

Leveraging Low-Cost **FPGA Prototyping** for Validation of Highly Threaded Server-on-Chip

DV Club - July 2009

Jai Kumar,
Verification Technologist
Sun Microsystems Inc.
jai.kumar@sun.com
<http://sun.com>

Outline

- Verification Challenges
- Emulation alternatives
- FPGA Prototyping Basics
- Prototyping Challenges
- Guidelines
- Results
- Summary

What's in it for you -

Managers:

- Requirements – effort, \$\$, Time, tools

Engineers:

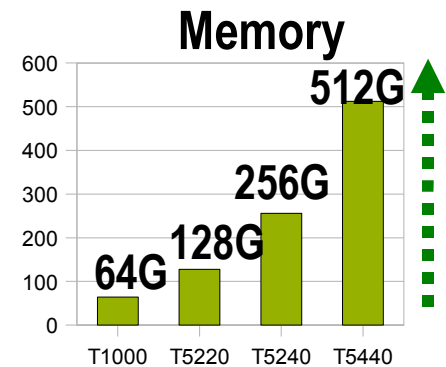
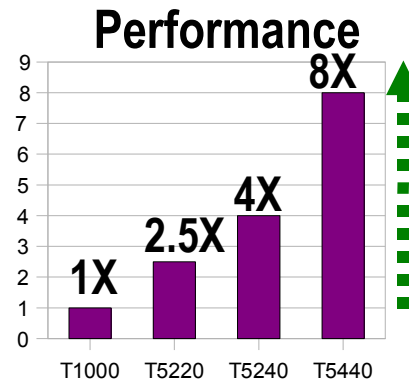
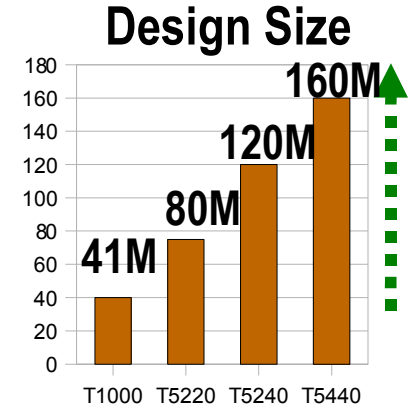
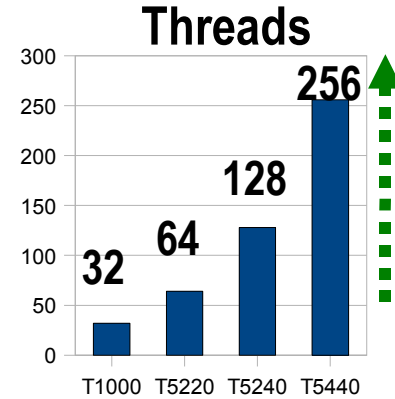
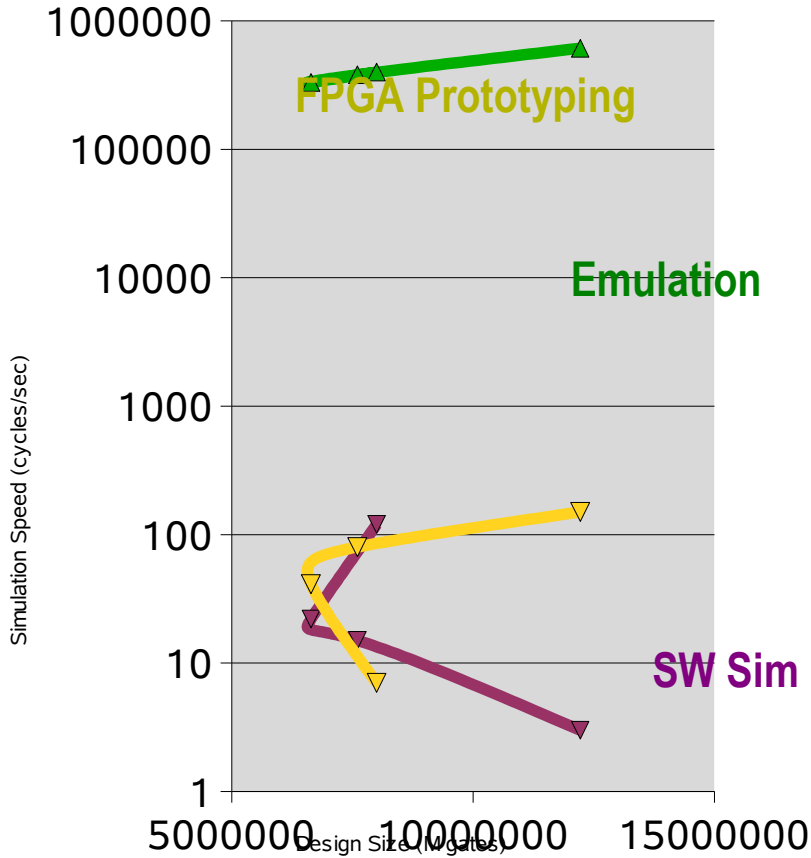
- Challenges

