SEGA OF AMERICA, INC.
Consumer Products Division

GENESIS TECHNICAL BULLETIN #1

TO: Developers and Third Parties

FROM: Technical Support

DATE: March 13, 1991

SUBJECT: **ERRORS WITHIN THE MICROTEC EXAMPLES**

There are three errors in examples that are provided. Please make the changes as noted.

In TESTC68K.bat:

The asm68k commands end with a semicolon, remove it and the file will assemble correctly.

The link command is incorrect. It should be:
LNK68K -c sieve.cmd -o sieve sieve

The C compiler documentation refers to a command line option of 'STRINGSINTEXT' for allocating string in code segment rather than data segment. The correct spelling for this option is 'STRINTEXT'.

125 Shoreway Road, San Carlos, CA 94070 • (415) 802-4400 • FAX (415) 802-4458

SEGA OF AMERICA, INC.
Consumer Products Division

GENESIS TECHNICAL BULLETIN #2

TO: Developers and Third Parties

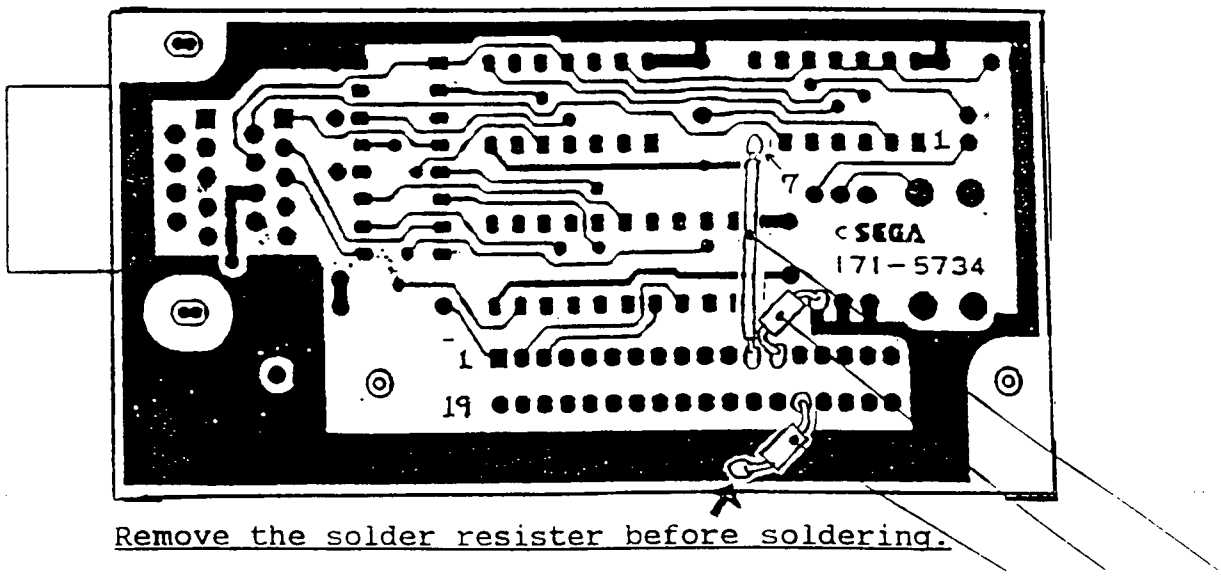
FROM: Technical Support

DATE: June 19, 1990

SUBJECT: **GENESIS LOADER BOARD**

In order to prevent loading errors, please have the following modification on the Genesis Loader Board. You need to have:

2 Resister 4.7kΩ 1/8W
1 Jumper Wire



Remove the solder resister before soldering.

1. Place 4.7k resister between Centronics Connector Pin #32 and +5V.

2. Place 4.7k resister between Centronics Connector Pin #13 and +5v.

3. Place jumper wire between Centronics Connector Pin #12 and ICI 74HC74 Pin #7 (GND).

125 Shoreway Road, San Carlos, CA 94070 • (415) 802-4400 • FAX (415) 802-4458

SEGA OF AMERICA, INC.
Consumer Products Division

GENESIS TECHNICAL BULLETIN #3

TO: Developers and Third Parties

FROM: Technical Support

DATE: August 2, 1990

SUBJECT: **PRECAUTION WHEN ACCESSING THE Z80 BUS FROM A 68000 MAIN (NON-INTERRUPT) ROUTINE**

If the following routine was executed and an interrupt occurred between Step 2 and 3, a data error would occur IF the interrupt also access the Z80 bus.

1. Send Z80 bus request
2. Check acknowledge
3. Writing data to Z80 bus
4. Release Z80 bus

Essentially, the main routine sends a bus request, but before it can conduct its data transfer, the interrupt sends its own bus request. When the interrupt completes, it releases the Z80 bus, as it should. Unfortunately, the main routine expects the bus to be available; thus, an error occurs. When this error occurs, the Z80 RAM is corrupted. One symptom of this error is a sound is SOMETIMES interrupted.

The solution is really quite simple, before Step 1 above, disable interrupts and re-enable when done.

125 Shoreway Road, San Carlos, CA 94070 • (415) 802-4400 • FAX (415) 802-4458

```
;           :           :           ( TC )
;           :           ( TA )
;           ( ST )
;sw_data1+1 is the address for edge data of port_1

read_sw:

                    z80_diw                         ; set busreq
                    bsr.b       rs_sub
                    z80_ei
                    rts

          rs_sub

                    lea         sw_data1,a5         ;switch data store address
                    lea         port_1,a6           ;port for p1
                    bsr         ??rs_0
                    addq.w      #2,a6               ;a6 = port_2

;
??rs_0:
                    move.b      #$0,(a6)            ;set TH = 0
                    nop                             ;wait
                    nop
                    move.b      (a6),d7             ;input data when TH = 0
                    asl.b       #2,d7               ;chane data when TH = 0 & wait
                    move.b      #$40,(a6)           ;set TH = 1
                    andi.w      #11000000b,d7       ;chane data when TH = 0 & wait
                    move.b      (a6),d6             ;input data when TH = 1
                    andi.w      #00111111b,d6
                    or.b        d6,d7
                    not.b       d7                  ;push -> 1
                    move.b      (a5),d6             ;copy
                    eor.b       d7,d6
                    move.b      d7,(a5)+            ; switch data
                    and.b       d7,d6
                    move.b      d6,(a5)+            ;switch edge data
                    rts

;*******< port initial  >******;
sw_init:                                            ;port initial
                    moveq       #$40,d7             ;set TH out
                    move.b      d7,cont_1
                    move.b      d7,cont_2
                    move.b      d7,cont_3
                    rst
;*****************************************************;
;                                                     ;
;-----------------------------------------------------;
;     input       a6.1 = controller port address      ;
;     output      d7.b == $00       Modem             ;
;                         $0d       Mega Drive Joy Stick;
;                         $0e       Ram Disk          ;
;                         $0f       ETC. or None      ;
;     use         d0,a0   |                           ;
;*****************************************************;
cont                set     $0006
```

```
c:  _port:

                movem.l d1-d2/a5,-(sp)          ;register push
                dis
                z80_diw
                lea     ??and_data(pc),a5       ;data table address set
                move.b  (a5),cont(a6)           ;Th bit set output
                moveq.l #0,d7                   ;rerturn register initial
                moveq.l #$08,d1                 ;counter set
```

SEGA OF AMERICA, INC.
Consumer Products Division

GENESIS TECHNICAL BULLETIN #4

TO: Developers and Third Parties

FROM: Technical Support

DATE: September 7, 1990

SUBJECT: **READING THE CONTROLLER PADS**

When reading a location within the range of $A10000-$A100FF, which includes the controller pads, the Z80 must have a bus request. If not, the wait state will change from 250ns to 110ns. The shorter time will cause the Z80 to misread ALL further data. Once the ports have been read, the Z80 may be released.

Included with this bulletin is a reprint of the routine that reads the control pads in the Logo Demo program. As you can see, this routine sends a bus request, reads the pads, and then, releases the Z80.

125 Shoreway Road, San Carlos, CA 94070 • (415) 802-4400 • FAX (415) 802-4458

SEGA OF AMERICA, INC.
Consumer Products Division

GENESIS TECHNICAL BULLETIN #5

TO: Developers and Third Parties

FROM: Technical Support

DATE: November 26, 1990

SUBJECT: **CARTRIDGE IDENTIFICATION**

Every product must have, at location 100h, an IDTABLE. The table is slightly different for each type of developer. The difference is small, affecting the product code and the copyright, both important for product approval. Please replace the following pages in your Sega Software Manual.

125 Shoreway Road, San Carlos, CA 94070 • (415) 802-4400 • FAX (415) 802-4458

```
;************************************************
;*     SEGA GENESIS CARTRIDGE ID TABLE          *
;*                                              *
;*     STANDARD FORMAT FOR SEGA OF AMERICA      *
;*                                              *
;*     NOV 26 1990 SEGA OF AMERICA              *
;*                                              *
;************************************************

check_sum equ  $0000
     ORG $0100
;-------------------------------------------------------------------------
     dc.b 'SEGA GENESIS    '         ;100
     dc.b '(C)SEGA 199X.XXX'         ;110 release year.month
     dc.b 'your game tile  '         ;120 Japan title
     dc.b '                '         ;130
     dc.b '                '         ;140
     dc.b 'your game title '         ;150 US title
     dc.b '                '         ;160
     dc.b '                '         ;170
     dc.b 'GM MK-XXXX -XX'           ;180 product #, version
     dc.w check_sum                  ;18E check sum
     dc.b 'J               '         ;190 controller
     dc.l $00000000,$0007ffff,$00ff0000,$00ffffff     ;1a0
     dc.b '                '         ;1B0
     dc.b '                '         ;1C0
     dc.b '                '         ;1D0
     dc.b '                '         ;1E0
     dc.b 'U               '         ;1F0
```

```
;*************************************************
;*     SEGA GENESIS CARTRIDGE ID TABLE           *
;*                                               *
;*     STANDARD FORMAT FOR THIRD PARTIES         *
;*                                               *
;*     NOV 26 1990 SEGA OF AMERICA               *
;*                                               *
;*************************************************

check_sum equ  $0000
     ORG $0100
;-------------------------------------------------------------------------
     dc.b 'SEGA GENESIS    '         ;100
     dc.b '(C)T-XX 199X.XXX'         ;110 release year.month
     dc.b 'your game tile  '         ;120 title
     dc.b '                '         ;130
     dc.b '                '         ;140
     dc.b 'your game title '         ;150 title
     dc.b '                '         ;160
     dc.b '                '         ;170
     dc.b 'GM T-XXXXXX XX'           ;180 product #, version
     dc.w check_sum                  ;18E check sum
     dc.b 'J               '         ;190 controller
     dc.l $00000000,$0007ffff,$00ff0000,$00ffffff     ;1a0
     dc.b '                '         ;1B0
     dc.b '                '         ;1C0
     dc.b '                '         ;1D0
     dc.b '                '         ;1E0
     dc.b 'U               '         ;1F0
```

ADDENDUM 1 - ABSTRACT FROM PAGE 2

Joysticks/Peripherals

It is possible to connect various peripherals to the I/O Port other than joysticks. Each peripheral has its own ID Code and it is possible for the CPU to check the code to identify what is connected. Be aware that there are some Master System peripherals, which cannot be identified by this method.

