Palette 0 ~ 3 , Color Code 0 ~ 15
Color Palette 3 , Color Code 0 ~ 13



Where S/TEN = 1 when the Priority bit of both Scroll A and Scroll B is 0, there will be Shadow. For the color status, refer to the color palette.

S/TEN = 1

Sprite Color Palette 3 , Color Code 15

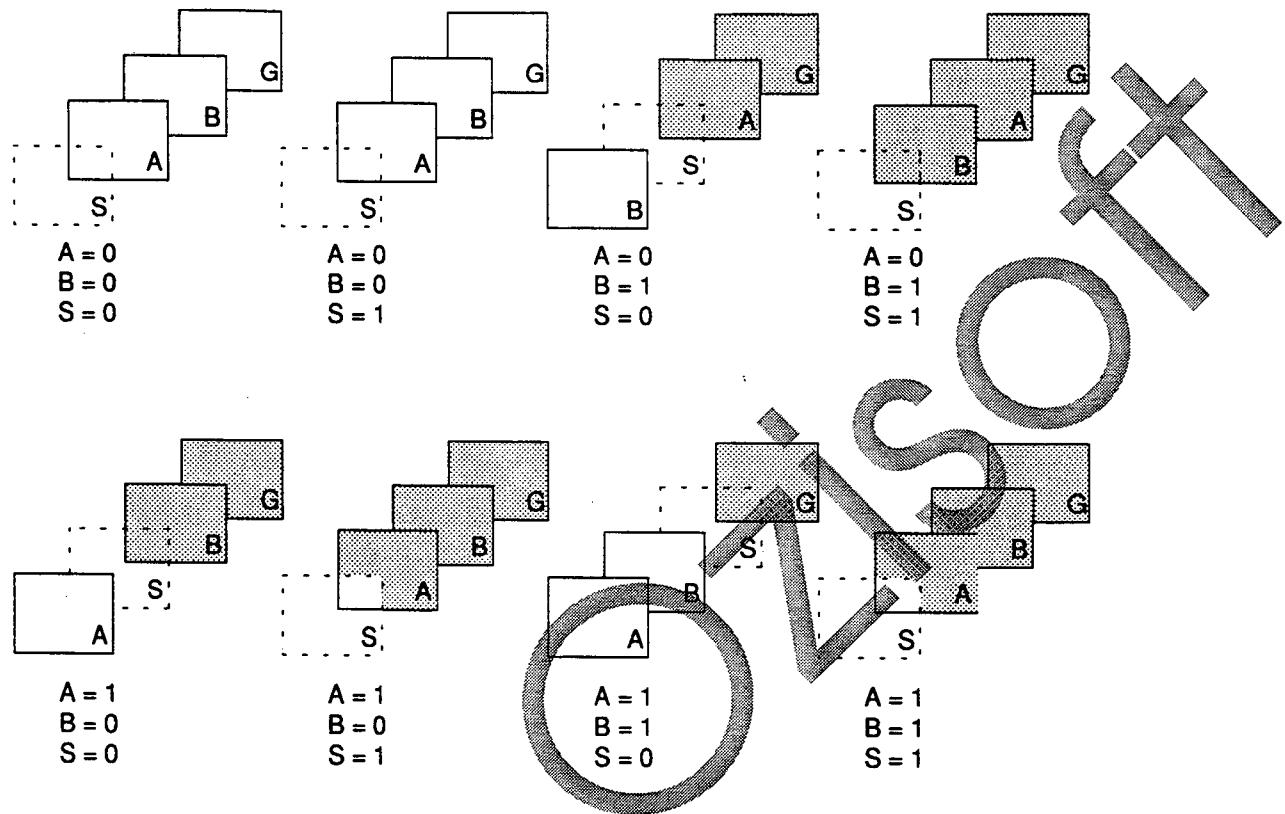


The dots for the Sprite Color Code 15 work as a Shadow operator on the screen, the Priority of which is lower than the Sprite.

Since Sprite dot works as an operator, this will not be displayed.

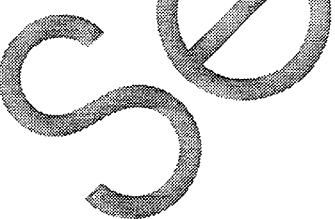
S/TEN = 1

Sprite Color Palette 3 , Color Code 14



The dots of Sprite Color Code 15 work as an operator on the screen, the priority of which is lower than Sprite.

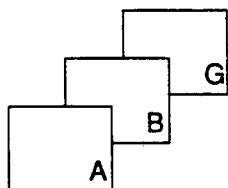
Since Sprite dots work as an operator, this will not be displayed.



