Exploring How the Internet of Things is Changing the Verification Challenge

Nick Heaton
Distinguished Engineer
Verification Futures April 2017
Agenda

• What do we mean by IoT?
• Verification Trends
• New challenges we are facing
• What are Cadence doing to address these challenges?
  – Simulation
  – FPGA Prototyping
  – SoC Use-case Creation
What do we mean by IoT?
What do we mean by IoT?
Verification Trends
Verification Trends
Leading edge design size

More and more emphasis on SoC Integration Verification
Introduction of multiple coherent processor clusters significantly increased SoC integration verification complexity.

Growing virtualization trend is further adding to the SoC integration challenges.
**Verification Trends**

**Simulation methodology**


- **Directed Testing**
- **Constrained Random (UVM)**
- **Use-case Driven**

**Evolution of eRM, OVM into UVM for Simulation based Metric-Driven Verification**

**Accellera Portable Stimulus Standard** reflects emergence of technology which is enabling portability across execution platforms.
New Challenges
New Challenges We Are Facing

- **Sensors**
  - Security
  - Mixed-Signal scalability, large A and D
  - Ultra-low Power with complex power management

- **Connectivity**
  - Security
  - Complex Protocol Support for multitude of standards
  - Performance Verification of Networking, Storage and Custom Accelerators
  - Latency Critical for business success
  - Digital Simulation Performance
  - HW/SW use cases demand long runs

- **Processes**
  - Security
  - Back-office Performance

---

**Verification Challenges**

- Security – across all levels from IP, through Infrastructure, SoC and System
- Functional Safety
- Simulation Scalability – Always more needed
- HW/SW Use-cases
  - Better core engine support
  - Productivity improvements in Use-case creation
- Full support for Performance Analysis across Simulation and Emulation
- Low Power Support – across simulation, emulation, debug

How is Cadence Addressing These Challenges?
Three Generations of Simulation
Ushering in a new era of parallel simulation

Interpreted Simulators
Ex. Verilog-XL

Compiled Simulators
Ex. Incisive

Performance and scalability via multi-core

Use Cases

Introducing Xcelium
Highest performance 3rd generation parallel simulator

REVOLUTIONARY

PROVEN

Incisive® Enterprise Simulator

OPTIMIZED

- Multi core engine improvement
- 2x average single core speed-up
- Direct kernel engine integration
- New randomization engine
- Agile release process for quality

3X+ RTL  5X+ Gate  10X+ DFT
Xcelium Parallel Simulation Architecture

- Supports all Incisive use cases
  - Xcelium developed for ease of adoption, migration

- Behavioral engine runs Single-Core
  - Average 2X faster over Incisive refactored engines
  - Runs testbench, low power, mixed signal, VHDL

- Multi-Core engine with direct kernel integration
  - Runs gate-level zero delay, RTL, X-prop, SVA, …

- Essential signal debug maximizes speed
  - Non-essential signals interpolated on demand
Xcelium Multi-Core Enables More Efficient SoC Verification

- Single-core simulation slows with more events
- To compensate, verification teams narrowed tests
- But SoC functions operate concurrently
- Multi-core simulation is more effective with higher event density

- Multi-core simulation enables test methodology better suited for SoC
- Perspec™ System Verifier can be used to create concurrent scenarios
- Creates test alignment between Xcelium™ multi-core and Palladium® Z1 acceleration

<table>
<thead>
<tr>
<th>Simulation Function</th>
<th>Event Density</th>
<th>SoC HW Run Time</th>
<th>Single-Core</th>
<th>Multi-Core</th>
<th>Multi-Core Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boot Sequence</td>
<td>1.0</td>
<td>28 ms</td>
<td>16.8 hr</td>
<td>5.7 hr</td>
<td>2.9X</td>
</tr>
<tr>
<td>Test Scenarios</td>
<td>4.3</td>
<td>15 ms</td>
<td>37.3 hr</td>
<td>6.8 hr</td>
<td>5.5X</td>
</tr>
<tr>
<td>Overall</td>
<td>2.1</td>
<td>43 ms</td>
<td>54.1 hr</td>
<td>12.5 hr</td>
<td>4.3X</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Concurrent Test Scenarios</th>
<th>Single-Core</th>
<th>Multi-Core</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.6 hr</td>
<td>0.2 hr</td>
</tr>
<tr>
<td>2</td>
<td>1.0 hr</td>
<td>0.3 hr</td>
</tr>
<tr>
<td>3</td>
<td>1.4 hr</td>
<td>0.4 hr</td>
</tr>
<tr>
<td>4</td>
<td>1.7 hr</td>
<td>0.5 hr</td>
</tr>
<tr>
<td>5</td>
<td>2.1 hr</td>
<td>0.6 hr</td>
</tr>
</tbody>
</table>
What is FPGA-Based Prototyping?

- **Primary platform for pre-silicon software development and validation**
  - Maps a digital ASIC, ASSP, SoC design or part thereof into one or more FPGAs

- **Allows SW to simulate in real world environments**
  - Provides pre-silicon execution speeds in MHz
  - Enables connectivity to real peripherals
    - Runs real world traffic flows including interrupts and unpredictable events
    - Runs error conditions and handling errata with other system components
FPGA-based Prototyping Is Hard To Do

Interfaces

Clocking

Debug

Memories

Software
Really, Really Hard To Do

FPGA-based prototyping has become the methodology of choice for early software development.

