WPI

NEHWS 2025

FaultDetective

Explainable to a Fault, from the Design Layout to the Software

Zhenyuan (Charlotte) Liu, Dillibabu Shanmugam and Patrick Schaumont

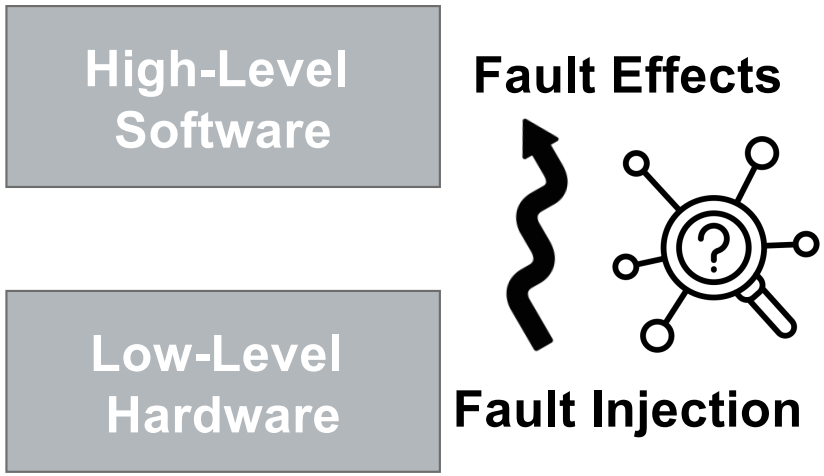
WPI Vernam Lab

CHES 2024

April 2025

What is Fault Root-Causing?

Explain Low-level Hardware Effects from High-level Software Observations



Why do we care?

- Three fault effects: correct output, no effects, faulty output
- Unpredictable fault behavior

Understand more than just the immediate output effects.

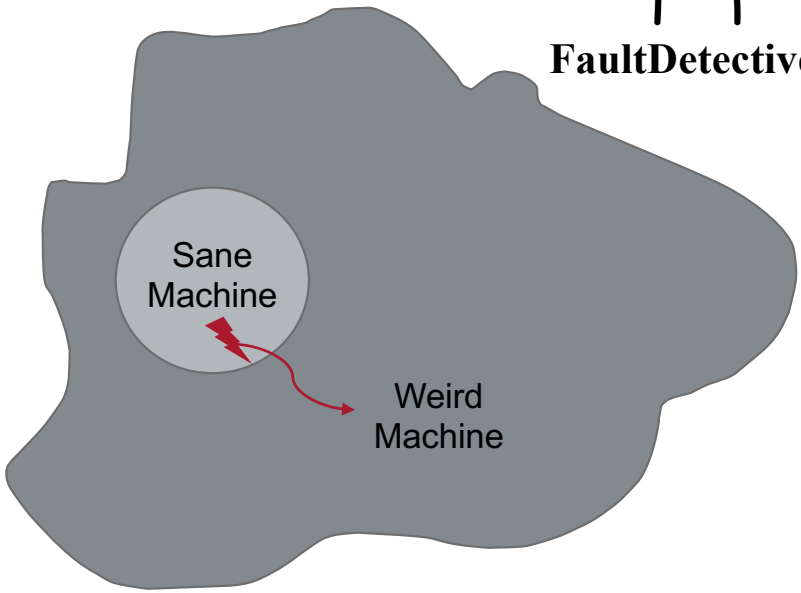
- ✓ Initial fault (the root-cause)
- ✓ Fault propagation from hardware to software
- ✓ Design improvements

