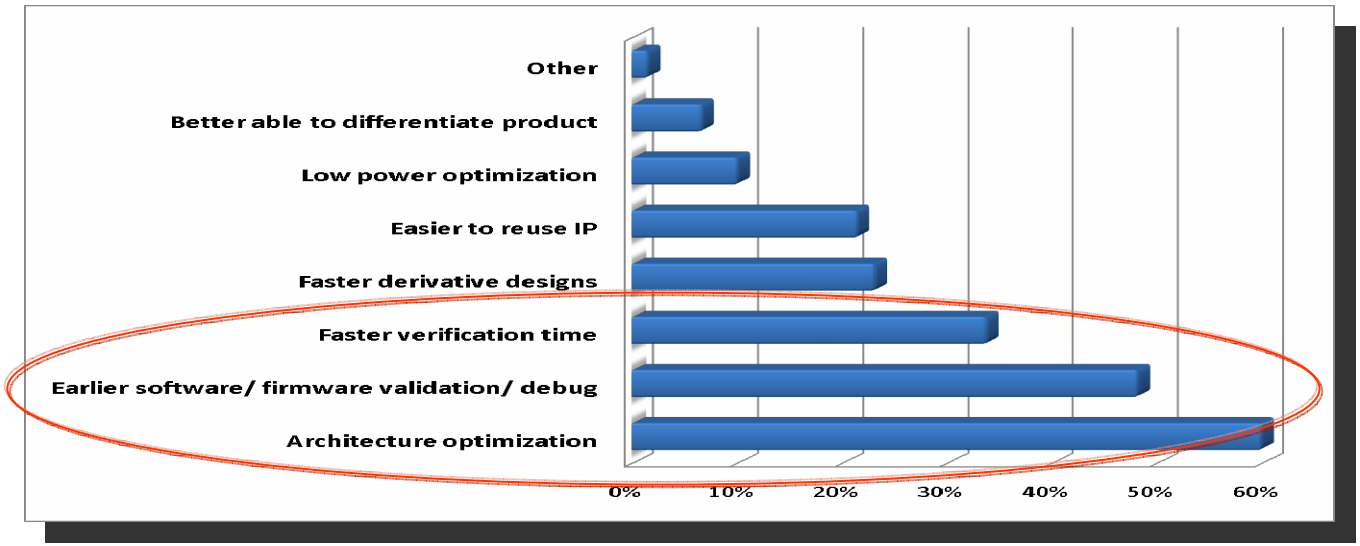
TLM Platforms Re-Define Hardware Virtualization

Jon McDonald

Why Build Transaction Level Platform?

- **The higher you go toward system-level abstraction, the greater the leverage**
  - **Optimize system architecture** for area, performance and power
  - Enable early **software validation/debug** against fast abstracted model
  - Reduce **verification time**
    - By quickly building and validating a transaction level model
    - By reusing the model for validating the implementation



Source: 3rd party ESL survey, Jan 2009

System Architecture Optimization

- Optimizing those **system architecture attributes** impacting area, timing and power
  - **10X power variation** resulted from system architecture tradeoffs in the following example

Transmitter Design (IFFT Block)	Min. Freq. to Achieve Req. Rate	Average Power (mW)
Combinational	1.0 MHz	3.99
Pipelined	1.0 MHz	92
Folded (16 bfy4s)	1.0 MHz	
Folded (8 bfy4s)	1.5 MHz	
Folded (4 bfy4s)	3.0 MHz	40
Folded (2 bfy4s)	6.0 MHz	21.10
Folded (1 bfy4)	12.0 MHz	34.6

~X10

Source: Chip Design Magazine ESL Synthesis + Power Analysis By Holly Stump and George Harper

IEEE 802.11a

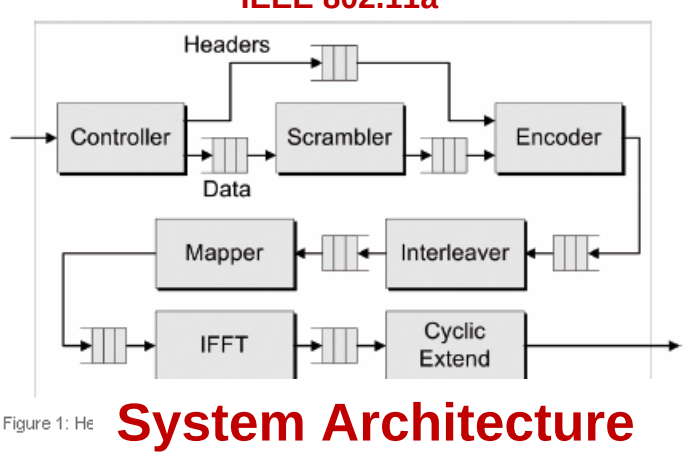


Figure 1: He

System Architecture

Only Transaction Level Platform Allows Users to Quickly Modify and Evaluate Various System Architectures

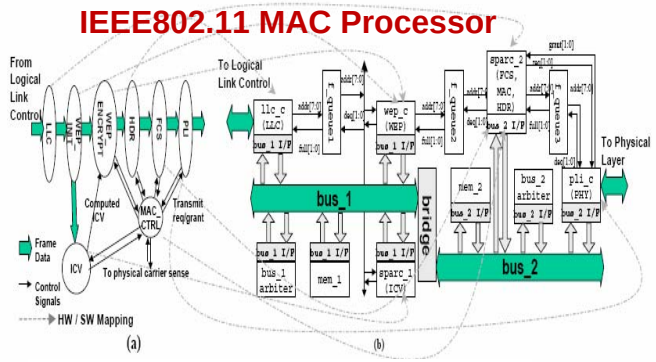
Hardware/Software Optimization

- Addressing **hardware/software architecture** tradeoffs
  - **2X power variation** resulted from hardware/software tradeoffs in the following example

Table 1: Results from power profiling of candidate architectures for the IEEE 80 2.11 MAC processor

Architecture	*Average Power*			*Power Profiling*		*CPU time*
	HW/SW co-estimation (mW)	Trace based profiling (mW)	Relative error (%)	Avg. absolute error (mW)	Avg. relative absolute error (%)	Trace Based Profiling (sec)
All HW, Single bus	380.7	380.8	0.03	1.8	0.4	0.21
All HW, Two busses	[illegible].2	414.9	0.80	4.1	2.2	0.36
ICV in SW, Single bus	[illegible]4.2	543.0	0.18	24.0	4.1	0.26
ICV in SW, Two bus	[illegible]6.2	592.9	0.55	13.0	3.4	0.42
WEP in S, One bu	~X2	614.0	0.16	45.0	7.3	0.25
WEP in S, Two bus		625.5	0.45	41.8	6.2	0.37
FCS in SW, Single bus	[illegible]7.2	512.2	0.97	19.0	3.2	0.22
FCS in SW, Two busses	[illegible]0.7	587.4	0.22	4.6	2.2	0.37
FCS, ICV in SW, Single bus	[illegible].2	672.2	0.00	26.3	3.5	0.26
FCS, ICV in SW, Two busses	766.8	763.5	0.43	23.3	4.3	0.42

