Hardware Modeling using Virtual Platforms

The ESA HW/SW Co-Verification SoC Validation Platform

T. Schuster, R. Meyer
19-Sep-11
Outline

Introduction to Virtual Platforms and TLM
The SoCRocket SystemC Modeling Library
Models Verification
Introduction to Virtual Platforms and TLM
The SocRocket VP
Verification
Design & Productivity Gap

How to manage the rising complexity?

Source: www.itrs.net (2010)
Advantages of Virtual Platforms

Virtual Platforms are abstract hardware models that are simulated by software:

Availability
- Software development can start early

Productivity
- Can be easily duplicated and packaged

Accessibility
- Software is easy to observe and control

Consistency
- Gradual HW refinement / keep your tests
TLM: Abstraction of communication using function calls.

Source: www.osci.org (2011)
Hardware Verification
Cycle-Timed Modeling

Master VHDL

Slave VHDL

address bus
control signals
data bus

address
control
data

Slave samples A1/C1
Slave samples A2/C2
Master samples D2

0 ns 10 ns 20 ns 30 ns 40 ns 50 ns
Some concurrent processes synchronizing at significant points in time.
Software Development
TLM 2.0 Loosely Time Coding Style (LT)

Master

Transaction 1

Transaction 2

Slave

Blocking transport

Very little concurrency – blocking communication only.
Drawbacks of VP Approach

Availability of Simulation IP
Additional EDA costs

In Aerospace domain trusted components are reused many times.

The long IP life cycle repays the simulation models.
Outline

SoCRocket

Introduction to Virtual Platforms and TLM
SoCRocket VP
Verification and Documentation
SoCRocket – VP for Aerospace applications

TLM Simulation Models for Aeroflex GRLIB library
Modeling library and tools

http://www.esa.int/TEC/Microelectronics/SEM151AMT7G_0.html
Library structure

- models
- templates
- tools
- platforms
Library structure

Models

Simulation Models for LT and AT abstraction
LEON processor simulator
Component-level tests
Library structure

Templates for platform models
(e.g. single core, multi-core)
Structural information – top level C code
Parameters for Design Space Exploration
Library structure

Tools

Platform Configurator
(generates platforms from templates)
Co-Simulation Adapters
Power Monitor
Timing Monitor
Library structure

Platforms

- models
- templates
- tools
- platforms

Platform simulators are generated here
Library structure

WAF – The META build system

Python based build system
(code.google.com/p/waf/)
Manages all tools of the library
Configuring the Library

./waf configure

GreenSoCs (GreenSockets, GSPrams, GreenReg)
Open Virtual Platform Infrastructure
www.greensocs.org
ARM/Carbon AMBAKit
TLM2.0 Sockets with AMBA Extensions
www.carbonsystems.org
SystemC/TLM 2.0 Installation
Mentor ModelSim Simulator (optional)
Multi-language simulator for RTL Co-Simulations
www.mentor.com
Python
GCC, GDB, other GNU
Setting up a system
./waf generate

Select an architecture template
(e.g. single core, multi-core, custom)
Configure parameters like cache size, address masks, ...
Generate the platform simulator
Compiling the platform

./waf --target=singlecore

Top-level of SystemC simulation (sc_main)
Instantiation of all components

- Parameter definitions (settings from Platform Configurator)
- Defines constructor parameters of all IP blocks

- Boot code for LEON processor
- Linker script for boot code

- Linker script for application

- WAF build script

Result is executable simulator: ./singlecore boot.elf program.elf
Simulating the platform

Verbosity & Debug

Verbosity of debug output may be configured in four stages (error, warn, info, debug)
Analyze results
Event Monitoring

Event Monitor provides API for Event Recording
IPs register at begin of simulation
\[ EM::registerIP(this, "type") \]

At runtime time-stamps for begin and end of important events are send to the monitor
\[ EM::send(this, "event", 1, sc_time_stamp()) \]

Event Monitor maintains hierarchical database of events
After simulation database is analyzed
\[ EM::analyze("power.dat") \]
Power Profile
Example GPTimer

Annotation of power information to events

Counters enabled
Underflow (counters ticking)

Static Power
Outline

SoCRocket

Introduction to Virtual Platforms and TLM
SoCRocket VP
Verification
Verification
Behavior & Timing

1. Co-Simulation of Original IP

- StimGen
- Transactor
- RTL Reference
- Transactor
- Monitor

2. Simulation of TLM Model

- StimGen
- TLM DUT
- Monitor

TLM Testbed is re-used for both simulations
Timing Monitor keeps track of simulation time
**Example:** Simulation of AHBCTRL with 16 masters and 16 slaves

<table>
<thead>
<tr>
<th>Phase</th>
<th>LT Start</th>
<th>LT End</th>
<th>AT Start</th>
<th>AT End</th>
<th>RTL Start</th>
<th>RTL End</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>10100</td>
<td>100</td>
<td>10100</td>
<td>100</td>
<td>10100</td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>10100</td>
<td>100</td>
<td>20100</td>
<td>100</td>
<td>20100</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>Total</td>
<td>21000</td>
<td>359990</td>
<td>360140</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

LT Timing is far off – LT Router does not model arbitration
Code Coverage

GCOV/LCOV

LCOV - code coverage report

Current view: top level
Test: lcov.info
Date: 2011-07-26

<table>
<thead>
<tr>
<th>Directory</th>
<th>Line Coverage</th>
<th>Functions</th>
<th>Branches</th>
</tr>
</thead>
<tbody>
<tr>
<td>common</td>
<td>100.0 %</td>
<td>74.9 %</td>
<td>50.0 %</td>
</tr>
<tr>
<td>contrib/transactors</td>
<td>15.1 %</td>
<td>40.7 %</td>
<td>5.6 %</td>
</tr>
<tr>
<td>models/archctrl</td>
<td>66.1 %</td>
<td>19.5 %</td>
<td>44.4 %</td>
</tr>
<tr>
<td>models/cppp</td>
<td>88.5 %</td>
<td>36.1 %</td>
<td>46.7 %</td>
</tr>
<tr>
<td>models/ireg</td>
<td>98.0 %</td>
<td>25.6 %</td>
<td>70.2 %</td>
</tr>
<tr>
<td>models/memr</td>
<td>46.2 %</td>
<td>22.9 %</td>
<td>32.0 %</td>
</tr>
<tr>
<td>models/mmu_cache/lib</td>
<td>68.2 %</td>
<td>39.5 %</td>
<td>47.6 %</td>
</tr>
<tr>
<td>models/syswire</td>
<td>54.0 %</td>
<td>41.5 %</td>
<td>33.2 %</td>
</tr>
<tr>
<td>models/utils</td>
<td>70.3 %</td>
<td>45.2 %</td>
<td>55.6 %</td>
</tr>
<tr>
<td>signalkit/signalkit_h</td>
<td>85.1 %</td>
<td>55.2 %</td>
<td>32.9 %</td>
</tr>
</tbody>
</table>

Generated by: LCOV version 1.9

Code coverage still too low for most of the IPs. Verification is work in progress.
Summary

Virtual Platforms are abstract hardware models that are simulated by software.

Virtual Platforms and TLM can help to manage the raising complexity in system design.

The SoCRocket VP provides simulation models and infrastructure for Design Space Exploration of LEON based SoCs.

Project planned to be completed end of 2011