WPI

Current Research and Challenges

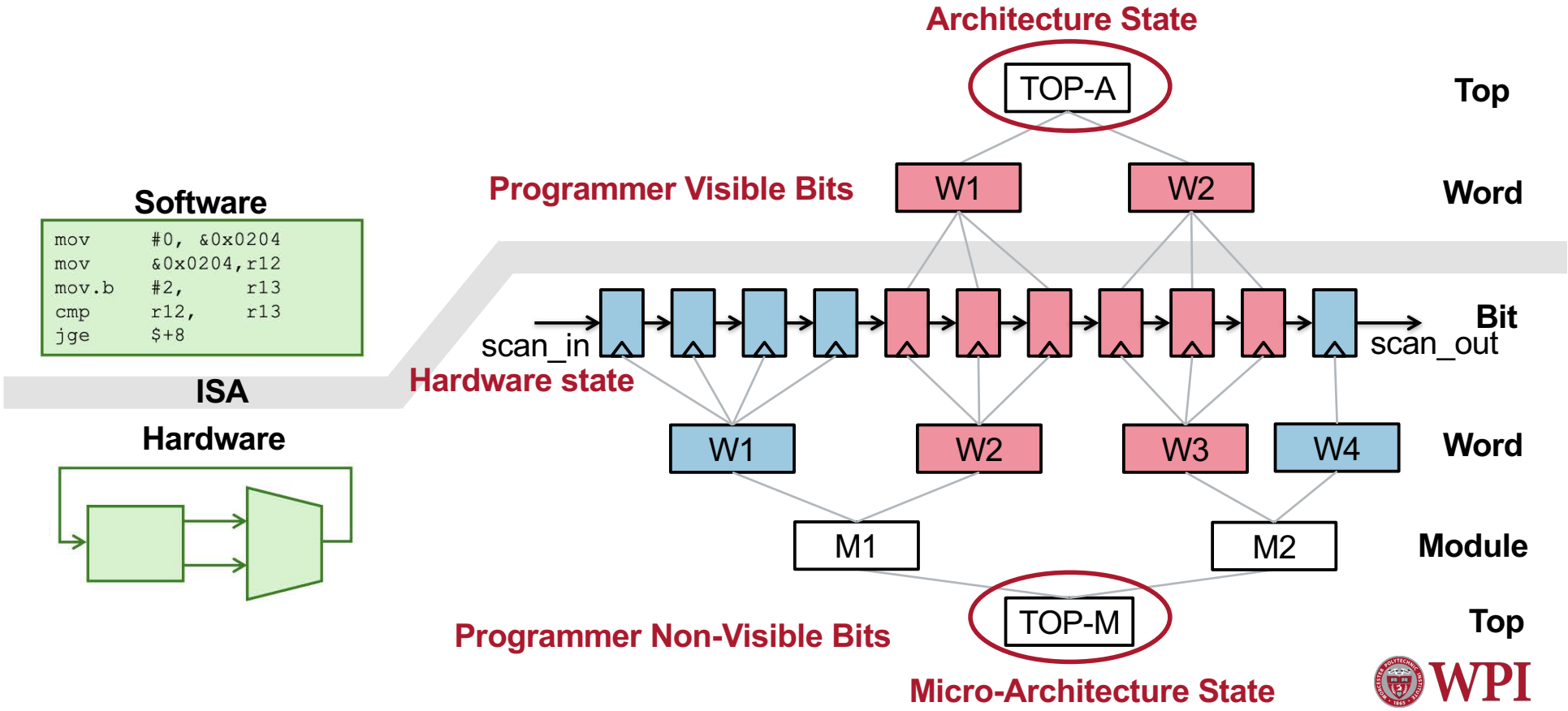
- Simulation-based research:
  - Based on a fault model (bit flipping, instruction skips …)
  - **Assumptions, depending on the accuracy of the fault model**
- Empirical research:
  - Observes the real-world effects of faults
  - **Limited visibility into hardware interactions**
  - **Cannot explain by the immediate output**
- The unpredictable fault behavior:
  - Sane machine
  - Weird machine
    - **Undefined state space**
    - **Difficult to model and analyze**
    - The state space for a 3000-bit processor contains 2^{3000} possible states

