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, 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 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 simply 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 with this document. Please address comments to :

SEGA of America, Inc., Developer Technical Support (att. Evelyn Merritt)
150 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.

SEGA OF AMERICA, INC.
Consumer Products Division

SCU DSP
Simulator
User's Manual

Doc. # ST-240-B-042795

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 ST-240-B-042795 **Date** _____

Document name SCU DSP Simulator User's Manual

Corrections:

Chpt.	pg. #	Correction

Questions/comments: _____

Where to send your corrections:

Fax: (415) 802-1717
Attn: Evelyn Merritt,
Developer Technical Support

Mail: SEGA OF AMERICA
Attn: Evelyn Merritt,
Developer Technical Support
150 Shoreline Dr.
Redwood City, CA 94065

1. Overview

The SCU DSP Simulator was designed to simulate the execution of DSP code under the MS-DOS or UNIX operating systems to aid DSP software development.

All simulator functions are executed via a command-line interface. Supported functions are listed below.

(1) Dump display of program and data areas and editing functions.
(2) S format compatible file load and save functions.
(3) Breakpoint setting and display functions.
(4) Program execution and pause functions.
(5) Single-step program execution function.
(6) Assembling and disassembling functions.
(7) Display and set functions for all registers.
(8) History log function for input commands .

2. Operating Environment

<A> MS-DOS Version
The program must be run on a unit that is compatible with MS-DOS Version 3.0 or higher, and will run with 640 KB of physical memory.

<B> UNIX Version
Compatible with SPARC-based and HP workstations.

3. List of Commands

[A] Assemble
[B] Set and display breakpoint
[D] Display memory dump
[E] Edit memory
[F] Fill memory
[G] Run program
[H] Set and display history
[L] Read file
[M] Move data within memory
[Q] Quit program
[R] Set and display register
[S] Step execution of program
[U] Disassemble
[V] Select development target
[W] Save contents of memory to file

4. Command Descriptions

A: Assemble

Input format: **A[start address]**
Function(s): This command converts mnemonic input to code and sets it in memory. Press **Return** to quit.

B: Set and display breakpoint

Input format: **B<address>**
Function(s): Sets breakpoint.

Input format: **B**
Function(s): Displays breakpoint setting information.

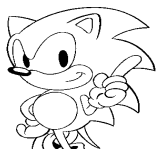
Input format: **B-[address]** or **BX[number]**
Function(s): Clears breakpoint.
If a specific address or number is not specified, all breakpoints will be cleared.

D: Display memory dump

Input format: **D[P|Rn|M][[start address[end address]]**
Function(s): Displays memory dump.
Set "P" to specify the program area, "R0"~"R3" for the data area, and "M" for the external bus. If the start address is not specified, the program will display the address which comes after the address last displayed.

E: Edit memory

Input format: **E[P|Rn|M]<address>[value]**
Function(s): Edits memory.
Set "P" to specify program area, "R0"~"R3" for data area, and "M" for the external bus. If a value is not specified, the program displays an address and waits for data input. To step forward through memory addresses, enter space; to back up, enter "^" and to cancel, enter "." only.



F: Fill memory

Input format: **F[P|Rn|M]<start address><end address><value>**
Function(s): Rewrites the area between the specified starting and ending addresses with the setting value. Set “P” to specify the program area, “R0”~”R3" for the data area, and “M” for the external bus.

G: Run program

Input format: **G[start address[end address]]**
Function(s): Runs the program from a specified address. If an address is not specified, the program is run from an address specified by the PC register. To cancel during execution, enter CTRL-C.

H: Set and display history

Input format: **H+**
Function(s): Allows data fetches from the history buffer (default).

Input format: **H-**
Function(s): Prohibits data fetches from the history buffer.

Input format: **H@**
Function(s): Clears the history buffer.

Input format: **H[number of steps]**
Function(s): Displays a specified number of execution results that precede this command. If no value is specified, the program sets 10 by default. The maximum is 512 steps.

L: Read file

Input format: **L[P|Rn|M][filename][transfer destination address]**
Function(s): Reads a file into memory.
Set “P” to specify the program area, “R0”~”R3" for the data area, and “M” for the external bus. If no filename is specified, the last file read using the L command is selected. The specified transfer address is used as an offset.

M: Move data within memory

Input format: **M[P|Rn|M]<start><end>[P|Rn|M]<transfer destination address>**

Function(s): Transfers data between the specified start and end address to the specified transfer destination address. Set "P" to specify the program area, "R0"~"R3" for the data area, and "M" for the external bus. If a program area in the transfer destination address is not specified, the same area specified in the original source address is used.

P: Change program size

Input format: **P[R|E]**

Function(s): Adjusts size of program memory.
If a parameter is not specified, the program displays the current setting. If "E" is specified, the program memory expands to store 2048 instructions, and when "R" is specified, it is returned back to 256 instructions.

Q: Quit program

Input format: **Q**

Function(s): Ends the simulator program

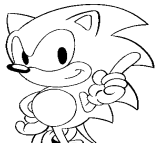
R: Set and display register

Input format: **R[<register name|flag name><value>]**

Function(s): Sets a value to the selected register or flag. If a parameter is not specified, values for all registers are displayed.
Values may be set to the following registers: **PC TOP LOP CT0 CT1 CT2 RX RY PH PL ACH ACL TN0 RA0 WA0 A1.**
Values may be set to the following flags: **PR EP T1 T0 S Z C V E ES EX**.

Input format: **R@**

Function(s): Sets all the registers to 0.



S: Step execution of program

Input format: **S[number of steps]**
Function(s): Runs the specified number of instruction steps from the address given by PC register. If a parameter is not specified, only one step is executed.

U: Disassemble

Input format: **U[start address[end address]]**
Function(s): Displays disassembled data for the selected area. If a start address is not specified, the address after the last displayed address is entered.

V: Select development targets

Input format: **V[S|M]**
Function(s): Selects the development target; specify "M" for MODEL-M or "S" for MODEL-S. Specifying this value enables assembly, disassembly, and execution functions on these targets. Default is set to "S". When a target is not specified, the current setting is displayed.

W: Save data to memory

Input format: **W[P|Rn|M]<start address><end address><filename>**
Function(s): Saves memory data between the specified start and end addresses to a file. Set "P" to specify the program area, "R0"~"R3" for the data area, and "M" for the external bus.

^: Command history display

Input format: **^ [number of lines]**
Function(s): Displays a specified number of input lines. If the number of lines are not specified, the program displays 20 lines. The maximum number of lines that can be displayed is 50. To repeat the commands, enter "!!", to repeat a command on a specific line, enter "!" followed by the line number.

5. Basic Specifications

(1) All numerical input, except for the number of times a command is repeated (entered in decimal), are represented in hexadecimal.

(2) Filenames that end with ".s" or ".mot," when read and saved by the L and W commands, are assumed to be in Motorola S format and the address information is processed simultaneously.

(3) To input a series of commands at once, specify the file which contains the commands as part of the start up parameter, or load the commands from a file by specifying "<file" from the command line.

(4) Addresses are in byte units when accessing external bus memory. When specifying an address as an internal address, appending "L" at the end of the address value in the command will increase the value by 4.

(5) To continue a display from a D, S, or U command, press **Return** only.

6. Tips for Use

(1) When transferring data from external to internal memory during a DMA simulation, be sure to input or transfer data to the external bus memory of the DSP simulator in advance.

(2) The disassemble function of the simulator orders commands according to the operation instruction type. Therefore, this feature can be used as a guide for optimization.