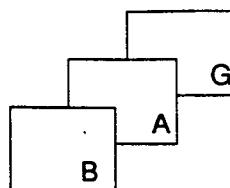
STATUS OF HIGHLIGHT

When SPRITE is not related to PRIORITY, the following PRIORITY applies:

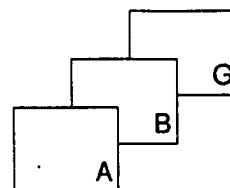
S/TEN = 0



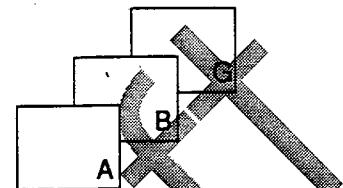
A = 0
B = 0



A = 0
B = 1

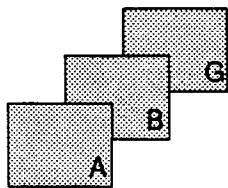


A = 1
B = 0

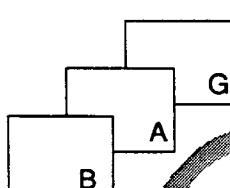


A = 1
B = 1

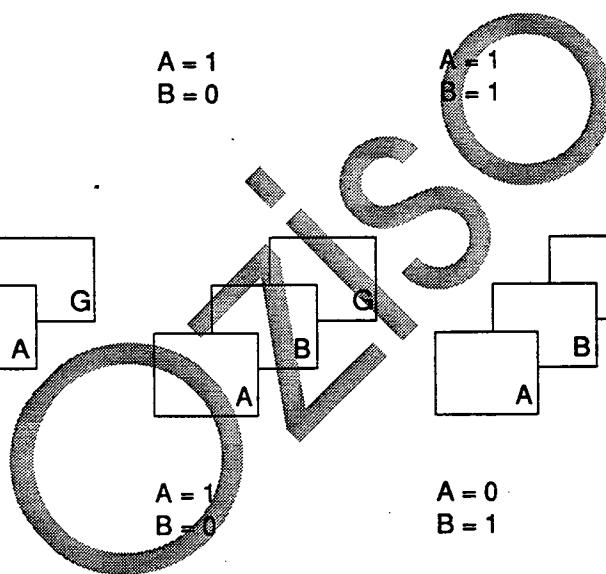
S/TEN = 1



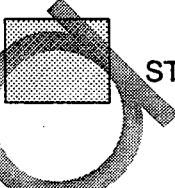
A = 0
B = 0



A = 0
B = 1



A = 0
B = 1



STATUS OF SHADOW

M. COLOR PALETTE

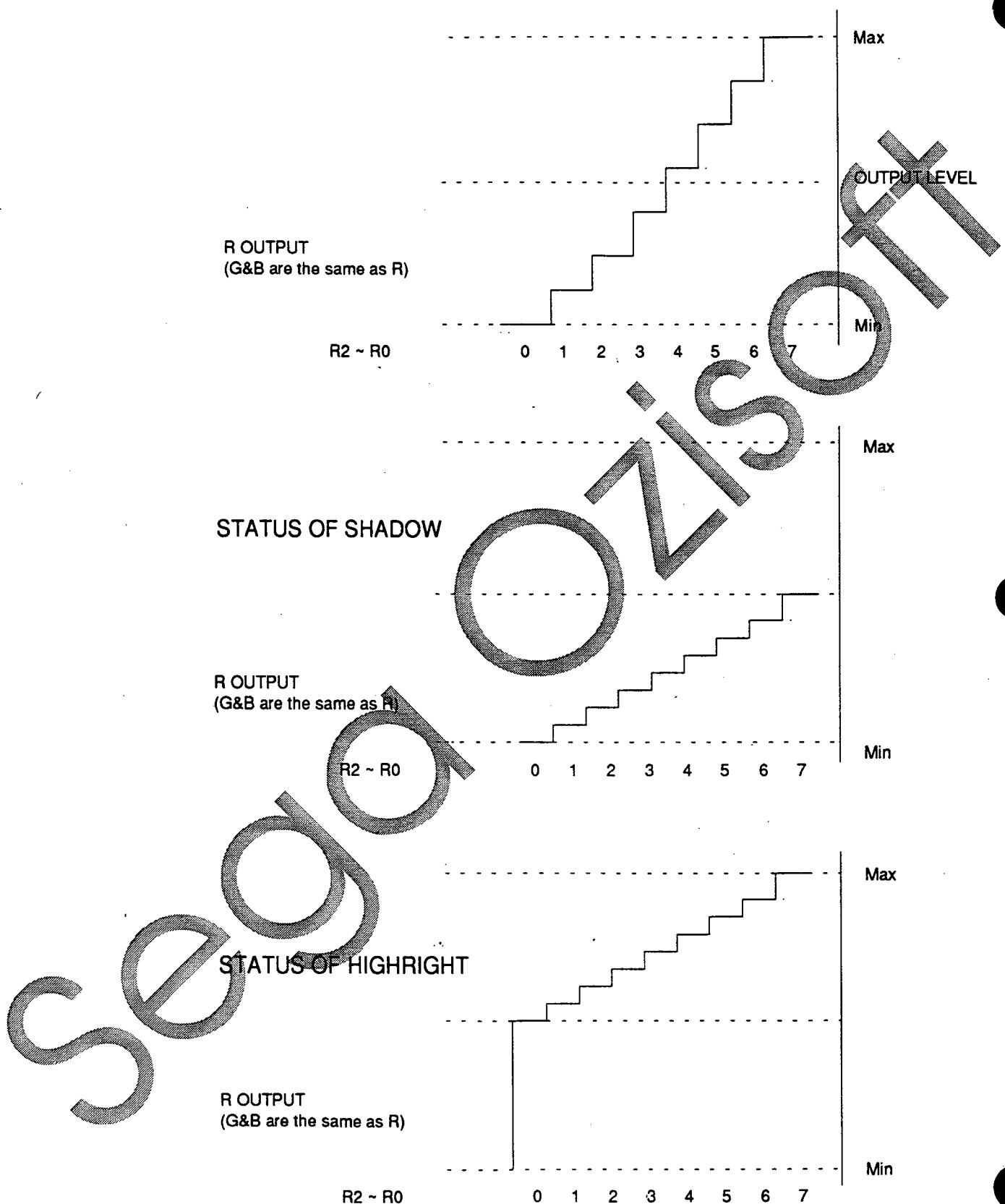
One dot is comprised of 4 bits and can designate the 0-15 colors. Also, 0-3 color palette can be designated by Scroll screen on a cell basis and by each Sprite. CRAM data are as follows. Since each of R, G and B has 3 bits, colors can be freely selected out of 512 colors.