S1 ID Code

The ID Code for the external ports, CTRL1, CTRL2, and EXT., which correspond to the I/O Port's DATA1, 2, 3 (PD3, PD2, PD0) are shown by a logical "or" operation. In concrete terms, setting the TH Pin to output mode and outputting either 1 or 0 as data will yield the following.

```
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 relationship between the ID Code and the peripheral hardware is shown below.

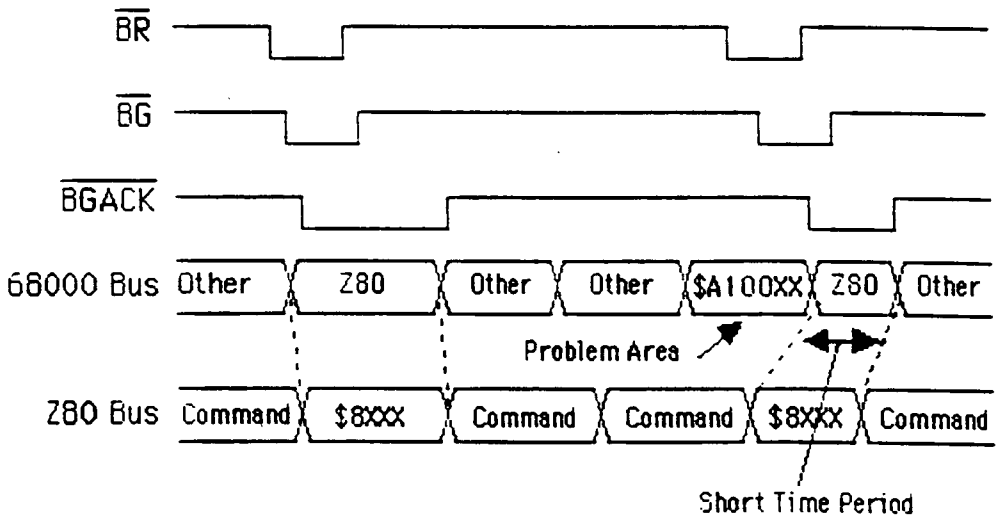
Device Name	ID3	ID2	ID1	ID0		
Old Style Joystick (2 trig):	1	1	1	1	($F)	If no joystick
Undefined:	1	1	1	0	($E)	
New Style Joystick (3 trig):	1	1	0	1	($D)	Power stick
Sega Reserved:	1	1	0	0	($C)	
Undefined:	1	0	1	1	($B)	
Sega Reserved:	1	0	1	0	($A)	
Undefined:	1	0	0	1	($9)	
Undefined:	1	0	0	0	($8)	
Undefined:	0	1	1	1	($7)	
Undefined:	0	1	1	0	($6)	
Sega Reserved:	0	1	0	1	($5)	
Undefined:	0	1	0	0	($4)	
Undefined:	0	0	1	1	($3)	
Undefined:	0	0	1	0	($2)	
Undefined:	0	0	0	1	($1)	
Sega Reserved:	0	0	0	0	($0)	

ADDENDUM 2 - NO ADDENDUM 2

ADDENDUM 3 - ABSTRACT FROM PAGE 1

1. When The Z80 Cannot Do A 68000 Bus Access

There are times when the Z80 cannot read or write good data in the 68000's addresses. If the 68000's bus cycle accesses $100XX prior to the Z80's interrupt, the bus cycle for the Z80 interrupt is shortened. When this occurs, the Z80 cannot read or write good data.



When The 68000 Is Accessing $A100XX

1. A BUSREQ is sent to the Z80. (Stops the Z80 bus access- $A11100=0100H)

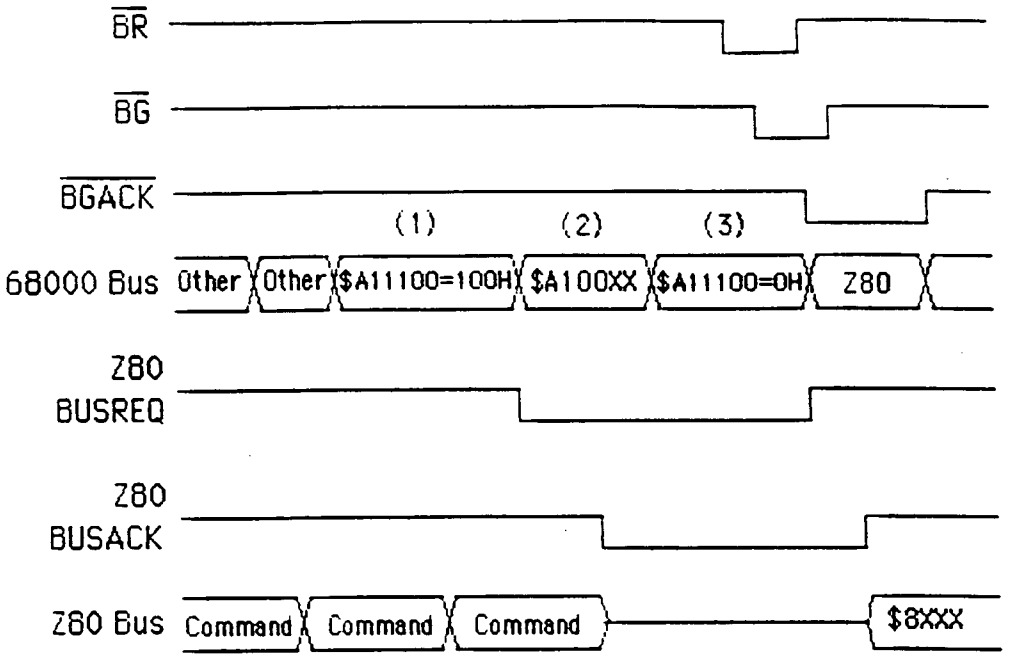
2. A100XX is accessed.

3. The BUSREQ from the Z80 is cleared. (The Z80 begins bus access- $A11100=0000H)

Change the 68000's programs to follow the steps shown above.

Miscellaneous

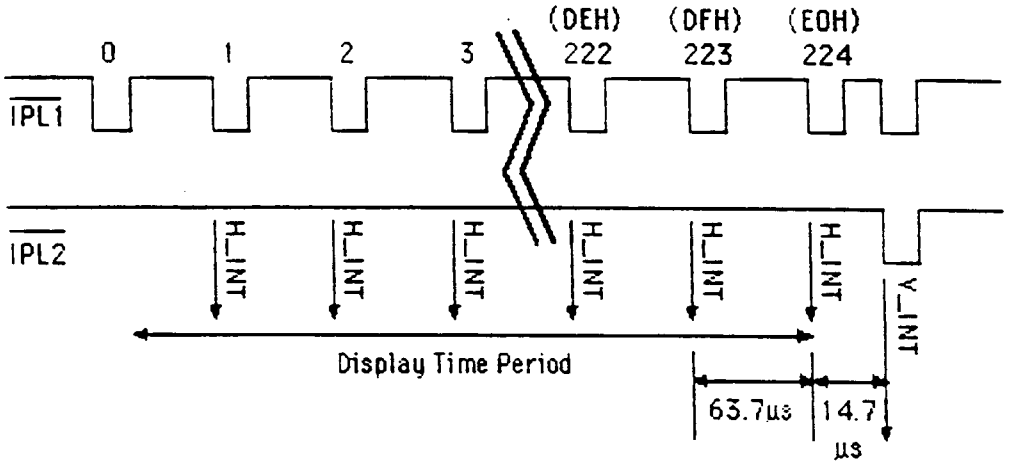
If the program structure uses something like V_INT for sync, there should be no problems as long as it is set up so that unfavorable bus access patterns are not created.


BR
BG
BGACK
(1)
(2)
(3)
68000 Bus
Other
Other
$A11100=100H
$A100XX
$A11100=0H
Z80
Z80
BUSREQ
Z80
BUSACK
Z80 Bus
Command
Command
Command
$8XXX

ADDENDUM 3 - ABSTRACT FROM PAGE 2

2. When Using H_INT And V_INT With A Genesis Program

When the VDP register #0's IE1=1 and register #10=00H,H_INT and V_INT are created with the timing scheme described below. However, a problem occurs during No.224H_INT's timing. Since the amount of time which is available before V_INT is only 14.7µs, the acceptance of the No.224H_INT means that the V_INT will miss the V_INT occurrence period. As a result, V_INT is cancelled. Moreover, when the V_INT is cancelled, the No.225H_INT occurs instead of the V_INT.



Solution A

1. Set register #0's IE1=0 before No.223H_INT ends in order to prevent No.224 from occurring.

2. Set register #0's IE1=1 before V_INT is accepted in order to make H_INT valid. (i.e., No.224 also becomes valid)

3. Process No.224H_INT.

Solution B

1. Set the 68000's SR=25XX before No.223H_INT ends in order to prevent No.224 from being processed.[1]

2. Restore the 68000's SR prior to the end of V_INT processing. (Enables No.224 processing)

3. Process No.224H_INT.

---

[1]**Warning:** When manipulating SR, make sure not to destroy the flag.

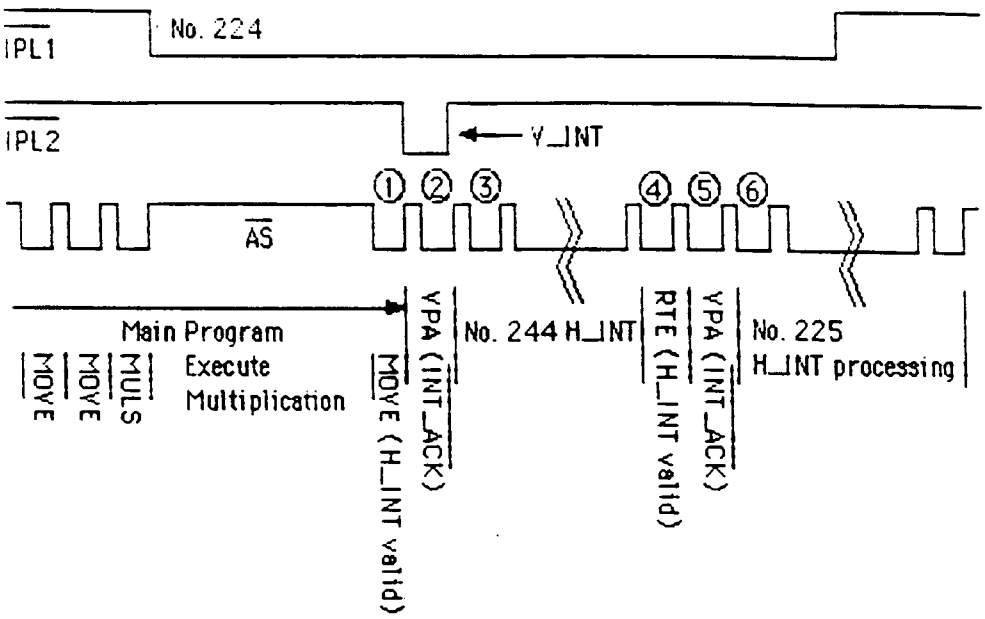


Diagram Of The Problem Caused By A Multiplication Command

1. The MOVE command given after multiplication readies the 68000 to process H_INT.

2. The 68000 creates a VPA (INT_ACK). Since the VDP (for processing No.224) does not see a VPA after H_INT occurs, H_INT is not processed. After V_INT occurs, the 68000 receives a VPA and mistakes this for V_INT processing. H_INT is stored.

3. Processing is executed since H_INT (No.224) was received in Part 1.

4. Given a RTE after processing H_INT, the 68000 readies to process H_INT again.

5. The 68000 creates a VPA (INT_ACK). The VDP (equivalent to No.225) processes the stored H_INT.

6. Receiving H_INT (No.225) in Part 4, processing occurs again.

ADDENDUM 3 - ABSTRACT FROM PAGE 4

4. The Interrupt Problem During Data Transfer

Part 2 discussed the problems encountered when using H_INT and V_INT in a program. The same kinds of problems occur during a) an INT involving H_INT data reception and b) an INT involving V_INT and data reception. (Refer back to Part 2 for details)

The "INT during data reception" does not occur in sync in relation to V_INT and H_INT; therefore, the solutions to the problem are different.

Solution In The Case of a)

During data reception, the receive flag is set at the same tine of the INT. The receive flag is cleared when the data is read by the "INT during data reception," which does not overlap with H_INT. When H_INT occurs again, it overlaps and passes as the second "INT during data reception." After this occurs, the receive flag is left in clear status.

Solve the problem by discriminating between the "INT during data reception" and the H_INT, which passes as the "INT during data reception." This can be determined by checking on the status of the receive flag.

Solution In The Case of b)

Set the mode, which prevents the occurrence of V_INT's, and then, use the VDP's V_BLK status bit to detect V-timing. Use this method to perform the same type of processing as V_INT.

Miscellaneous

It is possible to differentiate H_INT and V_INT timing by using H_BLK, V_BLK, and the HV_counter.

It is possible to distinguish between the "INT during data reception" and the "H and V that pass as the INT during data reception" by analyzing the receive flag.

Try to think of other solutions by using IE0, IE1, and IE2 settings within the VDP register.

ADDENDUM 4

1. Regarding The DMA

When using DMA's other than *FILL VRAM* and *VRAM COPY*, use the following method to program:

1) Send a bus request to the Z80.
2) Base the DMA destination address set section in RAM.
3) Rewrite the one leading word of data after the end of the DMA.

1) and 2) are used in all cases, except FILL VRAM and VRAM COPY.
3) is used for WRAM to VRAM, WRAM to COLOR RAM, and WRAM to VSRAM.

2. Compatibility With The MKIII Mode

VDP Registers #11 through #23 cannot be written over, unless bit 2 of Register #1 is not set to "1."

Reasons:

- Registers #11 through #23 are masked for MKIII mode's error trapping purposes.
- MKIII uses Registers #0 through #10.
- The Genesis mode is active when Register #1's bit 2=1.
- The registers are set to 0 on power up.

3. Warnings Regarding Use Of Back-Up RAM

- The back-up RAM data is unformatted when the cartridge (product) is shipped. Always make sure to initialize during power up. (In most cases, the back-up RAM data is cleared with $FF at the factory; however, don't count on this as a matter of fact.)
- Although this does not occur frequently, there may be occasions when the data becomes garbled. Because of this possibility, make sure to check the back-up data and perform reproduction/retrieval processing. Moreover, do not place critical data at leading and ending RAM addresses, since one word in both addresses have a high probability of becoming garbled.

SEGA OF AMERICA, INC.
Consumer Products Division

GENESIS TECHNICAL BULLETIN #6

TO: Developers and Third Parties

FROM: Technical Support

DATE: April 2, 1991

SUBJECT: **IMPOSSIBLE VALUES READ FROM THE CONTROLLER PAD**

It is possible for a well used controller pad to yield incorrect values, specifically: up AND down, left AND right. This is due to the small piece of rubber wearing out inside the pad.

Your routine that handles the values read should be written, so that if the number is NOT one of the expected values, it should re-read the pad to obtain a correct value.

As always, please call, if you have any questions or comments.

125 Shoreway Road, San Carlos, CA 94070 • (415) 802-4400 • FAX (415) 802-4458

I. Software Guidelines

1. When accessing the Z80 area from within the 68000's main routine, beware of the following:

1) Execute a bus request to the Z80.
2) Confirm execution.
3) An interrupt occurs.
4) A Z80 bus request is executed within the interrupt routine. (Execute for the purpose of reading the controller pad)
5) The bus request is cleared within the interrupt.
6) Return to main routine.
7) Write data to Z80 area.

In principle, the Z80 is left with the Z80 bus request in effect; however, in this example, it is cleared in Step 5).

When this occurs, the following things occur:

- The RAM of the Z80 gets damaged.
- Incorrect data is read during Z80 bank access.

A Fix For The Problem:

Disable interrupts prior to Step 1).

2. Problems During Sound Access

1) Sound output stops during the game.

Diagnosis Of The Problem:

The busy flag was read in the FM sound generator YM2612 like this (address 4001H was accessed in the Mega Drive):

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

However, in the case of the YM2612, its output is not regulated according to the conditions set up above. This results in the device being read as "not busy," and as a consequence, ends up outputting sound.

A Fix For The Problem:

When the FM sound generator's busy flag is read, do not access anything else other than:

(A1,A0)=(0,0) (Mega Drive address 4000H)

3. Repeated Resets

1) The software goes out of control when resets are repeated.

Diagnosis Of The Problem:

- When the reset occurs, the CPU is reset; however, the VDP is not reset.
- When the reset occurs during DMA, the VDP continues the DMA.
- The VDP is accessed right after the reset. If this is done while the VDP is executing a DMA, then this access becomes ineffective.

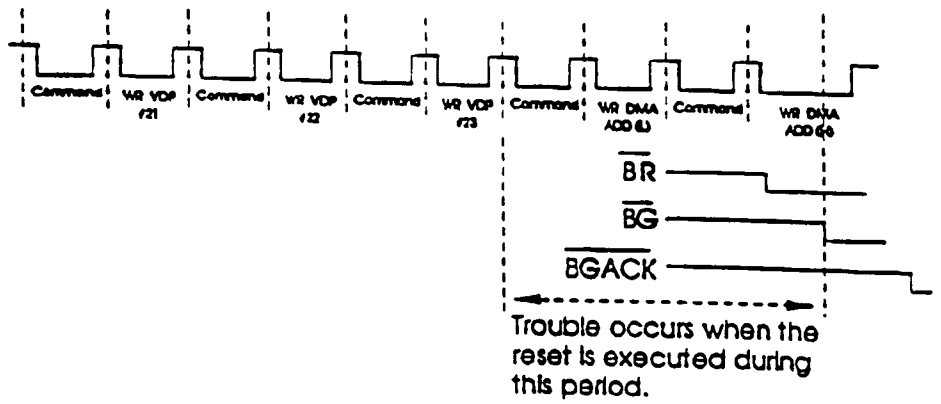
A Fix For The Problem:

Before accessing the VDP after the initialization program (ICD_BLK4), check the DMA BUSY status register. If a DMA is being executed, do not attempt to access.

2) The problem still persists even after the precautions above are taken.

Diagnosis Of The Problem:

The problem will occur when the reset takes place between the period of time when the CPU has finished setting the parameters to execute the DMA and the actual execution of the DMA itself.



A Fix For The Problem:

On top of the precautions taken in 1), as an example, make sure not to execute a DMA right after a reset.

II. Corrections To The Manual

When discussing VRAM, CRAM, VSRAM access, the manual states in Pages 22-27 that byte access is possible; however, in reality, this is false.[1] The reason for this is that after the VDP address register is set, an instruction for a CPU word access (ex. a fetch operation) causes the VDP to think during the next access that it is still a word access.

From now on, please consider the text regarding byte access in Pages 22-27 as being null and void.

Corrections to Mega Drive Addendum 6, Page 3

III. About Control Pads

Although the control pad is designed and produced so that simultaneous up/down or left/right input is impossible, there are times when simultaneous input occurs due to fatigue of the internal rubber parts or force on the pads, which exceed design specs. **Because of this, from now on, all software should have switch read routines that can process simultaneous directional input.**

Corrections to MD Manual, Page 77

IV. Bank Switching

Although the manual states that the 68000 may set the bank registers; however, in reality, this is not the case. Please execute bank switching in the Z80 from the Z80.

---

[1] **Translator's Note:** The page numbers are from the original Japanese-language manual.

SEGA SOFTWARE DEVELOPMENT
AND GAME STANDARDS

The following Sega of America Game Standards Policy should serve as a guideline for the development of Sega-licensed cartridges by defining the types of content and themes inconsistent with Sega's Corporate, Marketing, and Product Development philosophy. SOA will not approve cartridges (i.e., audio visual work, packaging, and instruction manuals)

- with sexually suggestive or explicit content;
- which reflect ethnic, racial, religious, nationalistic, or sexual stereotypes or language;
- which depict excessive graphic violence;
- which use profanity in any form or which incorporate language that could be offensive by prevailing public standards and tastes;
- which encourage or glamorize the use of drugs, alcohol, or tabacco;
- which make an overt political statement;
- which contain depictions of symbols or groups which are an anathema to racial, religious, or ethnic groups.
- which is a potential infringement of a legally protected copyright or trademark.

It is not possible for a single document to cover every possible eventuality in an area as broad and subjective as thematic content. For this reason, these standards of guidance should be one of common sense, practicality, and prevailing public taste.

GENESIS SOFTWARE DEVELOPMENT STANDARDS
(Revised 01/03/91)

I. INITIAL DISPLAY

A. The trademark, "SEGA TM", shall be displayed in the center of the screen. Logo code will be provided by SEGA.

B. Check the control pad(s) while SEGA logo is displayed. If no control pad is connected or only the 2P control pad is connected, make the loop as follows:

SEGA Logo - Title - Demo - Sega Logo

(While no button is pressed.)

If the START button on either controller is pressed, the game title should be displayed. If the Player 1 controller is connected, a button press will exit from the Logo to the title. Another press should exit the title.

II. TITLE DISPLAY

A. Display the title sequence and the text, "PUSH START BUTTON." If any button on either controller is pressed at this point, the Start/Option Select screen shall appear.

B. Copyright marking (user can exit via any button, but cannot bypass altogether).

1. Original software
   ©, the year of publication, copyright holder,
   e.g., ©1990 SEGA

2. Licensed software
   e.g., ©1987 TEKKON
   PROGRAMMED GAME ©1990 SEGA
   (or as required by contract)

C. After the title sequence, the screen automatically proceeds to the demonstration mode within 5-10 seconds.

III. DEMONSTRATION

A. If the player does not start the game during the title screen (5-10 seconds), the screen shall proceed to the demonstration.

B. Turn on the sound. (If not already on)

C. If the START button is pressed during the demo, the Logo screen shall appear again.

D. If the START button is not pressed during the demo, the SEGA logo shall appear after a certain time.

POWER ON	
SEGA	SEGA logo screen
	When the START Button is pushed or after an interval of about 2 seconds.
CHAMPION FINGER WRESTLING	Game software title
PUSH START BUTTON	
©1990 SEGA	©Copyright notice
	If START button is pressed, go on to game screens. Otherwise, in approximately 5-10 seconds.
DEMONSTRATION	10-20 seconds Return to SEGA logo screen. If START button is pushed, go to Logo screen.

IV. START/OPTION SCREEN

A. The Start/Option screen is required with the exception of transferred games and games which might lose their game play characteristics by having the screen.

B. Control pad maneuvers and screen display.

Selection shall be made by moving the D-button upward or downward or left and right and pressing the START button.

One Player

1 PLAYER START
OPTIONS

If the player selects **1 PLAYER START**, he can play alone.

Two player simultaneous play

1 PLAYER START
2 PLAYER START
OPTIONS

1. 1P Control Pad Only

   Although the screen indicates **2 PLAYER START**, the cursor shall be controlled, so that the player cannot select it. In order to make it clear (whether the player can select or not), different colors should be used.

   When the player selects **1 PLAYER START**, he can play alone.

2. 1P and 2P Control Pads

   a. When the 1P control pad's START button is pressed:

      1 PLAYER START
      2 PLAYER START
      OPTIONS

      The player can select all three; however, the player can only play with 1P control pad.

1 PLAYER START	The player can play alone with 1P side.
2 PLAYER START	Two players can play simultaneously (1P control pad controls the Player 1 and 2P control pad controls the Player 2).

