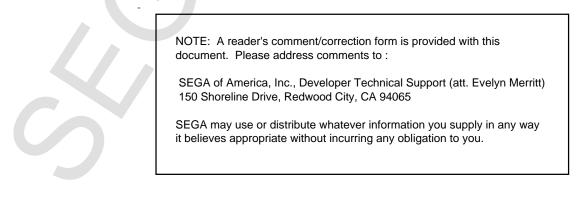
#### **General Notice**

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 nondisclosure and confidentiality agreement signed separately and in the possession of SEGA. If you have not signed such a nondisclosure 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, copyrights, trademarks, or other intellectual property rights of SEGA Enterprises, Ltd., SEGA of America, Inc., or any third party.
- 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 described 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.





# SEGA SATURN Address Checker Reference Manual

Doc. # ST-254-B-110395

© 1995-96 SEGA. All Rights Reserved.



REVISION-1 • Description of Commands • Technical Materials



SI Electronics 2-28-16 Shimomaruko, Ota-ku, Tokyo 146 TEL: 03-3756-4111 TEL: 03-3756-4114 (Tech Development) FAX: 03-3756-5377

Copyright SI Electronics Ltd. 1995. All rights reserved.

#### Notes:

- (1) This manual is copyrighted by SI Electronics Ltd.. Unauthorized reproduction in entirety or in part is prohibited by law.
- (2) Our best efforts have gone into preparation of this manual, but we welcome questions regarding the contents. Contents subject to change without notice.
- (3) Regardless of (2), no responsibility is assumed for any consequences of operation.

MS DOS is a registered trademark of Microsoft Corporation, USA.

Other program, system, and CPU names are trademarks registered trademarks of their respective manufacturers.

SI Electronics 2-28-16 Shimomaruko, Ota-ku, Tokyo 146 TEL: 03-3756-4111 TEL: 03-3756-4114 (Tech Development) FAX: 03-3756-5377



#### Introduction

This manual provides a description of commands used with the SEGA SATURN Address Checker, and technical specifications.

See the separate *SEGA SATURN Address Checker Operation Manual* (Doc. # ST-254-A-110395) for a description of Address Checker setup, program installation, and operation.

#### Notes and Symbols Used in This Manual

Explanatory symbol indicating a one-character space. Only used
to denote a space when particularly important.
Explanatory symbol indicating pressing the Enter key (carriage return key)
Explanatory symbol indicating a character to be actually input
from the keyboard.
ASCII symbol termed 'backslash' and assigned code 5C. In JIS
code,
displays '¥'(yen symbol).
Indicates pressing the 'A' key while holding down the control key.
Indicates, for example (^A) $\rightarrow$ 01, (^B) $\rightarrow$ 02, (^Z) $\rightarrow$ 1A.
Special keys are denoted by < >.
For example, besides 'U', the space key may be indicated by
<space>.</space>
'xxxxH' indicates that the value 'xxxx' is hexadecimal.



#### **Table of Contents**

Table Of	Content	5	
		Specifications	
1.1 SE	EGA System	n Specifications	9
1.2 Ac	Idress Cheo	cker Function Specifications	10
2 Detaile	d Descriptio	on of Commands	11
2.1	!	Execute Commands in the History Buffer	12
2.2	:	Edit and Execute Commands in the History Buffer	
2.3	@	MS-DOS Call and Command Shell Execution	
2.4	<	Command File Execution	
2.5	ALIAS	Define an Alias	
2.6	?ALIAS	Display an Alias	18
2.7	_ALIAS	Delete an Alias	
2.8	_ALIASU	Delete User-Defined Alias	20
2.9	AUTO	Execute Auto Re-test	21
2.10	BELL	Bell Sound	23
2.11	BB	Set Path Breakpoint	24
2.12	?BB	Display Path Breakpoint	25
2.13	_BB	Delete Path Breakpoint	26
2.14	CAT	Display File Contents	
2.15	CLS	Clear Command Window Screen	28
2.16	ECHO	Display Character String	29
2.17	_ECHO	Suppress Display	30
2.18	FK	Define Function Keys	
2.19	?FK	Display Function Key Definition Character String	
2.20	_FK	Delete Function Key Definition	33
2.21	GO	Execute Program	
2.22	HELP	Display Command Syntax	
2.23	LOG	Start Log Acquisition	
2-24	?LOG	Display Log File Name	
2.25	_LOG	End Log Acquisition	
2.26	MAN	Manual Display	
2.27	MIDX	Create Index File for Message Display	
2.28	MORE	MORE Format Display <filter command=""></filter>	
2.29	QUIT	Quit Address Checker	
2.30	REM	Comment Line	
2.31	RIDX	Read Index File for Message Display	
2.32	SET	Set Address Checker Operating Environment	
2.33	?SET	Display Address Checker Operating Environment	
2.34	STOP	Stop	
2.35	TDMP	Display Trace Data	
2.36	TRC	Set Trace Conditions	
2.37	?TRC	Display Trace Conditions	
2.38	_TRC	Delete Trace Conditions	
2.39	TSET	Set Trigger Detection Condition	
2.40	VRAM	Store Display Contents to a File	00
3. Detailed	d Descriptio	n of TRCMAP.DAT Contents	56
4. Error M	essages		58

### Appendices

SATURN System Address Map		
SMPC Register Map	64	
DCC Register Map		
CDROM Register Map	66	
SCSP Register Map	72	
VDP1 Register Map		
VDP2 Register Map		
SCU Register Map		

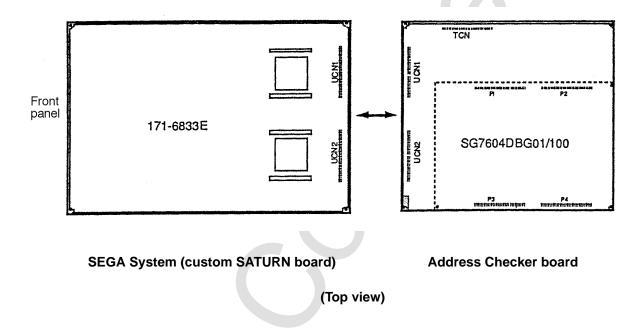


# 1. Address Checker Specifications

The Address Checker is used to perform address checks and debugging of programs such as games that run on the SEGA system (custom SATURN board 17106833E). For this purpose, the SEGA system and Address Checker board employ C-BUS connections.

#### 1.1 SEGA System Specifications

The Address Checker structure is illustrated below.



The SEGA System and Address Checker board are connected through the UCN1 and UCN2 connectors shown in the drawing.

#### **1.2 Address Checker Function Specifications**

#### I Command Specifications

Principal commands that can be used with the Address Checker are given below. See section §2 "Detailed Description of Commands" for details of commands.