Source: Fast System Level Power Profiling for Battery Efficient System Design Kanishka Lahiri Dept. of ECE UC San Diego , Anand Raghunathan C&C Research Labs NEC USA



Only Transaction Level Platform Allows Users to Quickly Modify and Evaluate Various Hardware/Software Configurations

Software Compiler Dependent Power

- Image Recognition Package running on an ARM
  - O0 - No Optimizations
  - O3 - Rename registers, inline functions
- Different compiler options may result in different profiles (cache access)

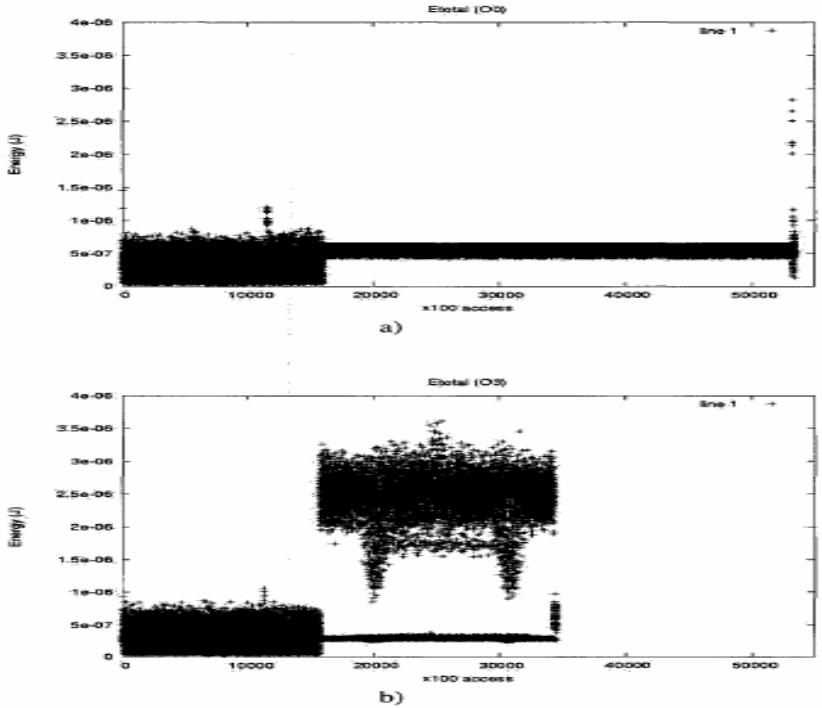
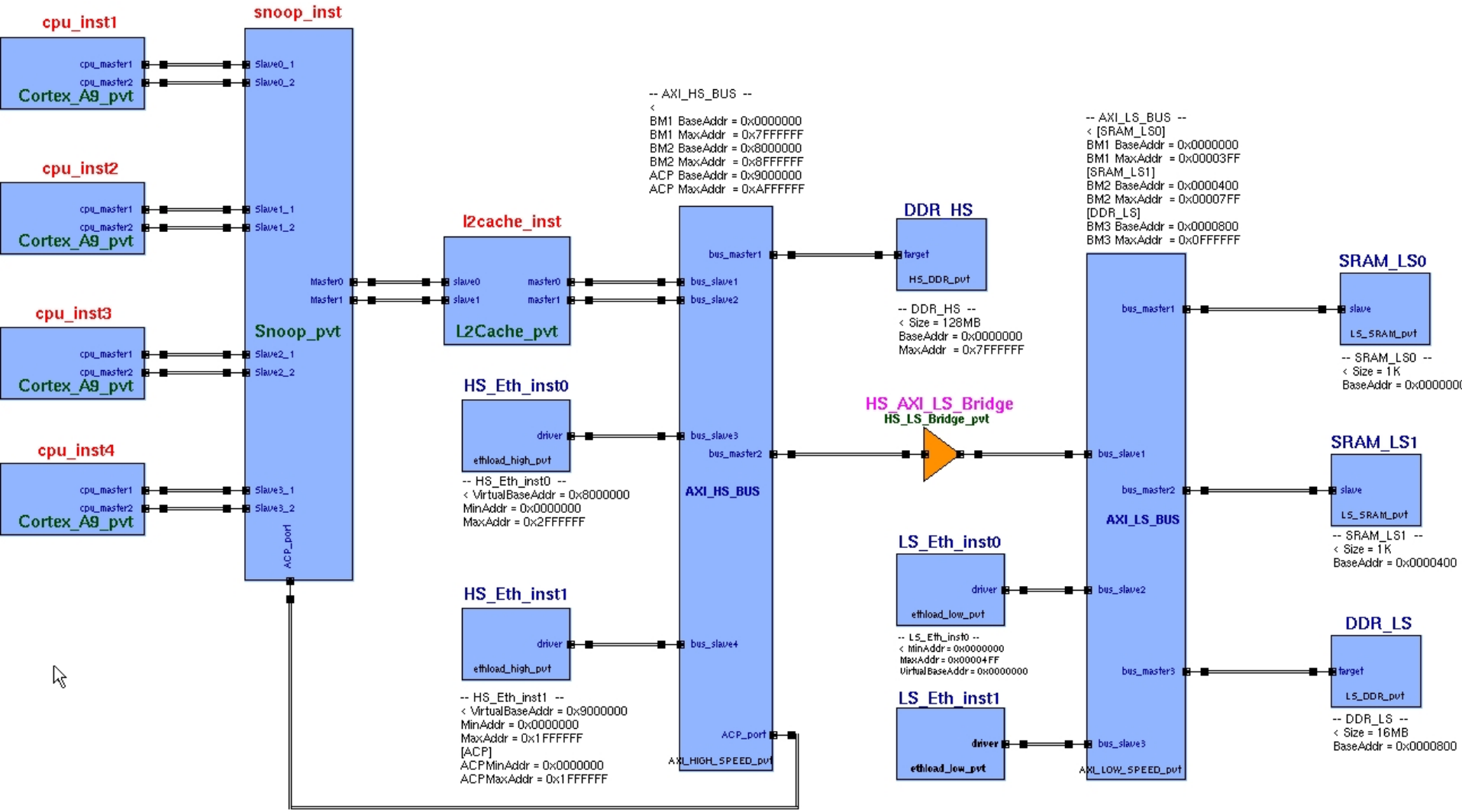