FaultDetective



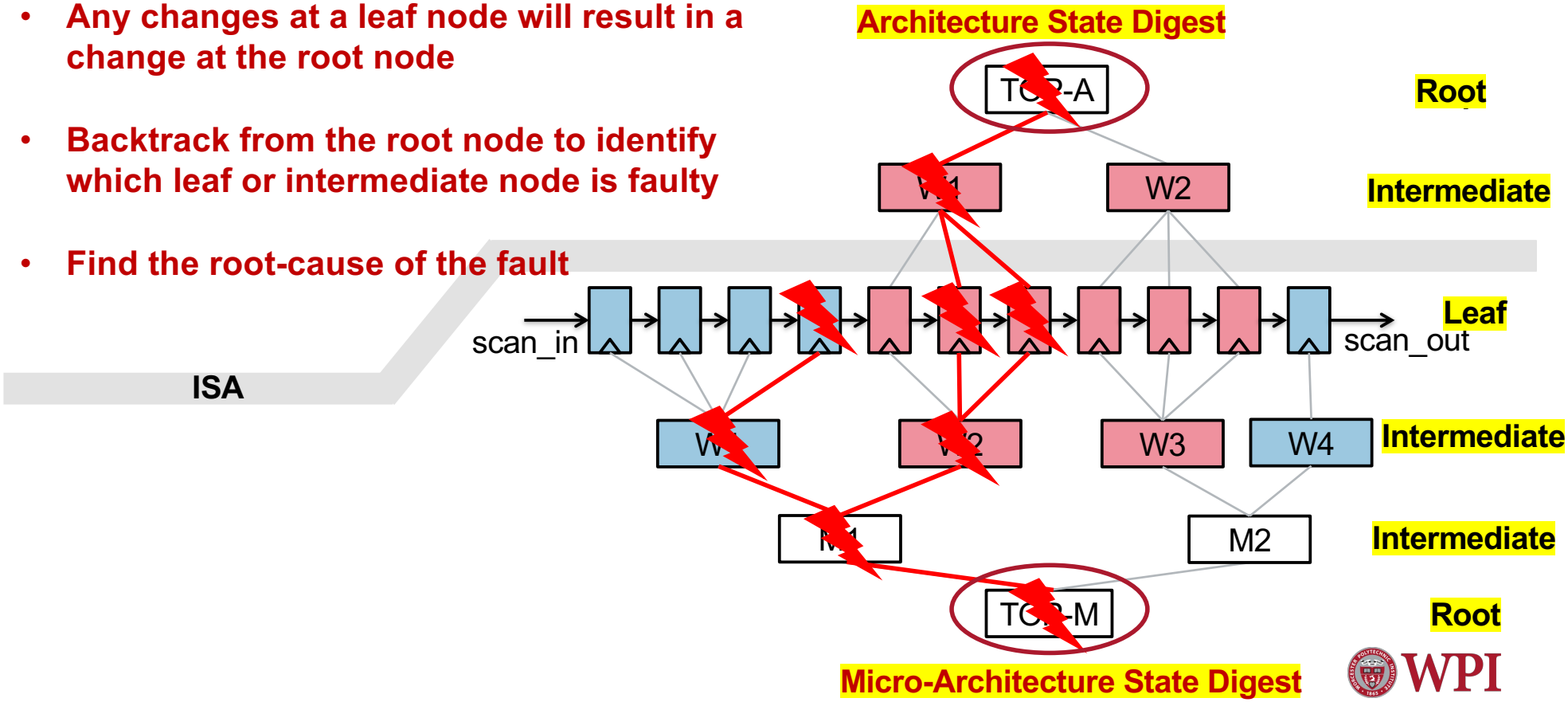
WPI

Architecture and Micro-Architecture States



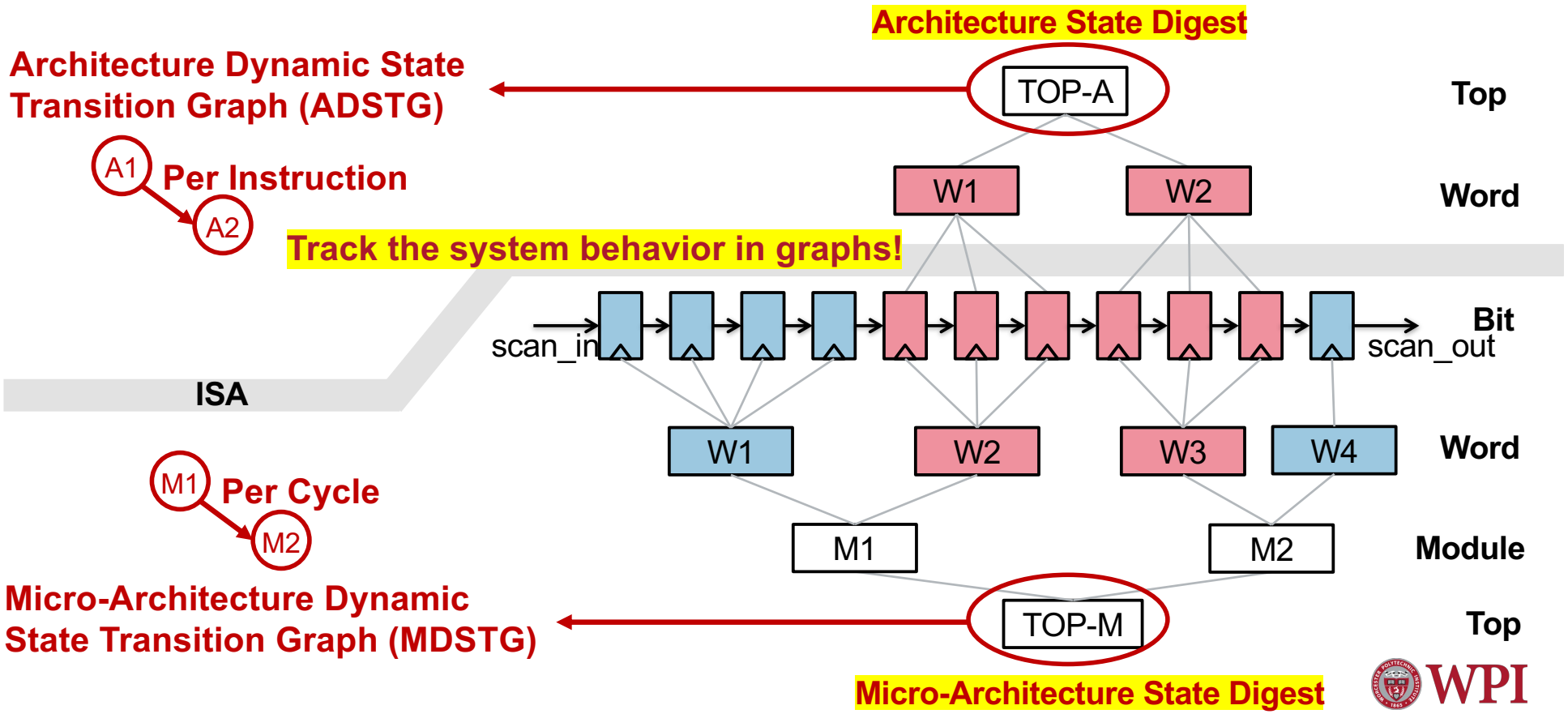
Hash Tree

- **Any changes at a leaf node will result in a change at the root node**
- **Backtrack from the root node to identify which leaf or intermediate node is faulty**
- **Find the root-cause of the fault**



*please refer to our paper and find more details on the hash function.

