Validation Strategies with pre-silicon platforms

Shantanu Ganguly

Synopsys Inc

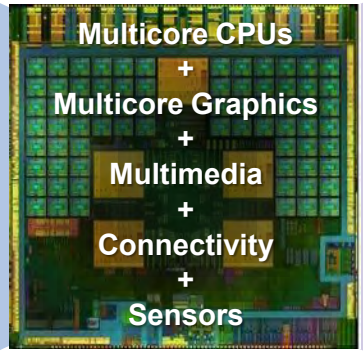
April 10 2014

Agenda

- Market Trends
- Emulation HW Considerations
- Emulation Scenarios
- Debug

Mobile & Internet-of-Things Driving Growth

Convergence → SoC complexity → New verification challenges



Verification Complexity

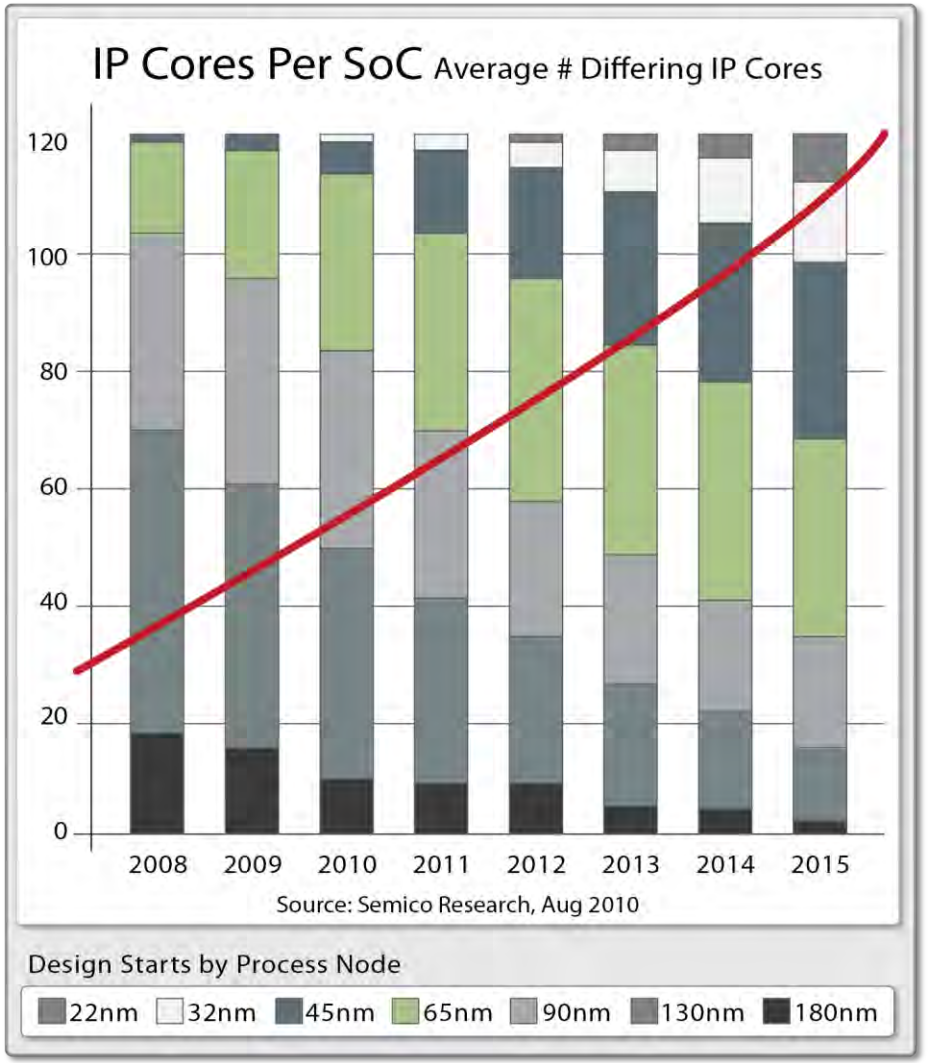
Power Efficiency

More Software

Time-to-Market

SoC Multi Core Architecture Trends

- Massive feature integration
  - Driven largely by Moore's Law (supply) and convergence (demand)
- Distributed architectures
  - Higher scalability (and independence)
  - Sharing memory
- Multiple processors
  - (Multicore) CPU
  - DSP
  - Special purpose (MPEG, GFX, …)
  - Always on controller
- Distributed DMA
  - Removes centralized DMA bottleneck
- Increasing software complexity
  - Re-use with multiple platform SoCs
  - Broader end use market coverage per SoC with software programmability



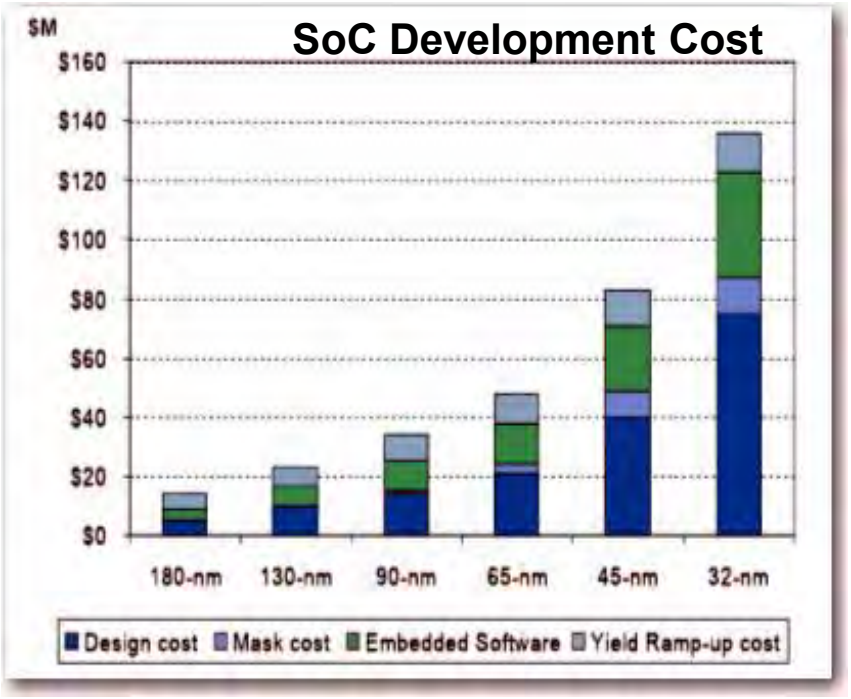
From NIT Alumni Meet Keynote Speech: Jim Hogan