- (1) GO command: start execution of user program (always reset before starting)
- (2) SET command: specifies bus master, specifies break at TRG detection
- (3) TRC command: specifies history acquisition conditions
- (4) TDMP command: displays history acquisition contents
- (5) BB command: specifies bus break conditions
- (6) ABORT command: aborts user program

#### II Address Error Detection (TRG Detection) Specifications

TRG detection is performed under the following conditions:

- (1) Free access area: no TRG detection, accessible area
- (2) Detection 1 area: area without memory allocation During READ: read not allowed During WRITE: write not allowed
- (3) Detection 2 area: VDP1, SCSP, SMPC areas
  When free bits and used bits are mixed
  During READ: unconditional access (free access)
  During WRITE: permits only "0" writes to free bits (bit-determined for D15 D0)
  When free bits and used bits are not mixed (free bit string=path width)
  During READ: unconditional access (free access)
  During WRITE: write not allowed
- (4) Detection 3 area: SCU, CDROM, VDP2 areas
  When free bits and used bits are mixed
  During READ: unconditional access (free access)
  During WRITE: unconditional access (free access)
  When free bits and used bits are not mixed (free bit string=path width)
  During READ: unconditional access (free access)
  During READ: unconditional access (free access)
  During WRITE: write not allowed
- (5) 5FE00AC'H 5FE00AF'H (in SCU area) During READ: read not allowed During WRITE: write not allowed



# 2. Detailed Descriptions of Commands

The commands are described in detail in this section. The format states the command syntax. The following symbols are used in the format.

$\{\mathbf{x} \mid \mathbf{y} \mid \mathbf{z}\}$	Specify one of the items x, y and z that are within the brackets.
[x]	The item within the square brackets can be omitted.
х	Parameters can be entered one after the other.
<>	This is used to make the parameters easier to spot and has no special
	meaning.

#### 2.1 ! Execute Commands in the History Buffer

2.1 ! E	xecute Commands	in the History Buffer
Format	! [ {numeral charact	er string} ]
Function	Executes a command r	ecorded in the command history.
Description	Searches for and execu as follows.	tes the command recorded in the command history
	! numeral	Executes the nth command recorded in the com mand history. If the value is negative, executes the command located that number of commands previous to the current command.
	! character string	Executes the last command executed that starts with the character string.
	! The command history	Executes the previously entered command.
Example	2	can be displayed with the HIS command. nand in the command history.
1	>!10	
	Execute the last BB0 cc >!BB0	ommand executed.



#### 2.2 : Edit and Execute Commands in the History Buffer

Format : [ {numeral|character string} ] Edits and executes a command recorded in the command history. Function Searches for and edits a command recorded in the command history as Description follows and executes it upon entry of a carriage return. Displays the nth command recorded in the ; numeral numerical command history and allows editing and command entry. If the value is negative, executes the command located that number of commands previous to the current command. Displays the last command executed that starts ; character string with the character string and allows editing and command entry. Displays the previously entered command and ; allows editing and command entry. The command history can be displayed with the HIS command. Execute the 10th command in the command history after editing: Example >:10 Execute the last BB1 command executed after editing. >:BB1

#### 2.3 @ MS-DOS Call and Command Shell Execution

Format @ [[/P]<space><MS-DOS command>] where /P indicates that key input for returning to the Address Checker is omitted.

Function Calls COMMAND.COM of MS-DOS.

Description Shifts control to MS-DOS as described below.

When <MS-DOS command> is not specified, the screen display disappears and the command line entry mode starts. When an MS-DOS command is entered in this state, COMMAND.COM is called and control shifts to MS-DOS. Upon returning from MS-DOS, the command line input mode resumes. Enter the EXIT command or the escape key to return. The command line input mode is managed by the Address Checker, so the command history and command editing are available.

When <MS-DOS command> is specified, the specified MS-DOS command is executed. Hitting any key thereafter will return to the Address Checker, which will wait for command entry. When option /P is specified, control returns to the Address Checker without any key input. A space is necessary between "@" and "<MS-DOS command>".

The amount of Address Checker that is MS-DOS resident is at least 40 Kbyte, provided there is a sufficient margin in protect mode memory. When "Not enough memory for environment" is displayed when a batch command is run, set the following environment variable before starting up the Address Checker.

A>SET IENV=<n>; <n> is a decimal numeral The function of the environment variable IENV depends on the MS-DOS version. In Ver. 3.1, specify 10 or more paragraph units where each is 16 bytes.



Example	
Zminpre	A>SET IENV=40 ; Sets an environment variable area of 640 bytes Starting from Ver. 3.3., specify 160 or more in byte units.
Example	A>SET IENV=640 ; Sets an environment variable area of 640 bytes The IENV function is not valid in MS-DOS ver. 2 or earlier. When COMMAND.COM is not on the MS-DOS path, MS-DOS commands cannot be executed.
Example	Display the MS-DOS directory: >@ DIR (Executes the DIR command of MS-DOS)
	(The DIR command display is presented.)
	<pre>push any key! (To return) When entry includes a reserved character of the Address Checker (;, &lt;, &gt;,  , -), surround it in quotation marks. &gt;@ "dir   sort &gt;&gt;logging.dat" Execute an MS-DOS command from the command line input mode: &gt;@ (Command input mode) A:\&gt;TIME (Executes various MS-DOS commands) A:\&gt;EXIT (Return with the EXIT command) &gt;G0 (Address Checker commands usable again.)</pre>

#### 2.4 < Command File Execution

Format < <filename>

**Function** Executes a command file.

**Description** Executes the specified <filename> as a command file.

Command files line up commands and parameters the same way they are when Address Checker commands are input by key.

#### Notes:

• The following commands cannot be executed in command files.

<
AUTO
\_ALIASU
\_CMDU
Commands for macros
(IF {, } ELSE {, WHILE{,}, BREAK)

• Log files obtained with the LOG command can be run as command files. The difference from AUTO commands is in whether the results of execution can be verified. See the example.

• When control codes for window control (such as control Z) are found in the file data, windows can be controlled.

# **Example** When a log is obtained from the log file, it can be executed with either the AUTO command or the < command.

>LOG ALOG	(Starts getting log)
>TRC ALL	
>GO	
>_LOG	(Finishes getting log)
>> ALOG.LOG	(Runs the command input recording file of the log file as a command file)



### 2.5 ALIAS Define an Alias

Format	ALIAS [ <alias> <character string=""> ]</character></alias>
Function	Defines aliases for specified command character strings.
Description	Defines the specified <character string=""> as <alias>. Thereafter, when the alias character string is entered as command input, the specified character string is run instead.</alias></character>
	When the parameters are omitted, the <alias> prompt is displayed, so entering the <alias> and <character string=""> can define the alias. See the example.</character></alias></alias>
Example	Define GO command under alias "RUN":
	>ALIAS RUN GO
	Save alias definition in a temporary file and read the file to redefine them:
	>ALIAS > ALIAS.SAV
	>ALIAS < ALIAS.SAV

#### 2.6 ?ALIAS Display an Alias

Format ?ALIAS [ <alias> ... ]

**Function** Displays the character string defined under the alias.

**Description** Displays the character string defined under the alias. When the alias is omitted, all defined aliases are displayed. When no aliases are defined, "alias not found" is displayed.