3. When the 2P control pad's START button is pressed:

   1 PLAYER START
   2 PLAYER START
   OPTIONS

   The player cannot select **1 PLAYER START** and he can only control the 2P control pad.

2 PLAYER START	Two players can play simultaneously (1P control pad controls the Player 1 and 2P control pad controls the Player 2).

4. Two player alternate play (if applicable)

a. 1P control pad only

1 PLAYER START
2 PLAYER START
OPTIONS

The player can select all three.

1 PLAYER START	The player can play along with 1P side.
2 PLAYER START	Two players can play alternate with 1P side control pad.

b. Option Screen

LEVEL	EASY
PLAYER	5
SOUND TEST	1
CONTROL	A JUMP B SHOT
	C SHOT
RAPID	ON
EXIT	

When the player selects **OPTIONS** on the Start/Option screen, the option screen shall appear.

The player can select items by pressing the D-Button upward or downward and change by pressing the D-Button sideways.

How to return to the title or option screen:

Select **EXIT** and press A, B, or C button or press the START button (the placement of cursor should not matter).

The following options shall be implemented when applicable:

LEVEL	Difficulty setting
PLAYER	Number of players
SOUND TEST	Sound test
CONTROL	Control setting
RAPID	Continuous shooting setting

V. PASSWORD/NAME ENTRY SCREEN

A. Select a clear font, so there will be no confusion.

e.g., O, 0, o, and Q, 1 and 1

Passwords should be 12 characters or less. It is a good idea to subgroup them and use different colors for each. Do not use punctuation marks or BOTH upper and lower case letters.

B. Maneuver

PASSWORD

1. To enter the password or name, select letters by D-Button and enter by button C.

2. When finished, put the cursor on END.

3. To delete letters, hold button C and move the cursor by the D-Button or use RTN (return) and PCD (proceed) function.

4. Even if the password is wrong, the letters entered shall not be erased.

5. Movement of the cursor shall be a loop (up-down-left-right).

VI. RESET/PAUSE

A. When the reset button is pressed, during DEMO, the screen shall return to SEGA logo (the high scores, the option settings, and the password should not be cleared).

B. During the play, the start button shall work as the pause button. During the 2 player simultaneous play, both control pads shall be able to pause and cancel.

C. During pause, the sound shall be silenced. The background music only shall continue immediately when the pause is released.

D. The pause function shall not work when the SEGA logo or the title logo are displayed or during the demonstration.

E. **PAUSE** shall be indicated on the screen during the pause.

NOTE: Adventure and role-playing games may not require the pause mode.

VII. BASIC MANEUVERS DURING THE GAME PLAY

A. General Maneuvers (May vary based on specific requirements of individual games)

START Button	Start/Pause
D-Button	Used to determine the directions
Button A	Special functions and rare maneuvers
Button B	Passive maneuvers, such as **CANCEL**
Button C	Active maneuvers, such as **DECIDE**

B. Sample Quick Maneuvers for action and shooting games

1. 2 actions (e.g., Ghouls 'n Ghosts, Thunder Force II)

Button A	Jump/Sub Shot
Button B	Shot/Main Shot
Button C	Jump/Sub Shot

2. 3 actions (e.g., Trojan, Altered Beast)

Button A	Punch
Button B	Kick
Button C	Jump

C. Sample Slow Maneuvers for role-playing, simulation, and adventure games.

1. 2 actions

Button A	Decide/Opening Windows
Button B	Cancel
Button C	Decide/Opening Windows

2. 3 actions

Button A	Check Special Functions
Button B	Cancel
Button C	Decide/Opening Windows

D. Others

Even if there are two actions for the game, all buttons should be utilized. It is convenient if maneuvers can be changed in the option mode.

VIII. CONTINUE/ENDING

A. CONTINUE

It shall appear when the game is over, if appropriate. (This may not apply to sports or RPG's, etc.)

The CONTINUE shall be limited to a certain number of times.

The screen should return to the SEGA logo after 10-20 seconds.

B. The ending screen shall not be cancelled, but it can be exited via the START button.

The screen shall return to the SEGA logo after the ending (after 10-20 seconds).

IX. GAME OVER

A. This shall appear for 60 seconds, and then, the screen shall display the SEGA logo.

B. Pressing the START button while GAME OVER is indicated shall cause the screen to change and show the Sega logo.

X. CONTENTS OF THE GAME

A. For important indications, such as the score, there shall be a two cell margin on the right and left side and a one cell margin on the top and the bottom of the screen (some monitors do not show these areas well). Do not put important score, text, or time information in the top, bottom, leftmost, or rightmost row or columns.

B. When background music composed by a Third Party is utilized, pay careful attention to the copyright in order to avoid any infringement therein. Copyrights shall be secured for all Third Party music when applicable.

C. Names, which are closely related to any particular manufacturer, product, and character, shall not be used.

D. Buttons shall work as soon as they are pressed, <u>not released</u>, unless there is a specific approved reason to violate this rule.

E. The "Start Button" shall work separately from other buttons.

F. High scores and previous players shall be up-to-date scores shown on TITLE, DEMONSTRATION screens.

If a 2-player game, then both scores shall be displayed simultaneously.

G. Title music shall start when the title appears and finish before the demonstration starts.

H. The title of the game and names used within the game shall not infringe upon the trademarks of Third Parties.

I. At the end of the game, any words or expressions, which imply "copyright," "patent," or "invention," shall not be made during credits.

J. Developer credits may be included at the end of the game, subject to approval by SEGA.

NOTE: In the case where any question arises as regards the ITEMS set forth in the SEGA GAME SOFTWARE DEVELOPMENT STANDARDS or matters for which no stipulation is made, the developer shall, from time to time as required, consult with the SEGA Software Department and resolve said problem subject to the outcome of such consultation.

SEGA OF AMERICA, INC.
Consumer Products Division

GENESIS TECHNICAL BULLETIN #11

TO: Developers and Third Parties

FROM: Technical Support

DATE: September 9, 1991

SUBJECT: **PROBLEMS DURING SOUND ACCESS**

Sound output stops during a game.

Problem:

The busy flag was read in the FM sound generator YM2616 like this (address 4001h)

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

However, in the case of the YM2612, it's output is not regulated according to the conditions set above. This results in the device being read as "not busy," and as a consequence, ends up outputting sound.

Fix:

When the FM sound generator's busy flag is read, do not access anything else other than:

(A1,A0) = (0,0) (address 4000h)

125 Shoreway Road, San Carlos, CA 94070 • (415) 802-4400 • FAX (415) 802-4458

SEGA OF AMERICA, INC.
Consumer Products Division

GENESIS TECHNICAL BULLETIN #12

TO: Developers and Third Parties

FROM: Technical Support

DATE: September 9, 1991

SUBJECT: **PROBLEMS WITH REPEATED RESETS**

The software seems to go out of control when resets are repeated.

Problem:

When the reset occurs, the CPU is reset; however, the VDP is not. When the reset occurs during a DMA, the VDP continues the DMA. The VDP is accessed right after the reset. If this is done while the VDP is executing a DMA, then this access is ineffective.

Fix:

Before accessing the VDP after the initialization program (ICD_BLK4), check the DMA BUSY status register. If a DMA is being executed, do not attempt to access the VDP.

If the problem persists, ensure that you are not executing a DMA right after a reset.

125 Shoreway Road, San Carlos, CA 94070 • (415) 802-4400 • FAX (415) 802-4458

SEGA OF AMERICA, INC.
Consumer Products Division

GENESIS TECHNICAL BULLETIN #13

TO: Developers and Third Parties

FROM: Technical Support

DATE: September 9, 1991

SUBJECT: **CORRECTIONS TO THE GENESIS SOFTWARE MANUAL**

When discussing VRAM, CRAM, and VSRAM access, the manual states in Pages 22-27 that byte access is possible. This is incorrect. Access is limited to word or long word.

On Page 77, it implies that the 68000 may set the bank switches. The bank switches MUST be set by the Z80.

These changes affect Version 1.0 of the manual, later versions will reflect this correction.

125 Shoreway Road, San Carlos, CA 94070 • (415) 802-4400 • FAX (415) 802-4458

SEGA OF AMERICA, INC.
Consumer Products Division

GENESIS TECHNICAL BULLETIN #14

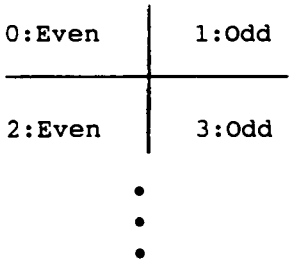
TO: Developers and Third Parties

FROM: Technical Support

DATE: September 5, 1991

SUBJECT: **ROM SPLITTING**

As we all know, Genesis products must be split into 128k odd and even pieces. Sega expects the ROM images to be in the following format:



For larger products, continue the above pattern. We would appreciate it if it all products would conform to this method of splitting. Please request the utility to split ROMs, M2B, or Split4, if your current tool can't output files in this manner.

125 Shoreway Road, San Carlos, CA 94070 • (415) 802-4400 • FAX (415) 802-4458

SEGA OF AMERICA, INC.
Consumer Products Division

GENESIS TECHNICAL BULLETIN #15

To: Sega Developers

From: Dave Marshall, Technical Support

Date: April 5, 1993

Re: Genesis Technical Information

==========

The attached document contains information on waiting during VRAM reading. The Super Target information applies to a CD development system.

125 Shoreway Road, San Carlos, CA 94070 • (415) 802-4400 • FAX (415) 802-4458

o WAIT during Mega Drive VRAM READ.

In the Mega Drive, during VRAM READ, when the following conditions are not met, data may not be read correctly.

<u>During display period and H-blank</u>

VRAM Address Set

VRAM READ

WAIT exceeding 116 CPU clock cycles

Next VRAM Address Set

<u>During V-blank</u>

VRAM Address Set

VRAM READ

Wait exceeding 12 CPU clock cycles

Next VRAM Address Set

As seen above, after the last VRAM READ is completed, WAIT is required before the next VRAM address is set. With respect to VRAM READ, it can be read continuously.

o SUPER TARGET *IC 19*

When 68000 CPU accesses an illegal area, "DATA ACKNOWLEDGE" signal may not be sent back - causing "hang up" in the production unit. Depending on some ROM emulators or ICE (+SUPER TARGET), even the emulator may hang up. To avoid such hang ups, measures have been taken so that SUPER TARGET would return ACKNOWLEDGE signal to the CPU without system hang ups. This is done via IC 19.
As a result, due to this IC function, for SUPER TARGET to function normally, the hang up occurs in the production unit.

Please take the following steps to correct the problem:

(1) Before checking the software through SUPER TARGET + CPU, remove IC 19 from the socket.

(2) When using ICE (ROM emulator), IC19 may be removed. Reinstall the IC should ICE (ROM emulator) act abnormally.

(3) The above measures are common with the MEGA DRIVE and the MEGA CD.

SEGA OF AMERICA, INC.
Consumer Products Division

GENESIS TECHNICAL BULLETIN #16

To: Sega Developers

From: Dave Marshall, Technical Support

Date: April 5, 1993

Re: Genesis Technical Information

==========

The attached document contains information on new peripherals and their protocol. The source files that this note references are available on the BBS in the Genesis conference. The files apply mainly to the 4P adapter. There is already driver code available for all other peripherals in the Genesis conference. This driver code is in a file called UNIVERS.ZIP

MEGA DRIVE
(January 7, 1993)

1. Regarding Mega Drive register #11 IE2 bit

This bit is set to allow or inhibit the external interrupt. Set this bit to "0" when not using external interrupt (GUN, etc.). When this bit is "1," error occurs on the Sega-made address checker.

2. Regarding Mega Drive Control Pad access

When accessing control pad, please strictly observe the following items:
(If not observed, the peripherals to be sold in the future may not operate properly.)

(1) Access Pad from within V-INT (routine)
Generally, pad should be accessed from within V-INT.
The access period is approximately 16 ms for NTSC and 20 ms for PAL.
The access interval should not exceed the above periods.

(2) Pad access through V-INT routine must be limited to one time only. Reading pad data repeatedly via V-INT could cause malfunction.

(3) Check the peripherals
Currently, the sale of 6-button pad micro trackball 4P adapter (all tentative names) is being planned. Therefore, even for the Pad-exclusive software, peripherals other than the Pad may be used or those devices may be switched. So please diligently check peripheral devices currently used and make sure to input data that meet those devices.
We will supply a sample program that checks the device ID at each V_INT and then executes controller input. Please use this program as your reference.

(4) Modify sample program only after understanding it completely.
Our sample program has been modified incorrectly into some software product that is for sale and suffer from erratic operation. Please modify the sample program after full understanding of its structure and operation.

(Translated by: J. Tabrizi, April 5, 1993)

How to accommodate peripherals

The 6-button pad, Mouse, and 4P TAP will soon go on sale. Whether or not each game will accommodate such peripherals, they must be made switchable once the game is in progress. 4P TAP has the selector function, therefore, it is very possible to be switched in mid-game. To this end, a sample program has been made to check the device ID at each V-INT and then execute the controller input. Please study this program. The file is <I0.ASM>. We also prepared a set of I/O check program.

The work structure may vary depending on the accommodation arrangement (the sample is described here). The unwanted areas may be deleted.

Accommodating 4P_TAP

I/O data

offset	use		label in MEM_MAP. ASS
+0	1byte	Connector-1 device ID	port_id1
+1	1byte	Connector-2 device ID	port_id2
+2	4b	4p tap connector info for Connector-1	mlt_info1
+2		CN1 of 4P-TAP from Connector-1	
+3		CN2 of 4P-TAP from Connector-1	
+4		CN3 of 4P-TAP from Connector-1	
+5		CN4 of 4P-TAP from Connector-1	
+6	4b	4p tap connect info for Connector-1	mlt_info2
+6		CN1 of 4P-TAP from Connector-1	
+7		CN2 of 4P-TAP from Connector-1	
+8		CN3 of 4P-TAP from Connector-1	
+9		CN4 of 4P-TAP from Connector-1	

10Bytes of data must be secured for controller data WORK. An amount equal to 8 units is secured. Whether or not Connector-1 is 4P_TAP, Connector-2 data is stored in "iodata_5."

+10	1P data or 4P_TAP CN1 data	iodata_1
+20	CN2 data at 4P_TAP	iodata_2
+30	CN3 data at 4P_TAP	iodata_3
+40	CN4 data at 4P_TAP	iodata_4
+50	2P data or 4P_TAP CN1 data	iodata_5
+60	CN2 data at 4P_TAP	iodata_6
+70	CN3 data at 4P_TAP	iodata_7
+80	CN4 data at 4P_TAP	iodata_8

iodata_x data varies by controller.

Note:
ST=start button
A=A button
B=B button
C=C button
X=X button
Y=Y button
Z=Z button
MD=Mode button
[...] describes each bit.

;3-button JOYPAD

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
+0	0...JOYPAD 3-button Controller ID							
+2	[ST	A		C	B	Right	Left	Down Up]
+3	+2 leading edge data							

(Translated by: J. Tabrizi, April 5, 1993)

+4 Not used

;6-button JOYPAD
+0 1...JOYPAD 3-button Controller ID
+2 [ST A C B Right Left Down Up]
+3 +2 leading edge data
+4 [0 0 0 0 MD X Y Z]
+5 +4 leading edge data
+6 Not used

;Mouse
+0 2...Mouse Controller ID
+2 [Center Middle Right Left Yover Xover Ysign Zsign]
+3 +2 leading edge data
+4 X data
+5 Y data
+6 X data sign
+8 Y data sign

The files have the following composition:

```
I-- IO_READ.ME        This file
I-- MEM_MAP.ASS       Memory map assign
I-- MACRO.MAC         Macro
I-- MAIN.ASM          Processing from Vector, ID, power supply ON
I-- ICD_BLK4.PRG      Included in Hard Initial Program <MAIN.ASM>
I-- INT.ASM           Interrupt program
I-- IO.ASM            This file is the center of I/O program
I-- SUB.ASM           Subroutine
I-- VDP.PRG           Subroutine for VDP related items
I-- SPCNT.ASM         Sprite control
I-- IO_CHK.ASM        Checking and displaying I/O data. Not to be minded
I-- EQU_EV.ASM          For reflecting mem_map.ass into EVT file
I-- ASCII.ASM         ASCII CG data
I-- IOSAMP.MK           Make definition file
I-- IOSAMP.LNK        Link definition file
I-- I.CMD               ICE initial batch
I-- MK.BAT              Batch file for MAKE
I-- XREF.1              file for Xref inclusion

I              I--EQU_EV.OBJ
I              I--MAIN.OBJ
I              I--INT.OBJ
I              I--IO.OBJ
I              I--SUB.OBJ
I--OBJ--    I--SPCNT.OBJ
               I--ASCII.OBJ
               I--IO_CHK.OBJ
               I--IOSAMP.SYM
               I--IOSAMP.EVT
               I--IOSAMP.S28
```

Reference:
This is for those who say "ID does the check, but only JOYPAD can do the control."

<3-button accommodation>
Essential WORK is 1 Byte only (2 Bytes when making edge data)

(Translated by: J. Tabrizi, April 5, 1993)

For program see <IO_JOY3.PRG>. Make sure to call only once from V_INT routine. Used in both cases of making edge and otherwise.

