



# **Post-Silicon Debug**

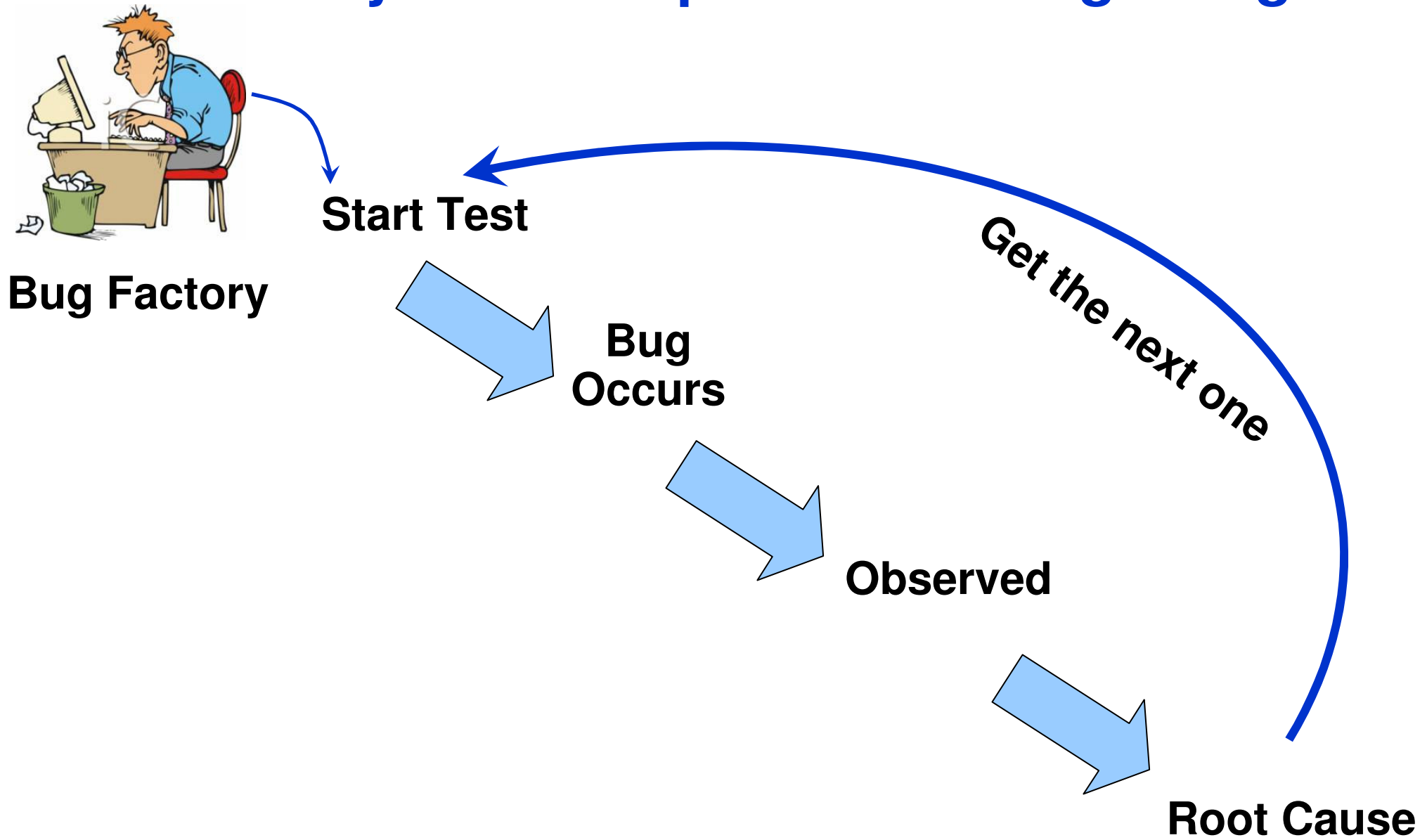
**Kevin Reick**

**Chief Engineer Power System Bring Up**

**June 7, 2012**



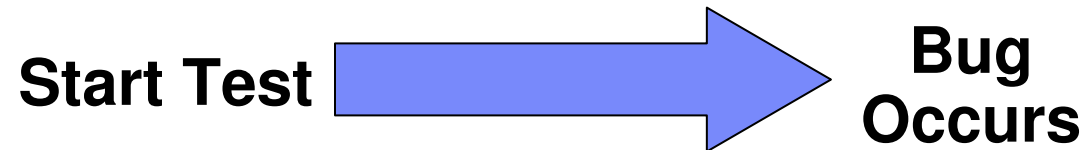
# Life Cycle of the post silicon logic bug