SoC Design Complexity & Cost – Out of Control

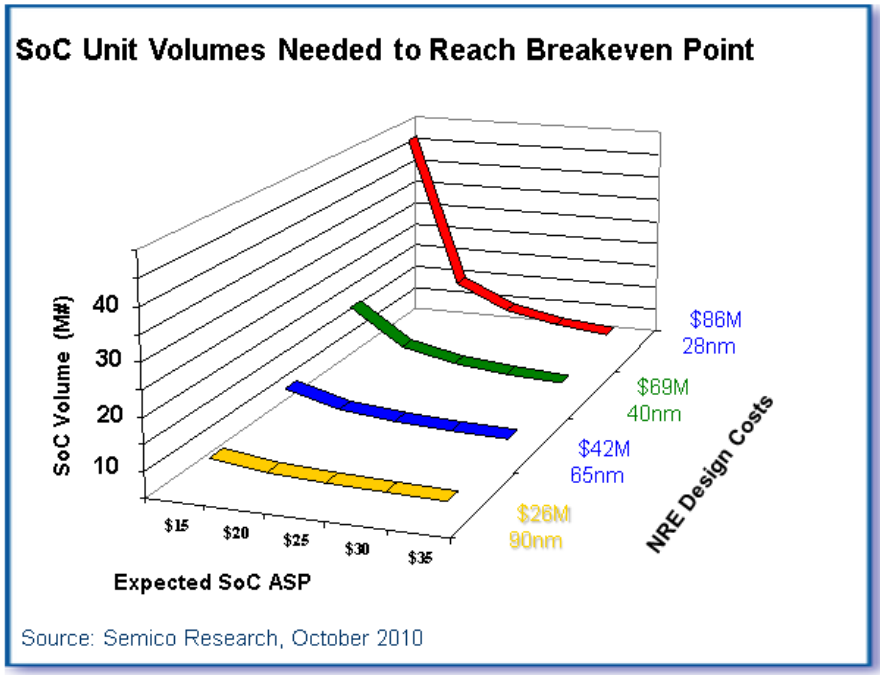
- Increasing complexity means increased risk
  - At 32nm, a typical design has ~50% chance to meet all objectives
  - At 22nm, that number drops to ~30%

Source: I.B.S. Inc.

- "Designer productivity must improve to match chip complexity"
  - The later a problem is detected, the more impact it will have on design schedules

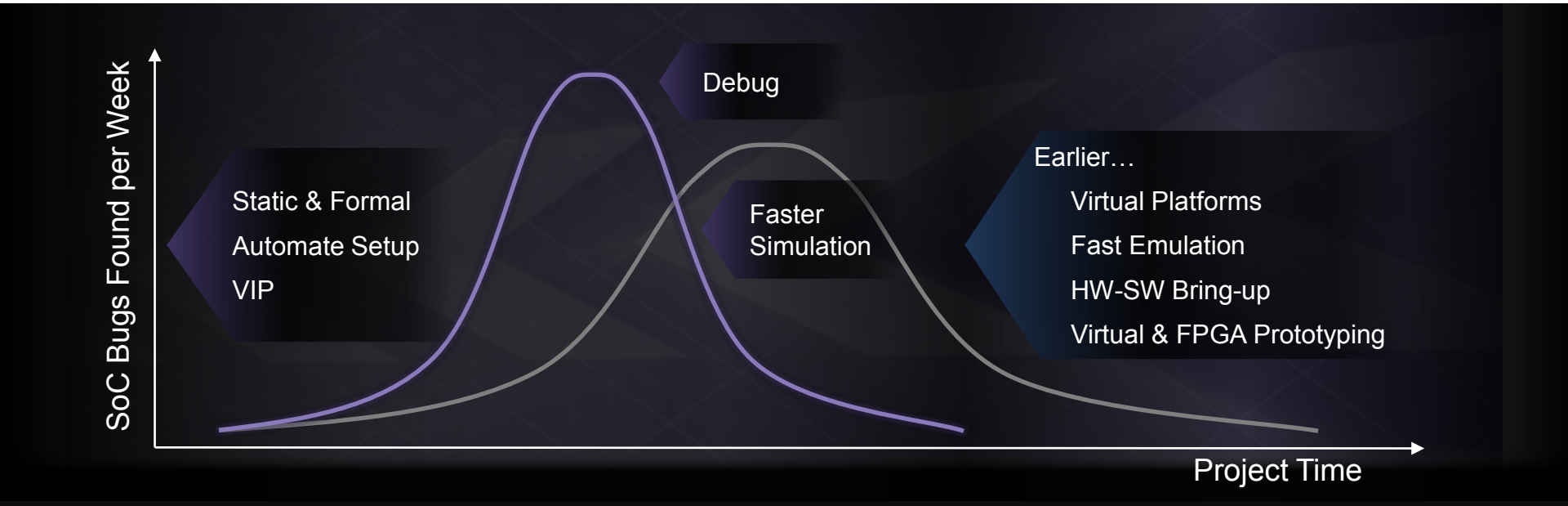


Source: Gartner



Need ‘Shift-Left’ for Faster Time-to-Market

Earlier HW verification, earlier SW bring-up



Smart Verification Strategy
- Static and Formal

Intelligent Verification Methodology
- Integrated, automated flows

Earlier HW-SW Bring-up
- Faster emulation

Verification Continuum
- Seamless flow

Agenda

- Market Trends
- Emulation HW Considerations
- Emulation Scenarios
- Debug

Three Ways to Emulate 256 Million Gates

- Emulation chip capacity accounts for the wide range of sizes.

Commercial FPGAs

SYNOPSYS®

1 meter

Custom Emulation Chip Advantages

	Custom FPGA	Custom Processor	Commercial FPGA
Compile Time	Faster due to emulation-specific interconnect architecture.	Faster due to processor-type architecture.	Slower due to chip place & route. Easily run on parallel on a small server farm.
Debug	Built-in: Change probes without recompile.	Built-in: Change probes without recompile.	Mix of built-in (readback) and FPGA resource. Some probe changes require recompile.
Emulation Compiler	Single-vendor: Fully integrated, all vendor-developed at vendor cost. Small user base.	Single-vendor: Fully integrated, all vendor-developed at vendor cost. Small user base.	Chip-level half by FPGA vendor's large team at no cost. Large user base good for quality.

Commercial FPGA based solution superior for Emulation

- Highest capacity per chip
  - ZeBu Server 3 module emulates 60M gates in 9 emulation chips.
    - Palladium XP needs 54 chips, Veloce 2 needs 75 chips.
  - Components fit better, fewer design nets get cut.
- Means less interconnect HW
  - Highest performance: 2 to 5 MHz
  - Low power, small size, reliable
- Latest process every two years
  - From Xilinx, no development cost

== Lowest TCO

  - Fastest, coolest, smallest, cheapest, most reliable logic emulation.

