

# Connecting Pre-silicon and Post-silicon Verification

**Sandip Ray and Warren A. Hunt, Jr.**

Department of Computer Sciences  
University of Texas at Austin

[{sandip, hunt}@cs.utexas.edu](mailto:{sandip, hunt}@cs.utexas.edu)

<http://www.cs.utexas.edu/users/{sandip, hunt}>

# Motivation

Formal analysis has shown promise in increasing reliability of computing systems.

- Can catch “high quality” bugs that are difficult to hit during simulation.
- Has been successfully applied to some industrial design components.
  - FP execution units
  - Control logic for out-of-order pipelines

**But formal analysis has primarily been restricted to pre-silicon**

- Typical targets are RTL models and netlists.
- Almost no connection with post-silicon verification.

**How do we make use of formal analysis to facilitate post-silicon design verification?**

# Post-silicon Verification

Post-silicon verification is the use of pre-production, physical circuits to determine logical bugs.

- Simulation speed may be 1,000,000,000 times faster than pre-silicon.
- Facilitates exploration of **very deep** states.

# Post-silicon Verification

**Post-silicon verification is the use of pre-production, physical circuits to determine logical bugs.**

- Simulation speed may be 1,000,000,000 times faster than pre-silicon.
- Facilitates exploration of **very deep** states.

## **BUT**

- Control is limited.
- Observability is **extremely limited**.

### **Factors limiting observability:**

- Limited number of pins
- Cost of additional DFD logic.
- ...

# Post-silicon Verification

**Post-silicon verification is the use of pre-production, physical circuits to determine logical bugs.**

- Simulation speed may be 1,000,000,000 times faster than pre-silicon.
- Facilitates exploration of **very deep** states.

## **BUT**

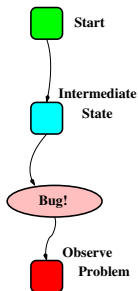
- Control is limited.
- Observability is **extremely limited**.

### **Factors limiting observability:**

- Limited number of pins
- Cost of additional DFD logic.
- ...

**Post-silicon verification is extremely expensive and tedious.**

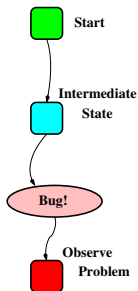
# Post-silicon Debug Process



- Start in a known state
- Quickly get to a *deep* state
- Continue until a bug occurs
  - Bug is unobserved
  - Bug may lay dormant
- Finally, observe a problem

**It can take substantial effort to find and fix a bug.**

# Post-silicon Debug Process



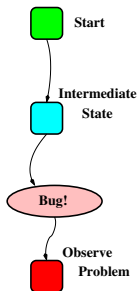
- Start in a known state
- Quickly get to a *deep* state
- Continue until a bug occurs
  - Bug is unobserved
  - Bug may lay dormant
- Finally, observe a problem

**It can take substantial effort to find and fix a bug.**

**Typical Approach:** Add extra hardware “hook” to improve observability.

- But the hooks are added on-demand without analysis of design invariants.
- Once added, they are carried over from one design to next.

# Post-silicon Debug Process



- Start in a known state
- Quickly get to a *deep* state
- Continue until a bug occurs
  - Bug is unobserved
  - Bug may lay dormant
- Finally, observe a problem

**It can take substantial effort to find and fix a bug.**

**Typical Approach:** Add extra hardware “hook” to improve observability.

- But the hooks are added on-demand without analysis of design invariants.
- Once added, they are carried over from one design to next.

**A more disciplined process of on-chip instrumentation is necessary.**



# Our Goal

**Facilitate post-silicon verification by pre-silicon analysis.**

# Our Goal

**Facilitate post-silicon verification by pre-silicon analysis.**

## Pre-silicon Models

- Allow complete visibility of internal state.
- Can be formally analyzed and reasoned about.

# Our Goal

**Facilitate post-silicon verification by pre-silicon analysis.**

## Pre-silicon Models

- Allow complete visibility of internal state.
- Can be formally analyzed and reasoned about.