```
MAKE_EDGE       set   1       ;if 1 then make edge, 0...only read
If so: Making edge
MAKE_EDGE       set   0       ;if 1 then make edge, 0...only read
If so: Not making edge
```

<Distinguishing between 3-button and 6-button>
Please see "get_joy" routine within <IO.ASM>

(Translated by: J. Tabrizi, April 5, 1993)

GENESIS TECHNICAL BULLETIN #17

To: Sega Developers

From: Bert Mauricio, Technical Support

Date: May 25, 1993

Re: Genesis Technical Information

==========

The attached document contains information on Eproms for use in Sega development boards.

Listed are manufacturers and model numbers of EPROM chips that are recommended to use with the Genesis/Game Gear boards. Please note chips are non-JEDEC.

1 megabit:	NEC	D27C1000A	120ns or 150ns
	AMD	AM27C100	150ns
	Fujitsu	MBM27C1000	150ns
	Intel	D27C100	150ns
2 megabit:	NEC	D27C2001D	150ns
	AMD	AM27C020	200ns
4 megabit	NEC	D27C4001D	150ns
	Toshiba	TC574000D	150ns

*note We have found the Intel 27C020 2 megabit chips to be unreliable.

There has also been problems when using 2 megabit chips on the standard 8M/16M EPROM board w/64k backup. (171-5663 M5 Test MB)

Dip switch settings:

1 megabit chips	1-on
	2-on
	3-on
	4-off
2-megabit chips	1-on
	2-on
	3-off
	4-on

If there are recurring problems, it may be more reliable to use 2 megabit chips on the new SRAM/EE backup board which accepts 1,2 and 4 megabit chips. (GN 837-8093)

Eprom source: America II Electronics (Florida) (800) 767-2637

GENESIS TECHNICAL BULLETIN #18

To: Sega Developers

From: Dave Marshall, Technical Support

Date: May 28, 1993

Re: Genesis Technical Information

==========

This is just a reminder that all Genesis carts must contain "lockout code". This code will prevent cartridges from being run on machines not listed in the Country Code field of the ID Block.

This lockout code is available on the Sega Technical Support BBS in a file called LOCK.ZIP in the Genesis and 3rdParty conferences. The zip file contains information on how and where to use the lockout code.

The test department does look for this feature and will bug any game that does not contain the lockout capability.

SEGA OF AMERICA, INC.
Consumer Products Division

GENESIS TECHNICAL BULLETIN #19

To: Sega Developers

From: Dave Marshall, Technical Support

Date: July 6, 1993

Re: Mega CD Technical Information

==========

This document contains requirements that must be followed when producing games for the non-U.S. market.

1. All software available in Japan should not contain a "TM" or "R" with the Sega Logo. All software available for the U.S. and European markets must have a "TM" on the right side of the Sega Logo.

2. Any SEGA games made available for the European market must be PAL compatible. The PAL version must be of the same graphic and sound quality as the NTSC version.

3. Any SEGA games made available in Japan must be compatible with both NTSC and PAL game systems.

130 Shoreline Drive, Redwood City, CA 94065 • (415) 508-2800 • FAX (415) 802-1448

SEGA OF AMERICA, INC.
Consumer Products Division

GENESIS TECHNICAL BULLETIN #20

To: Sega Developers

From: Dave Marshall, Technical Support

Date: July 8, 1993

Re: Genesis Technical Information

==========

The following tech note contains information for developing Genesis games that are larger than 16mbits.

130 Shoreline Drive, Redwood City, CA 94065 • (415) 508-2800 • FAX (415) 802-1448

MEGA DRIVE TECHNICAL INFORMATION

1. Regarding the VDP Access

In cases that the transmission to VRAM within the main routine is interrupted while in progress and the VDP is being accessed during the interruption, the writing to VRAM that was taking place in the main routine cannot function properly, thereby becoming the cause of a bug. Accordingly, when accessing towards VDP in the main routine, it is necessary to deal with it by using either 1 or 2.

1 Allow the main routine and the interrupt routine to run at the same time.

2 Prohibit interrupt while transmitting in the main routine.

Regarding 1:

When transmitting to VDP in the main routine, establish "a flag that says it is transmitting" and that flag must be confirmed in the interrupt routine. Access VDP only when VDP is not being accessed in the main.

Regarding 2:

When transmitting to VDP in the main routine, prohibit interrupt before performing the address set for the transmission, and permit interrupt after the data transmission has ended.

2. Regarding the Operation CHECK While Using EP-ROM

The enlarging of game capacity is anticipated hereafter as a trend, but in order to be certain that the accompanying EP-ROM operation is stabilized, be cautious of the following items.

1 Decrease the number of ROM that loads onto the ROM baseboard as much as possible by using large capacity devices.

Because the electric power consumption of ROM increases when the number of loads on the ROM increases, it is burdened and can be the cause of an operation malfunction.

The maximum number of possible loads varies according to:

- the ROM being used
- the type of base board being used
- the type of device being used

and cannot be specified but if kept about 4 will remain stabilized.

(Translated Bv: DDP. June 28. 1993)

2 Because the electric power consumption is great, EP-ROM's whose maximum use should be avoided are

•AMD Company made products
•INTEL Company made products

Specifically, the new product "Mega Drive 2" has a severe design on the electric power consumption. In the case that the worst operation malfunction occurs, check to see if 1 or 2 is relevant and can be applied. Furthermore, do all that can be done in the normal operation times to establish the operation circumstances referred to in the previous section.

3. Regarding the Production of Games over 16 Mbits

By using the present cartridge software corresponding to MEGA DRIVE, the capacity where present development is possible is as follows:

•16Mbit+256Kbit(max)
000000H~1FFFFFH(ROM) + 200000H~20FFFFH(BUP)

The methods corresponding to capacities larger than this are written on the last page. Development is in progress (release date pending).

If there is no backup, it is possible to develop up to the following capacity without using bank switch.

•Up to 32Mbit
000000H~3FFFFFH(ROM)

The S-RAM board is presently under development (release scheduled for summer of '93).

(Translated Bv: DDP. June 28. 1993)

Regarding MEGA DRIVE Bank Switching

Divide all 68,000 (ROM) space, every 4 Mbit.
(Bank 0 - bank 63; MAX 256 Mbit)

Divide the MD cartridge also into 8 areas of 4 Mbit and fix only area 0 that has vectors and it is possible to assign option banks to the remaining 7 areas.

The bank should be designated by the bank establishing register (odd number place of MD's $A13OF1~$A13OFF).

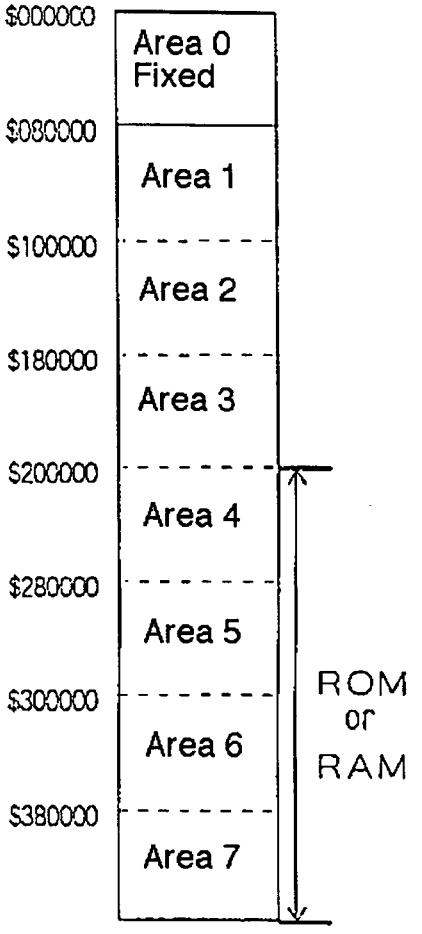
Register 0's 0th bit switches ROM/RAM after the $200000th place, and the number 1 bit designates the RAM light protect.

Register 1 thru register 7 corresponds to area 1 thru area 7.

The initial that depends on the hardware at the time of reset turns the light protect OFF (write in possible), as the cartridge area becomes 32 Mbit ROM space.

(Translated Bv: DDP. June 28. 1993)

MD Cartridge Area



Bank Setting Register

	D7	D6	D5	D4	D3	D2	D1	D0
REG. 0 $A130F1	0	0	0	0	0	0	0:W 1:P	0:ROM 1:RAM
REG. 1 $A130F3	0	0	BN5	BN4	BN3	BN2	BN1	BN0
REG. 2 $A130F5	0	0	BN5	BN4	BN3	BN2	BN1	BN0
REG. 3 $A130F7	0	0	BN5	BN4	BN3	BN2	BN1	BN0
REG. 4 $A130F9	0	0	BN5	BN4	BN3	BN2	BN1	BN0
REG. 5 $A130FB	0	0	BN5	BN4	BN3	BN2	BN1	BN0
REG. 6 $A130FD	0	0	BN5	BN4	BN3	BN2	BN1	BN0
REG. 7 $A130FF	0	0	BN5	BN4	BN3	BN2	BN1	BN0

BN0～BN5 : Bank Number

REG.0 , D1 • • 0 : WRITE
1 : PROTECT

(Translated Bv: DDP. June 28. 1993)

SEGA™

SEGA OF AMERICA, INC.
Consumer Products Division

MEGA Drive
Technical Info #9

Doc. # MD-TECH-09-100694

General Notice

When using this document, keep the following in mind:

1. This document is confidential. By accepting this document you acknowledge that you are bound by the terms set forth in the non-disclosure and confidentiality agreement signed separately and /in the possession of SEGA. If you have not signed such a non-disclosure agreement, please contact SEGA immediately and return this document to SEGA.

2. This document may include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new versions of the document. SEGA may make improvements and/or changes in the product(s) and/or the program(s) described in this document at any time.

3. No one is permitted to reproduce or duplicate, in any form, the whole or part of this document without SEGA'S written permission. Request for copies of this document and for technical information about SEGA products must be made to your authorized SEGA Technical Services representative.

4. No license is granted by implication or otherwise under any patents or other rights of any third party or SEGA Enterprises, Ltd.

