MULTICORE DESIGN SIMPLIFIED

imperas

RISC-V Models and Simulation Enable Early Software Bring Up

RISC-V

The 15th International System-on-Chip (SoC) Conference, Exhibit, and Workshops, October 2017, Irvine, California
www.SoCconference.com

Larry Lapides
Imperas Software Ltd

But first, a bit of history

- Array of processors architectures, with proprietary ISAs, have been proposed and built regularly over the last 20 years
- Very few have survived, let alone be successful

- While there are various reasons for the lack of success, one constant is the lack of a good software development environment for these devices
- A second constant was a love of the architectures by the founders

Quicksilver ?	*Equator ?*
Chameleon ?	*PACT ?*
Morphics ?	*Systolix ?*
Chromatic Research ?	*Mathstar ?*
Tabula ?	*Ambric ?*
Element CXI ?	*….*

Agenda

- With all due respect to the ISA, RISC-V success is all about the software ecosystem
- What is a virtual platform?
- Virtual platforms for software porting and bring up
- Virtual platforms for software debug and testing
- Status of Open Virtual Platforms (OVP) RISC-V models and platforms
- Demo of a RISC-V virtual platform running FreeRTOS
- Summary

Agenda

- With all due respect to the ISA, RISC-V success is all about the software ecosystem
- What is a virtual platform?
- Virtual platforms for software porting and bring up
- Virtual platforms for software debug and testing
- Status of Open Virtual Platforms (OVP) RISC-V models and platforms
- Demo of a RISC-V virtual platform running FreeRTOS
- Summary

Where Will RISC-V Be Used?

MULTICORE DESIGN SIMPLIFIED
Imperas

- IoT applications
  - Low cost, low power, performance requirements do not push the envelope
- Heterogeneous platforms with RISC-V as a “minion” processor, doing the power management, communications, and other secondary functions
  - Low cost, low power, performance requirements do not push the envelope
- Applications processors for mobile, networking, ADAS?
  - RISC-V features, maturity make winning this business unlikely in the near future

RISC-V Can Win Only If Software Ecosystem is Robust

- If RISC-V wins are in areas where low cost and low power are important, and performance requirements do not push the envelope, then how to win?
- Must combine technical and business advantages with ease of use
  - Technical advantages: competitive PPA (power performance area)
  - Business advantages: open source architecture, royalty-free licensing
  - Implementation: need to have an easy RTL flow
    - Processor IP and EDA companies can and are satisfying this requirement
  - Software: need to be able to easily move from existing devices to RISC-V based SoCs

The Measure of Success for RISC-V is Adoption by Embedded Systems Companies

MULTICORE DESIGN SIMPLIFIED
imperas

- Semiconductor vendors can build SoCs, but those SoCs need to be purchased and used in embedded systems
- Embedded systems derive more value from their software than the hardware
- For embedded systems adoption, the software ecosystem needs to be broad and deep
  - Tool chains
  - Operating systems
  - Development, debug and test tools
  - Libraries
  - Security
  - …

The Measure of Success for RISC-V is Adoption by Embedded Systems Companies

MULTICORE DESIGN SIMPLIFIED
imperas

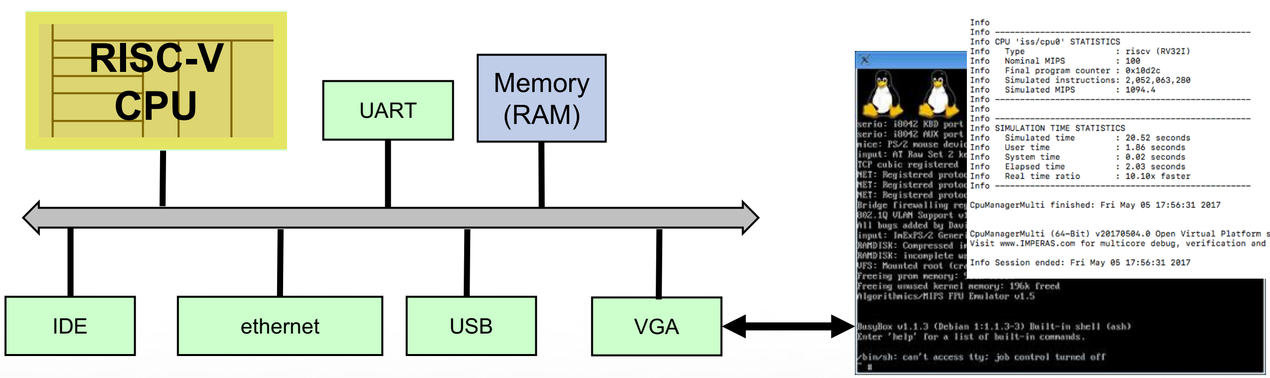
- Semiconductor vendors can build SoCs, but those SoCs need to be purchased and used in embedded systems
- Embedded systems derive more value from their software than the hardware
- For embedded systems adoption, the software ecosystem needs to be broad and deep
  - Tool chains
  - Operating systems
  - **Development, debug and test tools**
  - Libraries
  - Security
  - …

