Software Development using Virtual Platforms

Simon Davidmann
Imperas CEO
OVP founder

DAC VP09 Workshop, 29 July 2009
New Tools and Methodology Needed for Embedded Software Development

Adoption of new tools to produce higher quality embedded software

Move to Virtual Platforms for Software development

Multicore Proliferation

Software Complexity Explosion
Virtual Platforms Types

- Hardware Virtual Platforms
  - Timing / Cycle accurate
  - Used for architecture performance analysis, drivers, firmware
  - Models are very complex, slow, time consuming to build
  - SystemC etc
  - Main value is same performance as RTL, no Verilog license cost…

- Software Virtual Platforms
  - Instruction accurate
  - Used for OS, applications
  - Can be very fast, programmers views
  - Model only what is needed in peripherals
  - Complete system environment
New SoC Embedded Software Development market

- Older approach was to use prototypes, breadboards, FPGA, previous generation – this fails to satisfy for next generation complexity

The market need for new embedded software tools for single- and multi-core System on Chip ICs is large and growing

<table>
<thead>
<tr>
<th>Productivity and Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtual platform: multicore simulation, user modeling, model library</td>
</tr>
<tr>
<td>2009</td>
</tr>
<tr>
<td>Adoption over time</td>
</tr>
</tbody>
</table>

- Must address parallel programming and multi-core platform issues
- Needs to remove complexity and infrastructure difficulties and costs while improving verification ability
- Needs to target multi-core SoC software development and provide automation and efficiency with measurable quality building on virtual platforms
- Needs to be fast – to run full software, needs to be low cost – for massive deployment, needs to be easy to use for customers to be self sufficient, with full visibility to enable debug
Virtual Platforms (OVP) are the foundation for the next generation of embedded software development environments – especially for multicore.

Adoption over time

Must address parallel programming and multi-core platform issues

Needs to remove complexity and infrastructure difficulties and costs while improving verification ability

Needs to target multi-core SoC software development and provide automation and efficiency with measurable quality building on virtual platforms

OVP fulfils this need

Open Virtual Platforms

Open Virtual Platforms started March 2008, 28 partners
Today: 1400 registered users, 10,000 downloads, 200+ companies using
What is in OVP?

- **Modeling APIs**
  - Publishing of C OVP APIs for Processor, Peripheral, and Platform modeling
  - Documentation & header files
- **Open Source library of models**
  - C source of models written to C OVP APIs
  - Processor models of ARM, ARC, MIPS, OpenRisc OR1K, x86, …
  - Peripheral models of standard embedded devices
  - Example embedded platforms in C, C++, SystemC, TLM2.0
    - Including full platforms that boot operating systems like Linux, Nucleus
- **OVP reference simulator, free for non-commercial use**
  - Runs processor models fast, 500 MIPS typical
  - Interfaces to GDB via RSP/socket
  - MP Capable, scalable and very efficient
  - Can encapsulate existing processor models (ISS)
  - Callable with C/C++/SystemC wrapper
- **Website community/portal**
- **Ecosystem growing for model and tool developers and users**
OVPsim booting uClinux on ARM Atmel AT91sam7

OVPsim booting uClinux on ARM Atmel AT91sam7
OVPsim booting Nucleus on ARM Integrator / ARM920

Task 1 Event Injections: 22
RX Buffer:

Task 0 Time: 24
Timer Interrupts: 2400
Task 1 Messages Sent: 9503335
Task 2 Messages Received: 9503339
Task 2 Invalid Messages: 0
Task 3/4 Resource Owner: Task 4
Task 5 Event Injections: 23
RX Buffer:

**Easy to run**
telnet localhost 9999
OVPsim booting Linux on ARM Integrator / ARM926

Boot Linux

Page 9
OVPsim MIPS Linux platform

- Boot Linux on Windows PC...
- Easy to **run: platform** .exe vmlinux
Software debug on prototype: Run gdbserver on target and Eclipse on host to debug application on target.
Using a Virtual Platform from OVP provides exactly the same environment

TCP/IP

Host

Virtual Platform as Remote Target

Eclipse/CDT

Redhat Linux

Debian Linux 2.6

BlackBox Window Manager

gdbserver

application
Virtual Platforms are replacing hardware prototypes for software development

- It’s a new world, with familiar tools
- Virtual Platforms provide significant benefits
- Software development uses standard tools
  - Eclipse, GDB, gdbserver

Benefits of Virtual Platforms for software development

Easier to deploy, more reliable, deterministic

Often much faster

Easier to use for regression testing, improving quality

Available much sooner (developing SoC takes 2 years of delay…)

With OVP, the models are free, fast, and easy to use…
Thank you

For more information visit www.OVPworld.org