- Avoid Pitfalls

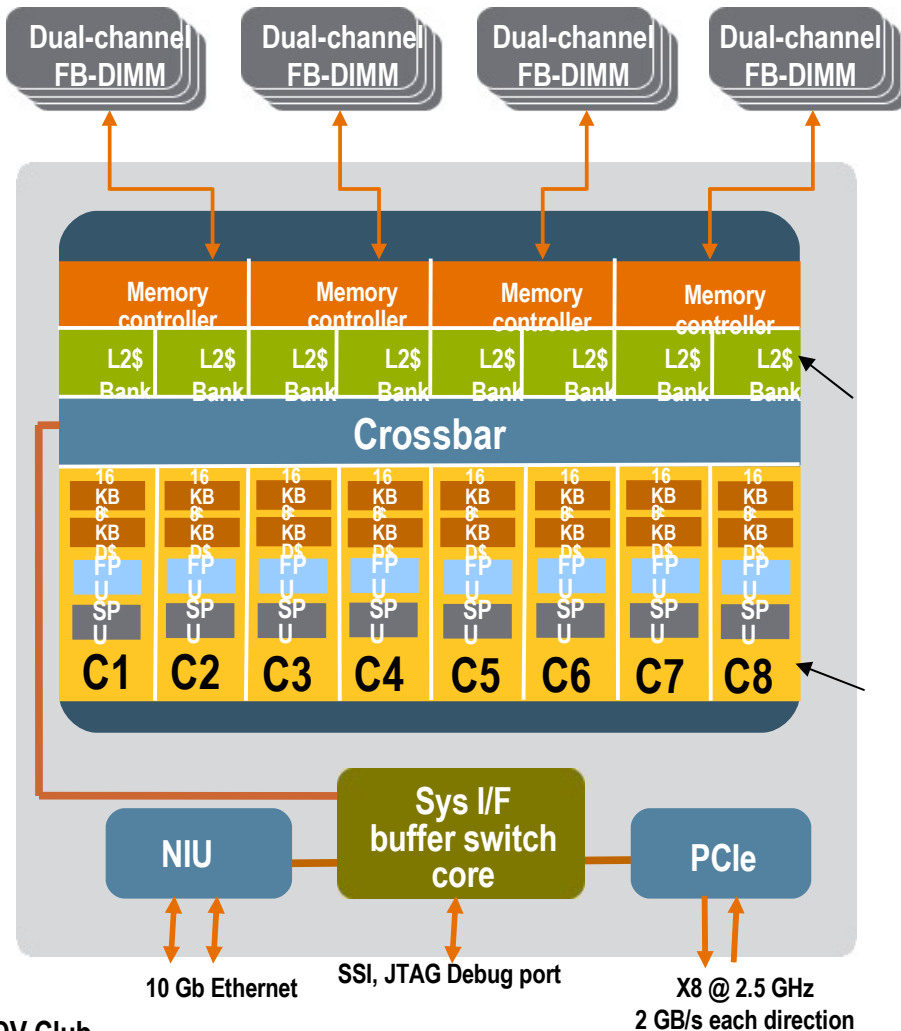
Vendors:

- Enhancements to simplify adoption

Design Challenges Impacting Verification



Server-on-Chip: Verification Complexity



- 2x+ performance over UltraSPARC T1, within the same power envelope
- Up to 8 cores @1.4GHz
- 2x the threads
 - > Up to 64 threads per CPU
- 2x the memory
 - > Up to 128GB memory
 - > Up to 16 full buffered Dimms
 - > 2.5x memory BW = 60+GB/S
- 8x FPUs, 1 fully pipelined floating point unit/core
- 4MB L2\$ (8 banks) 16 way set
- Security co-processor per core
 - > DES, 3DES, AES, RC4, SHA1, SHA256, MD5, RSA to 2096 key, ECC
- Powers SunFire T5120, T5220, T6320 Servers

