




Digital Sandbox Workshop

Improving Semi-custom Flow


Summer 2003






Outline

- Talk about what we accomplished
- What to do next year
 - Flow improvements (Tom)
 - Design specification and class improvements (Herm)



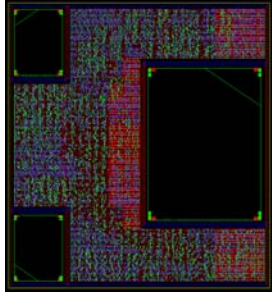

Sandbox Design Experience Course

- Accomplishments
 - Successfully completed Spring Course using NOC/LDPC design
 - Implemented new ASIC Design Flow
 - Incorporated OKI 0.16um Library and Memories
 - Shared server farm
 - Web Conferencing for “common” lectures
 - Web-based Design Reviews



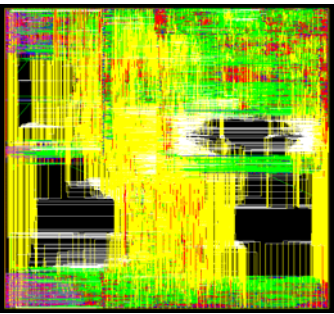

Layout from CMU1 Cnode Team

- OKI 0.16um technology
- 3 OKI memories
- Tool: Silcon Ensemble


Layout from PSU Team

- OKI 0.16um technology
- 5 OKI memories used
- Tool: SE

SDX Class ASIC Design Flow

Foundry	Sandbox or Open/Free	EDA Vendor Tools
Vendor Library Info (OKI Semiconductor)	Project Spec NOC / LDPC	HDL Simulation (ModelSim, Verilog)
	RTL Lab	Synthesis (Synopsys)
	Synthesis Lab Using Synopsys	Place & Route (Cadence)
	Floorplanning Lab for Silcon Ensemble including STA	Timing Analysis (PrimeTime)
		Web CVS



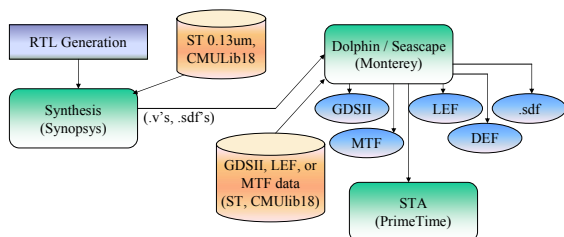
Cadence Experience

- Cadence Si Ensemble:
 - Tool is getting old and becoming less valuable since it cannot effectively handle 0.13um technology and below
 - Students had difficult time learning/using SE
 - Ability to change critical timing paths is difficult and iterative

Monterey Experience

- Monterey:
 - Less ECO interactive
 - Modifying critical timing paths is less iterative
 - Tcl based tool
 - Can effectively handle 0.13um technology
 - Accepts and generates GDSII data
 - This allows us to feed data back to Veruoso for Full Custom design

Proposed Design Flow

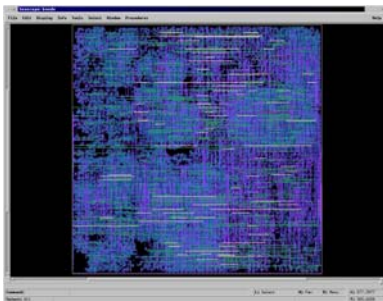


What we're doing to build this

- Student building "Best of SDX" in ST 0.13
- Using Monterey for full chip design

Layout of Bit Node

- ST 0.13um technology
- Cell Area is 592342
- Layout tool: Dolphin

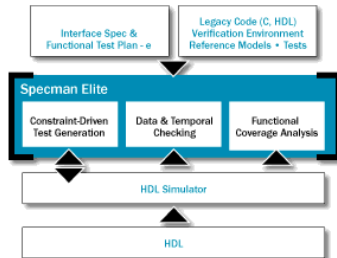


Next Issue: Verification

- We promoted no formal tool for verification
 - Verilog-based testbench authoring
 - Code line coverage
 - Use of OVL assertions in Verilog code
- Penn State used Verisity Specman E
- What tools are out there, results of that experiment

Verity Specman Elite

- Constraint-driven test generation
- Data and temporal checking
- Functional coverage analysis
- HW/SW co-verification support
- Supports all major Verilog and VHDL simulators
- Uses E language



Synopsys Vera

- Easy to learn
- Next Gen. Constraint solver finds bugs quickly
- Formal analysis engine
- Supports re-use across projects, geographies, and teams
- Uses VHDL, Verilog, and SystemC
- Access to OpenVera Verification IP

Open Vera

- OpenVera is an open hardware language
- Seamless tool integration and open distribution of verification IP.
- Goal is to establish a hardware verification language (HVL), called OpenVera™.
- Easy to learn
 - Combines the strengths of HDLs, C++ and Java
 - Additional constructs for functional verification making it ideal for developing testbenches
- OpenVera accelerates verification by providing high-level constructs designed for complex SoCs.
- Designers create testbenches using OpenVera and EDA vendors create tools that are interoperable.

