IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgment, including those pertaining to warranty, patent infringement, and limitation of liability.

TI warrants performance of its products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Customers are responsible for their applications using TI components.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance or customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, license, warranty or endorsement thereof.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations and notices. Representation or reproduction of this information with alteration voids all warranties provided for an associated TI product or service, is an unfair and deceptive business practice, and TI is not responsible nor liable for any such use.

Resale of TI's products or services with *statements different from or beyond the parameters* stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service, is an unfair and deceptive business practice, and TI is not responsible nor liable for any such use.

Also see: Standard Terms and Conditions of Sale for Semiconductor Products. www.ti.com/sc/docs/stdterms.htm

Mailing Address:

Texas Instruments
Post Office Box 655303
Dallas, Texas 75265

Preface

Read This First

About This Manual

This document is the user's guide for the TMS320C55x™ instruction set simulator, available within Code Composer Studio. This document describes the basic capabilities of the simulator and the features provided for configuring the it.

How to Use This Manual

This document contains the following chapters:

Chapter 1 discusses two simulation drivers that are available to simulate C55x DSP CPU and subsystems. This chapter also describes the capabilities and limitations of each driver.

Chapter 2 contains information on how to change stack configuration and also the differences related to C54x-compatible mode.

Chapter 3 discusses the importing and custom memory configuration setups.

Notational Conventions

This document uses the following conventions.

- Program listings, program examples, and interactive displays are shown in a `special typeface` similar to a typewriter's. Examples use a **`bold version`** of the special typeface for emphasis; interactive displays use a **`bold version`** of the special typeface to distinguish commands that you enter from items that the system displays (such as prompts, command output, error messages, etc.).

  Here is a sample program listing:

```
0011  0005  0001         .field    1, 2
0012  0005  0003         .field    3, 4
0013  0005  0006         .field    6, 3
0014  0006               .even
```

Here is an example of a system prompt and a command that you might enter:

```
C:  csr –a /user/ti/simuboard/utilities
```

- In syntax descriptions, the instruction, command, or directive is in a **bold typeface** font and parameters are in an *italic typeface*. Portions of a syntax that are in **bold** should be entered as shown; portions of a syntax that are in *italics* describe the type of information that should be entered. Here is an example of a directive syntax:

  .sect "*section name***"**

  .sect is the directive. This directive has one parameter, indicated by *section name*. When you use .sect, the first parameter must be a section name, enclosed in double quotes.

- Square brackets (**[** and **]**) identify an optional parameter. If you use an optional parameter, you specify the information within the brackets; you don't enter the brackets themselves. Here's an example of an instruction that has an optional parameter:

  ADD *[src,] dst*

  The ADD instruction has two parameters. The first parameter, *src*, is optional. The second parameter, *dst*, is required. As this syntax shows, if you use the optional first parameter, you must add a comma before the second parameter.

- Braces ({ and }) indicate a list. The symbol **|** (read as *or*) separates items within the list. Here's an example of a list:

  ```
  { * | *+ | *– }
  ```

  This provides three choices: *, *+, or *–.

  Unless the list is enclosed in square brackets, you must choose one item from the list.

- Some directives can have a varying number of parameters. For example, the .byte directive can have up to 100 parameters. The syntax for this directive is:

  .byte $value_1$ *[, ... ,* $value_n$*]*

  This syntax shows that .byte must have at least one value parameter, but you have the option of supplying additional value parameters, separated by commas.

Related Documentation From Texas Instruments

The following books describe the TMS320C55x devices and related support tools.

TMS320C55x Optimizing C/C++ Compiler User's Guide (literature number SPRU281) describes the C55x C/C++ compiler. This compiler accepts ANSI standard C/C++ source code and produces assembly language source code for the TMS320C55x device.

TMS320C55x Assembly Language Tools User's Guide (literature number SPRU280) describes the assembly language tools (assembler, linker, and other tools used to develop assembly language code), assembler directives, macros, common object file format, and symbolic debugging directives for TMS320C55x™ devices.

TMS320C55x DSP CPU Reference Guide (literature number SPRU371) describes the architecture, registers, and operation of the CPU for the TMS320C55x™ digital signal processors (DSPs). This book also describes how to make individual portions of the DSP inactive to save power.

TMS320C55x DSP Mnemonic Instruction Set Reference Guide (literature number SPRU374) describes the TMS320C55x™ DSP mnemonic instructions individually. Also includes a summary of the instruction set, a list of the instruction opcodes, and a cross-reference to the algebraic instruction set.

TMS320C55x DSP Algebraic Instruction Set Reference Guide (literature number SPRU375) describes the TMS320C55x™ DSP algebraic instructions individually. Also includes a summary of the instruction set, a list of the instruction opcodes, and a cross-reference to the mnemonic instruction set.