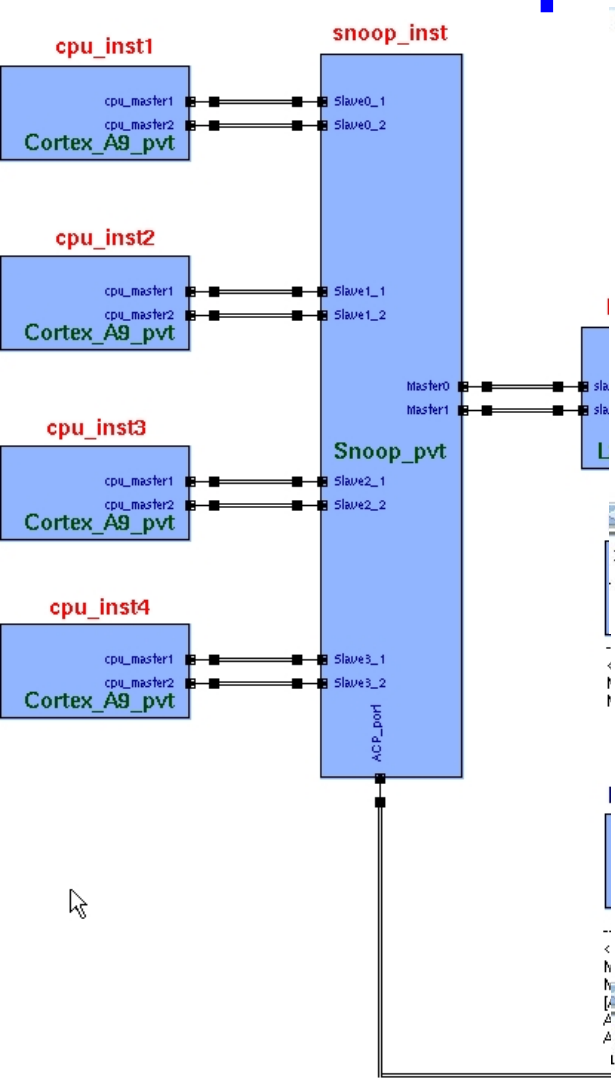
Figure 4.12: Influence of compiler optimizations: total energy per access a) *susan_O0*; b) *susan_O3*

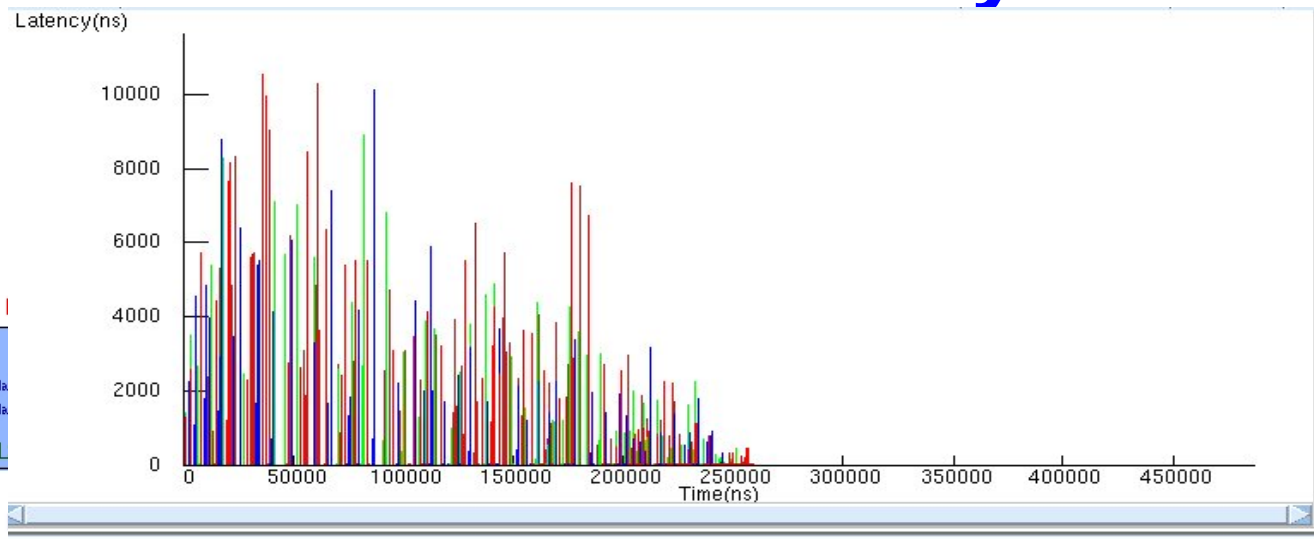
Source: POWER ESTIMATION AND POWER OPTIMIZATION POLICIES FOR PROCESSOR-BASED SYSTEMS José L. Ayala Rodrigo Universidad Politécnica de Madrid

Example Architecture



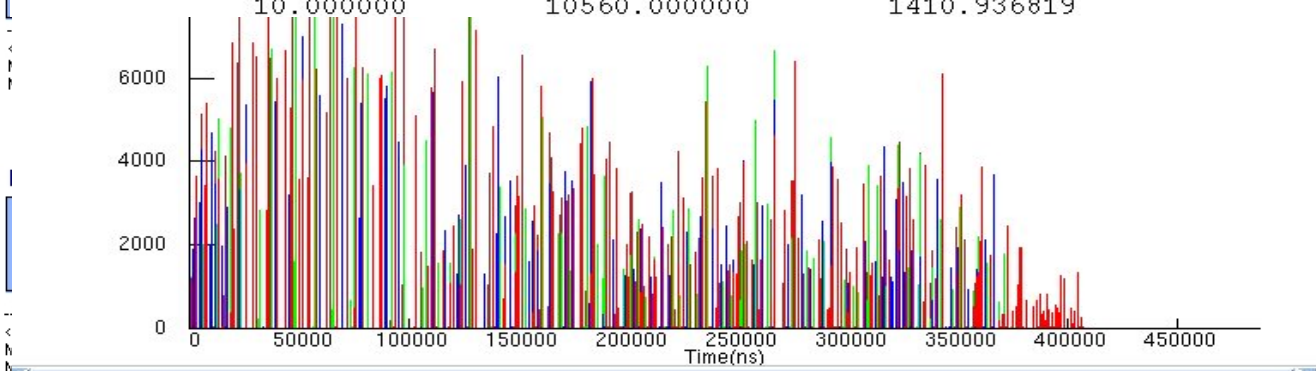
Example Architecture: Latency





ns

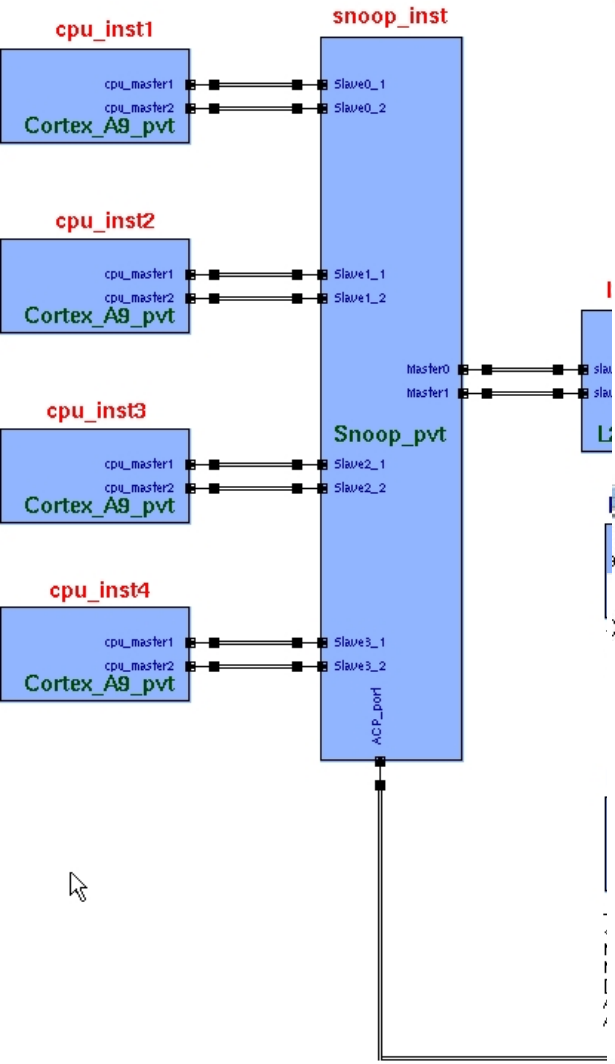
Count	Minimal	Maximal	Average
	10.000000	10560.000000	1410.936819

