

Chapter 1

DSP SIMULATOR

1.1 INTRODUCTION

The DSP simulator program is a software tool for developing programs and algorithms for Motorola digital signal processors (DSPs). This program exactly duplicates the functions of supported Motorola DSP chips, including all on-chip peripheral operations, all memory and register updates associated with program code execution, and all exception processing activity. The device's pipelined bus activity is exactly simulated. This enables the simulator to provide the user an accurate measurement of code execution time which is so critical in DSP applications.

The simulator executes object code which can be generated using either the device Macro Assembler program or the simulator's internal single-line assembler. The object code is loaded into the simulated device's memory map. The entire internal and external memory space of the DSP is simulated. During program debug the user can display and change any of the device's registers or memory locations. Instruction execution can proceed until a user defined breakpoint is encountered; or in a single-step mode, stopping after a specified number of instructions or cycles have executed.

1.2 FEATURES

Summary of simulator features:

- Multiple device simulation
- Source level symbolic debug of assembly and C source programs
- Conditional or unconditional breakpoints
- Program patching using a Single-Line Assembler/Disassembler
- Instruction and Cycle timing counters
- Session and/or Command Logging for later reference
- Input/Output ASCII files for device peripherals
- Help file and Help line display of Simulator commands
- Macro command definition and execution
- Display Enable/Disable of Registers and Memory
- Hexadecimal/Decimal/Binary calculator

1.3 OPERATING ENVIRONMENT

The minimum hardware requirements for the DSP simulator include:

- **IBM AT**^{*} (386 or 486 only) with **2Mb** of **RAM**
- **PC-DOS / MS-DOS**^{**} v3.0 or later.
- **IBM,AT** and **PC-DOS** are trademarks of International Business Machines.
- **MS-DOS** is a trademark of Microsoft Corp.

The simulator supports all of the external memory maps of the DSP. It is compiled with a compiler which supports extended and virtual memory on the PC. The file readme.mem will contain additional information for configuration of the PC to support the memory management.

Floppy diskette drives are adequate for small simulations. However, due to the virtual memory paging scheme and since many of the INPUT and OUTPUT commands reference disk files, a fixed disk drive is highly recommended.

If your simulation involves many assigned disk files, the operating system's limit of the number of open files may be reached. This will cause the simulation to slow down while files are closed and then reopened for accesses. In order to reduce the chance of this situation occurring, it is recommended that your operating system's CONFIG.SYS file be modified with the following MS-DOS configuration commands:

```
BUFFERS=32  
FILES=20
```

These commands increase the number of disk memory buffers and the maximum number of files that can be open at one time.

1.4 RUNNING THE SIMULATOR

The format for invoking the simulator is:

SIMDSP [macro command filename]

Although the name **simdsp** is used throughout this manual for example purposes, the actual name of the simulator is device dependent. For example, the DSP56000 and DSP56001 devices use the simulator named **sim56000**, while the DSP56116 device uses **sim56100**. See Chapter 8, Simulator Names for the actual name used for your device simulator.

The macro command filename is an optional parameter. The macro command file should contain a sequence of commands that the user wishes to execute upon simulator start-up and prior to command entry from the keyboard. If an incorrect command is encountered in the macro command file, the macro command will terminate and command entry will be enabled from the keyboard. Macro command files can be nested (a macro command file can call another macro command file) to any level.

If you do not specify a suffix in the macro command file name, the simulator will assume the suffix ".cmd".

EXAMPLES

SIMDSP

Invoke the simulator. Begin keyboard command input immediately (no macro file).

SIMDSP STARTUP

Invoke the simulator and run the macro file named "STARTUP.CMD".

SIMDSP SETUP.N5

Invoke the simulator and run the macro file named "SETUP.N5".

SIMDSP SETUP5.

Invoke the simulator and run the macro file named "SETUP5.".

1.5 USER INTERFACE

The bottom three screen lines function as the command line, an error message line, and a help line.

As each valid command is accepted from the command line, it and its results are scrolled into the display screen. The last 100 lines of display screen entry are available for review at any time by typing **Pg Up** (Ctrl-T), **Pg Dn** (Ctrl-V), **Up-Arrow** (Ctrl-U) or **Down-Arrow** (Ctrl-N).

1.6 COMMAND ENTRY

Upon entry into the simulator, several of the available commands are displayed on the help line. The remaining commands can be reviewed by pressing the **SPACE** bar when the cursor is at the start of the command line.

The simulator requires a minimum number of key strokes to recognize a simulator command. The minimum number of required characters for each command is shown highlighted on the help line. A command can be specified by typing the required characters followed by a space or by typing the entire command word followed by a space.

Entering the command key strokes followed by a space will activate the help line for that particular command. The help line shows the syntax for the remainder of the command. Additional help and examples of the current instructions can be obtained by typing a question mark at any point during the command entry.

Any text following a semicolon on the command line is considered to be a user comment. This provides the user a means of documenting session display.

Command execution begins when the user types the **ENTER** or **CARRIAGE RETURN** key. If the entered command is not one of the predefined simulator commands, the simulator interprets the command as a macro file name and executes the macro file. Macro command files can be created by logging command entries. This procedure is explained in the documentation of the simulator **LOG** command.

Command line editing is supported for command entry corrections. The cursor can be moved on the command line by using the **Left-Arrow** (Ctrl-L) and **Right-Arrow**

1.7 DISPLAY MODES

The simulator supports three display modes - register, assembly and source. These modes determine the simulator display at the termination of commands which initiate device cycle execution. The register display mode causes the display of register and memory locations enabled by the **DISPLAY** command. The assembly display mode causes the display of one full screen of disassembled instructions containing the instruction at the current execution address. The source display mode causes the display of one page of the original source file which contains the source line associated with the current execution address. In both the assembly and source display modes the position of the current execution address is marked by => in the left margin.

The source display mode requires symbol and line information in the object file that will normally be the result of assembling with the -g option of the assembler. See the assembler manual for instructions on the use of the -g option.

A display mode can be selected either by the simulator **VIEW** command, or by toggling among the display modes using the **Ctrl-W** key entry (hold down Ctrl and press w). In addition, simulator commands which display registers or memory, or otherwise create display to the register display window will select the register display mode; and the simulator **LOAD** and **LIST** commands will switch from the register display mode to the source display mode.