Generating/Programming/Validating the Uncore

- This is the 1-minute advertisement slide.
- Insert interesting graphic(s) here with minimal text!
- Change title to your project title as it is listed on the webpage
- Enter your project theme and task ID in the line at the bottom of this page
- This slide will correspond to your poster preview slide

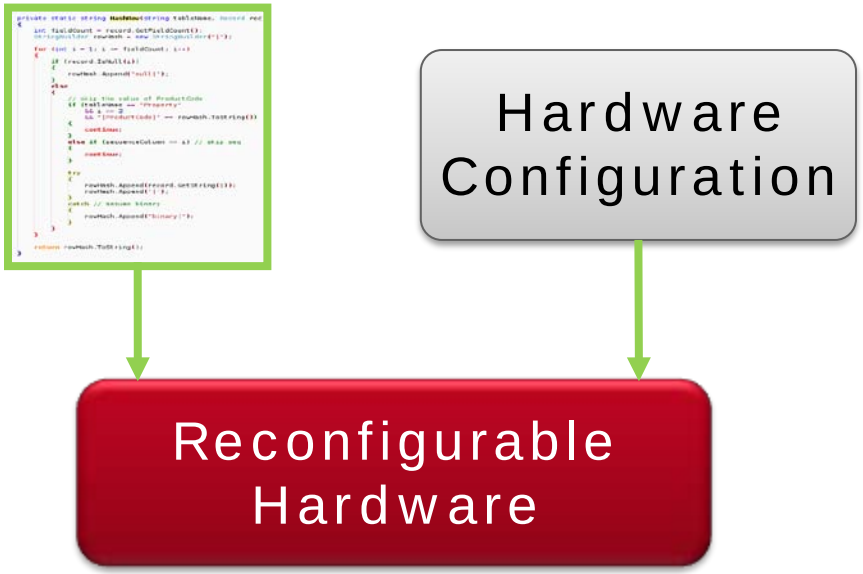
Abstract

- **Technical focus**
  - The lockstep scaling of cost, performance and energy has broken down. To improve performance under power constrains, we need to employ customization techniques.
  - Even though ASIC designs can efficiently provide good performance, they suffer from very high NRE costs.
  - In this work we develop a framework for automatic generation and customization of chip multi-processors. We develop new design methodologies and tools that incorporate optimization techniques, as well as tackle custom design & verification issues
- **Results**
  - We demonstrate the design and use of *Genesis2*, a tool for constructing and using Generators.
  - We investigate how architectural flexibility interacts with verification. In particular, we show that configurability can help the verification of generated instances of a design.

Challenges of Custom Designs

- **Must learn the design space**
  - System level simulator
  - Design exploration/optimization tools
- **Must create complete eco-systems**
  - Validation infrastructure
  - Software support (compiler, linker, runtime/driver)
- **Once exploration and tooling are done**
  - Freeze the architecture
  - Record the design

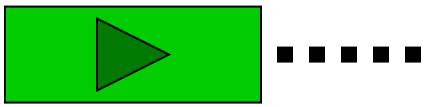
- **Current Solution: Reconfigurable Designs**
  - Application designers program the silicon substrate
  - Benefit: Amortize hardware development costs

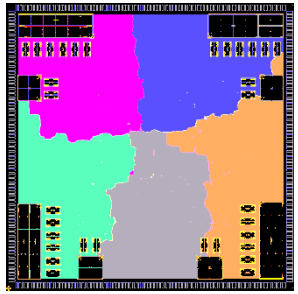


Chip Generator Idea

Instead of a reconfigurable chip, use a "virtually reconfigurable" chip, an optimization toolset, and a generate button…

- **Phase 1:**
  - Application developer "programs" the flexible framework
  - Makes his application meet the desired spec
- **Phase 2:**
  - The system "compiles" the configuration
  - A chip is automatically generated
  - The chip is tailored toward the desired application





Ready for fab.
chip database

Using a Generator: Phase 1

Architecture Parameters (#cpu, cache size, etc.)