DATA	0	0	0	0	B2	B1	B0	0	(D15~D8)
	G2	G1	G0	0	R2	R1	R0	0	(D7~D0)

The relationships between CRAM address, palette, and color code are as follows. However, in the case of each palette's color code 0, the color for the Scroll A, Scroll B Window, and Sprite is See Through irrespective of RGB designation.

ADDRESS	BLUE	GREEN	RED	PALETTE	CODE	REMARKS
00H					0	The 0-15 colors designated by RGB will be displayed
02H					2	
04H					:	
:					13	
1AH					4	
1CH					5	
1EH						
20H					0	
:					15	
3EH						Same as Palette 0
40H				2	0	
:					15	
5EH						Same as Palette 0
60H					0	
62H					1	
:					13	
7AH					14	
7CH					15	
7EH						For 14 and 15, refer to Priority

RGB bit and display are as follows:



N. INTERLACE MODE

Raster Scan Mode can be changed by setting LSM0 and LSM1 (RGB #12).

LSM1	LSM0	Raster Scan Mode
0	0	Non-interlace mode
0	1	In the non-interlace mode, the same pattern is displayed on the rasters of even and odd numbered files. (Interlace 1)
1	1	In the interlace mode, the different pattern is displayed on the rasters of even and odd numbered files. (Interlace 2)

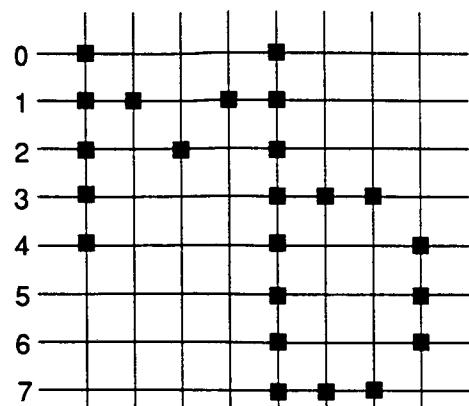
In the interlace mode and Interlace 1, one cell is defined by 8x8 dots and in Interlace 2, 8x16 dots. For Display, one cell consists of 8x8 dots in the non-interlace mode, and in the interlace mode, 8x16 dots.

In any case, number of cells in one screen are the same.

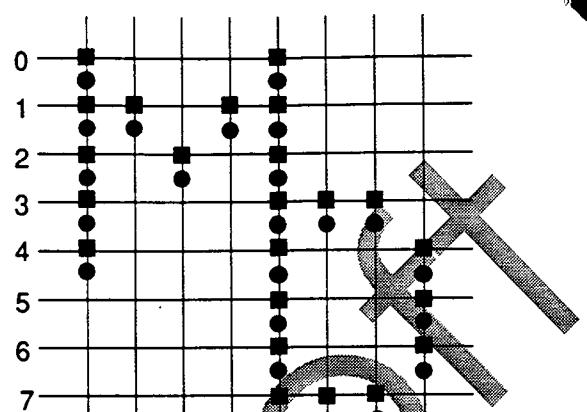
Depending on the type of display, in the case of interlace display, there may occur a serious blur in the vertical direction. Therefore, when using the display, pay careful attention in this regard.

