On-Chip Fault Diagnosis for Early-Life and Wear-Out Failures

Matthew Beckler

PhD Candidate

Outline

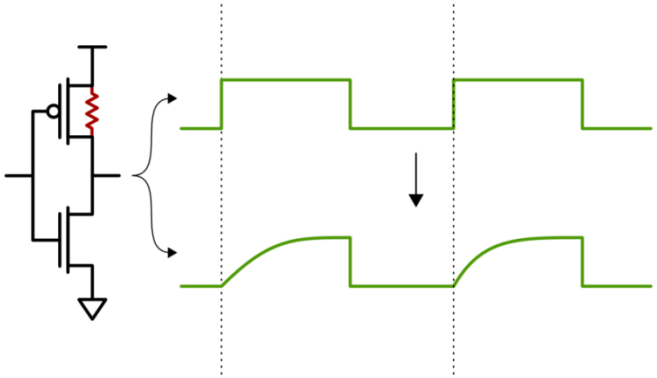
- Motivation
- Current Work
- Proposed Work
- Summary
- Discussion

Motivation

- Aging and early-life failures are…
  - Exhibiting larger effects with scaling
  - Not sufficiently detected by wafer/package test at *t*=0

A dominant PMOS aging mechanism is negative-bias temperature instability (NBTI)

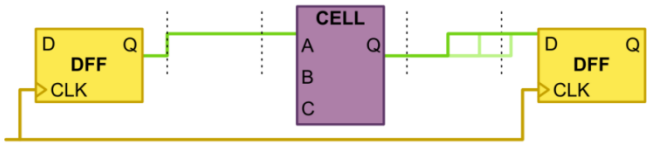
- Slows PMOS transitors over time
- Speed of chip can significantly degrade



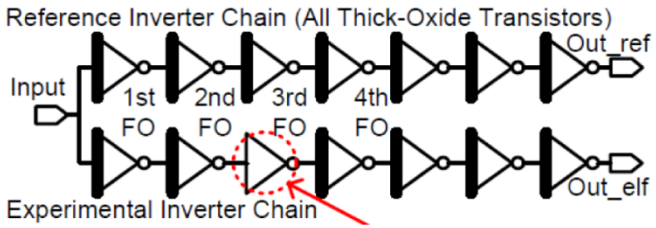
Agarwal, Paul, Zhang, Mitra, "Circuit Failure Prediction and Its Application to Transistor Aging", VTS 2007

Gate-oxide defects can result in early-life failures (ELF)

- Manifest as increased delay in standard cells.



- Experimental validation of behavior:

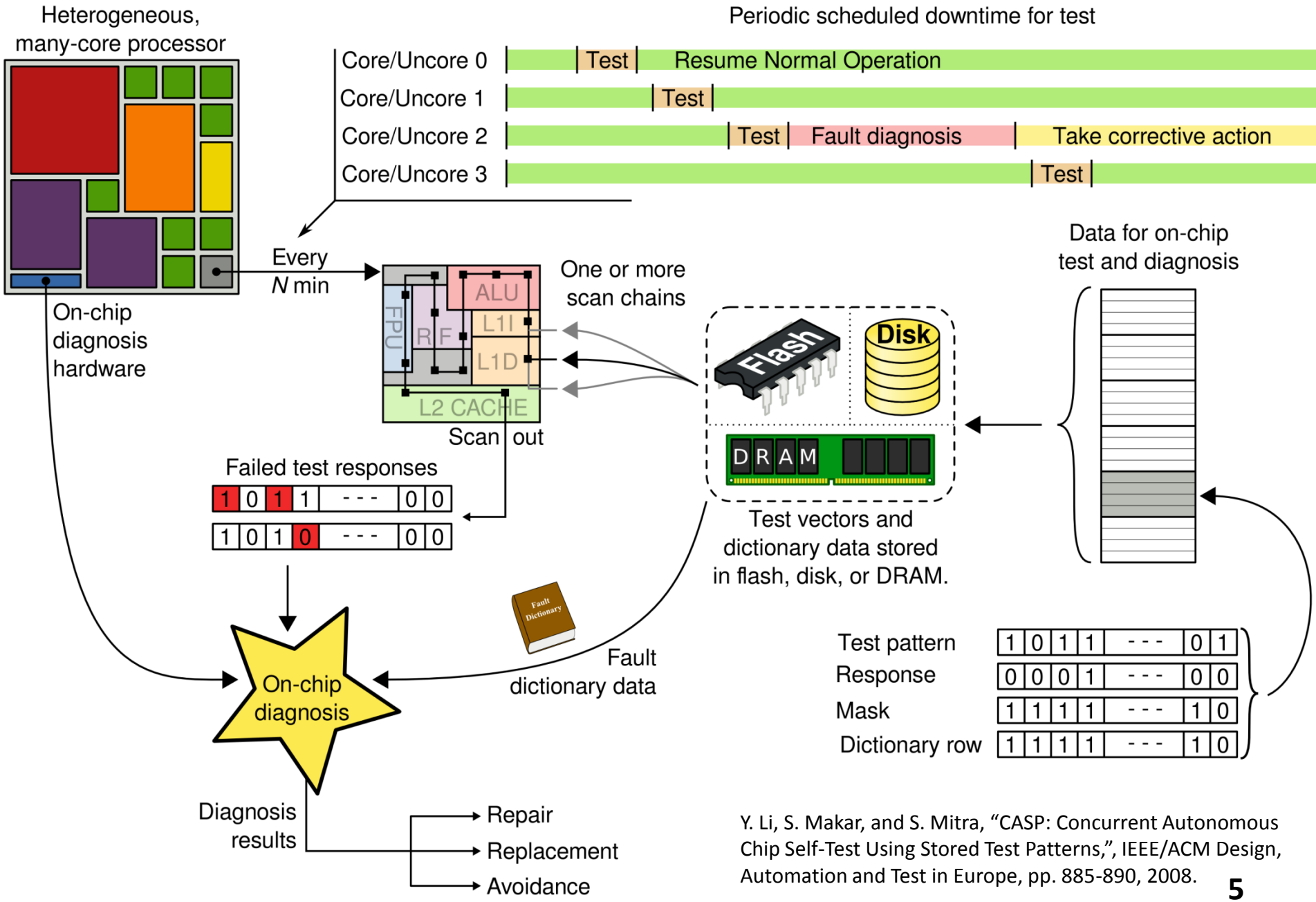


Kim, Chen, Kameda, Mizuno, Mitra, "Gate-Oxide Early-Life Failure Identification using Delay Shifts", VTS 2010

Motivation

- Other approaches
  - Conservative design
  - Design for best-case, cherry-pick devices that work
  - Fault tolerance (e.g., triple module redundancy)

- Goal: Ensure system robustness by:
  1. Periodic test during runtime (CASP) [Li 08]
  2. **Identify the location of any faults**
  3. Apply on-chip avoidance/redundancy/repair



Y. Li, S. Makar, and S. Mitra, "CASP: Concurrent Autonomous Chip Self-Test Using Stored Test Patterns,", IEEE/ACM Design, Automation and Test in Europe, pp. 885-890, 2008.