ExampleDisplay the character strings defined under CLEAR and RUN:<br/>>?ALIAS CLEAR RUN <CR>



### 2.7 \_ALIAS Delete an Alias

Format	_ALIAS [ <alias> ]</alias>
Function	Deletes the alias definitions.
Description	Deletes the alias definitions. When the alias is omitted, all defined charac- ter strings are deleted.
	Note:
	• When the alias is omitted, aliases defined (in BMINIT.DAT) during startup are also
	deleted. To delete all aliases the user has defined, use the _ALIASU command.
Example	Delete the already defined alias commands CLEAR and RUN:
	>_ALIAS CLEAR RUN <cr></cr>

### 2.8 \_ALIASU Delete User-Defined Alias

Format	_ALIASU
Function	Deletes all user-defined alias definitions.
Description	Deletes all user-defined alias definitions.
	<b>Note:</b> This command is defined by the ALIAS command.
Example	Delete the aliases defined by the user:



#### 2.9 AUTO Execute Auto Retest

Format Function	AUTO [ {/I /V /E} ] <filename> Where I: Log file command execution (default value) V: Verify execution results E: Display nonmatches of execution results Executes an automatic retest.</filename>
Description	Automatically re-runs a test after a first test is performed. Results can be
200017000	compared.
	<filename> specifies the name of the log file specified by the LOG command. See the example. This command opens a file that adds the extension ".LOG" to the name of the specified log file. It reads the contents of the file and automatically executes it as an Address Checker command. It can also automatically compare the results of execution to the contents of the log file (the log file name plus the extension ".OUT"). The following can be specified as options. I: Execute only the command without verifying results. The results of execution are displayed (default value) V: Display the results of execution while verifying. Also displays the locations of verification errors. E: Display only the locations of verification errors.</filename>
	Notes: • The following commands connet he evented in an AUTO command
	<ul> <li>The following commands cannot be executed in an AUTO command.</li> <li></li> <li>AUTO</li> </ul>
	_ALIASU _CMDU
	Commands for macros (IF {, } ELSE {, WHILE{,}, BREAK)

**Example** Record an executed command in the log file and automatically re-run the command:

>LOG LOGSAV	(Gets log. No extension added to file name.)
	(Executes various commands.)
>_LOG	(Finishes getting log)

>AUTO LOGSAV (Automatically executes a re-test.)

Display the results of execution while verifying. When the results of execution do not match, the result for the nonmatching lines and the log information are displayed.

>AUTO/V LOGSAV

.....

<result> (Execution results where nonmatches occurred.)

<log file> (Log contents where nonmatches occurred.)

.....

Save only the locations of nonmatches in a file. >AUTO/E LOGA > ERRLOG



#### 2.10 BELL Bell Sound

Format BELL Sounds a bell noise. Function Sounds a bell noise. Description If the contents of register R0 is 0, sound a bell noise: Example >cmd r0bell ?if { b\*.r0 == 0 ?bell ?} ?. ?macro name ? . >

#### 2.11 BB Set Path Break Point

Format	BBn DA/ <address></address>
	where

n: Break point number (0 to 3)

**Function** Sets a bus break point.

**Description** Records the specified address as a bus break condition. Up to 4 bus break points can be specified (0 to 3).

Example Break when address 1000 is accessed: >BB0 DA/1000



#### 2.12 ?BB Display Path Break Point

Format ?BB [n]

**Function** Displays the bus break points.

**Description** Displays the specified bus break point conditions. When n (break point number) is not specified, all bus break conditions are displayed.

Example Display all bus break conditions: >?BB

#### 2.13 \_BB Delete Path Break Point

Format \_BB [n]

**Function** Deletes the bus break points.

**Description** Deletes the specified bus break point conditions. When n (break point number) is not specified, all bus break conditions are deleted.

Example Delete the first bus break point: >\_BB1



#### 2.14 CAT Display File Contents

Format CAT <filename>

**Function** Displays the file contents.

**Description** Displays the contents of the specified file.

**Example** Display the contents of file FILE.DAT: >CAT FILE.DAT

#### 2.15 CLS Clear Command Window Screen

Format CLS [  $\{1|2|3\}$  ] where

- - 1: Command window screen clear 2: Screen refresh (same as [CTRL-])
  - 3: Command window screen clear and screen refresh (default)
- Clears the command window and refreshes the Address Checker screen. Function
- Clears the command display window and moves the cursor to the begin-Description ning of the command window.

Example

>CLS



### 2.16 ECHO Display Character String

Format	ECHO [ <character string="">]</character>		
Function	Displays a character string in the command window.		
Description	Displays the specified <character string=""> in the command window. Spaces can be used if surrounded by quotations marks ("). Use the charac- ters ¥n to get a carriage return. When no character string is specified as the parameter, the suppression of the display specified with the _ECHO com- mand is canceled.</character>		
Example	>ECHO " <thi< th=""><th>aracter string. <sup>S IS THE MESSAGE&gt;¥n"</sup> uppression of displays in the command window. (Nothing is displayed even when a command is entered.) (Cancels the suppression of display by input of an ECHO command. The ECHO command itself is not displayed.) (Now it is in the normal echo back and display mode.)</th></thi<>	aracter string. <sup>S IS THE MESSAGE&gt;¥n"</sup> uppression of displays in the command window. (Nothing is displayed even when a command is entered.) (Cancels the suppression of display by input of an ECHO command. The ECHO command itself is not displayed.) (Now it is in the normal echo back and display mode.)	

### 2.17 \_ECHO Suppress Display

Format	_ECHO		
Function	Suppresses display in the command window.		
Description	Restrains echo back and display of command execution results in the command window. The display suppression is canceled by the ECHO command.		
Example	Suppress display in the command window, then cancel. >_ECHO (Nothing is displayed even when a command is entered.) (ECHO) (Cancels the suppression of display by input of an ECHO command. The ECHO command itself is not displayed.) > (Now it is in the normal echo back and display mode.)		



### 2.18 FK Define Function Keys

2.18 FK	Define Function Keys
Format	FK <numeral> <character string=""> where</character></numeral>
	<numeral>:1 to 20 <character string="">: Up to 15 characters</character></numeral>
Function	Sets the function keys.
Description	Defines a character string to the function key of the specified number. <numeral> values 1 through 10 correspond to function keys F1 through F10. <numeral> values 11 through 20 correspond to Shift F1 through Shift F10.</numeral></numeral>
	Spaces can be used if surrounded by quotations marks ("). Use the characters $y_r$ to get a carriage return. The character string can be up to 15 characters.
	<b>Note:</b> The FK command of BMINIT.DAT defines the initial settings of the function keys. The contents of this file can be changed with an editor to set the function keys as the user desires.
Example	Define $GO < CR >$ for $F \bullet 1$ :
	>FK 1 "GO¥r"

# 2.19 ?FK Display Function Key Definition Character String

Format	<pre>?FK [<numeral> ] where      <numeral>: 1 to 20</numeral></numeral></pre>		
Function	Displays the character strings defined for the function keys.		
Description	Displays the character string defined for the specified function key. <numeral> values 1 through 10 correspond to function keys F1 through F10. <numeral> values 11 through 20 correspond to Shift F1 through Shift F10. When no number is specified, character string for all function keys are displayed.</numeral></numeral>		
Example	Display the character string defined for F10: >?FK 10 FK10 "GO¥r"		



