GENESIS

SOFTWARE MANUAL

SEGA ENTERPRISES

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)

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. $C0000000 (DATA PORT) 10
2. $C0000004 (CONTROL PORT) 11
3. $C0000008 (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

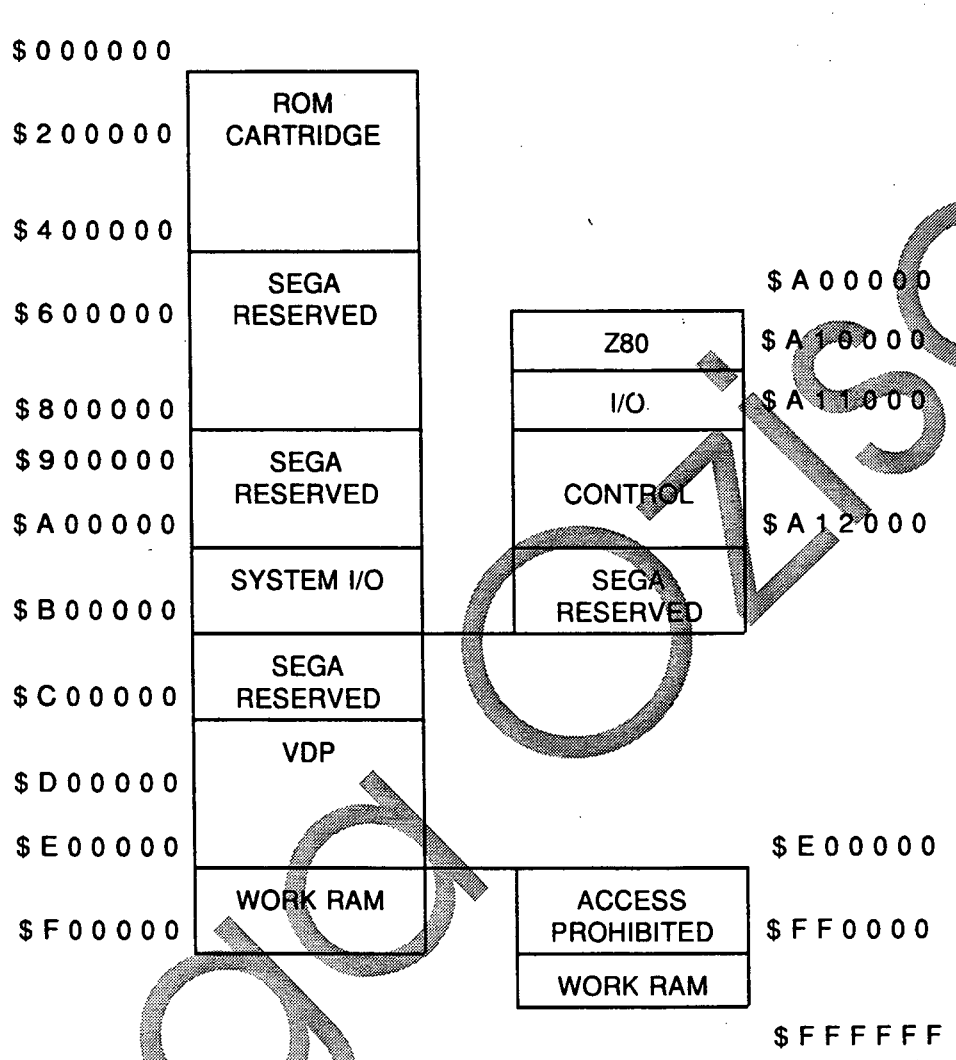
A. 280 MAPPING
B. 68K CONTROL OF Z80
C. FM SOUND CONTROL
D. PSG CONTROL
E. D/A CONTROL

VII. APPENDIX

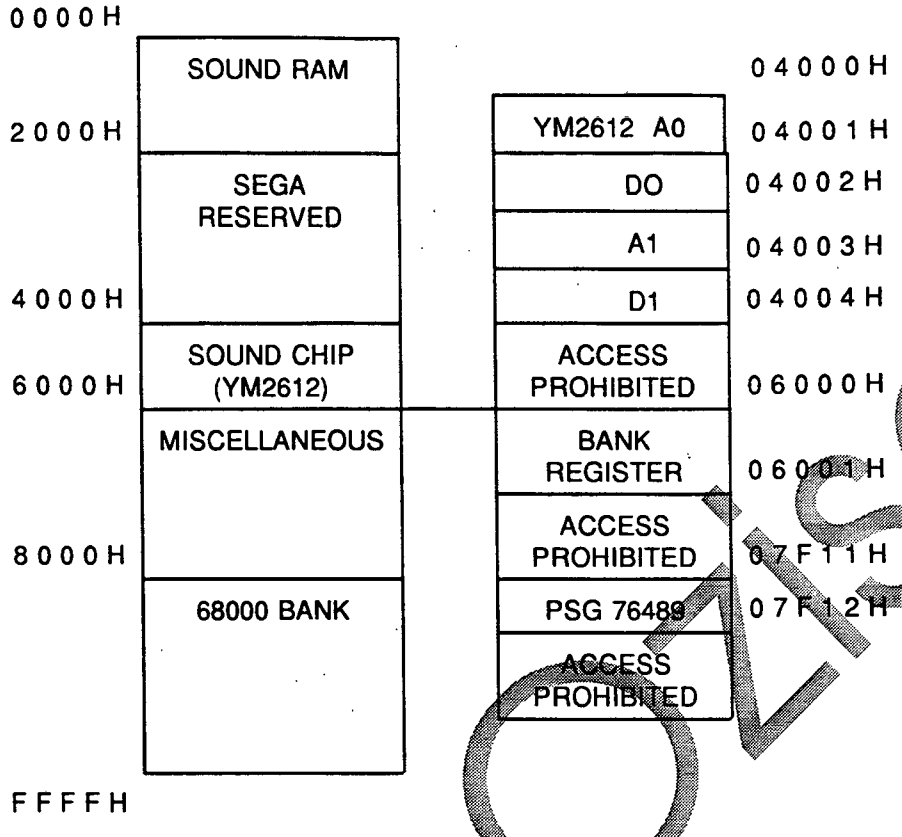
I. MEMORY MAP

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

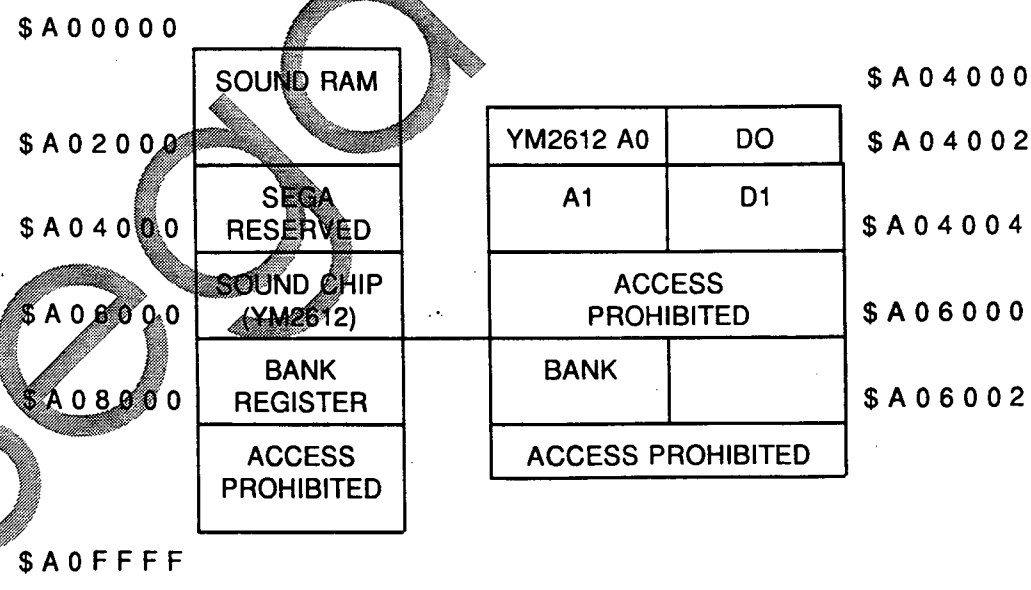
1. 68K MEMORY MAP



2. Z80 MEMORY MAP



3. 68000 ACCESS TO Z80 MEMORY



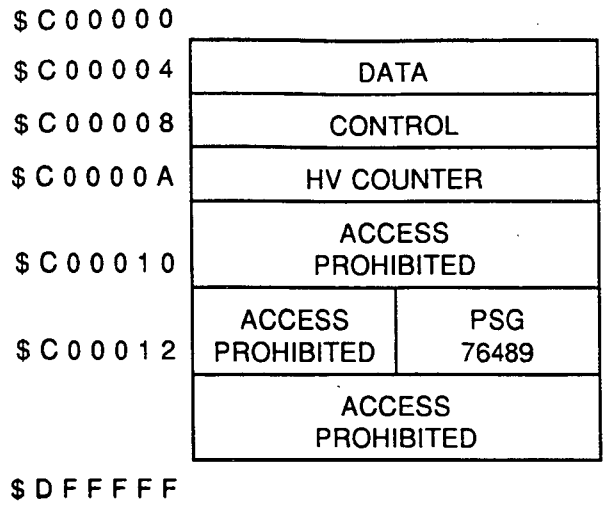
4. I/O AREA

$A10000		
$A10002		Version No.
$A10008		DATA (CTRL 1) DATA (CTRL 2) DATA (EXP)
$A1000E		CONTROL (1) CONTROL (2) CONTROL (E)
$A10014		T x DATA R x DATA (1) S - MODE
$A1001A		T x DATA R x DATA (2) S - MODE
$A10020		T x DATA R x DATA (E) S - MODE
	ACCESS PROHIBITED	
$A1FFFF		

5. CONTROL AREA

$A11000	
$A11002	MEMORY MODE
$A11100	ACCESS PROHIBITED
$A11102	Z80 BUSREQ
$A11200	ACCESS PROHIBITED
$A11202	Z80 RESET
	ACCCESS PROHIBITED
$A1FFFF	

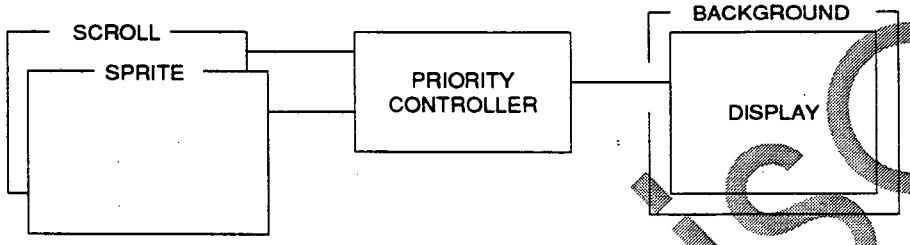
6. VDP AREA



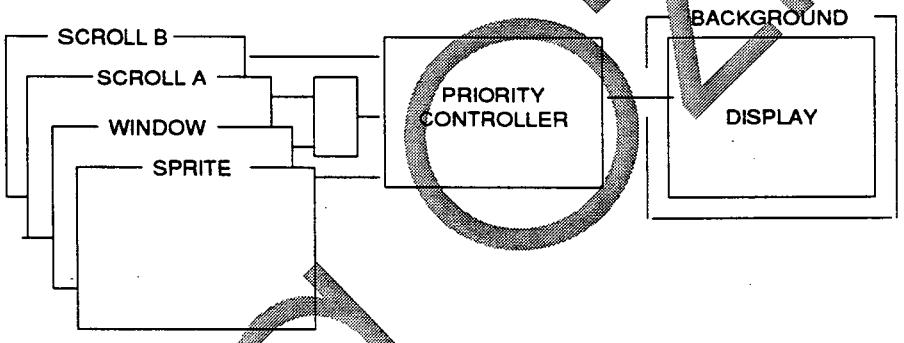
II. VDP 315-5313
(Video Display Processor)

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.

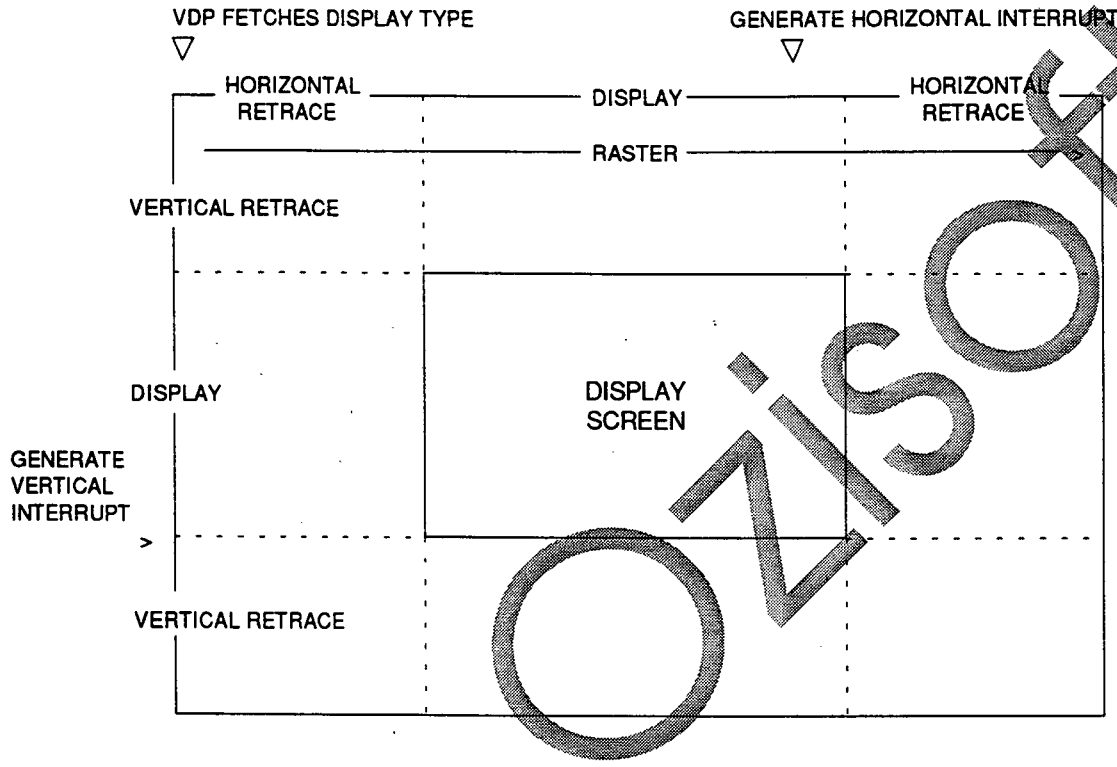
B. DISPLAY SPECIFICATION

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 NTCS 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.

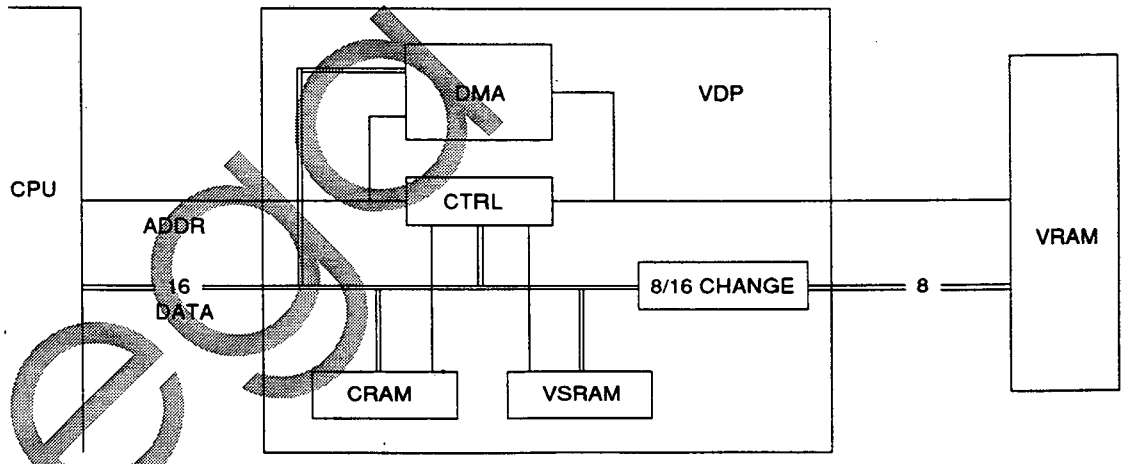
2. VRAM (VDP RAM)
   General purpose storage area for display data.

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.

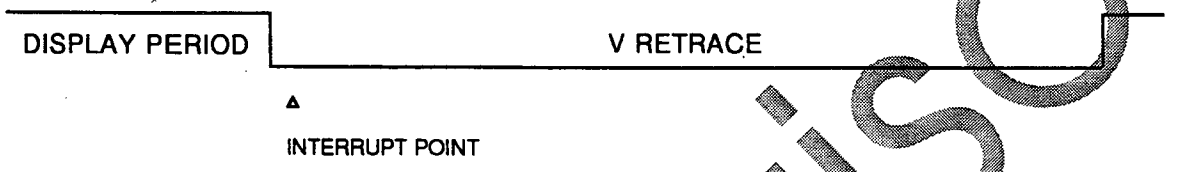


D. INTERRUPT

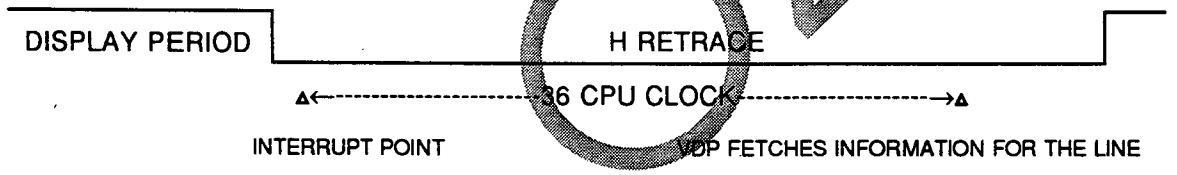
There are three interrupts, Vertical, Horizontal, and External. You can control each interrupt by the IEO, IE1, and IE2 bits in the VDP registers. The interrupts use the AUTO-VECTOR mode of the 68000 and are at levels 6, 4, and 2 respectively; the level 6 vertical interrupt having the highest priority.

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

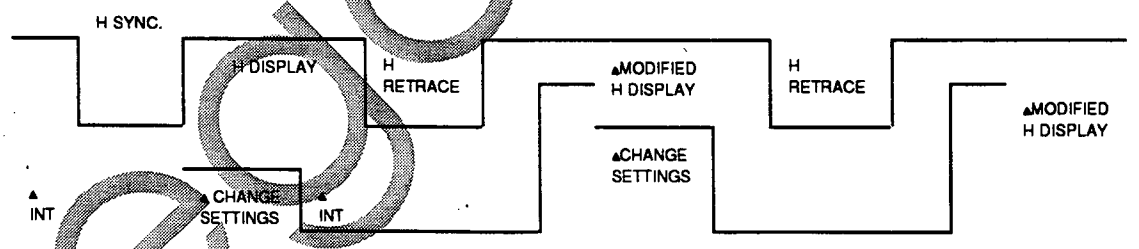
1. VERTICAL INTERRUPT (V-INT)
The vertical interrupt occurs just after V retrace.



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.



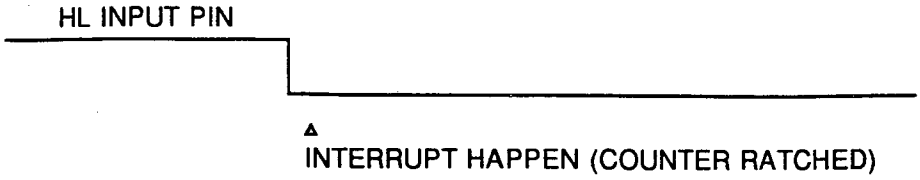
The horizontal interrupt is controlled by a line counter in register #10.

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.



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
$C00000	DATA PORT	
$C00002	"	
$C00004	CONTROL PORT	
$C00006	"	
$C00008	HV COUNTER	
$C0000A	PROHIBITED	
$C0000C	PROHIBITED	
$C0000E	PROHIBITED	
$C00010	PROHIBITED	PSG

1. $C00000 (DATA PORT)

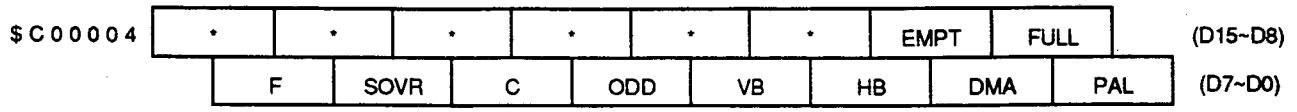
READ/WRITE: VRAM, VSRAM, CRAM

$C00000	DT15	DT14	DT13	DT12	DT11	DT10	DT9	DT8	(D15~D8)
	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0	(D7~D0)

* $C00000 and $C00002 are functionally equivalent.

2. $C00004 (CONTROL PORT)

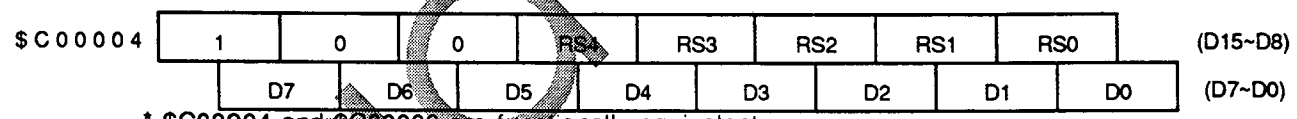
READ: STATUS REGISTER



* 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

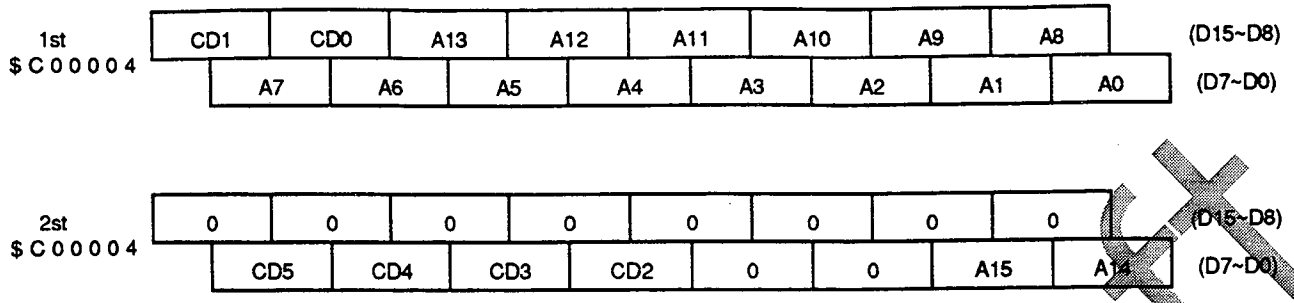


* $C00004 and $C00006 are functionally equivalent.

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



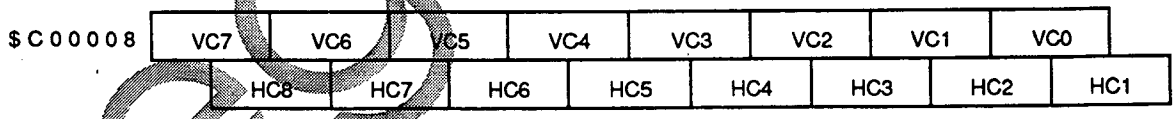
CD5 ~ CD0: ID CODE
A15 ~ A0: DESTINATION RAM ADDRESS

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	1	0	0	0
VSRAM READ	0	0	0	1	0	0

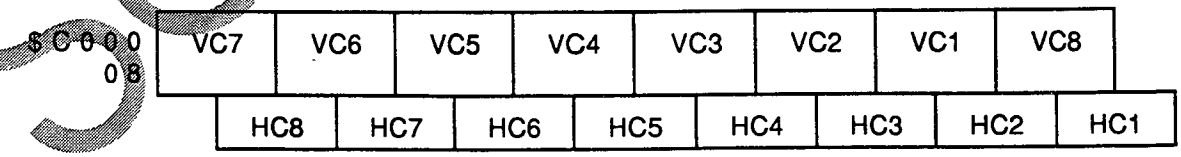
* You must use word or long word when performing these operations.