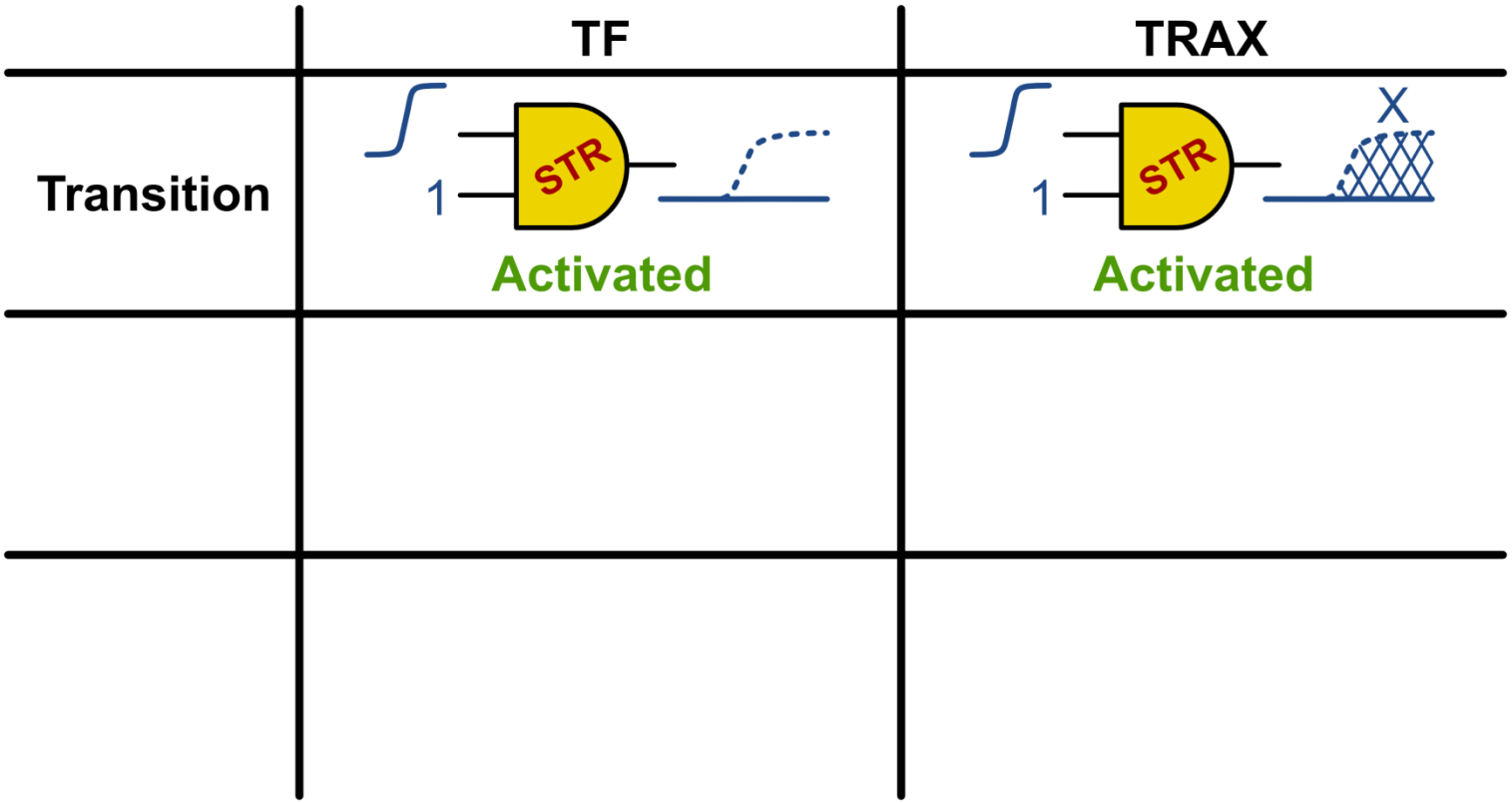
Outline

- Motivation
- Current Work
  - TRAnsition-X Fault Model (TRAX)
  - Hierarchical Fault Dictionary
  - Diagnosis Architecture
  - Experiment Validation
- Proposed Work
- Summary
- Discussion

Current Work No. 1 – TRAX Fault Model

TRAX has 3 features that go beyond the conventional:

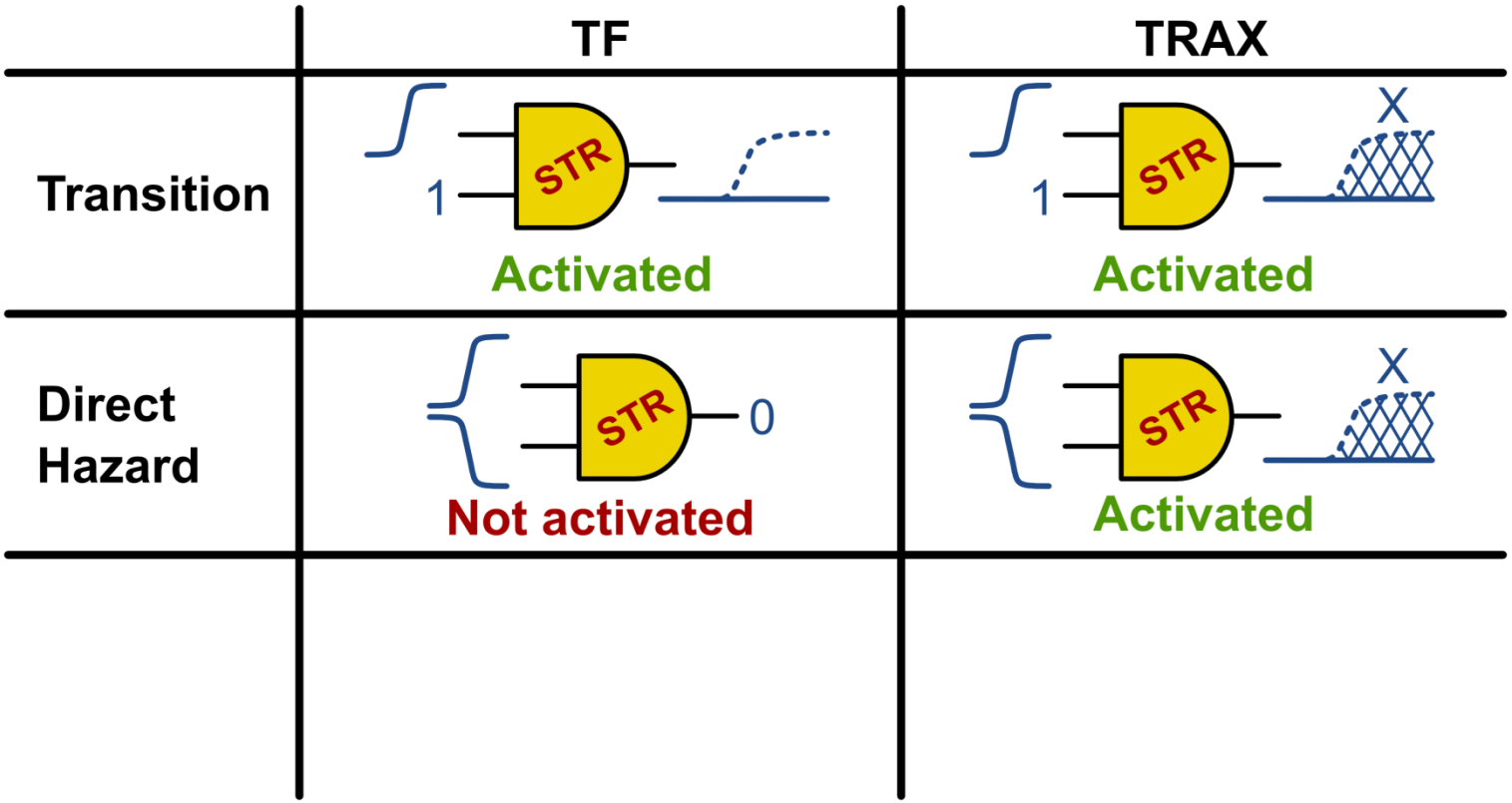
1. Activation creates an “X” instead of a slowed transition
2. Activation can be caused by transitions or hazards



TRAX Fault Model

TRAX has 3 features that go beyond the conventional:

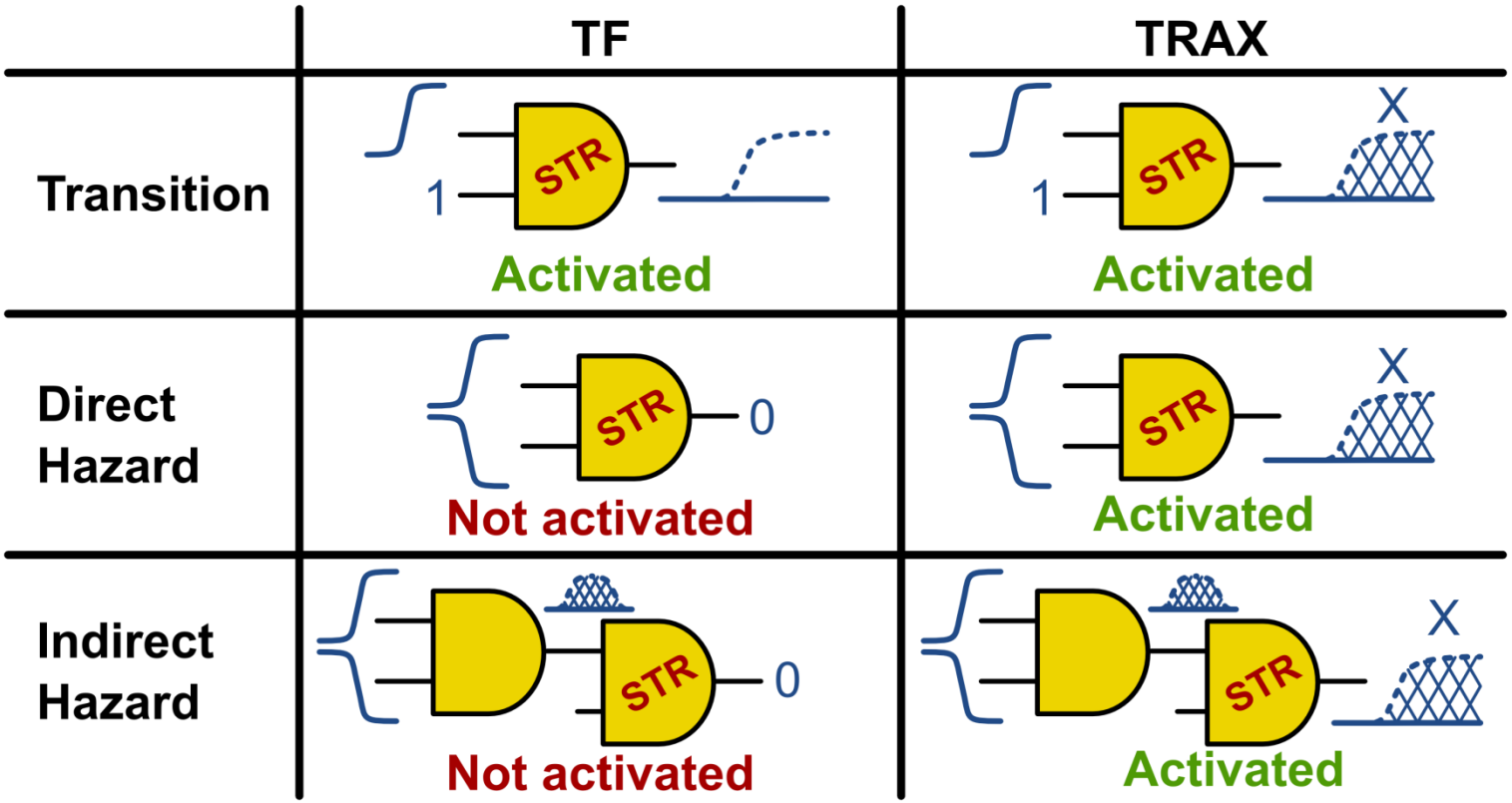
1. Activation creates an “X” instead of a slowed transition
2. Activation can be caused by transitions or hazards