### 2.20 \_FK Delete Function Key Definition

2.20 _fk	Delete Function Key Definition
Format	_FK <numeral> [ ] where</numeral>
	<numeral>: 1 to 20</numeral>
Function	Deletes function key settings.
Description	Deletes the function key definition for the specified function key. <numeral> values 1 through 10 correspond to function keys F1 through F10. <numeral> values 11 through 20 correspond to Shift F1 through Shift F10.</numeral></numeral>
F10. <numeral> values 11 through 20 correspond to Shift F1 through Shif</numeral>	

### 2.21 GO Execute Program

Format

**Function** Executes programs.

GO

**Description** Clears resets and executes programs.

Example Execute: >GO



#### 2.22 HELP Display Command Syntax

Format HELP [ {<command | KEY | EXPR } ]

**Function** Displays the command syntax.

**Description** When <command> is omitted, a list of commands from the manual is displayed and the cursor can be used to select one whose syntax to display. When KEY is specified, an explanation of the control keys and the like for key input is displayed. When EXPR is specified, an explanation of the expression is displayed.

To specify a command for a macro in <command name>, specify IF, WHILE OR BREAK. When <command> is specified, that command's syntax is displayed. The help window is used for the display, so the syntax can be viewed while inputting the command.

The command list is recorded in BM.HLP as text, so it may be freely added to or changed.

- **Note:** Enclose the command file execution command "<" in quotation marks (i.e., HELP "<").
- **Example** Display the GO command syntax: >HELP GO

Display a command selected from the command list: >HELP

# 2.23 LOG Start Log Acquisition

Format	LOG <filename></filename>		
Function	Records emulator commands and display results in a file.		
Description	Saves key input and its display results as log information in a file (log file). Do not place an extension on the file name. When the log is fetched, an automatic re-test can be run using the AUTO command. See the example.		
<b>Note</b> Command input and execution results are recorded in a file <filen Command input alone is recorded in a file <filename>.OUT.</filename></filen 			
Example	Record an executed test: >LOG LOGSAV 	d command in the log file and automatically re-run the (Gets log. No extension added to file name.) (Executes various commands.) (Finishes getting log) (Displays the previous execution results.) (Automatically executes a re-test.)	



### 2.24 ?LOG Display Log File Name

Format ?LOG

**Function** Displays the file being logged.

**Description** If logging is currently in progress, the log file name is displayed. If not currently logging, nothing is displayed.

Example

>LOG LOGSAV	(Gets log.)
	(Executes various commands.)
>?LOG	(Displays the log file name.)
LOG LOGSAV	

# 2.25 \_LOG End Log Acquisition

Format	_LOG
Function	Finishes getting the log.
Description	Finishes getting the log.
Example	Finished getting the log: >_LOG



# 2.26 MAN Manual Display

2.26 MAN	Manual Display
Format	MAN [ <command name=""/> ]
Function	Displays the contents of the manual.
Description	Displays the contents of the manual about the specified command. When <command/> is omitted, select an item with the cursor from the manual or command lists to display the manual contents.
	When KEY is specified, an explanation of the control keys and the like for key input is displayed. When EXPR is specified, an explanation of the expression is displayed. To specify a command for a macro in <command name=""/> , specify IF, WHILE OF BREAK.
	When <command/> is specified, the manual section for that command is displayed. When MAN <command/>   MORE is input, the manual is displayed a page at a time.
	Searches for the specified command start with system commands and then goes to user commands if the command is not found. The command list is recorded in BM.HLP as text, so it may be freely added to or changed.
	<b>Note:</b> When the command "<" is specified in command file execution, enclose it in quotation marks (i.e., MAN "<").
Example	Display the MAP command manual: >MAN TDMP   MORE
	Display after selecting from the command list: >MAN

# 2.27 MIDX Create Index File for Message Display

Format	MIDX { MAN   ERR } MAN: System manual file ERR: Error message file	
Function	Creates an index file that is used to display messages.	
Description	Creates an index file that is used to display messages. Select the following message types using the parameters. MAN: System manual file (file MAN.DOC) ERR: Error message file (file ERR.H)	
	The index file is needed by the RIDX command. (The RIDX command is required by the MAN, HELP and ERR commands.)	
Example	>MIDX MAN	



### 2.28 MORE MORE Format Display <Filter Command>

- Format MORE
- **Function** Controls output in page units.
- **Description** Does a MORE command display. First, 1 page worth of character string input by a pipe are displayed. Next, command input is waited for and pipe output (display) continues according to the command entered from the keyboard.
  - [n] <space> Displays the next n lines. When n is not specified, displays 1 page's worth.
  - [n] <CR> Displays the next n lines. When n is not specified, displays 1 line's worth.
  - [n] S Displays 1 page's worth after skipping the next n lines. When n is not specified, skips 1 page's worth.
  - Q Skips all lines and quits without displaying anything.

**Example** Display the trace buffer dump as a MORE display: >TDMP | MORE

### 2.29 QUIT Quit Address Checker

Format QUIT

**Function** Quits the Address Checker.

**Description** Quits the Address Checker and returns to MS-DOS.

Example

>QUIT A>



### 2.30 REM Comment Line

Format RI	EM [ <character< th=""><th>string&gt;]</th></character<>	string>]
-----------	--	----------

**Function** A comment line.

**Description** The line is handled as a comment and does nothing. This is convenient to use in the command file for comments.

**Note:** Do not use ";" in the character string.

#### Example

>REM EXECUTE GO >GO >REM THIS IS COMMENT!

# 2.31 RIDX Read Index File for Message Display

Format	RIDX { MAN   ERR } MAN: System manual file ERR: Error message file	
Function	Reads the index files that are used to display messages.	
Description	Reads the index files that are used to display messages. Select the follow- ing message types using the parameters. MAN: System manual file (file MAN.DOC) ERR: Error message file (file ERR.H)	
	Index file reading is required by the MAN, HELP and ERR commands. This command is also required when the user is adding expansion commands.	
Example	>RIDX MAN	



### 2.32 SET Set Address Checker Operating Environment

Format	<pre>SET <environment type="">/<mode></mode></environment></pre>	
Function	Sets the operating environment of the Address Checker.	
Description	Sets the operating environment of the Address Checker.	
	The following can be specified for <environment type="">: BM Bus master TRG Break when trigger is detected DSP Display speed mode (for DOS/V) The following can be specified for <mode>: (1) BM</mode></environment>	
	ALL All (SH2M, SH2S, SCU) SH2M SH2 (master) SH2S SH2 (slave) SCU SCU (2) TRG	
	BRK Break CNT Don't break	
	(3) DPS NRM SPD SPD Standard display mode Fast display mode (rewrites entire screen when com- mand ends)	
Example	The emulator operating environment information saved in the file can read and the same status operating environment set. See the example.	