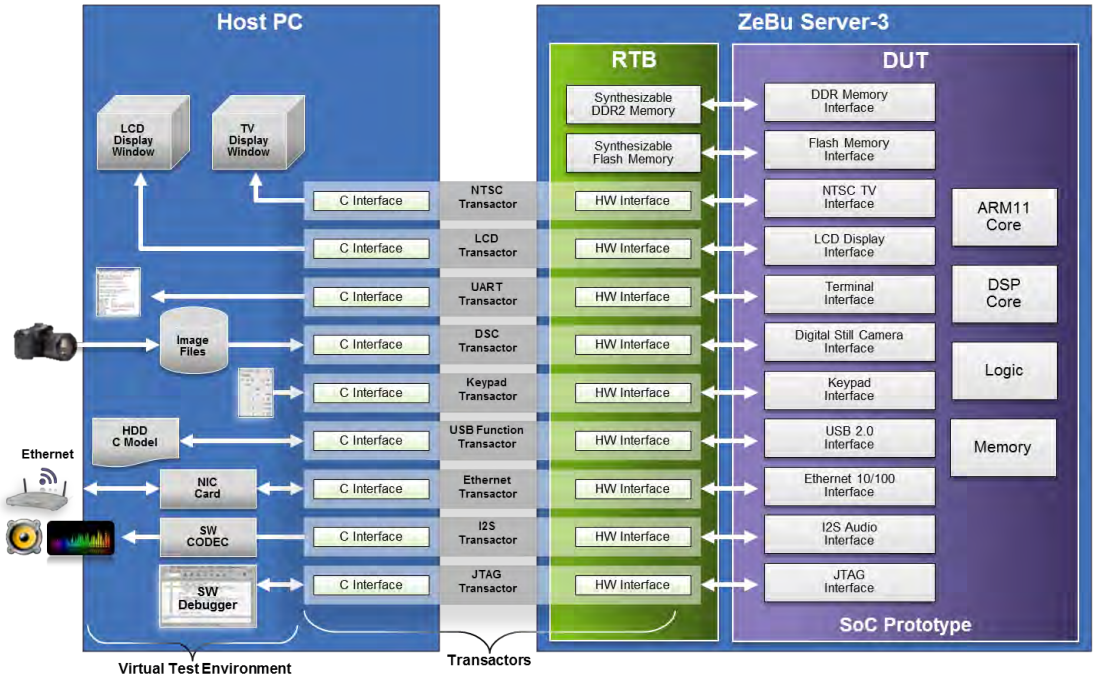
ZeBu Server-3 Hardware Performance Advantage

- 6-8X larger capacity emulation chips
  - fewer nets must cross chip-to-chip,
  - more nets stay on-chip where they are fast
- High bandwidth communication between emulation chips, modules, units, host
  - Each chip has 600 Gbps bandwidth to other chips
  - 33 Gbps bandwidth between modules
  - 640 Gbps bandwidth between units
  - 4 Gbps host communication bandwidth
- FPGA architecture advantages
  - Specialized HW for arithmetic operations in FPGA.
  - Wire-to-wire and gate-to-gate mapping, no modeling abstraction

Application	Design Size	Performance
GPU Quad Cluster	60 MG	5.0 MHz
GPU Dual Cluster	40 MG	4.95 MHz
GPU Single Cluster	33 MG	4.75 MHz
Customized GPU	50 MG	3.75 MHz
Communication Processor	60 MG	2.9 MHz
Consumer SoC	100 MG	2.8 MHz
Broadband SOC	80 MG	2.5 MHz

ZeBu Server-3 Throughput Advantage

- Highest raw performance hardware
- Multi-threaded runtime
- Truly concurrent communication message port
  - No blocking message transfer
- Dedicated high speed HW resources for implementation of transactors and communication ports.
- Each transactor can be modeled as separate process for maximum parallelism.



Agenda

- Market Trends
- Emulation HW Considerations
- Emulation Scenarios
- Debug

Verification: Architecture to Silicon

Accuracy

Architecture Exploration

Does the architecture meet performance and power requirements?

Processor & GPU selection
Memory System Dimensioning
Interconnect Configuration
Cache Coherency
Global Interrupts
Power Management

Design Technology:

Traffic Models
Transaction Level Models
Performance Visualisation
Hybrid-Simulation

HW / SW Integration

Prototype SW before first silicon

OS Integration
Driver Development
Virtualisation
Performance Optimisation

Tools:

Debugger
Transaction Level Models
systemC Virtual Prototypes
FPGA Prototypes
Performance Visualisation

Functional Verification

Does the design function correctly &meet performance and power requirements?

Protocol Compliance
Interconnect BW & Latency
Cache Coherency
Data integrity
Power & Clock Domains

Design Technology:

Traffic Models
Verification IP
System Monitor
Performance Visualisation

System Validation

Validate that the design functions as specified.

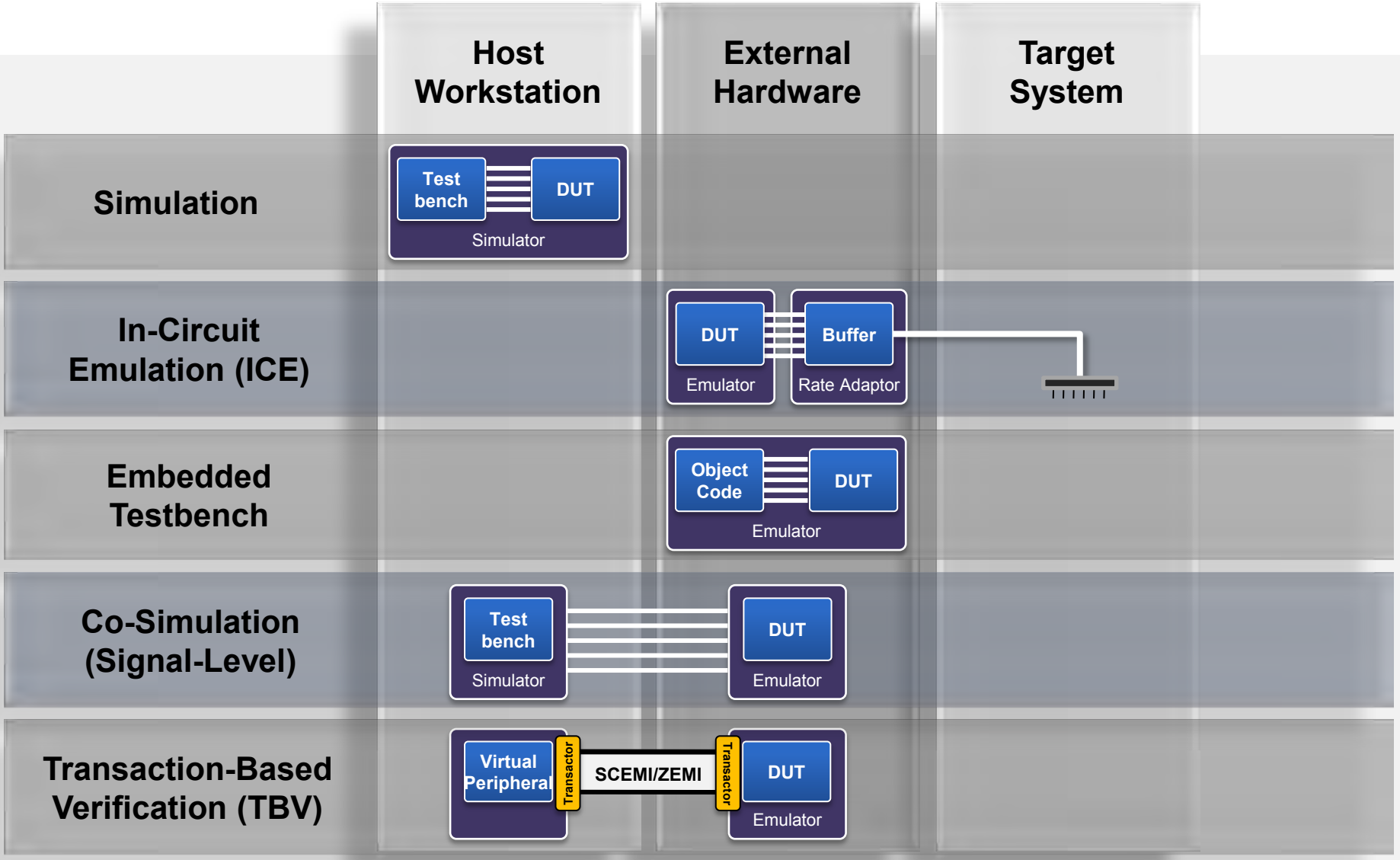
Graphics
Video
Audio
Browser Application
Gesture recognition
Phone functions
Camera functions