3. $C00008 (HV COUNTER)

NON INTERLACE MODE



INTERLACE MODE



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

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

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

	MSB							LSB
REG. #1	0	DISP	IEO	M1	M2	1	0	0

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 28 cell mode (PAL mode; always 0 in NTSC mode)

3. PATTERN NAME TABLE BASE ADDRESS FOR SCROLL A

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

VRAM ADDR %XXX0_0000_0000_0000

4. PATTERN NAME TABLE BASE ADDRESS FOR WINDOW

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

WDII 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

RSO.. .. .. 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 HILIGHT
0: Disable SHADOW and HILIGHT

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	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

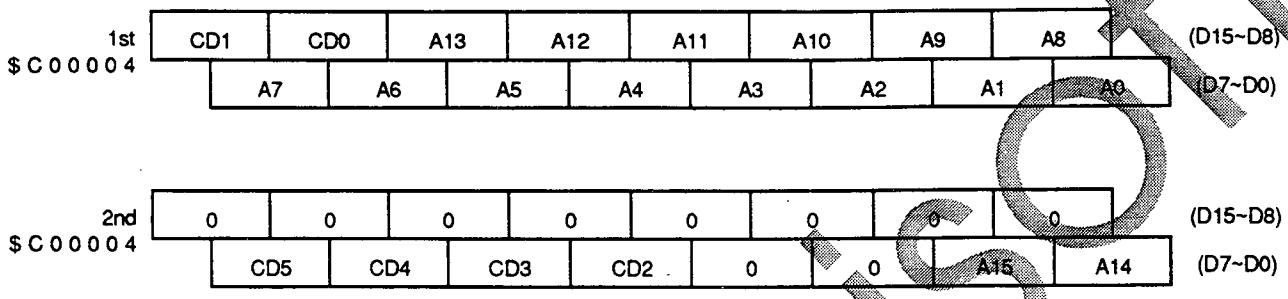
DMD1	DMD0	FUNCTION
0	SA23	MEMORY TO VRAM
1	0	VRAM FILL
1	1	VRAM COPY

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.



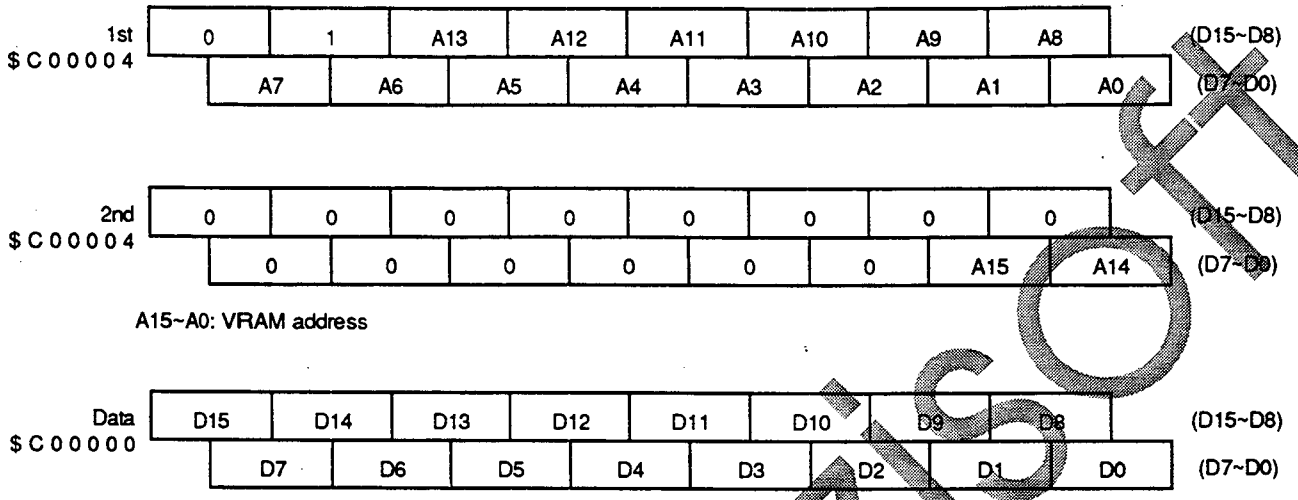
CD5~CD0: ID CODE
A15~A0: DESTINATION RAM ADDRESS

	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	0	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.

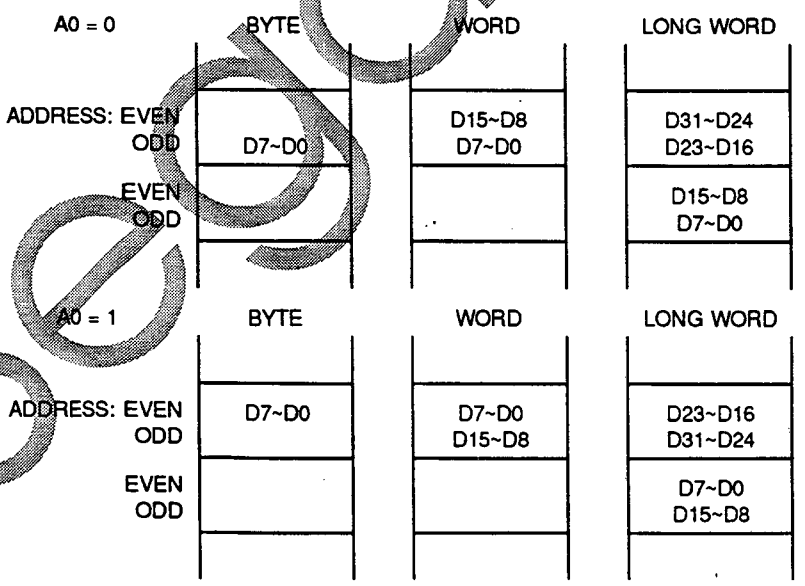
VRAM access addressing is as follows when writing:



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

START ADDRESS: 0 REG. #15=1

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

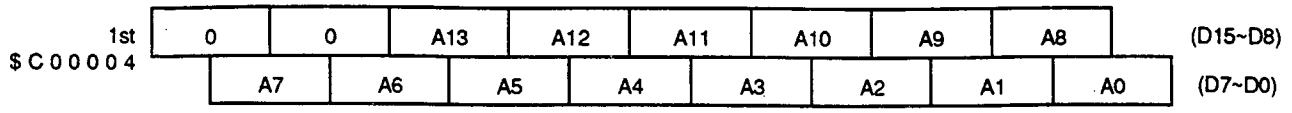
START ADDRESS: 1 REG. #15=2

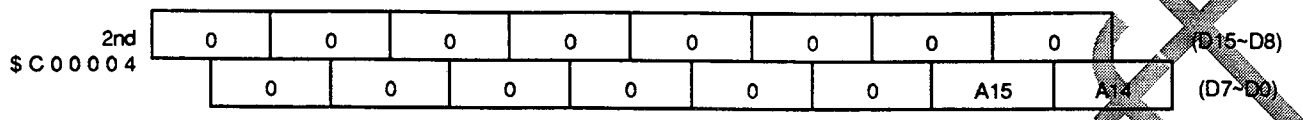
ADDRESS	BYTE	WORD	LONG WORD
0	1st D7~D0	1st D7~D0 D15~D8	1st D23~D16 D31~D24
1			
2	2nd D7~D0	2nd D7~D0 D15~D8	1st D23~D16 D31~D24
3			
4	3rd D7~D0	3rd D7~D0 D15~D8	2nd D23~D16 D31~D24
5			
6	4th D7~D0	4th D7~D0 D15~D8	2nd D23~D16 D31~D24
7			
8	5th D7~D0	5th D7~D0 D15~D8	3rd D23~D16 D31~D24
9			

START ADDRESS: 1 REG. #15=1

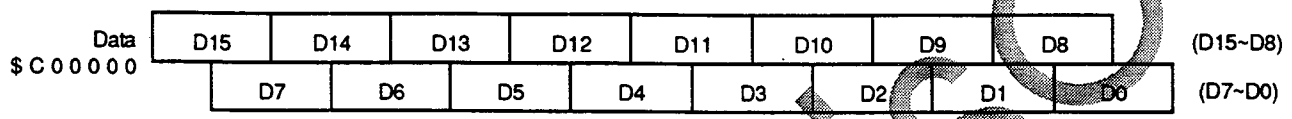
ADDRESS	BYTE	WORD	LONG WORD
0	1st D7~D7	1st D7~D0 D15~D8	1st D23~D16 D31~D24
1			
2	3rd D7~D7	3rd D7~D0 D15~D8	2nd D23~D16 D31~D24
3	2nd D7~D7		
4	5th D7~D7	5th D7~D0 D15~D8	3rd D23~D16 D31~D24
5	4th D7~D7		
6	7th D7~D7	7th D7~D0 D15~D8	4th D23~D16 D31~D24
7	6th D7~D7		
8	9th D7~D7	9th D7~D0 D15~D8	5th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

VRAM READ





A15~A0: VRAM ADDRESS



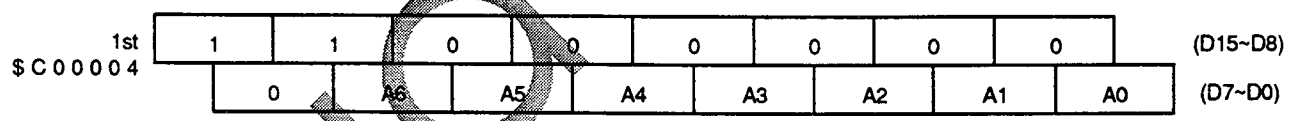
D15~D0: VRAM DATA

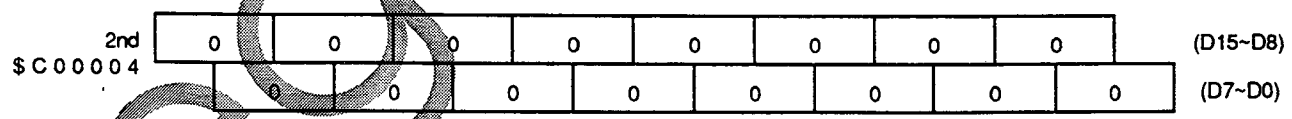
The data is always read in word units. A0 is ignored during the reading; no swap of bytes occurs if A0=1.

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

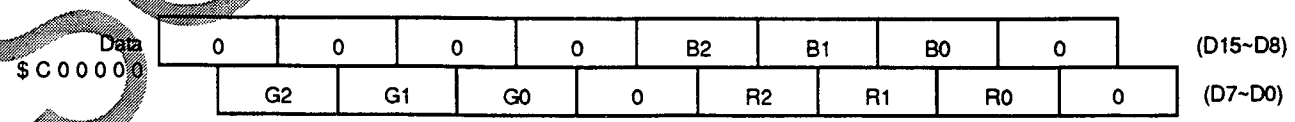
3. CRAM ACCESS

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





A6~A0: CRAM ADDRESS



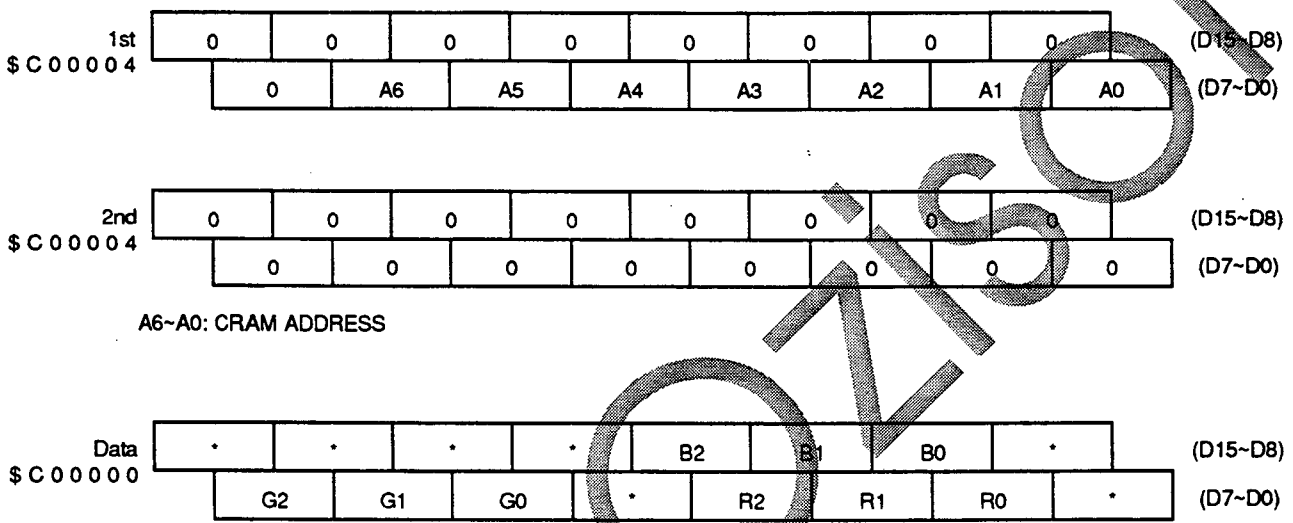
D15~D0 are valid when we use word for data set. If the writes are byte wide, write the high byte to $C00000 and the low byte to $C00001.

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.

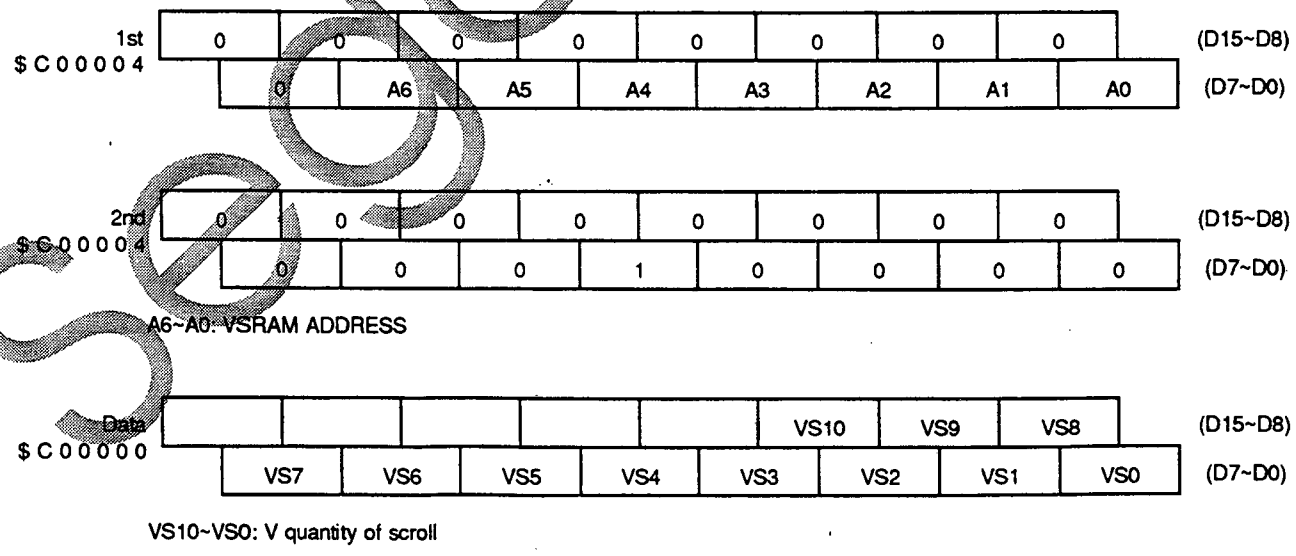
Note that A0 is used in the increment, but not in address decoding, resulting in some interesting side effects if writes are attempted at odd addresses.

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:

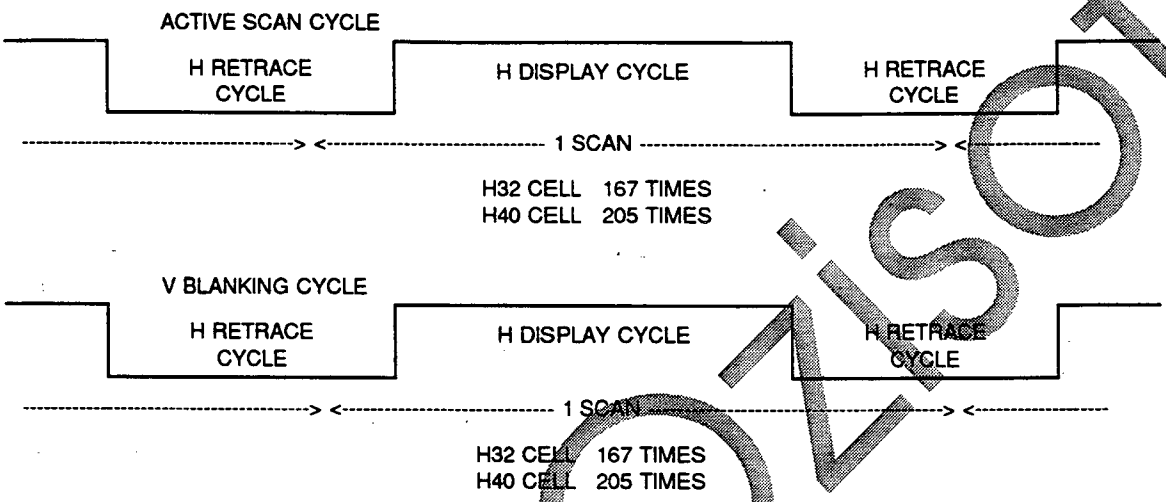


VS10~VS0: V quantity of scroll

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 acceess 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.

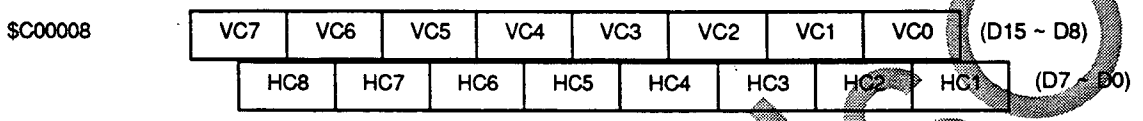
6. HV COUNTER

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.

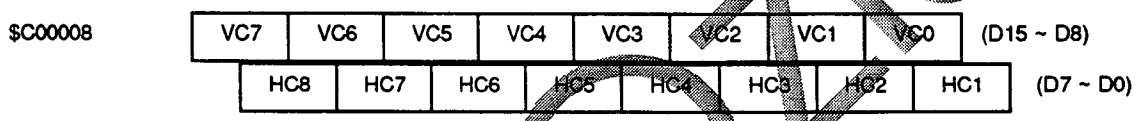
M3	Counter Latch Mode
0	Counter is not latched by trigger signal
1	Counter is latched by trigger signal

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

H-Counter: HC8 ~ HC1

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.

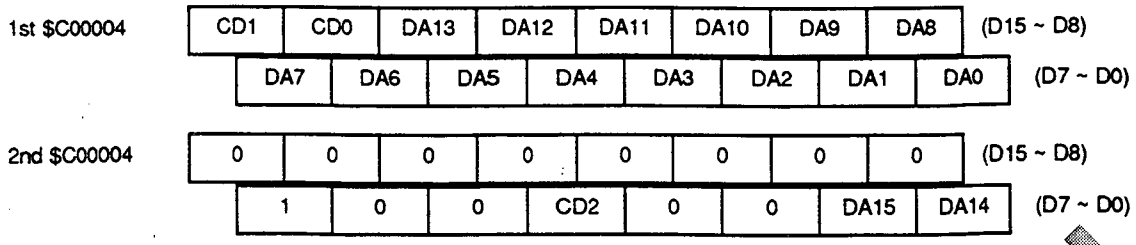
You must set M1=1 only during DMA; otherwise, we cannot guarantee the operation. Source addresses were increased with +2 and destination address increased with content of Register #15.

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



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 F.

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	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.)

VRAM address is even

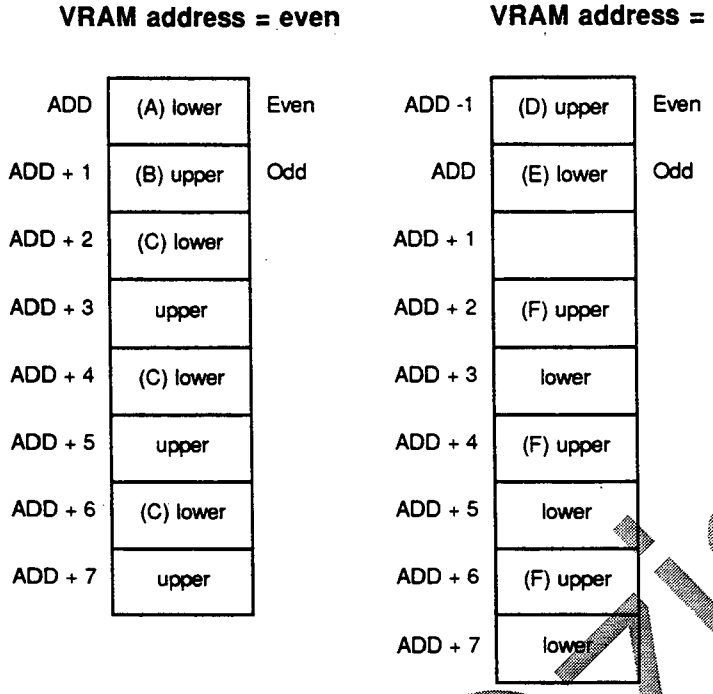
ADD	(A)	Even
ADD + 1	(B) (C)	Odd
ADD + 2	(C)	
ADD + 3	(C)	
ADD + 4	(C)	
ADD + 5	(C)	
ADD + 6	(C)	
ADD + 7		

VRAM address is odd

ADD -1	(D)	Even
ADD	(E)	Odd
ADD + 1	(F)	
ADD + 2	(F)	
ADD + 3	(F)	
ADD + 4	(F)	
ADD + 5	(F)	
ADD + 6	(F)	
ADD + 7		

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

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



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

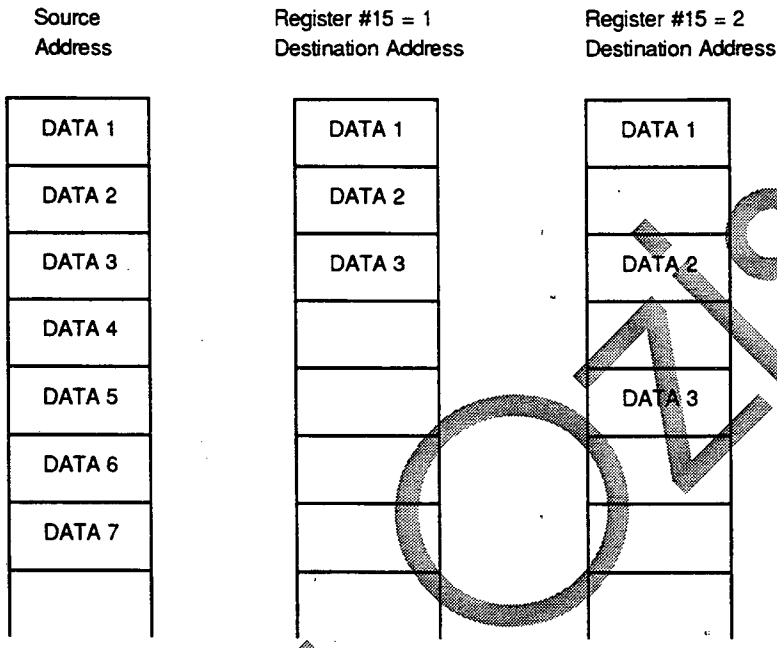
This function copies from source address to destination address by number of Copy byte.

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

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.