BUT …

Prototyping implementation and bring-up takes too long and there has, so far, not been any easy transition from simulation and emulation into FPGA-based prototyping.
Or is it?
Protium S1 – Addressing the prototyping challenges

- No RTL modifications needed
  - Clocking / number of clocks
  - Automated memory compilation and modeling

- Fully automatic, multi-FPGA partitioning
  - Optional manual optimization

- Pre-FPGA P&R model validation
  - Multiple design integrations per day
  - Avoids time-consuming FPGA P&R

- Fully integrated FPGA P&R
  - Automatic constraint generation
  - Guaranteed P&R success
Protium S1 Requires No ASIC RTL Modifications

• No ASIC RTL changes
  • Automatic conversion of latches and tri-states
  • Automatic memory compilation and modeling
  • Fully automated clock tree transformation
    • Automatic conversion of gated and multiplexed clocks
Comprehensive Memory Support

• FPGA built in & XSRAM
  – Benefits:
    – Automatic mapping of any memory type
    – Support for multi-port memories
    – Support for backdoor upload/download
  – XSRAM adds:
    – Increases FPGA internal memory from 80Mbits to 128MBytes

• XDRAM
  – Benefits:
    – Adds DDRx bulk memories
    – Supports LPDDR2/3/4; DDR3/4; HBM
    – No change to design memory controller and firmware
    – Support for backdoor upload/download
    – Acts as memory speed bridge (timing, refresh, etc.)

• Directly Connected or Full Custom
  – Daughter cards available for more custom approaches if required
SoC Verification Needs to Address:

- **Diverse Platforms**
  - Virtual Platform
  - Simulation
  - Emulation
  - FPGA Prototype
  - Silicon Board

- **Diverse Users**
  - Architect
  - HW Developer
  - SW Developer
  - Verification Engineer
  - SW Test Engineer
  - Post-Silicon Validation Engineer

- **Diverse Scopes (Integration)**
  - Middleware (Graphics, Audio, etc.)
  - OS & Drivers
  - Bare Metal SW
  - System on Chip (HW + SW)
  - Sub-System
  - IP

**Use Case Reuse**

**Vertical Reuse**

**Horizontal Reuse**

**Application-Specific Components**

- CPU subsystem
  - CPU, CPU L2 cache
  - Cache coherent fabric

- SoC interconnect fabric
  - DDR3
  - USB3.0
  - PCIe Gen 2-3
  - Ether net
  - HDMI
  - SATA
  - MIPI

- 3D GFX
- DSP A/V
- Boot processor

- Modem
  - GPIO
  - Display
  - UART
  - INT C
  - PCIe
  - SPI

- Other peripherals
  - PHY
  - 3.0 PHY
  - 2.0 PHY
  - MIPI

- Low-speed peripherals
  - PMU
  - JTAG
  - INT C
  - SPI

- High-speed, wired interface peripherals
  - CPU
  - L2 cache
  - CPU
  - L2 cache

- **Use Case Reuse**

- **Horizontal Reuse**
The Solution: Perspec™ System Verifier

Use Case Reuse

Diverse Scopes (Integration)
- Middleware (Graphics, Audio, etc.)
- OS & Drivers
- Bare Metal SW
- System on Chip (HW + SW)
- Sub-System
- IP

Vertical Reuse

Diverse Users
- Architect
- HW Developer
- SW Developer
- Verification Engineer
- SW Test Engineer
- Post-Silicon Validation Engineer

Abstract Model
- Reusable Use Cases
- Mapping to Targets

Perspec™ System Verifier

Delivers 10x Productivity Gain

Generated code
- Multi-Core Verification OS

Library provides built-in content (e.g., coherency stressing)

Tests capture user intent and use cases

Diverse Platforms
- Virtual Platform
- Simulation
- Emulation
- FPGA Prototype
- Silicon Board

Horizontal Reuse

All measurements as compared to hand-generated testcases on previous projects
From Partial Use-Case to Concrete Use-Case with UML GUI

Using GUI, users can create sophisticated scenarios including timing, repetition.

Scenario specification (goals)
- mem_buff_t
  - data.kind VIDEO

in_mem in_mem

Solver: check feasibility and randomize Data and Control Flow

Distribute available computing resources and sync as needed

Randomize a way to get a video buffer

Scenario instance (solution)

Select random attributes:
- video format convertible to mpg4
- accessible memory location

Randomize other video stream attributes

Randomize a display that can show mpeg4

These 4 were in the scenario specification
Productivity from **Built-In Content** with Perspec Libraries

- **Requirements/opportunities:**
  - Much of the SoC’s logic is modeled the same way
    - It is possible to model the generic aspects of an SoC
  - Consistent coding style and methodology can improve readability and reuse
  - Can build libraries for cache, distributed virtual memory and low power logic

- **Cadence libraries**

  - System Methodology Library (SML)
    - Captures system modeling including memories, processors, and more
    - User provides Excel configuration tables

  - PSlib for ARM Architecture
    - Provides built-in flexible content for ARM-related challenges
    - E.g., cache, DVM, and low-power scenarios
### Perspec System Verifier

**Satisfies Portable Stimulus requirements today and will meet the standard**

<table>
<thead>
<tr>
<th><strong>Productivity</strong></th>
<th>• 10X improvement for complex SoC test creation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Abstraction</strong></td>
<td>• UML-style use-case diagrams</td>
</tr>
<tr>
<td><strong>Content</strong></td>
<td>• Out-of-the-box content</td>
</tr>
<tr>
<td><strong>Portability</strong></td>
<td>• Reuse across all execution platforms</td>
</tr>
<tr>
<td><strong>Measurement</strong></td>
<td>• SoC-level hardware/software coverage metrics</td>
</tr>
</tbody>
</table>

All measurements as compared to hand-generated testcases on previous projects.