Energy Budget

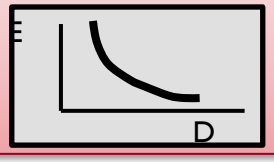
Rewrite Code

Simulator

Hardware Optimization

Adjust Hardware

Tradeoff Lib



Performance

Power Usage

Using a Generator: Phase 2

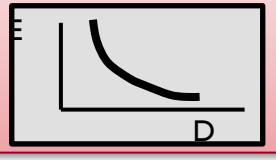
Architecture Parameters (#cpu, cache size, etc.)

Energy Budget

Simulator

Hardware Optimization & Generation

Tradeoff Lib

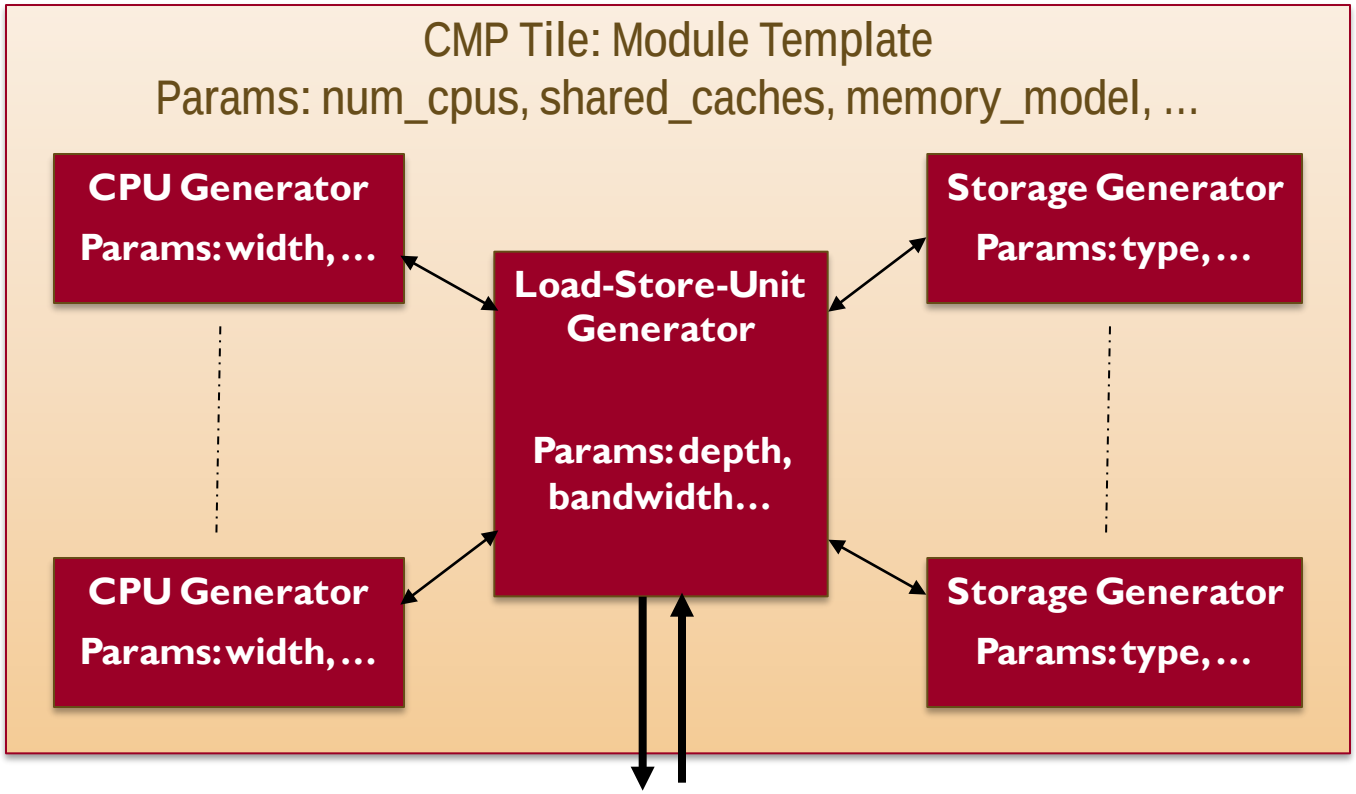


Verif. Collateral

RTL Code

Application Designer Sees a Simulator-Like Framework

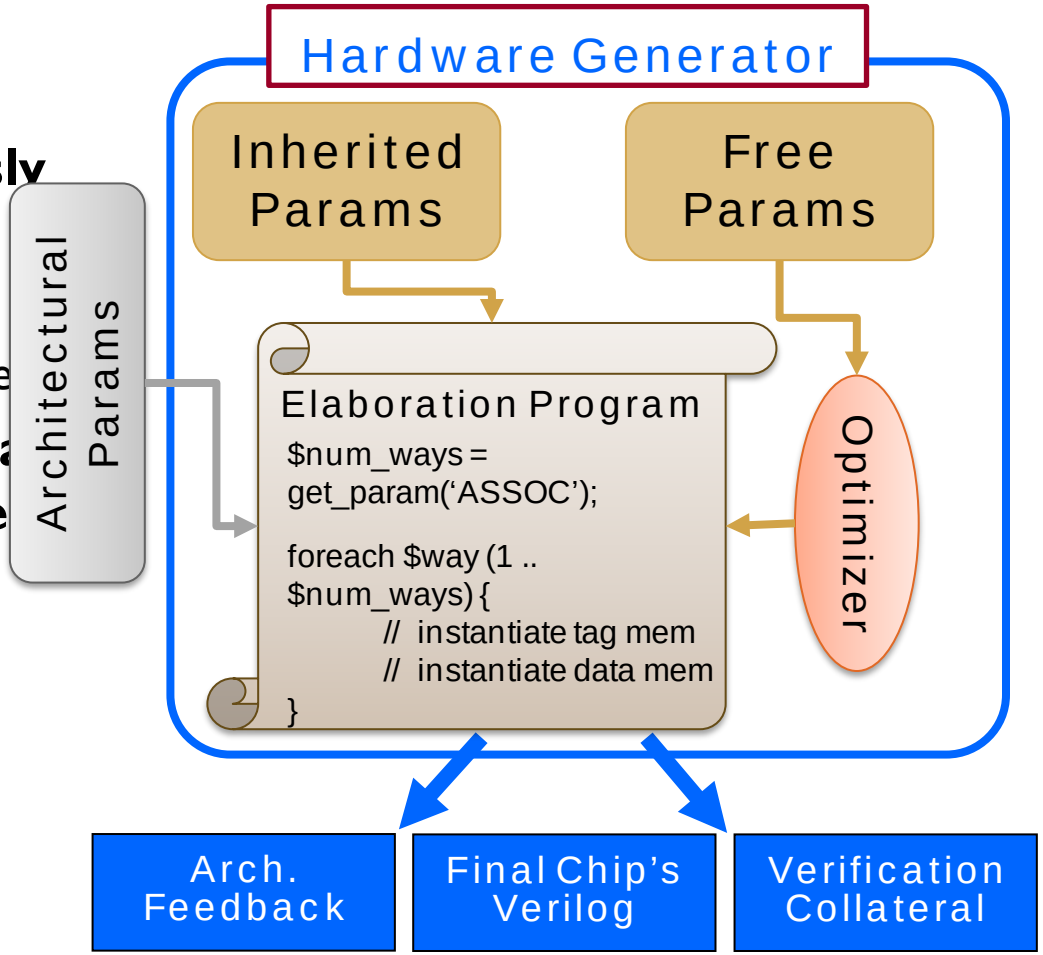
- **Fixed interconnection of flexible modules**
  - A design template / blueprint
- **<u>Not</u> a flexible configuration of fixed modules**