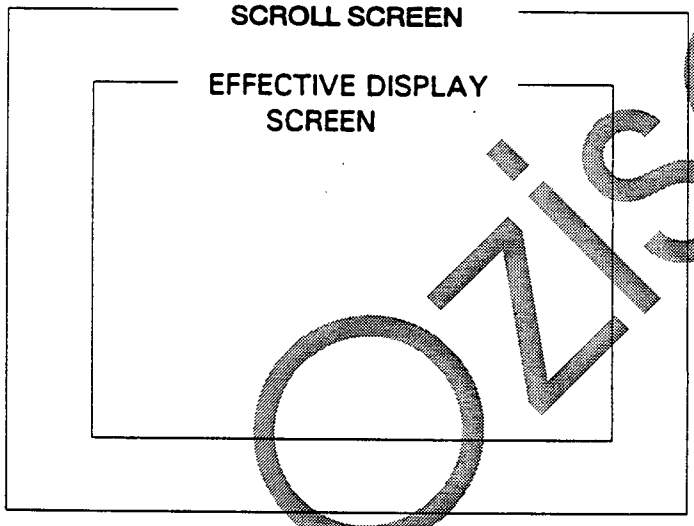
Note that when calculating, the transfer quantity in one screen (1/60 sec) varies depending on the number of lines during V Blank (refer to *Display Mode*) in the case of NTSC (video signal) and PAL systems.

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.



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

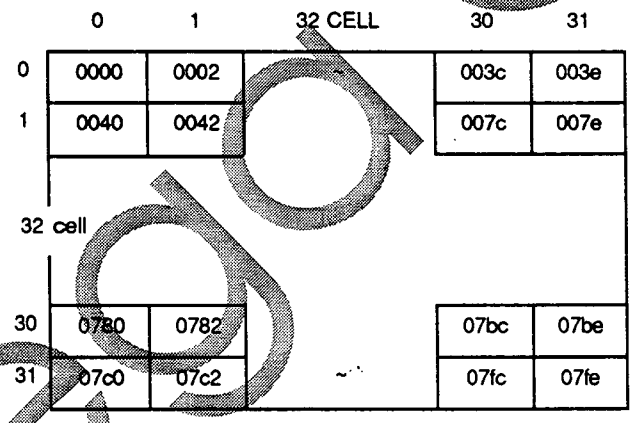
VSZ1	VSZ0	Function
0	0	V 32 cell
0	1	V 64 cell
1	0	Prohibited
1	1	V 128 cell

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 exits in the VRAM and is designated by Registers #2 and #4. Depending VRAM and Scroll Screen correspond to each other differently.

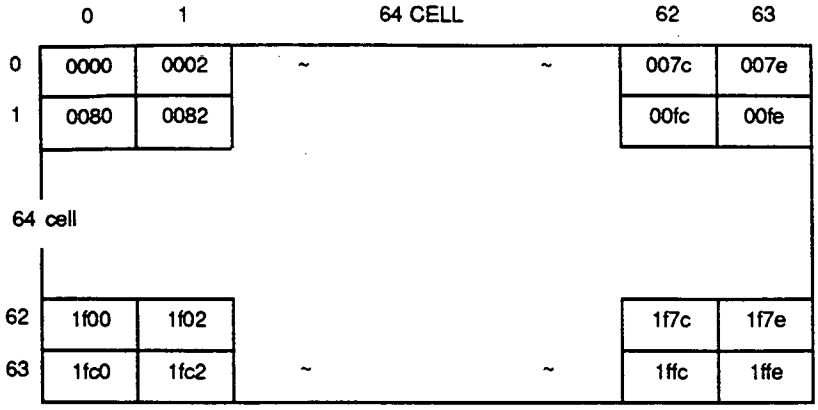
Example

Register #16 = 00H: 32*32 cell



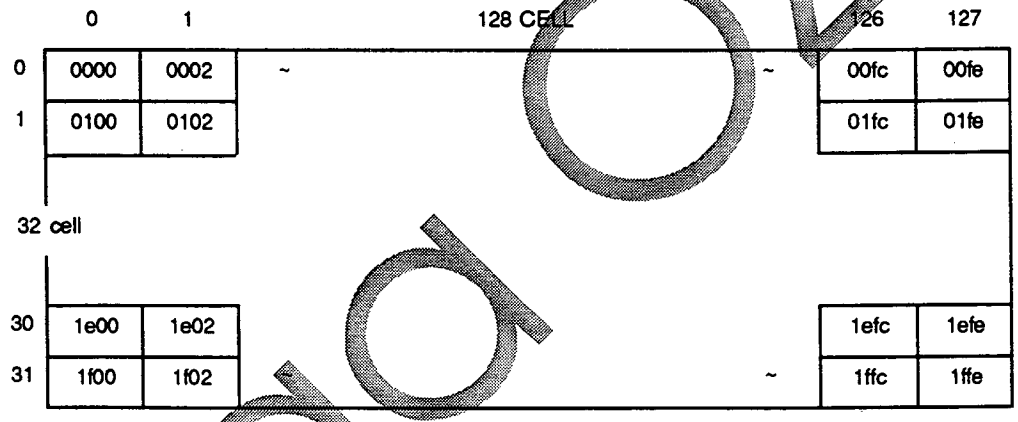
Example

Register #16 = 11H: 64*64 cell



Example

Register #16 = 03H: 32*128 cell



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

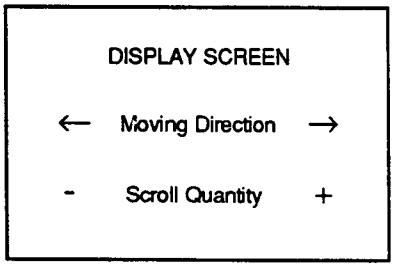
2. HORIZONTAL SCROLLING

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.

APPENDIX

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

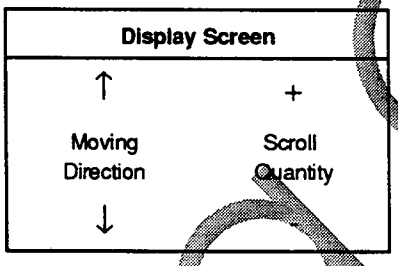
The scrolling quantity is equivalent to 11 bits (000H ~ 7FFH). However, it will be as shown below in the Interlace Mode.

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.

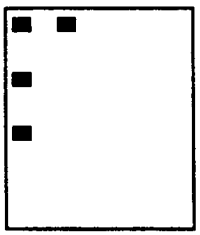
4. SCROLL PATTERN NAME

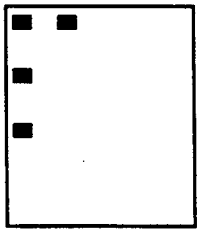
The Scroll Screen's name table is in VRAM and set by Registers #2 and #4. The Pattern Name requires 2 bytes (1 word) per cell the Scroll Screen. Depending on the Scroll Screen's size, VRAM and Scroll Screen correspond with each other differently. Refer to Scroll Screen Size.

Pattern Name	pri	cp1	cp0	vf	hf	pt10	pt9	pt8	(D15 ~ D8)
	pt7	pt6	pt5	pt4	pt3	pt2	pt1	pt0	(D7 ~ D0)

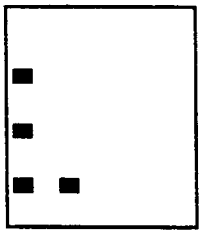
pri: Refer to Priority
cp1: Color palette selection bit (see Color Palette)
cp0: Color palette selection bit (see Color Palette)
vf: V Reverse Bit 1: Reverse
hf: H Reverse Bit 1: Reverse
pt10 ~ pt0: Pattern Generator Number

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

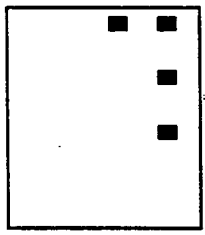




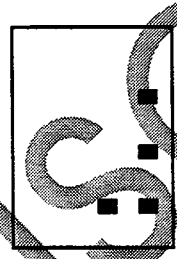
vf = 0
hf = 0



vf = 1
hf = 0



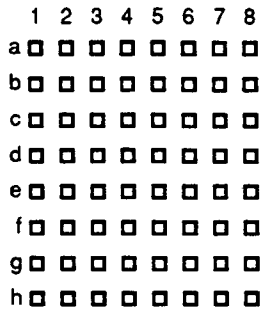
vf = 0
hf = 1



vf = 1
hf = 1

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:

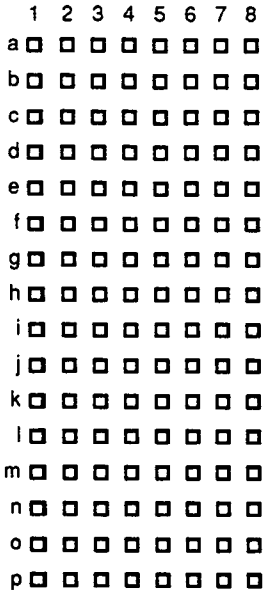


	0		1		2		3	
	D7	D0	D7	D0	D7	D0	D7	D0
00	a1	a2	a3	a4	a5	a6	a7	a8
04	b1	b2	b3	b4	b5	b6	b7	b8
08	c1	c2	c3	c4	c5	c6	c7	c8
0C	d1	d2	d3	d4	d5	d6	d7	d8
10	e1	e2	e3	e4	e5	e6	e7	e8
14	f1	f2	f3	f4	f5	f6	f7	f8
18	g1	g2	g3	g4	g5	g6	g7	g8
1C	h1	h2	h3	h4	h5	h6	h7	h8

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.



	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
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

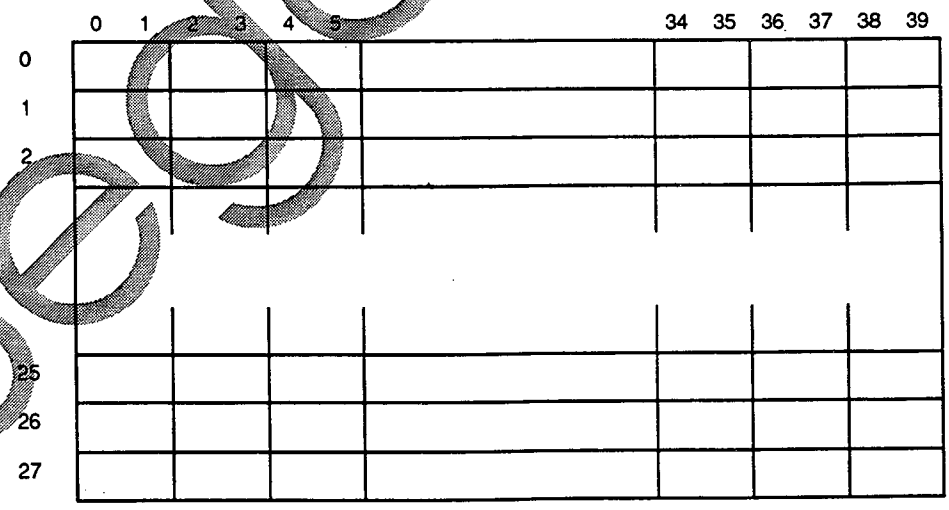
Window V Position

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 cell basis of H2 cells and V1 cell. The dividing position varies depending on resolution.



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.

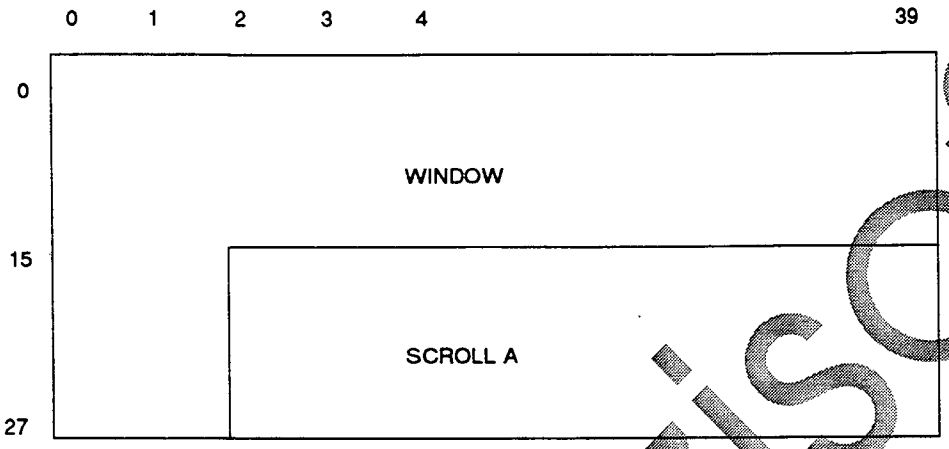
WHP5 ~ WHP1: H dividing position
WVP4 ~ WVP0: V dividing position

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

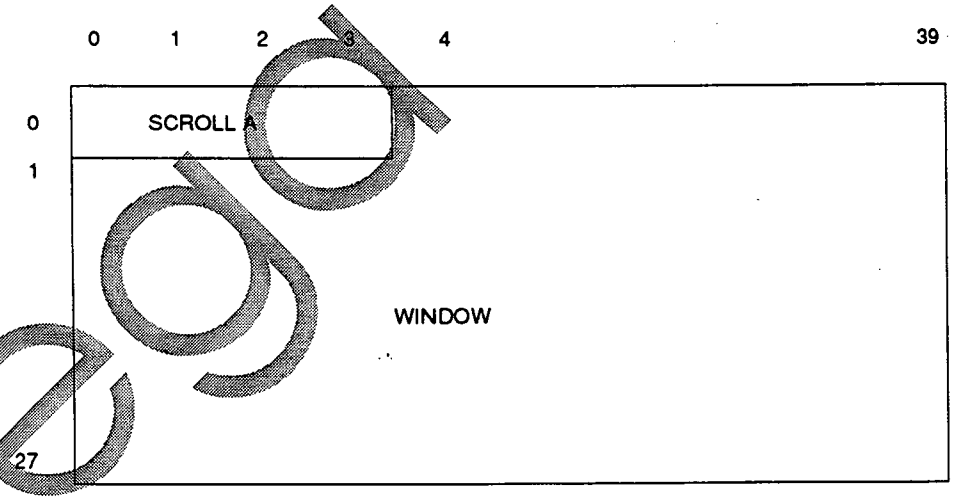
Setting Example

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



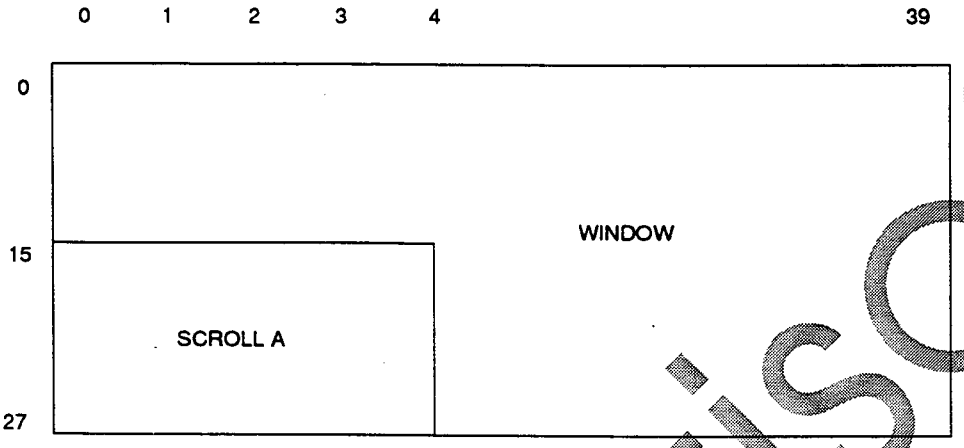
DISPLAY SCREEN: 40 X 28 CELL MODE

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



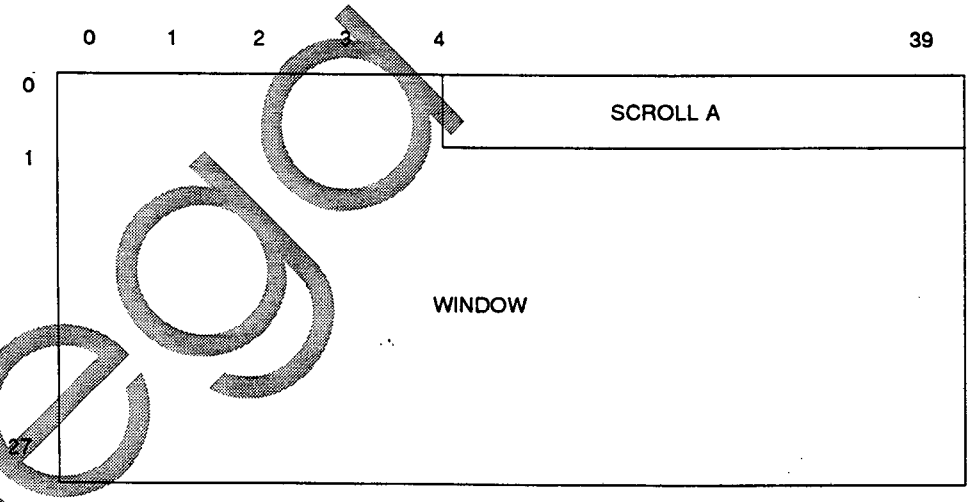
DISPLAY SCREEN: 40 X 28 CELL MODE

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



DISPLAY SCREEN: 40 X 28 CELL MODE

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

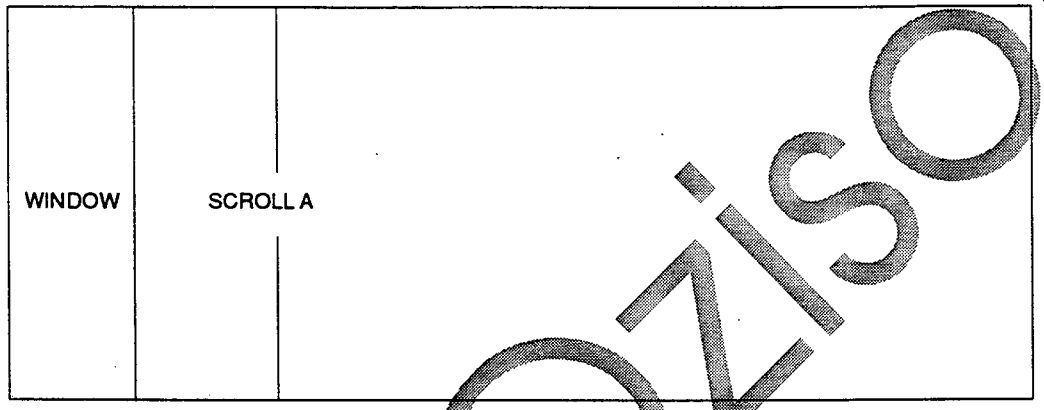


DISPLAY SCREEN: 40 X 28 CELL MODE

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.

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)

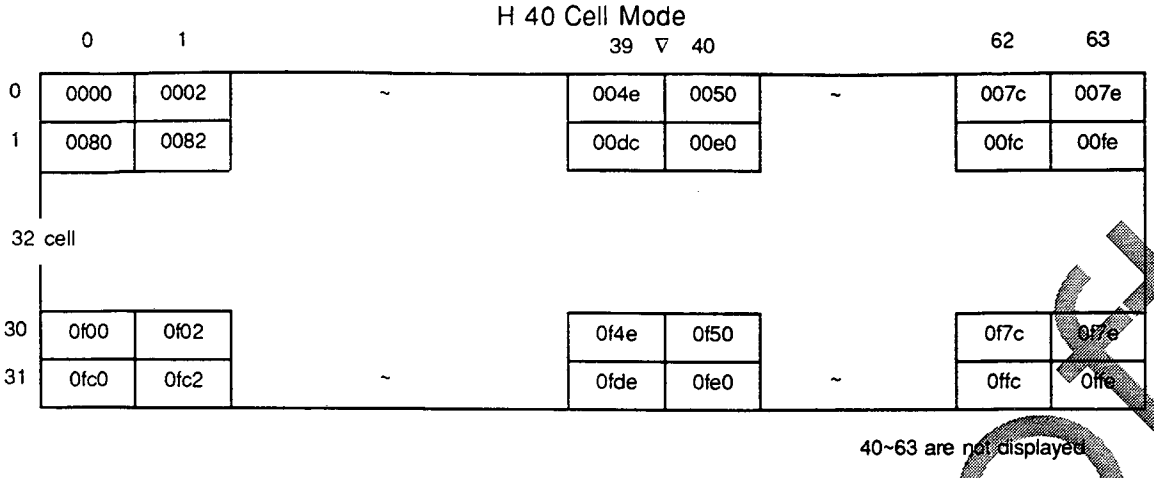
pri:	Refer to Priority
cp1:	Color palette selection bit
cp0:	Color palette selection bit
vf:	V reverse Bit 1: Reverse
hf:	H Reverse Bit 1: Reverse
pt10~pt0:	Pattern Generator Number

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

H 32 Cell Mode

	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

H 40 Cell Mode



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

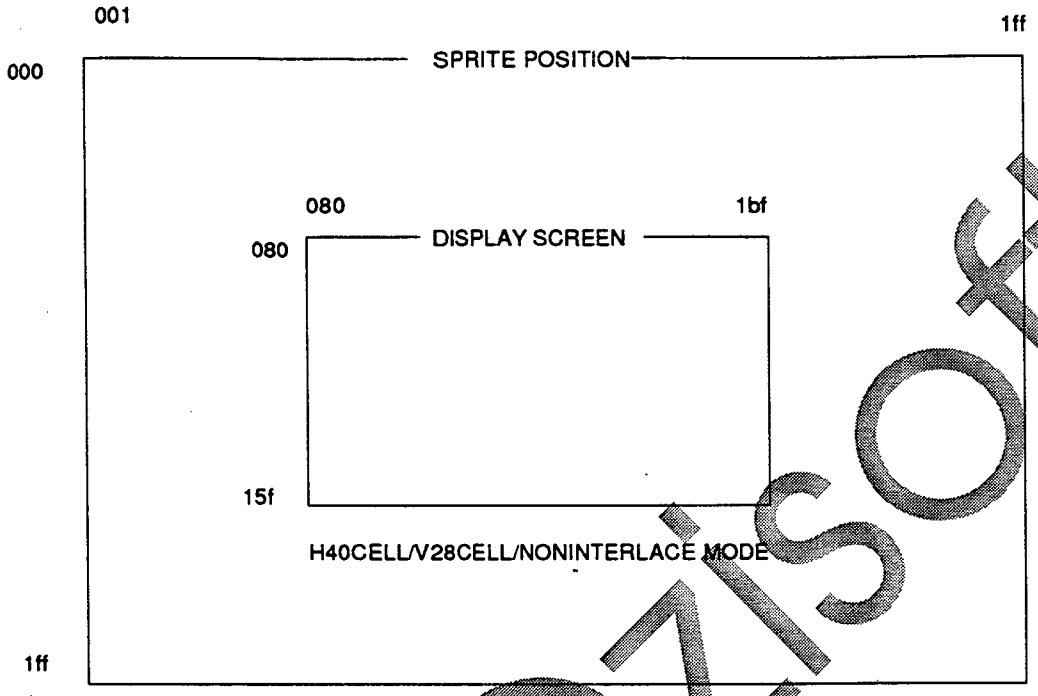
VRAM: Sprite Attribute Table Maximum 640 Bytes

1. DISPLAY POSITION

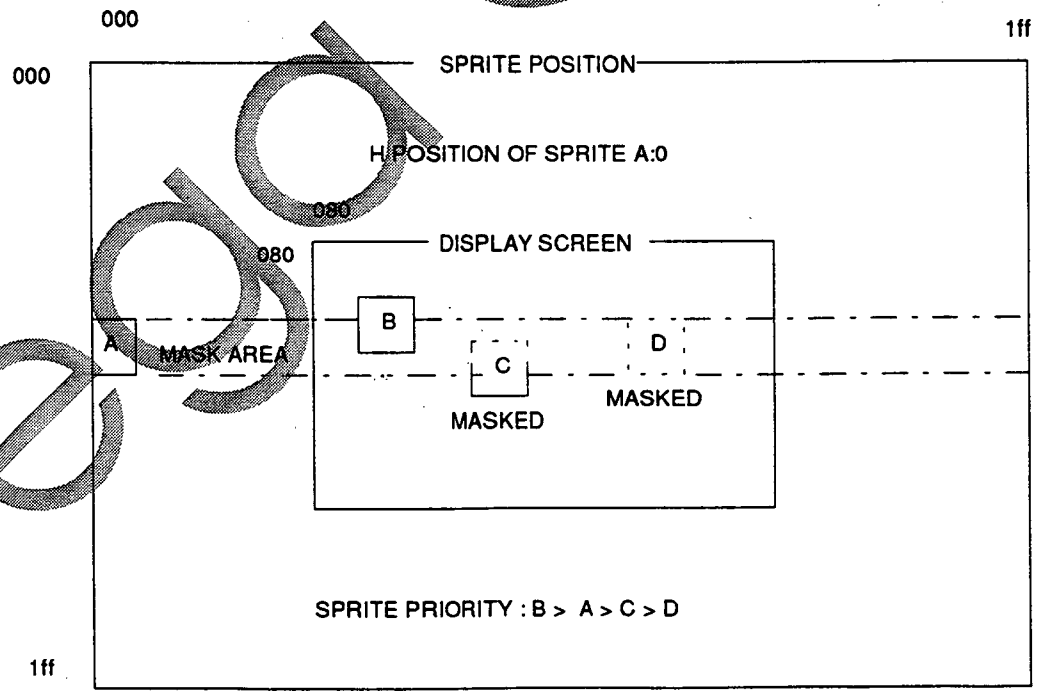
Sprite Position and Display Screen are as follows: When Sprite H position is 0, this is in a special mode. Therefore, pay careful attention to this point.