Agenda

- With all due respect to the ISA, RISC-V success is all about the software ecosystem
- What is a virtual platform?
- Virtual platforms for software porting and bring up
- Virtual platforms for software debug and testing
- Status of Open Virtual Platforms (OVP) RISC-V models and platforms
- Demo of a RISC-V virtual platform running FreeRTOS
- Summary

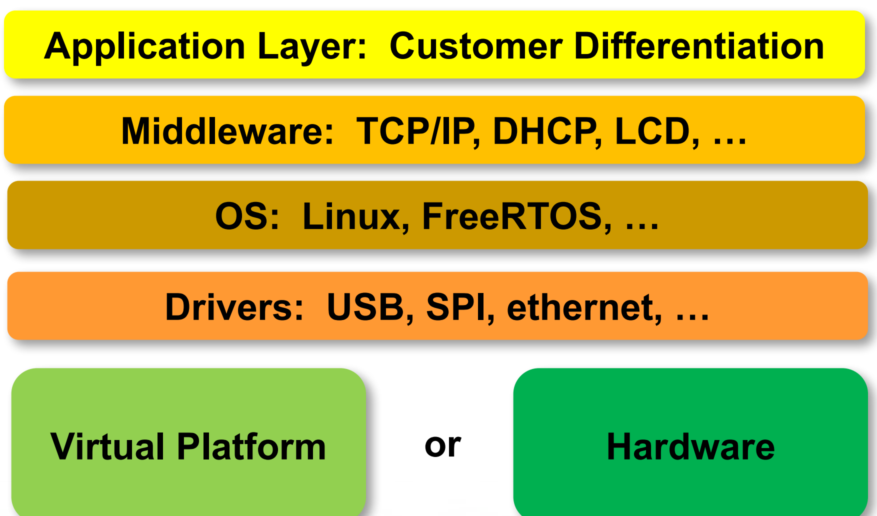
Virtual Platforms Provide a Simulation Environment Such That the Software Does Not Know That It Is Not Running On Hardware

- The virtual platform is a set of instruction accurate models that reflect the hardware on which the software will execute
  - Could be 1 CPU, 1 SoC, multiple SoCs, board, system; no physical limitations
- *Runs the executables compiled for the target hardware*
- Models are typically written in C or SystemC
- Models for individual components – interrupt controller, UART, ethernet, … – are connected just like in the hardware
- Peripheral components can be connected to the real world by using the host workstation resources: keyboard, mouse, screen, ethernet, USB, …
- Typical performance is 200-500 million instructions per second