# 2.33 ?SET Display Address Checker Operating Environment

Format	?SET		
Function	Displays the status of the Address Checker's operating environment.		
Description	Displays the status of the Address Checker's current operating environ- ment.		
Example	The emulator operating environment information saved in the file can be read and the same status operating environment set. See the example. Display current Address Checker operating environment:		
	>?SET SET BM/ALL TRG/CNT		
	Save the Address Checker operating environment once to a file, then read		
	the saved operating environment and set control with the same condi-		
	tions:		
	>?SET> SET.SAV	(Save current operating environment to file)	
		(Execute various commands.)	
	> < SET.SAV	(Read the previously saved operating environment from file and set the operating environment)	



### 2.34 STOP Stop

Format STOP

**Function** Aborts the program.

**Description** Stops an executing program.

Example Stop program execution: >STOP BREAK STOP! PC =00001000 SR=00000000:----\*\*-

### 2.35 TDMP Display Trace Data

2.35 томр Display Trace Data		
Format	TDMP [ <first frame="" number="">[<last frame="" number="">]]</last></first>	
Function	Displays the trace information.	
Description	Displays the trace information fetched from trace memory. Frame numbers are given in bus cycle units. The last bus cycle becomes frame 1 and the oldest frame has the largest number. The command displays the contents of trace memory from the <first frame="" number=""> to the <last frame="" number="">. When <first frame="" number=""> and <last frame="" number=""> are both omitted, the entire contents of the trace memory are displayed.</last></first></last></first>	
When searching large amount of trace contents, it is connected to first output the trace contents to a file, then display them with the VIEW con mand or use the editor with the @ command. See the usage example. The display format is shown below.		
.                     f-no add	displayed when trigger detected +	



CPU status

 Displays SH operating status
 RD Data read
 WR Data write

 DMA ack signal

 DACK0 DACK0
 DACK1 DACK1

ExampleDo a MORE display of all trace contents:<br/>>TDMP | MORE<br/>Display the last 12 frames from the final trace:<br/>>TDMP 1 12

### 2.36 TRC Set Trace Conditions

Format	<pre>(1) TRC ALL <trace control=""> (2) TRC SMP <trace control=""> (3) TRC TRG <trace control=""></trace></trace></trace></pre>	
Function	Format of <trace control="">: FUL/ {CNT STP} END/ {CNT BRK} Sets the trace conditions.</trace>	
Description	Sets the specified conditions as trace conditions. When the trace get condi- tions specified by <trigger condition=""> are reached during program execu- tion, the trace get starts according to <trace control="">.</trace></trigger>	
	<ul> <li>(1) For expression (1) TRC ALL <trace control=""> </trace></li> <li>When ALL is specified as the first parameter, the get of the trace information starts simultaneously with program execution.</li> </ul>	
	<ul> <li>(2) For expression (2) TRC SMP <trace control=""> </trace></li> <li>When SMP is specified as the first parameter, trace information is gotten only for trigger detections.</li> </ul>	
	<ul> <li>(3) For expression (3) TRC TRG <trace control=""> </trace></li> <li>When TRG is specified as the first parameter, the get of the trace information starts simultaneously with program execution and continues until a trigger is detected.</li> </ul>	
	<ul> <li>(4) Explanation of <trace control=""></trace></li> <li>Trace gets are controlled according to <trace control=""> as described below.</trace></li> <li>FUL/ {CNT   STP}</li> <li>Specifies processing when the trace buffer becomes full.</li> <li>CNT (default value): Continues the trace get. The trace buffer is used cyclically and the oldest trace information is erased.</li> <li>STP: Stops the trace get.</li> <li>END/ {CNT   BRK}</li> <li>Specifies processing when the trace get is stopped.</li> </ul>	
	CNT: Continues without breaking the execution of the user program. BRK (default value): Breaks execution of the user program.	
	Trace conditions can be temporarily saved to a file and then read later from the file to reset the same conditions. See the example.	
	<b>Note:</b> When this command is used, the current measured values of the 2 (hardware) bus counters are cleared to 0 (even when an error occurs during setting).	



**Example** Always get the trace. Break the program when the trace buffer becomes full.

>TRC ALL FUL/STP END/BRK

Always get the trace. Cycle the trace information gets when the buffer becomes full and continue the program. >TRC ALL FUL/CNT

Temporarily save trace conditions to a file, then read the conditions from the file and set them again:

(Saves the current trace conditions to a file.) (Execute various emulation commands.) (Reads the trace conditions from the file and sets them again.)

# 2.37 ?TRC Display Trace Conditions

Format	?TRC	
Function	Displays the trace condition	ons.
Description	Displays the current trace conditions. Trace conditions can be saved to a file and reset later.	
Example	Temporarily save trace conditions to a file, then read the conditions from the file and set them again:	
	>?TRC >TRACEX.SAV	(Saves the current trace conditions to a file.)
		(Execute various emulation commands.)
	> <tracex.sav< th=""><th>(Reads the trace conditions from the file and</th></tracex.sav<>	(Reads the trace conditions from the file and
		sets them again.)



### 2.38 \_TRC Delete Trace Conditions

**Function** Deletes the trace conditions.

**Description** Renders the currently set trace conditions invalid.

Example Delete the currently set trace conditions: >\_TRC

# 2.39 TSET Set Trigger Detection Condition

Format	TSET
Function	Sets trigger detection condition.
Description	Sets trigger detection condition according to the TRGMAP.DAT file contents.
	This is used to reset the trigger detection condition after updating the TRGMAP.DAT file.
Example	Update the TRGMAP.DAT file and reset the trigger detection condition.



### 2.40 VRAM Store Display Contents to a File

Format VRAM [<filename>]

**Function** Saves the contents of the current display to a file.

- **Description** Reads the contents of the currently displayed VRAM and saves them to a file. When PRN is specified as the file name, a hard copy goes to the printer. (Starting from MS-DOS V3.3, no hard copy can be made unless DEVICE=PRINT.SYS in CONFIG.SYS.) When the file name is omitted, the file is named VRAM.TAG.
- **Example** Saves the contents of the current display to a file called HCOPY: >VRAM HCOPY

# 3. Detailed Contents of TRGMAP.DAT

Among the System Program files, TRGMAP.DAT is the file that defines the address maps which dictate the Sega Saturn System trigger condition. The file contents are as follows:

Ad	dress	s Che				Defini	tion									
		Condition					I	7:			Free					
		Condition						<b>A</b> :					prohi	bitec	1	
		Condition						र:				d O				
				Condi					N:				te O			
			C	Condi	tion			Ι	.:			See	List			
<b>T</b> 1		1				• 417	1 .	г	1.	1 1						
							-byte	es. Fo	r de	tailed	setti	ngs,				
spe	ciry l	L, the	en re	fer to	tne	List.										
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	,,,,,,,,,,	,,,,,,,,,,,	,,,,,,,,,	,,,,,,,,,,,,	,,,,,,,	,,,,,,,,,,,,	,,,,,,,,	,,,,,,,,,,,,	,,,,,,,,	,,,,,,,,,,,,	,,,,,,,,,	,,,,,,,,,,,	,,,,,,,,,	,,,,,,,,,,,,	,,,,,,,,,	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
Address			(	Condi	tion			E	Bus V	Width						
00000000			F						6							
00080000			Ā	A												
00100000			L													
Set	unus	sed b	its to	o 1. (V	Vher	n all b	its a	re un	usec	l, wri	te is	prohi	bite	d.)		
R d	enot	es Re	ad C	Only;	Wd	enote	s Wr	ite O	nly							
	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
SMPC Re	giste	r														
000	FF	00W		<u>00</u> W		<u>00</u> W		00W		00W		<u>00</u> W		<u>00</u> W		FF
010 020	FF FF	FF 00R	FF	FF 00R	FF	FF 00R	FF FF	FF 00R	FF	FF 00R	FF FF	FF 00R	FF	FF 00R	FF	00W 00R
030	FF	00R	FF	00R	FF	00R	FF	00R	FF	00R	FF	00R	FF	00R	FF	00R
040 050	FF FF	00R 00R		00R 00R	FF FF	00R 00R		00R 00R		00R 00R	FF	00R 00R		00R 00R	FF FF	00R 00R
060	FF	00R		FE	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
070 ID	FF	FF	FF	FF	FF	80	FF	80	FF	80W	FF	80W	FF	FEW	FF	FEW
17																
2	A															
100080																
	A F															