TRAX Fault Model

TRAX has 3 features that go beyond the conventional:

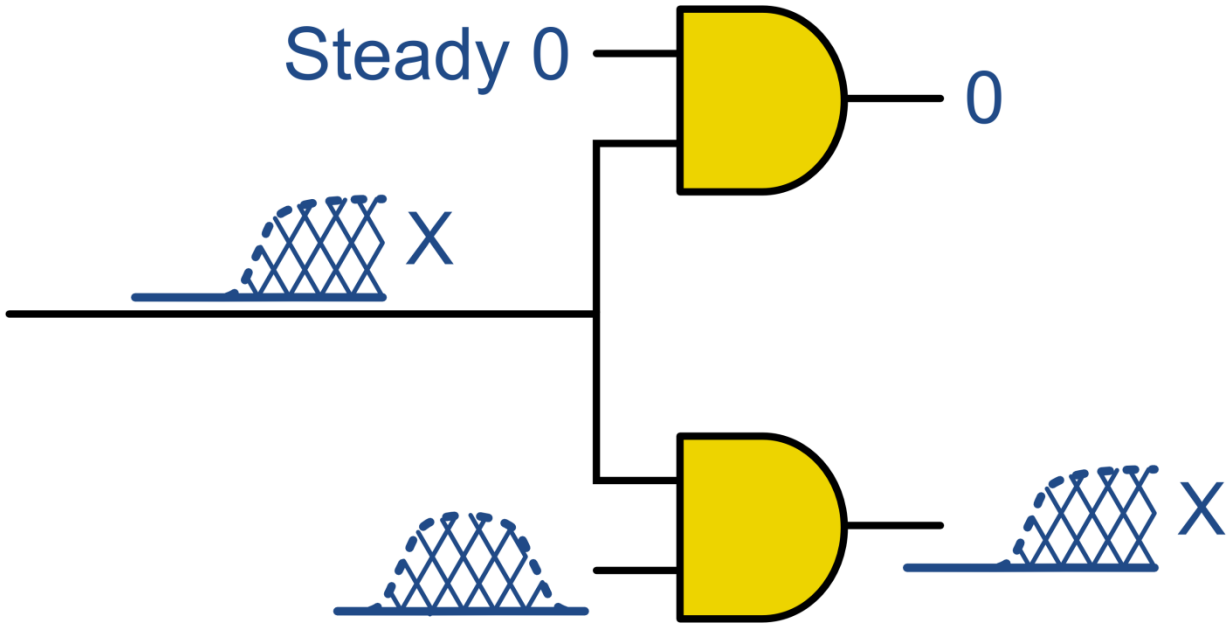
1. Activation creates an "X" instead of a slowed transition
2. Activation can be caused by transitions or hazards



TRAX Fault Model

TRAX has 3 features that go beyond the conventional:

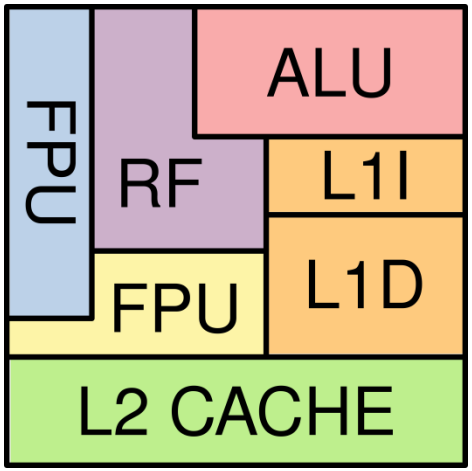
3. Side inputs for propagating: Any except stable controlling



Current Work No. 2 – Hierarchical Fault Dictionaries

- Repair/replacement/avoidance (RRA) employed above the gate/wire level (i.e., the module level)
  - Module = sub-circuit of 10 to 1000 gates
  - Module divisions determined by designer

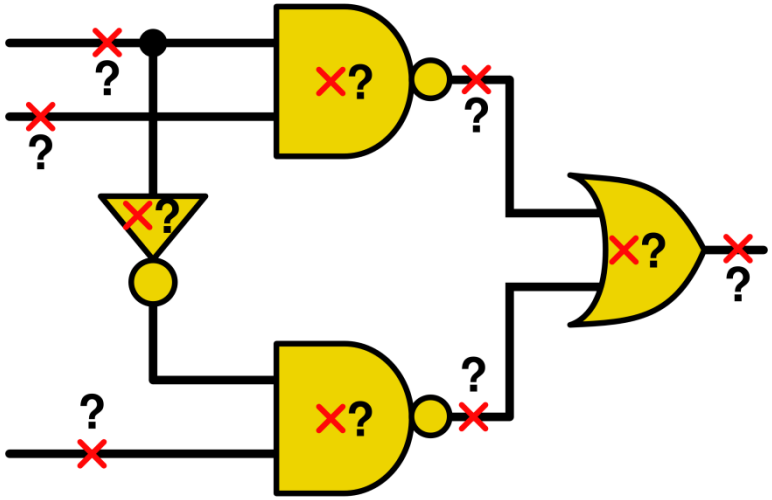
- Diagnostic precision does not need to surpass the RRA-level



Diagnosis Requirements

Conventional diagnosis

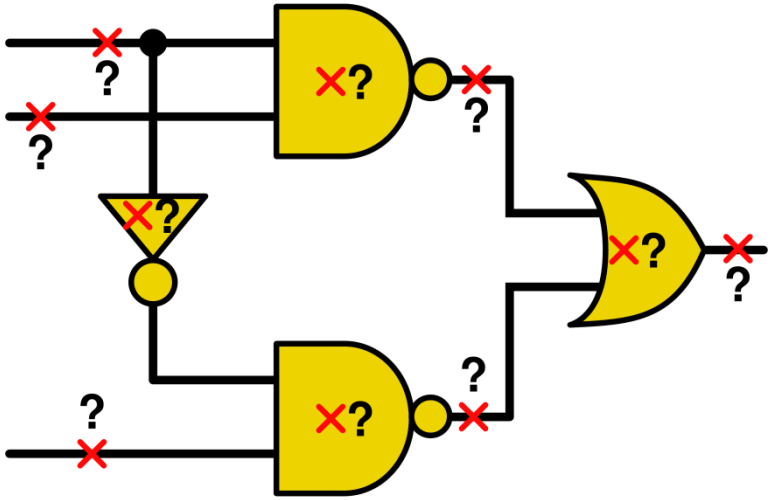
- Identify faulty gate/signal
- Fine-grained
- Failure analysis
- Improving DMT