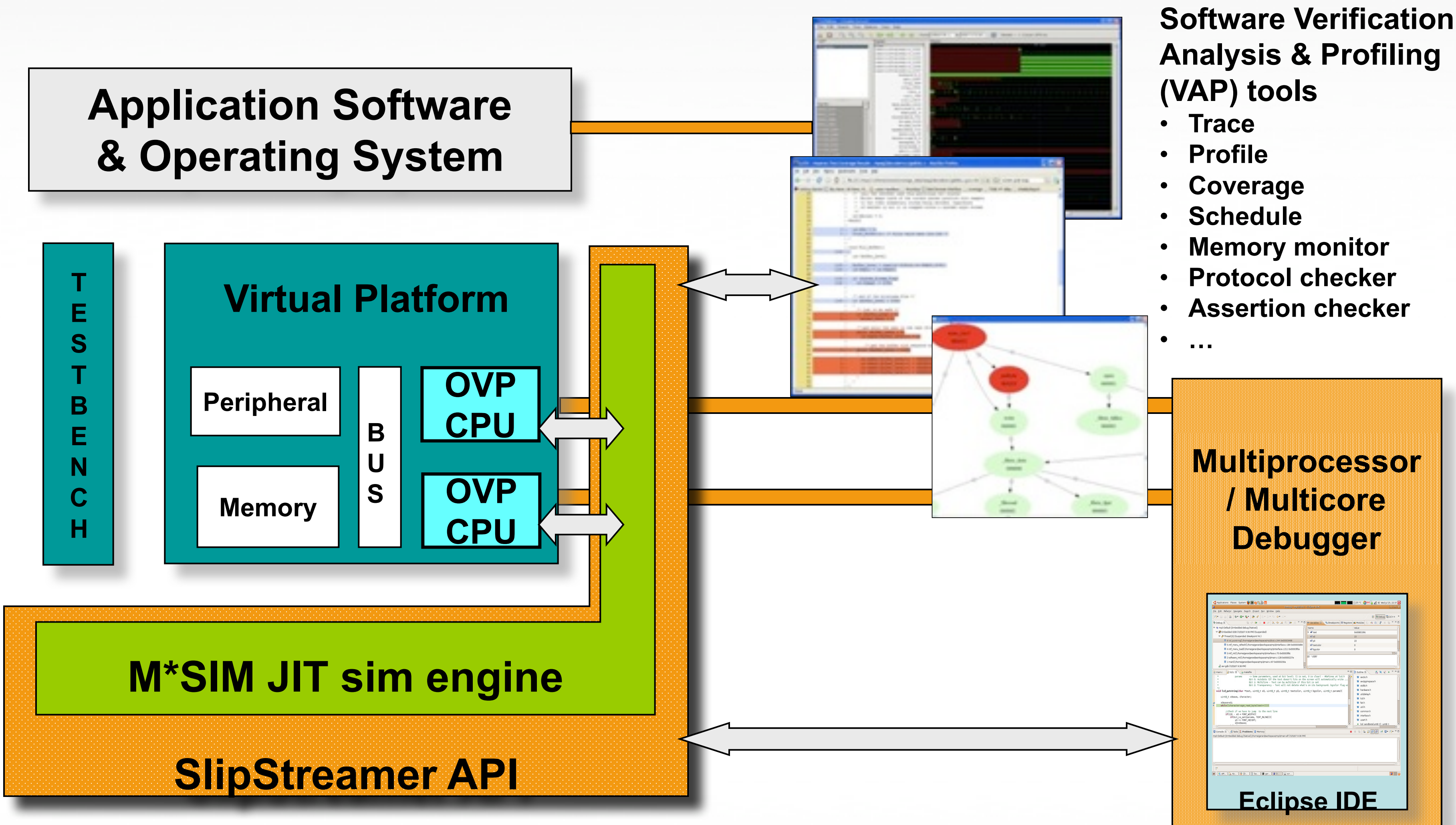
Virtual Platforms are an Integral Part of a Modern Embedded Software Development Methodology

- Virtual platform based methodology delivers controllability, visibility, repeatability, automation, access
  - 75-90% of bugs are functional, and can be found using software simulation testing
- Testing of timing sensitive software, and final testing, still needs to be done on hardware



Virtual platforms – software simulation – provide a complementary technology to hardware platforms

Simulation Architecture Can (Should) Include Tools



Agenda

MULTICORE DESIGN SIMPLIFIED
Imperas

- With all due respect to the ISA, RISC-V success is all about the software ecosystem
- What is a virtual platform?
- Virtual platforms for software porting and bring up
- Virtual platforms for software debug and testing
- Status of Open Virtual Platforms (OVP) RISC-V models and platforms
- Demo of a RISC-V virtual platform running FreeRTOS
- Summary

