The new ESA 32bit processor chipset is a fundamental building block for the next generation of space systems. Spacebel Informatique's Target Simulator is a crucial tool that helps in developing and testing characterised and dependable hardware and software systems.

ERC32 Target Simulator SPARC

Overview

The Target Simulator simulates a complete ERC32 based computer. The tool is exceptional in several aspects:

- Besides its simulation of the Integer Unit and its FPU and memory subsystem, it simulates more asynchronous events, such as timers, interrupts, DMA, UART, register level and data level I/O activity.
- It simulates precisely several ASIC's that have been developed for the space industry: the MEC (Memory Controller) and the ATAC (Ada Tasking Accelerator).
- The Target Simulator provides a clock cycle simulation timing accuracy while providing visibility on the internal hardware that exceeds the facilities of an In-Circuit Emulator. It allows to single step interrupt routines and inject interrupts and errors.
- It is based around the very productive Tcl command line interpreter, which is further used for the many additional interfaces: the I/O Simulator, the OS Emulator, the environment simulator and the scoreboard windows.
- The user defined scoreboard window provides the information that is really needed at each break.
- The tool has many flexible interfaces that provides integration facilities through UNIX pipes, which makes it possible to integrate the tool in a more global system-wide simulator and test environment.
- The Target Simulator provides seamless integration with the Thomson Software Products' ADA cross development tools. Moreover it can works in standalone mode too, using other compiler outputs. It supports all industry standard file formats.
- Ultra-fast loading and fast simulation, approaching 1 % of the real-time, enables improved characterisation and validation in all phases of the development cycle, while significantly reducing the need for simulation, prototype and flight hardware.
- The complete hardware context can be saved and reloaded, thereby allowing to repeat rapidly critical tests within long sequences, as required for deep space validation.

The Target Simulator has been designed as a productivity tool that can be adapted for a specific board and to a specific user, and to be integrated in development and validation benches.

Functionality

The Target Simulator is a tool that simulates a computer that is built around the ERC32 32bit core. It basically simulates the IU, FPU, MEC and ATAC. The Target Simulator simulates simultaneously input/output UART activity, watchdog, timers, interrupts (including errors) and DMA transfers with a resolution of one clock cycle.

Its generic marker detection system, that can activate breaks, traces and traps for the OS emulation and I/O simulation subsystems, provides the tool more versatility than modern In-Circuit Emulators, while being completely non-intrusive. During a break, the hardware is frozen. The tool provides a very good register visibility, a high accuracy and allows to single step interrupt routines and inject interrupts and errors.

The full hardware and software context can be saved and restored, which provides the possibility to bypass long preparation phases for complex test sequences.

It is fully user re-configurable in terms of clock speeds and memory banks, sizes and access speeds. PROM content and tool startup procedures are user definable.

User interfaces use symbolic addresses when possible.

All scripts, communication channels, windows and widgets can be altered and expanded by the tool user.

Besides the standard Tcl built-in procedures, the Target Simulator provides a rich set of versatile commands. All command windows have logging and history recall capabilities. The main command window provides direct access to UNIX.

Built-in support to help measuring time between interrupt arrival and servicing.

The power of the tool stems from its flexibility. All its interfaces are Tcl based and can be redirected to communicate with scripts and external processes.

- A very productive user interface with its integrated debugger or with the Thomson Software Products' ADA cross compiler system.
- A scoreboard window displays, after each command or break, user specified variables, such as registers, hardware status and user variables. The user can redefine the Tcl scoreboard procedure and tailor the display to his current needs. Tcl procedures can be used to calculate or validate variables.
- A register level I/O interface provides connectivity to external register level simulators.
- A data packet level I/O interface provides an easy interface to external simulators for DMA, 1553, UART, interrupts and errors.
- A built-in OS emulator provides a facility to quickly emulate characterised OS and external services. This emulator too can be connected with external processes.

Built in counters provide comprehensive statistics (word & byte accesses, IU & FPU activity, fetches, loads and stores, power down ...)

The use of the new generation ERC32 chipset is a major challenge for designers of complex systems. The Target Simulator is a valuable and versatile tool for harnessing and controlling that increased complexity.