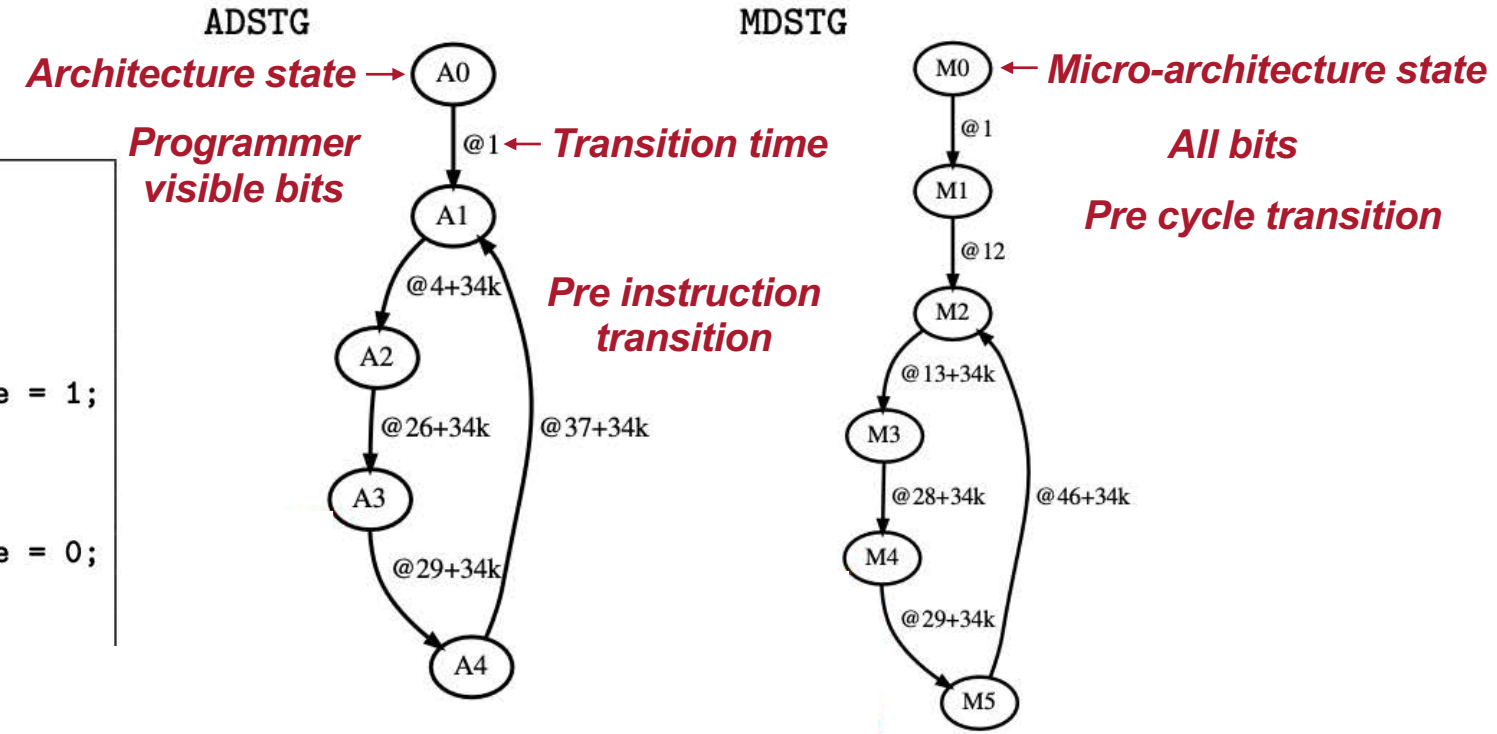
Dynamic State Transition Graph



Dynamic State Transition Graph – Before Fault

IMPLEMENTATION

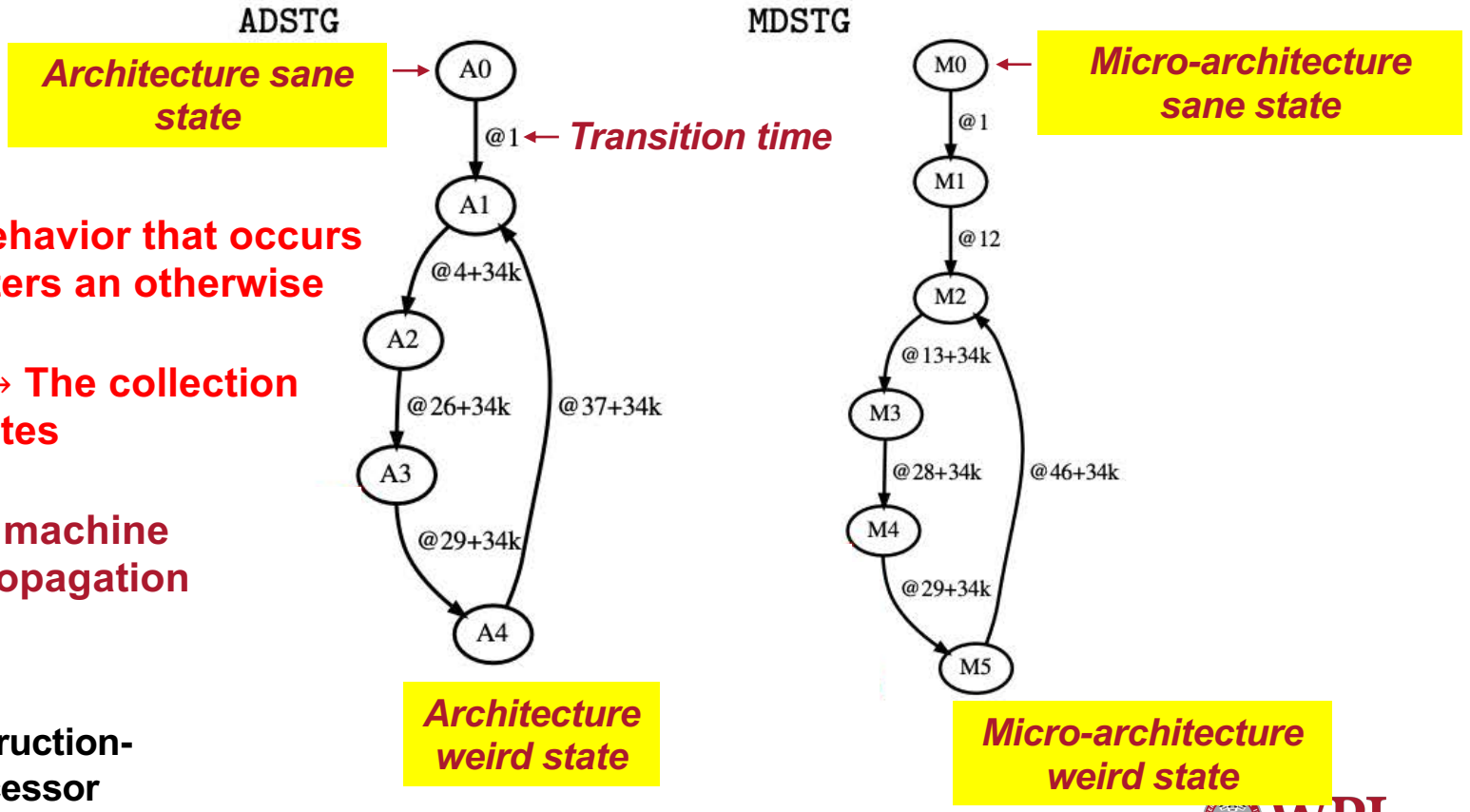
```
while (1) {

  switch (state) {
  case 0:
    P2OUT = 0x0;
    if (P2IN == 0x1) state = 1;
    break;
  case 1:
    P2OUT = 0x1;
    if (P2IN == 0x0) state = 0;
    break;
  }
```



- ***Sane State*** → **Achieved in accordance with the programmer's intent[1]**
- ***The Sane Machine*** → **The collection of all possible sane states**

[1] Thomas Dullien. Weird machines, exploitability, and provable unexploitability. *IEEE Transactions on Emerging Topics in Computing*, 8(2):391–403, 2020.