Design Technology:

FPGA Prototypes
HW Accelerated Simulation
Hybrid-Simulation
Verification IP
Transaction Level Models
IO Traffic

Turn Around Time

Common HW-Assisted Verification Modes



Summary of Verification Modes

	External Hardware	Challenges
Simulation	• Everyone has one • Perfect for 90% of designs	• Never fast enough
In-Circuit Emulation (ICE)	• Highest performance • Physical connections	• Rate adapter availability
Embedded Testbench	• Very fast • No physical connections	• Needs synthesizable TB • Limited to Software only
Co-Simulation (Signal-Level)	• Simple to use • Leverages existing TB • Good for DUT bring-up	• Little performance gain
Transaction-Based Verification (TBV)	• Highest performance • Works with virtual devices • No rate adapters needed	• Availability of transactors

One Emulator, Many Applications

Advanced verification use modes with ZeBu Server-3

Transaction-based Verification
system-level SoC verification

Hybrid Emulation
architecture optimization & early software development

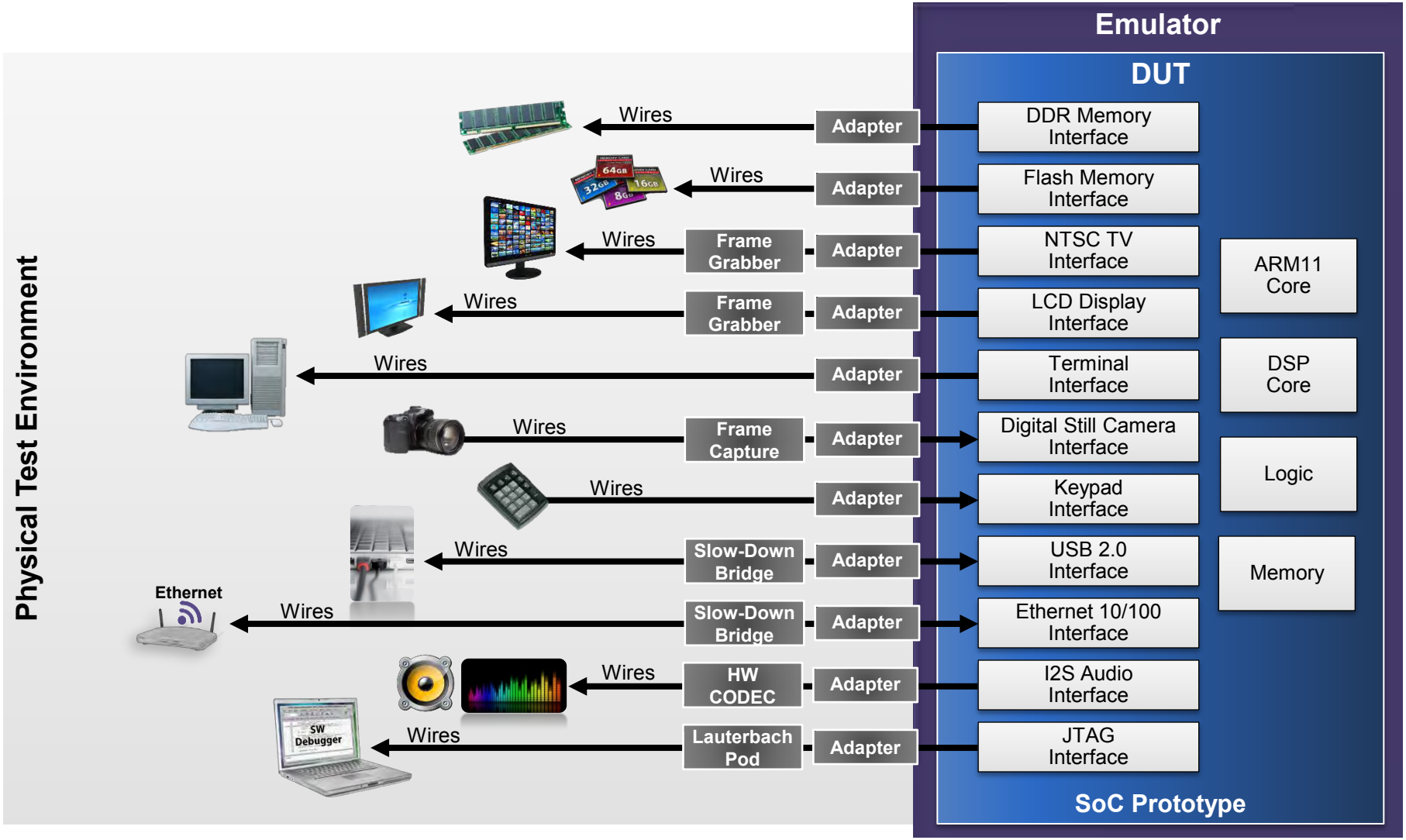
In-circuit Emulation
real-world connections

Simulation Acceleration
up to 100x simulation performance

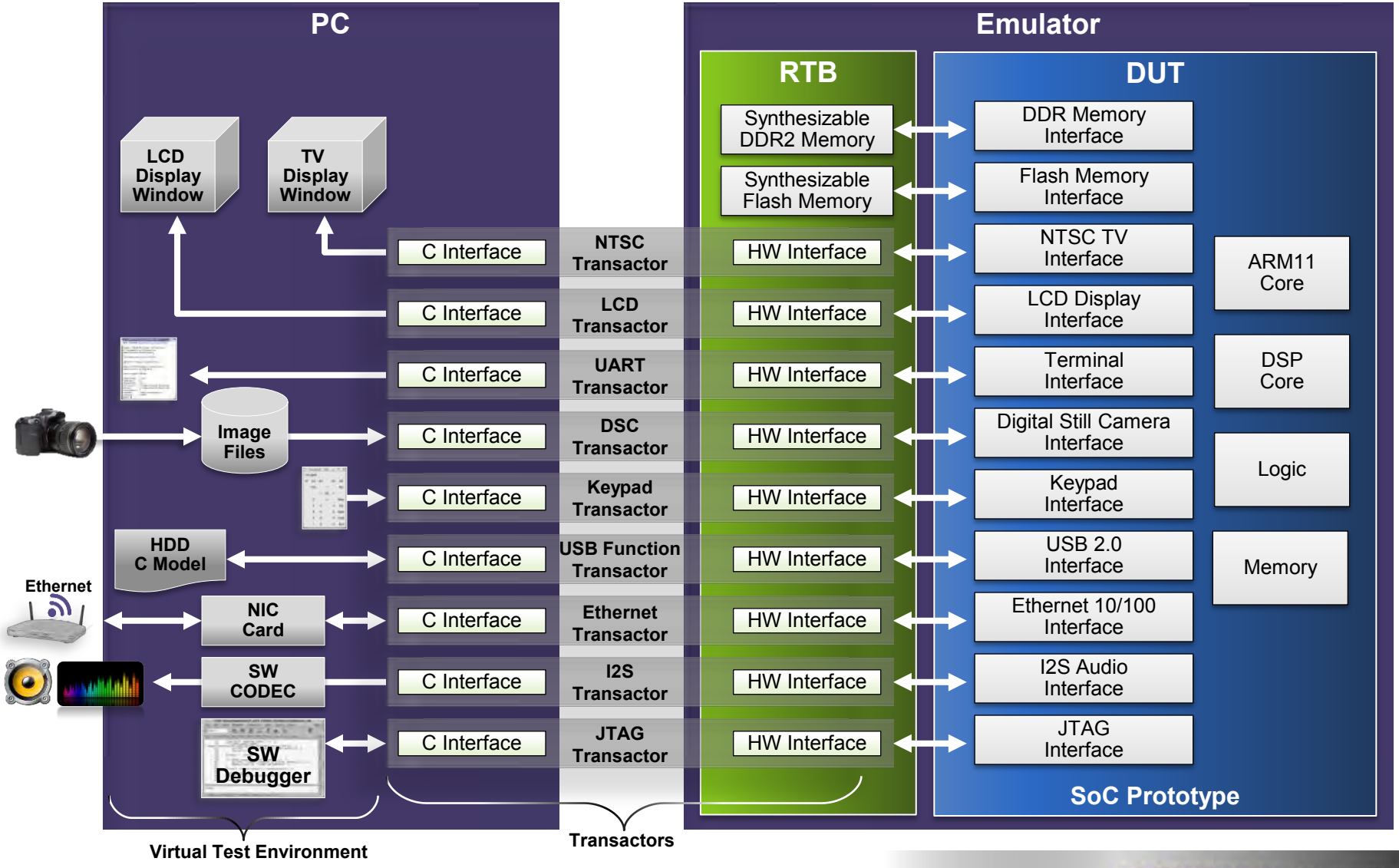
Power-aware Emulation
UPF support and SAIF output

