MULTICORE DESIGN SIMPLIFIED

imperas

Software Quality is Directly Proportional to Simulation Speed

CDNLive!
11 March 2014

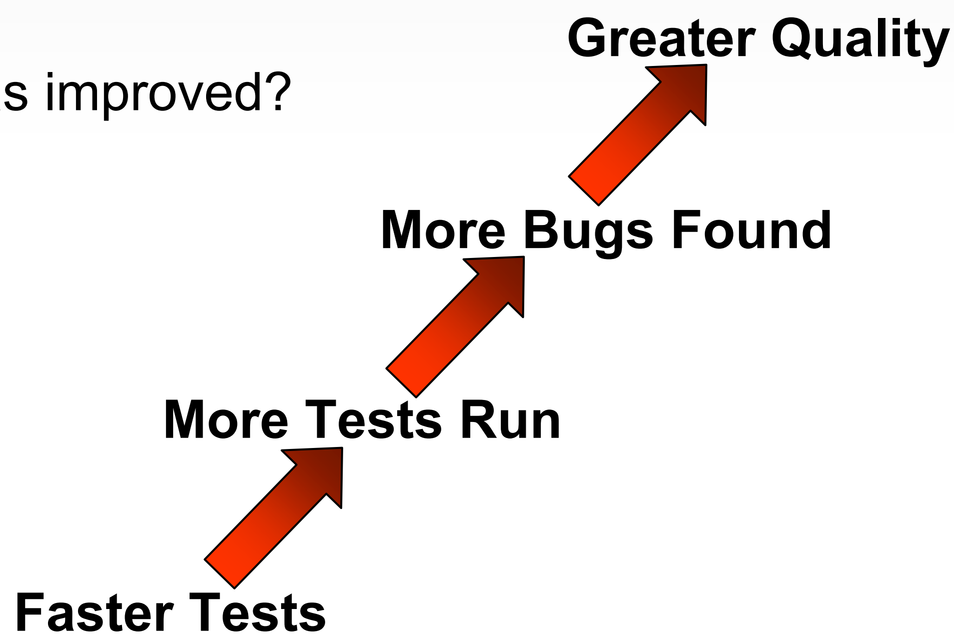
Larry Lapides

© 2014 Imperas Software Ltd

Software Quality is Directly Proportional to Test Speed

MULTICORE DESIGN SIMPLIFIED
Imperas

- Intuitively obvious (so my presentation is done!)
- How to achieve more speed?
- How to find more bugs?
- How to know that quality has improved?



© 2014 Imperas Software Ltd

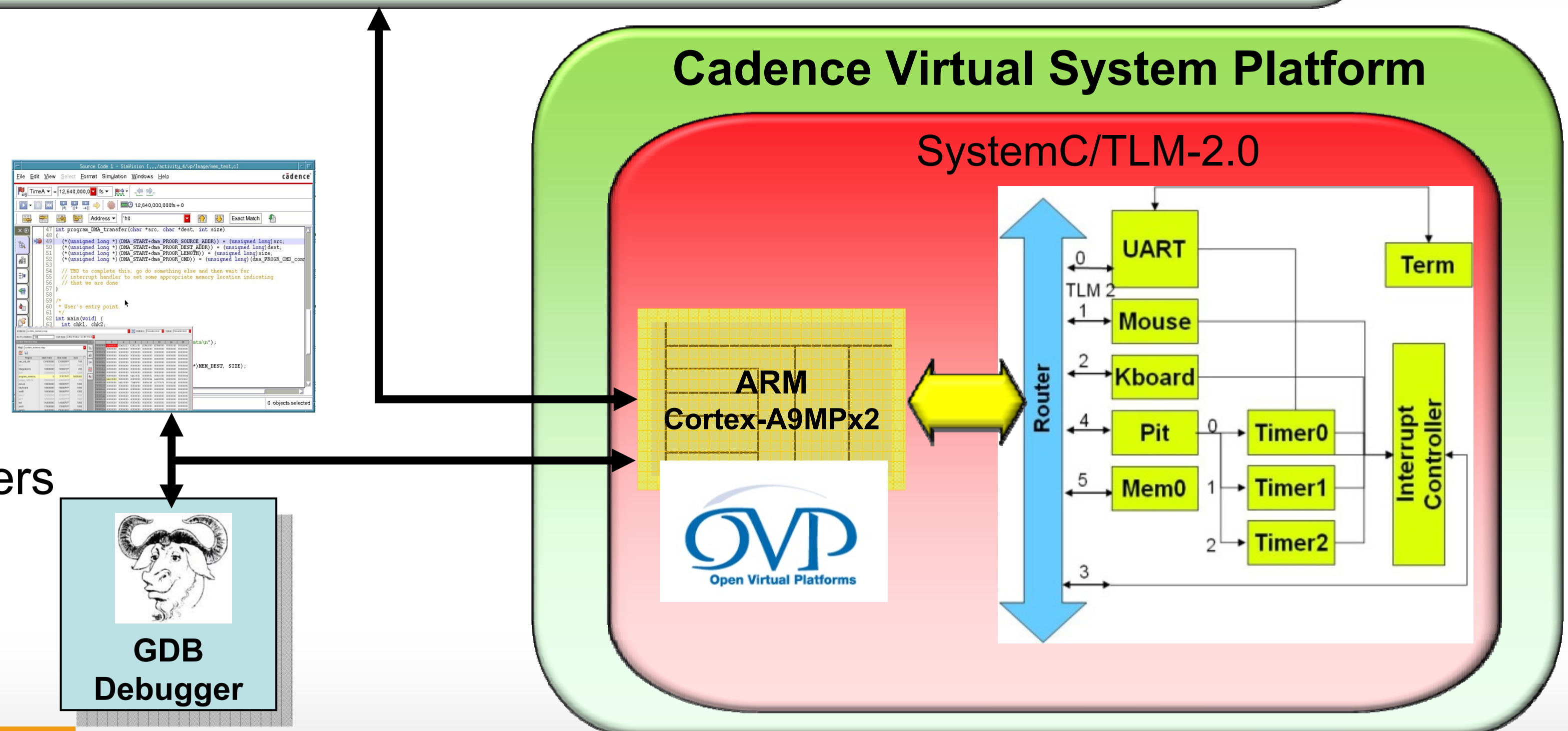
Cadence-Imperas Integration Supports Simulation, Debug and Software Development & Test Tools

MULTICORE DESIGN SIMPLIFIED

Verification, Analysis & Profiling (VAP) Tools

- CPU and OS awareness
- Tracing, profiling, coverage, memory analysis, …
    - Over 25 different tools
- User extendable