**Goal is to shorten the duration of each step**



## Step 1 – Finding the bug



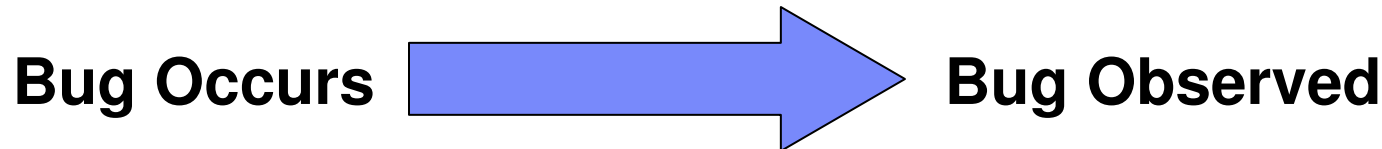
- ❑ **Duration can be from seconds to days**

### **Methods**

- **Coverage driven test programs**
- **Hardware Irritators**
  - **Modifies logic behavior**
  - **Requires upfront planning to add logic**



## Step 2 – Observation



- ❑ Duration can be from cycles to minutes
- ❑ Long duration -> Difficult to root cause

### Observed by

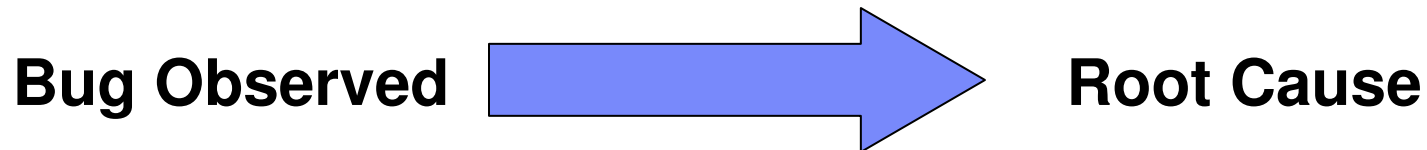
- Hardware
- Test Programs
- **System crash**
- **System freeze**

### Time Reduction Methods

- HW Error Detection
- Focused Test Programs
- Internal triggers
  - Logic Analyzers



## Step 3 – Root Cause



- ❑ **Quality of debug data is crucial**

### **Methods**

- **Internal logic analyzers**
- **External logic analyzers**
- **Transplanting hardware state into Simulation**



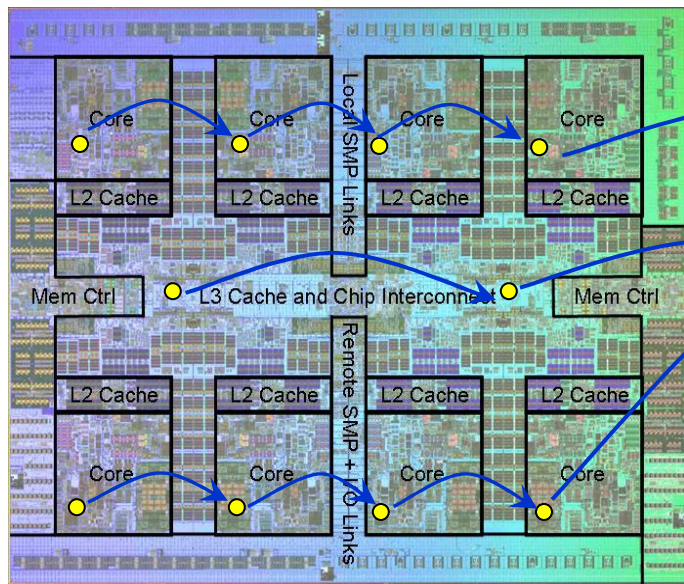
# Internal Logic Analyzers

## □ Pros

- Runs at cycle time
- Observe signals anywhere on the chip

## □ Cons

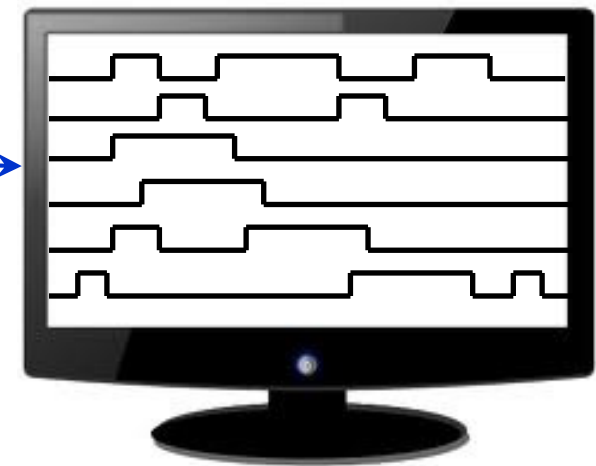
- Shallow depth. Mitigate with compression.
- Uses precious chip area
- Requires trigger logic to be useful



**Copy to file  
After trigger**



**Transfer State back  
Into Cycle Simulation  
Tool**





# External Logic Analyzers

## □ Pros

- Enables longer trace duration
- Works well for I/O and Memory busses

## □ Cons

- Requires dedicated systems
- Difficult to trace internal signals real time
- Uses precious chip I/O if larger debug busses are required.