sega

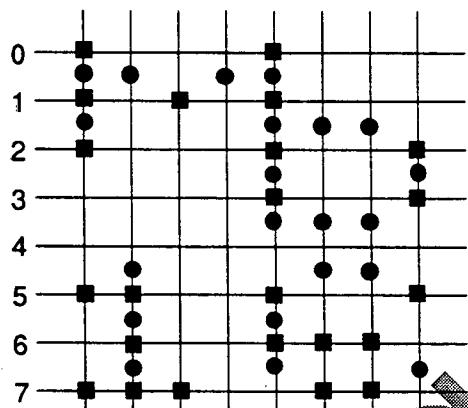
NON-INTERLACE



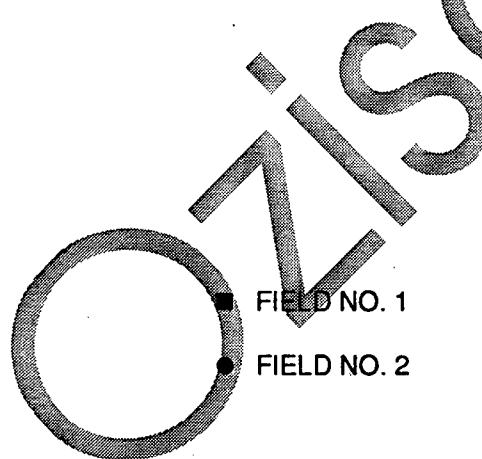
INTERLACE 1



INTERLACE 2



sega



III. BACKWARD COMPATIBILITY MODE

In the case of Backward Compatibility Mode, the Mega Drive differs from the original Mark III and Master System in the following points:

A. MARK III (MS-Japan)

OS-ROM is not incorporated

ROM cartridge/card selections are made by hardware in the same manner as in the case of Mark III. Start-up slot number is not written in 0C000H. Start-up SEGA logo is not displayed.

FM sound source is not incorporated

FM sound is incorporated in MS-Japan (standard) and Mark III (optional) (OPLL), however, Mega Drive has no option for that, although connection is possible.

Consider the Mega Drive's Japanese specifications as that of Mark III with MS-Japan's Joystick Port, or as MS-Japan without FM sound source and OS-ROM.

B. MASTER SYSTEM

OS-ROM is not incorporated

0C000H-0DFFFH RAM is not clear on power-up. RAM 0C000 has no meaningful value. Start-up Sega logo is not displayed.

FM sound source is not incorporated

FM sound source is incorporated in MS by option (OPLL). However, Mega Drive has no option, although connection is possible. Please regard the Mega Drive overseas version as a Master System without an Operating System ROM.

C. RAM BOARD

In the Mega Drive's Mark III and Master System backward compatibility mode, the RAM board for development (for which D-RAM was used) cannot be used due to the problem of Refresh. The other boards for development (which utilize S-RAM) can be used without any problem.

IV. SYSTEM I/O

Mega Drive System I/O area assignment starts from \$A00000, with the Z80 sub-CPU's memory area.

A. VERSION NUMBER

Indicates the Mega Drive's hardware version.

\$A10001

MODE	VMOD	DISK	RSV	VER3	VER2	VER1	VER0
------	------	------	-----	------	------	------	------

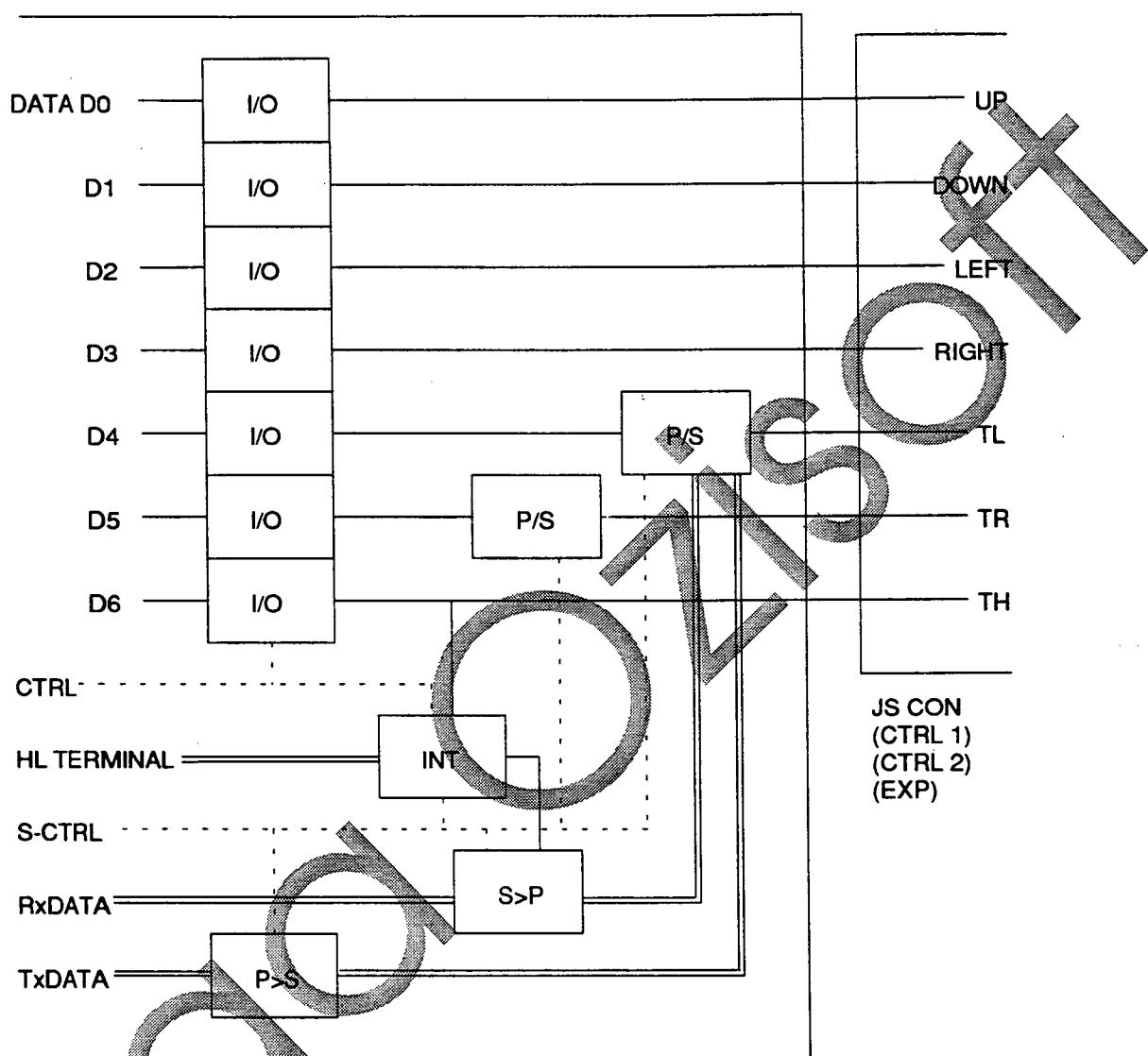
Mode (R) 0: Domestic Model
1: Overseas Model
VMOD (R) 0: NTSC CPU clock 7.67 MHz
1: PAL CPU clock 7.60 MHz
Disk (R) 0: FDD unit connected
1: FDD unit not connected
RSV (R) Currently not used
VER 3~0 (R) Mega Drive version is indicated by \$0-\$F.
The present hardware version is indicated by \$0.