Cadence and GDB debuggers supported



© 2014 Imperas Software Ltd

Agenda

MULTICORE DESIGN SIMPLIFIED Imperas

- Simulation speed
- Simulation based tools for finding bugs
- Quality metrics
- Case study: OS porting, bring up and verification on Altera Cyclone V SoC FPGA
- Summary

© 2014 Imperas Software Ltd

Agenda

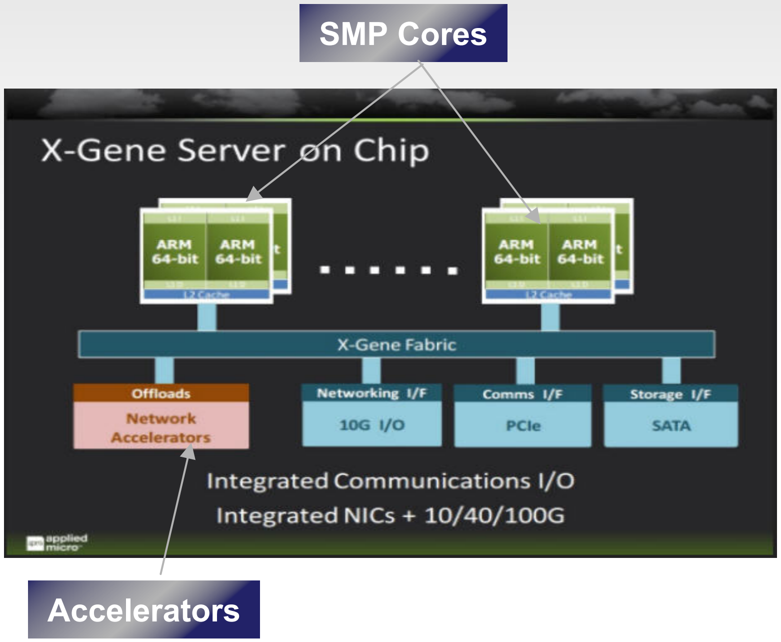
MULTICORE DESIGN SIMPLIFIED Imperas

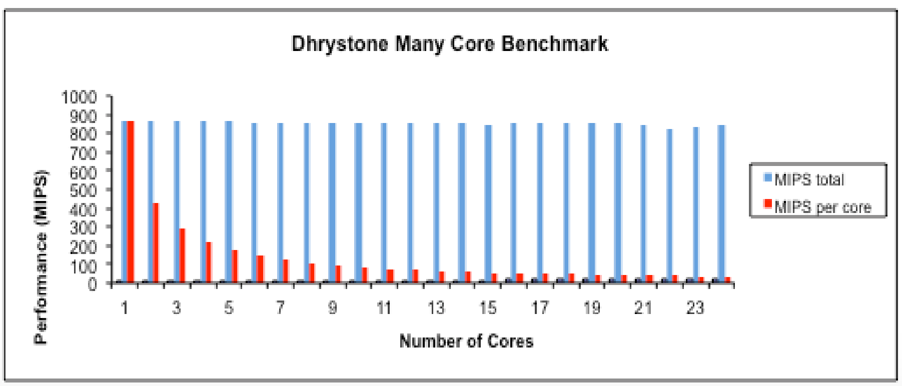
- Simulation speed
  1) Start with a faster simulation engine
  2) Use the multiple cores available in the host PC
- Simulation based tools for finding bugs
- Quality metrics
- Case study: OS porting, bring up and verification on Altera Cyclone V SoC FPGA
- Summary

© 2014 Imperas Software Ltd

Latest Many-Core Platforms Require Scalable Simulation

MULTICORE DESIGN SIMPLIFIED
imperas





- Server SoC software test suites can consist of 10s or 100s of tests, each executing 10s or 100s of billions of instructions
- Currently: Single threaded simulation does not scale with multicore platforms
- Simulation market leader's solution would take 1 week to simulate that test suite
- Challenges: Get needed simulation speed and fix platform simulation scaling problem

© 2014 Imperas Software Ltd

1) Start with a Faster Simulation Engine

MULTICORE DESIGN SIMPLIFIED
Imperas

	Altera Nios II			ARM32			Imagination MIPS32		
Benchmark	Simulated Instructions	Run time	Simulated MIPS	Simulated Instructions	Run time	Simulated MIPS	Simulated Instructions	Run time	Simulated MIPS
linpack	3,075,857,171	2.52s	1225	6,105,766,856	4.79s	1277	9,814,621,392	5.31s	1852
Dhrystone	1,810,082,387	1.18s	1547	2,250,079,359	2.32s	974	1,795,088,667	1.27s	1414
Whetstone	5,850,887,389	3.28s	1789	1,185,959,501	1.04s	1140	1,890,420,892	0.93s	2033
peakSpeed2	22,000,013,458	3.11s	7097	22,400,008,766	4.67s	4807	22,800,009,853	4s	5714
	Xilinx MicroBlaze			ARM AArch64			Imagination MIPS64		
Benchmark	Simulated Instructions	Run time	Simulated MIPS	Simulated Instructions	Run time	Simulated MIPS	Simulated Instructions	Run time	Simulated MIPS
linpack	6,386,275,159	3.77s	1699	594,945,589	1.01s	594*	1,558,856,686	0.83s	1901
Dhrystone	3,770,115,740	2.61s	1450	3,030,061,475	2.79s	1086	1,590,094,345	1.23s	1293
Whetstone	27,108,532,655	13.23s	2054	488,724,620	0.64s	759*	2,133,926,552	0.99s	2156
peakSpeed2	22,000,023,433	5.76s	3826	11,200,003,894	3.73s	3011	17,100,018,075	4.23s	4052
	PowerPC			Renesas v850			Synopsys ARC		
Benchmark	Simulated Instructions	Run time	Simulated MIPS	Simulated Instructions	Run time	Simulated MIPS	Simulated Instructions	Run time	Simulated MIPS
linpack	3,163,966,113	2.95s	1076	4,991,344,159	4.76s	1051	4,184,162,664	3.67s	1143
Dhrystone	2,205,068,239	1.75s	1260	6,410,133,101	4.01s	1603	3,155,082,476	2.75s	1148
Whetstone	6,424,865,755	3.97s	1622	10,296,940,591	7.41s	1393	7,883,567,047	4.4s	1796
peakSpeed2	22,400,002,937	5.6s	4007	22,400,007,569	3.53s	6364	22,000,002,100	4.05s	5446