Virtual Platforms Provide a Pre-Silicon Software Development Solution

MULTICORE DESIGN SIMPLIFIED
imperas

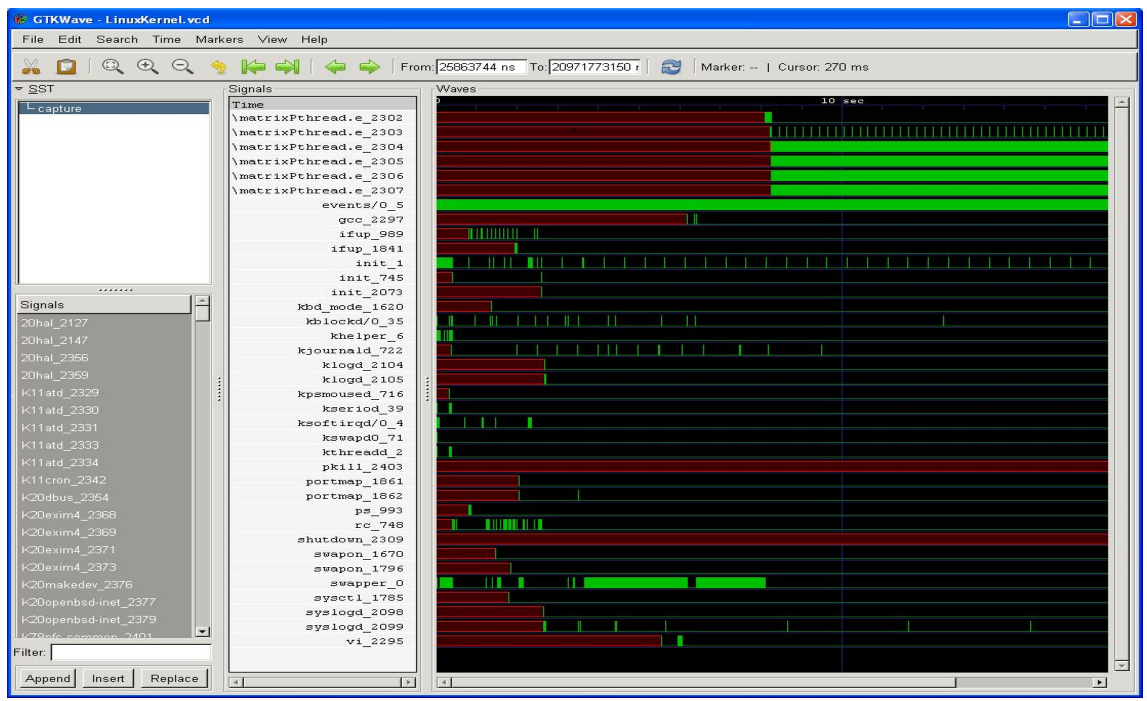
- Need to start software development earlier in project
  - Before silicon is available
  - Before RTL is available
- Need to port and bring up operating systems
- Need to develop drivers
- Need to develop firmware, test libraries, …

- Because virtual platforms do not require the same level of accuracy as RTL, the virtual platform can be ready months, typically 2-6 months, earlier
- This can mean significant schedule reduction, and/or more time for software testing (higher quality software)