Synthesizable Testbench
higher performance

Environment for In-Circuit Emulation

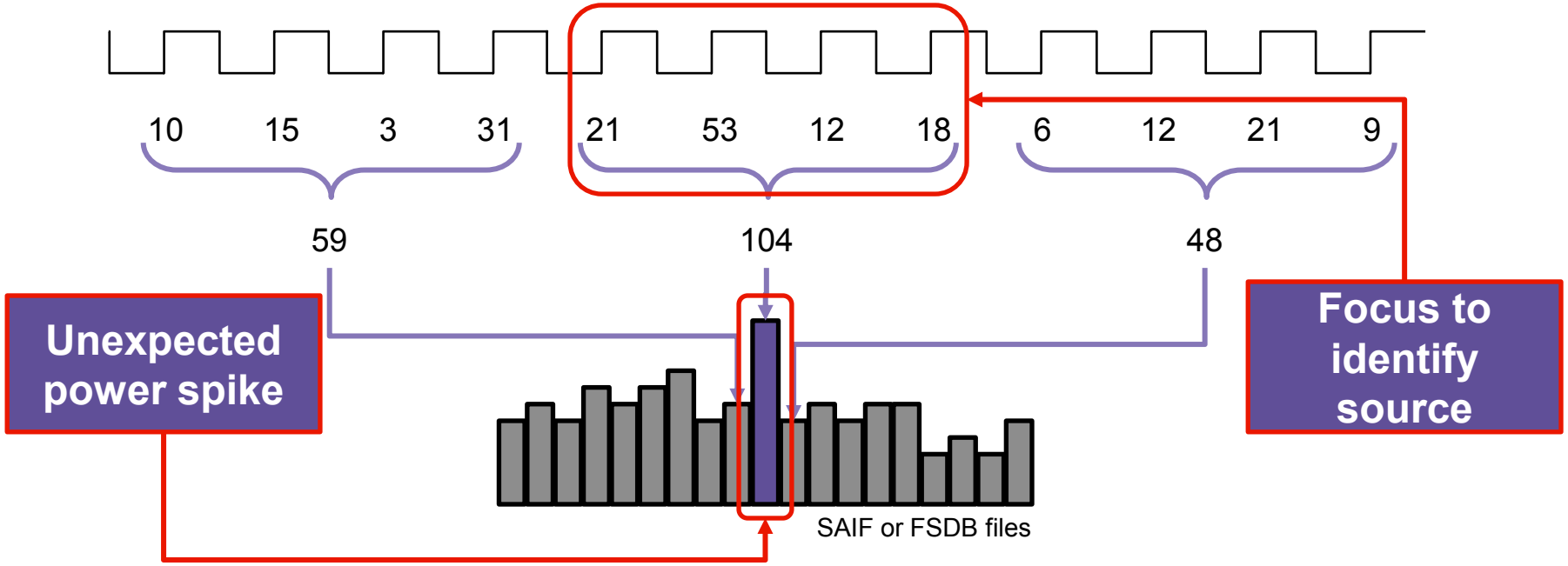


Transaction-based Verification Environment



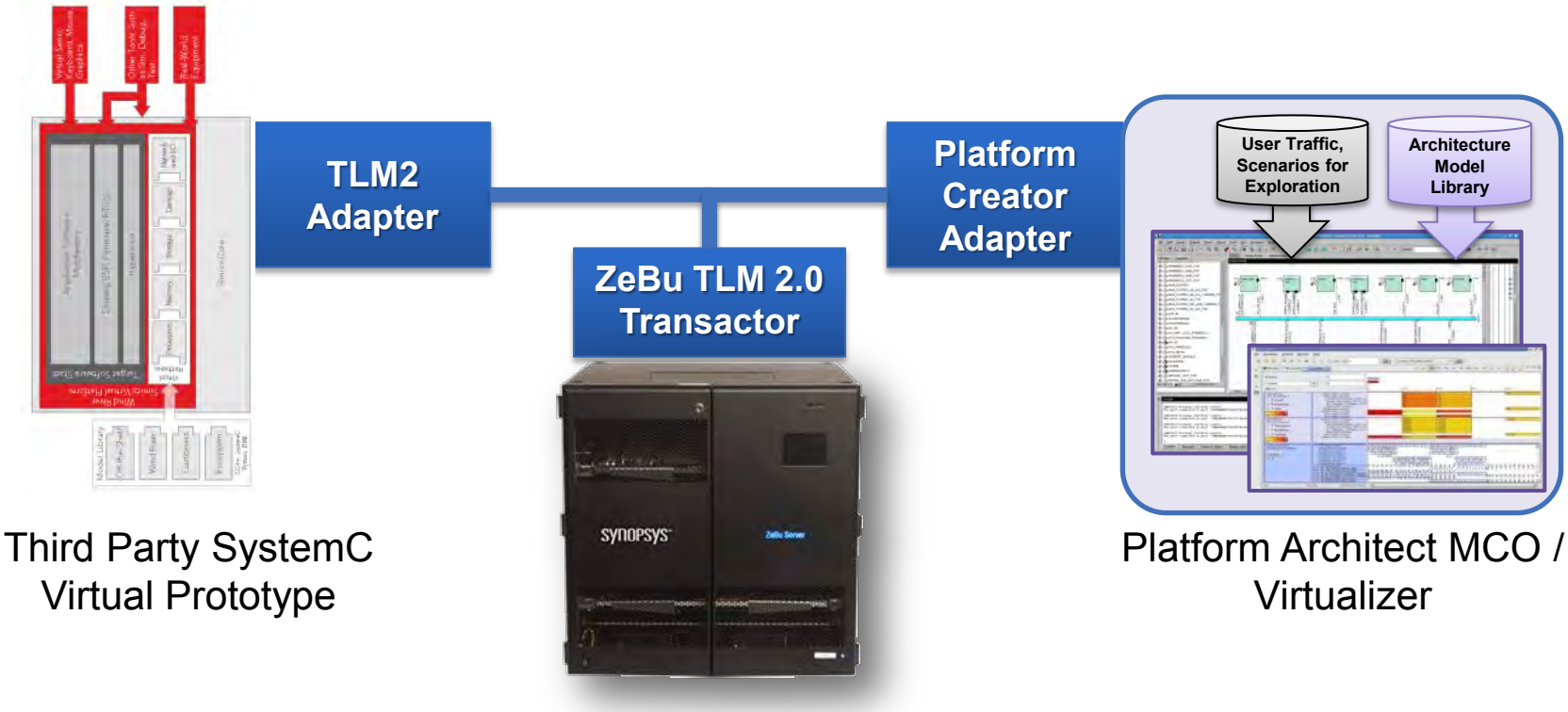
ZeBu Power Analysis Generates Power Profiles