All measurements on 3.40GHz Intel i7-3770, Linux, OVPsim 20140127.0

* Hardware Floating Point Instructions

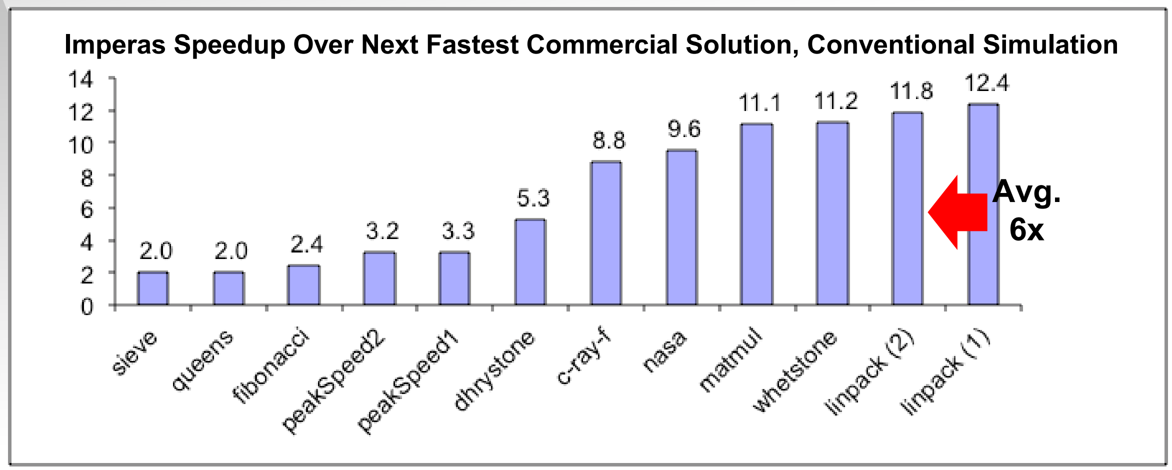
Imperas Simulator Benchmarks

© 2014 Imperas Software Ltd

Imperas: Fastest Virtual Platform Solution Available

MULTICORE DESIGN SIMPLIFIED
imperas

Just In Time (JIT) Code Morphing Simulator

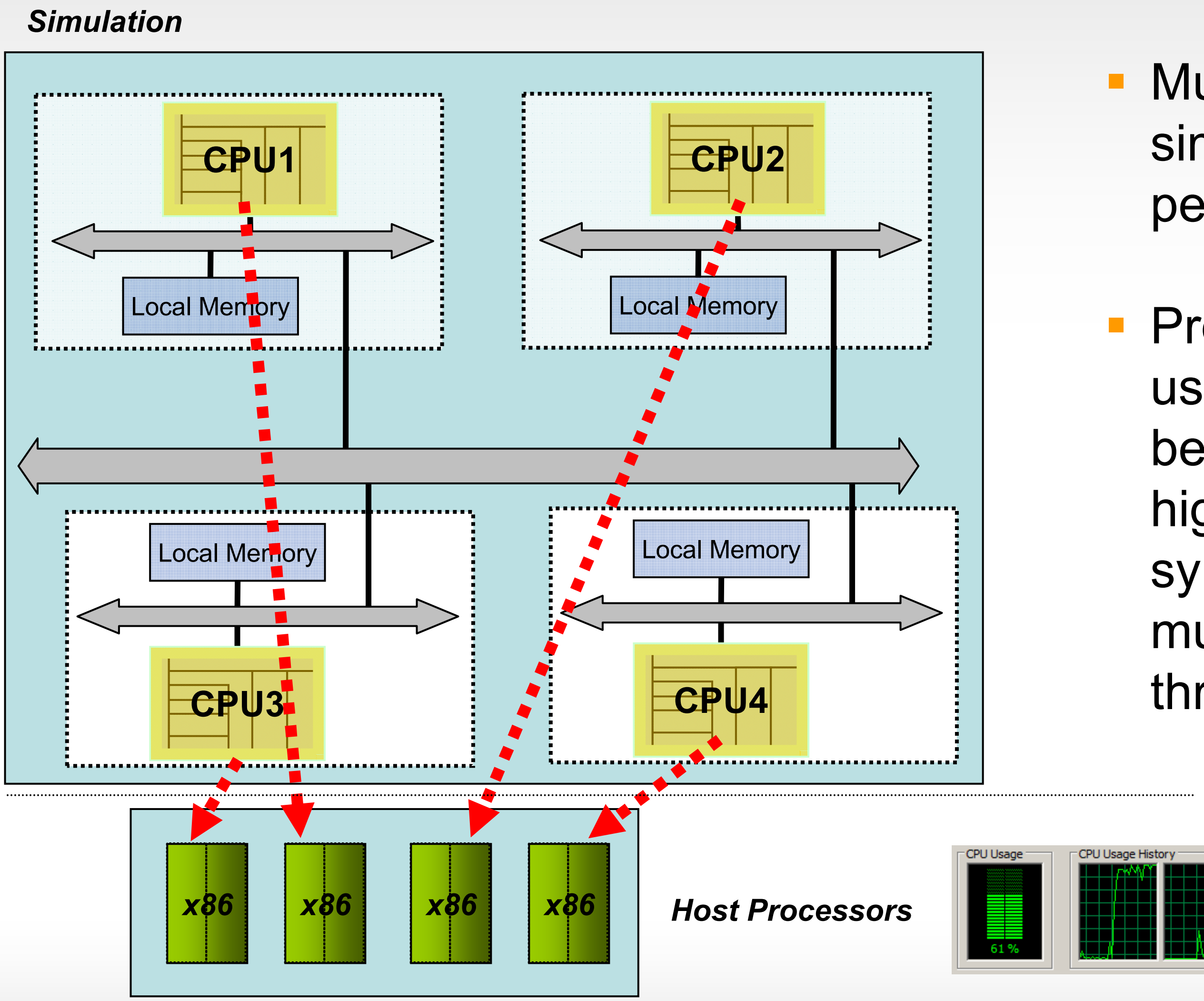


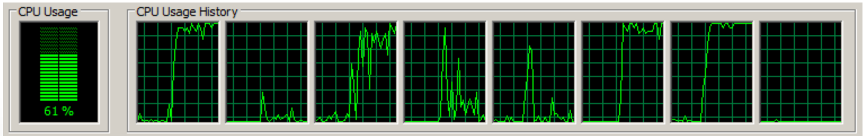
Virtual platform with ARM Cortex-A9, single thread simulation