5. Software, circuitry, and other examples described herein are meant merely to indicate the characteristics and performance of SEGA's products. SEGA assumes no responsibility for any intellectual property claims or other problems that may result from applications based on the examples describe herein.

6. It is possible that this document may contain reference to, or information about, SEGA products (development hardware/ software) or services that are not provided in countries other than Japan. Such references/ information must not be construed to mean that SEGA intends to provide such SEGA products or services in countries other than Japan. Any reference of a SEGA licensed product/program in this document is not intended to state or imply that you can use only SEGA's licensed products/ programs. Any functionally equivalent hardware/software can be used instead.

7. SEGA will not be held responsible for any damage to the user that may result from accidents or any other reasons during operation of the user's equipment, or programs according to this document.

NOTE: A reader's comment/correction form is provided at the back of this document. Please address comments to :

SEGA of America, Inc., Technical Translation/Publication Services
130 Shoreline Drive, Redwood City, CA 94065

SEGA may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

REFERENCES

In translating/creating this document, certain technical words and/or phrases were interpreted with the assistance of the technical literature listed below.

1. *KenKyusha New Japanese-English Dictionary*
   1974 Edition

2. *Nelson's Japanese-English Character Dictionary*
   2nd revised version

3. *Microsoft Computer Dictionary*

4. *Japanese-English Computer Terms Dictionary*
   Nichigai Associates
   4th version

Mega Drive Technical Info #9

1. Cartridge data (ID) specification changes
   Please note that the specifications below apply also to the 32X.

The following changes have been made in the code entered in 1FOh.

- Old Specifications: Entered data of corresponding countries
- New Specifications: 1 byte of operation hardware code entered in 1FOh in ASCII code.

  1FFh from 1F1h is buried in the space code (20h).
- Description
  When making distinctions between application hardware information ($A10001) and thereby creating operation restrictions, the operation hardware code shows which hardware is being programmed to enable operation. This code is used as confirmation information when Sega checks the compatibility of the hardware and application.

$A10001			Main Sales	Operation Hardware Code (figures from 0 to F below)															
Bit 7	Bit 6	Hard Specs.	Region	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	Japan, NTSC	Japan, S. Korea, Taiwan	x	O	x	O	x	O	x	O	x	O	x	O	x	O	x	O
0	1	Japan, PAL		x	x	O	O	x	x	O	O	x	x	O	O	x	x	O	O
1	0	Overseas, NTSC	N. America, Brazil	x	x	x	x	O	O	O	O	x	x	x	x	O	O	O	O
1	1	Overseas, PAL	Europe, Hong Kong	x	x	x	x	x	x	x	x	O	O	O	O	O	O	O	O

↑ Example 2 Example 1 ↑

O: Operates x: Does not operate

Example 1) When no application operation restriction are in force (common ROM), "F" which has all "O"s is entered.

Example 2) For applications operating only in Genesis which is sold in the U.S. $A10001 is operated at: Bit 7 = 1 and Bit 6 = 0. Operation hardware code becomes "4" according to the table above when not being operated by other hardware.

Address	1F0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Operation Hardware Code	4	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
ASCII Code	34	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20

NOTE: Please be aware that if there is an error in the code above, the application cannot perform master release.

READER CORRECTION/COMMENT SHEET

Keep us updated!

If you should come across any incorrect or outdated information while reading through the attached document, or come up with any questions or comments, please let us know so that we can make the required changes in subsequent revisions. Simply fill out all information below and return this form to the Developer Technical Support Manager at the address below. Please make more copies of this form if more space is needed. Thank you.

General Information:

Your Name _____ **Phone** _____

Document number MD-TECH-09-100694 **Date** _____

Document name MEGA Drive Technical Info #9

Corrections:

Chpt.	pg. #	Correction

Questions/comments: _____

Where to send your corrections:

Fax: (415) 802-3963
Attn: Manager,
Developer Technical Support

Mail: SEGA OF AMERICA
Attn: Manager,
Developer Technical Support
275 Shoreline Dr. Ste 500
Redwood City, CA 94065

SEGA™

SEGA OF AMERICA, INC.
Consumer Products Division

MEGA Drive/32X Address Checker Specification

(Not used with MEGA CD)

Doc. # MAR-57-102894

General Notice

When using this document, keep the following in mind:

1. This document is confidential. By accepting this document you acknowledge that you are bound by the terms set forth in the non-disclosure and confidentiality agreement signed separately and /in the possession of SEGA. If you have not signed such a non-disclosure agreement, please contact SEGA immediately and return this document to SEGA.

2. This document may include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new versions of the document. SEGA may make improvements and/or changes in the product(s) and/or the program(s) described in this document at any time.

3. No one is permitted to reproduce or duplicate, in any form, the whole or part of this document without SEGA'S written permission. Request for copies of this document and for technical information about SEGA products must be made to your authorized SEGA Technical Services representative.

4. No license is granted by implication or otherwise under any patents or other rights of any third party or SEGA Enterprises, Ltd.

5. Software, circuitry, and other examples described herein are meant merely to indicate the characteristics and performance of SEGA's products. SEGA assumes no responsibility for any intellectual property claims or other problems that may result from applications based on the examples describe herein.

6. It is possible that this document may contain reference to, or information about, SEGA products (development hardware/ software) or services that are not provided in countries other than Japan. Such references/ information must not be construed to mean that SEGA intends to provide such SEGA products or services in countries other than Japan. Any reference of a SEGA licensed product/program in this document is not intended to state or imply that you can use only SEGA's licensed products/ programs. Any functionally equivalent hardware/software can be used instead.

7. SEGA will not be held responsible for any damage to the user that may result from accidents or any other reasons during operation of the user's equipment, or programs according to this document.

NOTE: A reader's comment/correction form is provided at the back of this document. Please address comments to :

SEGA of America, Inc., Technical Translation/Publication Services
130 Shoreline Drive, Redwood City, CA 94065

SEGA may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

REFERENCES

In translating/creating this document, certain technical words and/or phrases were interpreted with the assistance of the technical literature listed below.

1. *KenKyusha New Japanese-English Dictionary*
   1974 Edition

2. *Nelson's Japanese-English Character Dictionary*
   2nd revised version

3. *Microsoft Computer Dictionary*

4. *Japanese-English Computer Terms Dictionary*
   Nichigai Associates
   4th version

MEGA Drive/32X
Address Checker Specification

Caution

- Objectives of the address checker
  Because the game malfunctions and does not operate if the MD function is expanded in the future, this prevents access to prohibited areas of the game program and illegal use of various registers. As a result, the address checker does not check for bugs in the program.

- Errors may not be detected by the address checker even when there is a bug in the program or if something not allowed by the manual is performed.

- The address checker can check only the part of the executed program.

1.0 Specifications

1) The Address Checker checks access to prohibited access areas by the 68000 CPU accompanying software development used with the Mega Drive (referred to hereafter as MD) and the 32X (referred to hereafter as S32X), and errors for data written to various registers.

2) The address, data, and program counter can be switched and displayed when an error occurs in combination with the R/W, UDS, and LDS conditions. (During a display, HALT is applied to the CPU.)

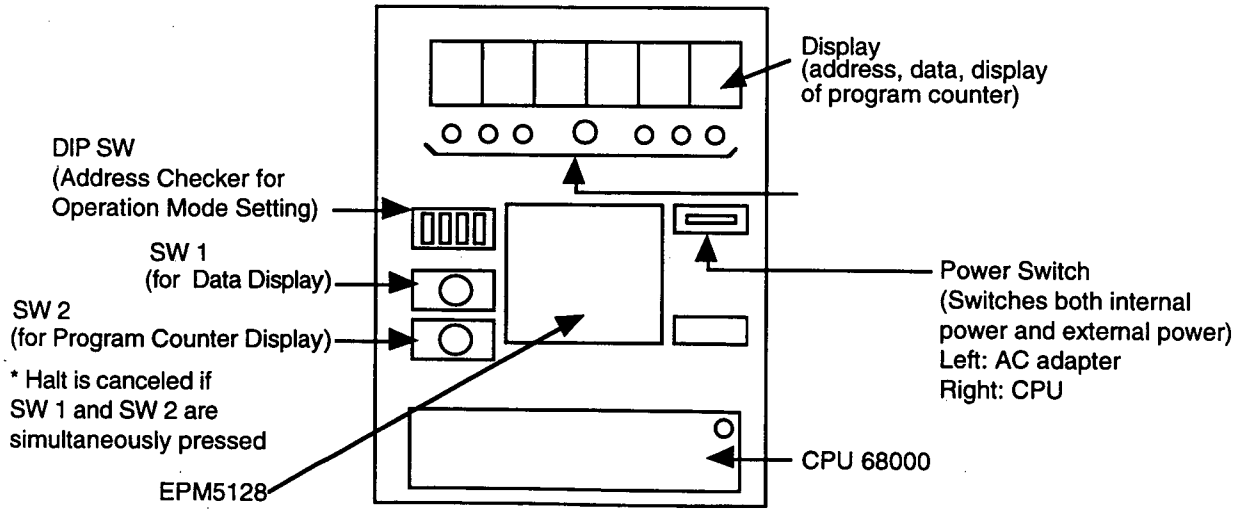
3) The Address Checker corresponds to the following specifications. Note that newer restrictions were also produced.

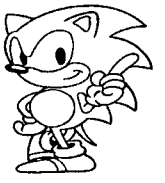