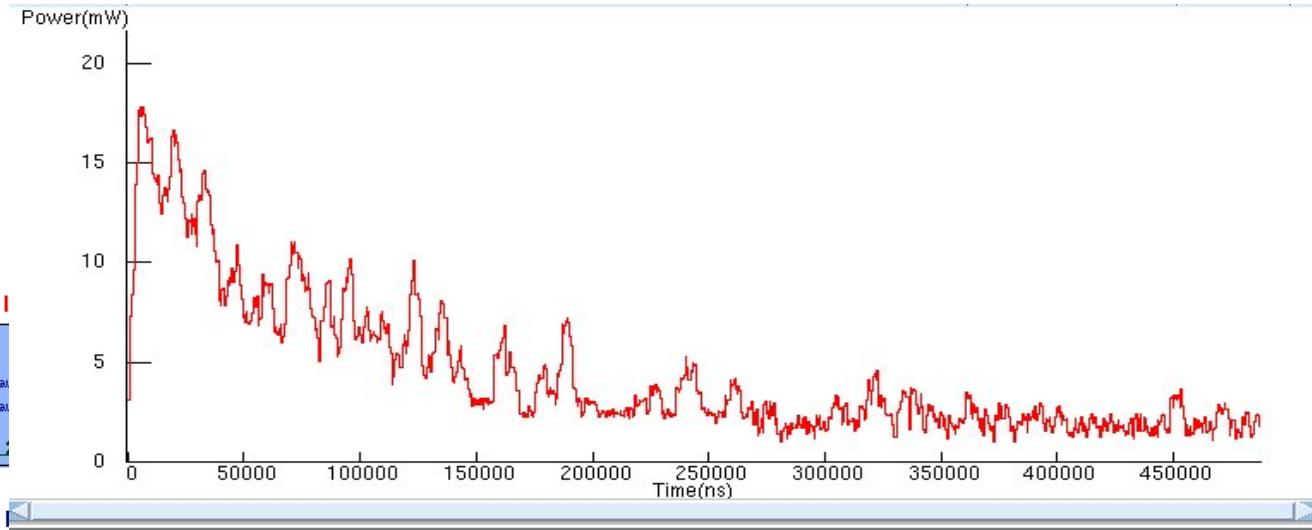


ns

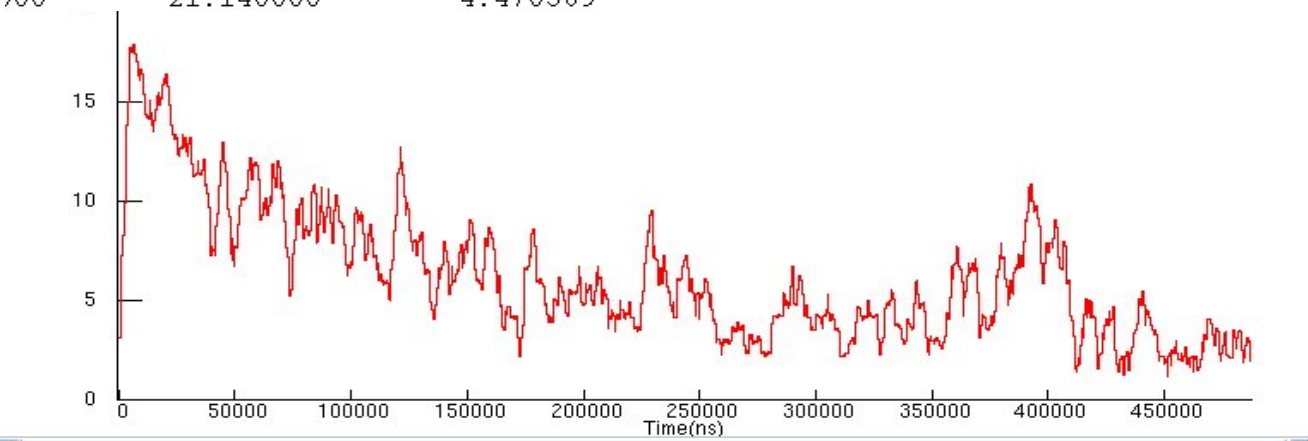
Count	Minimal	Maximal	Average
	10.000000	9390.000000	2013.231922