© 2014 Imperas Software Ltd

2) Use the Multiple Cores Available in the Host Machine

MULTICORE DESIGN SIMPLIFIED
Imperas



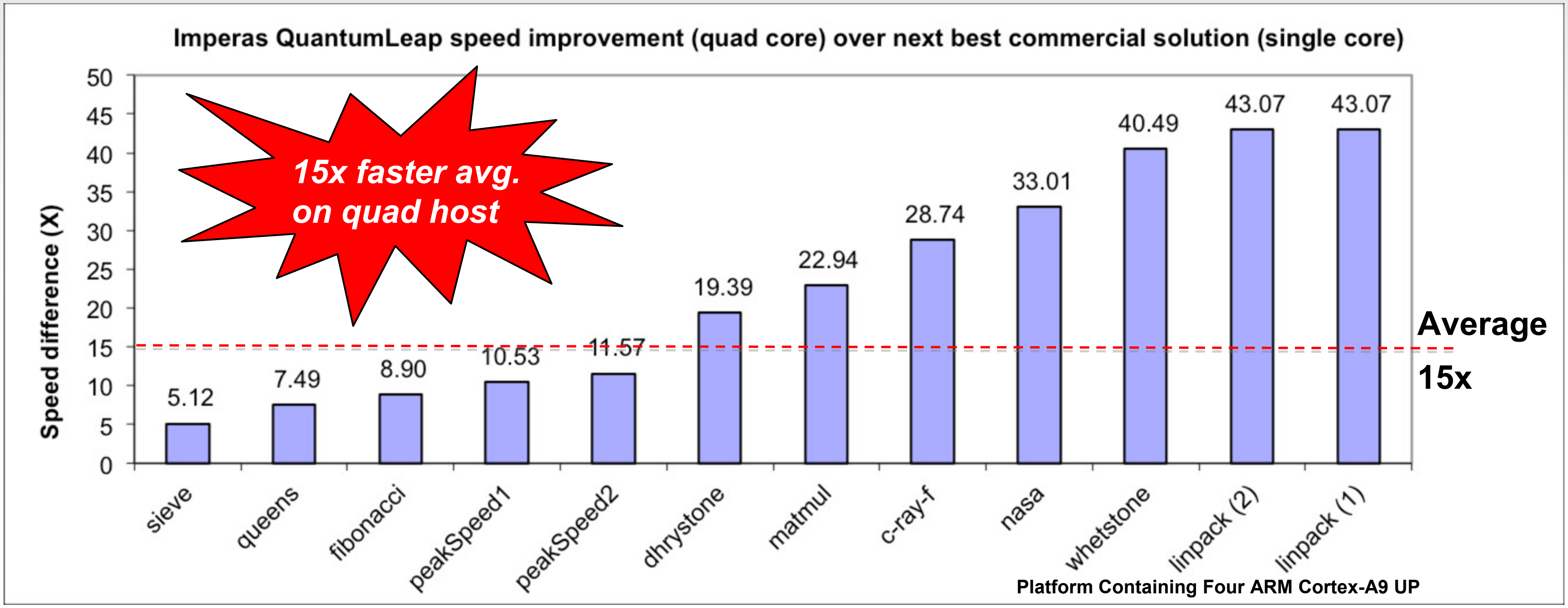


- Multiple cores for parallel simulation should result in performance gains
- Previous attempts at using multiple cores have been unsuccessful due to high overhead from synchronization of multiple simulation threads

© 2014 Imperas Software Ltd

QuantumLeap: 15x Faster Than Next Fastest Solution

MULTICORE DESIGN SIMPLIFIED



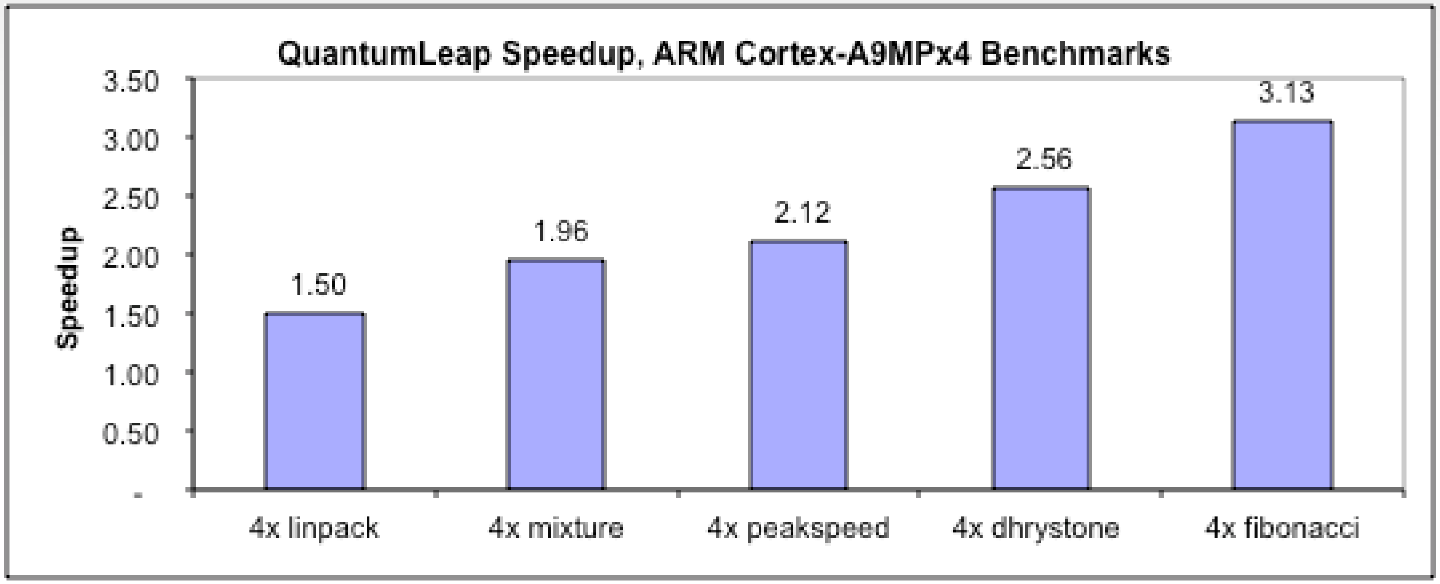
- Advanced parallel synchronization algorithm for SMP, AMP and hardware accelerators
- Transparent operation to user: No model, tool, software changes
- Total performance on benchmarks recorded up to 16K MIPS
- Accelerates execution 2-3x over current simulation performance (already 6x faster), 15x over nearest alternative solution

© 2014 Imperas Software Ltd

SMP Acceleration Results

MULTICORE DESIGN SIMPLIFIED
imperas

Simulated Applications Running Under Simulated Linux



All benchmarks run on ARM Cortex-A9MPx4 models

- QuantumLeap speeds up Imperas SMP models by 2.25x on average for quad core SMP and host
- Works for Imperas OVP Fast Processor Models of SMP cores even when used in a SystemC platform