SystemVerilog

- Language structures for assertions
- Not yet accepted into tools...

Discussion

- Functional Verification tools:
 - Good tools?
 - Falsely perceived silver bullets?
- Satisfied with Verity for Verification Class?

New Libraries

- Plan to using CMU18Lib
- One technology library across the curriculum
 - Full Views
 - DRC, LVS, etc
- Can be fabricated with MOSIS on TSMC

Problems/Issues with Libraries

- Memory Generators!
 - Nobody gives these out
 - Need: LEF, Liberty
 - Like to have: GDSII (unlikely)
 - May enlist undergraduates
- IOs / PLLs / Passives



SDX Class Improvements

- How did this class work?
 - Network-on-Chip
 - Team organization
- What could have been better?
- What should be the design objective this year?



Requirements for Success

- A **Firm** Specification of Design
- Something that gets students excited
- A Passionate "Consumer"
 - PipeRench was successful
 - picoJava was not
 - I think LDPC on NoC was successful



Industry Challenges to SoC Design

- Divide and conquer (hierarchy)
 - Reduce the atomic problem size
 - Improve the correspondence of wireload models
 - Allow multiple teams to work independently
 - Reduce time to market
- Design Re-use
- Decoupling of efforts
 - Specified Interfaces
 - Clear Match of Responsibilities and teams



Industry < Education

- Design Productivity is in crisis
 - Issues are even worse in academia
- Semester less than product design cycle
 - Reality: less than a semester
 - Reality: less than full-time
- Divide and conquer is even more important
 - Senioritis
 - Can't adjust if team doesn't have right skills
 - Failure is an option, but we still have to give a grade

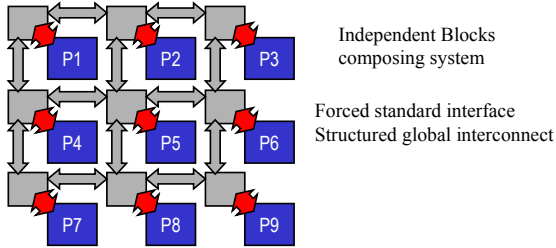


Network on a Chip Design

- Hot and Hip Topic in Industry
- Great for education
 - Highly decoupled design
 - Structured interfaces
 - If some team blows it off, still can evaluate others
- Good results:
 - Most teams successfully tested their component
 - Two of four chip designs passed some whole chip test



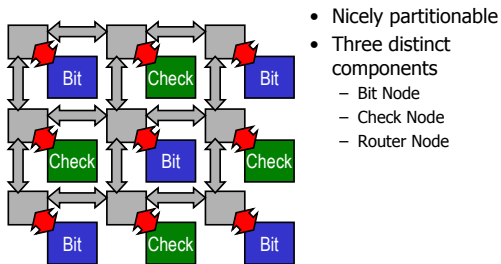
NoC Methodology



Spring 2003

- Taught Course at CMU, Pitt and PSU
 - Ivan Kourtev (Pitt)
 - Vijay Narayanan (Penn State)
- System spec: LDPC Decoder
 - Good fit for NoC Methodology
 - Exciting research topic

LDPC on NoC



Team Organization

- Chip Team (given one spec)
 - 4 Sub-Teams
 - Each sub-team builds different component
 - Architecture team simulates, determines system performance, and does system verification

	CMU1	CMU2	PITT	PSU
Bit				
Check	Compete			
Route	Route			
Arch				

CMU Results

- We had one flunky subteam
 - Bad personality mix
 - Two hard workers, two bums
 - Their problems didn't poison the well
- Good load balancing
 - Arch teams worked hard on verification
- My best Faculty Course Evaluation ever

Class Organization

- Right size design teams?
- Expected enrollment the same?
- Design review structure?

New Class Design

- Re-use NoC Methodology?
- Could re-use the existing design specs?
 - Cheating? / Novelty?
- Substantially Modified Spec?
 - Change the network topology?
 - Make it substantially bigger?
 - Make the processors more general purpose?
 - Ideas from research?
- Could attempt to develop a new app in NoC
 - A lot of effort



Class Logistics

- Conference Call Phone was not ideal
 - Students are not aware of mikes
 - Repeated questions... repeated answers...
 - We are planning to upgrade our classroom
- Other conferencing issues?



Class To Do List

- Schedules:
 - Do we have the same issues this year?
- Verilog:
 - Pitt and PSU get VHDL, learning Verilog was hard
 - Better labs
- Planning session in October...