B. I/O PORT

The Mega Drive has the three general purpose I/O ports: Ctrl 1, Ctrl 2, and Exp. Although each port differs from the others in physical shape, it functions in the same manner. Each port has the following five registers for control:

Data	(Parallel data)	:	R/W
Ctrl	(Parallel control)	:	R/W
S-Ctrl	(Serial control)	:	R/W
TxDATA	(Txd data)	:	R/W
RxDATA	(Rxd data)	:	R

The relationship between REGISTERs is as follows:



I/O : I/O change
P/S : PARALLEL/SERIAL MODE change
INT : INTERRUPT CONTROL
S>P : SERIAL-PARALLEL CONVERSION
P>S : PARALLEL-SERIAL CONVERSION

Mapping is as follows:

\$A10003	:	Data 1	(Ctrl 1)
\$A10005	:	Data 2	(Ctrl 2)
\$A10007	:	Data 3	(Exp)
\$A10009	:	Ctrl 1	
\$A1000B	:	Ctrl 2	
\$A1000D	:	Ctrl 3	
\$A1000F	:	TxData 1	
\$A10011	:	RxDATA 1	
\$A10013	:	S-Ctrl 1	
\$A10015	:	TxData 2	
\$A10017	:	RxDATA 2	
\$A10019	:	S-Ctrl 2	
\$A1001B	:	TxData 3	
\$A1001D	:	RxDATA 3	
\$A1001F	:	S-Ctrl 3	

Both Byte and Word access are possible. However, in the case of Word access, only the lower byte is meaningful.

Data shows the status of each port. The I/O direction of each bit is set by Ctrl and S-Ctrl.

DATA	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
	PD7 (RW)				PD3 (RW)	RIGHT		
	PD6 (RW)	TH			PD2 (RW)	LEFT		
	PD5 (RW)	TR			PD1 (RW)	DOWN		
	PD4 (RW)	TL			PD0 (RW)	UP		

Ctrl designates the I/O direction of each port and the Interrupt Control of TH.

CTRL	INT	PC6	PC5	PC4	PC3	PC2	PC1	PC0
------	-----	-----	-----	-----	-----	-----	-----	-----

INT	(RW)	0: TH-INT Prohibited 1: TH-INT Allowed
PC6	(RW)	0: PD6 Input Mode 1: PD6 Output Mode
PC5	(RW)	0: PD5 Input Mode 1: PD5 Output Mode
PC4	(RW)	0: PD4 Input Mode 1: PD4 Output Mode
PC3	(RW)	0: PD3 Input Mode 1: PD3 Output Mode
PC2	(RW)	0: PD2 Input Mode 1: PD2 Output Mode
PC1	(RW)	0: PD1 Input Mode 1: PD1 Output Mode
PC0	(RW)	0: PD0 Input Mode 1: PD0 Output Mode

S-Ctrl is for the status, etc. of each port's mode change, baud rate, and serial.

Ctrl designates the I/O direction of each port and the Interrupt Control of TH.