© 2014 Imperas Software Ltd

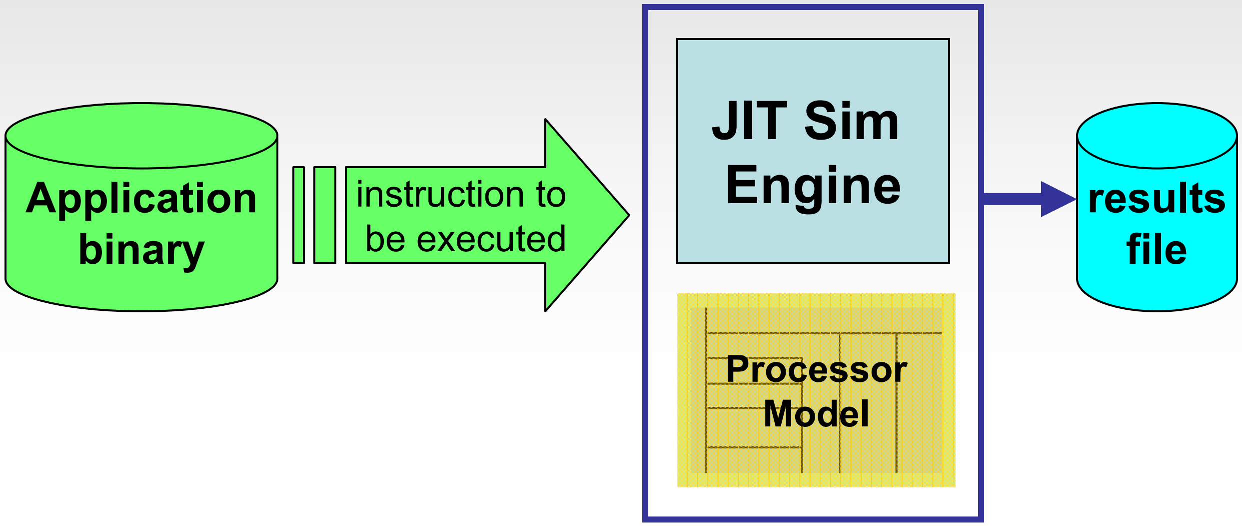
Agenda

MULTICORE DESIGN SIMPLIFIED
Imperas

- Simulation speed
- Simulation based tools for finding bugs
- Quality metrics
- Case study: OS porting, bring up and verification on Altera Cyclone V SoC FPGA
- Summary

© 2014 Imperas Software Ltd

Typical Software Simulator Execution

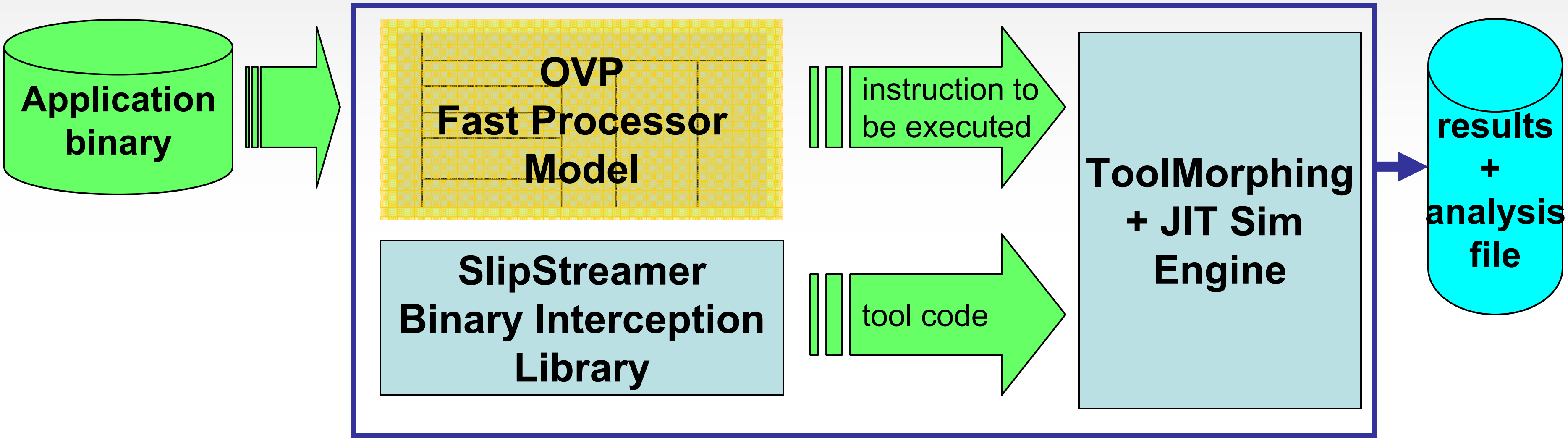


- Imperas technical conclusion at founding: typical software simulation flow is not adequate for software development, debug and test
- How to get full observability, controllability – *the promise of simulation* – from virtual platforms?
  - Need minimal overhead to maximize performance
  - Need to maintain order of instruction execution
  - Cannot introduce new “bugs” through the act of observation
- How to get near real time simulation performance?
- **Solution: innovation in both simulator engine and processor model**

© 2014 Imperas Software Ltd

Imperas Unique Model and Simulation Technology

MULTICORE DESIGN SIMPLIFIED
Imperas



- **Architect the simulation environment, from the beginning, for performance and tools; software tools should not be an afterthought**

- OVP Fast Processor Models contain special information for tools
- SlipStreamer libraries for tools
  - Non-intrusive: no modification of source code
  - Executes as native host code for minimal overhead
- ToolMorphing engine tightly integrates models and tools

© 2014 Imperas Software Ltd

ToolMorphing Technology Enables Tool Definition

MULTICORE DESIGN SIMPLIFIED
Imperas

VAP Tool (from Imperas) or User-Defined Tool:
Definition of the tool, written in C, included in simulation environment