Code Composer User's Guide (literature number SPRU328) explains how to use the Code Composer development environment to build and debug embedded real-time DSP applications.

Trademarks

TMS320C55x and Code Composer Studio are trademarks of Texas Instruments.

If You Need Assistance. . .

If you want to. . .	Do this. . .
Request more information about Texas Instruments Digital Signal Processing (DSP) products	Call the CRC† hotline: **(800) 336–5236** Or write to: Texas Instruments Incorporated Market Communications Manager, MS 736 P.O. Box 1443 Houston, Texas 77251–1443
Order Texas Instruments documentation	Call the CRC† hotline: **(800) 336–5236**
Ask questions about product operation or report suspected problems	Call the DSP hotline: **(713) 274–2320**
Report mistakes in this document or any other TI documentation	Fill out and return the reader response card at the end of this book, or send your comments to: Texas Instruments Incorporated Technical Publications Manager, MS 702 P.O. Box 1443 Houston, Texas 77251–1443

† Texas Instruments Customer Response Center

Contents

Figures

Chapter 1

Simulator Features and Limitations

This chapter discusses two simulation drivers that are available to simulate the C55x CPU and its subsystems. This chapter also describes the capabilities and limitations of each driver.

Topic **Page**

1.1 C55x CPU and Memory System Simulation Capabilities

1.1.1 Functional Capabilities

The simulator's functional capabilities are listed below.

- C55x CPU full instruction set architecture execution (except emulation instructions and IDLE instruction). For more information, see the *TMS320C55x DSP CPU Reference Guide* and the *TMS320C55x DSP Instruction Set Reference Guides*.
- Parallel instruction execution
- Configurable memory system simulation
  - If the memory configuration is not provided, a flat memory system (memory with no latency, no DARAM/SARAM) is used as a default.
  - Program/data memory with latency is supported.
  - If the memory map is provided in a configuration file, the driver uses the cycle accurate memory system (SARAM/DARAM). Support of SARAM and DARAM memory models follow the C55x memory protocol and access priorities. To use the memory system, you must set up the configuration file accordingly (see Chapter 3).
  - If a hole exists in the memory map, access to an unmapped location generates a bus error, and it is flagged in 8th bit of IFR1 register (INT24).
- The estop_1 instruction can be used in your code as a software breakpoint in addition to simulator breakpoints.
- The Port Connect tool supports external peripheral simulation (in I/O memory). For more information on Port Connect, select Help–>Contents in Code Composer Studio. Port Connect online help is listed in the Contents pane.

  Port Connect is supported only for I/O accesses from 0x0 to 0xFFFF.

  However, two functional timer modules are simulated at I/O memory ranges 0x1000 to 0x13FF and 0x2400 to 0x27FF. Port connect is not supported within these ranges.
- The Pin Connect tool supports external interrupt simulation. The following interrupts pins are supported: NMI, SINT2, SINT3 … SINT24. For more information on Pin Connect, select Help–>Contents in Code Composer Studio. Pin Connect online help is listed in the Contents pane.

  Note that SIN24 and SINT22 are not supported because internal timers use them.

- The simulator driver includes I/O memory (a placeholder for peripherals) that supports word reads/writes. This functionality can be used for general access and storage.

  I/O memory is supported for I/O accesses from 0x0 to 0xFFFF.

  However, two functional timer modules are simulated at I/O memory ranges 0x1000 to 0x13FF and 0x2400 to 0x27FF. I/O memory accesses not supported within these ranges.

- When the CPU is writing to an I/O address, the simulator first checks if there is a file connected to that address. The write happens to the file and also in the I/O memory. During the reading of I/O space, the simulator first checks if there is a file connected to that address and reads from there. If there is no file, then the simulator reads from the I/O memory.

- CPU internal registers are visible in the Code Composer Studio Register window. For more information, see the Code Composer Studio online help.

1.1.2 Functional Timer Support

Timer support includes the following:

- Setting up timer registers
- Count down and generation of interrupts
- Timer 0 generates SINT4 and Timer 1 generates SINT22

For more information on the timers, see the *TMS320C55x DSP Peripherals Reference Guide*.

1.1.3 RTDX Support

RTDX™ support includes the following:

- Host-target and target-host communication
- Both small and large memory models are supported

1.1.4 Limitations

The simulator has the following limitations:

- Port Connect is not supported for Data Memory.
- There is no Pin Connect support on Timer input pins.
- Memory map creation and deletion is not supported via the Code Composer Studio menu. However, you can configure the memory system in the simulator configuration file by following the correct syntax.