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
	Interlace 1		
	Interlace 2	000 ~ 3FFH	100 ~ 2BFH
V30 Cell	Non-interlace	000 ~ 1FFH	080 ~ 16FH
	Interlace 1		
	Interlace 2	000 ~ 3FFH	100 ~ 2DFH



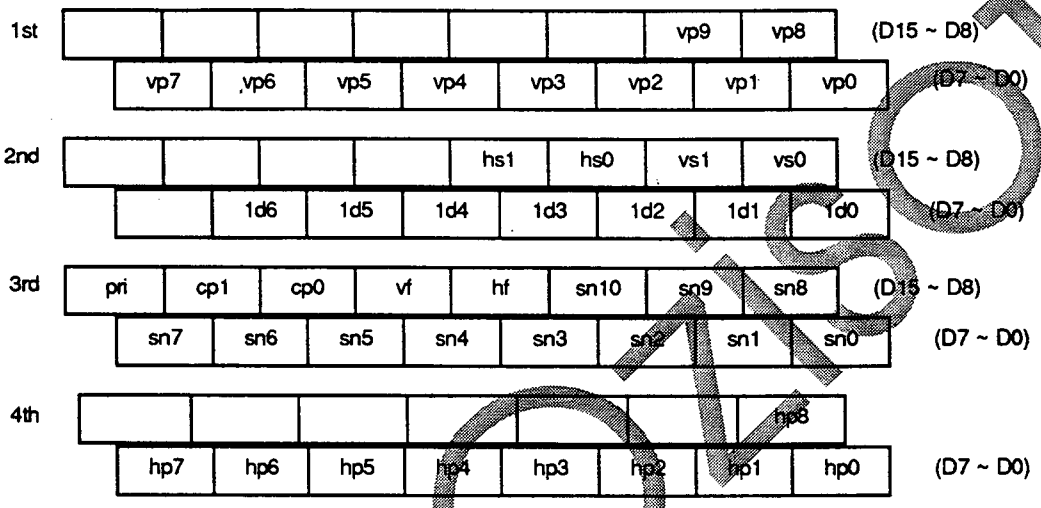
When 0 is set to the SPRITE H POSITION, a low priority sprite on the same line will not be displayed.



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.

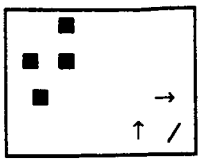
Starting from the beginning of the Attribute Table, numbers are given in the sequence of Sprite 0, Sprite 1, Sprite 2, Sprite 3, etc. Priority between Sprites is not determined by the sequence of Sprite number, but by each Sprite's Link Data, and thus becomes programmable.

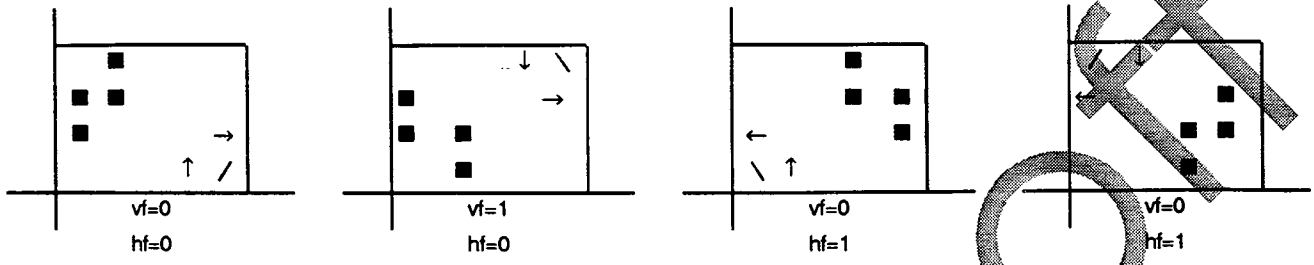


Blank portions can be utilized freely for software.

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

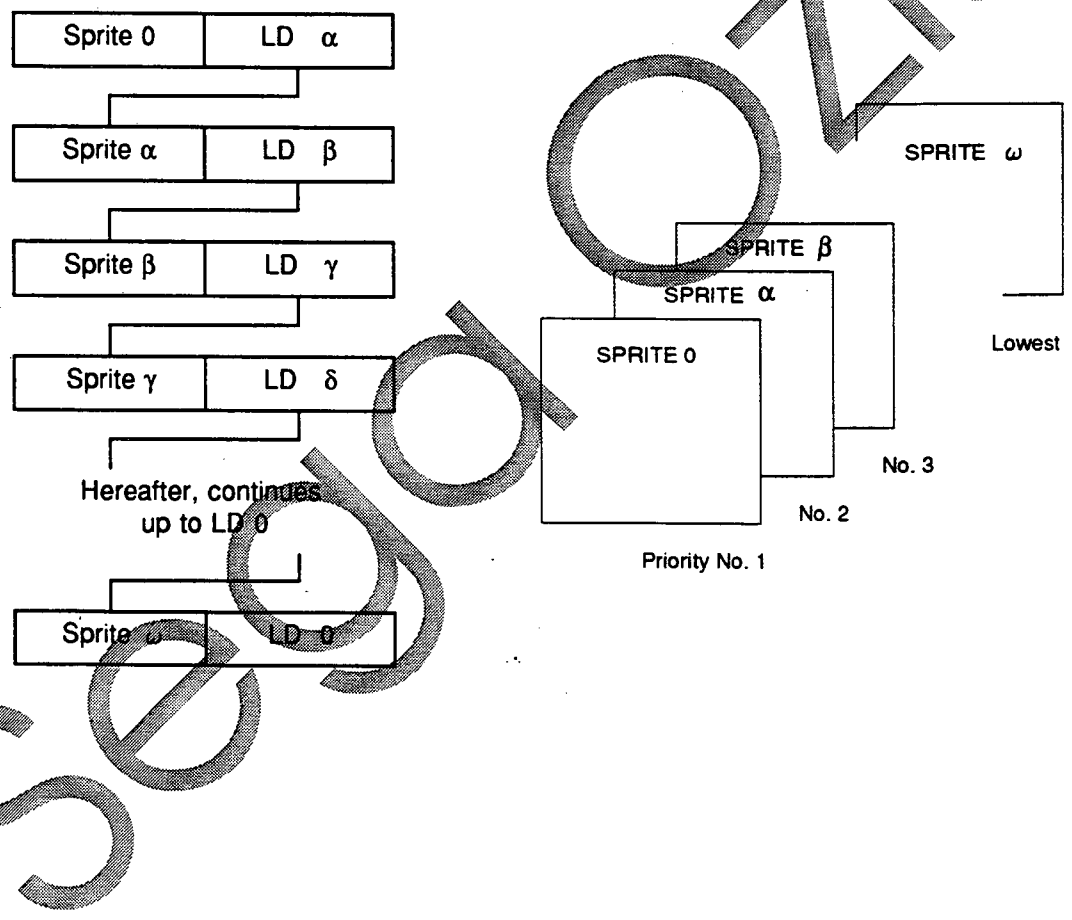
Priority between Sprites is designated by each Sprite's Link Data.

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

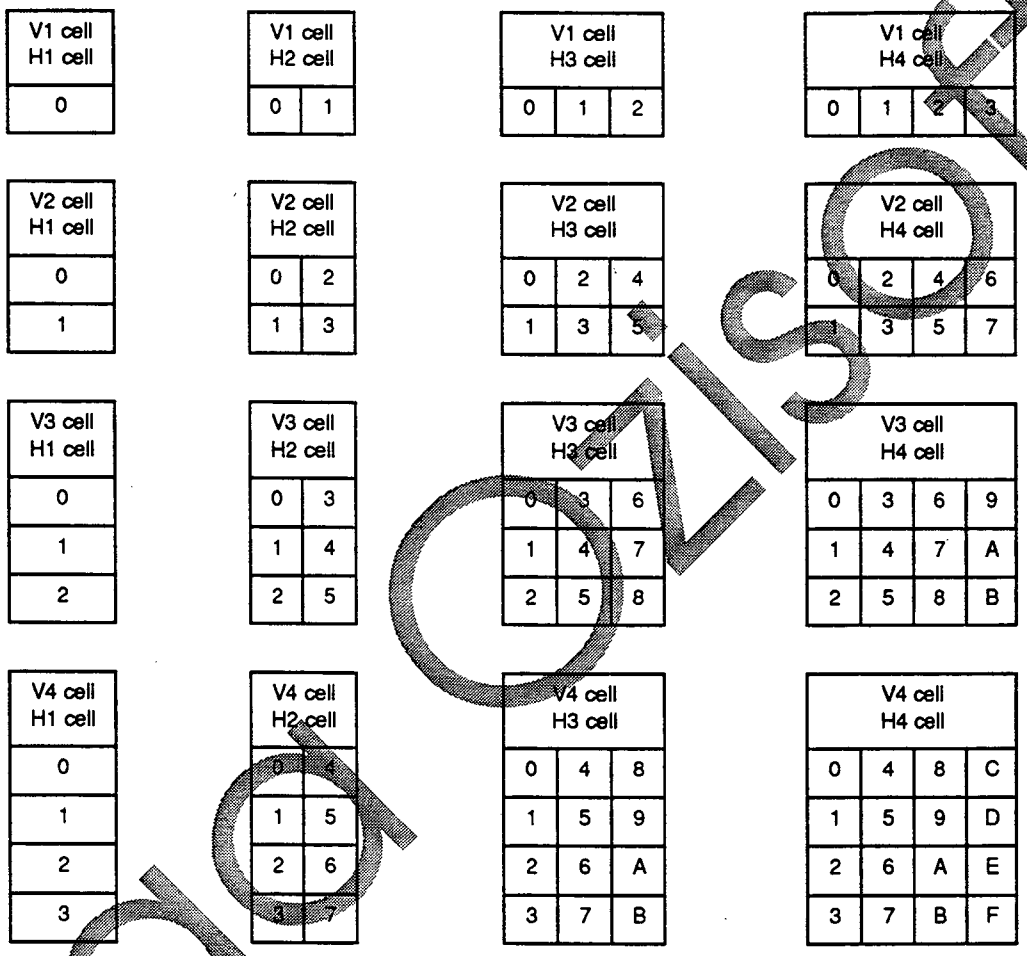
	Link Data
Sprite 0	2
Sprite 1	10
Sprite 2	1
Sprite 3	4
Sprite 4	5
Sprite 5	15
Sprite 6	—
Sprite 7	0
Sprite 8	—
Sprite 9	—
Sprite 10	11
Sprite 11	13
Sprite 12	—
Sprite 13	3
Sprite 14	—
Sprite 15	7
Sprite 16	—

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.

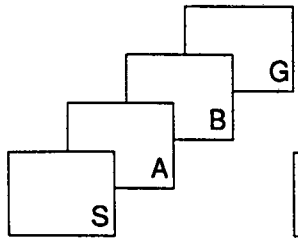
Priority can be designated by each Pattern Name and Attribute Priority bit. It will be set for the Scroll Screen on a cell unit basis and for each Sprite. By combining each priority bit, Priority will be as follows, however, the Background Priority is always the lowest.

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

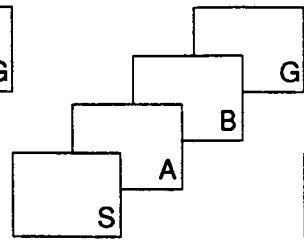
S: Sprite
A: Scroll A
B: Scroll B
G: Background

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

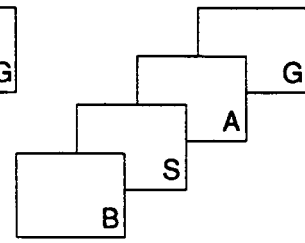
S/TEN = 0



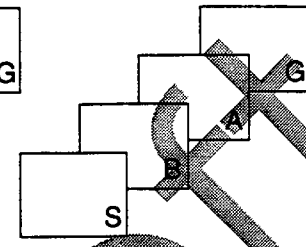
A = 0
B = 0
S = 0



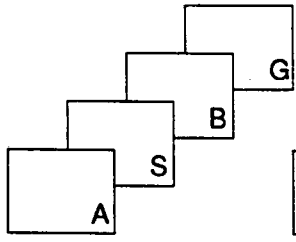
A = 0
B = 0
S = 1



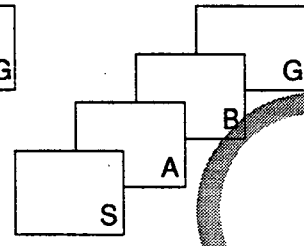
A = 0
B = 1
S = 0



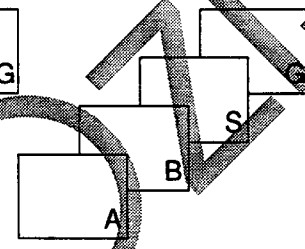
A = 0
B = 1
S = 1



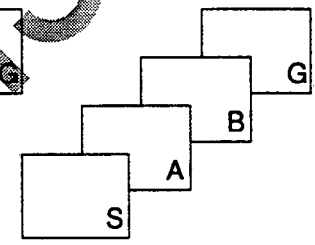
A = 1
B = 0
S = 0



A = 1
B = 0
S = 1



A = 1
B = 1
S = 0

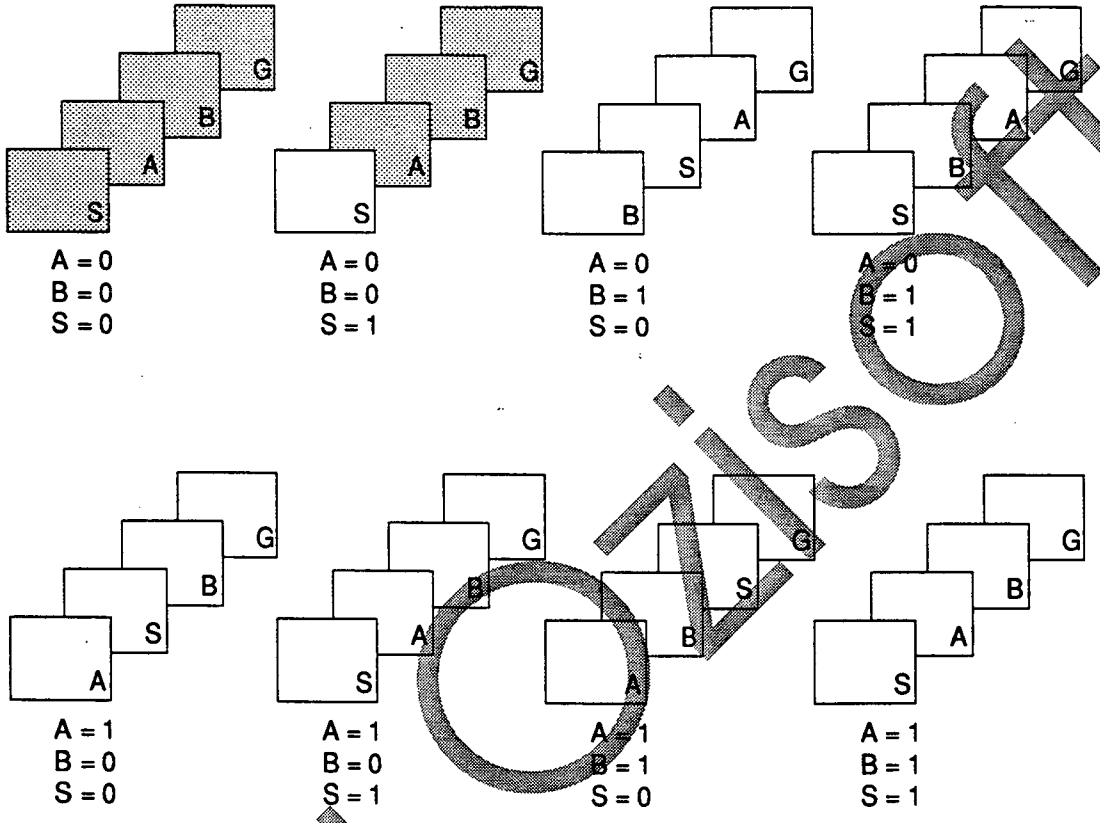


A = 1
B = 1
S = 1

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

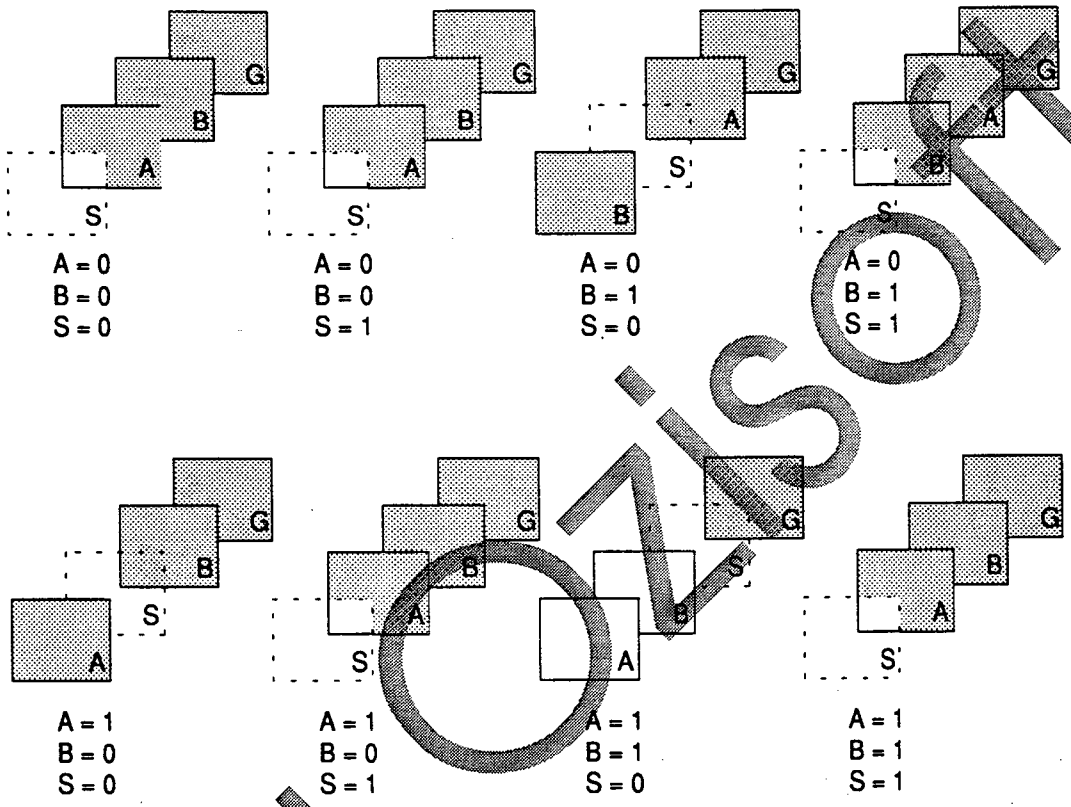


STATUS OF SHADOW

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



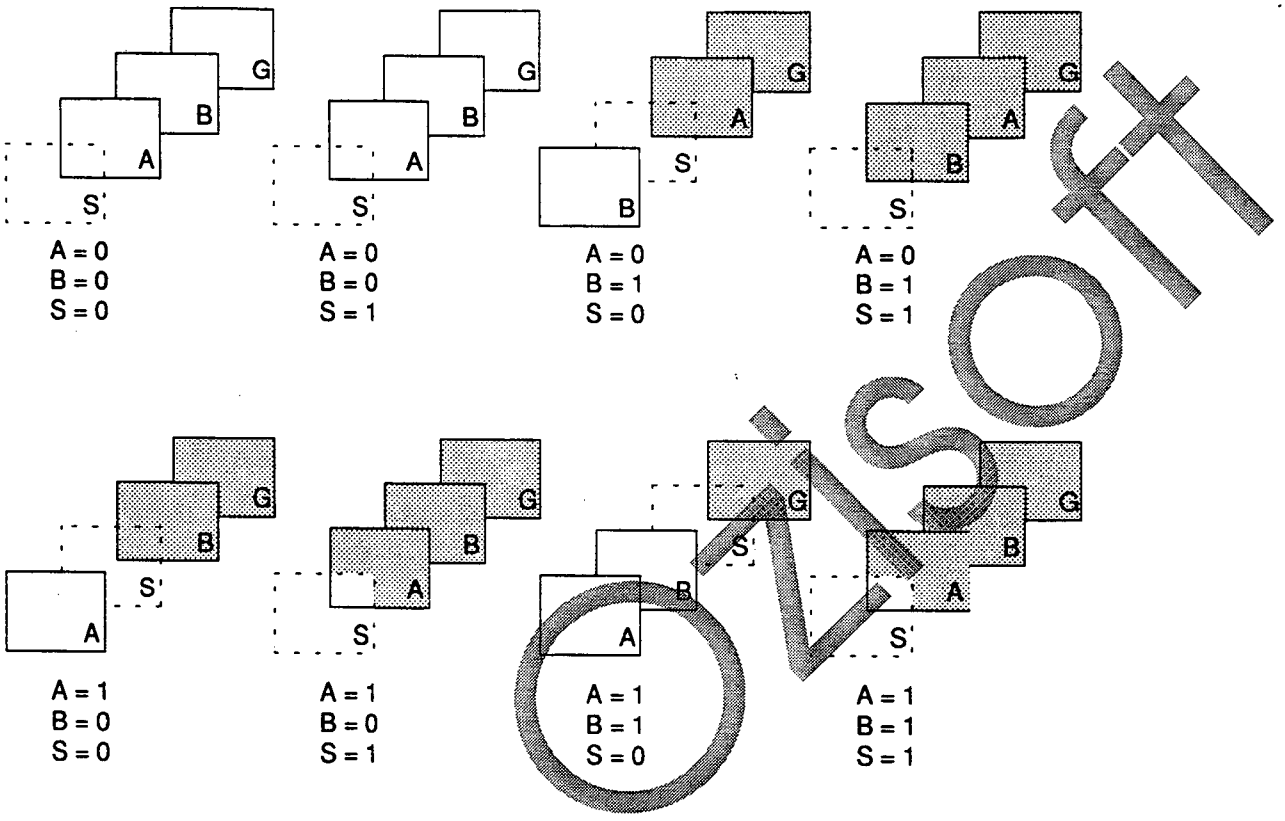
STATUS OF SHADOW

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



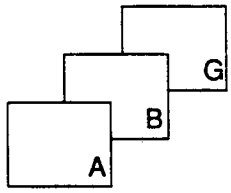
STATUS OF HIGHRIGHT

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

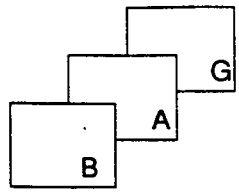
Since Sprite dots work as an operator, this will not be displayed.