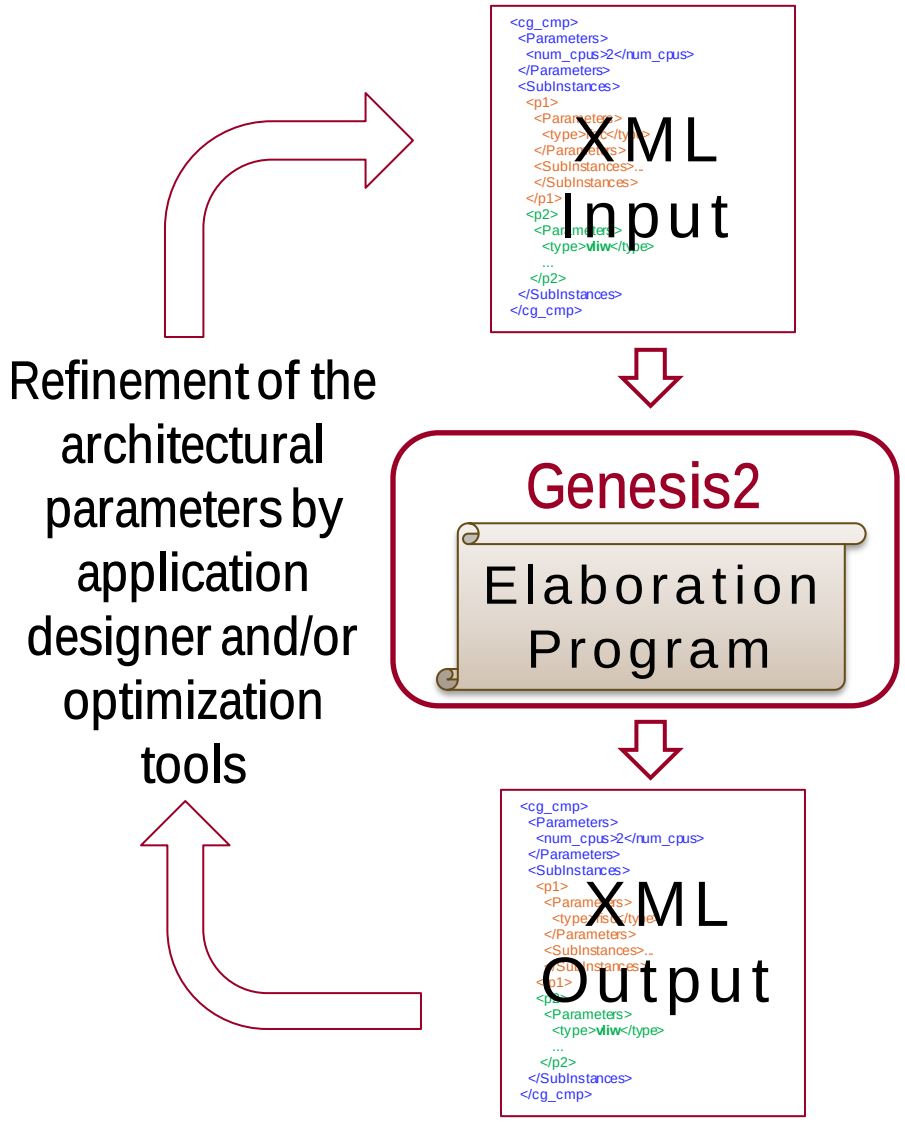

```
Architectural
Program
Tile 0:
  CPU0:
    width = 4
  CPU1:  ...
  Storage0:
    type = D-
Scratch
    Owner = CPU0
  Storage1:
    type = I-Cache
    owner = Shared
  LSU:  ...
```