Problem: cost of Emulation going up

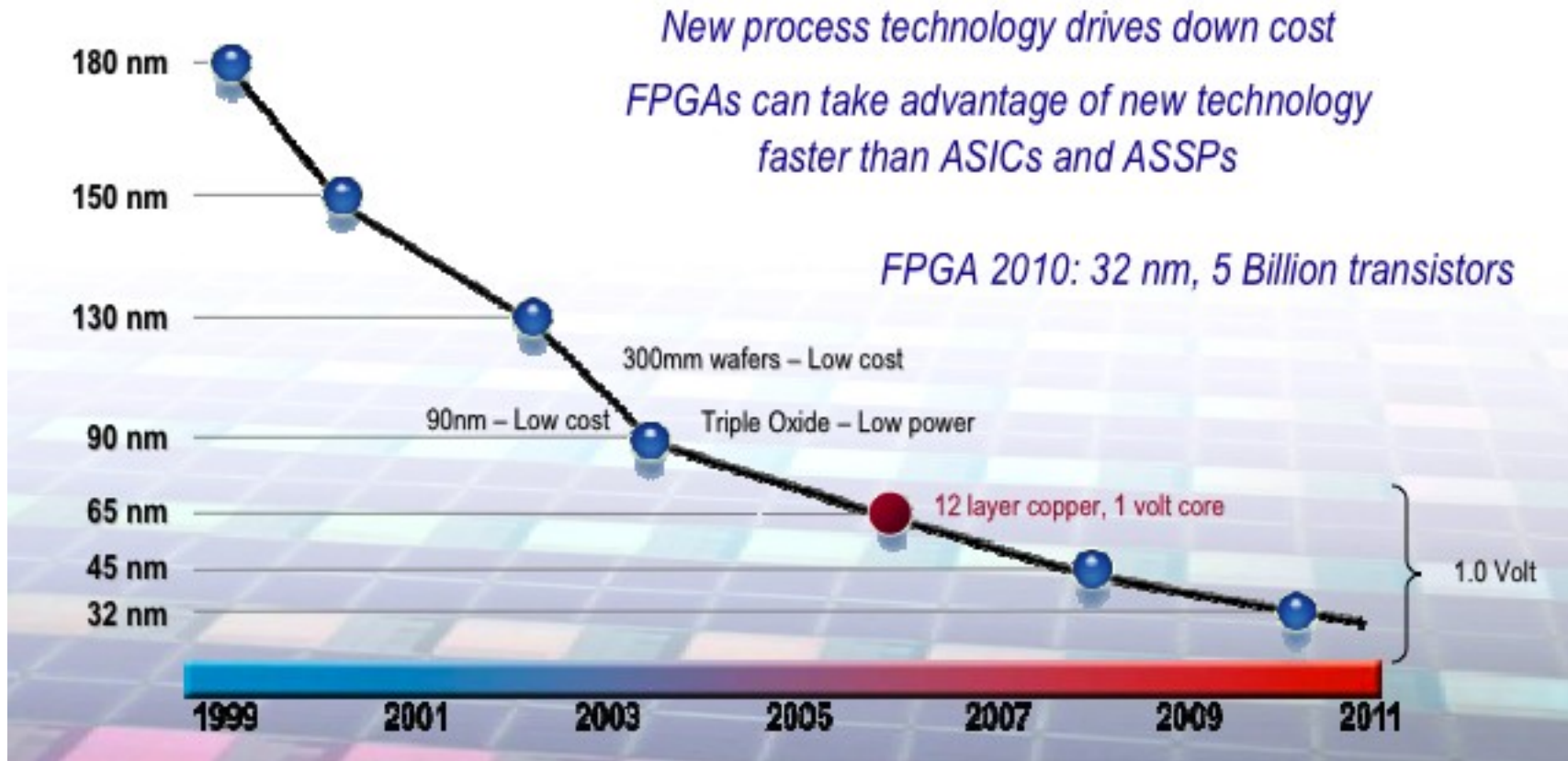


Emulator HW (big iron)



Gulfstream jet

FPGA Roadmap



Source: MPSOC Keynote 2006, Xilinx

FPGAs are getting bigger, cheaper and faster!