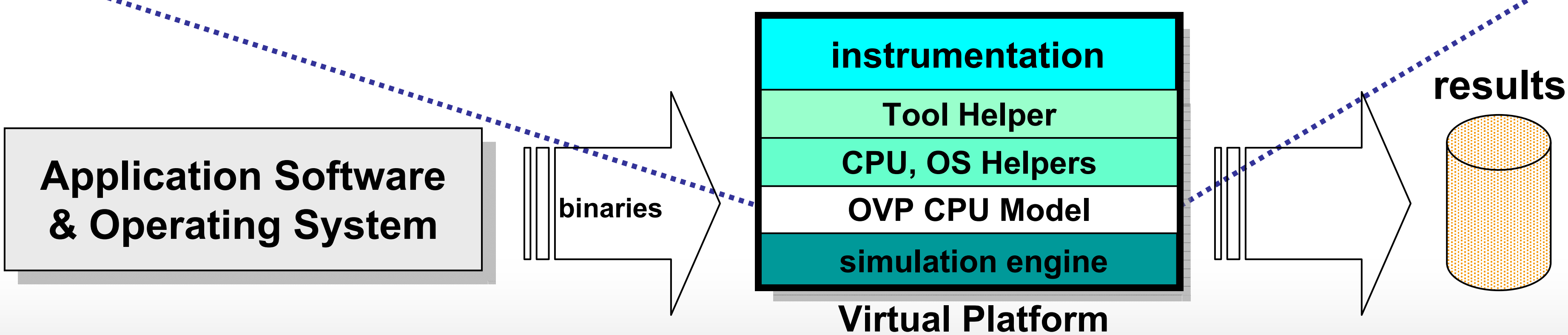
ToolMorphing Simulation Infrastructure

Tool Helper:
API enabling definition of software analysis tools

CPU and OS Helpers:
CPU and OS specific information

OVP Processor Model:
CPU functionality, predefined views, events, actions

Simulation Engine:
Just In Time (JIT) code morphing (binary translation)



© 2014 Imperas Software Ltd

Verification, Analysis & Profiling (VAP) Tool Suite for HDS Development

MULTICORE DESIGN SIMPLIFIED imperas

Operating System

Bare Metal Apps & Middleware

Platform (e.g. Drivers)

Processor

Trace coprocessor registers
Trace TLB trace exceptions
Trace modes
Trace service calls
Trace hypervisor calls
Trace secure monitor calls
Trace MT/MP extensions
Trace system calls
Trace timer
Trace cache instructions
Trace SIMD extensions
Trace instruction
Trace register change

Multi Processor Debug
Address space introspection
Virtual2physical mapping
Print CP registers
TLB dump
Break on exception
Break on mode
Break on register change
Break on instruction
Instruction coverage
Instruction profiling
Instruction fault Injection
Cache analysis

Bus connectivity view
Peripheral register view
Peripheral src debugger
Processor freeze control
Trace peripheral access
Memory coverage
Shared memory checks

Break on line
Break on function call
Elf introspection
Unlimited HW breakpoints
Memory region watchpoints
Trace source line
Trace context
Trace functions
Line Coverage
Function profiling
Heap checks
Stack checks
Malloc checks
Semaphore checks

Trace console
Trace execve
Trace scheduler
Trace tasks
Trace module loads
Trace printk

Simulator Break on messages TCL callbacks Full GDB command set

- Drivers
- Firmware
- Assembly libraries
- OS porting and bring up
- Hypervisors

- Multiprocessor, multicore, multithread, multi-everything
- Non-intrusive
- Low overhead
- User extendable

© 2014 Imperas Software Ltd

MULTICORE DESIGN SIMPLIFIED
Imperas

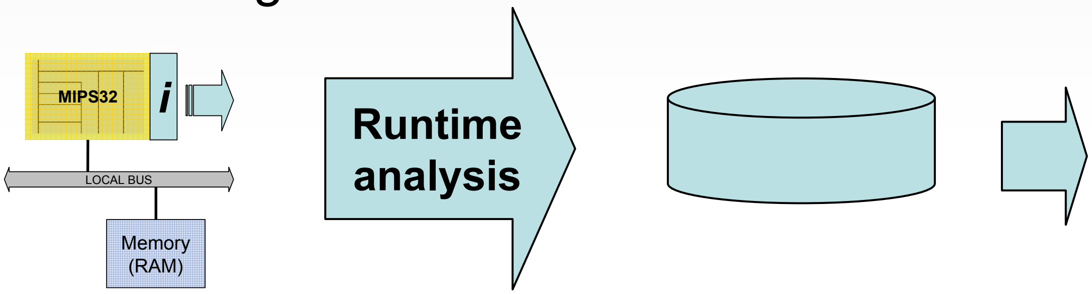
Agenda

- Simulation speed
- Simulation based tools for finding bugs
- Quality metrics
- Case study: OS porting, bring up and verification on Altera Cyclone V SoC FPGA
- Summary

© 2014 Imperas Software Ltd

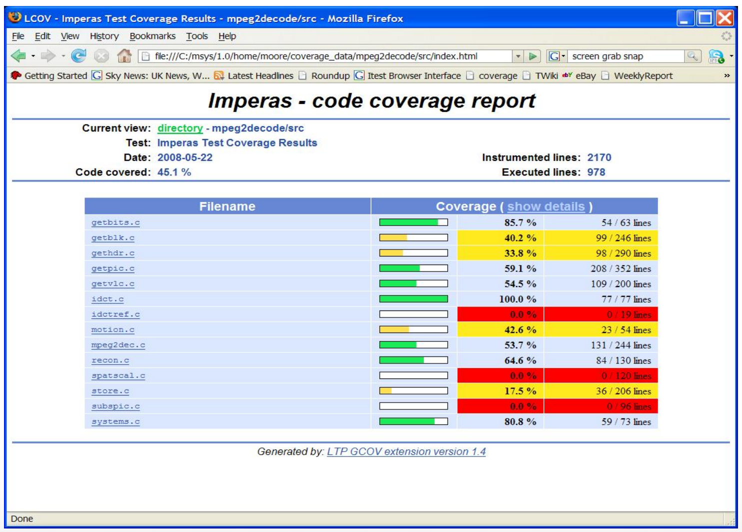
Code Coverage

- Non-intrusive statement and branch coverage analysis using intercepts
- Full multiprocessor, multicore, and peripheral code coverage



Standard .lcov file format

TN:cpuA.lcov
SP:cpuA
SF:/home/graham/mpeg2decode/src/getbits.c
DA:44,3
DA:45,3
DA:46,3
DA:47,3
DA:58,3
DA:59,3
DA:60,3
DA:63,1151
DA:66,1151
DA:67,1151
...