Genesis2: A Tools for Creating Generators

- **Removes artificial limitations from elaboration**
- **Enables designers to code in two languages simultaneously**
  - One that describes the hardware/logic
  - One that decides what hardware
- **Instead of coding a module, a designer codes the template that produces that module**
- **Template can incorporate parameters from many sources**



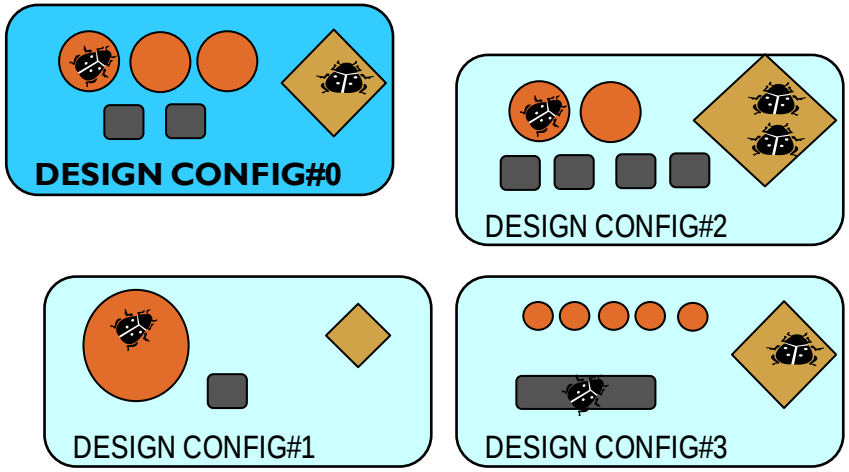
Genesis2: Allows Iterative User Input



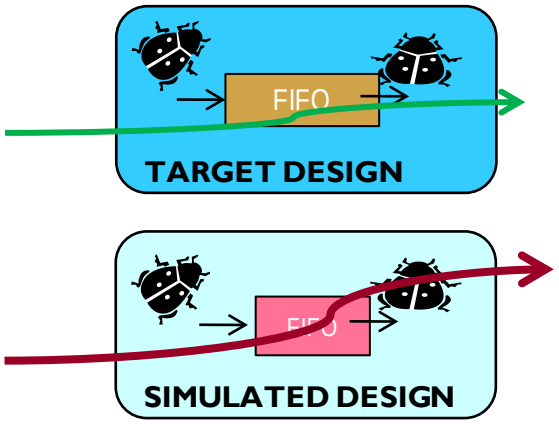
- **Completely implemented and functional**
- **Currently in use for:**
  - Designing the first ever chip multiprocessor generator
  - Assisting in the design of mixed signal chips
  - Used by Stanford's "Design Projects in VLSI Systems" class
- **In development:**
  - Interactive GUI based front end
  - CMP memory-system protocol design framework front end

Using Configurability to Aid Verification

- **Does configurability worsen the verification problem?**
  - Increased flexibility: more verification space?
  - Key insight: only verify INSTANCE(s) of the generator