S-CTRL	BPS1	BPS0	SIN	SOUT	RINT	RERR	RRDY	TFUL
--------	------	------	-----	------	------	------	------	------

SIN	(RW)	0: TR - Parallel Mode 1: TR - Serial In
SOUT	(RW)	0: TL - Parallel Mode 1: TL - Serial Out
RINT	(RW)	0: Rxd Ready - Interrupt Prohibited 1: Rxd Ready - Interrupt Allowed
RERR	(R)	0: 1: RxdError
RRDY	(R)	0: 1: RxdReady
TFUL	(R)	0: 1: TxdFull

BPS1	BPS0	bps
0	0	4800
0	1	2400
1	0	1200
1	1	300

C. MEMORY MODE

The Mega Drive is able to generate internally the Refresh signal for the D-RAM development cartridge. When using the development cartridge, set to D-RAM mode. In the case of a production cartridge, set to ROM mode.

Only D8 of address \$A11000 is effective and for Write only.

\$A11000 D8 (W) 0: ROM mode
1: D-RAM mode

Access to \$A11000 can be based on byte.

D. Z80 CONTROL

1. Z80 BusReq

When accessing the Z80 memory from the 68000, first stop the Z80 by using BusReq. At the time of power on reset, the 68000 has access to the Z80 bus.

\$A11100 D8 (W) 0: BusReq cancel
1: BusReq request
(R) 0: CPU function stop accessible
1: Functioning

Access to Z80 area in the following manner.

- a. Write \$0100 in \$A11100 by using a Word access.
- b. Check to see that D8 of \$A11100 becomes 0.
- c. Access to Z80 area.
- d. Write \$0000 in \$A11100 by using a Word access.

Access to \$A11100 can also be based on byte.

2. Z80 Reset

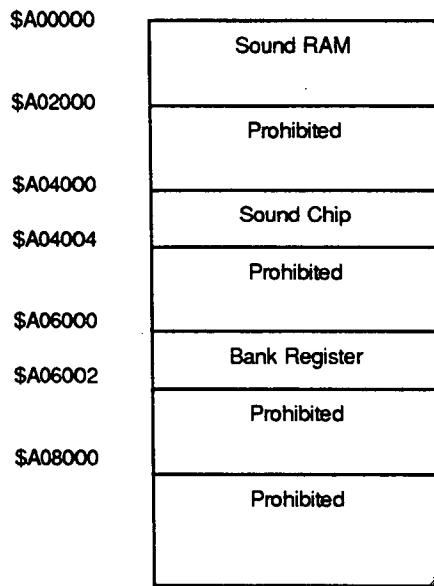
The 68000 may also reset the Z80. The Z80 is automatically reset during the Mega Drive hardware's power on reset sequence.

\$A11200 D8 (W) 0: Reset request
1: Reset cancel

Access to \$A11100 can also be based on byte.

E. Z80 AREA

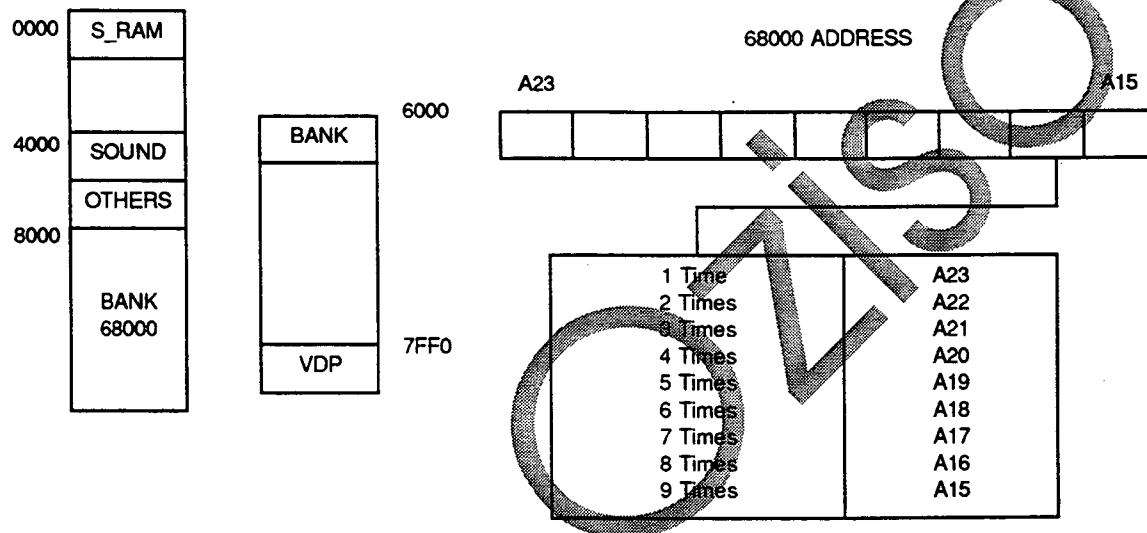
Mapping is performed starting from \$A00000 for Z80, a sub-CPU. As viewed from 68000, the memory map will be as follows:



1. **Sound RAM**
This is for the Z80 program. Access from 68000 by byte.
2. **Sound Chip**
This is the mapping area for FM sound source (YM2612). When accessing from 68000, use byte, due to timing problem.
3. **Bank Register**
Access to the 68000 side Memory Area from Z80 will be based on a 32 K byte unit. At this time, this register sets which bank is to be accessed. Registering from 68000 can be set, however, do not access to Z80 Bank Memory Area by 68000.

