QED: Quick Error Detection Tests for Effective Post-Silicon Validation

Jiong Xue

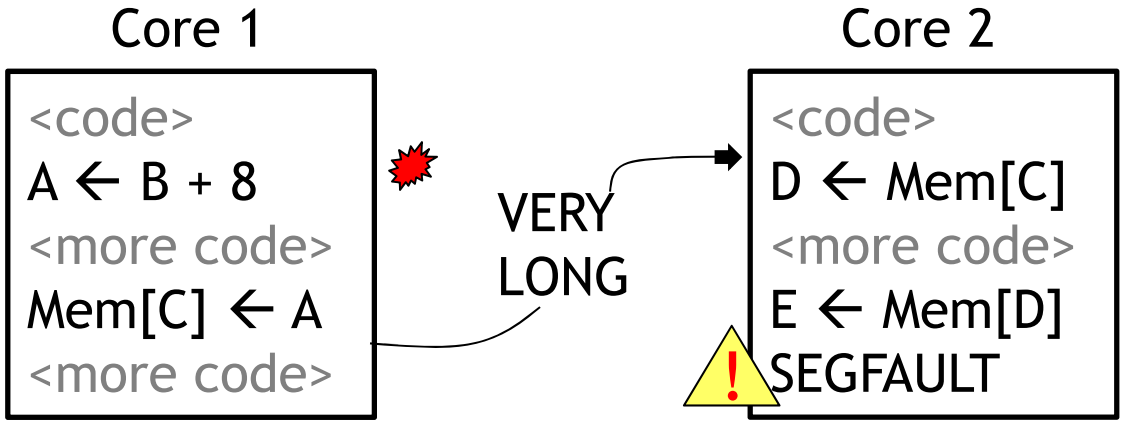
11/12/2015

Outline

- Background
- Proposed Solution - QED
  - EDDI-V
  - RMT-V
- Experimental Evaluation
  - Hardware experimental results
  - Simulation results
- Conclusion