ERC32 Target SPARC Simulator

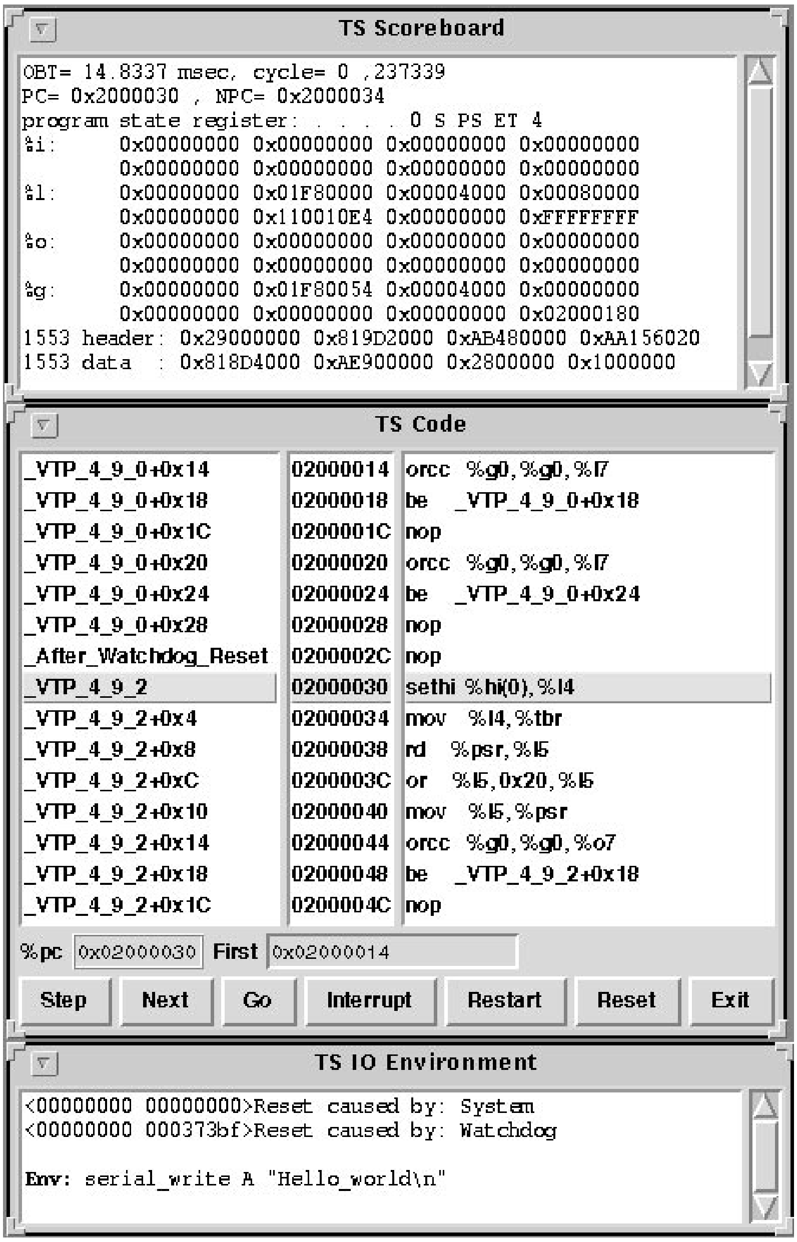
The Target Simulator simulates accurately a complete ERC32 based computer, including input/output systems, with a very good performance. It provides complete hardware visibility, full debugging and ICE capability, without the associated cost and overhead.

Interfaces

The following basic tool commands are available:

assign - Assign value(s) to one or more memory locations or registers
condition - Inquire or update information on debugging conditions
default - Specify the default formats for outputs
disassemble - Disassemble a range of memory words
display - Display logging information in the Scoreboard window
dma - Schedule a DMA access
dump - Dump a memory range to an ASCII file
examine - Examine the contents of one or more memory addresses or registers
exception - Cause a MEC Error
go - Resume the program execution
interrupt - Cause a processor interrupt
load - Load an object file into simulated memory
lookup - Convert a value in a particular format
marker - Inquire or update information on debugging markers
next - Execute the next program instruction, stepping over sub-routines
range_clear - Clear markers and watchpoints in a range of addresses
reset - Reset the board
restart - Reinitialise the simulator
run - Initialise the processor and start execution
serial_write - Queue a message string for UART reception
step - Execute the next program instruction
ticks - Set the number of ticks consumed in the simulation routine
wait_time - Control the timing behaviour of the environment simulation
watchpoint - Inquire or update information on watchpoints