**Description** ";" denotes comments. "address" denotes the starting address.

> "Condition" denotes the trigger detection condition; "F" denotes Free Access; "A" denotes Access prohibited; "R" denotes Read Only; "W" denotes Write Only; "L" denotes See List. In the List, conditions can be specified in single bytes (this is normally in 4 bytes.)

"Bus Width" denotes the effective bus width. When abbreviated, the previous setting will be enabled. The List allows the selection of unused bits, Read Only, and Write Only. All unused bit addresses prohibit Write. The List starts with the L row or the L2 row and ends with the END row.

When the List starts with the L2 row, for addresses where unused bits and used bits coexist, the unused bit "1" Write is prohibited. There are Lists for 8 bits, 16 bits, and 32 bits.

# 4. Error Messages

Error messages are displayed in Japanese, but error messages that require annotation and how to deal with them are described below.

• The contents of the configuration file are illegal.

See section 2.6, "Setting the Configuration," and correct the contents of the file.

• Hardware is not connected.

Check to see if the power switch to the Address Checker is on. Also check to see if the interface board is connected to the host personal computer. Finally, check that BANK is correct.



)00H   ROI	Access Area	512Kbyte	ROM	<b>1</b> 	CS0
оон	cess Prohibited			A	
) ООН	PC Area	128Kbyte	Register		→ [SMPC Register]
——— НО8С Асо	cess Prohibited	Access	Prohibited	B	
000H	ckup RAM Area	32Kbyte	S-RAM		
	cess Prohibited	Access	Prohibited	C	
000H   Wo	rk RAM-L Area	1Mbyte	D-RAM		
НОО( Асс	cess Prohibited	Access	Prohibited	D	
)ООН	C Area (MINIT)	4byte	Register		
)02H   Acc	cess Prohibited	Access	Prohibited	E	
	C Area (SINIT)	4byte	Register		
•	cess Prohibited	Access	Prohibited	F↓	
	Bus ACSO	32Mbyte	Cartridge ROM	†	C S 1
)оон	Bus ACS1	   16Nhuta	Cartridge RAM	+     +	<b>C S 2</b>
оон					0.5.2
A-  	Bus Reserved		Access Prhbted		
	Bus ACS2	1Mbyte	External-1/0 & CD Block		→ 【CD Register】
	cess Prohibited	Access	Prohibited	 	
	UND Area	512kbyte	RAM		
,	cess Prohibited		Prohibited		

	SOUND Area	3812byte	Register	<u>}</u>	→ [SCSP Register]
5B00EE4H	Access Prohibited	• Access	Prohibited	   J	
5C00000н	VDP1 Area	768Kbyte	VRAM&Frm Buffr		
5 <b>сс</b> 0000н	Access Prohibited	• Access	Prohibited	K	
5D00000н	VDP1 Area	24byte	Register	{   	→ 【VDP1Register】
5D00018H ⊣ 	Access Prohibited	• Access	Prohibited	   L	
5E00000н	VDP2 Area	512Kbyte	VRAM		
5E80000H	Access Prohibited	Access	Prohibited	   M	
5F00000H	VDP2 Area	4Kbyte	Color RAM	 	
5F01000H	Access Prohibited	• Access	Prohibited	   N	
5F80000H	VDP2 Area	288byte	Register	 	→【VDP2 Register】
5F80120H	Access Prohibited	• Access	Prohibited	   0	
5FE0000н  - 	SCU Area	208byte	Register	{    }	→【SCU Register】
5FE00D0H	Access Prohibited	Access	Prohibited	   P↓	-
6000000н —	Work RAM-H Area		S-DRAM	   ↑ (	C S 3
6100000н  - 			Prohibited	       0	
8000000н <sup>′</sup>				1 †	



### **Addendum: Register Names**

#### SMPC (System Management & Peripheral Control)

Saturn system reset management; Control Pad and other peripheral devices interface management. Each has a backup battery and real-time clock.

#### SCSP (Saturn Custom Sound Processor)

Sound source LSI for game, which integrates PCM sound source and DSP for sound.

#### VDPI (Video Display Processor 1)

LSI for drawing sprites and polygons. The VRAM and frame buffer are connected and data is thus transferred. Also, data is sent to VDP2 when requested.

**VDP2 (Video Display Processor 2)** Manages scroll planes and includes priority functions.

#### SCU (System Control Unit)

Includes CPU I/F, A-Bus I/F, and B-Bus I/F Controllers. Includes DMA controller, interrupt controller, and DSP.

SATURN Access Area	Data Access Width	SH2 Data Bus Width
ROM Access Area	×, w, ×	
SMPC Area Register	Х, W, В	
Backup RAM Area		ı  -   CSO 16Bit Bus Width
Work RAM-L Area		
DCC Area Register	imes, w, $ imes$	
A-Bus ACSO	L,W,B	CS1 32Bit Bus Width
A-Bus ACS1		
A-Bus ACS3		
25818000H		   
A-Bus ACS2 Register	imes, w, $ imes$	1
SOUND Area Work RAM	L, W, B	1
SOUND Area Register		   . 
VDP1 Area VRAM	~	  CS2 32 Bit Bus Width 
VDP1 Area Register	×, w, ×	1:   
VDP2 Area VRAM	L, W, B	1
VDP2 Area Color RAM	L,W, $\times$	1 
VDP2 Area Register		 
SCU Area Register	L, W, B	1
Work RAM-H Area		CS3 32Bit Bus Width



◆Data Access Width (Data width that can be accessed within a program)

- L [32bit] Long Word Access
- W [16bit] Word Access
- B [ 8bit] Byte Access
- $\times$  ······Access Prohibited Width

Access size setting of [SATURN C Bus] according to the SH Bus State Controller CSO Area0 0000000'H~1FFFFFF'H 1 6 Bit Data Bus Width

- CS1 Area1 2000000'H~3FFFFF'H 3 2 Bit Data Bus Width
- CS2 Area2 4000000'H~5FFFFF'H 32 Bit Data Bus Width
- CS3 Area3 6000000'H~7FFFFF'H 32 Bit Data Bus Width