Solution: Supplement Emulation with cheaper FPGA prototyping alternatives

- Why use FPGA prototyping?
 - Not enough \$\$ for HW Emulators (big iron) – R&D dollars
 - Need to run at close to real-time speed
 - New advancements in FPGA technology creates opportunity for leverage
- Benefits
 - Availability of standard off-the-shelf, mix-n-match FPGA HW/SW tools (small iron)
 - Allows you to stretch your R&D dollars
 - Deploy many replicates – multiple systems in parallel
 - Supplements your emulators (big iron) – does not replace

Think Small, Fast and Many

FPGA Prototyping 101

What is Prototyping:

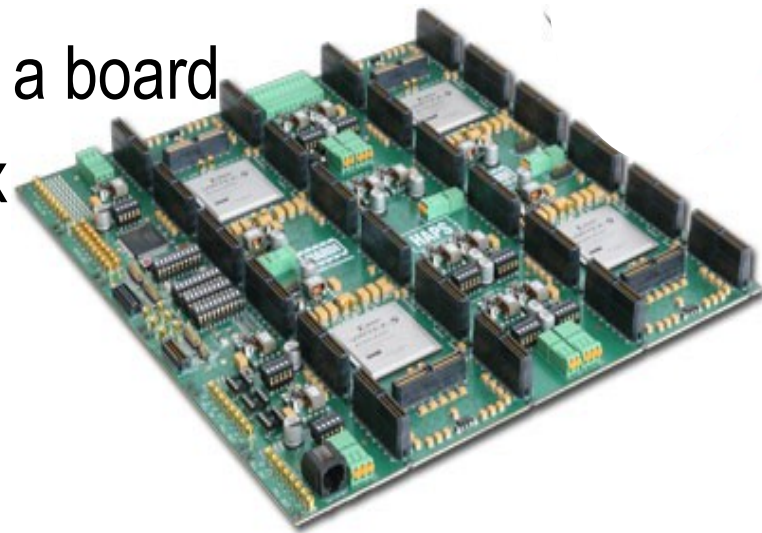
- Process of mapping RTL functionality to FPGAs

Hardware:

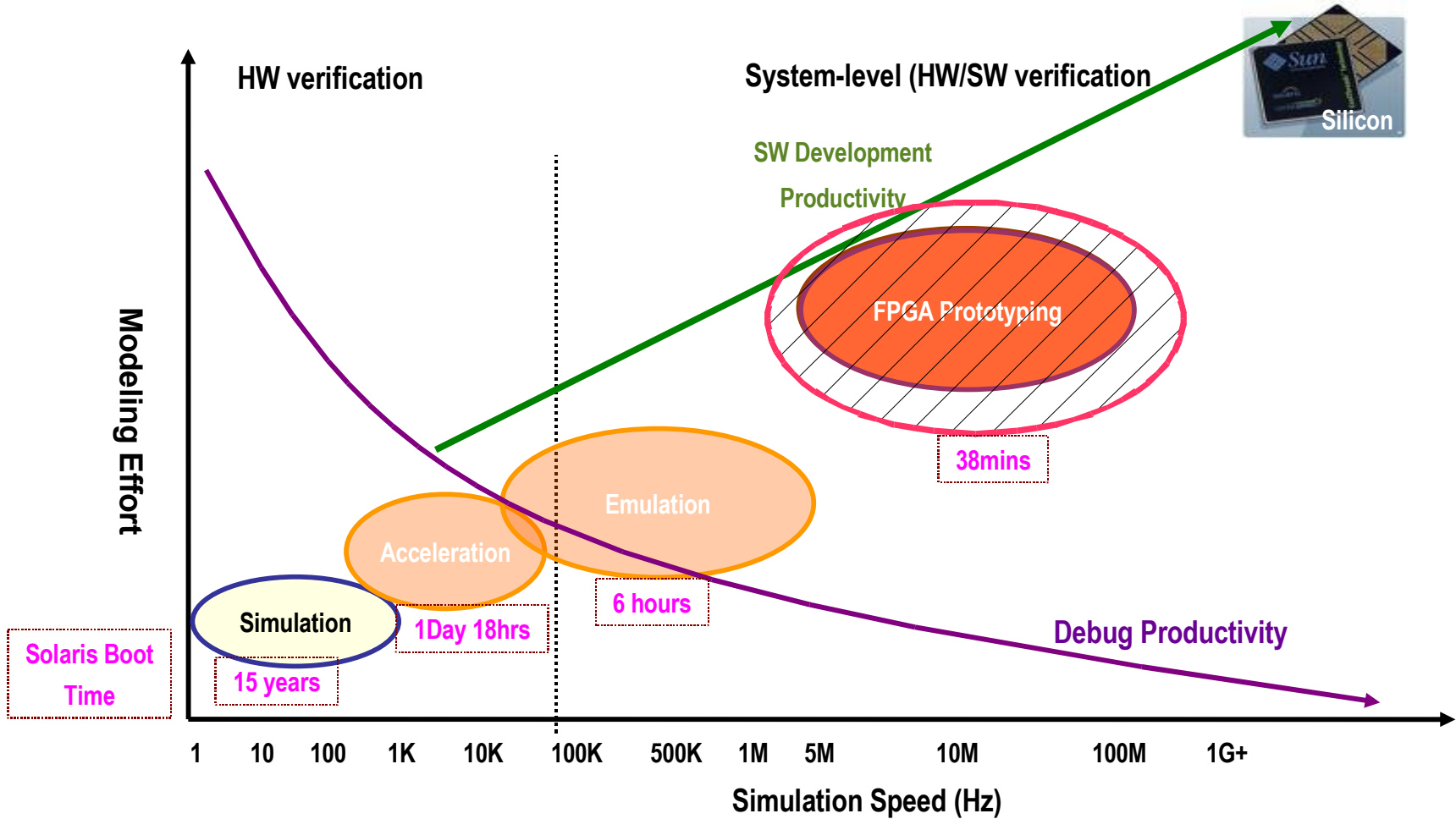
- Multiple Latest, Largest FPGAs on a board
- Two Major Vendors: Altera & Xilinx
- Capacity: 3-150M Gates
- Performance: 5 to 50MHz

Software:

- Synthesis, Design Partition, FPGA P&R
- Debug Tools



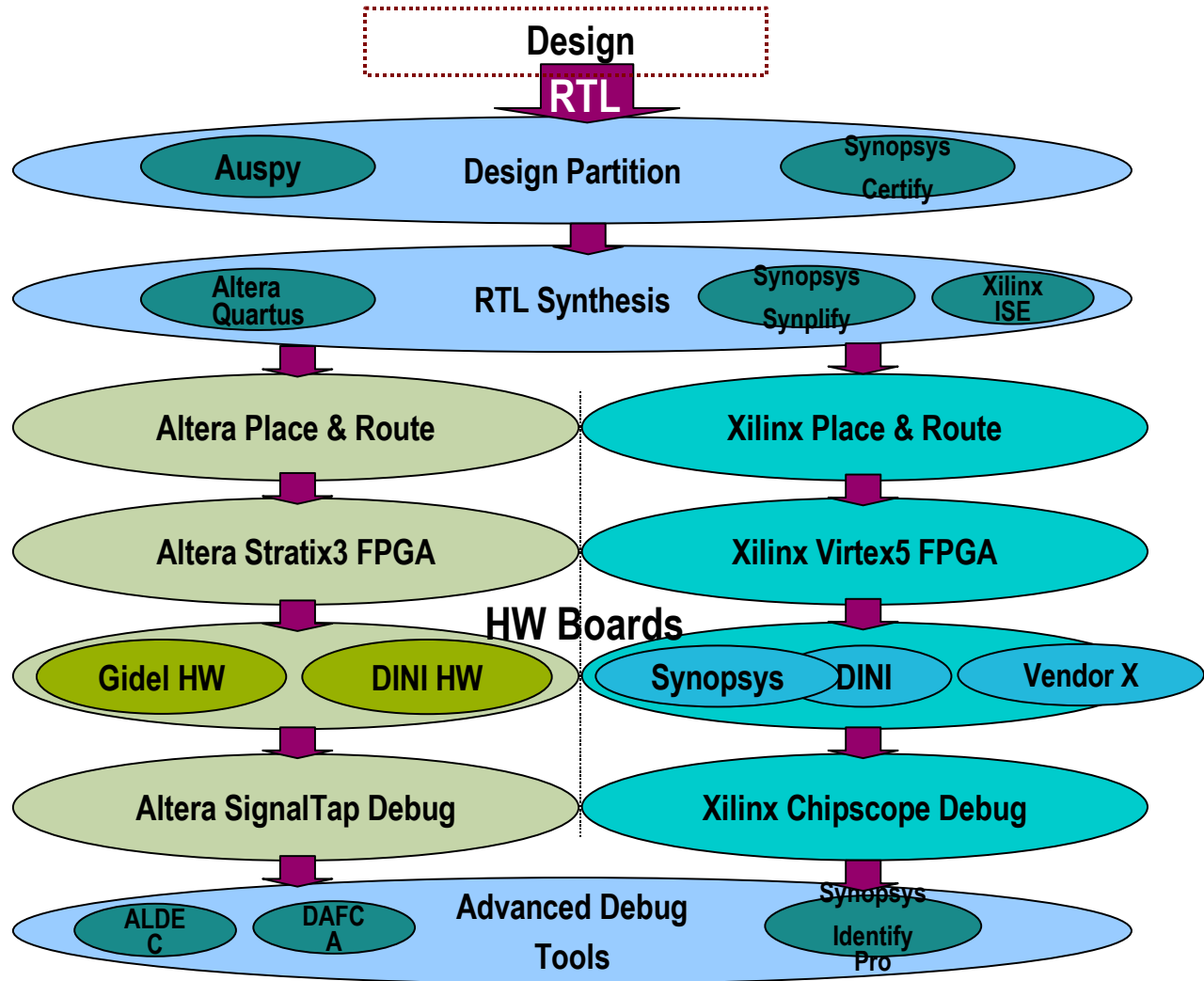
Big Picture



FPGA Prototyping Vs. Emulation

Features	FPGA Prototype	Emulation
General:		
Capacity Expandability	Good	Very Good
Memory Capacity	Very Good	Good
Ease of use	Low	Very Good
Cost	Low	High
Model Build Efficiency:		
Compile Time	OK	Very Good
Model Size	Smaller	Bigger
RTL Flexibility	OK	Good
Test bench support	OK	Very Good
Simulation Efficiency:		
Simulation Speed	Very Good	Good
Save/Restore	No	Very Good
IO Expandability (PCIE, Ethernet etc)	Very Good	Good
Debug Efficiency:		
Signal Visibility	Limited	Very Good
Waveforms w/o re-run	No	Very Good

FPGA Tools



Off-the-Shelf, Mix-n-Match FPGA Emulation HW/SW Tools

Deployment Strategy

- Understand platform capabilities and limitations
 - > Build your use model
 - > Set management, user expectations
- Identify Applicable Model Configurations
 - > Size limited to small capacity (<16MGates)
- Identify Workload
 - > Primary Platform for SW Development
 - > Secondary Platform for RTL/IO Verification
- Design Mapping
 - > Automated FPGA RTL Coding enforcements
- Leverage simulators/emulators for debug