OS Porting and Bring Up

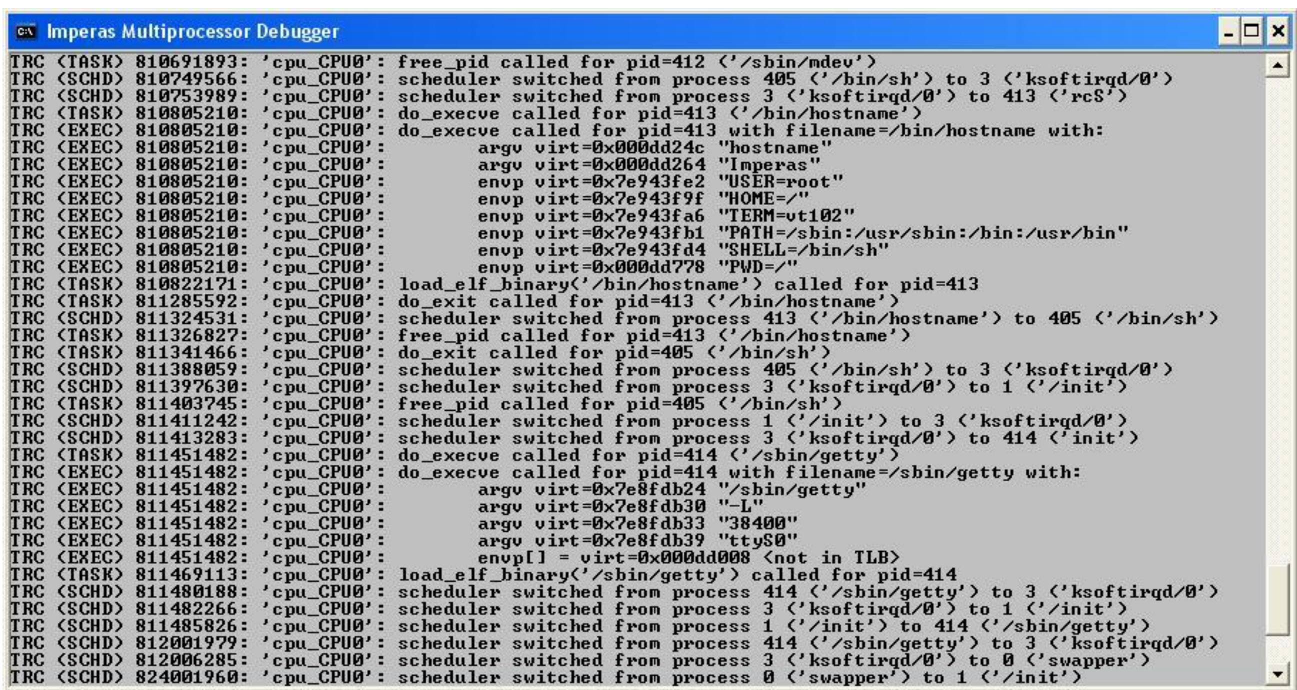
MULTICORE DESIGN SIMPLIFIED
imperas

- Non-intrusive (no modification of OS source) tools for trace of
  - process creation
  - context switch
  - process deletion
- Captures communications between processes
- Supported OS include Linux, FreeRTOS, µC/OS, iTron, …
  - < 1 week to support new RTOS
- View in waveform viewer



OS-Aware Tools Used to Find Bugs

- Use OS tracing [task, execve, schedule, context, …] to trace at the OS level, not instruction level
  - Higher level of abstraction makes debug easier: ~700,000,000 instructions to boot Linux, however, only ~700 tasks
- OS-aware tools can reduce debug time by 5x over hardware-based debug
- Simulation overhead due to OS-aware tools < 10%



Agenda

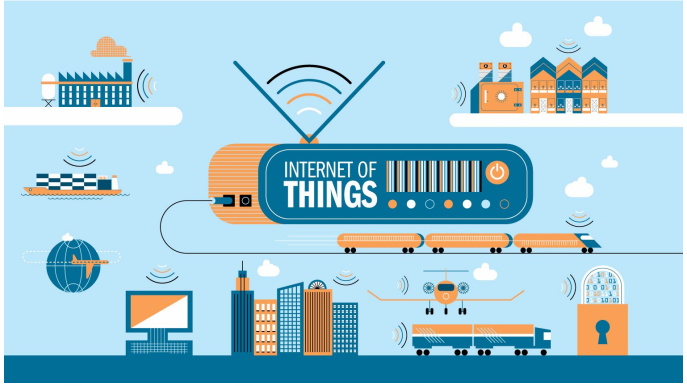
MULTICORE DESIGN SIMPLIFIED
imperas

- With all due respect to the ISA, RISC-V success is all about the software ecosystem
- What is a virtual platform?
- Virtual platforms for software porting and bring up
- Virtual platforms for software debug and testing
- Status of Open Virtual Platforms (OVP) RISC-V models and platforms
- Demo of a RISC-V virtual platform running FreeRTOS
- Summary

Virtual Platforms Enable Improved Software Testing

MULTICORE DESIGN SIMPLIFIED
Imperas

- In many markets reliability, safety and security are critical to product success
- Need more comprehensive software testing
  - Automation: Continuous Integration (CI) and regression testing
  - Code coverage, profiling, memory monitoring and analysis, …
  - Fuzzing and fault simulation
  - Need to demonstrate compliance with standards
  - Silicon may be available, however, silicon lacks visibility, controllability, repeatability, ease of automation, ease of access/update