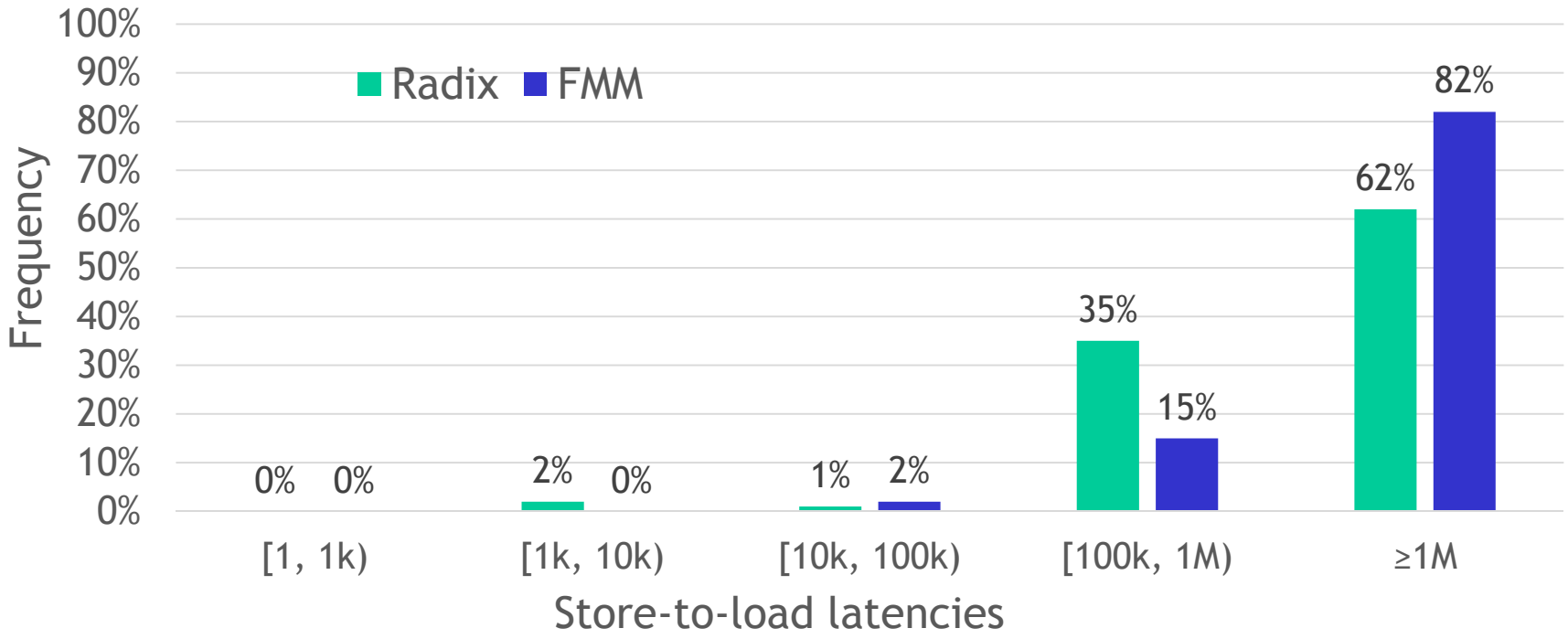
Background

- Post-Silicon Validation
- Electrical Bugs
- Error Detection Latency



Inter-core store-to-load latency

Motivation



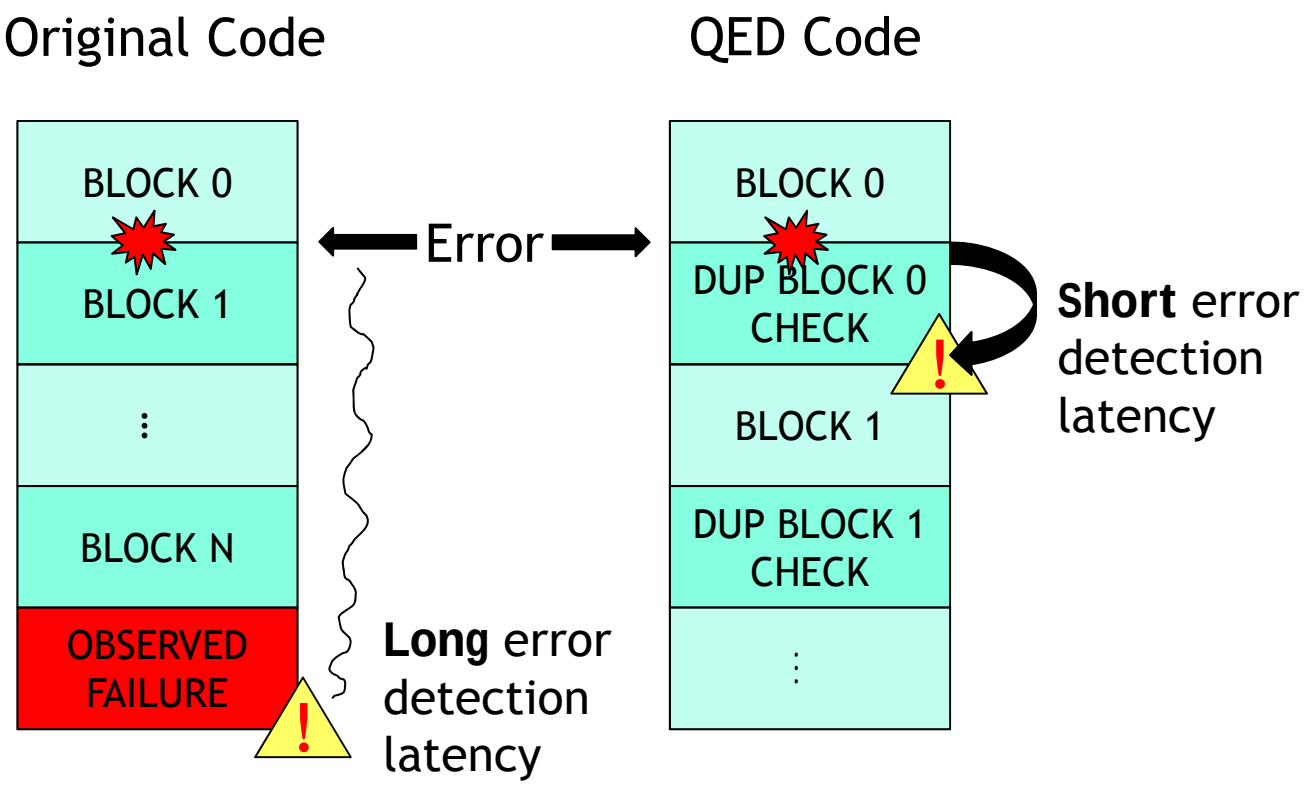
Long latencies limit the effectiveness of debug

Outline

- Background
- Proposed Solution - QED
  - EDDI-V (Error Detection by Duplicated Instructions for Validation)
  - RMT-V
- Experimental Evaluation
  - Hardware experimental results
  - Simulation results
- Conclusion