Example Architecture: Power



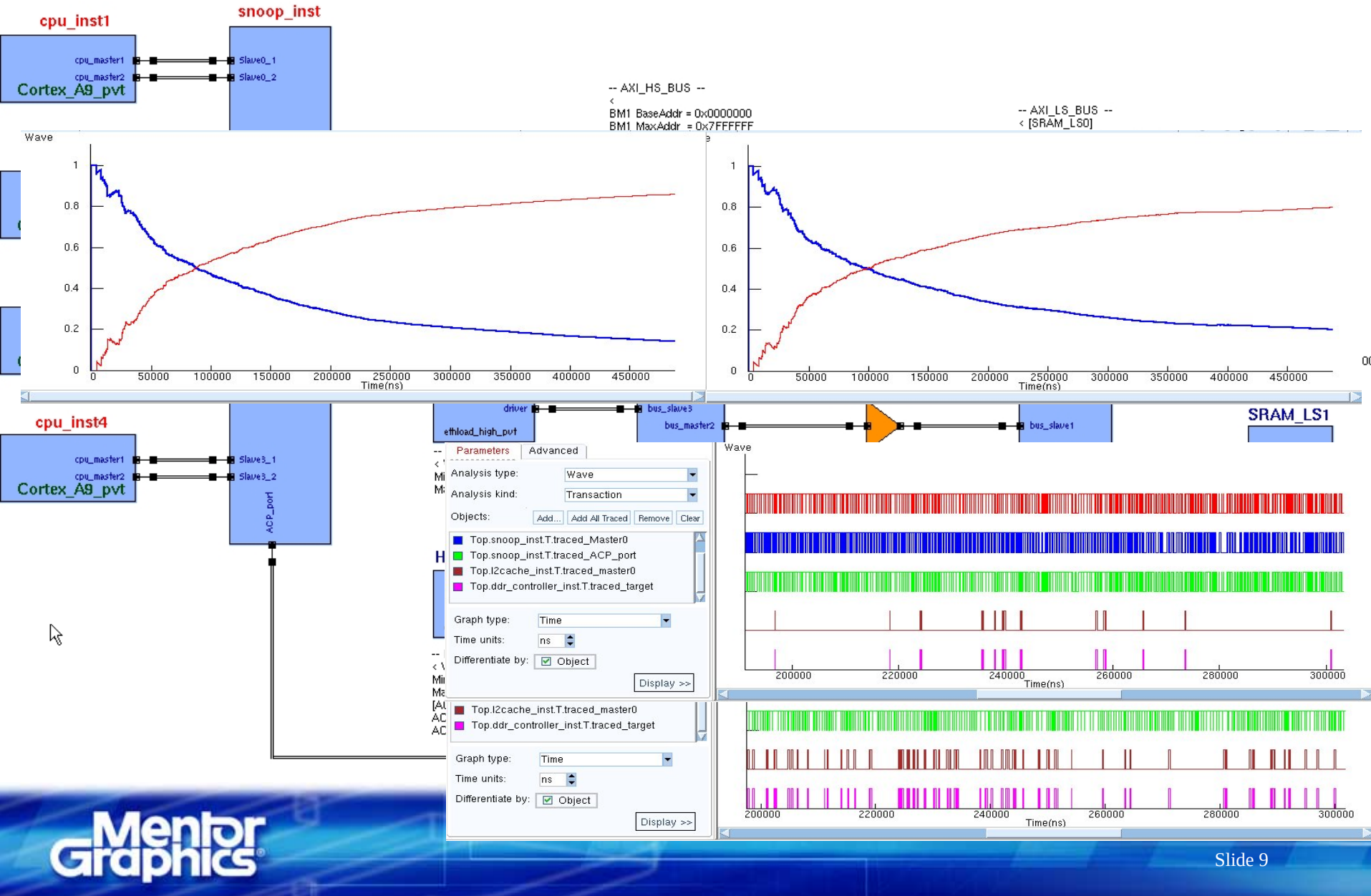


ns	Peak	Average
000	21.140000	4.470589



ns	Peak	Average
000	20.540000	6.180625

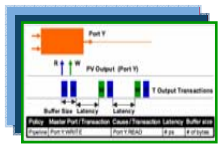
Example Architecture: Cache, DDR

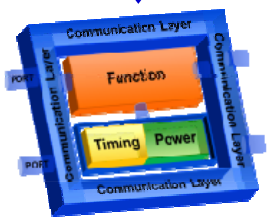


Vista™ Main Capabilities

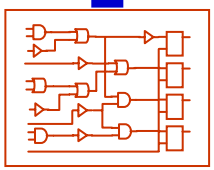
Modeling | **Assembly** | **Debug** | **Analysis**