- Ideal for block and system-level analysis with hardware and software
- Captures every transition within each clock period
  - User-programmable: blocks, registers, buses, entire SoC
- Includes cumulative total spanning user-defined clock numbers
  - Large span for highest performance
  - Short span for highest accuracy
- Works seamlessly with PrimeTime



ZeBu Hybrid Emulation

Architecture optimization and early software development



Third Party SystemC Virtual Prototype

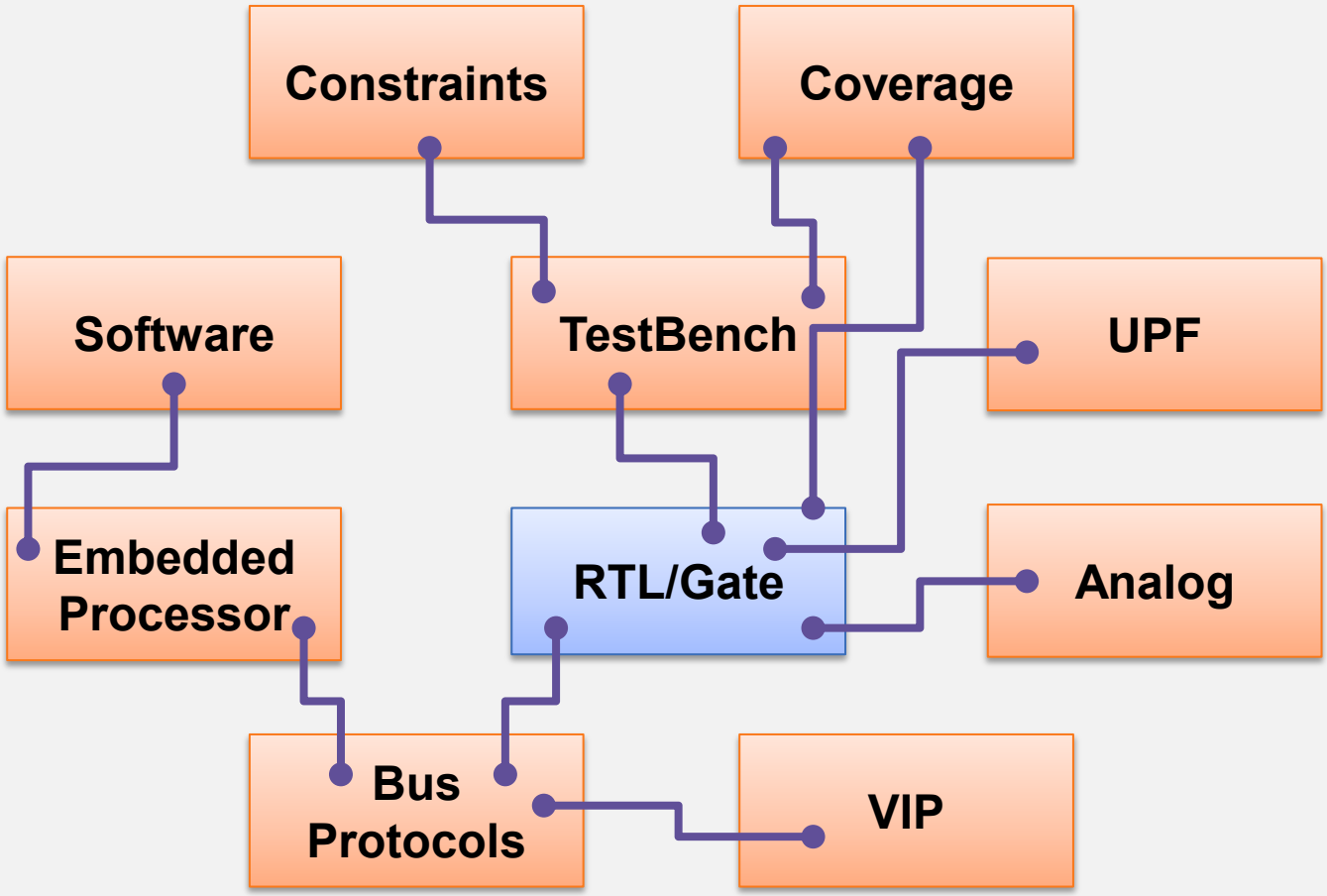
Platform Architect MCO / Virtualizer

- RTL runs at high speed in ZeBu while processor model or other components run in virtual prototype
- Reduces need to have high level models for all components

Agenda

- Market Trends
- Emulation HW Considerations
- Emulation Scenarios
- Debug

Why Is SoC Debug so Complex?