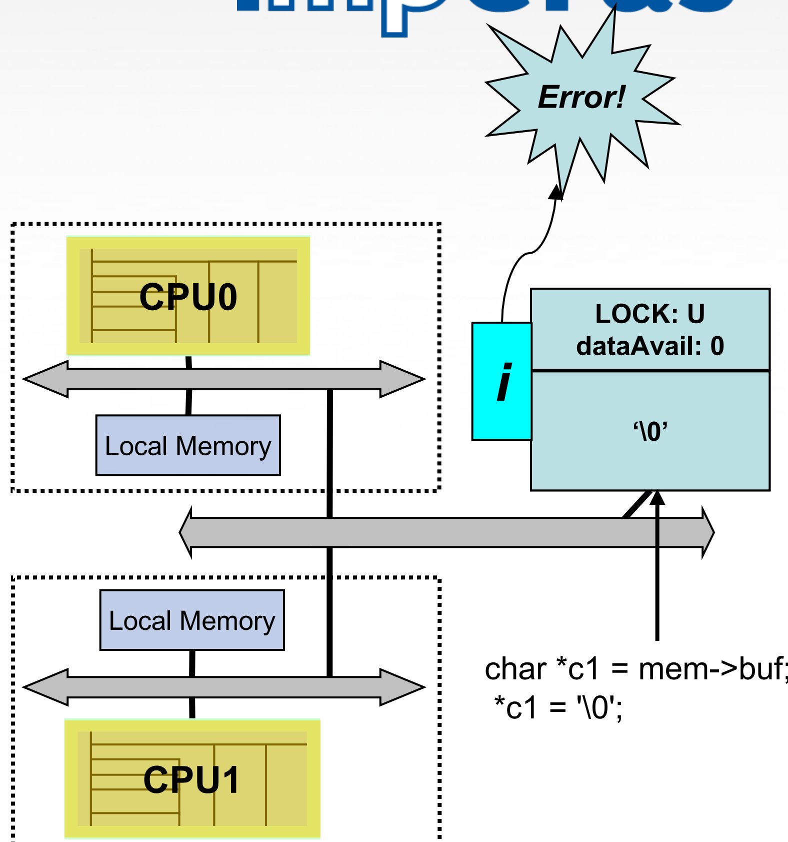
Jeep hacked in 2015

Key Tools & Technologies for Security Testing

MULTICORE DESIGN SIMPLIFIED

imperas

- Code coverage
- Fault injection
- Hypervisor-aware debug and analysis
- Custom tools, e.g. assertion checkers
  - SlipStreamer API enables users to develop custom tools
    - Tools are written in C, compiled to host machine, loaded into simulation environment
  - Example: checker for shared memory protocol



- Test automation is needed

- *i* is assertion checker
- If access is made without a lock by that CPU, error issued

Custom Memory Access Monitor Easily Identifies Illegal Accesses

MULTICORE DESIGN SIMPLIFIED imperas

- Memory access monitor is just C code, less than 350 lines, loaded into simulation environment
- When simulation is run, monitor produces warning if memory access rules are violated

```
//
// Define watch areas for memory and peripherals defined in the platform
//
memWatchT amcWatch[] = {
//   name                               watchLow        watchHigh       allowedCPUs
    { "Linux memory",                  0,              0x2fffffff,     LINUX_CPU   },
    { "uCOS memory",                   0x30000000,     0x31ffffff,     UCOSII_CPU  },
    { "gmac0",                         0xff700000,     0xff700fff,     LINUX_CPU   },
    { "emac0_dma",                     0xff701000,     0xff701fff,     LINUX_CPU   },
    { "gmac1",                         0xff702000,     0xff702fff,     LINUX_CPU   },
    { "emac1_dma",                     0xff703000,     0xff703fff,     LINUX_CPU   },
    { "uart0",                         0xffc02000,     0xffc02fff,     LINUX_CPU   },
    { "uart1",                         0xffc03000,     0xffc03fff,     UCOSII_CPU  },
    { "CLKMGR",                        0xffd04000,     0xffd04fff,     LINUX_CPU   },
    { "RSTMGR",                        0xffd05000,     0xffd05fff,     LINUX_CPU   },
    { "SYSMGR",                        0xffd08000,     0xffd08fff,     LINUX_CPU   },
    { "GIC",                           0xfffec000,     0xfffedfff,     LINUX_CPU   },
    { "L2",                            0xfffef000,     0xfffeffff,     LINUX_CPU   },
    { 0 } /* Marks end of list */
};
```

```
Warning (AMPCHK_MWV) cpu_CPU0: AMP write access violation in uart1 area. PA: 0xffc03008 VA: 0xffc03008
Warning (AMPCHK_MWV) cpu_CPU0: AMP write access violation in uart1 area. PA: 0xffc0300c VA: 0xffc0300c
Warning (AMPCHK_MWV) cpu_CPU0: AMP write access violation in uart1 area. PA: 0xffc03010 VA: 0xffc03010
Warning (AMPCHK_MRV) cpu_CPU1: AMP read access violation in Linux memory area. PA: 0x00000020 VA: 0x00000020
```

Agenda

- With all due respect to the ISA, RISC-V success is all about the software ecosystem
- What is a virtual platform?
- Virtual platforms for software porting and bring up
- Virtual platforms for software debug and testing
- Status of Open Virtual Platforms (OVP) RISC-V models and platforms
- Demo of a RISC-V virtual platform running FreeRTOS
- Summary