Diagnosis Requirements

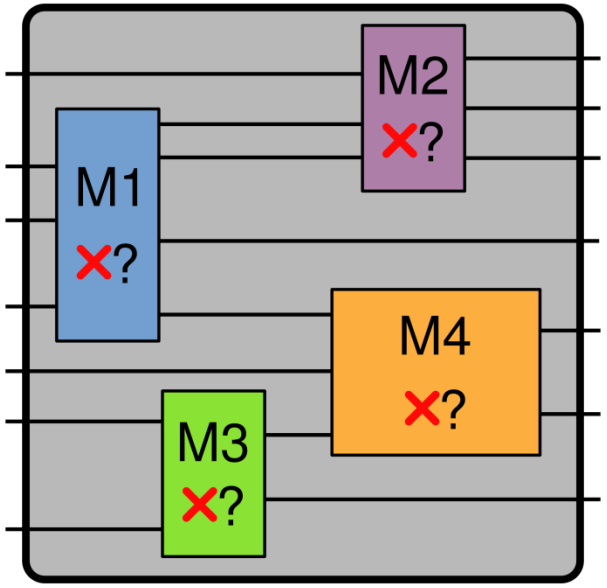
Conventional diagnosis

- Identify faulty gate/signal
- Fine-grained
- Failure analysis
- Improving DMT



Module-level diagnosis

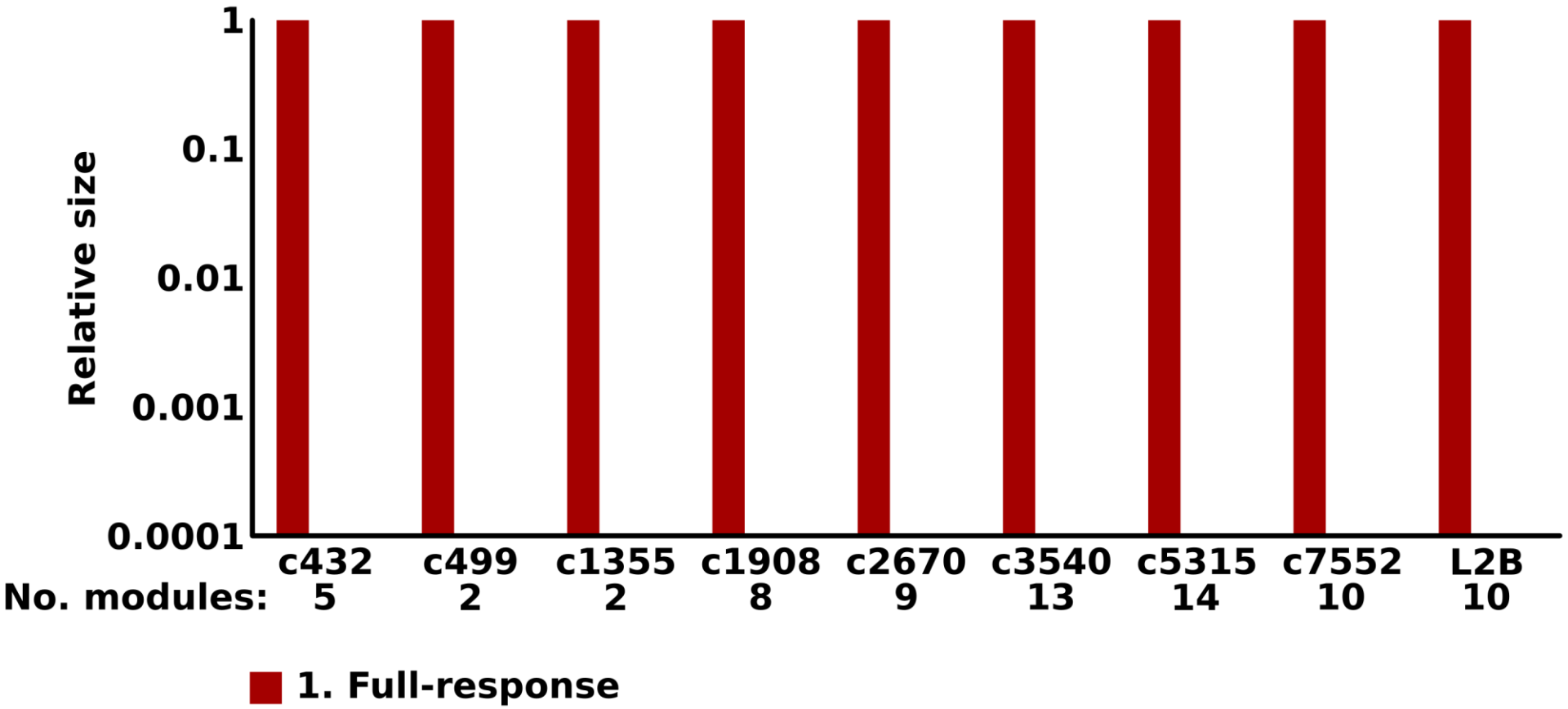
- Identify faulty module
- Coarse-grained
- Repair, replace, avoid
- Suitable for on-line use



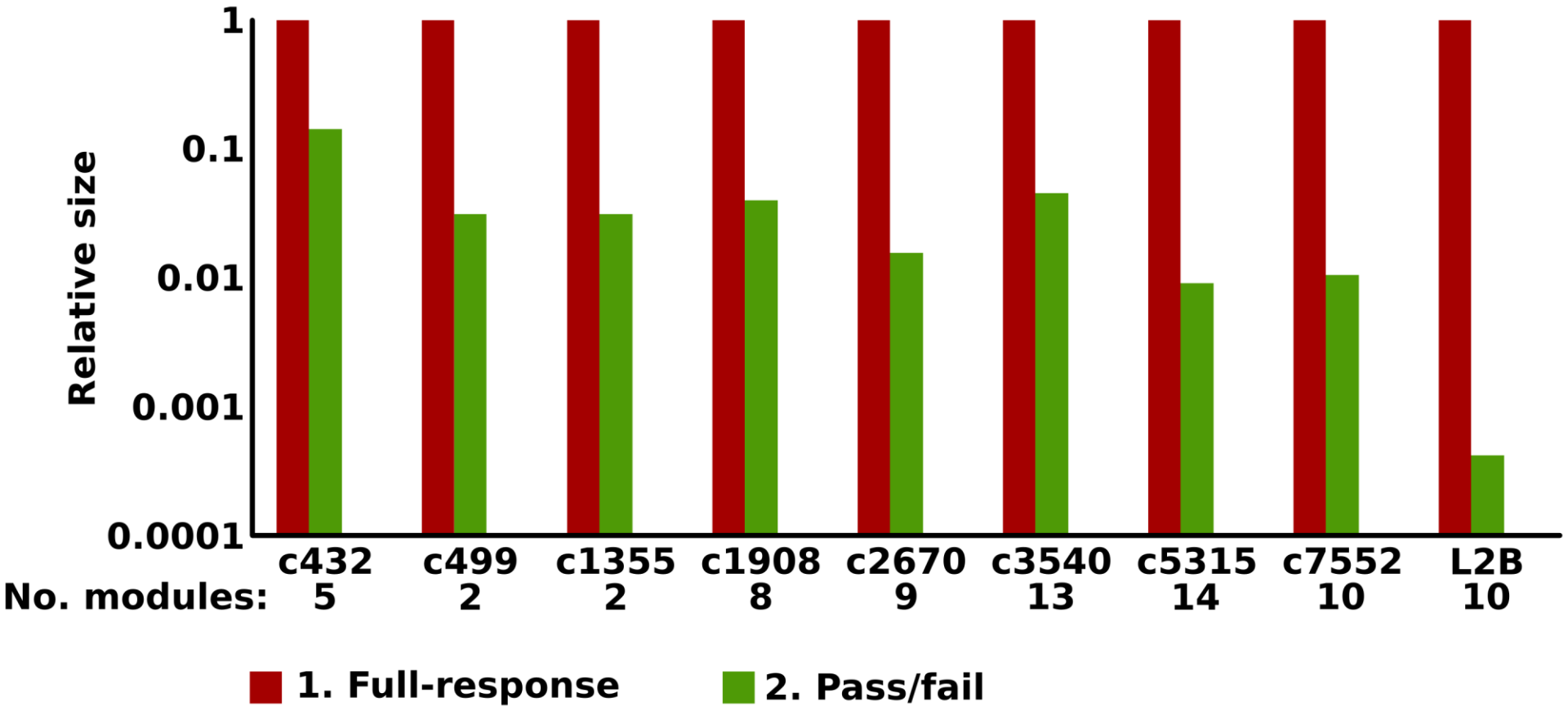
High-Level Fault Dictionaries

- Used relaxed precision to compact dictionary
  - Must still detect **all** faults (cannot remove tests)
  - **No need to distinguish intra-module faults**
  - i.e., collapse faults via equivalence and subsumption
- Benefits of a smaller fault dictionary
  1. Less off-chip storage required
  2. Less time to load dictionary for on-chip use
  3. Less time for performing on-chip diagnosis