Policies

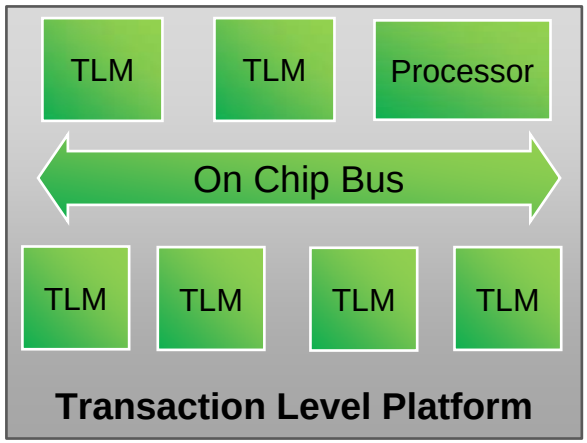


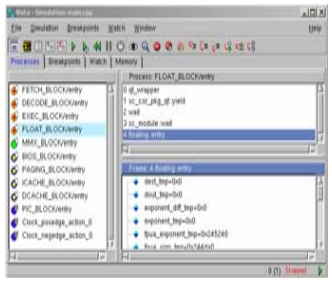


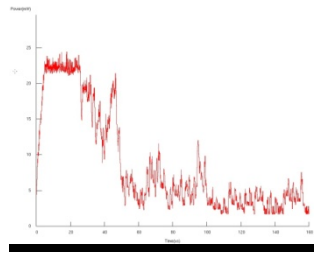
Vista Model Builder



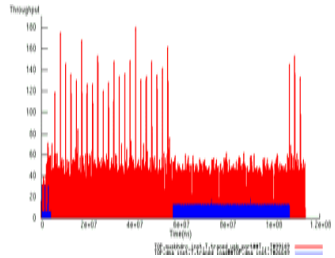
RTL IP







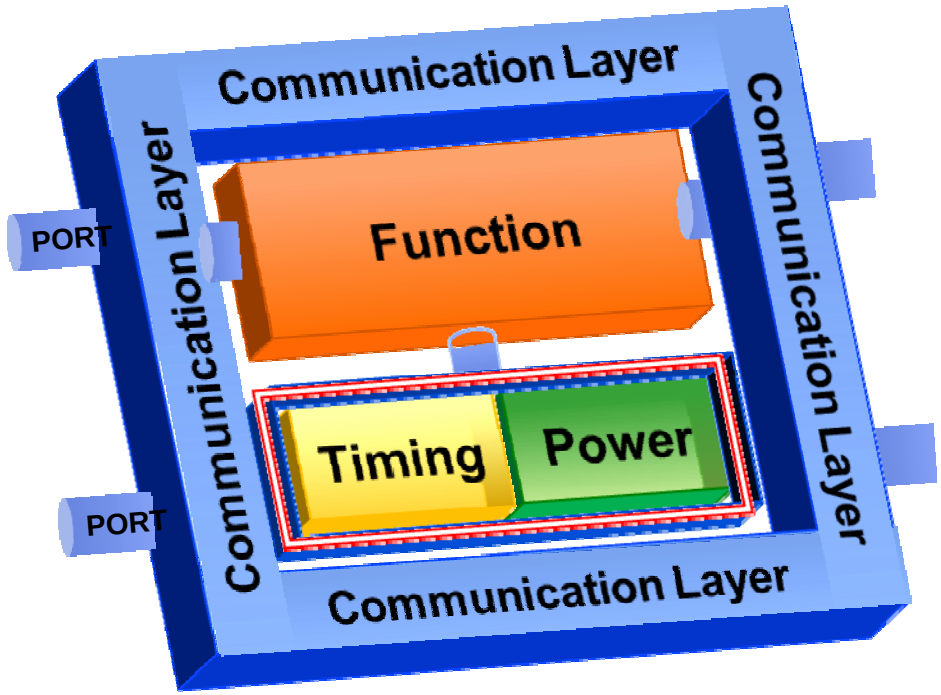
Power



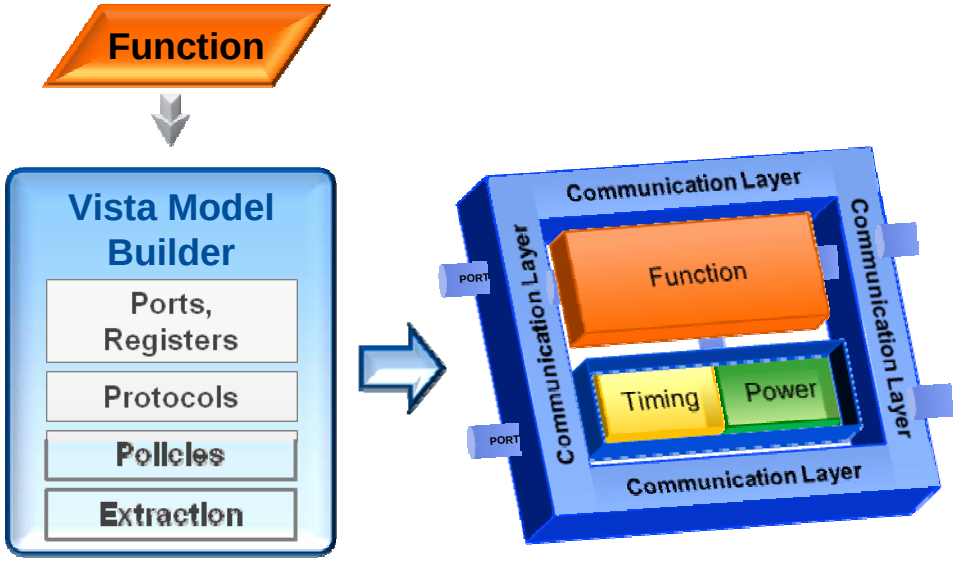
Performance
(latency, utilization, bandwidth)

Vista Scalable TLM Models

- Based on SystemC and the **OSCI TLM 2.0 standard**
- Scalability is accomplished by
  - Modeling the core **Function**
  - Providing the **Communication Layer**
  - Adding a separate **Timing/Power** model

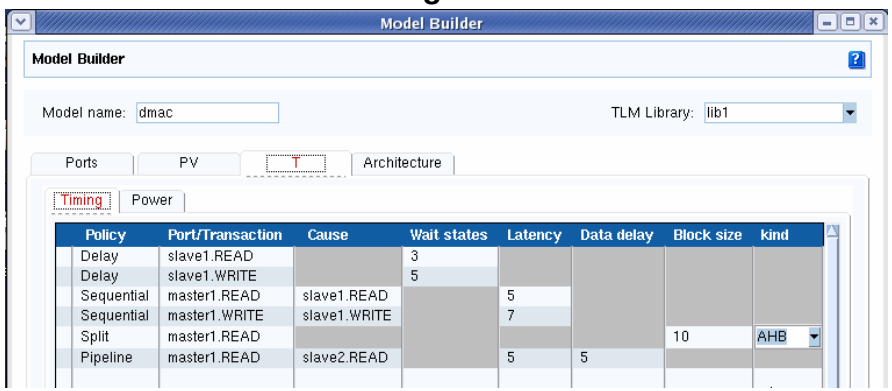


Vista Scalable TLM Timing Model

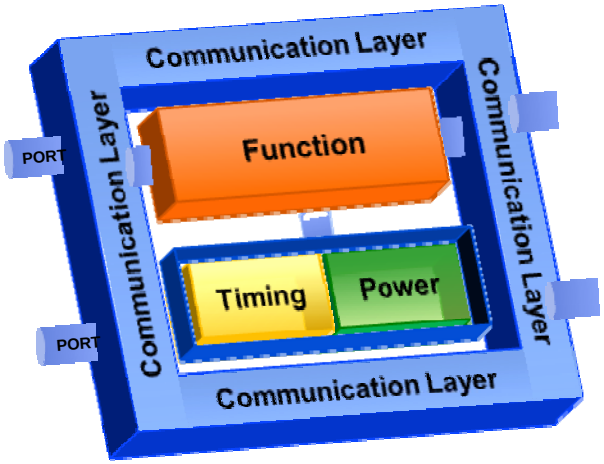


- Single transaction level timing model supporting
  - LT (loosely time)
  - AT (approximated time)
- GUI based timing definition policies
- Policy types:
  - Delay, Sequential, Split, Pipeline
- Timing values:
  - Wait states, Latency, Data Delay

Modeling Policies



Vista Scalable TLM Power Model



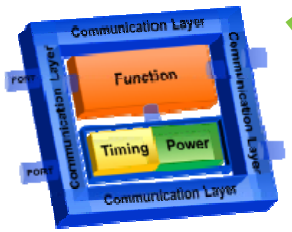
- Transaction-level modeling of all power types
  - Static (leakage) power
  - Clock tree power
  - Dynamic power (per transaction)
- Dynamic power is assigned to each transaction type
- Vista TLM power model is
  - **Reactive** to incoming traffic and inner states
  - Supports **voltage and frequency scaling**

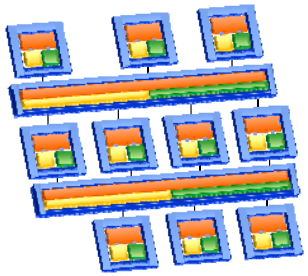
Power Policies

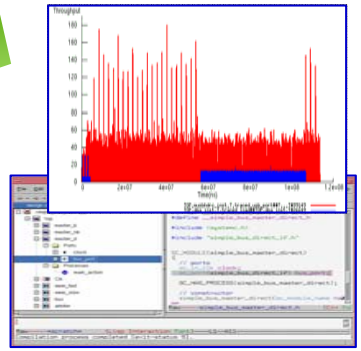
Dynamic Power		
Transaction	**Power**	**Time Interval**
USB.READ	0.16mw	200
USB.WRITE	0.44mw	20
Clock Tree Power		
0.5mw		
Static (Leakage) Power		
0.2mw		

Architectural Power Optimization Flow

Model Builder







Dynamic Power		
Transaction	**Power**	**Time Interval**
USB.READ	**0.16mw**	**200**
Clock Tree Power		
0.5mw		
Static (Leakage) Power		
0.2mw		

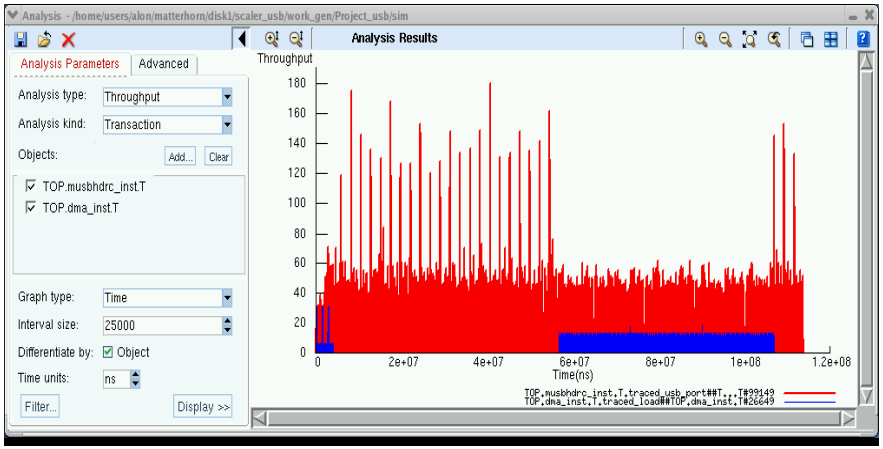
- Model TLM Timing and Power via policies
- Assemble the transaction level reference platform
- Analyze Timing/Power in a system context
- Modify Timing/Power policies or the platform architecture
- Quickly iterate optimizing for timing and power