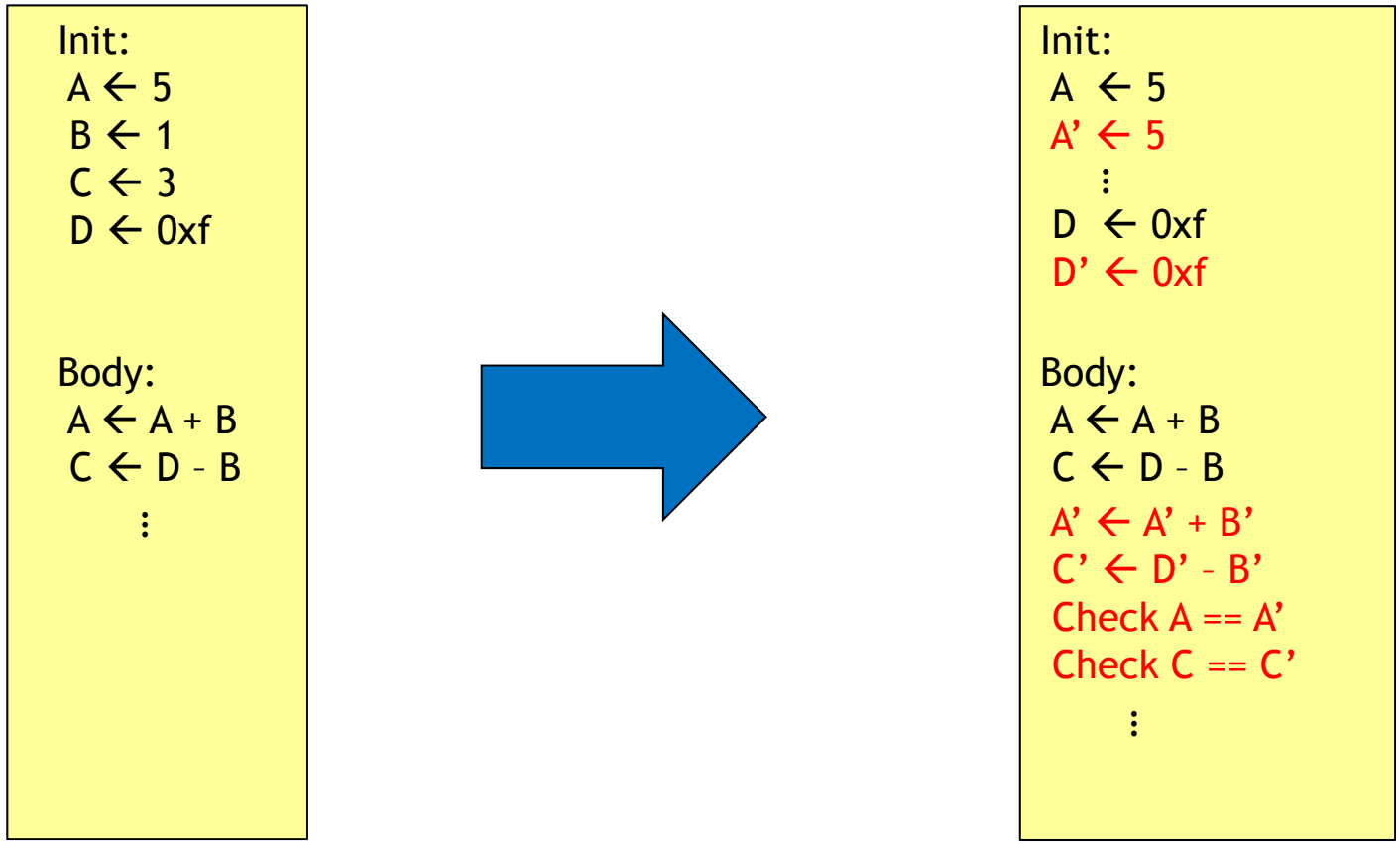
EDDI-V

Duplicate instructions and compare the results



EDDI-V Implementation

Reserve half of general purpose registers

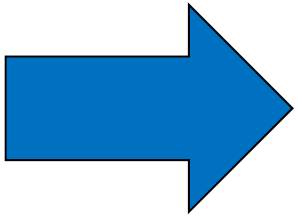


EDDI-V Implementation

Reserve half of memory

```
Init:
 A ← 5
 B ← 1
 C ← 3
 D ← 0xf

Body:
 A ← A + B
 C ← D – B
      ⋮
```



```
Init:
 MEM[0:3] ← {5,1,3,0xf}
 MEM[4:7] ← {5,1,3,0xf}

Body:
 (A,B,C,D) ← MEM[0:3]
 A ← A + B
 C ← D – B
 MEM[0:3] ← (A,B,C,D)
 (A,B,C,D) ← MEM[4:7]
 A ← A + B
 C ← D - B
 Check A == MEM[0]
 Check C == MEM[2]
 MEM[4:7] ← (A,B,C,D)
      ⋮
```