Setting Method

When accessing to the 68000 side addresses from Z80 side, all the addresses can be classified into Banks. Bank can be set by writing 9 times in 0 bit of 6000 (Z80 address). The 9 bits correspond to 68000 address 15-23 as shown below:



V. VRAM MAPPING

In VRAM, there are various tables and pattern generators as stated below. Among those, the base address of Pattern Generator Table and Sprite Generator Table are 0000H and fixed. However, the other base addresses can be freely assigned in VRAM by setting VDP Register. Also, Area can be overlapped, therefore, Table can be commonly used by Scroll screen and Window, for example.

- Scroll A Pattern Name Table, maximum 8 K byte
Base address designated by Register #2.
- Scroll B Pattern Name Table, maximum 8 K byte
Base address designated by Register #4.
- Window Pattern Name Table, varies by H resolution
Base address designated by Register #3.
- H Scroll Data Table, 1 K byte
Base address designated by Register #13.
- Sprite Attribute Table, varies by H resolution
Base address designated by Register #5.
- Pattern Generator Table
Base address is 0000H (fixed).
- Sprite Generator Table
Base address is 0000H (fixed).

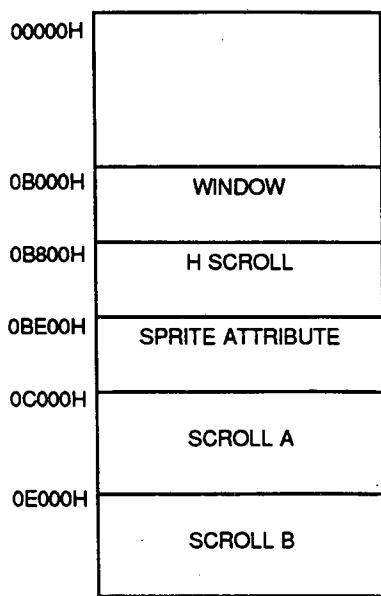
There are 1 K bytes for H Scroll Table, however, as for display 896 bytes in V28 Cell Mode and 960 bytes in V30 Cell Mode. There are 2 K bytes for Window Pattern Name Table in H32 Cell Mode, and 4 K byte area in H40 Cell Mode. For details, refer to Window. There are 512 bytes for Sprite Attribute Table in H32 Cell and 1 K byte area in H40 Cell Mode. However, as for display, there are 640 bytes in H40 Cell Mode.

Setting examples

H32 Cell Mode

- Scroll A Pattern Name Table
8 K bytes from 0C000H: Register #2 = \$30
- Scroll B Pattern Name Table
8 K bytes from 0E000H: Register #4 = \$07
- Window Pattern Name Table
2 K bytes from 0B000H: Register #3 = \$2C
- H Scroll Data Table
1 K byte from 0B800H: Register #13 = \$2E
- Sprite Attribute Table
512 bytes from 0BE00H: Register #5 = \$5F

Unoccupied area is used as Pattern Generator and Sprite Generator.



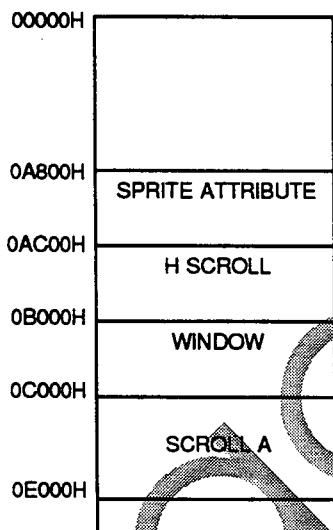
seg a

zisoft

H40 Cell Mode

- Scroll A Pattern Name Table
8 K bytes from 0C000H: Register #2 = \$30
- Scroll B Pattern Name Table
8 K bytes from 0E000H: Register #4 = \$07
- Window Pattern Name Table
4 K bytes from 0B000H: Register #3 = \$2C
- H Scroll Data Table
2 K bytes from 0AC00H: Register #13 = \$2B
- Sprite Attribute Table
1 K byte from 0A800H: Register #5 = \$54

Unoccupied area is used as Pattern Generator and Sprite Generator.



Precautions for M5 Software Programming

When programming the M5 software, pay attention to the following:

1. The program od DMA (RAM, ROM→VRAM, CRAM, VSRAM) should be resident in RAM or it should be as in LIST1 for example. However, in either one of the above two cases, a long word access is not possible as regards the last VRAM address set.
2. ID should be as in the next page.
3. Put LIST2 at your program's start. This is the U.S. security software.