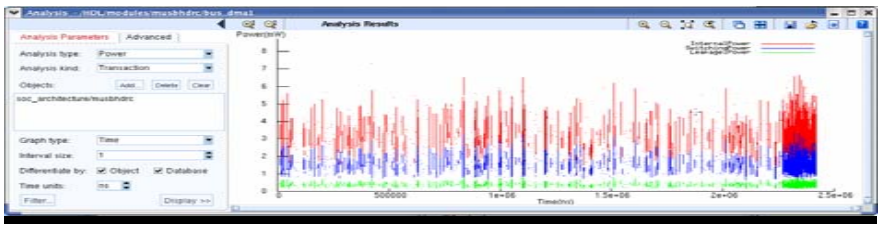
Quick Optimization of Power and Performance before RTL S15

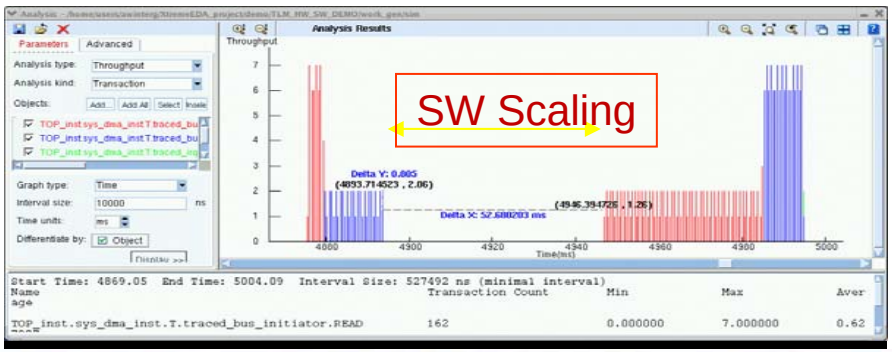
S15

Lisa,
Please highlight the message at the bottom
Shabtay, 7/22/2009

Vista Wide Range of Analysis Toolsets

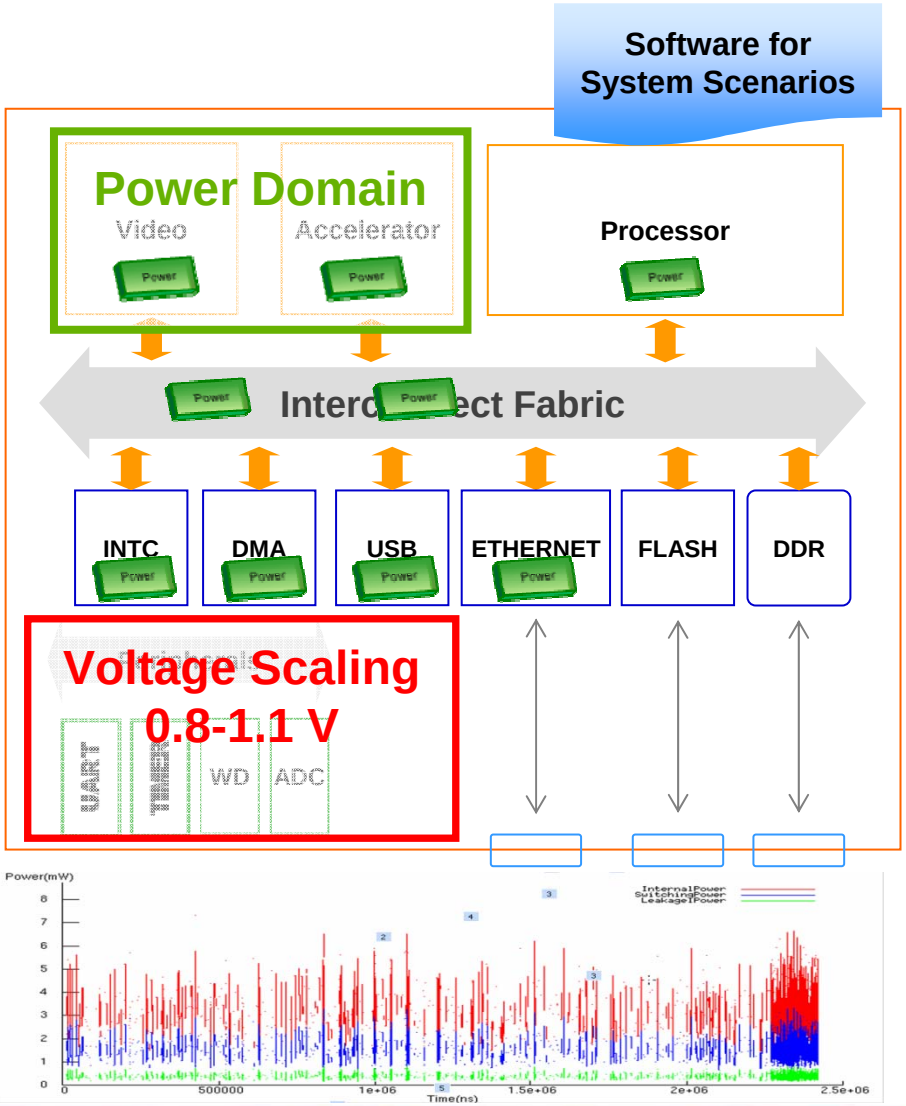






- Functional Analysis
  - Data ID and Tracing
- Timing/Performance Analysis
  - Throughput
  - Latencies
  - Bus Utilization
  - State Distribution
- Power Analysis
  - Dynamic, Static and Clock Power
  - Instance Power
  - Mean and Peak values
  - System and software power profiles
  - Hot Spot analysis
  - Voltage and frequency scaling (DVFS)
  - Power domain management