◆A-Bus ACS3 Area is usually Access Prohibited

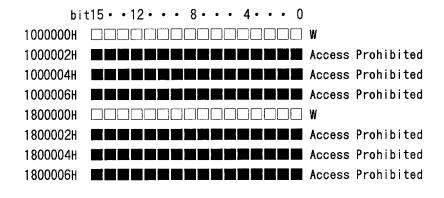
ж (smpc	Register Map】
bit	7•• 4••• 0
0100000н	
0100001H	
0100002H	
0100003н	
0100004H	
0100005H	W DODDER
0100006н	
0100007н	
0100008н	
0100009н	<b>W</b>
010000AH	
010000BH	
010000 <b>C</b> H	
010000DH	
010000EH	
010000FH	
0100010H	
0100011H	
0100012H	
0100013H	
0100014H	
0100015H	
0100016н	
0100017H	
0100018H	
0100019Н	
010001AH	
010001BH	
010001CH	
010001DH	
010001EH	
010001FH	
0100020H	
0100021H 0100022H	
0100022H 0100023H	
0100023H 0100024H	
0100024H 0100025H	
0100025H 0100026H	
0100020H	
0100027H 0100028H	
01000200	

bit	7 · · 4 · · · 0	
0100 <b>040</b> H		
0100041н		R
0100042н		
0100043H		R
0100044H		
0100045H		R
0100046H		_
0100047H		R
01000488		
0100049H		R
010004AH		D
010004BH 010004CH		R
0100040H 010004DH		R
010004DH		n
010004En		R
0100050H		n
0100051H		R
0100052H		
0100053H		R
0100054H		
0100055н		R
0100056н		
0100057н		R
0100058н		
0100059н		R
010005AH		
010005BH		R
010005 <b>ch</b>		
010005DH		R
010005EH		
010005FH		R
0100060H		
0100061H		R
0100062H		
0100063H		R/W
0100064H		
0100065H		
0100066H		
0100067H		
0100068H		



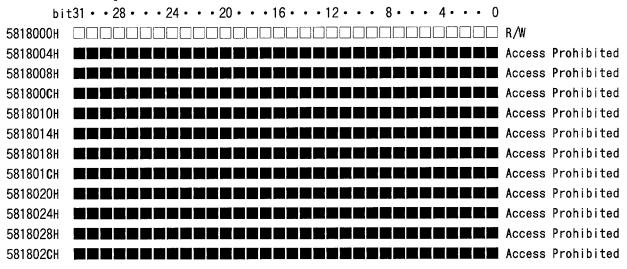
0100029н	R	0100069н		
010002AH		010006AH		
010002BH	R	010006BH		
010002 <b>c</b> h		010006 <b>c</b> h		
010002DH		010006DH		
010002EH		010006EH		
010002FH	R	010006FH		
0100030H		0100070н		
0100031Н		0100071н		
0100032H		0100072H		
0100033H		0100073н		
0100034Н		0100074H		
0100035H		0100075H	R/W	
0100036H		0100076H		
0100037H		0100077H	R∕₩	
0100038H		0100078H		
0100039н		0100079н	W	
010003AH		010007AH		
010003BH		010007BH	W	
010003 <b>c</b> h		010007 <b>c</b> h		
010003DH		010007DH	W	
010003EH		010007EH		
010003FH		010007FH	W	

#### ※ 【DCC Register Map】



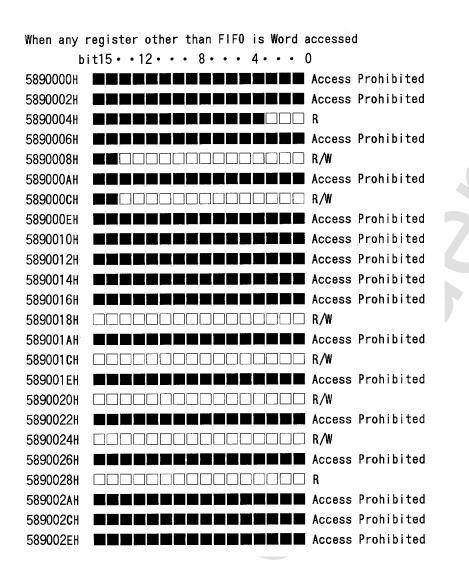
※ 【CDROM Register Map】

When FIFO is Long Word accessed





When FIFO	is Word accessed t15・・12・・・ 8・・・ 4・・・ 0	
5898000H		R/W
5898002H		Access Prohibited
5898004H		Access Prohibited
5898006H	ک کا کا کا کا بین کا کا کا پر ان کا کا کا ک	Access Prohibited
5898008H		Access Prohibited
589800AH	و و و و و و و و و و و و و و و	Access Prohibited
589800 <b>c</b> h	و و و م و و و و و و و و و و و	Access Prohibited
589800EH		Access Prohibited
5898010H		Access Prohibited
5898012H		Access Prohibited
5898014H		Access Prohibited
5898016H		Access Prohibited
5898018H		Access Prohibited
589801AH	ک کر کے کا کر کا کا کا کا کا کا کا کا ک	Access Prohibited
589801 <b>c</b> h		Access Prohibited
589801EH		Access Prohibited
5898020H		Access Prohibited
5898022H		Access Prohibited
5898024H		Access Prohibited
5898026H		Access Prohibited
5898028H	ہ ہے کا ک	Access Prohibited
589802AH		Access Prohibited
589802 <b>c</b> h	ت و م و تو بر تر و و و و و و و و	Access Prohibited
589802EH		Access Prohibited

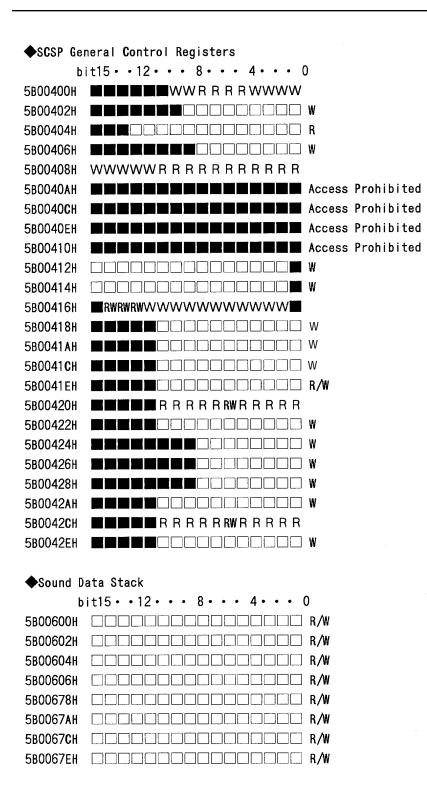


SEGA









### ※ 【SCSP Register Map】 ◆DSP Control Register

COEF bi	$t15 \cdot \cdot 12 \cdot \cdot \cdot 8 \cdot \cdot \cdot 4 \cdot \cdot \cdot 0$	
5B00700H		R∕₩
5B00702H		R/W
5B00704H		R/W
5B00706H		R/W
5B00778H		R/W
5B0077AH		R/W
5B0077 <b>c</b> h		R/W
5B0077EH		R/W