LIST 1

DMA-RAM:

```
lea           ; vdp_cmd = $c00000
              ; An = Address Register
; Set source Address to VDP Register
; Set Data Length to VDP Register
move.1 xx,ram0      ; xx: Destination Address
                     ; ram0: Work RAM
move.w ram0, (An)
move.w ram0+2, (An) ; Pay careful attention to the sequential order of
                     ; 1st word and 2nd word.
                     ; Destination Address should be set by Word and
                     ; not by Long Word.
```

LIST 2

```
move.b $a10001,d0    ; Get version number
andi.b #$0f,d0
beq.b ?0
move.1 # 'SEGA' $a14000
?0:                   ; If not version #0
                     ; Output ASCII
```

ROM Cartridge Data For Mega Drive

Write in ROM's 100H-1FFH

100H:	'SEGA MEGA DRIVE'	1
110H:	'(C) SEGA 1988.JUL'	2
120H:	Game Name (domestic)	3
150H:	Game Name (overseas)	4
180H:	'GM XXXXXXXX-XX'	5
18EH:	\$XXXX	6
190H:	Control Data	7
1A0H:	\$000000, \$XXXXXX	8
1A8H:	\$FF0000, \$FFFFFF	9
1B0H:	External RAM Data	10
1BCH:	Modem Data	11
1C8H:	Memo	12
1F0H:	Country in which the product can be released	13

- 1: SEGA, system name and Title in common with all ROMs.
- 2: Copyright notice and year/month of release (firm name in ASCII, 4 character).
- 3: Game name for domestic (JIS KANJI Code o.k.).
- 4: Game name for overseas market (JIS KANJI Code o.k.).
- 5: Type of cartridge and products, number, version number:

Type	Game:	GM
	Education:	AI
Number	Product No.	
Version	Data varies depending on the type of ROM or software version	

6. Check Sum

7. I/O use support data

Joystick for MS	:	0
Joystick	:	J
Keyboard	:	K
Serial (RS232C)	:	R
Printer	:	P
Tablet	:	T
Control Ball	:	B
Paddle Controller	:	V
FDD	:	F
CDROM	:	C

8. ROM Capacity Start Address, End Address

9. RAM Capacity Start Address, End Address

10. When no external RAM is mounted, fill the address by a space code; when it is mounted, do the following:

1B0H: dc.b 'RA',%1x1yz000,%00100000
x 1 for Backup and 0 if not Backup
yz 10 if even address only, 11 if odd address only
1B4H: dc.1 RAM start address, RAM end address

11. If corresponding to modem, fill it by space code; if not, do the following

1BCH: dc.b 'MO','xxxx','yy.z'
xxxx Firm name, the same as in 2
yy Modem number
z Version

13. Data on the countries in which the product can be released

Japan : J
USA : U
Europe : E

Be sure to input a space code in the unoccupied 1-7-9-13 space.

segad

onisoft

How to Obtain a Check Sum

The Check Sum obtaining program is shown as follows. The program starts with 0FF8000H, RAM space.

First, fill game capacity by -1 (0FFH) and then load all of the programs. Next, load the Check Sum program and run the program from 0FF8000H.

After a while, stop running the program. At this time, the lower Word of Data Register 0 (d0) is the Check Sum value. Note that Break in Memory should be cancelled in advance.

Also, when burn-ins to ROM, first fill the game capacity by -1 (0FFH).

```
end_addr    equ      $1a4
org        -$8000
start:
        move.1    (a0),d1
        addq.1    #$1,d1
        movea.1   #$200,a0
        sub.1     a0,d1
        asr.1     #1,d1      ; counter
        move      d1,d2
        subq.w   #$1,d2
        swap      d1
        moveq   #$0,d0
•?12:
        add      (a0)+,d0
        dbra    d2,?12
        dbra    d1,?12
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
?1e:
        nop
        nop
        bra.b   ?1e
```

Memory Mapping for Emulation

For the 68000 Emulation

All address should be disabled initially: 0 to 0FFFFFF

Required areas should then be enabled as follows:

1. Program and Data are in 0 to 007FFFF
2. S-RAM is for Z80 in 0A00000 to 0A01FFF
3. FM sound chip interface is in 0A04000 to 0A04FFF
4. I/O and Z80 control port are in 0A10000 to 0A11FFF
5. VDP and sound control port are in 0C00000 to 0C00FFF
6. Scratch RAM is in 0FF0000 to 0FFFFFF

RAM Card (No. 171-5642-02)

This board has two memory areas:

Main Memory	(D-RAM) \$000000 - \$0FFFFFF
Backup Memory	(S-RAM) \$200000 - \$203FFFF

1. Initialize
Write 0100H into \$0A11000
Write 1 into \$0A130F0
(Green LED light up)
2. Write Protect
Write 3 into \$0A130F0
(Red LED light up)
3. Read/Write
Write 1 into \$0A130F0
(Red LED turns off)
4. Note: Emulator access to these ports should be enabled before the writes, then disabled after words.

Mega Drive Registers Fixed Bits (40 Cell and NTSC Mode)

RO	0	0	0		0	1	0	0
1	0				0	1	0	0
2	0	0				0	0	0
3	0	0						0
4	0	0	0	0	0			
5	0							
6	0	0	0	0	0	0	0	0
7	0	0						
8	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0
10								
11	0	0	0	0	0			
12	1	-	-	-				1
13	0	0						
14	0	0	0	0	0	0	0	0
15								
16	0	0			0	0		
17	0	0						
18	0	0						
19								
20								
21								
22								
23								

* DMA cannot be performed emulated ROM or RAM on most ICES.

sega ~~onisoft~~