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

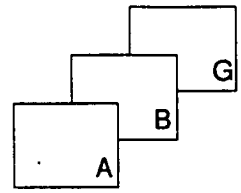
S/TEN = 0



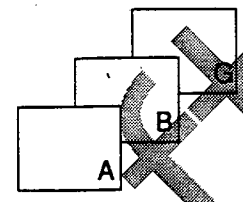
A = 0
B = 0



A = 0
B = 1

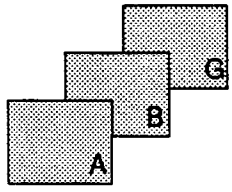


A = 1
B = 0

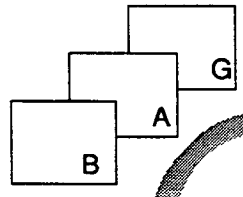


A = 1
B = 1

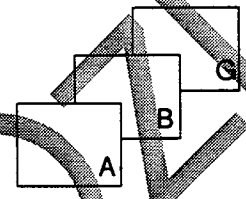
S/TEN = 1



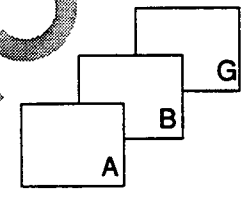
A = 0
B = 0



A = 0
B = 1



A = 1
B = 0

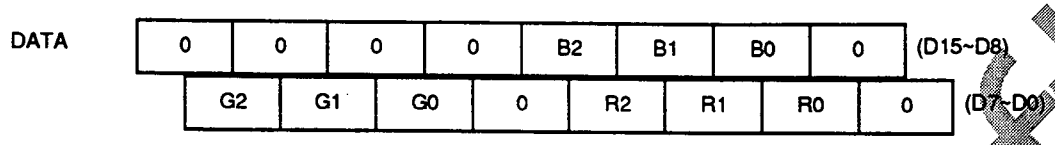


A = 0
B = 1

STATUS OF SHADOW

M. COLOR PALETTE

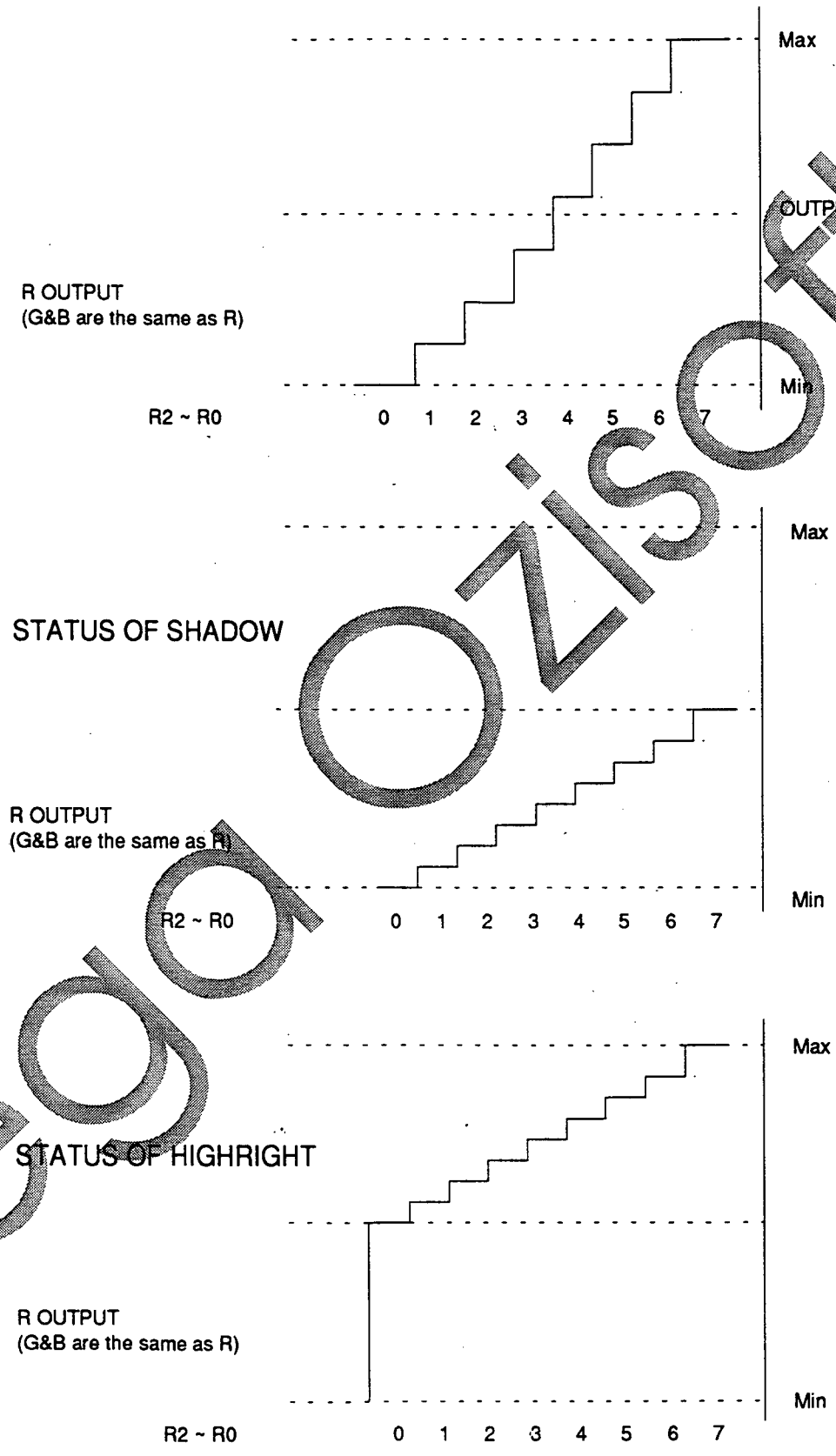
One dot is comprised of 4 bits and can designate the 0-15 colors. Also, 0-3 color palette can be designated by Scroll screen on a cell basis and by each Sprite. CRAM data are as follows. Since each of R, G and B has 3 bits, colors can be freely selected out of 512 colors.



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 RBG designation.

ADDRESS	BLUE	GREEN	RED	PALETTE	CODE	REMARKS
00H 02H 04H : : : 1AH 1CH 1EH				0	0 1 2 : : : 13 14 15	The 0~15 colors designated by RGB will be displayed
20H : 3EH				1	0 : 15	Same as Palette 0
40H : 5EH				2	0 : 15	Same as Palette 0
60H 62H : 7AH 7CH 7EH				3	0 1 : 13 14 15	Same as Palette 0 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).

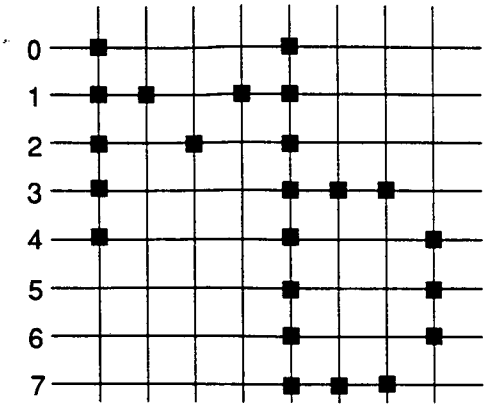
LSM1	LSM0	Raster Scan Mode
0	0	Non-interlace mode
0	1	In the non-interlace mode, the same pattern is displayed on the rasters of even and odd numbered files. (Interlace 1)
1	1	In the interlace mode, the different pattern is displayed on the rasters of even and odd numbered files. (Interlace 2)

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.

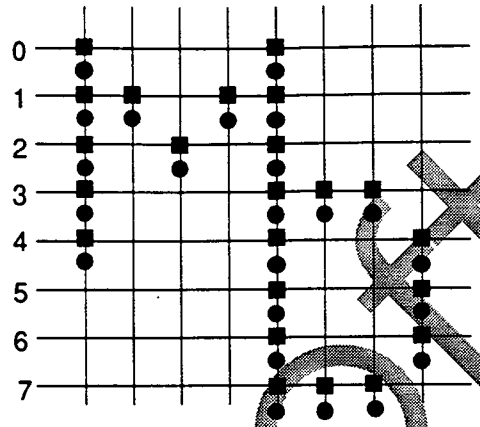
In any case, number of cells in one screen are the same.

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.

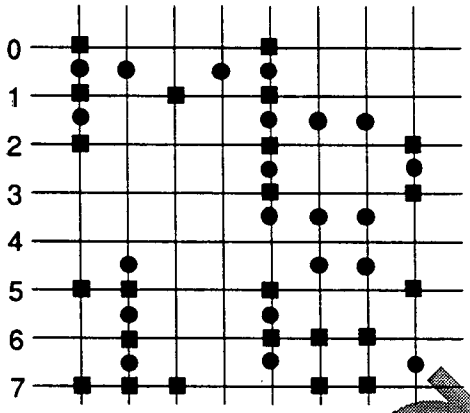
NON-INTERLACE



INTERLACE 1



INTERLACE 2



■ FIELD NO. 1

● FIELD NO. 2

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

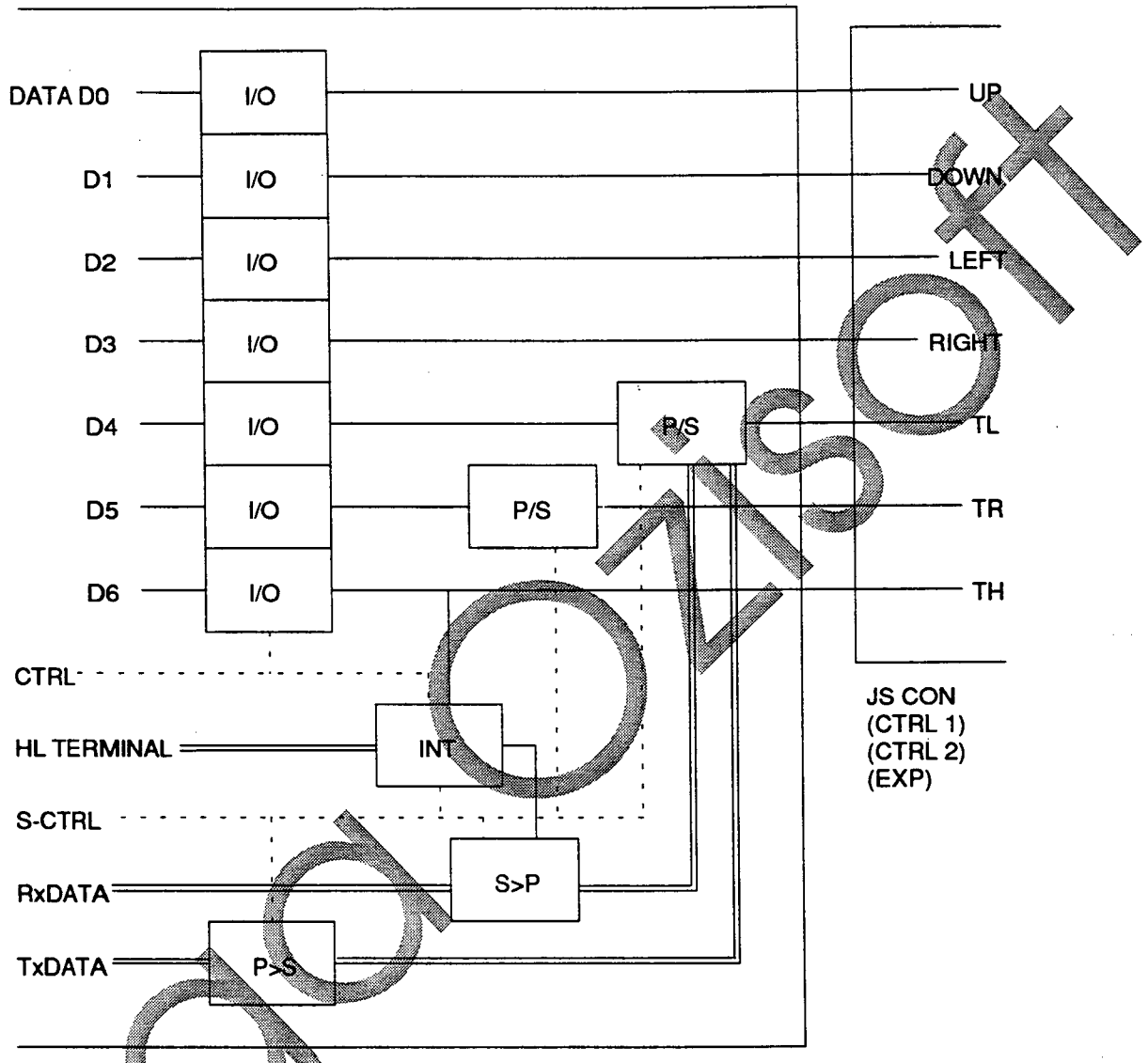
Mode (R) 0: Domestic Model
1: Overseas Model
VMOD (R) 0: NTSC CPU clock 7.67 MHz
1: PAL CPU clock 7.60 MHz
Disk (R) 0: FDD unit connected
1: FDD unit not connected
RSV (R) Currently not used
VER 3~0 (R) Mega Drive version is indicated by $0~$F.
The present hardware version is indicated by $0.

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

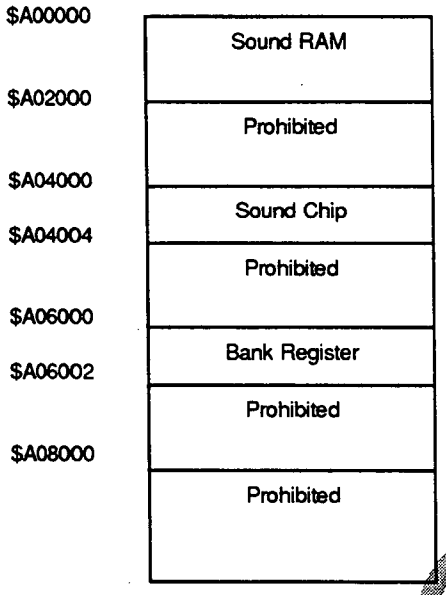
The 68000 may also reset the Z80. The Z80 is automatically reset during the Mega Drive hardware's power on reset sequence.

$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:



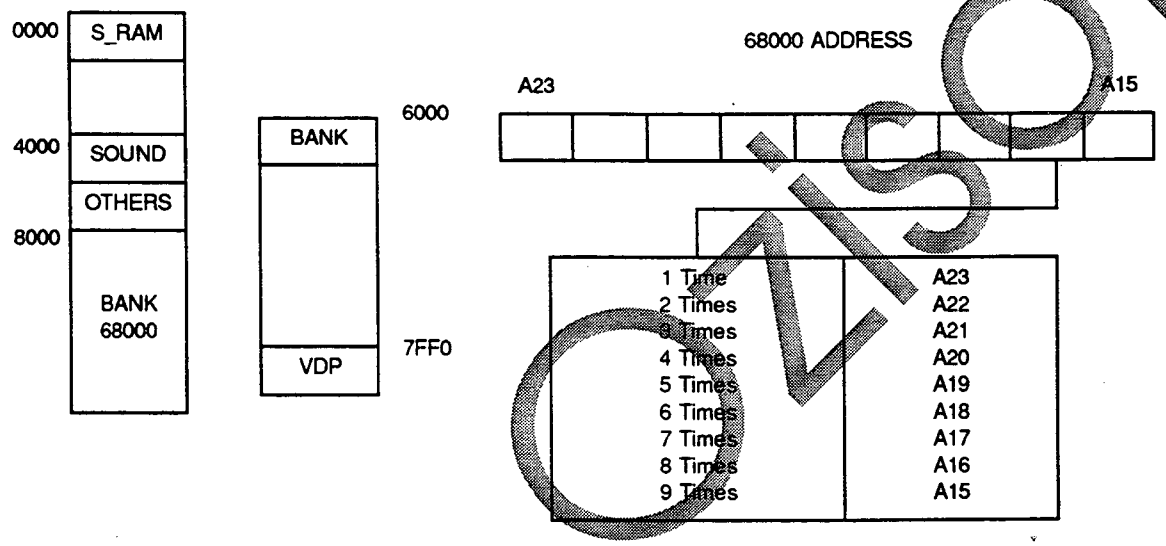
1. Sound RAM
   This is for the Z80 program. Access from 68000 by byte.

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).

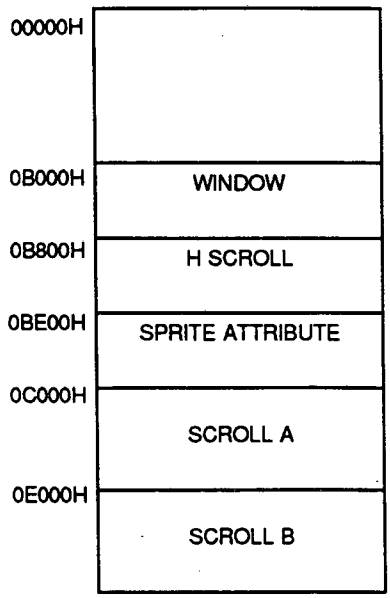
There are 1 K bytes for H Scroll Table, however, as for display 896 bytes in V28 Cell Mode and 960 bytes in V30 Cell Mode. There are 2 K bytes for Window Pattern Name Table in H32 Cell Mode, and 4 K byte area in H40 Cell Mode. For details, refer to Window. There are 512 bytes for Sprite Attribute Table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

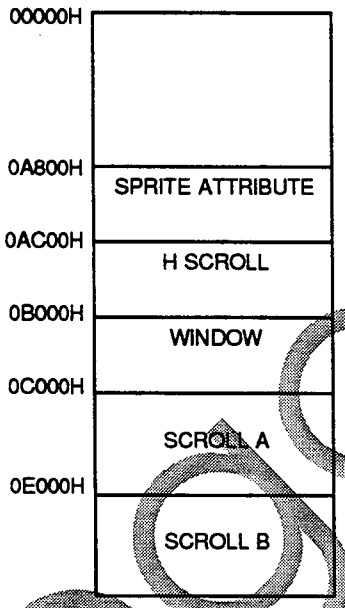
Unoccupied area is used as Pattern Generator and Sprite Generator.



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

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.

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
        ?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 isin 0FF0000 to 0FFFFFF

RAM Card (No. 171-5642-02)

This board has two memory areas:

Main Memory	(D-RAM)	$000000 - $0FFFFF
Backup Memory	(S-RAM)	$200000 - $203FFF

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.

GENESIS

SOUND SOFTWARE MANUAL

INDEX

I. Z80 MAPPING
- A. Z80 Memory Map
- B. Interrupt

II. 68K CONTROL OF Z-80
- A. Z80 Start-up
- B. Z80 Handshake

III. FM SOUND CONTROL
- A. 68K Access FM Chip
- B. Z80 Access FM Chip

IV. PSG CONTROL

V. D/A CONTROL

This manual explains memory mapping and way of accessing especially. FM sound generation and PSG are explained in another manual.

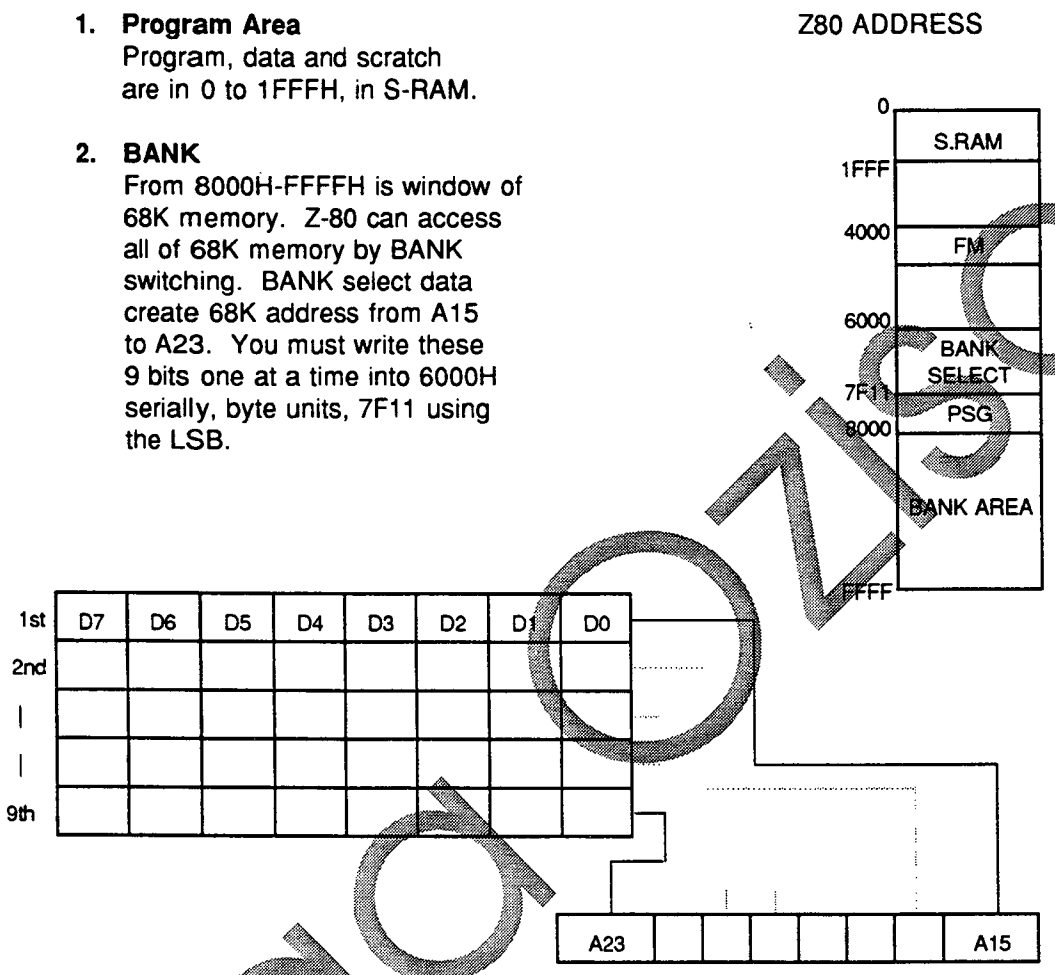
I. Z80 MAPPING

A. Z80 Map

We show the memory at right.
I/O is contained in memory map.

1. **Program Area**
   Program, data and scratch are in 0 to 1FFFH, in S-RAM.

2. **BANK**
   From 8000H-FFFFH is window of 68K memory. Z-80 can access all of 68K memory by BANK switching. BANK select data create 68K address from A15 to A23. You must write these 9 bits one at a time into 6000H serially, byte units, 7F11 using the LSB.



3. **I/O**
   4000H FM1 register select (Channel 1-3)
   4001H FM1 DATA

   4002H FM2 register select (Channel 4-6)
   4003H FM2 DATA

   PSG address is in 7F11H.

B. Interrupt

Z-80 gets the only VIDEO vertical interrupt. This interrupt is generated 16ms period and 64ms length.

II. 68K CONTROL OF Z-80

A. Z80 Start-Up

Z-80 Operation Sequence:

1. BUS REQ ON
2. BUS RESET OFF
3. 68K copies program into Z-80 S-RAM
4. BUS RESET ON
5. BUS REQ OFF
6. BUS RESET OFF

BUS REQUEST

- BUS REQ ON
  DATA 100H (WORD) → $A11100
- BUS REQ OFF
  DATA 0H (WORD) → $A11100

RESET Z-80

- RESET ON
  DATA 0H (WORD) → $A11200
- RESET OFF
  DATA 100H (WORD) → $A11200

  This period requires 26ms.
  Also FM sound source is cleared at the same time.

CONFIRMATION OF BUS STATUS

This information is in $A11100, bit 0.

0 - Z-80 is using
1 - 68K can access

B. Z80 Handshake
If you access the HANDSHAKE area (A00000 - A07FFF) you must use BUS REQ. 68K has to access the Z-80 S-RAM by byte.

III. FM SOUND CONTROL

A. 68K Accesses the FM Source
68K needs BUS REQ when accessing the FM source, because this memory is controlled by Z-80.

B. Z80 Accesses the FM Source
Z-80 normally controls the FM (4000H - 4003H).

IV. PSG CONTROL

PSG accepts access of 68K and Z-80 any time, but you have to coordinate 68K and Z-80 accesses.

PSG is in $C00011 from 68K and in 7F11H from Z-80.

OVERVIEW

The Yamaha 2612 Frequency Modulation (FM) sound synthesis IC resembles the Yamaha 2151 (used in SEGA's coin-operated machines) and the chips used in Yamaha's synthesizers.

Its capabilities include:

- 6 channels of FM sound
- An 8-bit Digitized Audio channel (as replacement for one of the FM channels)
- Stereo output capability
- One LFO (low frequency oscillator) to distort the FM sounds
- 2 timers, for use by software.