   Corresponding Specifications:
   - ROM16 Mbits or less + Backup Ram
   - ROM16 Mbits or greater (bank switch used) + Backup RAM S32X corespondent

   Precautions:
   Please note that the following items are not detected as errors.
   (a) When writing to the ROM area.
   (b) When accessing an address that is greater than the Backup RAM capacity of the mass production board.
   (c) When an even numbered address is accessed while accessing the Backup RAM.

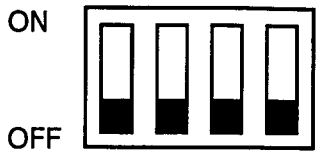
2.0 Function and uses of each part

2.1 Part Names





2.2 DIP SW Initial Settings and Functions



No. 1	ON	When not operating as the address checker, this switches to normal operation even if the CPU socket is not replaced.
	OFF	Operates as an address checker
No. 2	ON	Checks address when in the MD mode
	OFF	Cannot be set
No. 3		Not used
No. 4		Not used

2.3 Usage Method

1) The address checker is attached to the CPU socket of the MD.

2) Set and execute the software. (Confirm that the power lamp is on.)

3) After running all routines of a program and HALT is not turned on, the program is judged as not performing various prohibited items.

4) When HALT is on, look at the display and LED and decide the condition of the program.

5) By pressing switches 1 and 2 at the same time, the program restarts from the address following the spot where the error occurred.

※ To display the address, data, and program counter by applying HALT to the CPU without relevancy to errors, start the address checker after turning DIP SW 1 to ON and press the SW 2 to an arbitrary location. Also, when cancelling HALT at this time, press SW 1 and SW 2 at the same time; when releasing, press SW2 first.

2.4 How to View the Display and LED while in use

1) If the power to the address checker and the MD are turned on, all lamps light up along with the display and LED during the CPU reset interval. Except for power lamp, all lamps turn off when reset ends.

2) If an error occurs, the error address is first displayed on the display screen (6 digits.)
Error data is displayed only while the SW 1 is pressed. (Lower 4 digit.)
The value of the program counter in which an error occurred is displayed only while SW 2 is pressed (6 digit.)

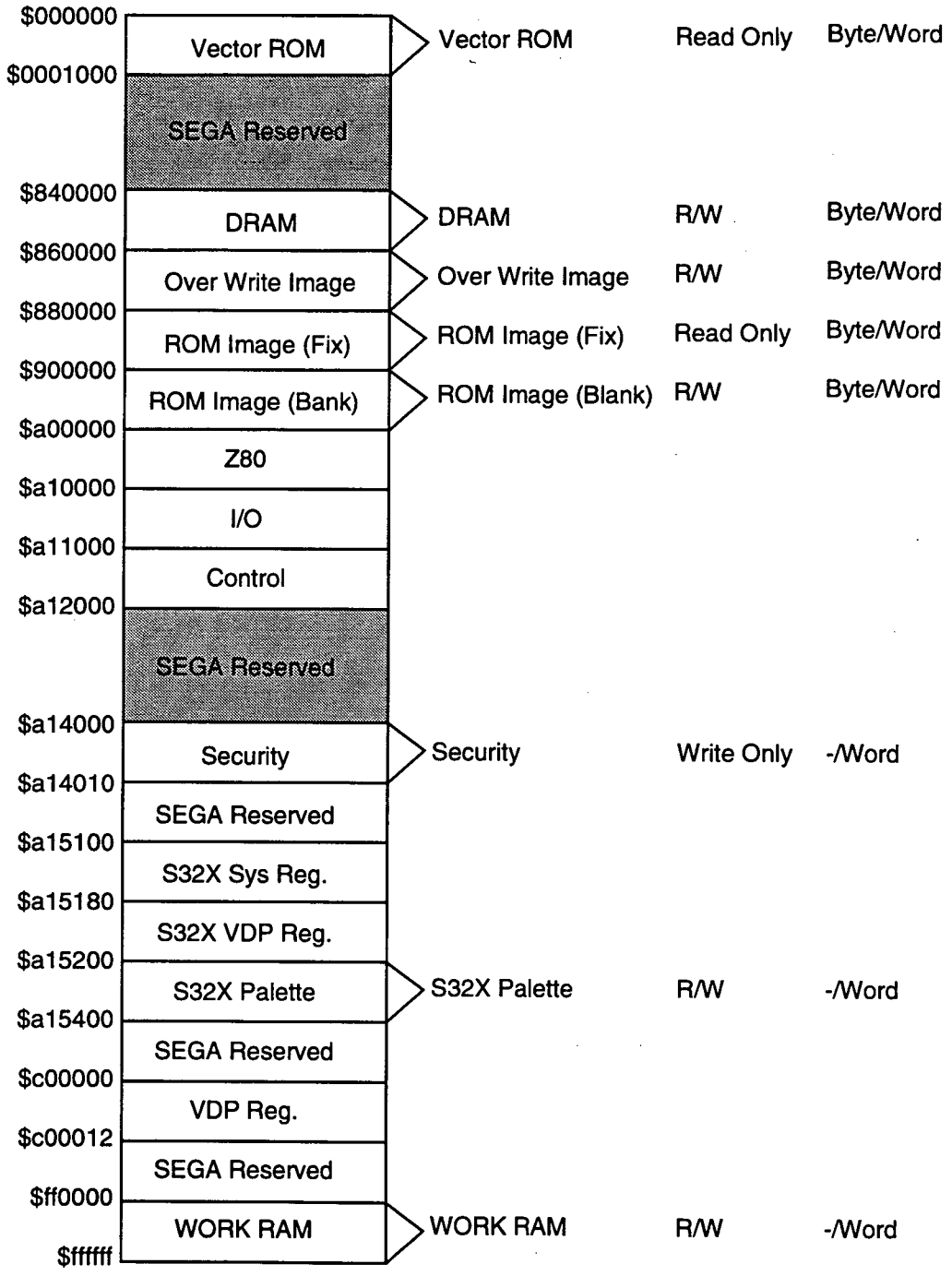
3) LED display (these are lamps that displays CPU I/O conditions when an error occurs.)
   - R/W ON Indicates that the status has been Write
     OFF Indicates that the status has been Read
   - LDS ON Shows the I/O status of low byte data.
     OFF No input and output of data low bytes.
   - UDS ON Shows the status of low byte data.
     OFF No input and output of data high bytes.

* Both LDS and UDS are on during word access.
* For the 68000, because the next operating value is already entered in the executing program counter register, the error is suspected to have been caused before the PC value is actually displayed.
* Because the address checker only checks the mapping check and the value that is set in the register, it does not include any function that discovers bugs hidden in the program.

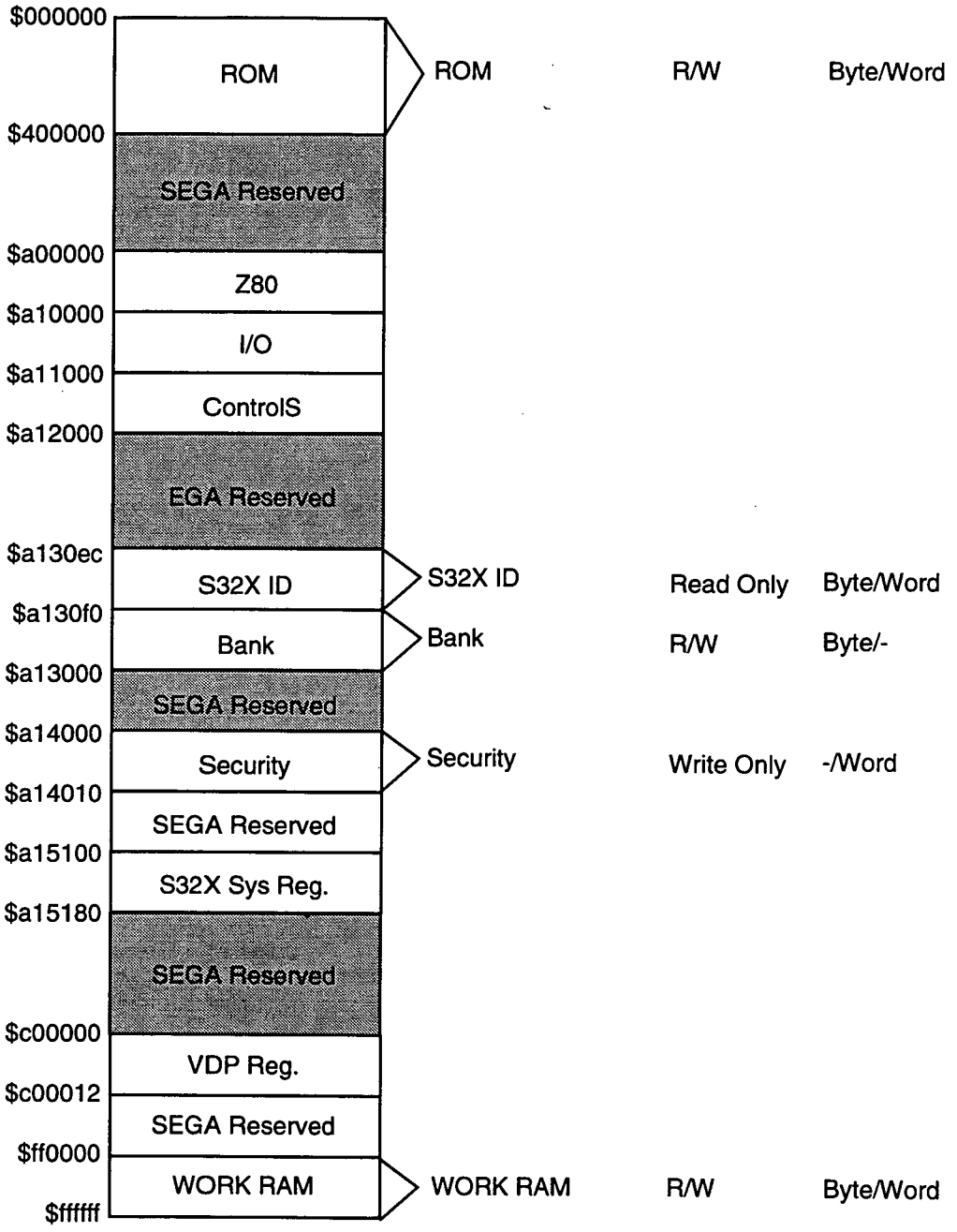
3.0 Checking Items with Address Checker

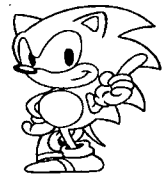
3.1 Memory Mapping

(1) Address Checker Memory Mapping MD+S32X Mode (68000 side)



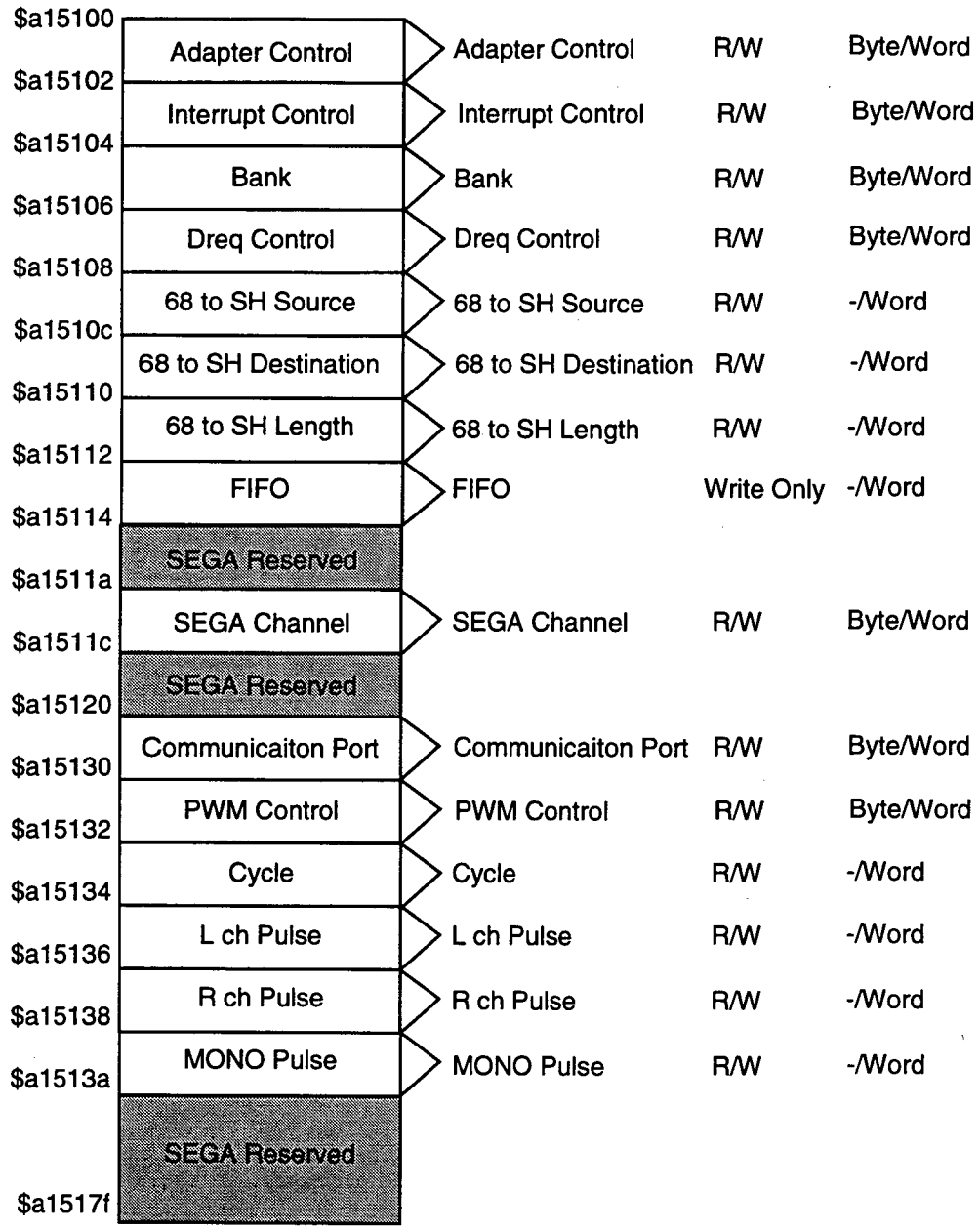
(2) Address CheckeMemory Mapping



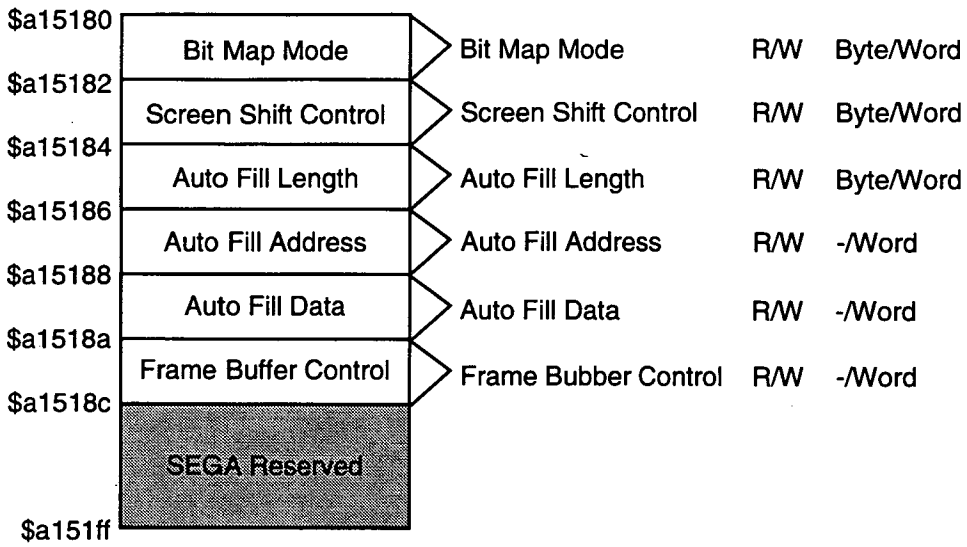


3.2 Register Mapping

(1) Address Checker Register Mapping S32X System Register (68000 side)

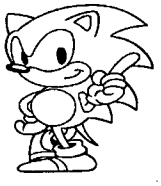


(2) Address Checker Register Mapping S32X VDP Register (68000 side)



3.3 All Register Set Values

When data other than values that can be set for all registers is written, HALT is applied to the main CPU together with the display of an error See the Register Set Value Table of section 5.0,"Supplement" for the values set in the VDP register and the main CPU register.



4.0 Analysis Example when using ICE

Caution ! When analyzing with the ICE, a HALT could apply when an error occurs and the address checker operates. To view the history, make sure to stop the program and press SW1 and SW2 at the same time, then release HALT.

☆ 1, 2 : Error generated part per address checker
(PC 1, 2) : Address checker PC display part

4.1 Analysis Example (1) Register Set Error

1) Access Address : C00004
2) Data (when SW 1 is pressed) : 8008
3) PC (when SW 2 is pressed) : 0601A8
4) ON Lamps : UDS, LDS, R/W

1: HD (History Dumping)

Point	T Address	St	Data	Memory	Opcode	Operand
0014	06019E		207C00C00004	SP	MOVEA.L	#$00C00004, A0
0011	0601A4		30BC8008	SP	MOVE.W	#$8008,(A0)
0008 *1	C00004	MW	8008	SD		
0009 (PC)	0601A8		30BC8124	SP	MOVE.W	#$8124,(A0)

2 : H M (History Map)

Point	T Address	St	Data	Memory	Opcode	
0014	06019E	MR	207C	SP		
0013	0601A0	MR	00C0	SP	MOVEA.L	#$00C00004,A0
0012	0601A2	MR	0004	SP		
0011	0601A4	MR	30BC	SP	MOVE.W	#$8008,(A0)
0010	0601A6	MR	8008	SP		
0009 (PC2)	0601A8	MR	30BC	SP	MOVE.W	
0008 *2	C00004	MW	8008	SD		

Cause: Writing use-prohibited register data.
8008H is written to address $C00004 at the locations of ☆ 1 and 2. (POINT 8)
8008H data is not permitted in the MD Register Set Table. Therefore, errors occur in the address checker.

3 : Disassemble / Source Program

== Supervisor Program Memory ==

Address Code		Mnemonic	
06019E	207C00C00004	MOVEA.L. # $00C00004,A0	;A0←$C00004
0601A4	30BC8008	MOVE.W #$8008,(A0)	;VDP RGSTR SET

4.2 Analysis Example (2) Address Set Error

1) Access Address : C00004
2) Address Set Data (before pressing SW 2) : 4010
3) PC (when SW 2 is pressed) : 0614A8
4) ON Lamps : UDS, LDS, R/W, 2ND

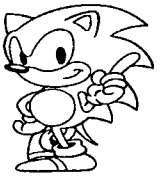
1: HD (History Dumping)

Point	T Address	St	Data	Memory	Opcode	Operand
0016	06149E		23FC4000401000C00004	SP	MOVE.L	#$40004010,$C00004.L
0010	C00004	MW	4000	SD		
0009 *1	C00006	MW	4010	SD		
0011 (PC)	0614A8		2038B3B8	SP	MOVE.L	$FFB3B8.W,D0

2 : H M (History Map)

Point	T Address	St	Data	Memory	Opcode	
0016	06149E	MR	23FC	SP		
0015	0614A0	MR	4000	SP		#$40004010,$C00004.L
0014	0614A2	MR	4010	SP	MOVE.L	
0013	0614A4	MR	00C0	SP		
0012	0614A6	MR	0004	SP		
0011 (PC2)	0614A8	MR	2038	SP	MOVE.L	
0010	C00004	MW	4000	SD		
0009 *2	C00004	MW	4010	SD		

Cause: Writing use-prohibited register data.
$4010 is written to address $C00004 at the locations of ☆ 1 and 2. (POINT 9)
$4010 data is prohibited in the MD address set table. Therefore, an error occurs in the address checker.



3 : D I (Disassemble / Source Program

== Supervisor Program Memory ==

ADDRESS	CODE	Nnemonic
06149E	23FC4000401000C00004	MOVE.L #$40004010,$C0004.L ;ADRS RGSTR SET

5.0 Supplement

Caution !

* Sections 5.1 and 5.2 are the same set value table during the MD and S32X modes.

5.1 Table of VDP Register Set Values

VDP REGISTER #0 ($C00004)

	MSB															LSB
Name	1	0	0	RG4	RG3	RG2	RG1	RG0	0	0	0	IE0	0	M4	M3	0
BINARY	1	0	0	0	0	0	0	0	0	0	0	X	0	1	X	0
HEXA	8				0				0,1				4, 6			

VDP REGISTER #1 ($C00004)

	MSB															LSB
Name	1	0	0	RG4	RG3	RG2	RG1	RG0	0	BLNK	IE0	M1	M2	M5	0	0
BINARY	1	0	0	0	0	0	0	1	0	X	X	X	X	1	0	0
HEXA	8				1				0to7				4, C			

VDP REGISTER #2 ($C00004)

	MSB															LSB
Name	1	0	0	RG4	RG3	RG2	RG1	RG0	0	0	SA15	SA15	SA13	0	0	0
BINARY	1	0	0	0	0	0	1	0	0	0	X	X	X	0	0	0
HEXA	8				2				0to3				0,8			

VDP REGISTER #3 ($C00004)

	MSB															LSB
Name	1	0	0	RG4	RG3	RG2	RG1	RG0	0	0	WD15	WD14	WD13	WD12	WD11	0
BINARY	1	0	0	0	0	0	1	1	0	0	X	X	X	X	X	0
HEXA	8				3				0to3				0,2,4,6,8,A,C,E			

VDP REGISTER #4 ($C00004)

	MSB															LSB
Name	1	0	0	RG4	RG3	RG2	RG1	RG0	0	0	0	0	0	WD15	WD14	WD13
BINARY	1	0	0	0	1	1	0	0	0	0	0	0	0	X	X	X
HEXA	8				4				0				0to7			

VDP REGISTER #5 ($C00004)

	MSB															LSB
Name	1	0	0	RG4	RG3	RG2	RG1	RG0	0	AT15	AT14	AT13	AT12	AT11	AT10	AT9
BINARY	1	0	0	0	0	1	0	1	0	X	X	X	X	X	X	X
HEXA	8				5				0to7				0toF			

VDP REGISTER #6 ($C00004)

	MSB															LSB
Name	1	0	0	RG4	RG3	RG2	RG1	RG0	0	0	0	0	0	0	0	0
BINARY	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
HEXA	8				6				0				0			

VDP REGISTER #7 ($C00004)

	MSB															LSB
Name	1	0	0	RG4	RG3	RG2	RG1	RG0	0	0	CPT1	CPT0	COL3	COL2	COL1	COL0
BINARY	1	0	0	0	0	1	1	1	0	0	X	X	X	X	X	X
HEXA	8				7				0to3				0toF			

VDP REGISTER #8 ($C00004)

	MSB															LSB
Name	1	0	0	RG4	RG3	RG2	RG1	RG0	0	0	0	0	0	0	0	0
BINARY	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
HEXA	8				8				0				0			

VDP REGISTER #9 ($C00004)

	MSB															LSB
Name	1	0	0	RG4	RG3	RG2	RG1	RG0	0	0	0	0	0	0	0	0
BINARY	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
HEXA	8				9				0				0			

VDP REGISTER #10 ($C00004)

	MSB															LSB
Name	1	0	0	RG4	RG3	RG2	RG1	RG0	HIT7	HIT6	HIT5	HIT4	HIT3	HIT2	HIT1	HIT0
BINARY	1	0	0	0	1	0	1	0	X	X	X	X	X	X	X	X
HEXA	8				A				0toF				0toF			

VDP REGISTER #11 ($C00004)

MSB LSB

Name	1	0	0	RG4	RG3	RG2	RG1	RG0	0	0	0	0	IE2	VPSC	HPSC	LSCR
BINARY	1	0	0	0	1	0	1	1	0	0	0	0	X	X	X	X