Verdi³'s Unique Technology

Design

Optimized for debug

Combining

Uncluttered

Debug Oriented DB

De facto Standard FSDB

Simulator Verification and Platform Independent

Clock Tree Extraction/Debug

Power-Aware Debug

Siloti

Trace This Value

Trace Active X

Assertion Analysis

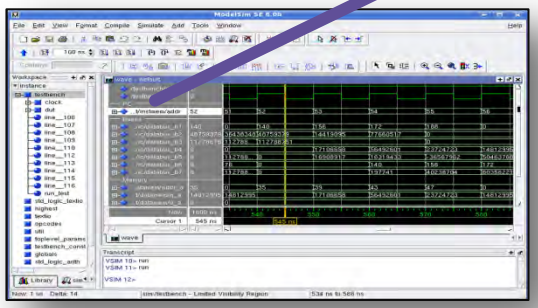
Verdi Behavior Analysis Engine

Verdi³ Visualization

Debug Automation

- Database in not optimized for debug

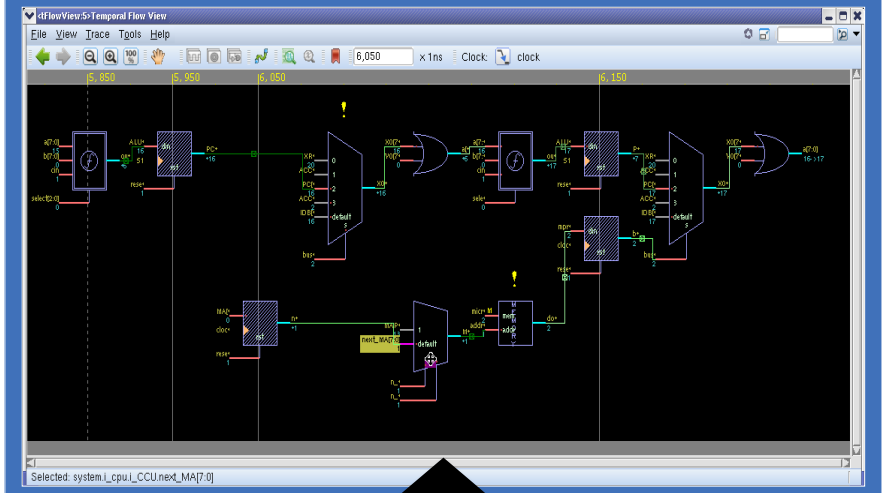
Native Debug A



Native Debug B

ZeBu Simulator-Like Debug with Verdi^3

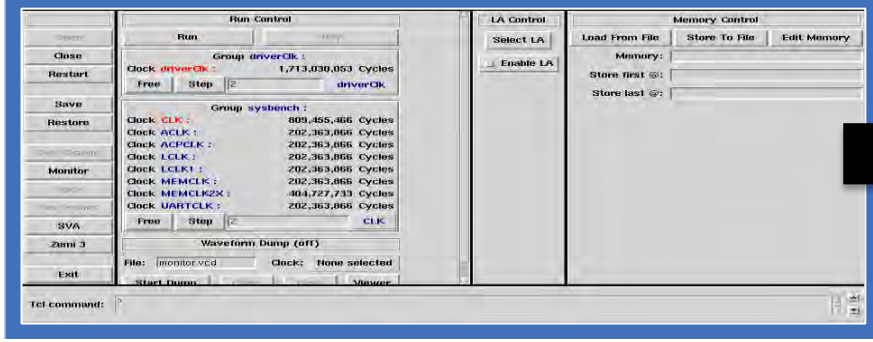
RTL & Gate-Level Hierarchical Debug



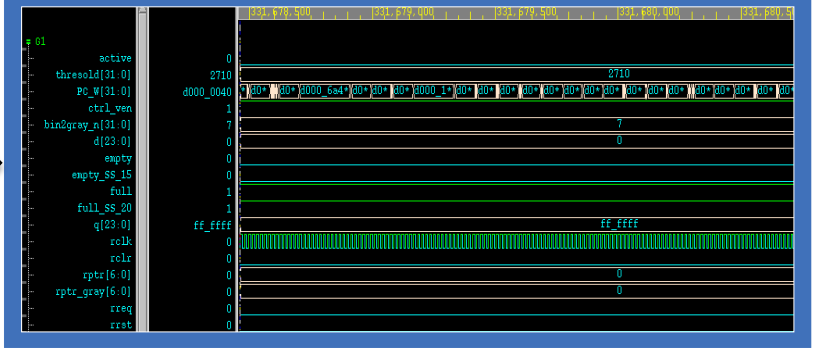
- **Full visibility (RTL & gate level)**
  - All registers, nodes, memories
  - Run-time and post-run debug
  - No recompiles required
- **Open standard support**
  - FSDB, VCD, etc.
- **Transaction level debug**
- **iCSA integration for fast time to waveform**

New!