EDDI-V Advantages

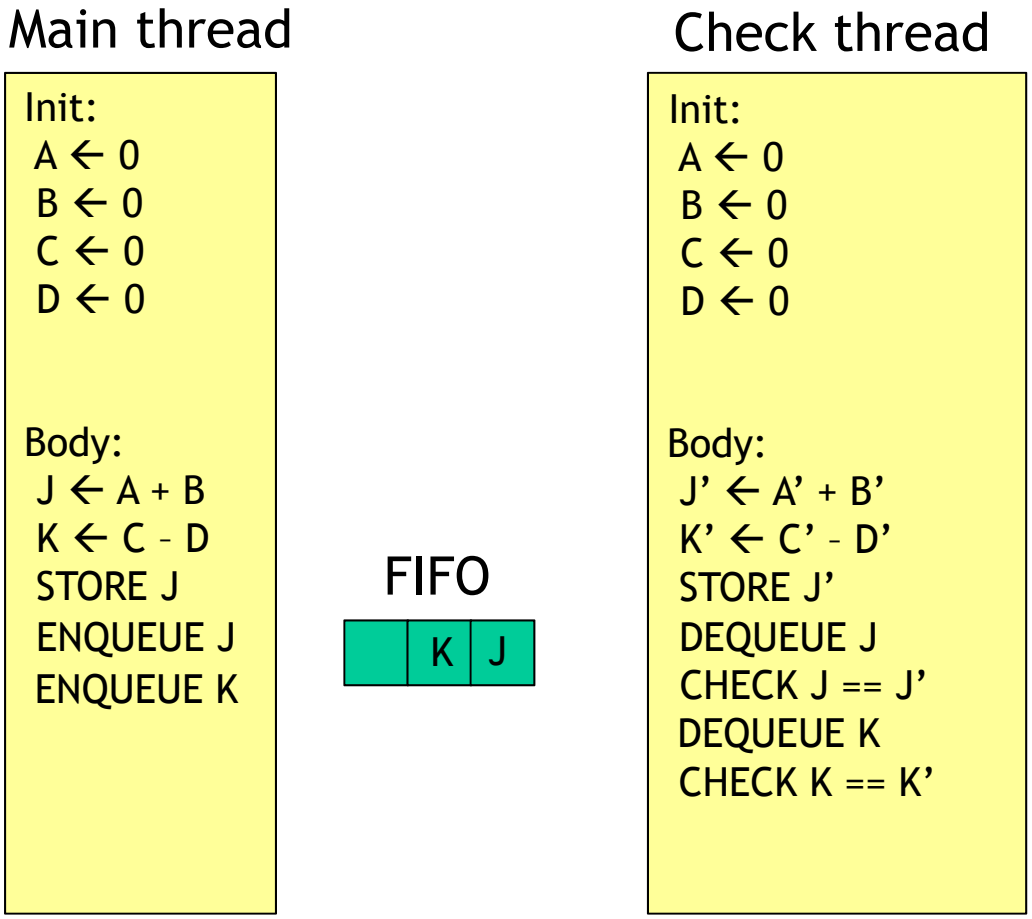
- Inst_min vs Inst_max
  - Minimum and maximum of original instructions inserted before QED code
- Bounded error detection latency
- Tradeoff between latency and intrusiveness

Outline

- Background
- Proposed Solution - QED
  - EDDI-V
  - RMT-V (Redundant Multi-Threading for Validation)
- Experimental Evaluation
  - Hardware experimental results
  - Simulation results
- Conclusion