### MADRS

5B00780H	R/W
5B00782H	R/W
5B00784H	
5B00786H	
5B007B8H	
5B007BAH	
5 <b>8007BC</b> H	
5B007BEH	

#### MPR0

5B00800H	R/W
5B00802H	R/W
5800804H	R/W
5B00806H	R∕W
5BOOBF8H	R/W
5BOOBFAH	R/W
5B00BFCH	R/W
5B00BFEH	R/W

#### TEMP

5B00C00H	R/W
5B00C02H	R∕W
5B00C04H	R/W
5B00 <b>C0</b> 6H	R∕W
5B000F8H	R/W
5BOODFAH	R/W
5B00DF <b>c</b> h	R∕W
5BOODFEH	R∕W



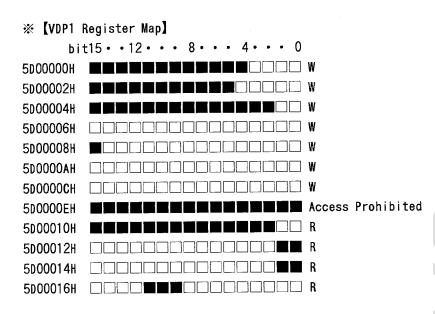
#### MEMS

5B00E00H	R∕₩
5B00E02H	R/W
5B00E04H	R/W
5B00E06H	R∕W
5B00E78H	R/W
5 <b>B00E7A</b> H	R/W
5 <b>B00e7c</b> h	R/W
5B00E7EH	R/W

### MIXS

5B00E80H	R/W
5B00E82H	R/W
5B00E84H	R/W
5B00E86H	R∕₩
5B00EB8H	R/W
5BOOEBAH	R/W
5800EBCH	R/W
5B00EBEH	R/W

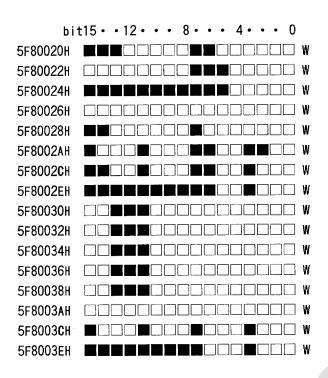
EFREG b	it15••12••• 8••• 4••• 0
5b00ec0h	R/W
5B00E <b>c</b> 2h	R/W
5B00E <b>C4</b> H	
5b00e <b>c</b> 6h	
5BOOED8H	R/W
5BOOEDAH	R/W
5BOOEDCH	R/W
5B00EDEH	R/W

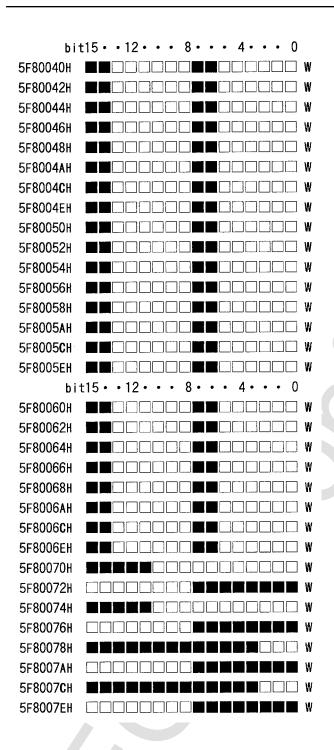


#### % [VDP2 Register Map]

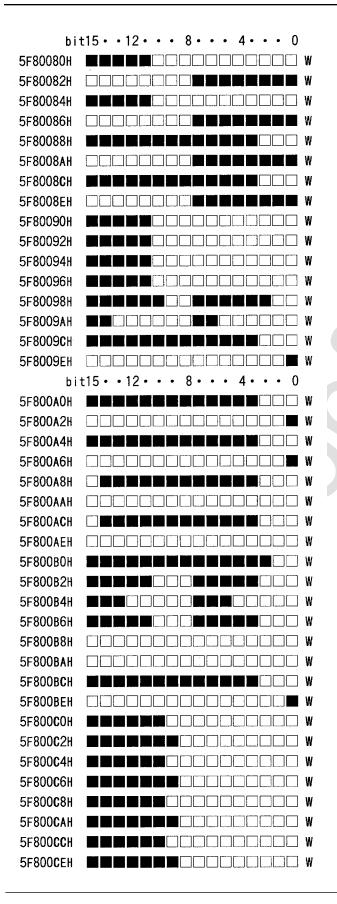
bi	$t15 \cdot \cdot 12 \cdot \cdot \cdot 8 \cdot \cdot \cdot 4 \cdot \cdot \cdot 0$	
5F80000H		R/W
5F80002H		R∕W
5F80004H		R
5F80006H		
5F80008H		R
5F8000AH		R
5F8000 <b>c</b> h		W
5F8000EH		R/W
5F <b>800</b> 10H		W
5F80012H		W
5F80014H		W
5F80016H		W
5F80018H		W
5F8001AH		W
5F8001 <b>C</b> H		W
5F8001EH		W

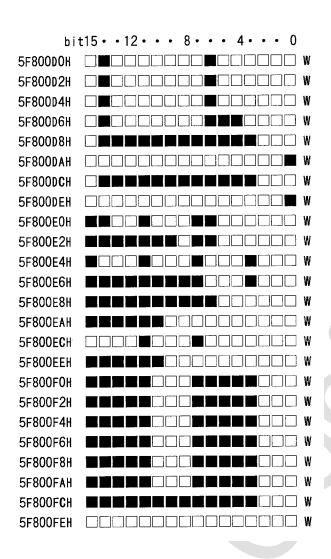




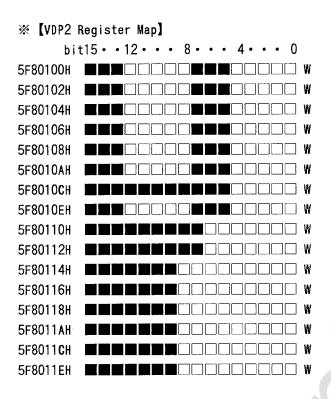






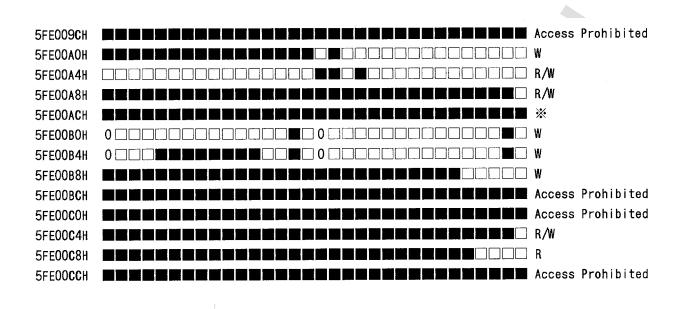












\* ····READ Prohibited (If this register is read, the application will become unstable.)

0 ····Always enter 0 when writing.

- ••••Access Prohibited Register.
   Only O is allowed when writing.
   When the row is all I, then access is prohibited.
- Both O and 1 are allowed.