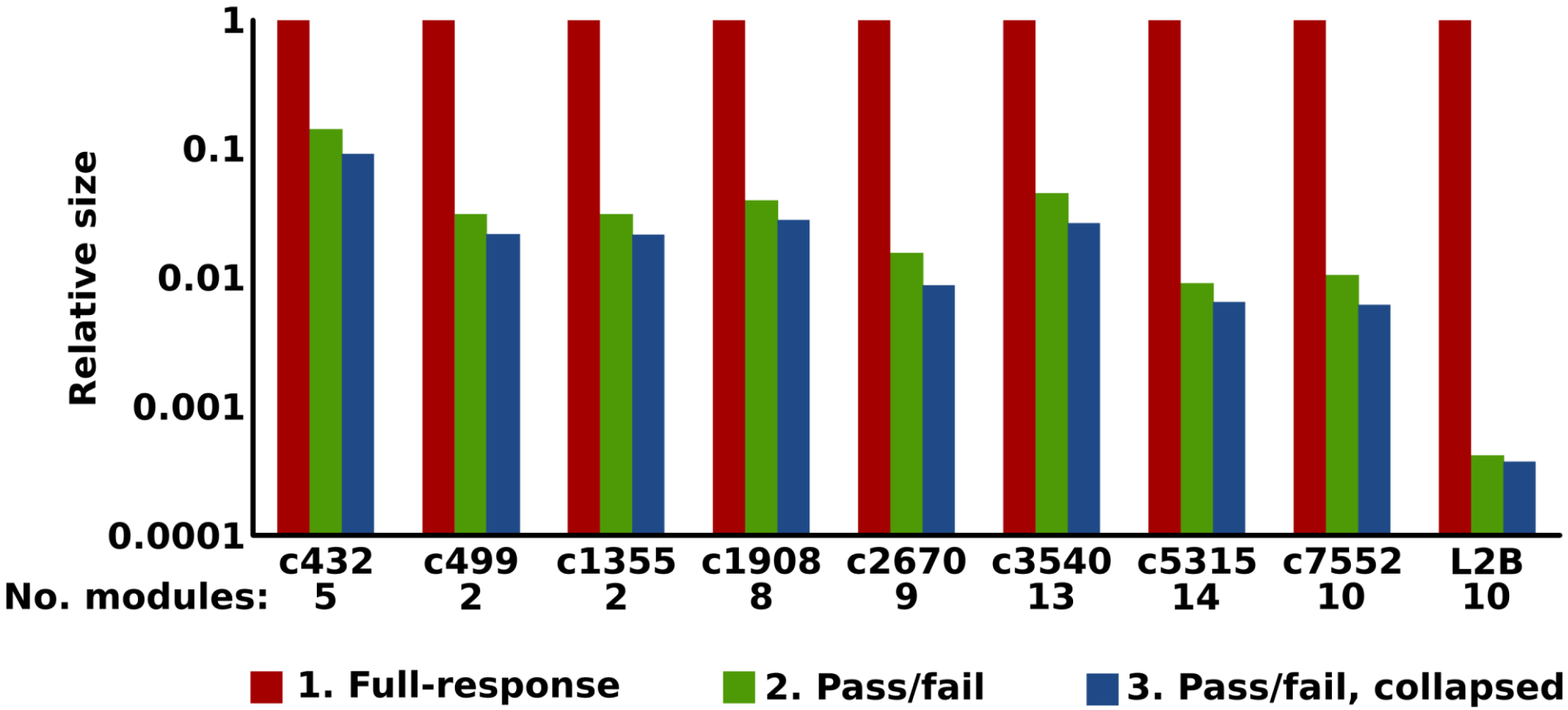
Fault Dictionary Compaction Results



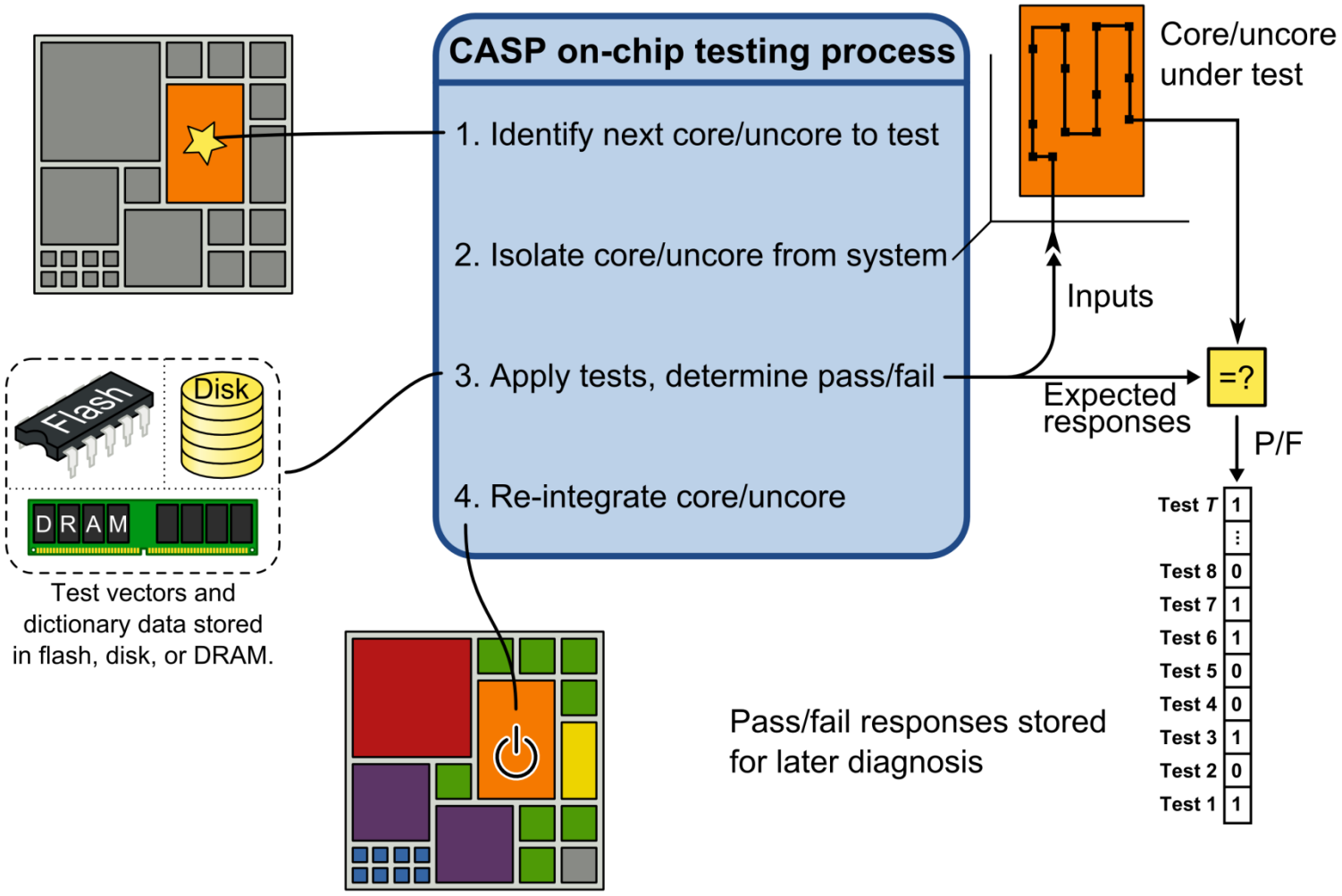
Fault Dictionary Compaction Results



Fault Dictionary Compaction Results

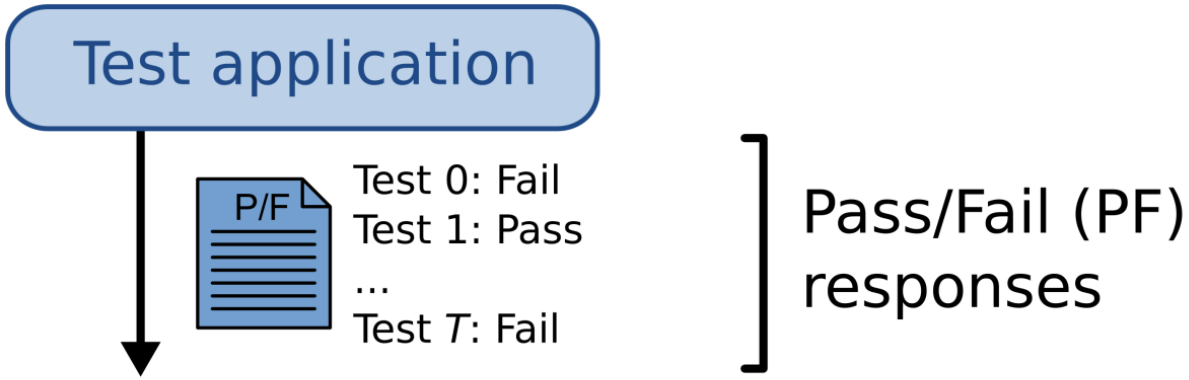


Current Work No. 3 - On-Chip Architecture

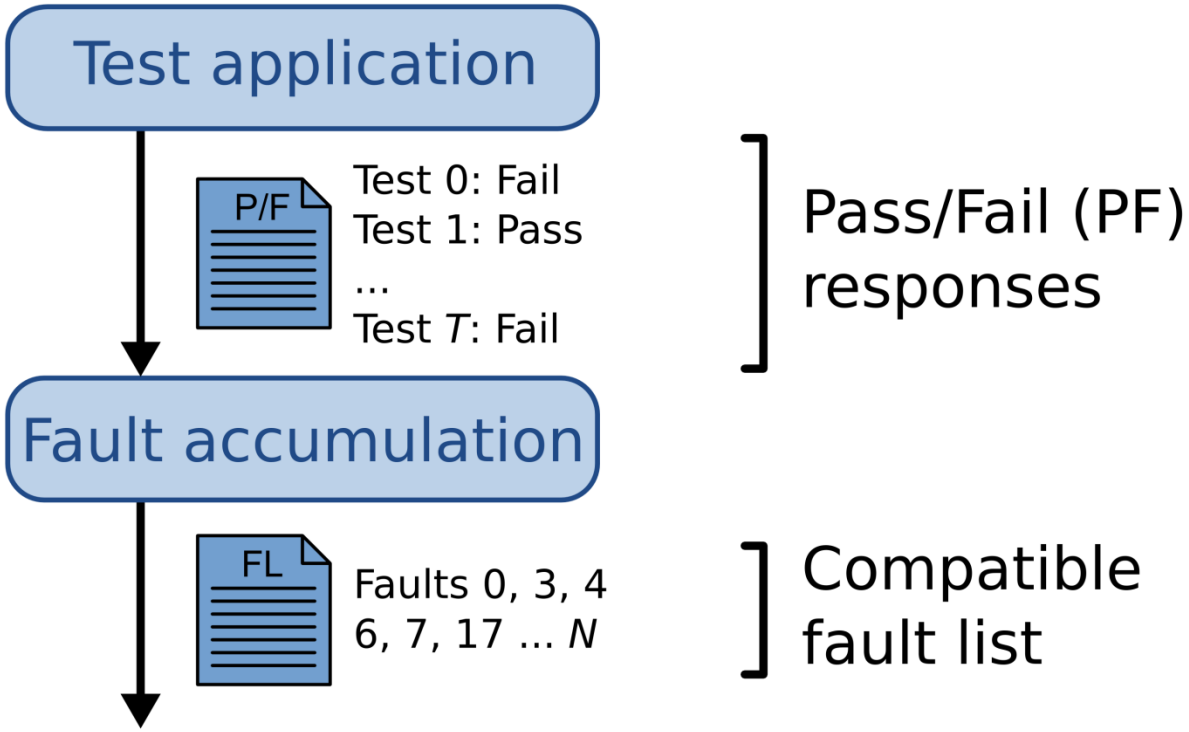


Y. Li, S. Makar, and S. Mitra, "CASP: Concurrent Autonomous Chip Self-Test Using Stored Test Patterns,", IEEE/ACM Design, Automation and Test in Europe, pp. 885-890, 2008.

