

# Taint Analysis of Microarchitecture Synthesis Software

Not Using It to Secure Microarchitectures is Foolish

Zhiyang Ong

Department of Electrical and Computer Engineering  
Dwight Look College of Engineering,  
Texas A&M University  
College Station, TX

September 26, 2016



# Background

Microarchitecture synthesis software is used to generate cycle-accurate/RTL designs of microarchitectures.

Taint analysis has been used to determine security loopholes of software.

Project Goal: Apply taint analysis to determine the security loopholes of the microarchitecture synthesis software.

Questions:

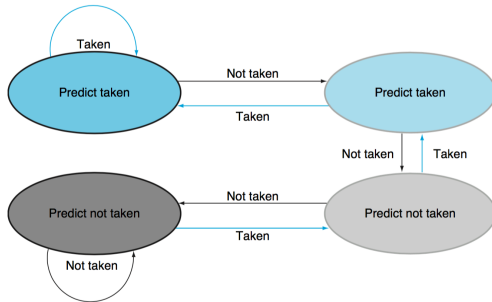
- 1 Can I force a finite state machine (FSM) to enter a non-existent/invalid state?
- 2 Can I force the processor to write back to memory out of order?
- 3 Can I degrade the performance of many-core processors?



# Timeline of Milestones

- Benchmarks (now):
  - ➊ Rocket chip: <https://github.com/ucb-bar/rocket-chip>
  - ➋ OpenSoCFabric: <https://github.com/ucb-bar/OpenSoCFabric>
  - ➌ chisel-torture: <https://github.com/ucb-bar/chisel-torture>
- Instruction set architecture: RISC-V ISA
- Learning Chisel: Early October
- firrtl parser: Mid October
- Taint analysis:
  - ➊ Late October - Early November
  - ➋ Multi-cycle processor's control logic
  - ➌ Pipelined processor's control logic
  - ➍ Branch prediction FSM
  - ➎ Cache coherence FSM
- Design for Testability (DFT) Synthesis: Late November
- More taint analysis on DFT components: Late November
- Regression testing: Early December

# Pictures



**FIGURE 4.63 The states in a 2-bit prediction scheme.** By using 2 bits rather than 1, a branch that strongly favors taken or not taken—as many branches do—will be mispredicted only once. The 2 bits are used to encode the four states in the system. The 2-bit scheme is a general instance of a counter-based predictor, which is incremented when the prediction is accurate and decremented otherwise, and uses the midpoint of its range as the division between taken and not taken.

Figure: Branch prediction FSM

# Pictures

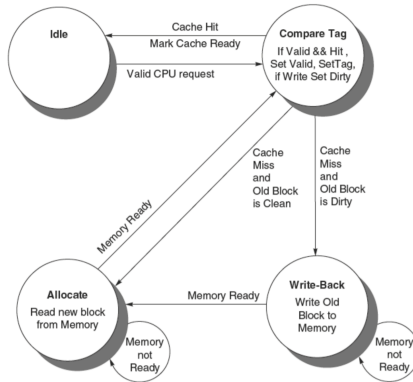


FIGURE 5.40 Four states of the simple controller.

Figure: Cache coherence FSM

# Pictures

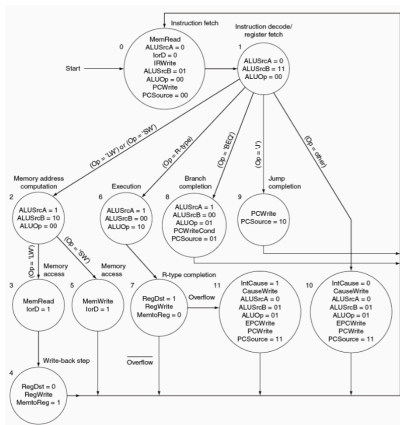


Figure: FSM of multi-cycle processor's control logic