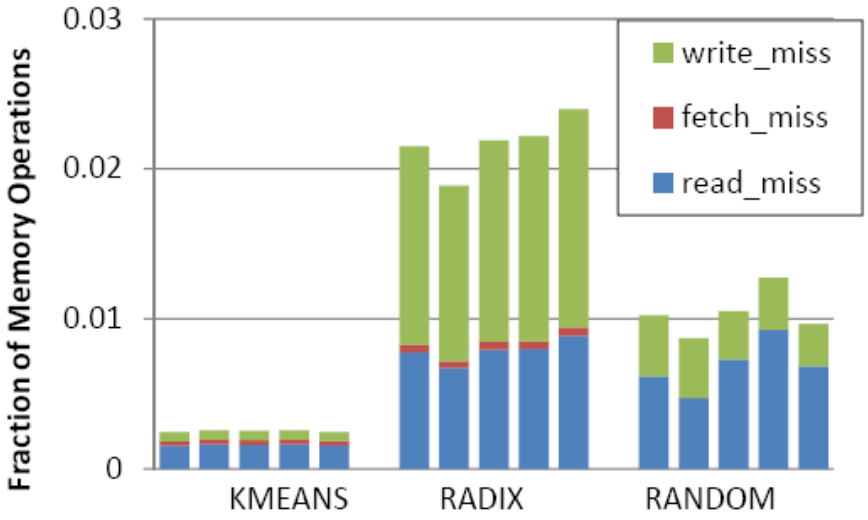


- **Extend common practice in verification**
  - Artificially resizing resources (like FIFOs) to induce corner cases
  - Changing configuration of one module can expose bugs in another

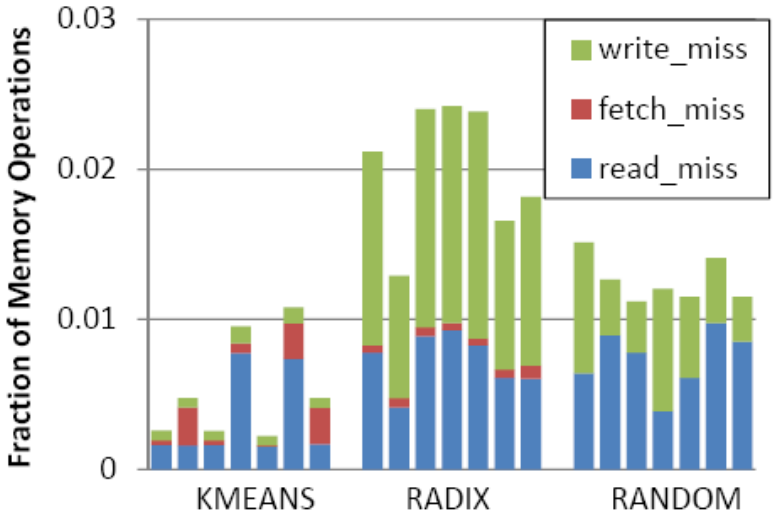


Configuration's Effect on Coverage

- Random configuration is a force multiplier for existing tests
- Changing configuration requires less effort than adding a test