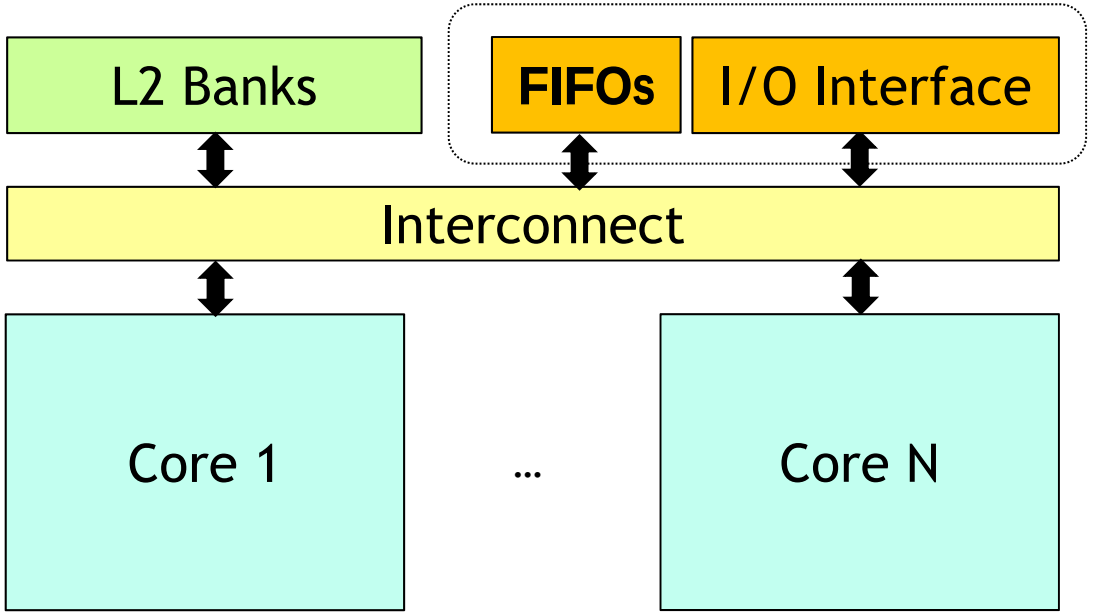
RMT-V

Duplicate the original thread



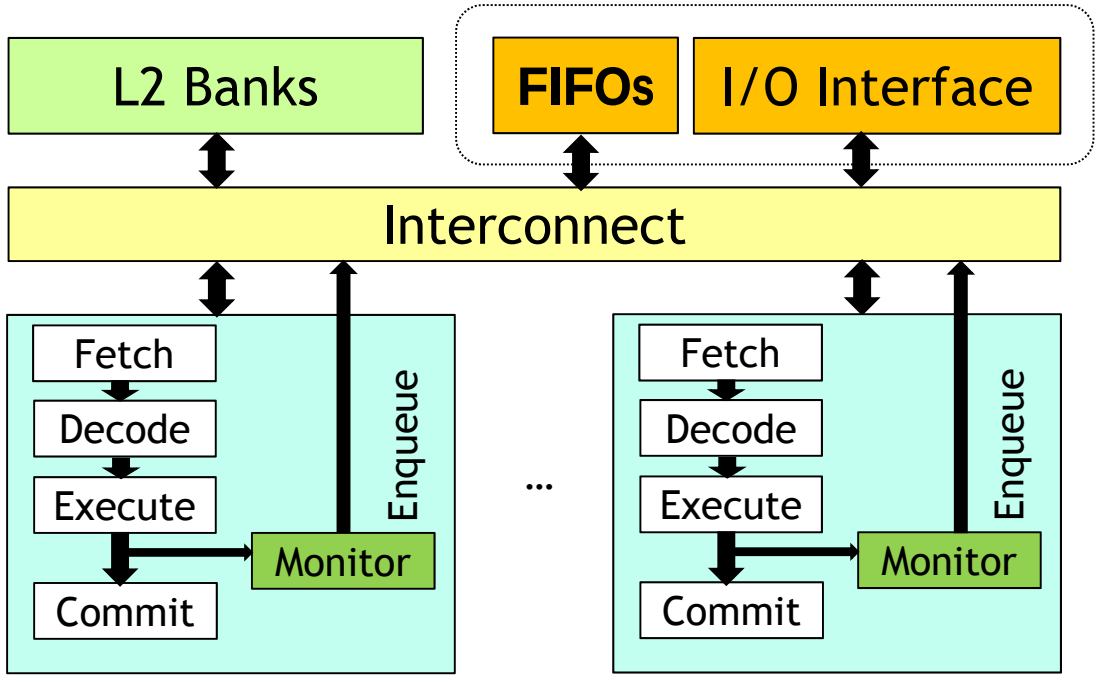
RMT-V Implementations

- Software RMT-V (S-RMT-V)
  - Lock-free queues implemented in software
  - Three instructions per enqueue operation
- S-RMT-V with Hardware Queues (S-RMT-V-HQ)