Dynamic State Transition Graph – After Fault



- ***Weird State*** → **The behavior that occurs when the sane state enters an otherwise unreachable state[1]**
- ***The Weird Machine*** → **The collection of all possible weird states**

- ✓ **Visualize sane/weird machine**
- ✓ **Initial fault and its propagation**

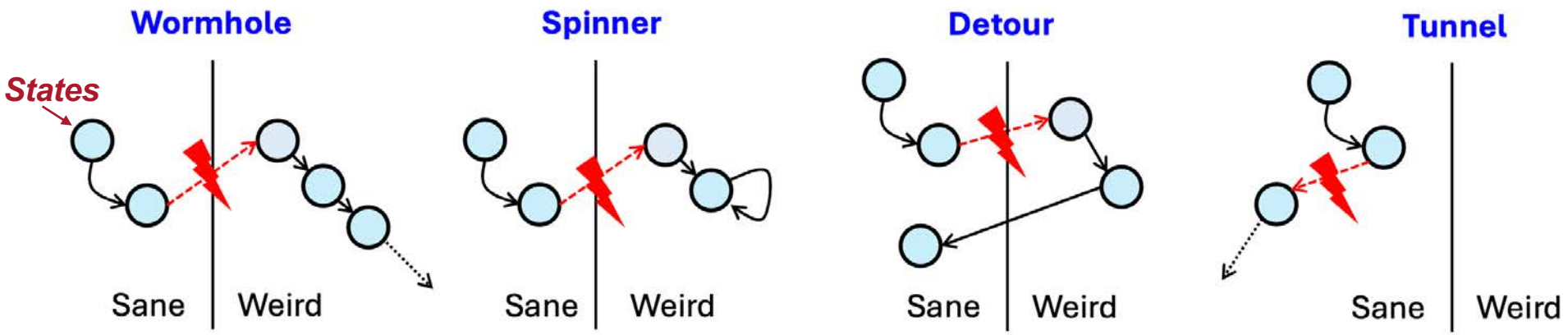
Register R2
03 → 13
Disables further instruction-fetch, halt the processor

[1] Thomas Dullien. Weird machines, exploitability, and provable unexploitability. *IEEE Transactions on Emerging Topics in Computing*, 8(2):391–403, 2020.

WPI

Visualized Fault Patterns

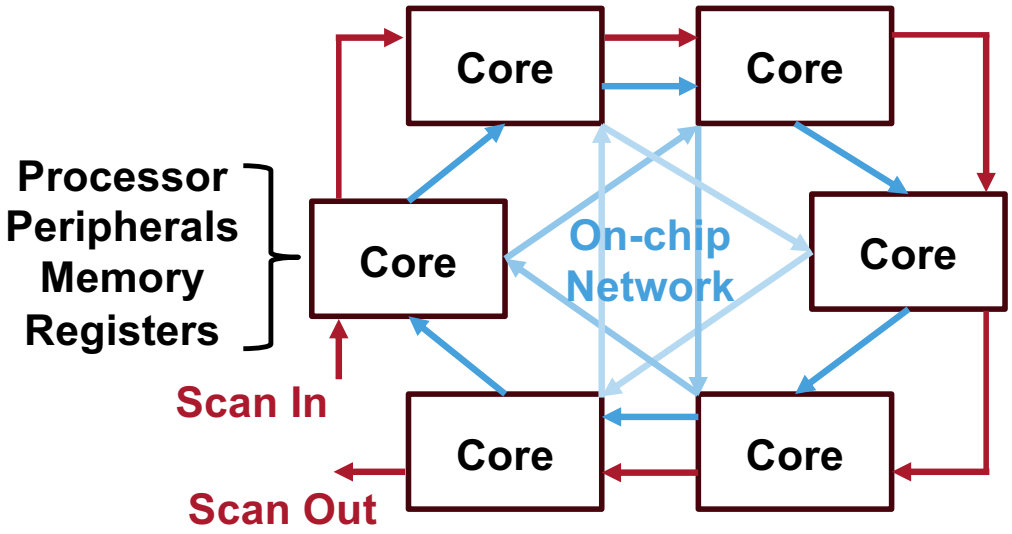
- Four fault patterns



Weird machine does exist!