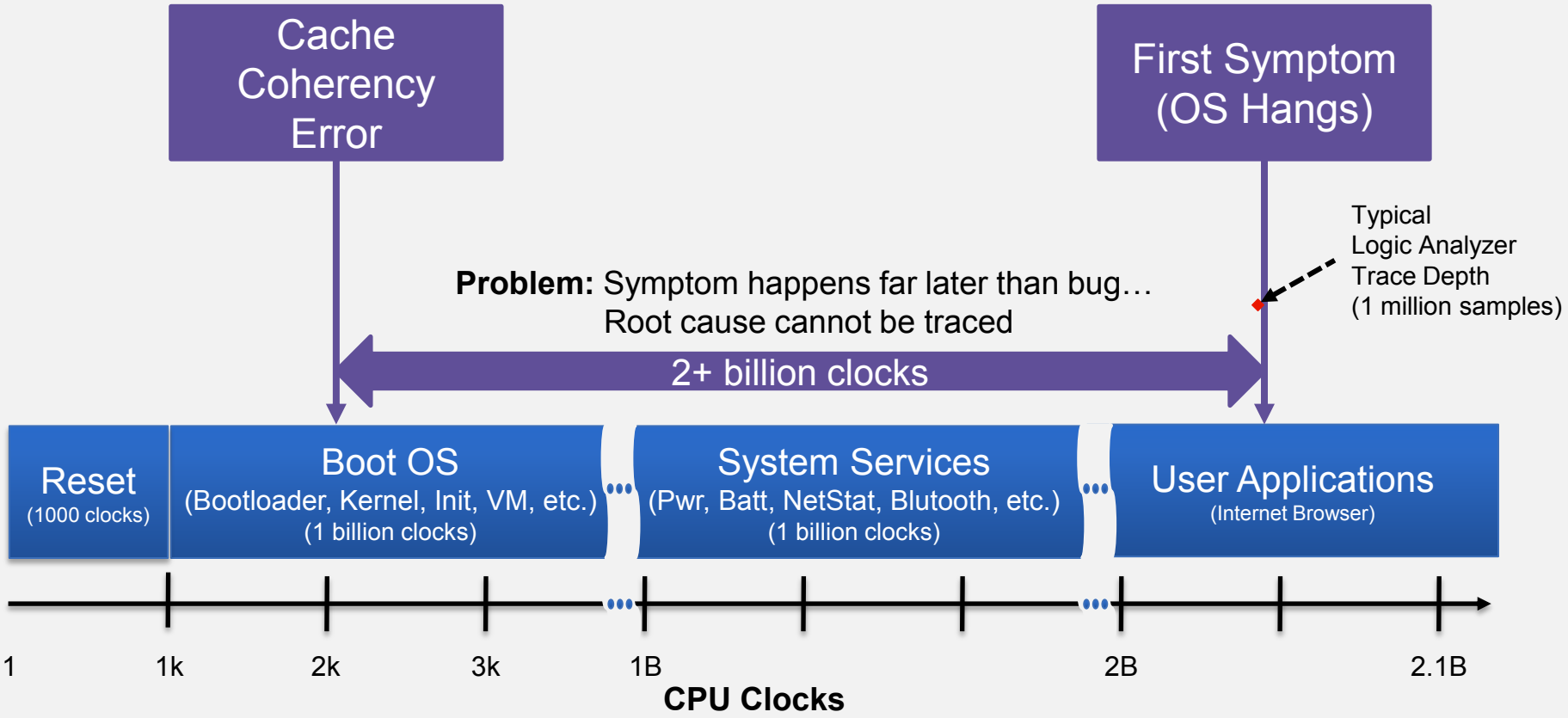
Runtime Control



Verdi^3 Waveform



Complex Debug Scenarios

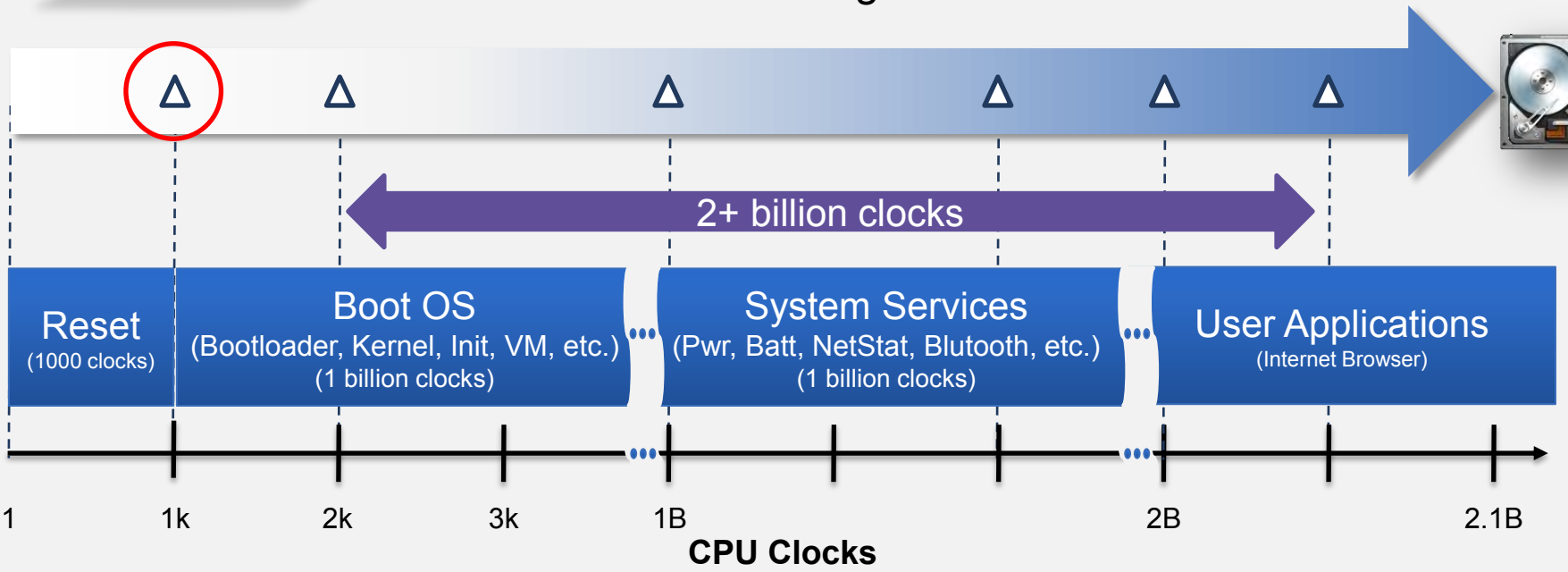


ZeBu Post Run Debug

Billion-Cycle Full Visibility, Optimized for HW/SW Co-Verification

- Testbench captures DUT state periodically
- DUT inputs captured on every clock
- Data stored on Host PC disk drive
- To debug, user selects a restore point…
  and loads it into ZeBu to generate waveforms

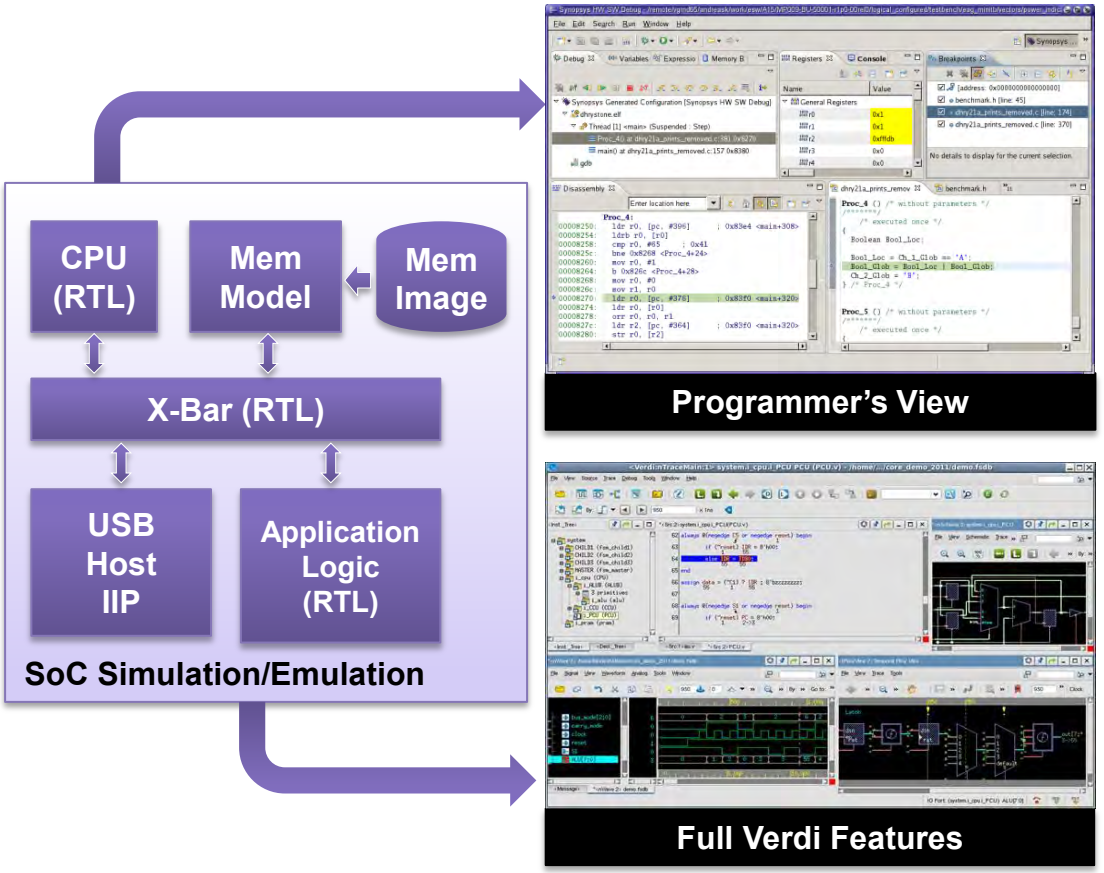
Host Machine



HW/SW Debug Overview

Embedded Processor Debug with Synchronized RTL, C, Assembly

- Enables co-debug between RTL and SW
- HW and SW debug synchronized in time
- View C/Assembly source, C variables, stack, memory
- Debugs multiple core simultaneously
- Supports all popular cores
- Easy to support additional cores or custom cores
- Custom Core support without exposing CPU internals



HW/SW Debug Use Models

Verification Environment with C-Tests

- Part of SoC verification schedule
- Hardware debug with C-tests /stimulus
- C-tests may have minimal OS or boot code
- Requires concurrent software and hardware debug

Use HW/SW Debug for this task

Driver Development

- Software (driver) development
- Fast speed is required (>1MHz)
- Approximate hardware is acceptable

Use Virtualizer for this task

Prepare for First Silicon Bring-Up

- Debug synthetic tests mimic specific use scenarios
- Tests run on a bare-metal OS
- Develop and bug tests on a pre-silicon model
- Get ready for silicon bring up

Use HW/SW Debug for this task

First Silicon Debug

- Observed failure running test on first silicon
- Debug the failure and isolate a design bug
- Create/Validate software/firmware workaround

Use HW/SW Debug for this task

The End