RMT-V Implementations

- Hardware RMT-V (H-RMT-V)
  - Monitor automatically enqueues the results

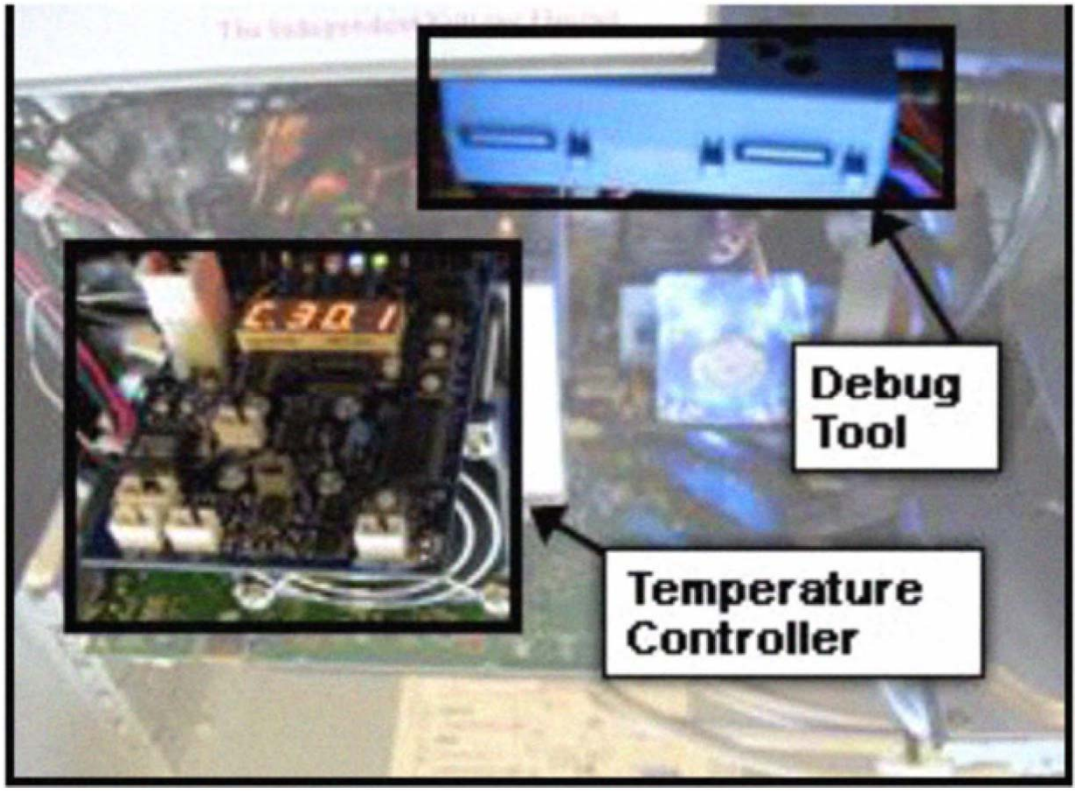


Outline

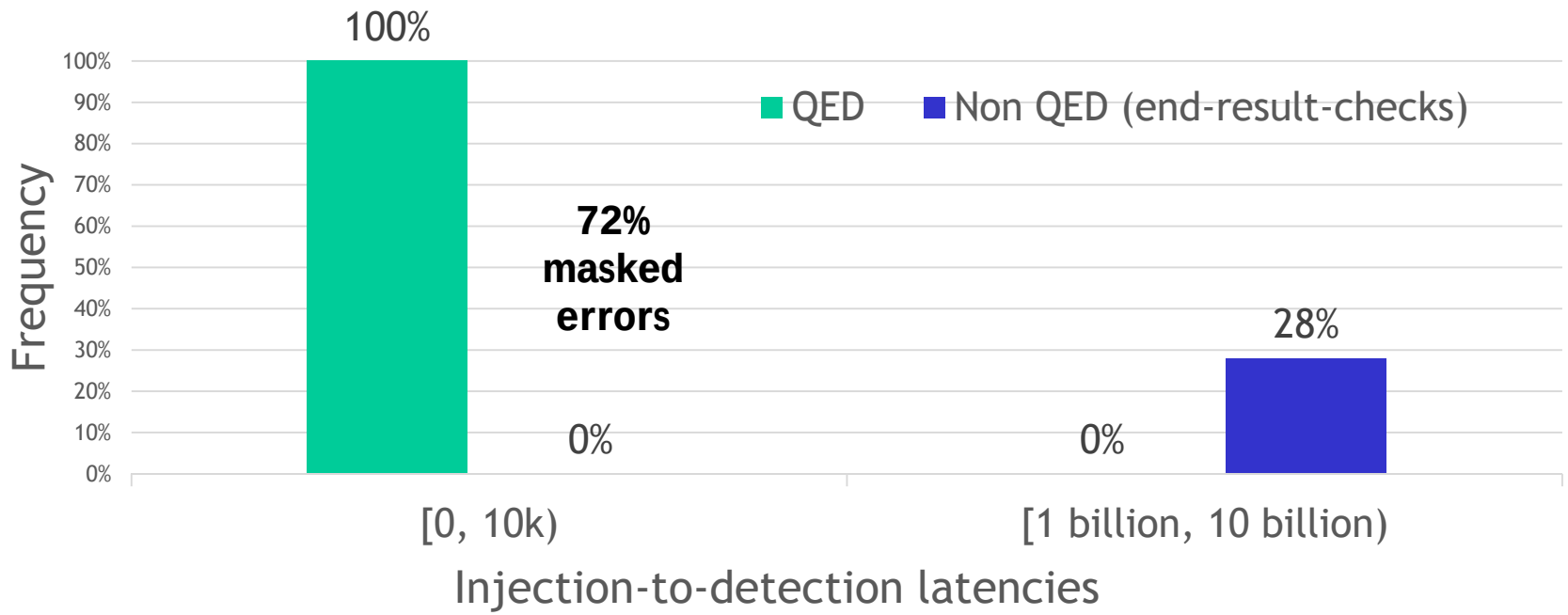
- Background
- Proposed Solution - QED
  - EDDI-V
  - RMT-V
- Experimental Evaluation
  - Hardware experimental results
  - Simulation results
- Conclusion

Hardware Experiments

- Quad-core Intel Core i7 Processor Platform
- Voltage and frequency can be changed
- Temperature remains constant