Fault Root-Cause Analysis: FaultDetective

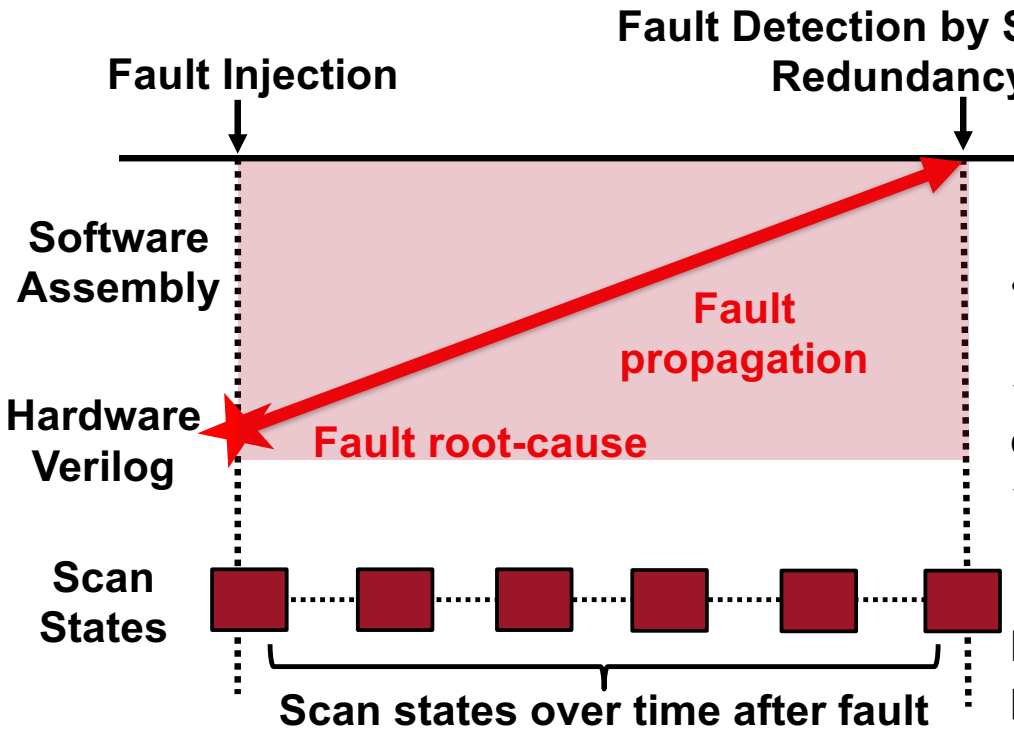
- Hardware Redundancy + Scan Chain



✓ **Lock-step execution**
✓ **Minimal hardware in on-chip network**
✓ **Fault checking in software by redundancy**
✓ **Once a fault is detected, halt all cores**
✓ **Scan chain provides visibility into all flip-flop bits**

WPI

Fault Root-Cause Analysis: FaultDetective



- Initial fault (root-cause)

✓ Detect differences among cores

✓ Earliest fault detection

Document hardware state periodically

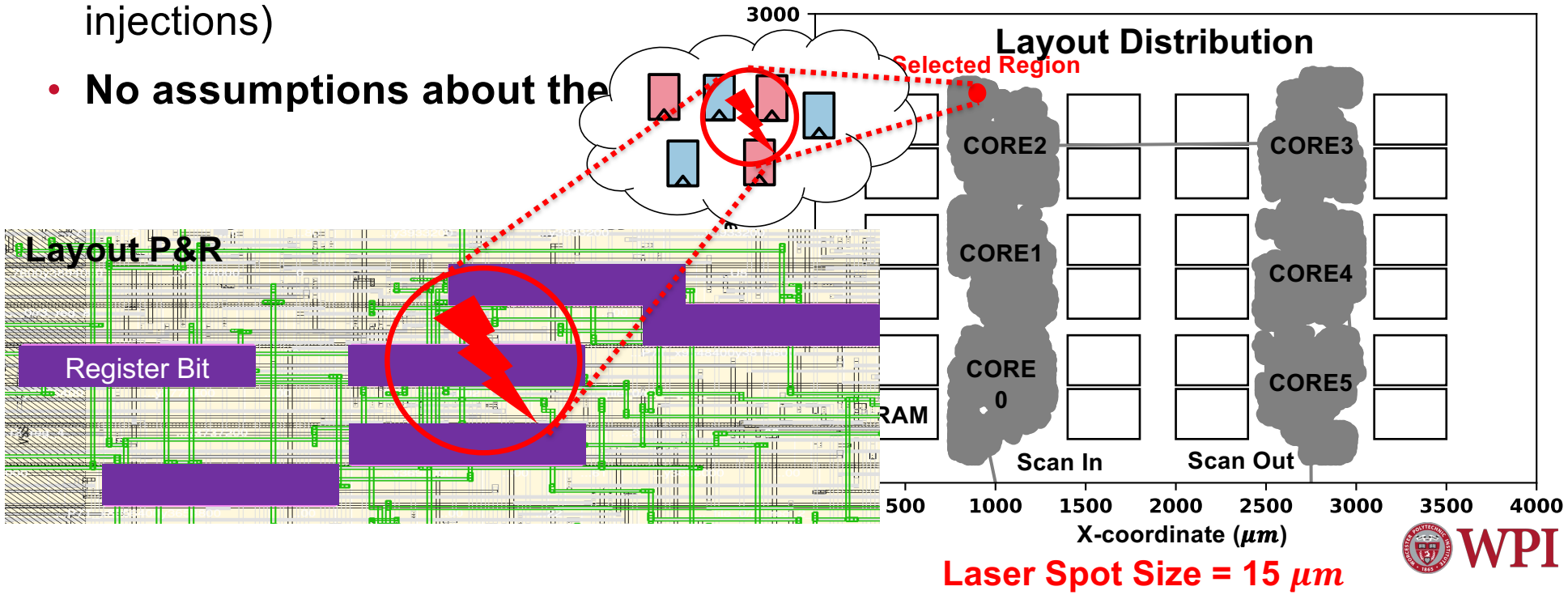
✓ Plot the ADSTG/MDSTG for both sane and weird machine