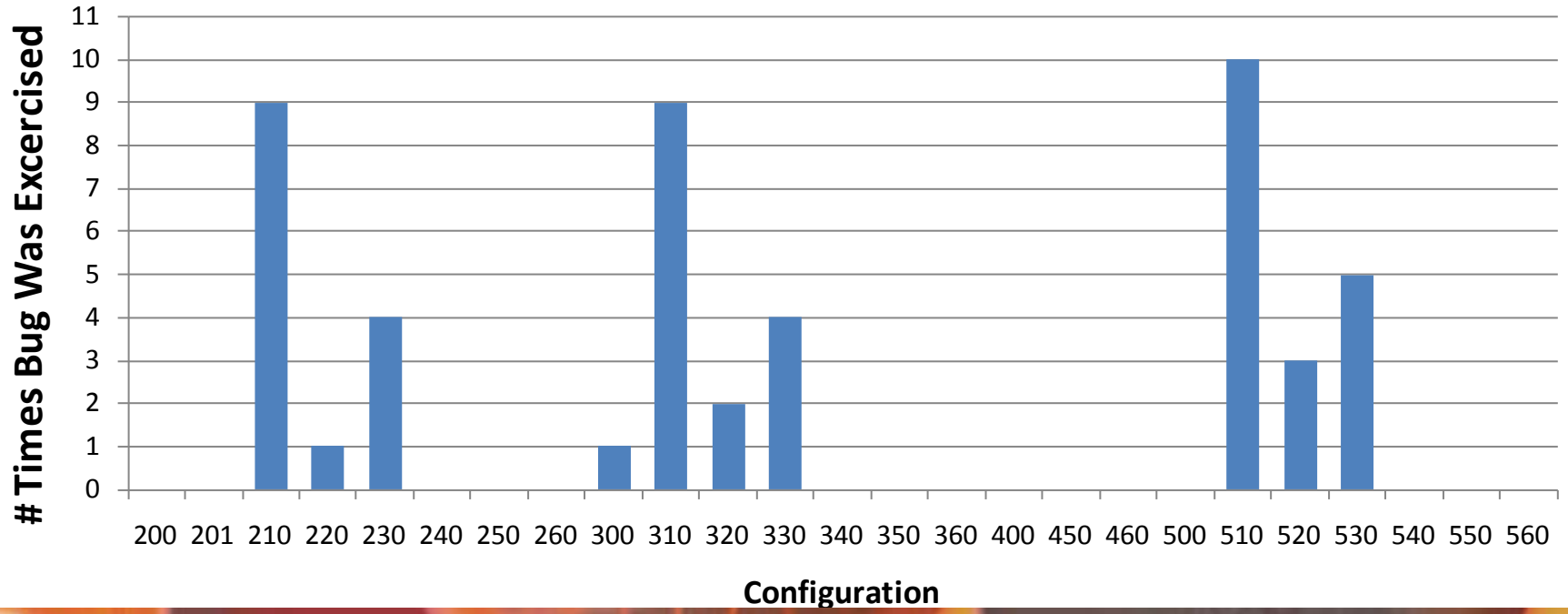
Vary Test & Seed Only



Vary Test, Seed & Configuration

Configuration's Effect on Finding Bugs

- **Example: Real Bug #458 on taped-out Smart Memories Chip**
  - originally discovered by a random test :10,000 loads, stores, and prefetches
  - transactions randomized by operation, address, value, and delay
- **Error: One submodule couldn't handle back-to-back messages**
- **Ran test 6250 times (250 runs x 25 configurations)**



Summary & References

Summary

- We present an approach to reducing the high NRE cost of chip design
- We have created a new tool, Genesis2, for designing and using generators.
- New methodologies to tackle verification issues

Related References

- O. Shacham, O. Azizi, M. Wachs, S. Richardson, M. Horowitz. "Why Design Must Change: Rethinking Digital Design". *Micro IEEE,* Vol. PP-99, Sept 2010.
- R. Hameed, W. Qadeer, M. Wachs, O. Azizi, A. Solomatnikov, B.C. Lee, S. Richardson, C. Kozyrakis, M. Horowitz. "Understanding Sources of Inefficiency in General-Purpose Chips." *ISCA-37* , June 2010.
- A. Solomatnikov, A. Firoozshahian, O. Shacham, Z. Asgar, M. Wachs, W. Qadeer, S. Richardson, M. Horowitz. "Using a Reconfigurable Processor Generator for Computer Architecture Prototyping." *MICRO-42*, December 2009.
- A. Firoozshahian, A. Solomatnikov, O. Shacham, Z. Asgar, S. Richardson, C.Kozyrakis, M. Horowitz. "A Memory System Design Framework: Creating SmartMemories" *ISCA'-36*, June 2009
- O. Shacham, Z. Asgar, H. Chen, A. Firoozshahian, R. Hameed, C. Kozyrakis,W. Qadeer, S. Richardson, A. Solomatnikov, D. Stark, M. Wachs, M.Horowitz. "Smart Memories Polymorphic Chip Multiprocessor." Winner of the46th DAC/ISSCC Student Design Contest. *DAC'09*, July 2009.
- O. Shacham, M. Wachs, A. Solomatnikov, A. Firoozshahian, S. Richardson, M.Horowitz. "Verification of Chip Multiprocessor Memory Systems Using A Relaxed Scoreboard." *MICRO-41*, November 2008.
- A. Solomatnikov, A. Firoozshahian, W. Qadeer, O. Shacham, K. Kelley, Z. Asgar, M. Wachs, R. Hameed, M. Horowitz. "Chip Multi-Processor Generator," *DAC '07.* June 2007