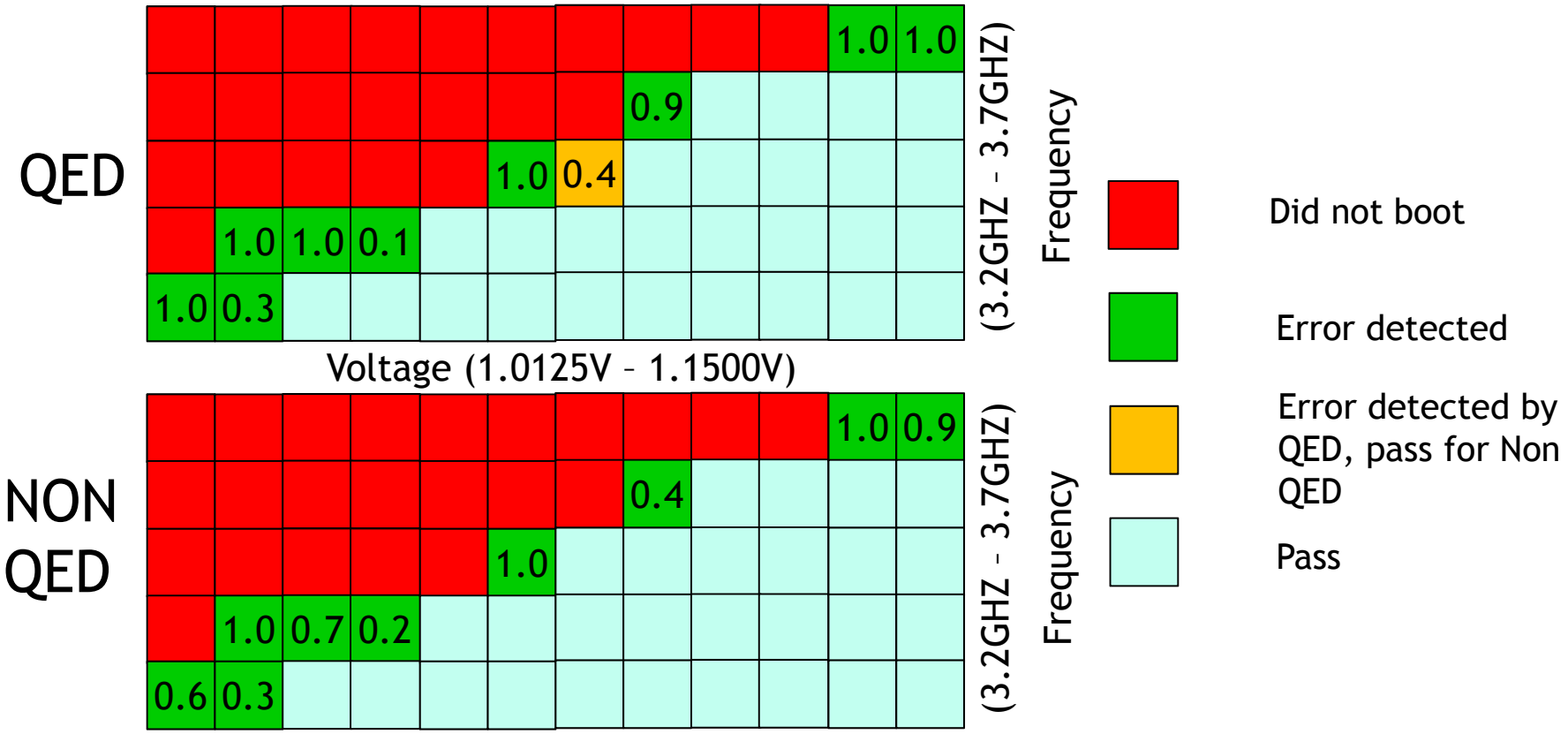


Error Detection Latency Results



- Error detection latencies ↓ by six orders of magnitude
- Masked errors can be detected

Electrical Bug Coverage Analysis



- QED can improve coverage
- Coarse-grained assertions may not be sufficient to reduce error detection latencies

Outline

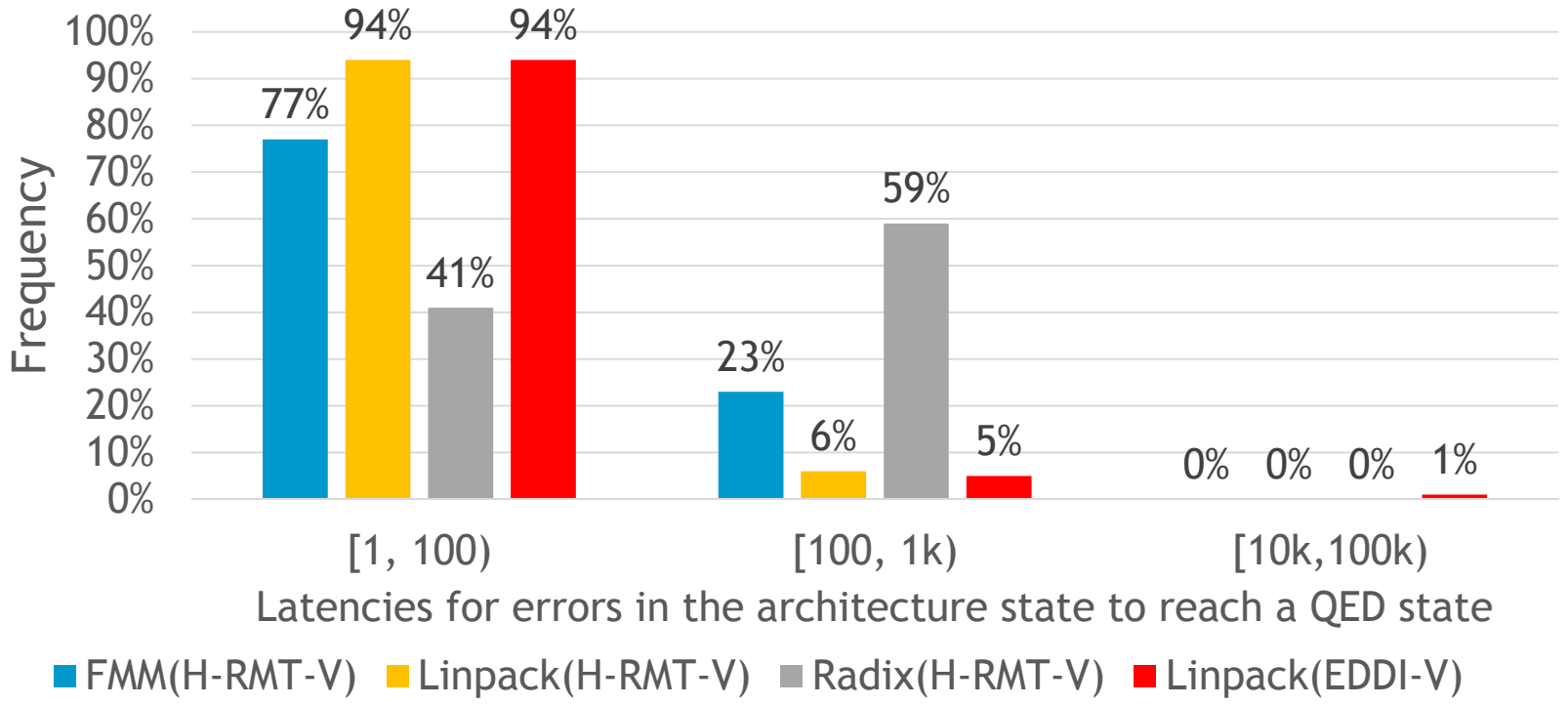
- Background
- Proposed Solution - QED
  - EDDI-V
  - RMT-V
- Experimental Evaluation
  - Hardware experimental results
  - Simulation results
- Conclusion