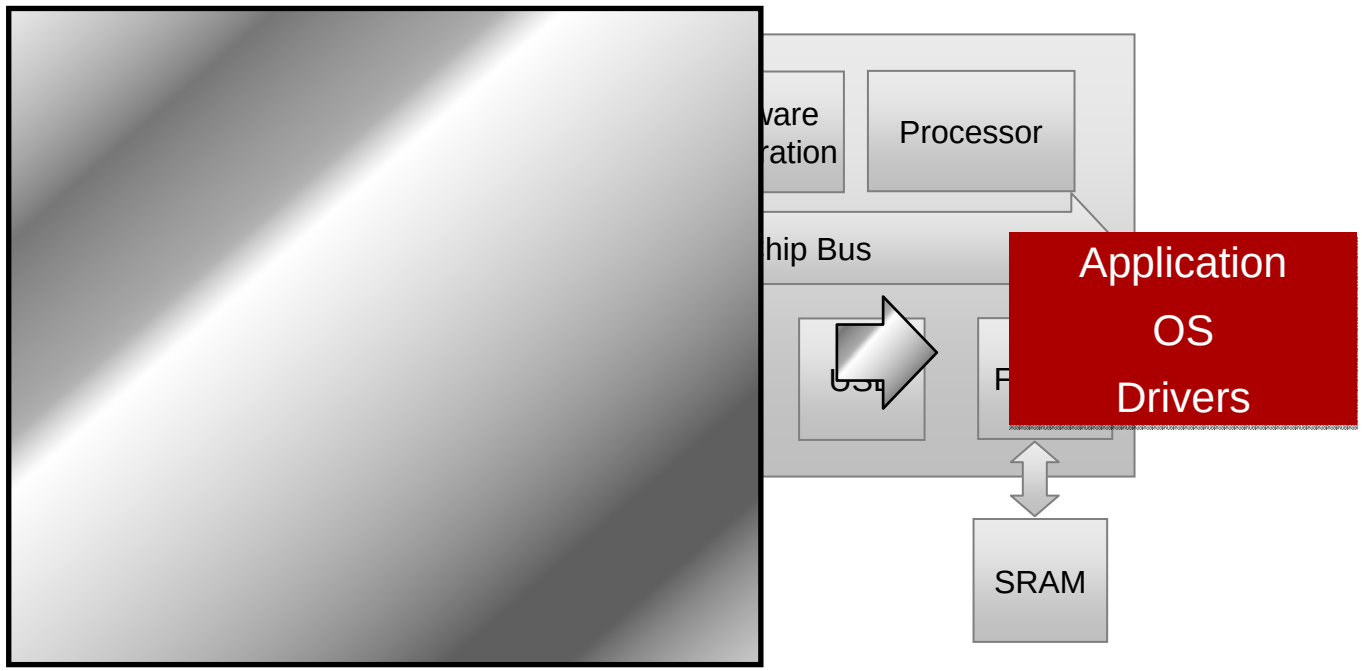
Vista - Optimization Philosophy



- ✓ Use policies to model power/timing of new IP
- ✓ Use accurate power/timing models for legacy IP
- ✓ Analyze power/timing profiles
  - ✓ typical system scenarios
  - ✓ running software application
- ✓ Explore various power/timing domain management strategies and voltage/frequency scaling techniques

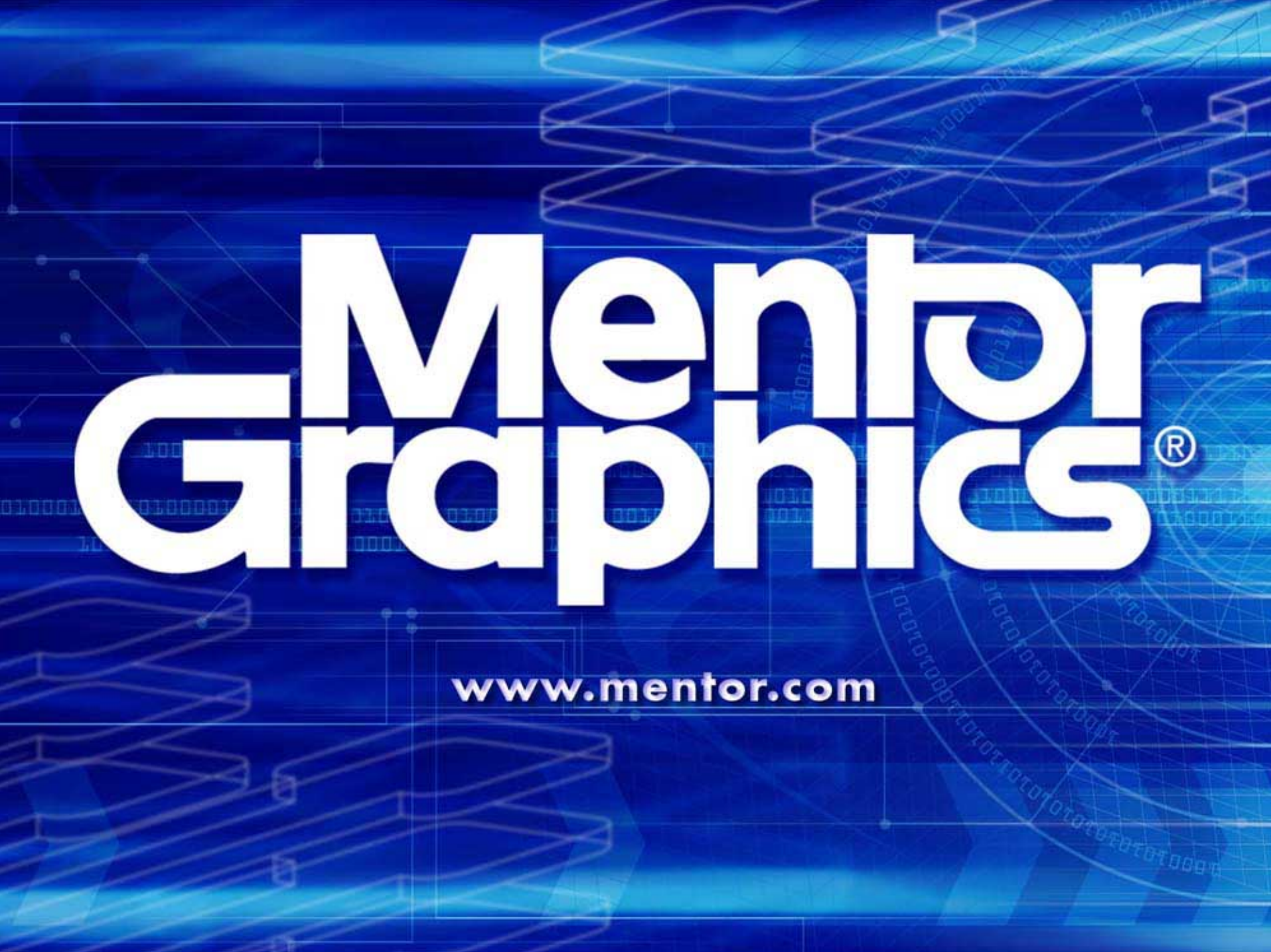
Vista enables HW/SW Co-Development

- Vista produces SystemC virtual reference platform
  - Available for early software development and validation
  - Accurate enough for tuning SW for power and performance



Vista: Power Design and Optimization at the ESL

- Architectural analysis, exploration and optimization for power/timing
  - Unique power/timing modeling capabilities
  - Wide range of analysis toolsets
- Improved model accuracy enables
  - Analysis of system architecture changes
  - HW/SW trade-off
  - Tuning the application software
  - Correlating performance with system workload


Mentor Graphics®
www.mentor.com