# Transplant Hardware state back into verification model

## ☐ Pros

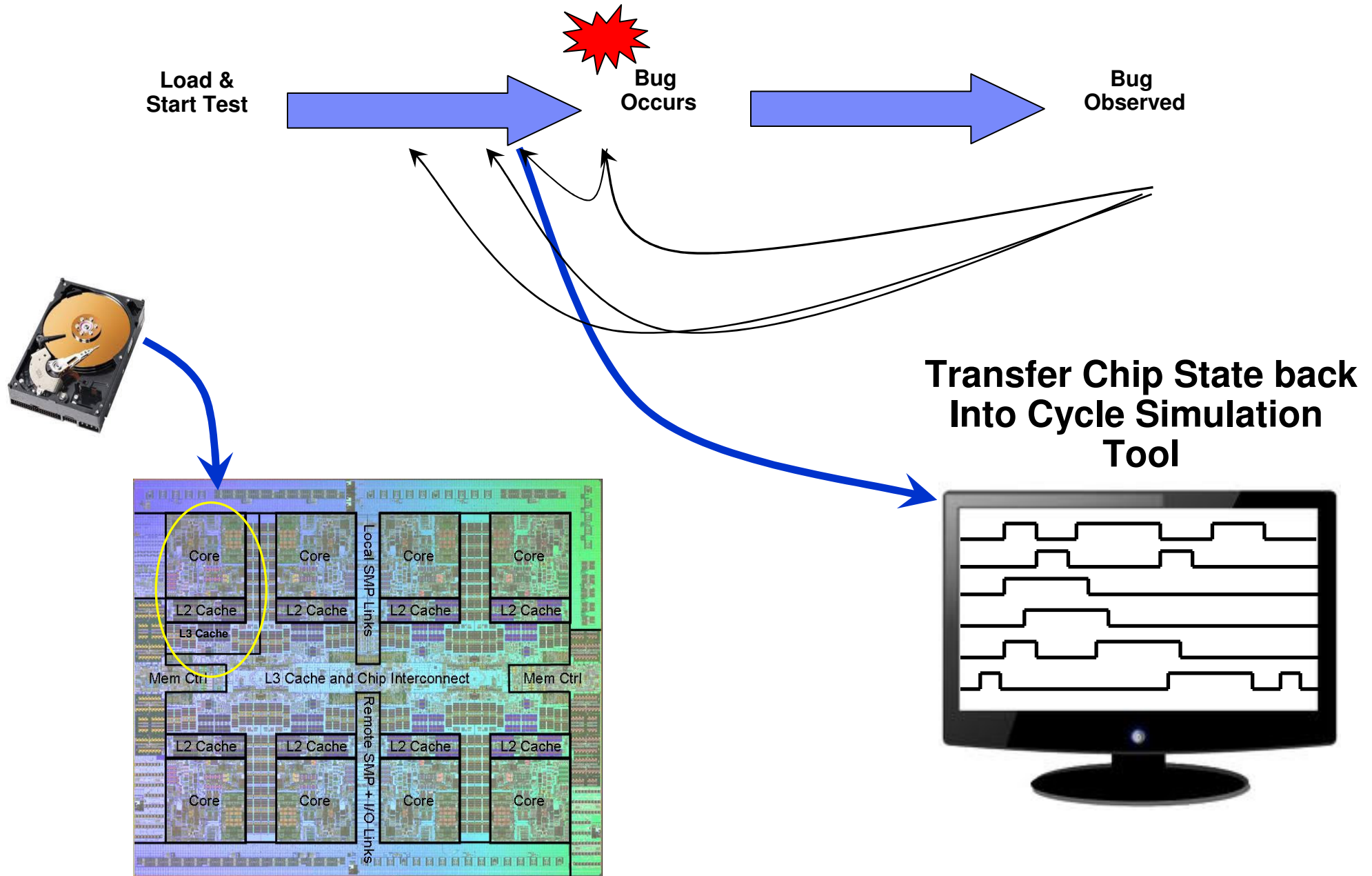
- Works for any test program that runs in clock synchronous domain

## ☐ Cons

- Does not work well for asynchronous driven events



# Transplant Hardware state back into verification model





- ☐ **Found the Bug. What's next?**
- ☐ **Severity can range from errata to mask change**
  - **Leverage capability in hardware to work around bug**
    - **Modes to modify chip behavior**
      - **Must be planned and implemented early**
    - **Chip performance is an issue**
    - **Method must be “surgical”**
  - **Firmware**
  - **Operating System**
- ☐ **Crucial Final Step**
  - **Recreate in verification to confirm mask change**



# Bring up starts early, before tape out

## ❑ Virtual Lab

### ➤ Hardware Accelerators

### ➤ Looks and feels like the lab

- Tools and scripts work unmodified when run on real hardware

### ➤ Validation of tools and scripts

### ➤ Validation of debug data

### ➤ Validation of work around capability

### ➤ Training ground for the lab team

### ➤ Validation of Test Programs and infrastructure

- Coverage is key



## Conclusion

- ☐ No one method can cover entire state space
- ☐ Quality of debug data is crucial
- ☐ Use a balanced approach
- ☐ Debug must be part of High Level Design
- ☐ Early training key to lab success