Simulation Experiments

- 4-core 4-way out-of-order MIPS processor

- Goal:
  - Estimate error detection latency
  - Characterize error detection latency for H-RMT-V

Simulation Results



- Latencies are within 1k cycles
- Simulation results are consistent with hardware experimental results

Outline

- Background
- Proposed Solution - QED
  - EDDI-V
  - RMT-V
- Experimental Evaluation
  - Hardware experimental results
  - Simulation results
- Conclusion

Conclusion

- Improve error detection latencies by six orders of magnitude
- Improve the coverage of post-silicon validation tests

Questions?

Debate

- Since QED detects only electrical bugs, is it good to combine QED with ISA diversity method?
- Which approach is better, EDDI-V or RMT-V?

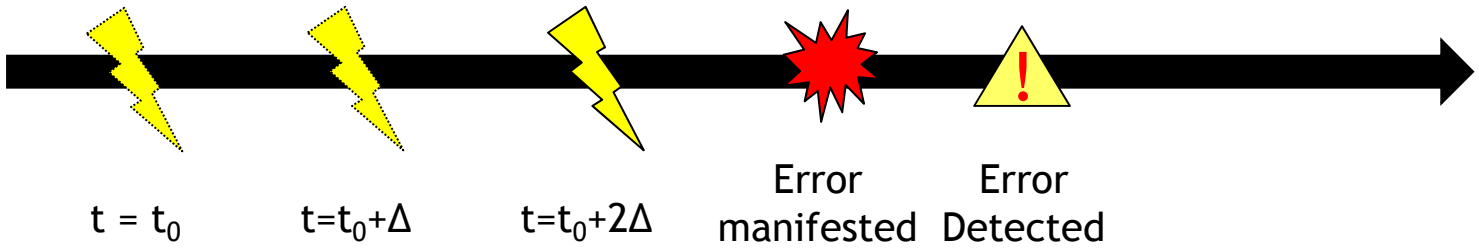
Thank you!

Backup

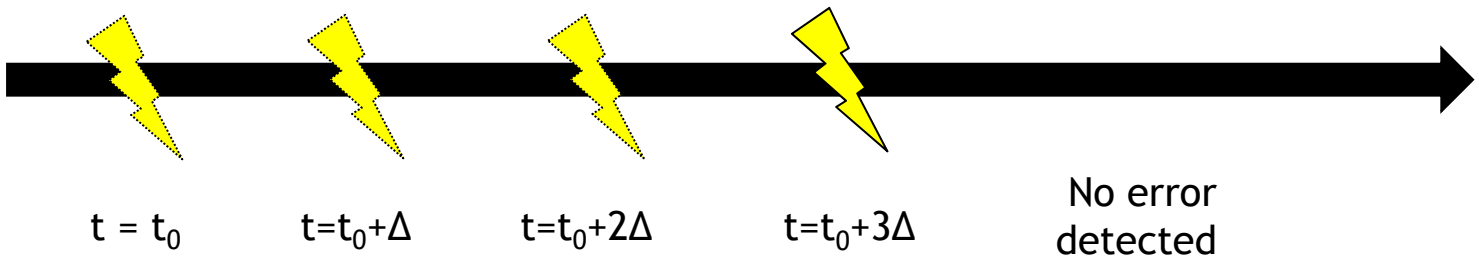
Error Detection Latency Measure

Error Detection Latency Measure

Error Detection Latency Measure



Error Detection Latency Measure



Error Detection Latency Measure