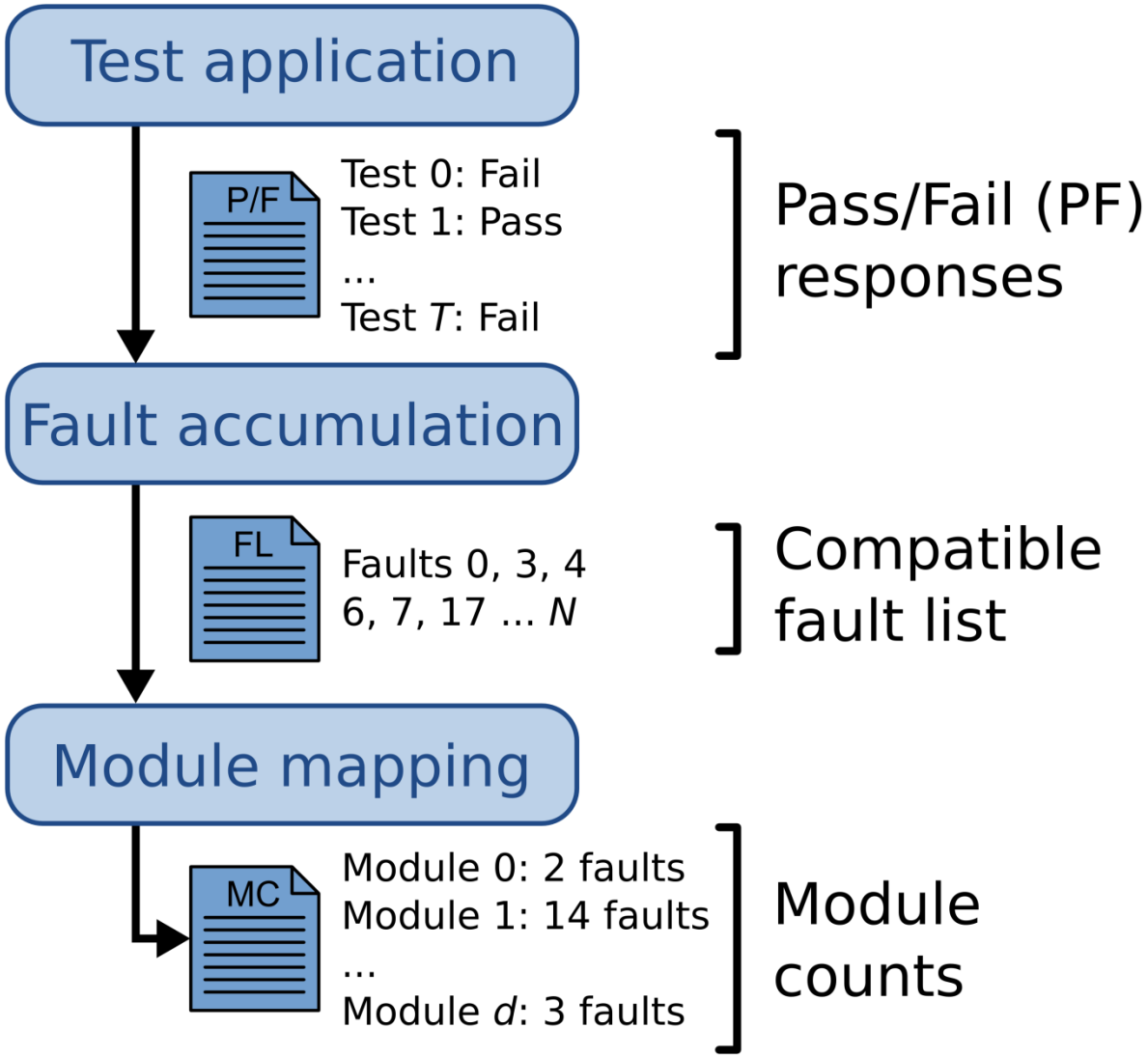
Test and Diagnosis Process



Test and Diagnosis Process

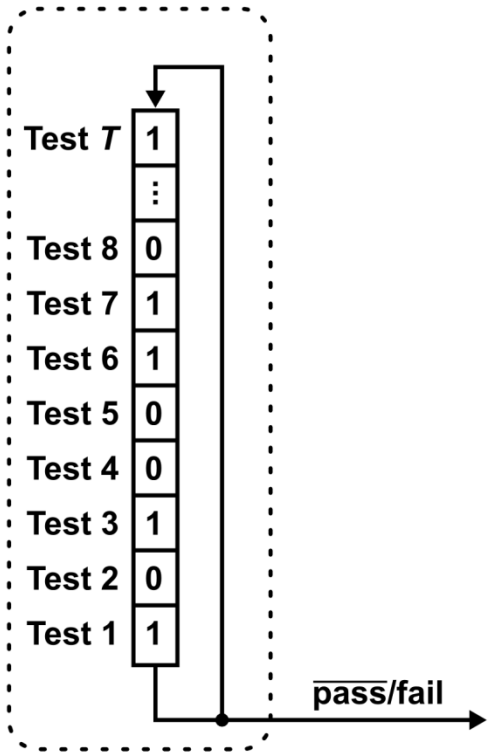


Test and Diagnosis Process

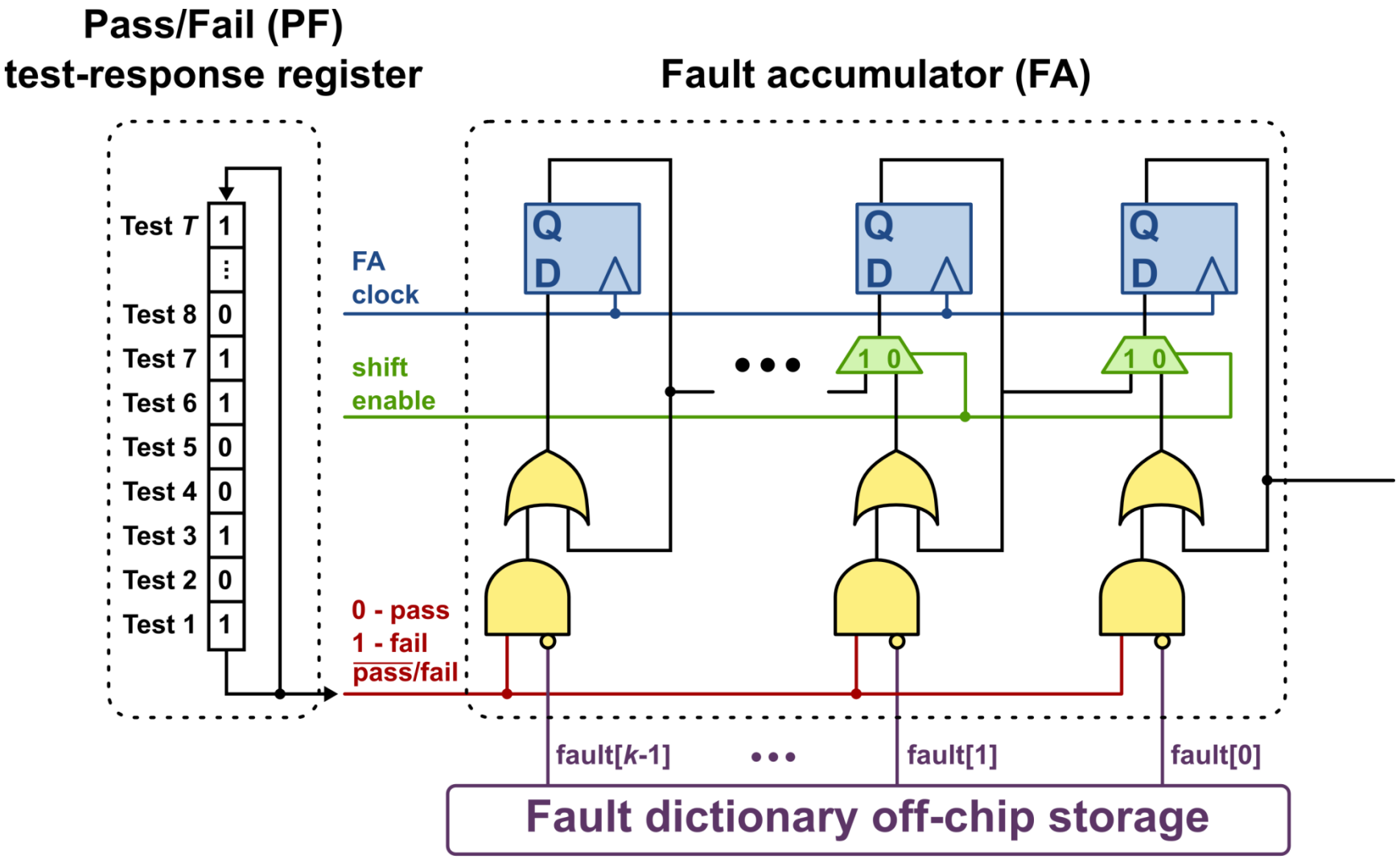


Diagnosis Architecture

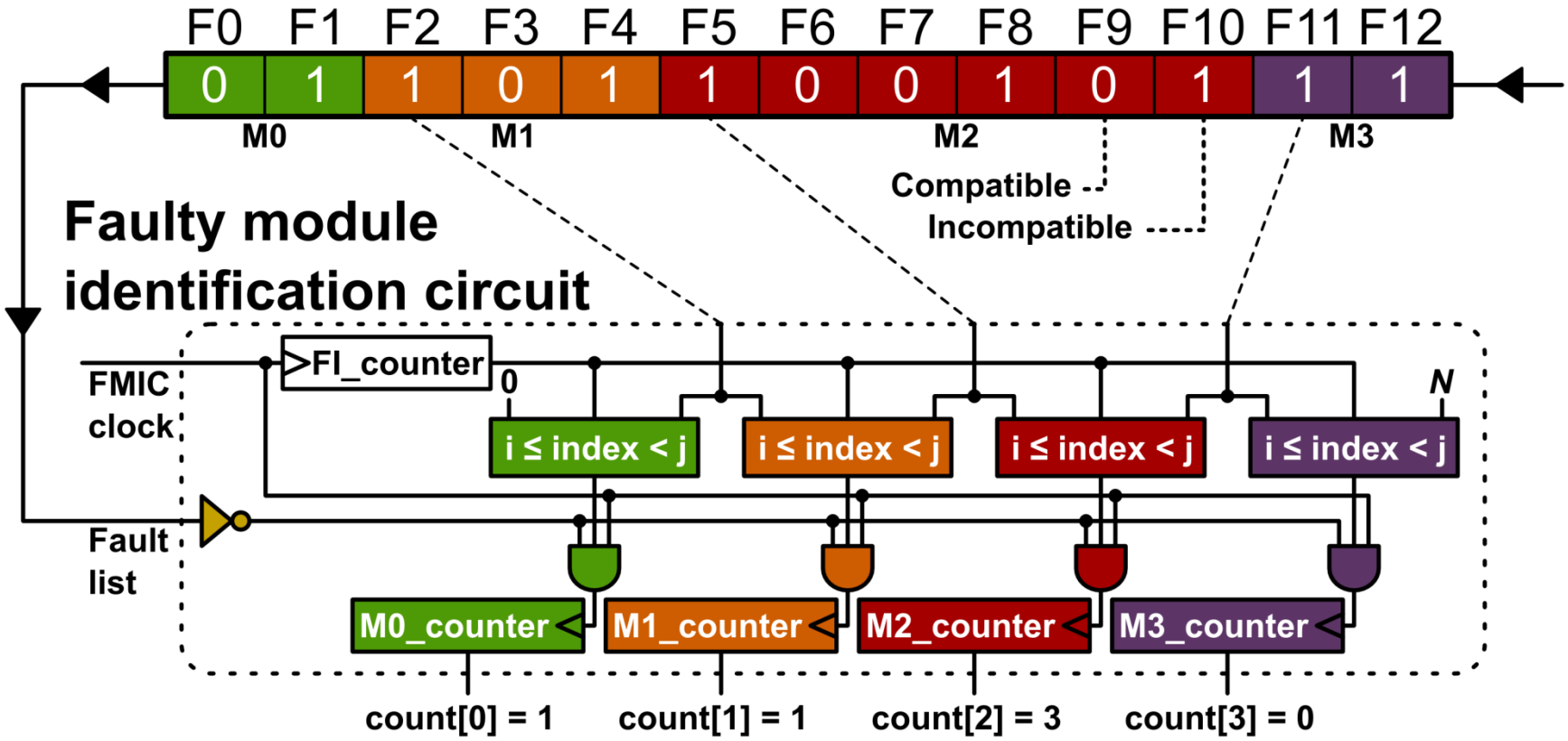
Pass/Fail (PF)
test-response register



Diagnosis Architecture cont…

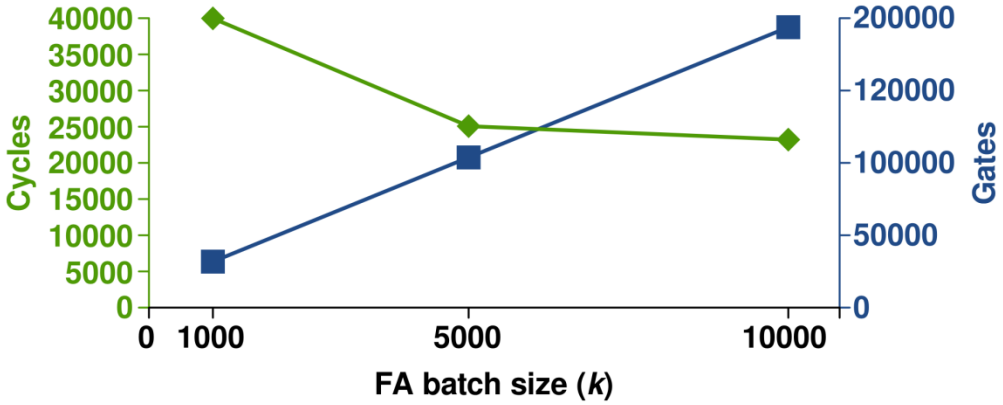


Diagnosis Architecture cont…



Diagnosis Hardware overhead

- Processing overhead:
  - About 25,000 cycles for diagnosis (k = 5000)
  - Can ideally be performed in the background
- Area overhead:
  - $(15 \times F \times M_{max} + 16 \times M_{max} + 13 \times F - 11) + (12 \times T_{max} + 4) + (18 \times k - 3)$
  - About 100k gates, (i.e. 0.12% of T2 design) (k = 5000)



Tmax	Max no. of tests	931
Mmax	Max no. of modules	10
Fmax	Max no. of faults	25514
F	$\log_2$(Fmax)	15
k	No. faults in parallel	5000

Outline

- Motivation
- Current Work
  - TRAnsition-X Fault Model (TRAX)
  - Hierarchical Fault Dictionary
  - Test and Diagnosis Architecture
  - Validating Experiment
- Proposed Work
- Summary
- Discussion

Experiment Overview

- Gate-injected delay faults in L2B
- Commercial ATPG for 2-pattern TF tests
- 99.95% TF coverage
- Experiment process:

```
simulate all tests to determine clock
for i in number_of_experiments:
  select TF site randomly
  while (no output discrepancy detected):
    increase delay in gate, apply tests
  record responses, perform diagnosis
```

Experiment Results

- **Example diagnosis result:**

M1: 35	M2: 74	M3: 97	M4: 23	M5: 12	M6: 00	M7: 17	M8: 03

- **Accuracy**
  - Accurate = injected module has a non-zero count
  - Ideal accuracy: Injected module has largest count
- **Resolution**
  - Resolution = no. of modules with non-zero fault count
  - Ideal resolution: Only one module with non-zero count