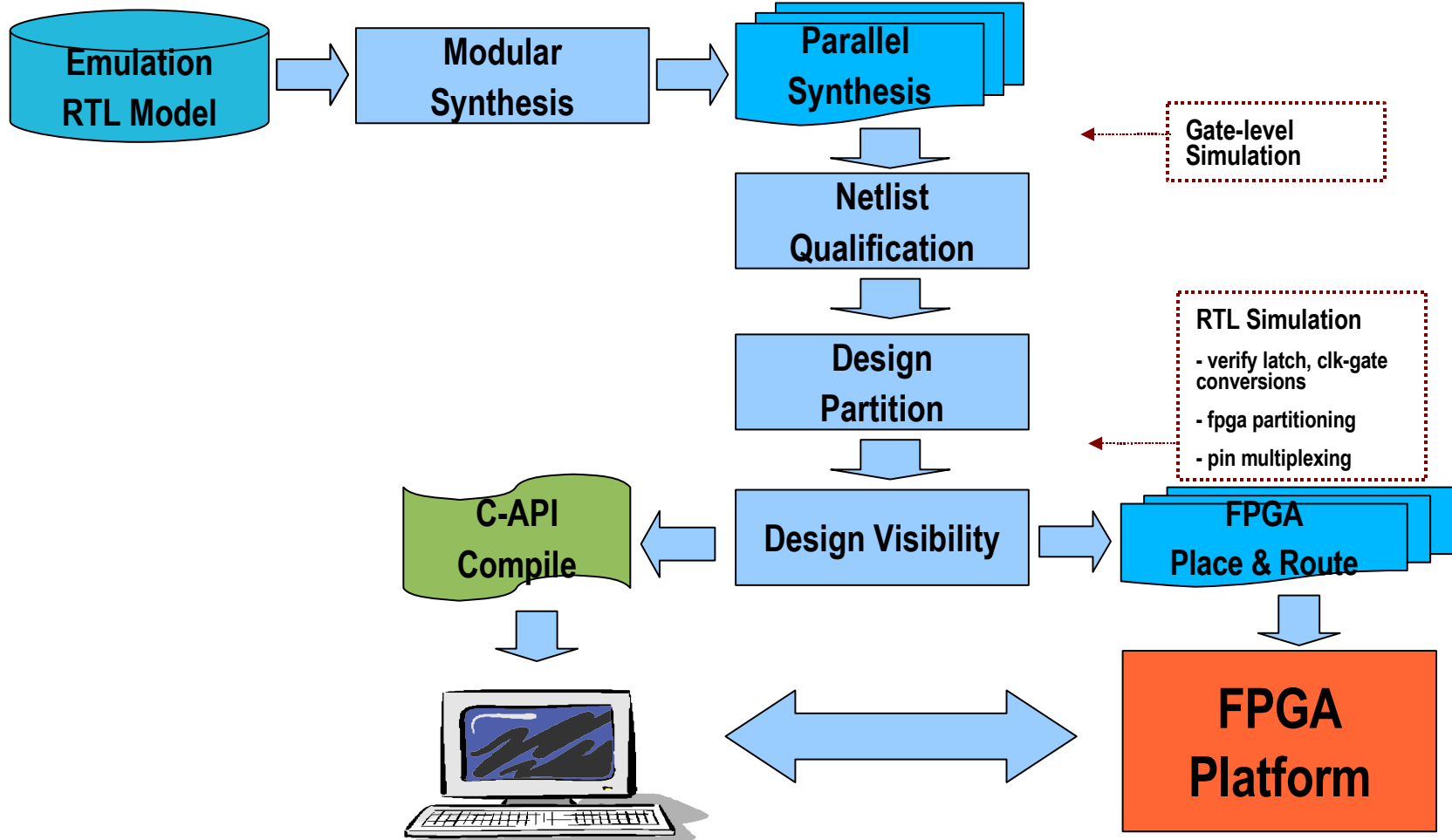
Prototyping Challenges

- Design Mapping – Size, Style
 - > Limit to 4-6 FPGAs (~16M Gates)
- Memory Mapping
 - > RTL Arrays (custom logic) – BLK RAM inferencing
 - > Multi-ported arrays – over clocking
 - > Large system memory - mapping to DDR
- Verification Infrastructure
 - > TestBench – synthesizable, self-checking
 - > Initialization - Use back-door access to download/upload big memories
 - > Monitors, SVA, \$display is not supported – use LA triggers
- Mapping Transformation Verification
 - > Gate-level Simulation at every stage