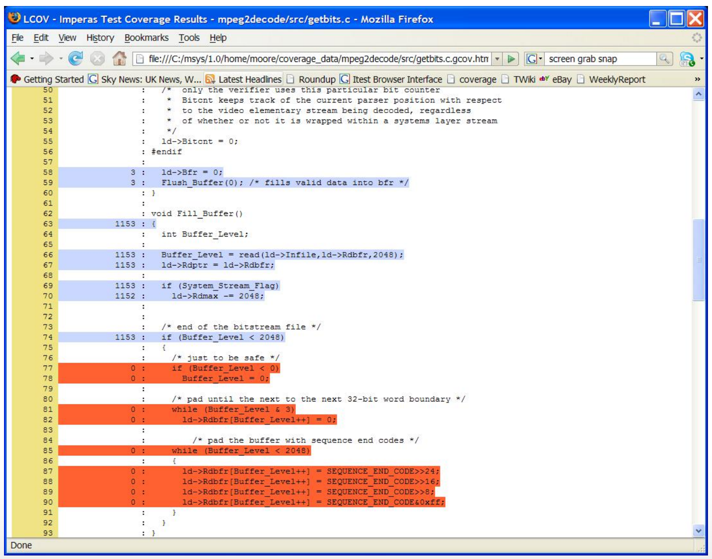
LCOV - Imperas Test Coverage Results - mpeg2decode/src - Mozilla Firefox

Imperas - code coverage report

Current view: directory - mpeg2decode/src
Test: Imperas Test Coverage Results
Date: 2008-05-22 Instrumented lines: 2170
Code covered: 45.1 % Executed lines: 978

Filename	Coverage (show details)	
getbits.c	85.7 %	54 / 63 lines
getblk.c	40.2 %	99 / 246 lines
gethdr.c	33.8 %	98 / 290 lines
getpic.c	59.1 %	208 / 352 lines
getvlc.c	54.5 %	109 / 200 lines
idct.c	100.0 %	77 / 77 lines
idctref.c	0.0 %	0 / 19 lines
motion.c	42.6 %	23 / 54 lines
mpeg2dec.c	53.7 %	131 / 244 lines
recon.c	64.6 %	84 / 130 lines
spatscal.c	0.0 %	0 / 120 lines
store.c	17.5 %	36 / 206 lines
subspic.c	0.0 %	0 / 96 lines
systems.c	80.8 %	59 / 73 lines

Generated by: LTP GCOV extension version 1.4



© 2014 Imperas Software Ltd

■ MULTICORE DESIGN SIMPLIFIED
Imperas

Agenda

- Simulation speed
- Simulation based tools for finding bugs
- Quality metrics
- Case study: OS porting, bring up and verification on Altera Cyclone V SoC FPGA
- Summary

© 2014 Imperas Software Ltd

OS Porting, Bring Up and Verification on Altera Cyclone V SoC FPGA

MULTICORE DESIGN SIMPLIFIED
imperas

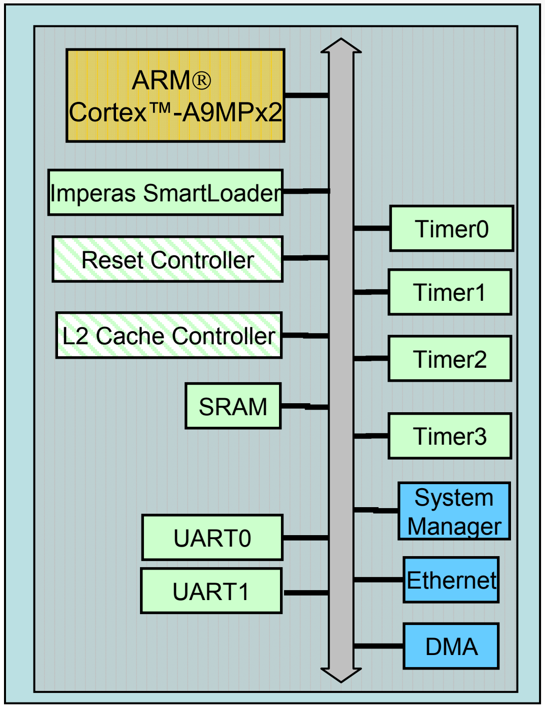
1) Linux boot on single core ARM Cortex-A9

2) SMP Linux boot on dual core ARM Cortex-A9

3) RTOS boot on single core ARM Cortex-A9

4) AMP boot on dual core ARM Cortex-A9

Altera Cyclone V SoC FPGA



© 2014 Imperas Software Ltd

Cyclone V SoC FPGA Virtual Platform

MULTICORE DESIGN SIMPLIFIED
Imperas

- Top level virtual platform built using Open Virtual Platforms (OVP, www.OVPworld.org) ICM API
- ARM Cortex-A9MPx2 and Altera Nios II processor core models from the OVP Library
- Peripheral models
  - Some models available in the OVP Library
  - Remaining models of peripheral components developed using OVP APIs
- OVP APIs written for C language
- Simulation engine: Imperas M*SDK

- All OVP processor and peripheral models include both native OVP and native SystemC/TLM2 interfaces, so all the following results could have been achieved using the OSCI SystemC simulator plus Imperas M*SDK product
  - Peripheral models could have been written in SystemC
  - M*SDK tools require OVP processor core models for ToolMorphing capability

© 2014 Imperas Software Ltd

1a) Linux Boot on Single Core ARM Cortex-A9

MULTICORE DESIGN SIMPLIFIED Imperas

- Use Linux from Altera: Altera-3.4
- Use default configurations
- Use default device trees
  - Comment out a few peripherals not yet modeled
- Bug found in Linux kernel preemptive scheduling
  - Running multiple applications under Linux part of standard Imperas bring up testing
  - Linux boots and runs, but does not switch tasks properly
  - Not observed in previous virtual platform (different virtual platform vendor) using much slower model of ARM Cortex-A9MPx2
    - Could not run multiple applications for long enough simulation to observe the bug