**We use pre-silicon analysis to determine post-silicon observation points.**

- Exploit the connection between pre- and post- silicon models.
- The number of observation points depends on the desired logical guarantee

# Our Goal

**Facilitate post-silicon verification by pre-silicon analysis.**

## Pre-silicon Models

- Allow complete visibility of internal state.
- Can be formally analyzed and reasoned about.

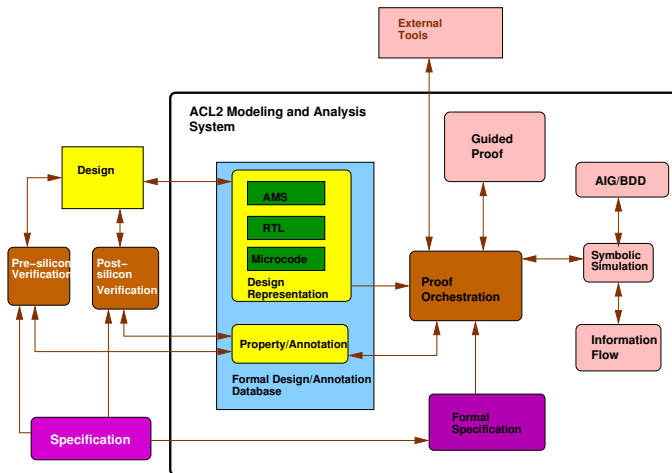
**We use pre-silicon analysis to determine post-silicon observation points.**

- Exploit the connection between pre- and post- silicon models.
- The number of observation points depends on the desired logical guarantee

**Eventual goal is a post-silicon verification methodology that**

- provides high correctness assurance.
- helps comprehend post-silicon execution results.
- provides clear trade-offs between logical guarantees and DFD support.

# Overall Vision



**We envision a single, unified, formal framework for specification, evaluation, and verification of computing systems.**

# An Approach: Partition Trace Analysis

**Partition post-silicon trace analysis into two components.**

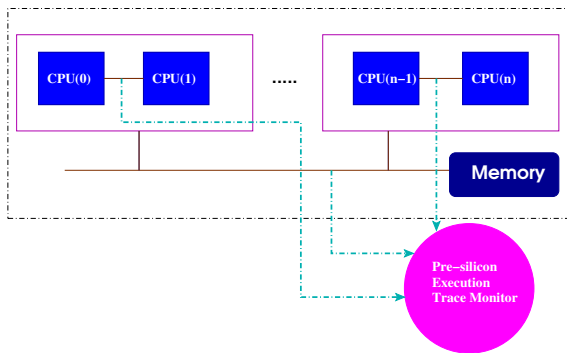
- small on-chip **integrity unit** that has full observability
- an off-chip **partial trace analyzer**

The off-chip component can assume that in-silicon analysis has succeeded.

**Formal analysis guarantees that the components together are equivalent to a monitor that has full observability.**

**We applied the partitioning approach for post-silicon analysis of a multiprocessor memory system.**

# A Multiprocessor Memory System

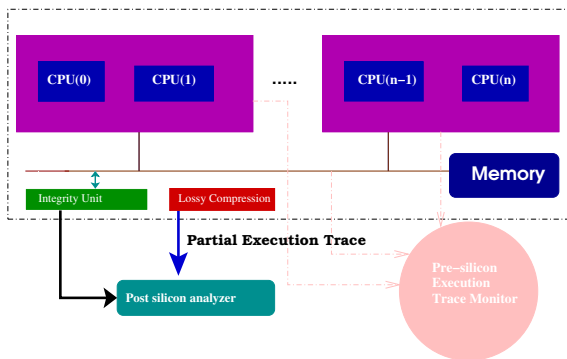


**The pre-silicon monitor checks for bounded coherence.**

- Has **full observability** of all bus transactions.
- Obviously impractical for post-silicon.

# Post-silicon Analysis

A post-silicon trace is a subsequence of a pre-silicon trace with lossy compression.



- The integrity unit keeps track of internal bus transactions.
- It is sufficient to externally observe only a small number of **critical events**.