RISC-V Models & Platforms Available From Open Virtual Platforms (OVP)

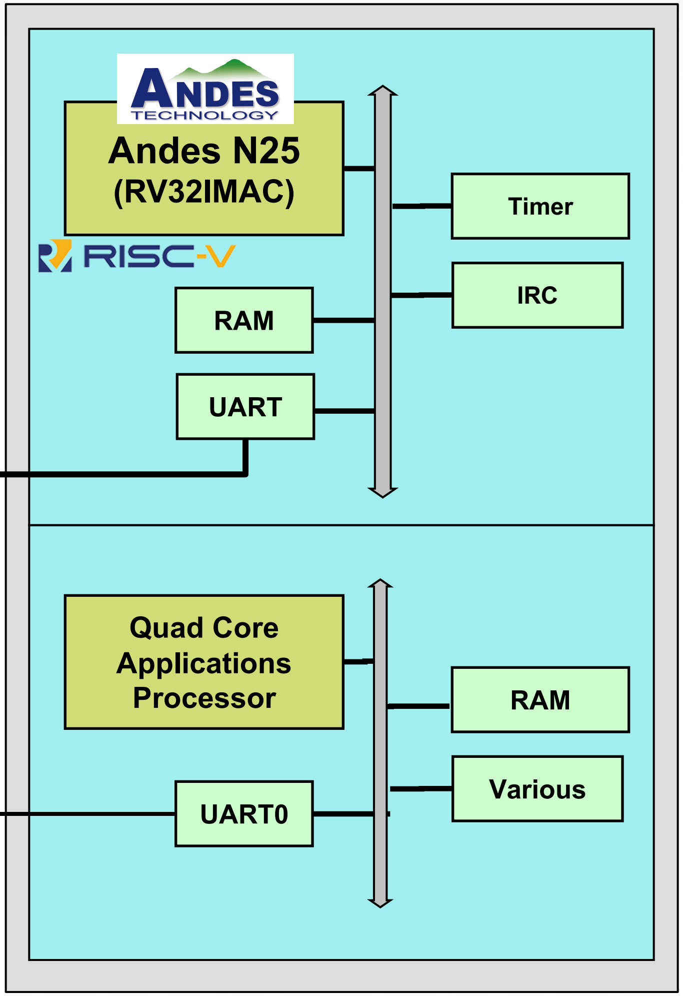
MULTICORE DESIGN SIMPLIFIED
imperas

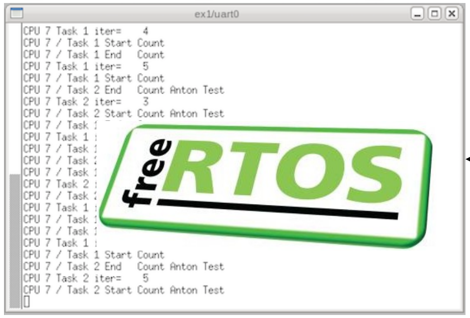
- Models and platforms are open source (Apache 2.0 license)
- Processor models
  - Generic RISC-V models
    - RV32IMAC
    - RV64IMAC
  - SiFive
    - E31
    - E51
  - Andes
    - N25
    - NX25
- Extendable Platform Kits (EPKs)
  - Microsemi SmartFusion2 RISC-V FreeRTOS EPK
  - Andes FreeRTOS EPK
  - Andes – Linux Heterogeneous EPK

www.OVPworld.org

Andes RISC-V – Linux EPK

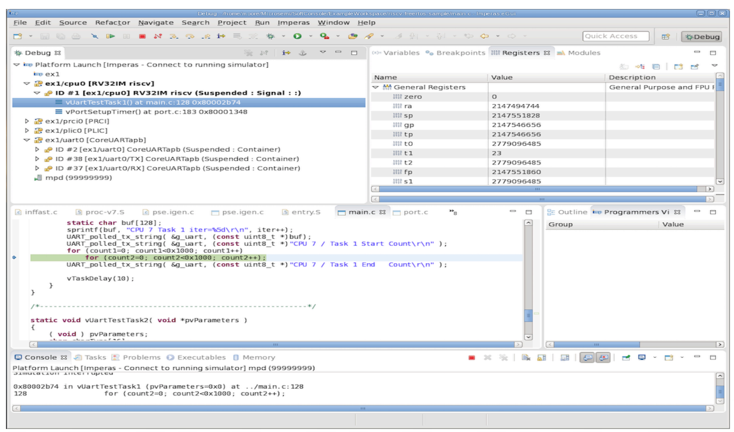
- Extendable Platform Kits (EPKs) are virtual platforms, with software running, to help users start quickly
- EPKs include
  - Individual models, binary and source
  - Platform model, binary and source
  - Software and/or OS running on platform