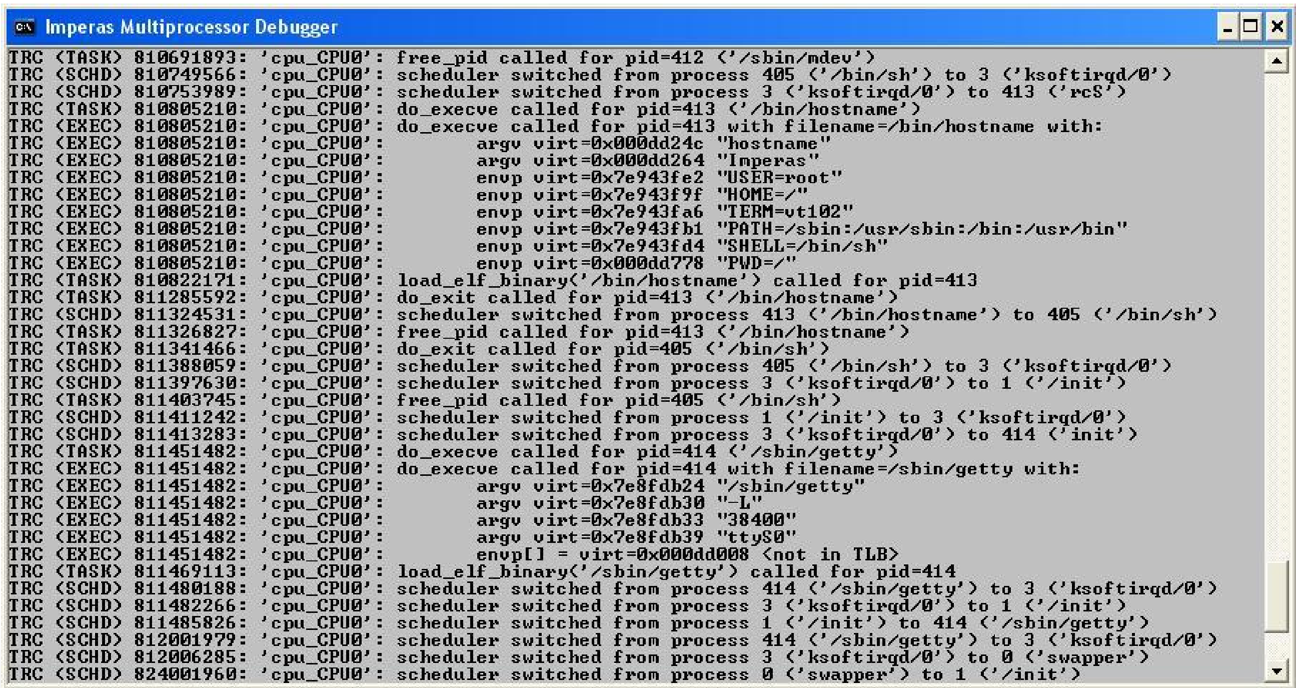
- Approximately 2 man weeks effort to build virtual platform able to boot Linux
- Virtual platform boots Linux in under 5 sec on standard PC, Windows or Linux

© 2014 Imperas Software Ltd

1b) OS-Aware Tools Used to Find the Bug

MULTICORE DESIGN SIMPLIFIED
Imperas

- 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 to boot Linux, however, only ~700 tasks
- OS-aware tools debug in hours, once the bug was observed
- Simulation overhead due to OS-aware tools < 10%



© 2014 Imperas Software Ltd

2) SMP Linux Boot on Dual Core ARM Cortex-A9

MULTICORE DESIGN SIMPLIFIED
imperas

- Use Linux from Altera: Altera-3.6
- Use default configurations
- Use default device trees
  - Comment out a few peripherals not yet modeled
- No problems in SMP Linux bring up on virtual platform

© 2014 Imperas Software Ltd

3a) Micrium µCOS-II Boot on Single Core ARM Cortex-A9

MULTICORE DESIGN SIMPLIFIED
Imperas

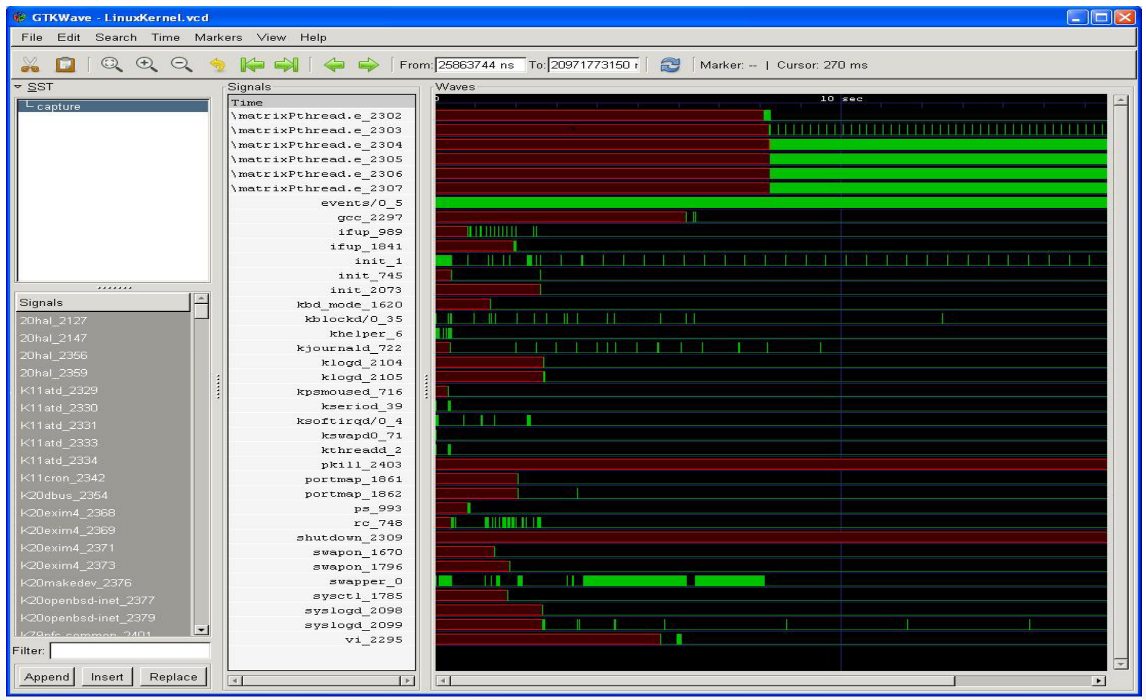
- Use Altera µCOS-II release
- Bugs found and fixed in GIC register accesses using OS-aware tools
  - Access ICDICER1 to 8 when only 0 to 7 exist
  - Access ICDIPTR08 to 63 when only 00 to 55 exist
- Typically < 1 week effort to add support for new RTOS
- RTOS OS-aware tools include event scheduler viewing as waveform

© 2014 Imperas Software Ltd

3b) OS Porting and Bring Up

MULTICORE DESIGN SIMPLIFIED
Imperas

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



© 2014 Imperas Software Ltd

4a) AMP boot on Dual Core ARM Cortex-A9

MULTICORE DESIGN SIMPLIFIED
Imperas

- Linux booting on first core, µC/OS-II on second core
- Bug found in Linux accesses of GIC registers
- Virtual platform debug took 2 days versus 2 weeks on hardware platform (5x improvement)
- Also need to ensure that different operating systems do not access forbidden memory segments
  - Bugs found using custom memory access monitor

© 2014 Imperas Software Ltd

4b) Custom Memory Access Monitor Accelerates AMP Platform Debug

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

© 2014 Imperas Software Ltd

Summary

MULTICORE DESIGN SIMPLIFIED Imperas

- More processor cores, more complex systems ⇒ more tests are needed
- Simulation speed is critical for running more tests
- Also need tools and metrics, architected into the simulation environment from the start
- Results were shown for SMP and AMP systems on Altera Cyclone V SoC FPGA

Software quality is proportional to simulation speed!

© 2014 Imperas Software Ltd