HEXA	8				B				0				0toF			

VDP REGISTER #12 ($C00004)

MSB LSB

Name	1	0	0	RG4	RG3	RG2	RG1	RG0	PCH2	0	0	0	S/TE	LSM1	LSM0	RS40
BINARY	1	0	0	0	1	1	0	0	A	0	0	0	X	X	X	A
HEXA	8				C				A = 0 : 0 A = 1 : 8				A = 0 : Even number A = 1 : Odd number			

Note: Bit pattern must be PCH2 = RS40.

VDP REGISTER #13 ($C00004)

MSB LSB

Name	1	0	0	RG4	RG3	RG2	RG1	RG0	0	0	HS15	HS14	HS13	HS12	HS11	HS10
BINARY	1	0	0	0	1	1	0	1	0	0	X	X	X	X	X	X
HEXA	8				D				0to3				0toF			

VDP REGISTER #14 ($C00004)

MSB LSB

Name	1	0	0	RG4	RG3	RG2	RG1	RG0	0	0	0	0	0	0	0	0
BINARY	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
HEXA	8				E				0				0			

VDP REGISTER #15 ($C00004)

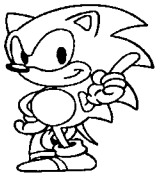
MSB LSB

Name	1	0	0	RG4	RG3	RG2	RG1	RG0	INC7	INC6	INC5	INC4	INC3	INC2	INC1	INC0
BINARY	1	0	0	0	1	1	1	1	X	X	X	X	X	X	X	X
HEXA	8				F				0toF				0toF			

VDP REGISTER #16 ($C00004)

MSB LSB

Name	1	0	0	RG4	RG3	RG2	RG1	RG0	0	0	VSZ1	VSZ0	0	0	HSZ1	HSZ0
BINARY	1	0	0	1	0	0	0	0	0	0	X	X	0	0	X	X
HEXA	9				0				0to3				0to3			



VDP REGISTER #17 ($C00004)

MSB LSB

Name	1	0	0	RG4	RG3	RG2	RG1	RG0	RIGT	0	0	WHP4	WHP3	WHP2	WHP1	WHP0
BINARY	1	0	0	1	0	0	0	1	X	0	0	X	X	X	X	X
HEXA	9				1				0,1,8,9				0toF			

VDP REGISTER #18 ($C00004)

MSB LSB

Name	1	0	0	RG4	RG3	RG2	RG1	RG0	DOWN	0	0	WVP4	WVP3	WVP2	WVP1	WVP0
BINARY	1	0	0	1	0	0	1	0	X	0	0	X	X	X	X	X
HEXA	9				2				0,1,8,9				0toF			

VDP REGISTER #19 ($C00004)

MSB LSB

Name	1	0	0	RG4	RG3	RG2	RG1	RG0	LG7	0LG6	LG5	LG4	LG3	LG2	LG1	LG0
BINARY	1	0	0	1	0	0	1	1	X	X	X	X	X	X	X	X
HEXA	9				3				0toF				0toF			

VDP REGISTER #20 ($C00004)

MSB LSB

Name	1	0	0	RG4	RG3	RG2	RG1	RG0	LG15	0LG14	LG13	LG12	LG11	LG10	LG9	LG8
BINARY	1	0	0	1	0	1	0	0	X	X	X	X	X	X	X	X
HEXA	9				4				0toF				0toF			

VDP REGISTER #21 ($C00004)

MSB LSB

Name	1	0	0	RG4	RG3	RG2	RG1	RG0	SA8	SA7	SA6	SA5	SA4	SA3	SA2	SA1
BINARY	1	0	0	1	0	1	0	1	X	X	X	X	X	X	X	X
HEXA	9				5				0toF				0toF			

VDP REGISTER #22 ($C00004)

MSB LSB

Name	1	0	0	RG4	RG3	RG2	RG1	RG0	SA16	SA15	SA14	SA13	SA12	SA11	SA10	SA9
BINARY	1	0	0	1	0	1	1	0	X	X	X	X	X	X	X	X
HEXA	9				6				0toF				0toF			

VDP REGISTER #23 ($C00004) DMA ROM TO VRAM

MSB LSB

Name	1	0	0	RG4	RG3	RG2	RG1	RG0	DMD1	DMD0	BA22	BA21	BA20	BA19	BA18	BA17
BINARY	1	0	0	1	0	1	1	1	0	0	0	X	X	X	X	X
HEXA	9				7				0,1				0toF			

VDP REGISTER #23 ($C00004) DMA WORK-RAM TO VRAM

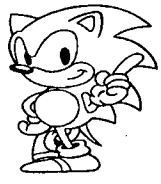
MSB LSB

Name	1	0	0	RG4	RG3	RG2	RG1	RG0	DMD1	DMD0	BA22	BA21	BA20	BA19	BA18	BA17
BINARY	1	0	0	1	0	1	1	1	0	1	1	1	1	1	1	1
HEXA	9				7				7				F			

VDP REGISTER #23 ($C00004) DMA FILL VRAM/VRAM COPY

MSB LSB

Name	1	0	0	RG4	RG3	RG2	RG1	RG0	DMD1	DMD0	BA22	BA21	BA20	BA19	BA18	BA17
BINARY	1	0	0	1	0	1	1	1	1	X	0	0	0	0	0	0
HEXA	9				7				8,C				0toF			



5.2 VDP Address Set Value Table

CRAM READ ($C00004)

MSB — Upper WORD — LSB

Name	CD1	CD0	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
BINARY	0	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X
HEXA	0				0				0to7				0toF			

MSB — Lower WORD — LSB

Name	0	0	0	0	0	0	0	0	CD5	CD4	CD3	CD2	0	A16	A15	A14
BINARY	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
HEXA	0				0				2				0			

CRAM WRITE ($C00004)

MSB — Upper WORD — LSB

Name	CD1	CD0	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
BINARY	1	1	0	0	0	0	0	0	0	X	X	X	X	X	X	X
HEXA	C				0				0to7				0toF			

MSB Lower WORD LSB

Name	0	0	0	0	0	0	0	0	CD5	CD4	CD3	CD2	0	A16	A15	A14
BINARY	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HEXA	0				0				0				0			

CRAM DMA ADDRESS SET ($C00004)

MSB — Upper WORD — LSB

Name	CD1	CD0	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
BINARY	1	1	0	0	0	0	0	0	0	X	X	X	X	X	X	X
HEXA	C				0				0to7				0toF			

MSB — Lower WORD — LSB

Name	0	0	0	0	0	0	0	0	CD5	CD4	CD3	CD2	0	A16	A15	A14
BINARY	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
HEXA	0				0				8				0			

VRAM READ ($C00004)

MSB Upper WORD LSB

Name	CD1	CD0	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
BINARY	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X
HEXA	0to3				0toF				0toF				0toF			

MSB Lower WORD LSB

Name	0	0	0	0	0	0	0	0	CD5	CD4	CD3	CD2	0	A16	A15	A14
BINARY	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X
HEXA	0				0				0				0to3			

VRAM WRITE ($C00004)

MSB Upper WORD LSB

Name	CD1	CD0	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
BINARY	0	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X
HEXA	0				0				7				0toF			

MSB Lower WORD LSB

Name	0	0	0	0	0	0	0	0	CD5	CD4	CD3	CD2	0	A16	A15	A14
BINARY	0	0	0	0	0	0	0	0	0	0	0	1	0	0	X	X
HEXA	0				0				1				0			

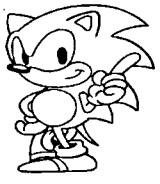
VRAM DMA ADDRESS SET ($C00004)

MSB Upper WORD LSB

Name	CD1	CD0	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
BINARY	0	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X
HEXA	4,5,6,7				0toF				0toF				0toF			

MSB Lower WORD LSB

Name	0	0	0	0	0	0	0	0	CD5	CD4	CD3	CD2	0	A16	A15	A14
BINARY	0	0	0	0	0	0	0	0	1	0	0	0	0	0	X	X
HEXA	0				0				8				0to3			



VSRAM READ ($C00004)

MSB — Upper WORD — LSB

Name	CD1	CD0	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
BINARY	0	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X
HEXA	0				0				0to7				0toF			

MSB — Lower WORD — LSB

Name	0	0	0	0	0	0	0	0	CD5	CD4	CD3	CD2	0	A16	A15	A14
BINARY	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
HEXA	0				0				1				0			

VSRAM WRITE ($C00004)

MSB — Upper WORD — LSB

Name	CD1	CD0	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
BINARY	0	1	0	0	0	0	0	0	0	X	X	X	X	X	X	X
HEXA	4				0				0to7				0toF			

MSB — Lower WORD — LSB

Name	0	0	0	0	0	0	0	0	CD5	CD4	CD3	CD2	0	A16	A15	A14
BINARY	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
HEXA	0				0				1				0			

VSRAM DMA ADDRESS SET ($C00004)

MSB — Upper WORD — LSB

Name	CD1	CD0	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
BINARY	0	1	0	0	0	0	0	0	0	X	X	X	X	X	X	X
HEXA	4				0				0to7				0toF			

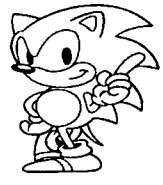
MSB — Lower WORD — LSB

Name	0	0	0	0	0	0	0	0	CD5	CD4	CD3	CD2	0	A16	A15	A14
BINARY	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0
HEXA	0				0				9				0			

VSRAM COPY ($C00004)

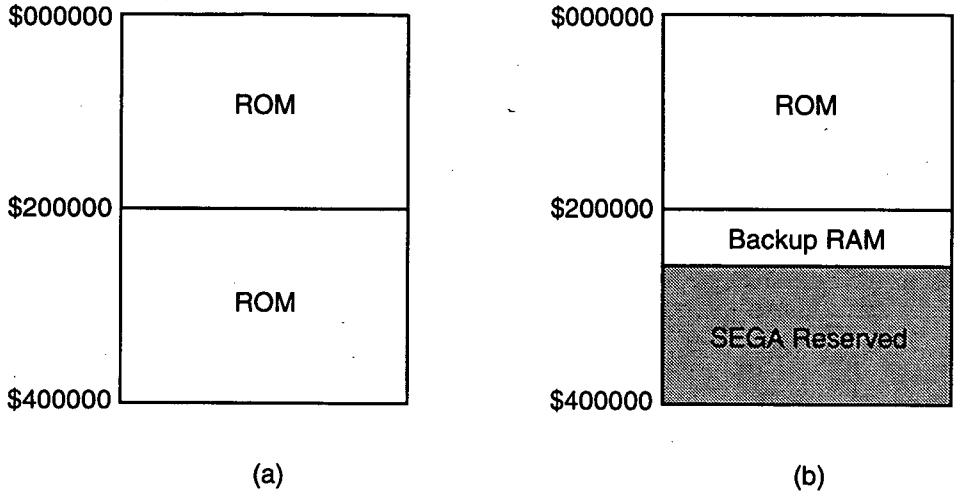
	MSB							Lower WORD							LSB
Name	CD1	CD0	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
BINARY	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X
HEXA	0to3				0toF				0toF				0toF			

	MSB							Lower WORD							LSB
Name	0	0	0	0	0	0	0	0	CD5	CD4	CD3	CD2	0	A16	A15	A14
BINARY	0	0	0	0	0	0	0	0	1	1	0	0	0	0	X	X
HEXA	0				0				C				0to3			



Addendum

1. P 2 1-(3) Precautions Supplement of (b)



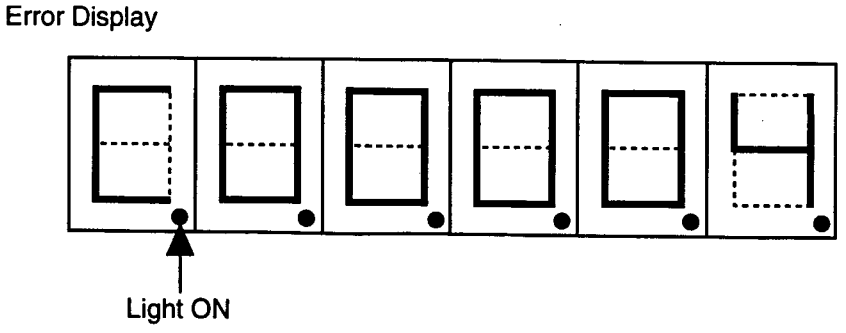
Two mappings exist, as mentioned above, when bank switching is used. However, the address checker is not able to distinguish the difference between (a) and (b). Thus, even if the area shown by the slanted lines (b) is accessed, it is not shown as an error because it is recognized equally with the ROM area in (a).

2. Additional Functions

When all dots in the lower area of the display are lit.

There are two methods for access to the VDP control port ($C00004), register set (1 word access) and address set (2 word access). When all display dots are lit, it shows abnormal access by address set access.

Example of Error Display



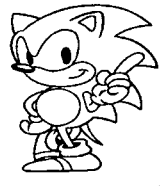
When this error is displayed, the following abnormal access is considered.

(1) The access modes of the 1st word and 2nd word do not match.
Although an address set in the control port must be accessed by 2 word set, set of the access mode in which 1st and 2nd word are each dif ferent is performed.

Example: In the case when VRAM READ is set in the 1st word and VRAM WRITE is set in the 2nd word.

(2) Only the 1st Word is set.
Among 2 word set of the address set, 2nd word set is not performed and register set is performed in the control port after the 1st word is set.

Note: When an address set error occurs, check is not possible in address checker units. Errors should be analyzed with the ICE history function.



READER CORRECTION/COMMENT SHEET

Keep us updated!

If you should come across any incorrect or outdated information while reading through the attached document, or come up with any questions or comments, please let us know so that we can make the required changes in subsequent revisions. Simply fill out all information below and return this form to the Developer Technical Support Manager at the address below. Please make more copies of this form if more space is needed. Thank you.

General Information:

Your Name _____ **Phone** _____

Document number MAR-57-102894 **Date** _____

Document name MEGA Drive/32X Address Checker Specification

Corrections:

Chpt.	pg. #	Correction

Questions/comments: _____

_____

_____

_____

Where to send your corrections:

Fax: (415) 802-3963
Attn: Manager,
Developer Technical Support

Mail: SEGA OF AMERICA
Attn: Manager,
Developer Technical Support
275 Shoreline Dr. Ste 500
Redwood City, CA 94065