To define these terms more carefully, an FM channel is capable of expressing, with a high degree of realism, a single note in almost any instrument's voice. Chords are generally created by using multiple FM channels.

The standard FM channels each have a single overall frequency and data for how to turn this frequency into the complex final waveform (the voice). This conversion process uses four dedicated channel components called "operators," each possessing a frequency (a variant of the overall frequency), an envelope, and the capability to modulate its input using the frequency and envelope. The operator frequencies are offsets of integral multiples of the overall frequency.

There are two sets of three FM channels, named channels 1 to 3 and 4 to 6, respectively. Channels 3 and 6, the last in each set, have the capability to use a totally separate frequency for each operator rather than offsets of integral multiples. This works well (we believe) for percussion instruments, which have harmonics at odd multiples such as 1.4 o 1.7 of the fundamental.

The 8-bit Digitized Audio Channel (DAC) exists as a replacement of FM channel 6, meaning that turning on the DAC turns off FM channel 6. Unfortunately, all timing must be done by software — meaning that unless the software has been very cleverly constructed, it is impossible to use any of the FM channels at the same time as the DAC.

Stereo output capability means that any of the sounds, FM or DAC, may be directed to the left, the right, or both outputs. The stereo is output only through the headphone jack.

The LFO, or Low Frequency Oscillator, allows for amplitude and/or frequency distortions of the FM sounds. Each channel elects the degree to which it will be distorted by the LFO, if at all. This could be used, for example, in a guitar solo.

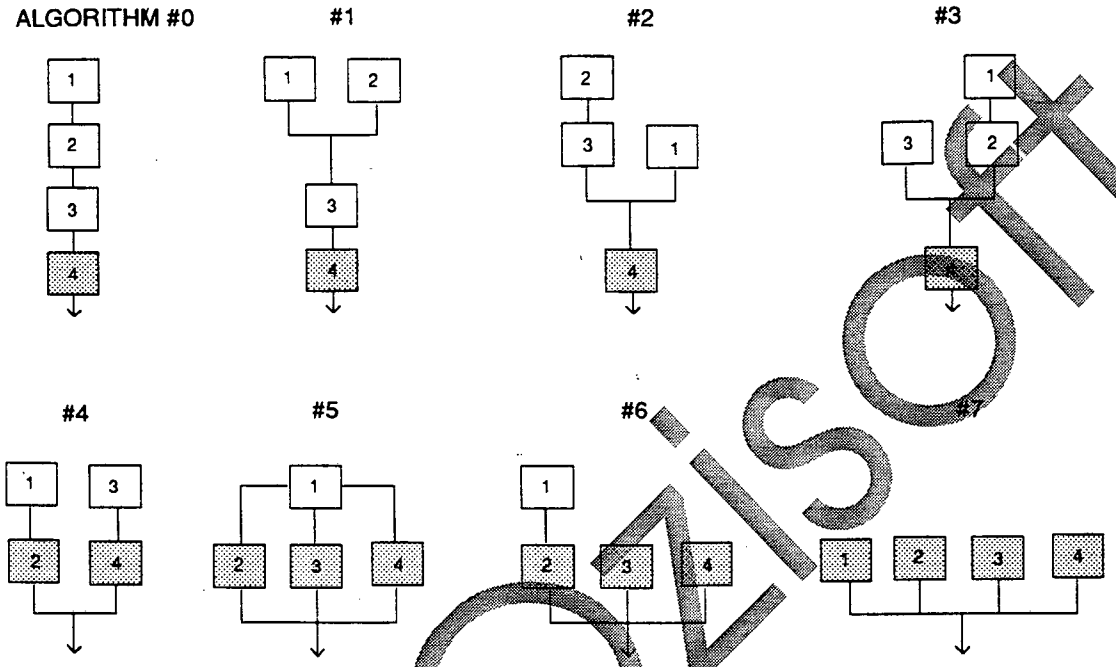
Finally, the system has two software timers which may be used as an alternative to the Z80 VBLANK interrupt. Unfortunately, these two timers do not cause interrupts — they must be read by the software to determine if they have finished counting.

A LITTLE BIT ABOUT OPERATORS

There are four dedicated operators assigned to every channel, with the following properties:

- An operator has an input, a frequency and an envelope (with which to modify the input), and an output.
- The operators have two types: those whose outputs feed into another operator, and those that are summed to form the final waveform. The latter are called "slots."
- The slots may be independently enabled, although Sega's software always enables or disables them all simultaneously.
- Operator one may feed back into itself, resulting in a more complex waveform.

These operators may be arranged in eight different configurations, called "algorithms." Following is a diagram of the algorithms.



SLOTS ARE INDICATED BY SHADING

Algorithm 0 - distortion guitar, "high hat chopper" (?) bass
Algorithm 1 - harp, PSG (Programmable Sound Generator) sound
Algorithm 2 - Bass, electric guitar, brass, piano, woods
Algorithm 3 - strings, folk guitar, chimes
Algorithm 4 - flute, bells, chorus, bass drum, snare drum, tom-tom
Algorithm 5 - brass, organ
Algorithm 6 - xylophone, tom-tom, organ, vibraphone, snare drum, base drum
Algorithm 7 - pipe organ

REGISTER OVERVIEW

The system is controlled by means of a large number of registers. General system registers are:

- timer values and status, software use
- LFO enable and frequency, to distort the FM channels
- DAC enable and amplitude
- output enables for each of the six FM channels
- number of frequencies to be used in FM channels 3 and 6.

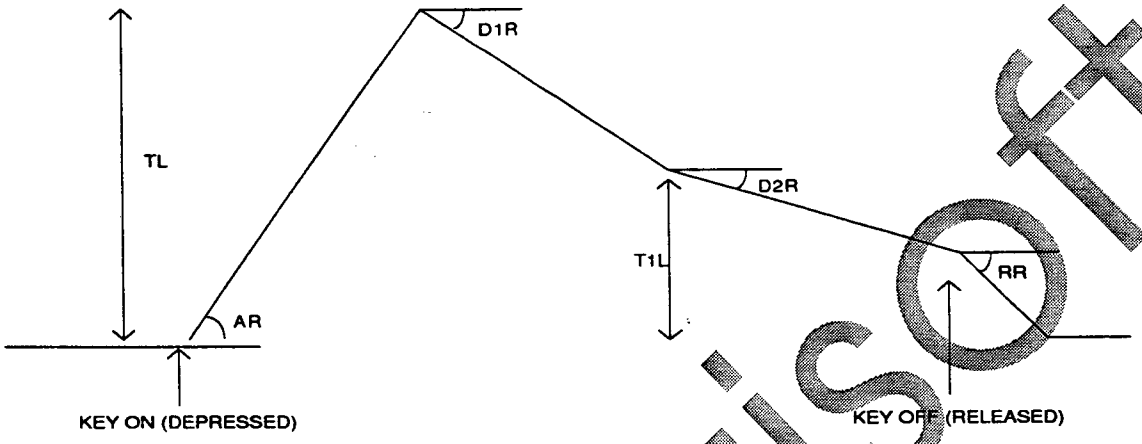
Usually, an FM channel has only one overall frequency, but if so elected, FM channels 3 and 6 use four separate frequencies, one for each operator.

The remainder of the registers apply to a single FM channel, or to an operator in that channel. Registers that refer to the channel as a whole are:

- frequency number (in the standard case)
- algorithm number
- extent of self-feedback in operator 1
- output type, to L, R, or both speakers. This can only be heard if headphones are used.
- the extent to which the channel is distorted by the LFO.

Registers that refer to each operator make up the remainder. The four operators' connections are determined by the algorithm used, but the envelope is always specified individually for each operator. In the case of FM channels 3 and 6, the frequency may be specified individually for each operator.

ENVELOPE SPECIFICATION



The sound starts when the key is depressed, a process called "key on." The sound has an attack, a strong primary decay, followed by a slow secondary decay. The sound continues this secondary decay until the key is released, a process called "key off." The sound then begins a rapid final decay, representing, for example, a piano note, after the key has been released and the damper has come down on the strings.

The envelope is represented by the above amplitudes and angles, and a few supplementary registers. Used in the above diagram are:

TL — Total level, the highest amplitude of waveform.

AR — Attack rate, the angle of initial amplitude increase. This can be made very steep if desired. The problem with slow attack rates is that if the notes are short, the release (called "key off") occurs before the note has reached a reasonable level.

DIR — The angle of initial amplitude decrease.

T1L — The amplitude at which the slower amplitude decrease starts.

D2R — The angle of secondary amplitude decrease. This will continue indefinitely unless "key off" occurs.

RR — The final angle of amplitude decrease, after "key off."

Additional registers are:

RS — Rate scaling, the degree to which envelopes become shorter as frequencies become higher. For example, high notes on a piano fade much more quickly than low notes.

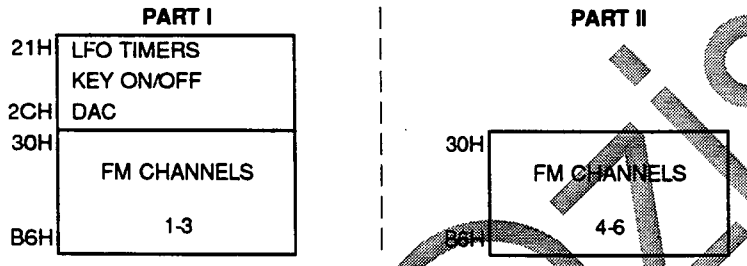
AM — Amplitude Modulation enable, whether or not this operator will allow itself to be modified by the LFO. Changing the amplitude of the slots changes the loudness of the note; changing the amplitude of the other operators changes its flavor.

SSG-EG — A proprietary register whose usage is unknown. It should be set to zero.

The FM-2612 may be accessed from either the 68000 or the Z-80. In both cases, however, the bus is only 8 bits wide.

The FM-2612 is accessed through memory locations 4000H - 4003H in the Z80 case, or A04000H - A04003H in the 68000 case. These will be referred to as 4000 to 4003.

The internal registers of the FM-2612 are divided as follows:



To write to Part I, write the 8-bit address to 4000 and the data to 4001. To write to Part II, write the 8-bit address to 4002 and the data to 4003.

Caution: Before writing, read from any address to determine if the YM-2612 I/O is still busy from the last write. Delay until Bit 7 returns to 0.

Caution: In the case of registers that are "ganged together" to form a longer number — for example the 10-bit Timer A value or the 14-bit frequencies — write the high register first.

READ DATA: Reading from any of the four locations.

BUSY — 1 if busy, 0 if ready for new data.
OVERFLOW — 1 if the timer has counted up and overflowed. See Register 27H.

PART I MEMORY MAP

22H	X	X	X	X	LFO EN	LFO FREQ		
24H	TIMER A							
25H	X	X	X	X	X	X	TIMER A	
26H	TIMER B							
	CH3		RESET		ENABLE		LOAD	
27H	MODE	B	A	B	A	B	A	
28H	OPERATOR			X	CHANNEL			
2AH	DAC							
2BH	DAC EN	X	X	X	X	X	X	X

30H+	X	DT1		MUL				
40H+	X	TL						
50H+	RS		X	AR				
60H+	AM	X		X	D1R			
70H+	X	X		X	D2R			
80H+	D1L				RR			
90H+	X	X		X	SSG-EG			

30H	CH1, OP1							
31H	CH2, OP1							
32H	CH3, OP1							
34H	CH1, OP2							
34H	CH2, OP2							
36H	CH3, OP2							
38H	CH1, OP3							
39H	CH2, OP3							
3AH	CH3, OP3							
3CH	CH1, OP4							
3DH	CH2, OP4							
3EH	CH3, OP4							

Each of 30H-90H has twelve entries, three channels x four operators.

Channels 1-3 become channels 4-6 in Part II.

PART I MEMORY MAP (cont.)

A0H+	FREQ. NUM							
A4H+	X	X	BLOCK	FREQ. NUM				
A8H+	CH 3 SUPPLEMENTARY FREQ. NUM							
ACH+	X	X	CH 3 SUPP BLOCK	CH3 SUPP FREQ NUM				
B0H+	X	X	FEEDBACK	ALGORITHM				
B4H+	L	R	AMS	X	FMS			

Each of the above has three entries. All follow the pattern

A0H	CH1
A1H	CH2
A2H	CH3

with the exception that A8H and ACH follow the pattern

A8H	CH3, OP2
A9H	CH3, OP3
AAH	CH3, OP4

"PART II" is a duplication of 30H-B4H, where channels 1-3 are replaced by 4-6.

The Registers:

22H	X	X	X	X	LFO EN	LFO FREQ	

LFO EN — 1 is enabled, 0 disabled.

LFO FREQ

	0	1	2	3	4	5	6	7
Hz	3.98	5.56	6.02	6.37	6.88	9.63	48.1	72.2

The LFO (Low Frequency Oscillator) is used to distort the FM sounds' amplitude and phase. It is triply enabled, as there is:

a. a global enable in Register 22H
b. a sensitivity enable on a channel by channel basis, in Registers B4H-B6H
c. an amplitude enable on an operator by operator basis in Registers 60H-6EH.

If the LFO is desired, enable it by Register 22H. Next, select which channels will be affected by the LFO, to what degree, and whether their amplitude or frequency is affected, by setting Registers B4H-B6H. Finally, if a channel's amplitude is affected, make sure that it is only the "slots" that are affected by setting Registers 60H-6EH.

24H	TIMER A MSBs						
25H	X	X	X	X	X	X	TIMER A LSBs

Registers 24H and 25H are ganged together to form 10-bit TIMER A, with Register 25H containing the least significant bits. They should be set in the order 24H, 25H. The timer lasts

18* (1024 - TIMER) microseconds

TIMER A = all 1's →18μs = 0.108 ms
TIMER A = all 0's →18,400μs = 18.4 ms

26H	TIMER B

8-bit TIMER B lasts

288 * (256 - TIMER B) microseconds

TIMER B = all 1's →0.288 ms
TIMER B = all 0's →73.44 ms

27H	CH3 MODE	RESET		ENABLE		LOAD	
		B	A	B	A	B	A

Register 27H controls the software timers and the Channel 3 (and 6) mode, two entirely separate items.

CH3 MODE	D7	D6	
NORMAL	0	0	Channel 3 is the same as the others.
SPECIAL	0	1	Channel 3 has four separate frequencies.
ILLEGAL	1	X	

A normal channel's operators use offsets of integral multiples of a single frequency. In SPECIAL mode, each operator has an entirely separate frequency. Channel 3 operator 1's frequency is in Registers A2 and A6. Operators 2 and 4 are in Registers A8 and AC, A9 and AD, and AA and AE, respectively.

No one at Sega has used the timer feature, but the Japanese manual says:

LOAD 1 starts the timer, 0 stops it.

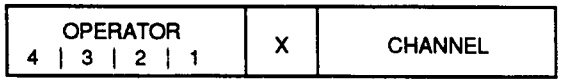
ENABLE 1 causes timer overflow to set the read register flag. 0 means the timer keeps cycling without setting the flag.

RESET writing a 1 clears the read register flag, writing a 0 has no effect.

28H	OPERATOR	X	CHANNEL

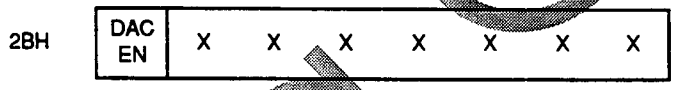
This register is used for "key on" and "key off." "Key on" is the depression of the synthesizer key. "Key off" is its release. The sequence of operations is: set parameters, key on, wait, key off. When key off occurs, the FM channel stops its slow decline and starts the rapid decline specified by "RR", the release rate.

In a single write to Register 28H, one sets the status of all operators for a single channel. Sega always sets them to the same value, on (1) or off (0). Using a special channel 3, I believe it is possible to have each operator be a separate note, so there is possible justification for turning them on and off separately.



D2	D1	D0	
0	0	0	Channel 1
0	0	1	2
0	1	0	3
1	0	0	Channel 4
1	0	1	5
1	1	0	6

Register 2AH contains 8 bit DAC data.



If the DAC enable is 1, the DAC data is output as a replacement for channel 6. The only channel 6 register that affects the DAC is the stereo output portion of Register B4H.

Registers 30H-90H are all single-operator registers. Please see page 8 for how the twelve channel-operator combinations are arranged.

Both DT1 (Detune) and MUL (multiple) relate the operator's frequency to the overall frequency.

MUL ranges from 0 to 15_{10}, and multiplies the overall frequency, with the exception that 0 results in multiplication by 1/2. That is, MUL = 0 to 15 gives x 1/2, x 1, x 2, ... x 15.

DT1 gives small variations from the overall frequency x MUL. The MSB of DT1 is a primitive sign bit, and the two LSBs are magnitude bits. See the next page for a diagram.

D6	D5	D4	MULTIPLICATIVE EFFECT
0	0	0	No change
0	0	1	x (1 + E)
0	1	0	x (1 + 2E)
0	1	1	x (1 + 3E)
1	0	0	No change
1	0	1	x (1 - E)
1	1	0	x (1 - 2E)
1	1	1	x (1 - 3E)

where E is a small number

40H+	X	TL

TL (total level) represents the envelope's highest amplitude, with 0 being the largest and 127_{10} the smallest. A change of one unit is about 0.75 dB.

To make a note softer, only change the TL of the slots (the output operators). Changing the other operators will affect the flavor of the note.

50H+	RS	X	AR

Register 50H contains RS (rate scaling) and AR (attack rate). AR is the steepness of the initial amplitude rise, shown on page 4.

RS affects AR, D1R, D2R and RR in the same way. RS is the degree to which the envelope becomes narrower as the frequency becomes high.

The frequency's top five bits (3 octave bits and 2 note bits) are called KC (key code) in the following rate formulas:

RS=0 ⇒ Final Rate = 2 * Rate + (KC/8)
RS=1 ⇒ Final Rate = 2 * Rate + (KC/4)
RS=2 ⇒ Final Rate = 2 * Rate + (KC/2)
RS=3 ⇒ Final Rate = 2 * Rate + KC**

** Always rounded down.

As rate ranges from 0-31, this means that the RS influence ranges from small (at 0-3) to very large (at 0-31).

60H+	AM	X	X	D1R

D1R (first decay rate) is the initial step amplitude decay rate (see page 4). It is, like all rates, 0-31 in value and affected by RS.

AM is the amplitude modulation enable, whether or not this operator will be subject to amplitude modulation by the LFO. This bit is not relevant unless both the LFO is enabled and Register B4's AMS (amplitude modulation sensitivity) is non-zero.

70H+	X	X	X	D2R

D2R (secondary decay rate) is the long tailoff of the sound that continues as long as the key is depressed.

90H+	X	X	X	X	SSG-EG

This register is proprietary and should be set to zero.

The final registers relate mostly to a single channel. Each register is tripled; please see the diagram on page 9.

A0H+	FREQ NUM			
A4H+	X	X	BLOCK	FREQ NUM
A8H+	CH 3 SUPP FREQ NUM			
ACH+	X	X	CH 3 SUPP BLOCK	CH 3 SUPP FREQ NUM

Channel 1's frequency is in A0 and A4H.
Channel 2's frequency is in A1 and A5H.
Channel 3, if it is in normal mode (please see page 12) is in A2 and A6H.

If channel 3 is in special mode:
Operator 1's frequency is in A2 and A6H
Operator 2's frequency is in A8 and ACH
Operator 3's frequency is in A9 and ADH
Operator 4's frequency is in AA and AEH.

The frequency is a 14-bit number that should be set high byte, low byte (e.g., A4H then A0H). The highest 3 bits, called the "block," give the octave. The next 10 bits give position in the octave, and a possible 12-tone sequence is:

Low	617
	653
	692
	733
	777all in base 10
	823
	872
	924
	979
	1037
	1099
High	1164

This sequence should be used inside each octave.

B0H+	X	X	FEEDBACK	ALGORITHM

Feedback is the degree to which operator 1 feeds back into itself. In the voice library, self feedback is represented as this:

The ALGORITHM is the type of inter-operator connection used. Please see the list of the eight operators on page 3.

B4H+	L	R	AMX	X	FMS

Register B4H contains stereo output control and LFO sensitivity control.
L — Left output, 1 is on, 0 is off
R — Right output, 1 is on 0 is off
Note: The stereo may only be heard by headphones.

AMS (amplitude modulation sensitivity) and FMS (frequency modulation sensitivity) are the degree to which the channel is affected by the LFO. If the LFO is disabled, this register need not be set. Additionally, amplitude modulation is also enabled on an operator-by-operator level.

AMS	0	1	2	3
dB	0	1.4	5.9	11.8

FMS	0	1	2	3	4	5	6	7
% of a halftone	0	± 3.4	± 6.7	± 10	± 14	± 20	± 40	± 80

TEST PROGRAM

Here a tested power-on initialization and sample note in the "Grand Piano" voice (page 27).

Register	Value	Comments
22H	0	LFO off
27H	0	Channel 3 mode normal
28H	0	Off
28H	1	Off
28H	2	Off
28H	4	Off
28H	5	Off
28H	6	Off
2BH	0	DAC off
30H	71H	DT1/MUL
34H	0DH	
38H	33H	
3CH	01H	
40H	23H	Total Level
44H	2DH	
48H	26H	
4CH	00H	
50H	5FH	RS/AR
54H	99H	
58H	5FH	
5CH	94H	
60H	5	AM/D1R
64H	5	
68H	5	
6CH	7	
70H	2	D2R
74H	2	
78H	2	
7CH	2	

Register	Value	Comments
80H	11H	D1L/RR
84H	11H	
88H	11H	
8CH	A6H	
90H	0	Proprietary
94H	0	
98H	0	
9CH	0	
B0H	32H	Feedback/Algorithm
B4H	C0H	Both speakers on
28H	00H	Key off
A4H	22H	Set frequency
A0H	69H	
28H	F0H	Key on
<wait>		
28H	00H	Key off

Notes:
1. Write address then data.
2. Loop until read register D7 becomes 0.
3. Follow MSB/LSB sequence.

Programmable Sound Generator (PSG)

The PSG contains four sound channels, consisting of three tone generators and a noise generator. Each of the four channels has an independent volume control (attenuator). The PSG is controlled through output port $7F.

Tone Generator Frequency

The frequency (pitch) of a tone generator is set by a 10-bit value. This value is counted down until it reaches zero, at which time the tone output toggles and the 10-bit value is reloaded into the counter. Thus, higher 10-bit numbers produce lower frequencies.

To load a new frequency value into one of the tone generators, you write a pair of bytes to I/O location $7F according to the following format:

First Byte:	1	R2	R1	R0	d3	d2	d1	d0
Second Byte:	0	0	d9	d8	d7	d6	d5	d4

The R2:R1:R0 field selects the tone channel as follows:

R2	R1	R0	Tone Channel
0	0	0	#1
0	1	0	#2
1	0	0	#3

10-bit data is: (msb) d9 d8 d7 d6 d5 d4 d3 d2 d1 d0 (lsb)

Noise Generator Control

The noise generator uses three control bits to select the "character" of the noise sound. A bit called "FB" (Feedback) produces periodic noises or "white" noise:

FB	Noise Type
0	Periodic (like low-frequency tone)
1	White (hiss)

The frequency of the noise is selected by two bits NF1:NF0 according to the following table:

NF1	NF0	Noise Generator Clock Source
0	0	Clock/2 (higher pitch, "less coarse")
0	1	Clock/4
1	0	Clock/8 (lower pitch, "more coarse")
1	1	Tone Generator #3

Note: "Clock" is fixed in frequency. It is a crystal controlled oscillator signal connected to the PSG.

When NF1:NF0 is 11, Tone Generator #3 supplies the noise clock source. This allows the noise to be "swept" in frequency. This effect might be used for a jet engine runup, for example.