Guidelines

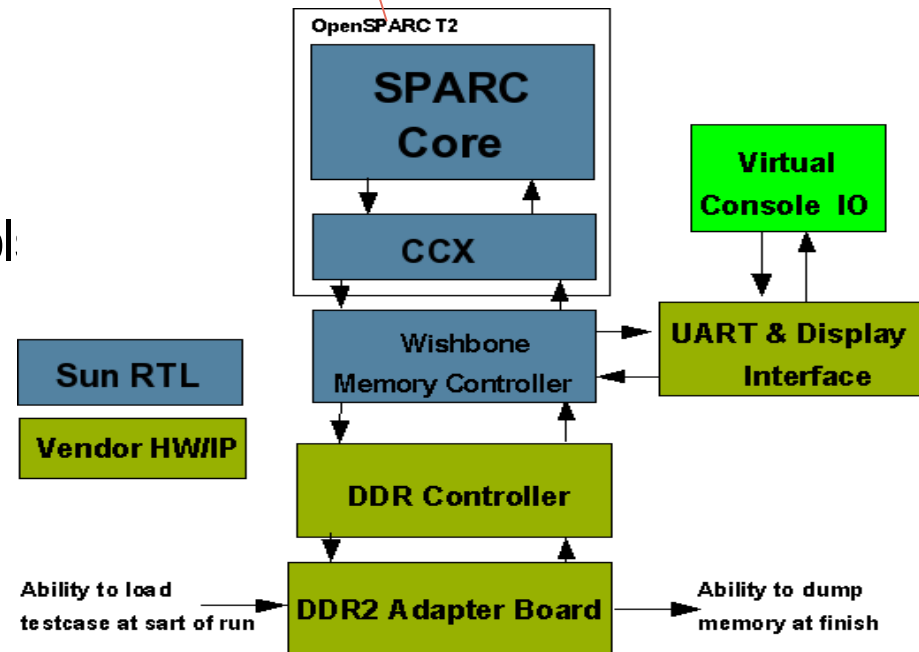
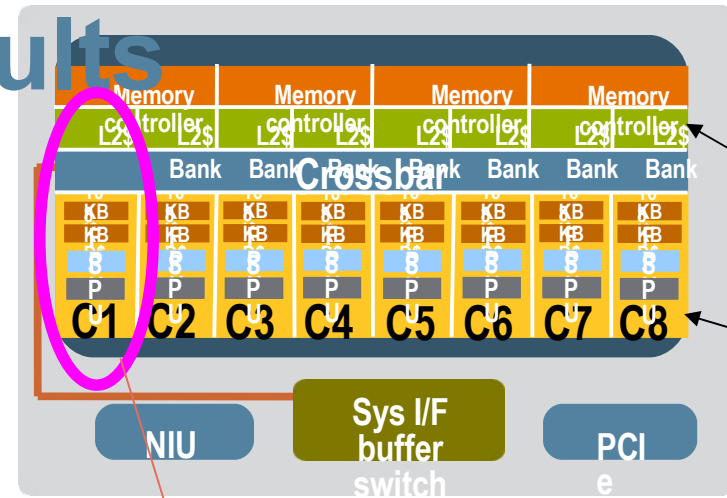
- RTL Coding Guidelines for FPGAs
 - > No XMRs, no force/release, avoid latches, clock gating
 - > No initializations (constant inits results in undesired synth optimizations)
 - > Perform FPGA RTL Linting Check
- Stand-alone Synthesis & Verif of custom logic
 - > check for RAM utilization & reduced CLK domains
 - > Mixed-mode RTL-Gate Simulations
- Perform full-chip gate simulations at different stages
 - > After synthesis, after partitioning, after insertion of signal multiplexing logic

FPGA Flow



FPGA Prototyping Results

- OpenSPARC T2 Model
 - > 3.8M Gates, Runs @8MHz
 - > Being opensourced soon – opensparc.net
- Hardware:
 - > 6M Gates
 - > 2 Altera Stratix III SL340 FPGAS
- Software:
 - > RTL Partitioner, Bundled FPGA tool.
- Effort:
 - > 1 engineer; 3 months
- Applications:
 - > Verify Core, SOC, IO
 - > Verify Firmware (HV/OBP), Solaris, Application



Platform improvements – to ease adoption

- Bridge gap between Emulator and FPGA Prototyping
 - > Learn from advances in the emulator space
 - > Ease of model build
 - > Support for RTL, SVA, TB constructs
 - > Seamless RTL partitioning
 - > Eliminate need for gate-simulations
- Support for Verification infrastructure
 - > XMRs, preserve net names, ports
- Enhance Debug experience
 - > Improve debug tools, offload to simulators

Summary

- Low cost FPGA prototyping supplements expensive emulators
- Collaborate with vendors to implement feature-set for your use models
- FPGA Prototyping is effort-intensive, but will pay off in cost savings & higher performance
- Benefit:
 - > Higher HW & SW coverage (fewer silicon respins)
 - > Debug Bringup Tools before TO (faster bringup; productization time savings)

Leveraging Low-Cost **FPGA Prototyping** for Validation of Highly Threaded Server-on-Chip

DV Club - July 2009

Jai Kumar,
Verification Technologist
Sun Microsystems Inc.
jai.kumar@sun.com
<http://sun.com>