# Post-silicon Certification

**Theorem.** If the integrity unit does not interrupt, then any post-silicon trace that passes the post-silicon analysis is a subsequence of a trace that would pass pre-silicon analysis under full observability.

**The theorem is proven is ACL2.**

- Makes use of underlying protocol invariants.

# Post-silicon Certification

**Theorem.** If the integrity unit does not interrupt, then any post-silicon trace that passes the post-silicon analysis is a subsequence of a trace that would pass pre-silicon analysis under full observability.

**The theorem is proven is ACL2.**

- Makes use of underlying protocol invariants.
- Proven by exploiting a decidable subclass of the logic.

# Post-silicon Certification

**Theorem.** If the integrity unit does not interrupt, then any post-silicon trace that passes the post-silicon analysis is a subsequence of a trace that would pass pre-silicon analysis under full observability.

**The theorem is proven is ACL2.**

- Makes use of underlying protocol invariants.
- Proven by exploiting a decidable subclass of the logic.

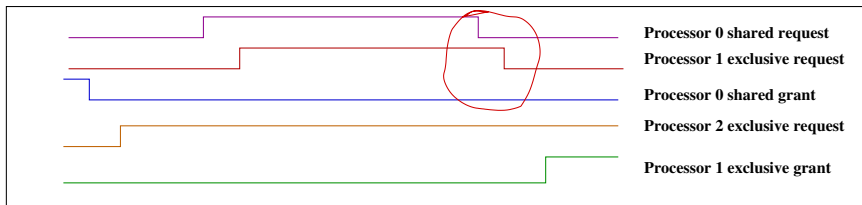
**The theorem formally connects post-silicon verification with pre-silicon analysis.**

# Using the System

**The system can identify subtle design errors.**

# Using the System

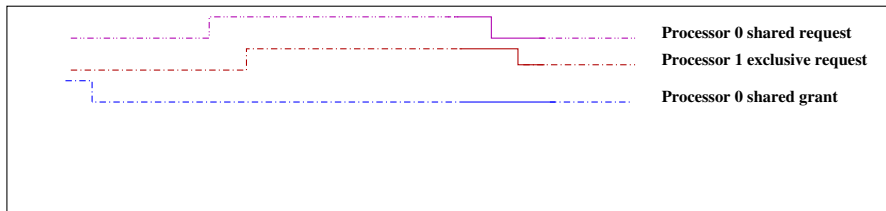
The system can identify subtle design errors.



Such errors are very difficult to exercise in simulation because of the non-determinism in the protocol.

# Using the System

**The system can identify subtle design errors.**



Such errors are very difficult to exercise in simulation because of the non-determinism in the protocol.

**The system identifies the error even under very poor observability.**

# Related Work

- [Gopalakrishnan and Chou](#): Limited observability checkers based on constraint solving and abstract interpretation.
- [Aschlager and Wilkins](#): Model checking to generate a short trace containing an observed bug.
- [Safarpour et al.](#): SAT solving to automatically find and repair stuck-at faults.
- [De Paula et al.](#): SAT solving to develop a “backspace” from a crashed state.

**Our approach is to introduce some of the analysis or checking into the silicon.**

# Conclusion and Future Work

**To our knowledge our work is the first effort on connecting pre-silicon and post-silicon verification through formal proofs.**

- Provides a flexible mechanism for making use of pre-silicon analysis in post-silicon verification.
- Makes use of **existing** design artifacts to facilitate post-silicon analysis.



# Conclusion and Future Work

**To our knowledge our work is the first effort on connecting pre-silicon and post-silicon verification through formal proofs.**

- Provides a flexible mechanism for making use of pre-silicon analysis in post-silicon verification.
- Makes use of **existing** design artifacts to facilitate post-silicon analysis.

**Of course, the results are preliminary.**

## **Future work:**

- Exploit information flow for automatic signal winnowing.
- Automate partitioning, given an observability and hardware bound.
- Tighten connection between pre-silicon and post-silicon.
  - Exploit faster post-silicon simulation to facilitate pre-silicon analysis.