To load these noise generator control bits, write the following byte to I/O port $7F:

Out ($7F):

1	1	1	0	0	FB	NF1	NF0

Attenuators

Four noise attenuators adjust the volume of the three tone generators and the noise channel. Four bits A3:A2:A1:A0 control the attenutation as follows:

A3	A2	A1	A0	Attenuation
0	0	0	0	0 db (maximum volume)
0	0	0	1	2 db Note: a higher attenuation results in a quieter sound
0	0	1	0	4 db
0	0	1	1	6 db
0	1	0	0	8 db
0	1	0	1	10 db
0	1	1	0	12 db
0	1	1	1	14 db
1	0	0	0	16 db
1	0	0	1	18 db
1	0	1	0	20 db
1	0	1	1	22 db
1	1	0	0	24 db
1	1	0	1	26 db
1	1	1	0	28 db
1	1	1	1	-Off-

The attenuators are set for the four channels by writing the following bytes to I/O location $7F:

Tone Generator #1:	1	0	0	1	A3	A2	A1	A0
Tone Generator #2:	1	0	1	1	A3	A2	A1	A0
Tone Generator #3:	1	1	0	1	A3	A2	A1	A0
Noise Generator:	1	1	1	1	A3	A2	A1	A0

EXAMPLE

When the Mk3 is powered on, the following code is executed:

```
LD  HL,CLRTB      ; clear table
LD  C,PSG_PRT     ; psg port is $7F
LD  B,4           ; load four bytes
OTIR
(etc.)
.
.
CLRTB  defb $9F, $BF, $DF, $FF
```

This code turns the four sound channels off. It's a good idea to also execute this code when the PAUSE button is pressed, so that the sound does not stay on continuously for the pause interval.

GENESIS SOFTWARE DEVELOPMENT MANUAL

<<COMPLEMENT>>

9. JUL 1991 REVISION

SEGA ENTERPRISES, LTD.

VER2.0

CONTENTS

1. REQUIRED ITEMS TO SET UP THE GENESIS ... 1

§ 1 CARTRIDGE 1

§ 2 INITIAL PROGRAM 2

§ 3 INDICATION OF EXTERNAL RAM INFORMATION .. 3

§ 4 INDICATION OF ID MODEM INFORMATION 4

§ 5 CHECK SUM 5

2. VDP SETUP DATA ... 6

3. SETTINGS TO PRESERVE COMPATIBILITY DURING GENESIS DEVELOPMENT ... 7

4. 68000 COMMANDS AND PERIPHERAL DEVICES ... 8

§ 1 VDP AND 68000 COMMANDS 8

§ 2 JOYSTICK AND PERIPHERAL DEVICES 9
◇ ID codes ◇ 9
◇ Periphral devices ◇ 10
◇ JOY PAD ◇ 12

5. PRECAUTIONS FOR SOFTWARE DEVELOPMENT ... 14

§ 1 Z80 CANNOT PROPERLY ACCESS THE BUS ON THE 68000 ... 14

§ 2 H_INT AND V_INT ARE USED IN GENESIS PROGRAMS ... 15

§ 3 TROUBLES DUE TO INTERRUPTS DURING COMMUNICATION ... 16

§ 4 PRECAUTIONS WHEN THE CONTROL A AND B SCREENS ARE USED ... 17

§ 5 OTHER PRECAUTIONS 17
●Precautions for the main program and Z80 bus requests ... 17
●Precautions for sound access 18
●When the reset button is pressed repeatedly . 18
●Notes on the control pad and read-programs .. 19

6. NOTES ON MEGA DRIVE EUROPE VERSION ... 20

§ 1 PAL 20
●PAL 20
●Differences in development of MEGA DRIVE software and precautions. ... 20

§ 2 VERTICAL 30-CELL MODE 21

7. NOTES ON THE BACKUP RAMs ... 22

1. REQUIRED ITEMS TO SET UP THE GENESIS

§ 1 CARTRIDGE

The following data (hereafter ID) must always be entered from the address 100H.

100H	'SEGA MEGA DRIVE'	: 1
110H	'(C)SEGA 1991.MAR'	: 2
120H	'GAME NAME FOR DOMESTIC USE'	: 3
150H	'GAME NAME FOR OVERSEAS USE'	: 4
180H	'GM XXXXXXXX-XX'	: 5
18EH	$XXXX	: 6
190H	CONTROL DATA	: 7
1A0H	$000000,$XXXXXX	: 8
1A8H	$FF0000,$FFFFFF	: 9
1B0H	EXTERNAL RAM INFORMATION	:10
1BCH	MODEM INFORMATION	:11
1C8H	INHIBITED TO BE USED	:12
1F0H	COUNTRY NAME	:13

●Descriptions on the above items

1. Hardware name : 'SEGA MEGA DRIVE' or 'SEGA GENESIS'

2. A company name of four characters long or company code is entered after (C).
   (C) SEGA when made or ordered by SEGA.
   (C) T-XX(company code) when made by a third party.

3. A domestic game name is entered (shifted JIS kanji character codes can be used; however, the name should be entered with as many ASCII characters as possible).

4. A overseas game name is entered (shifted JIS kanji character codes can be used; however, the name should be entered with as many ASCII characters as possible).

5. Type of the cartridge, product number and version number.
   Cartridge type : GM for game
   AI for education
   Product number : Unique number for each game
   Version : This number should be increased when the product is upgraded.

6. Check sum (see 'check sum')

7. Information on supports for I/O

MARK III Joy Stick	: 0
GENESIS Joy Stick	: J
Keyboard	: K
Serial (RS232C)	: R
Printer	: P
Tablet	: T
Track ball	: B
Paddle controller	: V
FDD	: F
CDROM	: C
Analog Joy Stick	: A

8. ROM capacity : Start address,End address
**Example

	Start address	End address
2M :	$000000	$03FFFF
4M :	$000000	$07FFFF
8M :	$000000	$0FFFFF

9. RAM capacity (fixed) from $FF0000 to $FFFFFF

10. External RAM information (see below)

11. Modem information (see below)

12. Inhibited to be used (filled with spaces)

13. Country name
    Japan : J
    USA : U
    Europe : E

※Note※ Blank addresses must always be filled with spaces.

- The data must be entered carefully, since the ID data will be used when the product is checked.
- The ID data and the initial program must always be input during the check sample process (see '2 Initial program').

§ 2 INITIAL PROGRAM

The initial program (see ICD_BLK4.PRG, in the sample program disc) must always be stored in every product, in order to set every product to the same status when the GENESIS is powered on, and to execute the hardware security process and other required processes which must be performed at the resetting.
This program must always be input as it is, starting with program start. Remenber that any product cannot be released, if its initial program is modified, or if it is not input starting with program start.

§ 3 INDICATION OF EXTERNAL RAM INFORMATION

```
1B0H :  dc.b    'RA',%1x1yz000,%abcdefgh
1B4H :  dc.l    Start address,End address
```

- x : Data preservation
  1 ... Non-volatile ... Backup, EEPROM, etc. where data will not be destroyed even if the power is turned off.
  0 ... Volatile ..... Data will be lost when the power is turned off.

- yz : Data size
  10 .... Even byte (D15 to D8)
  11 .... Odd byte (D7 to D0)
  00 .... Word
  01 .... Others (EEPROMs which are serially accessed, RAMs with 4-bit data bus, etc.)

- abc : Device type
  000
  001 ... SRAM
  010 ... EEPROM
  011
  100
  101
  110
  111

- defgh : Spare which is filled with 0s.

●Start addrss,End address
Addresses where RAMs are installed or areas of the control ports.

●Indication examples●

○ 8k byte backup RAM cartridge

```
+
+  SRAM backup memory
+  Odd bytes only
+  8k byte capacity
+  Addresses $200001,$203FFF
+

   dc.b    'RA',%11111000,%00100000
   dc.l    $200001,$203FFF
```

○ EEPROM cartridge (for modem)

Since the EEPROM for modem is accessed serially, the data size fits into 'others.' Its data preservation fits into '1' since data in EEPROMs will not be destroyed when the power is turned off. The address is set to $200001 for the control ports.

```
+
+  EEPROM memory
+  Serial access
+  1k bit capacity (128 × 8)
+  Addresses $200001,$200001
+

   dc.b    'RA',%11101000,%01000000
   dc.l    $200001,$200001
```

§ 4 INDICATION OF ID MODEM INFORMATION

1BCH: 'MO','company name','xx,y','zz'

- Company name : Same as the company name or code which is entered from 110H.
- xx : Game number (if this number is the same, communication can be performed)
  - 00 TELTEL stadium
  - 01 TELTEL majong
  - 02 MEGA answer
  - 03 SEGA NET GAME
  - ⋮
- y : Version No.
- zz : Modem information
  - 00 Japan only : No microphone installed
  - 10 Japan only : Microphone installed
  - 20 Overseas only : No microphone installed
  - 30 Overseas only : Microphone installed
  - 40 Common to Japan and overseas : No microphone installed
  - 50 Common to Japan and overseas : Microphone installed
  - 60 No microphone installed for Japan; microphone installed for overseas
  - 70 Microphone installed for Japan; no microphone installed for overseas
  - 80
  - 90

●Indication examples●

① If SEGA released a base ball game 'abcde' with a microphone, then the modem information is like this.

1BCH : 'MO','SEGA','00,0','10'

② After that, TOTO (third party with a company code of T-01) released the software 'xyz' with a microphone for Japan use only, which can communicate with '① abcde'. Then the modem information is like this:

1BCH : 'MO','T-01','00,1','10'

xx is common, but the version number in y is changed.
Since xx is common, the information indicates that ① and ② can communicate each other.

Note that the game number and the version number are managed by SEGA of JAPAN.

§ 5 CHECK SUM

The program to examine the check sum is given below. The program starts at 0FF8000H in the RAM space.
First fill the space to be used for this game with -1 (0FFH), and then load all the programs. After this, load this program (including the check sum program), and start it at 0FF8000H. Wait a while and stop the program. The check sum value is stored in the lower word of data register 0 (d0). It is a good idea to release the break points in the memory.

The space to be used for the game must also be filled with -1 (0FFH) before the programs are stored in the ROM.

```
end_addr        equ     $1a4
        org     -$8000

start:
                lea     end_addr,a0
                move.l  (a0),d1
                addq.l  #$1,d1
                movea.l #$200,a0
                sub.l   a0,d1
                asr.l   #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
        ?1e:
                nop
                nop
                bra.b   ?1e
```

2. VDP SETUP DATA

The followings must always be fixed.

(40-cell mode and NTSC mode)

R0	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	0	0	0				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								

☆

☆ The DMA registers should be set in such a way that only the actual memory is accessed. Don't attempt to use the techniques taking advantage of image, etc.

3. SETTINGS TO PRESERVE COMPATIBILITY DURING GENESIS DEVELOPMENT

・I1 mode on the 68000 side (also applies to ER-ICE, 178 and 378)

① Cartridge in use

```
MA 0000000,0FFFFFF=NO  (No Memory is assigned for all areas)
MA 0000000,007FFFF=US  (Only required portion of the DRAM area is assigned)
MA 0A00000,0A01FFF=US  (Z80 S-RAM area)
MA 0A04000,0A04FFF=US  (FM sound source area)
MA 0A10000,0A11FFF=US  (I/O and Z80 control)
MA 0C00000,0C00FFF=US  (VDP and sound control)
MA 0FF0000,0FFFFFF=US  (work RAM area)
```

【Note】 When the write protect function is used on the DRAM board, the following steps need to be conducted.

```
STEP    1. MA 0A13000,0A13FFF=US
        2. E 0A130F0=XXXX        ( Setting up or releasing)
        3. MA 0A13000,0A13FFF=NO
```

It is easier to defined a macro by conducting the above steps.

・I2 mode on the Z80 side (also applies to 178, 278 and 378)

```
MA 00000,0FFFF=NO  (No memory is assigned for all area)
MA 00000,01FFF=US  (Z80 SRAM area)
MA 04000,043FF=US  (YM2612 area)
MA 06000,063FF=US  (bank register area)
MA 07C00,07FFF=US  (PSG on the VDP side)
MA 08000,0FFFF=US  (68000 memory area)
```

4. 68000 COMMANDS AND PERIPHERAL DEVICES

§ 1 VDP AND 68000 COMMANDS

The TAS command is not supported on the GENESIS. The TAS command serves to test and set up the byte operands which are assigned by the effective address fields. If this command is issued, the test is executed and completed, but the setting will be ignored and not be written.

When VRAM access is made, reading in the write mode or writing in the read mode cannot be executed.

- If and attempt is made to read in the write mode when VRAM access is performed, the VDP suspends the 68000 (the VDP will not return DTACK to the 68000).
- If an attempt is made to write in the read mode when VRAM access is performed, the address for the VRAM access is only increased, but no data is written in the VRAM.

For these reasons, the following commands cannot be used to access the VRAM.

Example: CLR command
The CLR command attempts to write 0 after reading the target memory. The VDP suspends the 68000, since the read access occurs when the write access is made to the VRAM.

** otheres **
CLR, NBCD, NEG, NEGX, NOT, Scc, TAS, (single operand commands),
BCHG, BSET, BCLR,)bit handling commands)
ASR, ASL, LSR, LSL, ROR, ROL, ROXR, ROXL (shift and rotate commands), ADDI, ADDQ, ANDI, CMPI, EORI, ORI, SUBI, SUBQ (immediate commands)
(The above commands cannot be used to access the VRAM, since read and write operations are made to a single destination in these commands).

§ 2 JOYSTICK AND PERIPHERAL DEVICES

In addition to the joy stick, various peripheral devices can be connected to the I/O port. Since such devices have their own ID codes, the CPU can judge which devices are connected to the I/O port by examining their ID codes. However, remember that some peripheral devices of the MARK III & MS cannot be known by this method.

◇ ID codes ◇

The ID code is expressed by the OR sum logic of DATA 1, 2 and 3, PD3, PD2, PD1 and PD0 which correspond to the external pins, CTRL1, CTRL2 and EXT in the I/O port. The OR sum value is expressed in the following way by data when 1 is output and by data when 0 is output, provided that the TH pin is set in the output mode.

```
ID3 = (PD6 = 1) AND (PD3 OR PD2)
ID2 = (PD6 = 1) AND (PD1 OR PD0)
ID1 = (PD6 = 0) AND (PD3 OR PD2)
ID0 = (PD6 = 0) AND (PD1 OR PD0)
```

The relationships between the ID code and the peripheral device are given below.

Device name	ID3	ID2	ID1	ID0	
Old joy stick (2TRIG)	1	1	1	1	($F)
Not defined	1	1	1	0	($E)
New joy stick (3TRIG)	1	1	0	1	($D)
SEGA RESERVED	1	1	0	0	($C)
Not defined	1	0	1	1	($B)
SEGA RESERVED	1	0	1	0	($A)
Not defined	1	0	0	1	($9)
Not defined	1	0	0	0	($8)
Not defined	0	1	1	1	($7)
SEGA RESERVED	0	1	1	0	($6)
Not defined	0	1	0	1	($5)
Not defined	0	1	0	0	($4)
Not defined	0	0	1	1	($3)
Not defined	0	0	1	0	($2)
SEGA RESERVED	0	0	0	1	($1)
	0	0	0	0	($0)

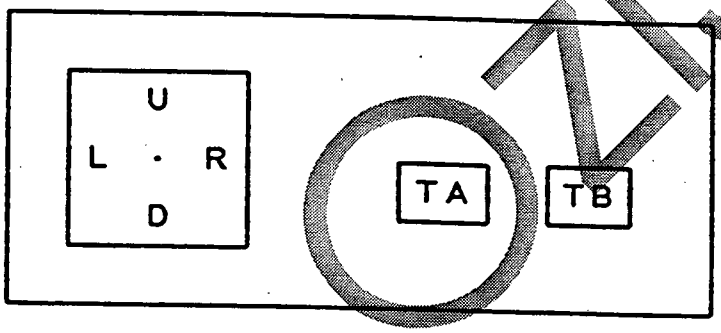
◇ Periphral devices ◇

● ID = $F : Old joy stick

The standard joy stick of MARK III & MS contains the 4-direction switch and A and B triggers. Each switch is configured as shown below, which is set to 0 when it is pressed.

	CTRL1	CTRL2	EXT.
CTRL	$A10009	$A1000B	$A1000D
DATA	$A10003	$A10005	$A10007

	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
CTRL	0	0	0	0	0	0	0	0
DATA	*	*	TB	TA	R	L	D	U

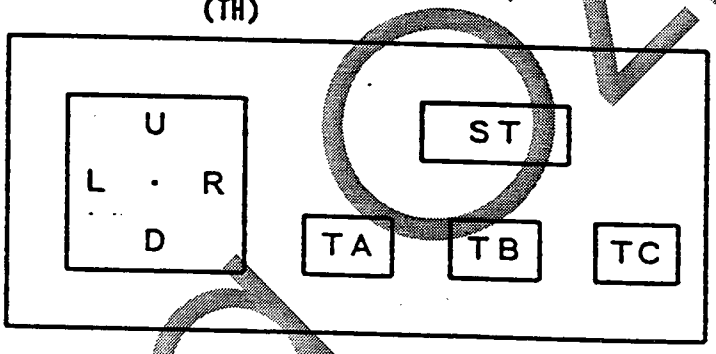


- ID = $D : New joy stick

The joy stick of the GENESIS contains the 4-direction switch, A, B and C triggers and Start.
The mode can be changed by setting 0 or 1 after the TH pin is set in the output mode.
Each switch is configured as shown below, which is set to 0 when it is pressed.

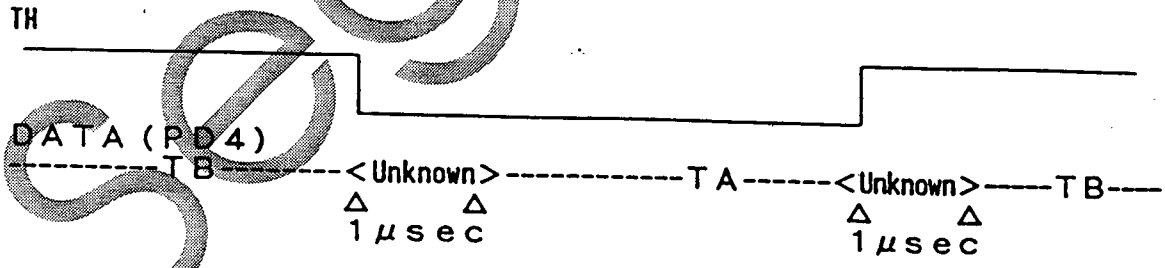
	CTRL1	CTRL2	EXT.
CTRL	$A10009	$A1000B	$A1000D
DATA	$A10003	$A10005	$A10007

		PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
	CTRL	0	1	0	0	0	0	0	0
TH=1	DATA	*	1	TC	TB	R	L	D	U
TH=0	DATA	*	0	ST	TA	0	0	D	U



It takes about 1μsec from the time the TH is changed until the data is fixed (corresponding to the elapse time of two NOP commands).

Example: The time difference from the time when the TH is changed until the switch value appears at PD4 of DATA.



◇ JOY PAD ◇

Player's button operations are recognized by reading data in $A10003 and $A10005 of the I/O addresses via CONTROL1 and CONTROL2. 7 bits of D6 to D0 in the I/O are sent to CONTROL1 and CONTROL2, but there are eight buttons. So, the D6 bit is used as the select bit, and the button operation is read in two steps.

** SAMPLE PROGRAM **

```
        MOVE.B  #$40,$A10009        ; Enter D5 to D0 of CONTROL1.
                                    ; D6 is set to output (select bit).
        MOVE.B  #$40,$A10003        ; 1 is output to D6 (C, B, right, left, down or up
                                      is selected).
        NOP
        NOP
        MOVE.B $A10003,D0           ; D5 to D0 values are read into register D0.
        MOVE.B #$00,$A10003         ; 0 is output to D6 (start, A, down or up is
                                      selected).
        NOP
        NOP
        MOVE.B  $A10003,D1          ; D5 to D0 values are read into register D1.
```

※ Each button outputs 0 when it is pressed.

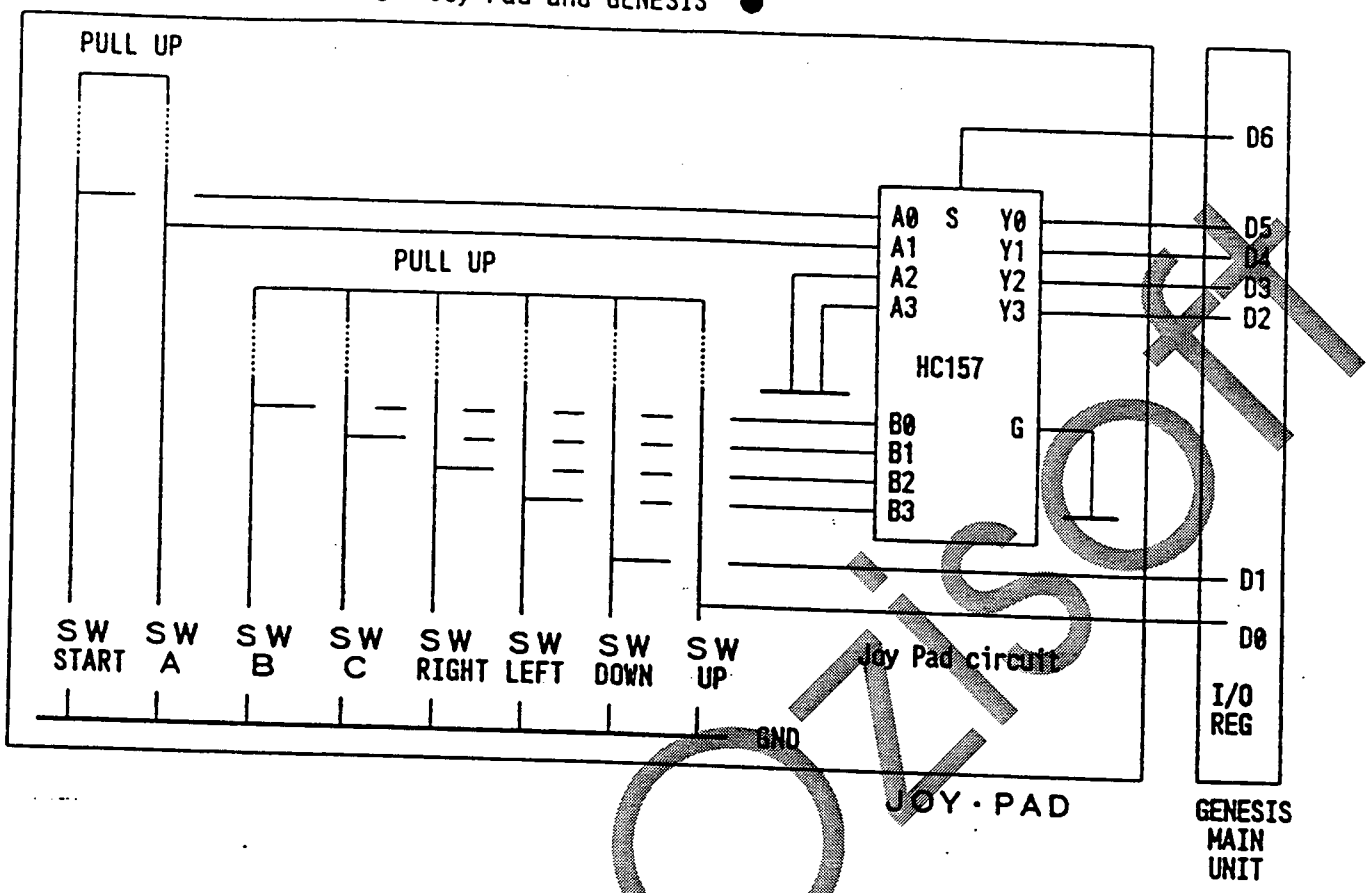
Register values after the program is executed.