Andes – Imperas Collaboration

- Imperas has developed models of the RISC-V based Andes NX25 and N25 processor cores, with real-time simulation performance
- These models, together with Imperas tools, enable simulation-based development, debug and test of software running on the Andes cores
- The Imperas solution supports heterogeneous platforms
- Use of virtual platforms enables early (pre-silicon) software development, shaving months off typical project schedules



Imperas platform-centric debugging allows coherent debug of the entire platform

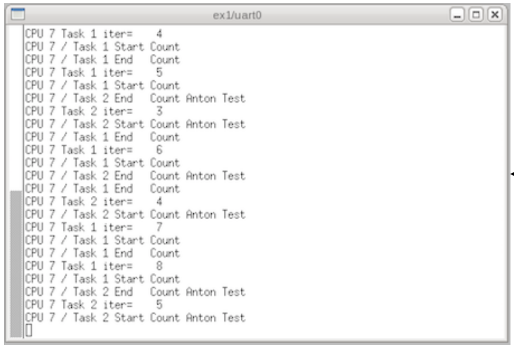
Agenda

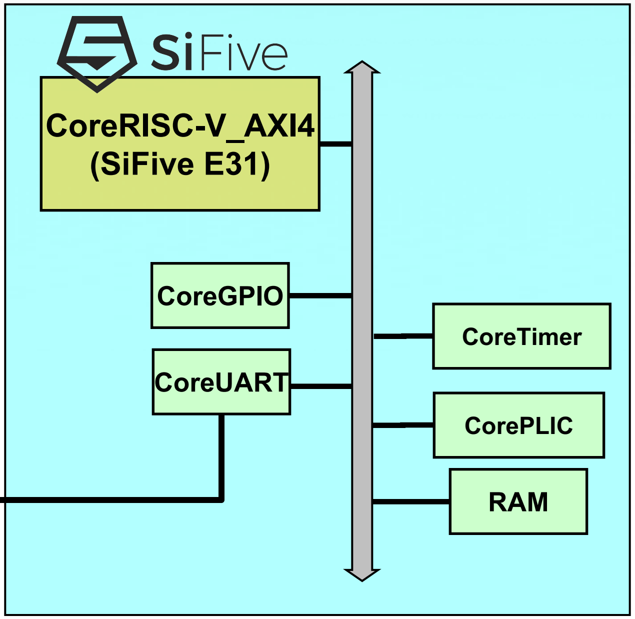
- With all due respect to the ISA, RISC-V success is all about the software ecosystem
- What is a virtual platform?
- Virtual platforms for software porting and bring up
- Virtual platforms for software debug and testing
- Status of Open Virtual Platforms (OVP) RISC-V models and platforms
- Demo of a RISC-V virtual platform running FreeRTOS
- Summary

Microsemi SmartFusion2 / SiFive E31 / FreeRTOS EPK

MULTICORE DESIGN SIMPLIFIED
imperas

- Extendable Platform Kits (EPKs) are virtual platforms, with software running, to help users start quickly
- EPKs include
  - Individual models, binary and source
  - Platform model, binary and source
  - Software and/or OS running on platform





freeRTOS

Microsemi
Power Matters.™

RISC-V

Agenda

MULTICORE DESIGN SIMPLIFIED
imperas

- With all due respect to the ISA, RISC-V success is all about the software ecosystem
- What is a virtual platform?
- Virtual platforms for software porting and bring up
- Virtual platforms for software debug and testing
- Status of Open Virtual Platforms (OVP) RISC-V models and platforms
- Demo of a RISC-V virtual platform running FreeRTOS
- Summary

Summary

MULTICORE DESIGN SIMPLIFIED
imperas

- RISC-V success is dependent on the software ecosystem
- Virtual platforms complement and supplement hardware-based testing
  - Schedule reduction
  - More comprehensive testing
- RISC-V processor models and virtual platforms are starting to be available

MULTICORE DESIGN SIMPLIFIED

imperas

Thank you