A Common Software Development Environment for Many-core RISC-V based Hardware and Virtual Platforms

RISC-V 7th Workshop – Barcelona
Tuesday May 08, 2:00pm

Simon Davidmann – Imperas Software
Gajinder Panesar – UltraSoC
Agenda

- Embedded Software Development Challenges
- Traditional SW Debug Environment using Hardware
- Modern SW Debug using Imperas Simulation
- Giving more visibility to SW Debug using UltraSoC Hardware
- Imperas/UltraSoC collaboration provides common solution
Agenda

- Embedded Software Development Challenges
- Traditional SW Debug Environment using Hardware
- Modern SW Debug using Imperas Simulation
- Giving more visibility to SW Debug using UltraSoC Hardware
- Imperas/UltraSoC collaboration provides common solution
New Markets with new Software Requirements

- Schedule
- Quality
- Reliability
- Security
- Safety
- Engineering productivity / automation
- Predictability on software development schedules
- Unknown / unmeasurable software delivery risk
Chip designs get more complex inc.
Asymmetric Multi-Core…

- Example… SMP CPU groups, AMP CPU, many peripherals and other processors…
Embedded Software Increasingly Important & Engineering Intensive

- Functionality of SW is defining embedded products
- SW codebase size, complexity, quality requirements exploding
- Engineering schedules & costs under pressure and harder to manage

![Engineering Cost vs SoC Process Node]

*Source: System Level Design Community 2013*
Agenda

- Embedded Software Development Challenges
- Traditional SW Debug Environment using Hardware
- Modern SW Debug using Imperas Simulation
- Giving more visibility to SW Debug using UltraSoC Hardware
- Imperas/UltraSoC collaboration provides common solution
Traditionally use a hardware breadboard... Using GDB(s)...

Well.. You use one GDB per processor with JTAG giving access...
Traditionally use a hardware breadboard… Using GDB(s)…

- one GBD per CPU
- little visibility: each GDB only sees the limited memory space of the attached CPU
- non-deterministic (bugs move around…)
- poor control (hard to set specific places to break)
- scheduling and synchronized events difficult to reproduce…
- what is going on in the peripherals? …
- pretty un-manageable…
Agenda

- Embedded Software Development Challenges
- Traditional SW Debug Environment using Hardware
  - Modern SW Debug using Imperas Simulation
  - Giving more visibility to SW Debug using UltraSoC Hardware
  - Imperas/UltraSoC collaboration provides common solution
Simulating Hardware (Virtual Platforms) with Imperas

- Virtual Platforms built using processor, peripheral and platform models using Open Virtual Platforms (OVP) APIs
  - Models are open source
- Simulated unmodified production binaries
- Software does not know it is not on hardware
- Runs very fast, 100-2,000MIPS

Imperas U54-MC Virtual Platform

Under 10 seconds to get to booted Linux login prompt!
Imperas MPD full platform debugger

- Platform aware, multiprocessor / multicore aware
- Driver-peripheral software-software/hardware co-debug
- Event-based debug, e.g. using assertions
- OS-aware debug, e.g. breakpoints on OS events

Select CPU or peripheral in target window and see source, programmers registers, variables, …

Complete visibility in platform…
Agenda

- Embedded Software Development Challenges
- Traditional SW Debug Environment using Hardware
- Modern SW Debug using Imperas Simulation
- Giving more visibility to SW Debug using UltraSoC Hardware
- Imperas/UltraSoC collaboration provides common solution
Advanced debug/monitoring for the whole SoC

Interconnect (AXI, ACE, ACE-lite, OCP, NoC)

Portfolio of Analytic Modules
Flexible & Scalable Message Fabric
Family of Communicators

System Block
UltraSoC IP

© 2018 UltraSoC
Agenda

- Embedded Software Development Challenges
- Traditional SW Debug Environment using Hardware
- Modern SW Debug using Imperas Simulation
- Giving more visibility to SW Debug using UltraSoC Hardware
- Imperas/UltraSoC collaboration provides common solution
Collaboration

- Common Software Development Environment

- Hardware with UltraSoC Debug IP

- Imperas MPD Debugger

- Eclipse GUI

- UltraSoC Interface
Collaboration allows use from concept to production

- Views to control and debug and observe throughout the design process
Summary

- Imperas and UltraSoC collaborating
  - Developing common software development environment
  - Same for hardware based and simulation based teams

- Start with simulation on virtual platforms
- Use with RTL as design progresses
- Use prototypes when available
- Use with silicon pre- and post-production