The user can write Tcl procedures to expand on this command set. Tcl shell users have quick access to the history through the cursor keys.
Supported object formats are: a.out, ELF, COFF, S_Record and the tools' ASCII dump format.
Online help through a HTML browser widget.
Interfaces seamlessly with the Thomson Software Products' ADA cross compilation system, which allows for symbolic ADA debugging.



```
OBT= 14.8337 msec, cycle= 0 ,237339
PC= 0x2000030 , NPC= 0x2000034
program state register: . . . . 0 S PS ET 4
%i:     0x00000000 0x00000000 0x00000000 0x00000000
        0x00000000 0x00000000 0x00000000 0x00000000
%l:     0x00000000 0x01F80000 0x00004000 0x00080000
        0x00000000 0x110010E4 0x00000000 0xFFFFFFFF
%o:     0x00000000 0x00000000 0x00000000 0x00000000
        0x00000000 0x00000000 0x00000000 0x00000000
%g:     0x00000000 0x01F80054 0x00004000 0x00000000
        0x00000000 0x00000000 0x00000000 0x02000180
1553 header: 0x29000000 0x819D2000 0xAB480000 0xAA156020
1553 data  : 0x818D4000 0xAE900000 0x2800000 0x1000000
```

```
_VTP_4_9_0+0x14       02000014 orcc  %g0,%g0,%l7
_VTP_4_9_0+0x18       02000018 be    _VTP_4_9_0+0x18
_VTP_4_9_0+0x1C       0200001C nop
_VTP_4_9_0+0x20       02000020 orcc  %g0,%g0,%l7
_VTP_4_9_0+0x24       02000024 be    _VTP_4_9_0+0x24
_VTP_4_9_0+0x28       02000028 nop
_After_Watchdog_Reset 0200002C nop
_VTP_4_9_2            02000030 sethi %hi(0),%l4
_VTP_4_9_2+0x4        02000034 mov   %l4,%tbr
_VTP_4_9_2+0x8        02000038 rd    %psr,%l5
_VTP_4_9_2+0xC        0200003C or    %l5,0x20,%l5
_VTP_4_9_2+0x10       02000040 mov   %l5,%psr
_VTP_4_9_2+0x14       02000044 orcc  %g0,%g0,%o7
_VTP_4_9_2+0x18       02000048 be    _VTP_4_9_2+0x18
_VTP_4_9_2+0x1C       0200004C nop
```

%pc 0x02000030 First 0x02000014

```
<00000000 00000000>Reset caused by: System
<00000000 000373bf>Reset caused by: Watchdog

Env: serial_write A "Hello_world\n"
```

Required Workstation

Any SUN workstation that runs Solaris 2.x and Open Windows or Motif, 8 MByte of free disk space.
The display part of the product can be on another workstation or X-terminal.
The products' performance is directly related to the workstation performance (currently up to 1% of real time)
Multi-CPU workstations can be used to significantly improve the performance of systems where several simulators are coupled.

More?

More Information

It is believed that the contents of this data sheet is correct at publishing time. As the product still evolves, future versions of the product might have slightly different characteristics and features.
Early access available through Estec evaluation programme.

web: http://www.spacebel.be (01/11/96)
email: esa32bit@spacebel.be
SPACEBEL Informatique s.a.
111, rue Colonel Bourg
1140 Brussels - Belgium
Tel: +32 2 730 46 11 Fax: +32 2 726 85 13

A licensing scheme, maintenance and hot-line support is available. Consulting services are negotiable. Help for customisation to other chips and subsequent integration and validation can be arranged.

SPACEBEL INFORMATIQUE s.a.

SPACEBEL Informatique is an engineering company that specialises in dependable, real-time, parallel, reusable object oriented systems for the various space segments