DATA REG	D7	D6	D5	D4	D3	D2	D1	D0
REGISTER D0	0	1	C button	B button	RIGHT	LEFT	DOWN	UP
REGISTER D1	0	0	START	A button	0	0	DOWN	UP

HC 157 logic

Input		Output
S	G	Yx
X	1	-Z-
0	0	Ax
1	0	Bx

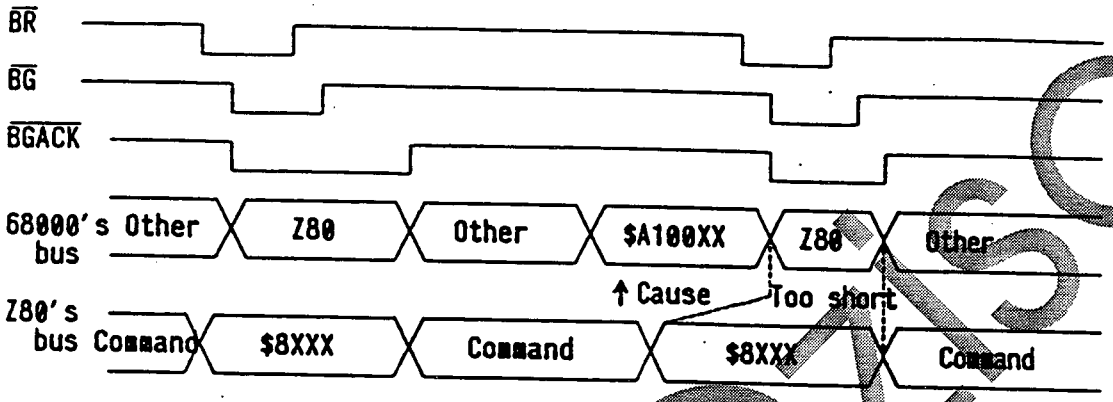
● Joy Pad and GENESIS ●



5. PRECAUTIONS FOR SOFTWARE DEVELOPMENT

§ 1 Z80 CANNOT PROPERLY ACCESS THE BUS ON THE 68000.

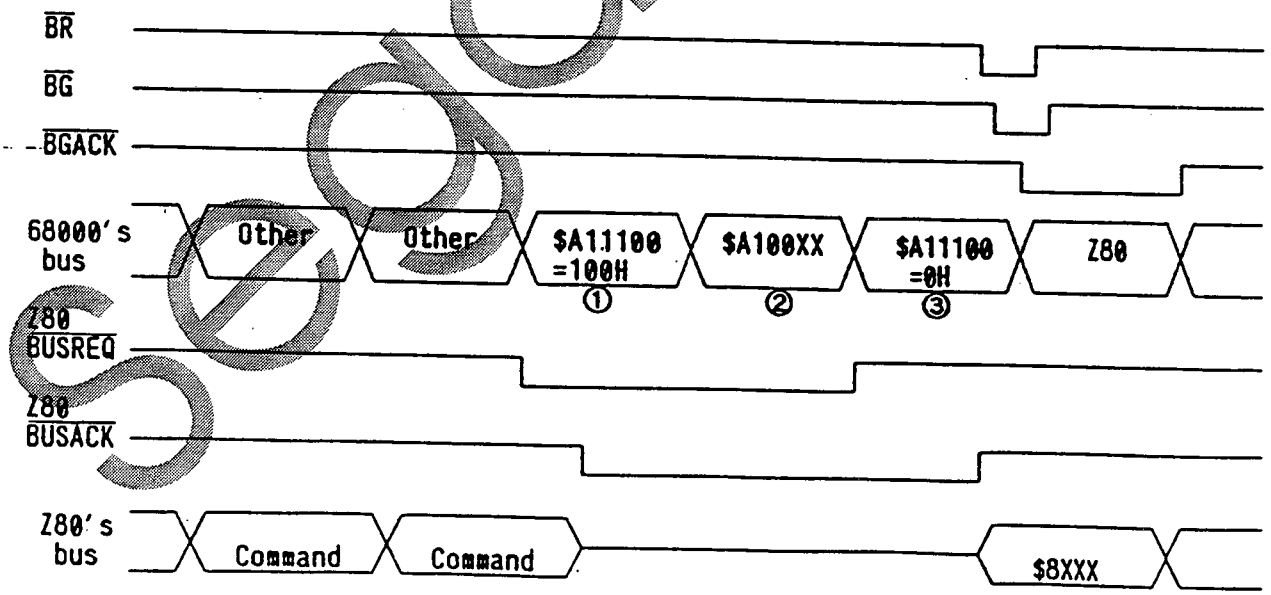
When the Z80 tries to enter the bus cycles of the 68000 to read/write a specific address on the 68000, and when the access to $A100xx is made in the 68000's bus cycle immediately before the bus cycle which the Z80 tries to enter, the cylce of the Z80's entry may become too short to read/write data properly.



Measure...When the 68000 accesses $A100xx.
①Send BUSREQ to the Z80 (suspend the Z80's bus access...$A11100 = 0100H)
②Access $A100xx.
③Release the BUSREQ of the Z80 (the Z80 starts the bus access....$A11100 = 0000H).

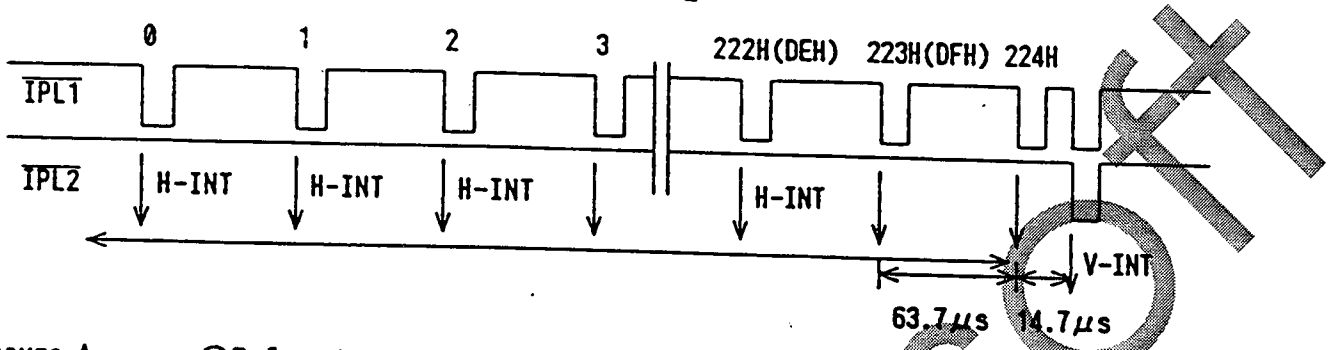
Modify the program to execute the above steps.

Note: it will be no problem if the program is designed to be synchronized with V_INT, etc. to eliminate a bad timing for the bus access.



§ 2 H_INT AND V_INT ARE USED IN GENESIS PROGRAMS

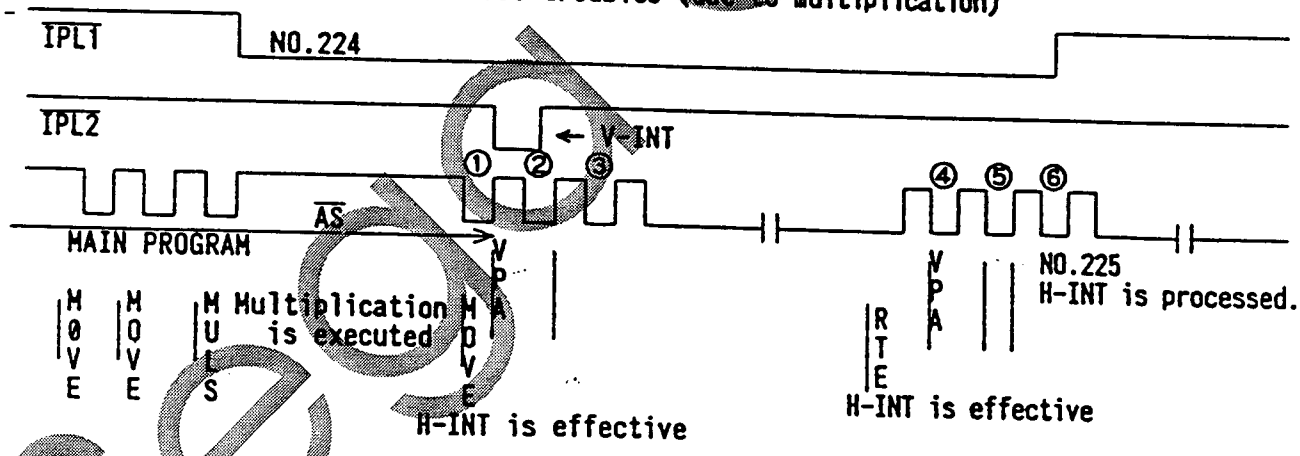
● When IE1 = 1 and #10 = 00H are set for VDP register #0, H_INT and V_INT occur at the timings below. Care is required for the timing of H_INT at No.224, in which case, the next V_INT occurs in only 14.7μsec. If the receipt of H_INT at No.224 is prolonged and the opportunity of the V_INT occurrence is missed, that V_INT is canceled. Once the V_INT is caceled, H_INT occurs at No.225 instead of V_INT.



Measure A
①Before H_INT at No.223 is completed, set IE0 = 0 for register #1 to suppress the next occurrence at No.224.
②Before the receipt of V_INT is completed, set IE0 = 1 for register #1 to make H_INT effective (No.224 also becomes effective).
③Receive H_INT at No.224.

Measure B
①Before H_INT at No.223 is completed, set SR = 25xx for the 68000 not to receive No.224.
②Before the receipt of V_INT is completed, return SR of the 68000 to the previous value (so that No.224 can be received).
③Receive H_INT at No.224.
※Don't give damages to the flag when you are handling SR.

●The case which was known to cause troubles (due to multiplication)



①After the multiplication, the 68000 decides to receive H_INT by means of MOVE.
②The 68000 generates VPA (INT-ACK). (No.224 is received.)
The VDP cannot find the VPA after the H_INT occurrence, and so the H_INT is treated as 'un-processed.' The VPA is received after the occurrence of V_INT. This receipt is assumed that V_INT has been received and the H_INT is preserved.
③Since H_INT at No.224 was received in ①, the processing is executed.
④The 68000 decides to receive H_INT again after the H_INT is proccessed at the occurrence of RTE.
⑤The 68000 generates VPA (INT-ACK). The VDP assumes that the VPA is the preserved H_INT, and proccesses it (at No.225).
⑥H_INT (at No.225) is received in ④, and the process is executed again.

§ 3 TROUBLES DUE TO INTERRUPTS DURING COMMUNICATION

The similar troubles to 'H_INT and V_INT are used' (see section 2) occur in:
a) 'H_INT and INT during data communication', and
b) 'V_INT and INT during data communication'.
Since 'INT during data communication' occurs asynchronously with V_INT and H_INT, the measures differ from those described in section 2.

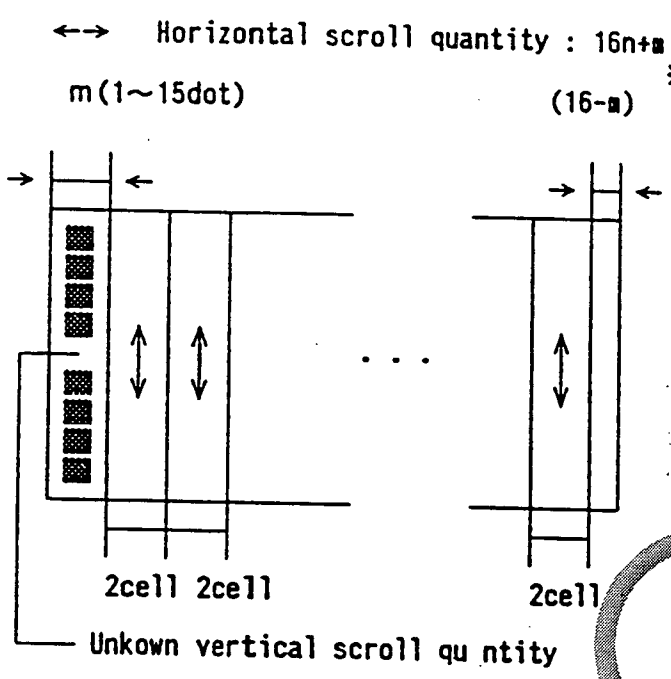
Measure for a)
INT and the receive flag are simultaneously set when data communication starts. The receive flag is cleared when received data is read by means of 'INT during data receipt' which does not overlap with H_INT. If it overlaps ,with H_INT and this H_INT is treated as the second 'INT during data receipt', the receive flag remains cleared. Hence, the proper measure should be selected by using the status of the receive flag to judge whether it is treated as 'INT during data receipt' or 'data receive INT'.

Measure for b)
Set the mode in which V_INT does not occur and execute the same treatment as that of V_INT by detecting the V timing using the V_BLK bit for VDP status.

○Others○
The timing for H_INT or V_INT can be judged by using H_BLK, V_BLK, HV_counter, etc. The receive flag can be used to judge whether 'INT during data receipt' or 'INT during data receipt as which H or V is treated' occurs. Other items and measures should be judged by combing IE0, IE1 and IE2 in the VDP registers, etc.

§ 4 PRECAUTIONS WHEN THE CONTROL A AND B SCREENS ARE USED

When vertical 2-cell-unit scroll and horizontal scroll are used together, remember that up to 15 dots at the left of the screen cannot be vertically scrolled if a number which is not a multiple of 16 (i.e., 1 to 15, 17 to 31, etc.) is assigned for the horizontal scroll quantity.



※When the horizontal scroll quqntity=16n+m (1≦m≦15) is set, the vertical scroll quantity becomes unkown in the left m-columns.

※Reason

When the horizontal scroll quantity= 16n+m(1≦m≦15) is set on the A or B screen, 21 words need to be securd in the RAM for the vertical scroll setting. But the actual capacity of the RAM is only 20 words.

§ 5 OTHER PRECAUTIONS

●Precautions for the main program and Z80 bus requests

Care is required for the following items in the main routine on the 68000 when the Z80 area is aacessed.

```
        .
        .   Main routine
        .
        .
```

①A bus request is sent to the Z80.
②An acknowledgement signal is checked.
③The interrupt is achieved.
④A bus request is released within INT.
(for example, for reading the control pad, etc.).
⑤The bus request is released within INT.
⑥Operation returns to the main routine.
⑦The data is written in the Z80 area.
However, the Z80 is usually locked with a bus request, but it has been released in (5).

Hence, the following troubles may occur.
- RAMs of the Z80 may be broken.
- Wrong data may be read when a Z80 bank is accessed.

※Measure※
The interrupt is set to be disabled before step (1) is executed.

●Precautions for sound access

①Sound stops during game play.

☆Cause and measure
When the busy flag of status data of the FM sound source (YM2612) is read,

(CS,RD,WR,A1,A0)=(0,0,1,0,1)

has been set (the address 4001H of the MEGA drive is accessed).
However, no provisions are given to the output of the YM2612 under the above conditions. It happens that 'not busy' is set and adta is successfully read,

※Measure
When the busy flag of the FM sound is read, any address other than 4000H address in the MEGA drive must not be accessed.

●When the reset button is pressed repeatedly

①Software runs away when the reset button is pressed repeatedly.

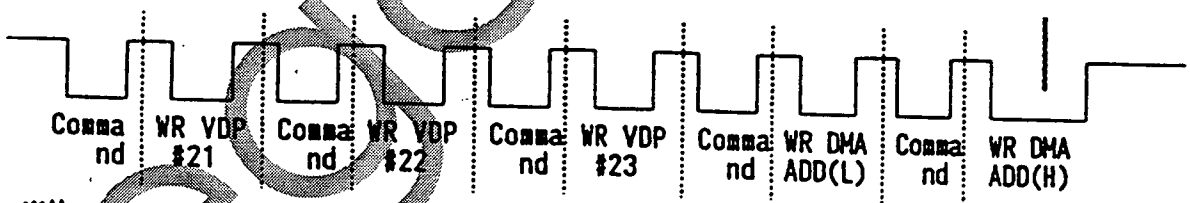
☆Cause and measure

- Pressing the reset button resets the CPU, but not the VDP.
- When the reset button is pressed while DMA is performed, the VDP continues the DMA.
- If the VDP is performing DMA when the VDP is accessed immediately after resetting, this access is ignored.

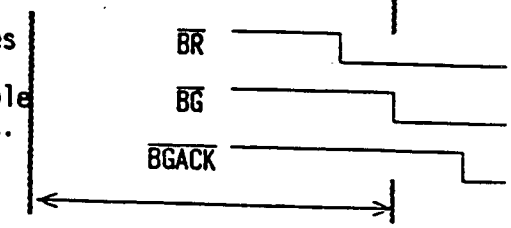
※Measure※
Before accessing the VDP after the initial program (ICD_BLK4), check 'DMA busy' in the status register. If DMA is underway, no access is made.

②The software still runs away in some cases even if the above measure is taken.

☆Cause and measure
Such troubles can occur, if resetting is made from the time when the parametter set for executing DMA by the CPU is terminated until the first DMA starts.



※Measure※
Although it is very rare for such troubles to occur, take the measure of ① and another appropriate measure of, for example not executing DMA immediately after reset.



Trouble occurs if the system is reset during this interval.

●Notes on the control pad and read-programs

The control pad is designed and manufactured in such a way that the up and down signals or the left and right signals are entered simultaneously. However, they can rarely be entered simultaneously, when the internalrubber is aged too much or strong forces exceeding the designed tolerance level are applied. So, design the software not to accept such simultaneous signals.

●Make a revision of the Manual

Regarding the access to VRAM, SRAM, VSRAM, please disregard the statement in the manual that you can implement byte access. You can only implement word access and long-word access.

Make the change in the manual

6. NOTES ON MEGA DRIVE EUROPE VERSION

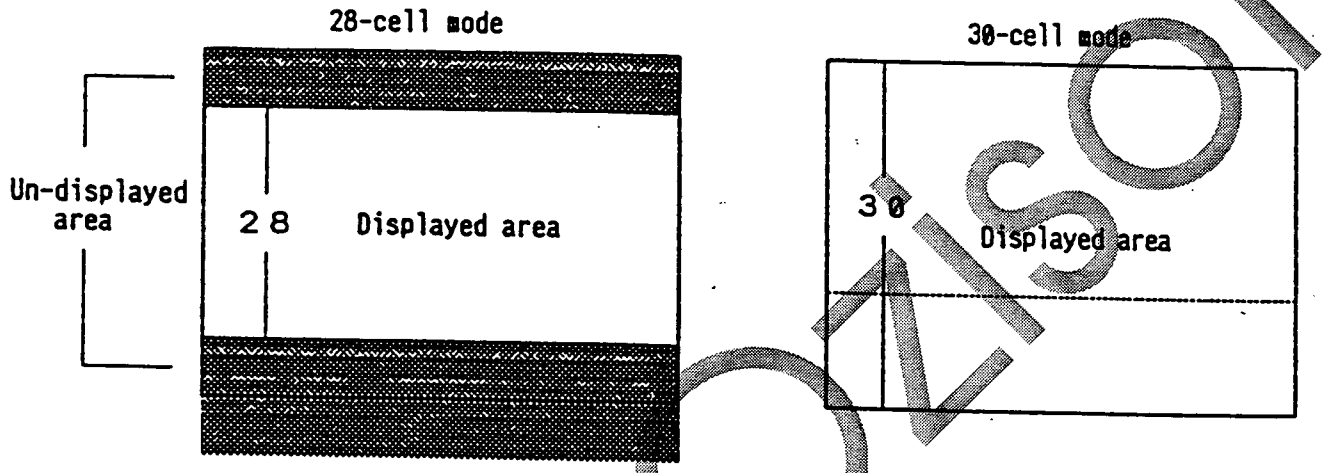
§ 1 PAL

●PAL
- ○The vertical size is increased by two cells.
- ○Intervals between interrupts are increased (16msec to 20msec).

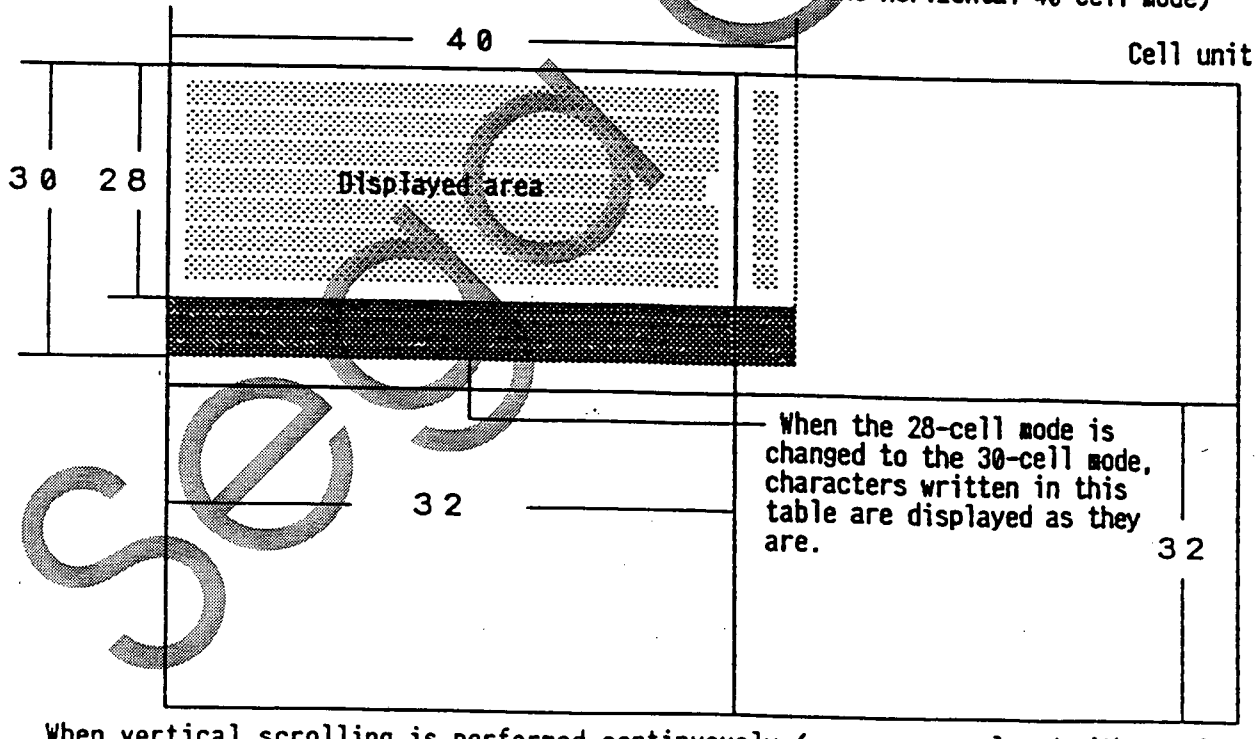
●Differences in development of MEGA DRIVE software and precautions.
- ○The vertical 30-cell mode can be used.
- ○Since the color change appears at the bottom of the screen, wait about 3msec after entering the V-interrupt, and then change the color.
- ○Since every operation goes slow when synchronized in interrupts, adjust movements and rates so that motions on the screen appear naturally.
- ○When sounds are produced in synchronization with interrupts, the tempo needs to be increased. The PAL data table is required.
- ○When sounds are produced by the FM timer, no modifications for NTSC or PAL are required.

§ 2 VERTICAL 30-CELL MODE

1. When the vertical 28-cell mode is changed to the vertical 30-cell mode on PAL, the 30-cell mode displays all the cells which have been displayed in the 28-cell mode and the two cells at the bottom which have not been displayed in the 28-cell modes.
2. The 28-cell mode does not display one cell each at the top and the bottom; the 30-cell mode displays the entire screen.
3. The V-blank period is about 5.6 msec in the 28-cell mode; it is about 4.6 msec (shorter) in the 30-cell mode 'approximately same as that on NTSC).
4. By developing software in awareness given to the PAL 30-cell mode, the software, which is common to NTSC and PAL and has minimized border part even in the PAL mode, can easily be developed.



◎Scroll size 64 × 64 (V-scroll quantity = 0 in the horizontal 40-cell mode)



When vertical scrolling is performed continuously (e.g., games played with vertical scr olling), it is delieved that there will be no problem even if the 28-cell mode is changed to the 30-cell mode.

7. NOTES ON THE BACKUP RAMs

There are two types of backp RAMs:

- Odd addresses in 200000H～203FFFH · · 64k bits
- Odd addresses in 200000H～20FFFFH · · 256k bits

The same access methods as those for the work RAMs are used. However, care is required for the following points.

- Data (initial values) in the backup RAMs is not known when the cartridge (product) is shipped. The initialization must always be executed when the power is turned on. (In many cases, they are cleared to $FF when shipped from the factory, but this cannot apply for all cases.)
- Data in the backup RAMs may rarely be destroyed. For this reason, the data must always be check and reproduced. Any important data must not be stored in the first word (in the first address) and last word (in the last address), since there is a high possibility that data in these word may destroyed.