✓ Apply to both simulation and measurement fault experiments

WPI

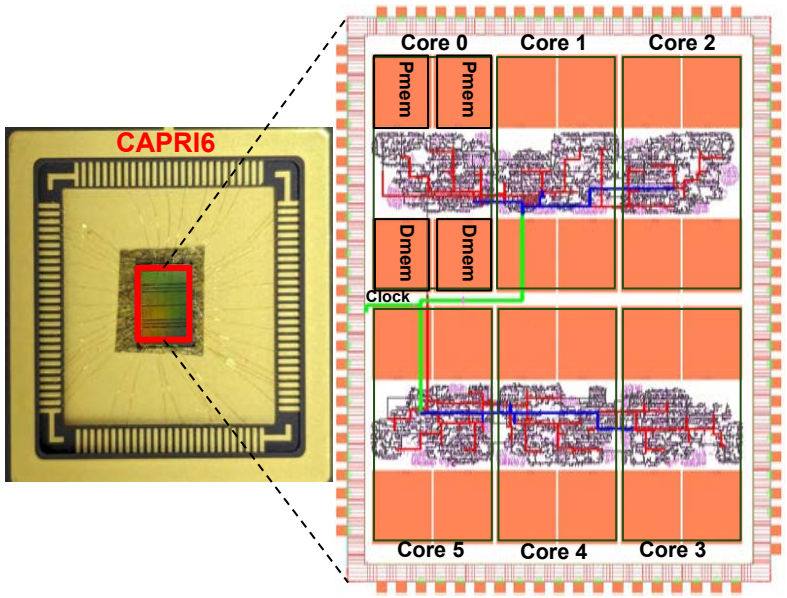
Layout-Aware Fault Analysis in Simulation

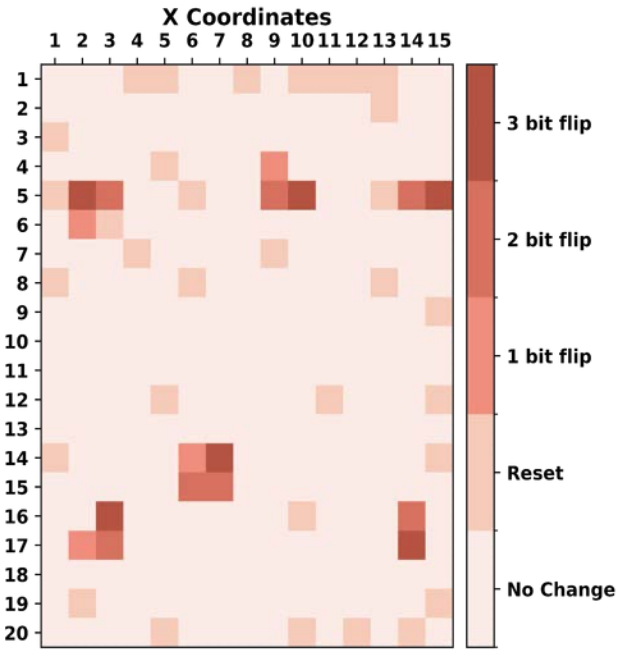
- Realization in ASIC (six-core MSP430)
- Implement real-world fault injection using layout data (laser and clock fault injections)
- **No assumptions about the**



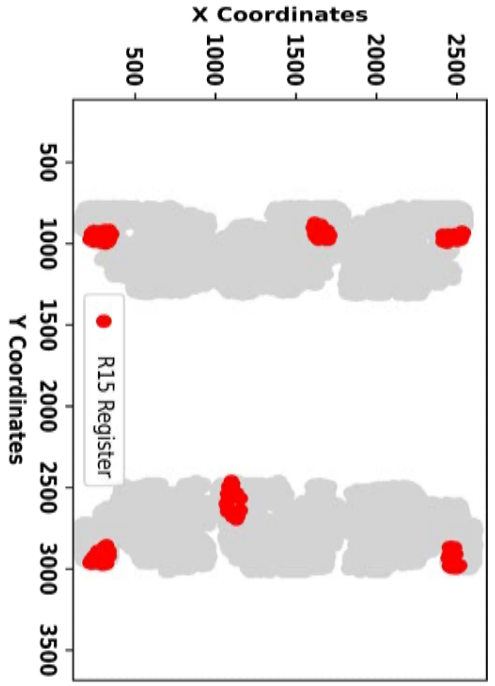
WPI

Measurement Result with EM-fault Injection





EM Heatmap



Faulty Scan Bits

WPI

WPI

Any Questions?

Thank you

zliu12@wpi.edu
dshanmugam@wpi.edu
pschaumont@wpi.edu

NEHWS 2025