Experiment Results

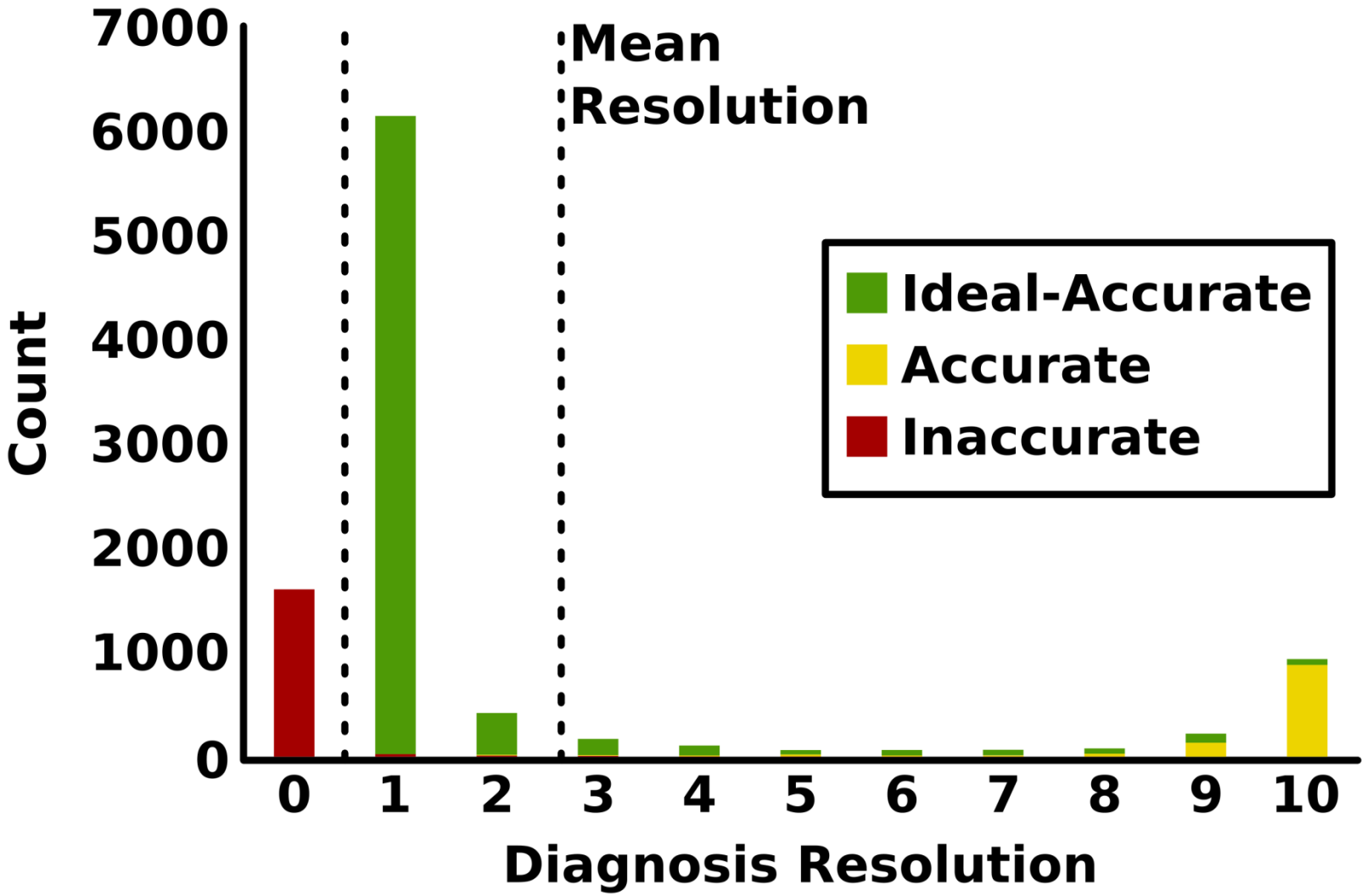
L2B: 931 tests, 25514 faults, 10 modules

10,000 injected faults

Empty diagnoses	**15.82%**
Accurate diagnoses*	**99.05%**
Ideal accurate diagnoses*	**73.96%**
Mean resolution*	**2.68**
Ideal resolution*	**73.13%**

****non-empty diagnoses only***

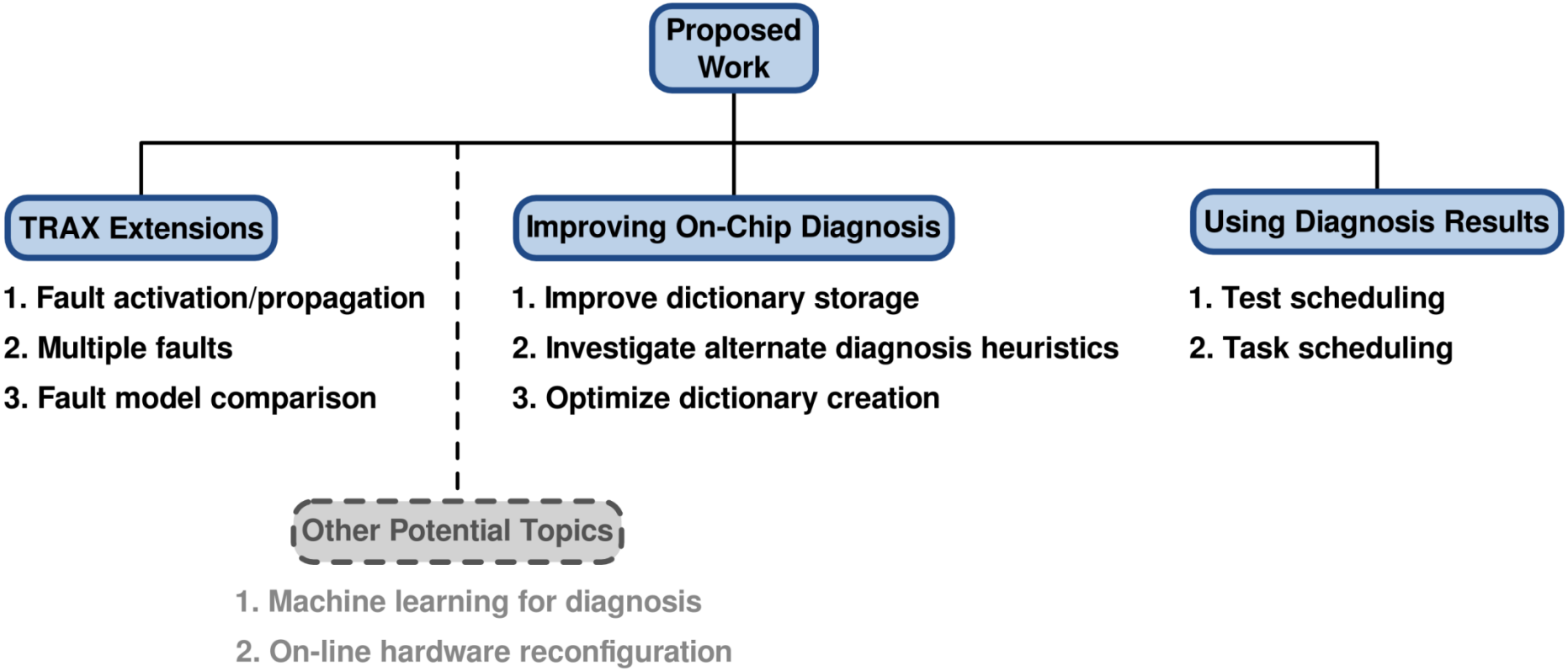
Experiment Results



Outline

- Motivation
- Current Work
- Proposed Work
  - TRAX Fault Model Improvements
  - Improving On-Chip Diagnosis
  - Using Diagnosis Results
  - Other Potential Topics
- Summary
- Discussion

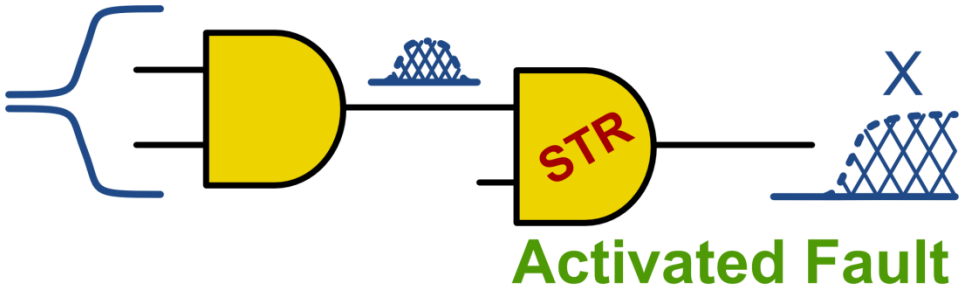
Proposed Work



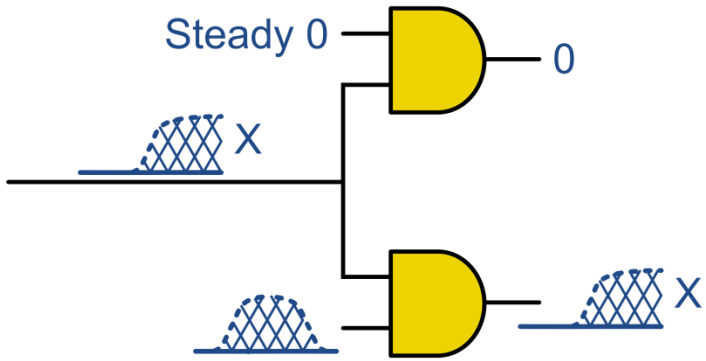
Proposed Work No. 1 – TRAX Improvements

Extensions to improve and extend TRAX model

1. Fault activation and propagation requirements
   A. Activation due to indirect hazards not yet supported



   B. Fault effect propagation too limited in current tools



Proposed Work No. 1 – TRAX Improvements

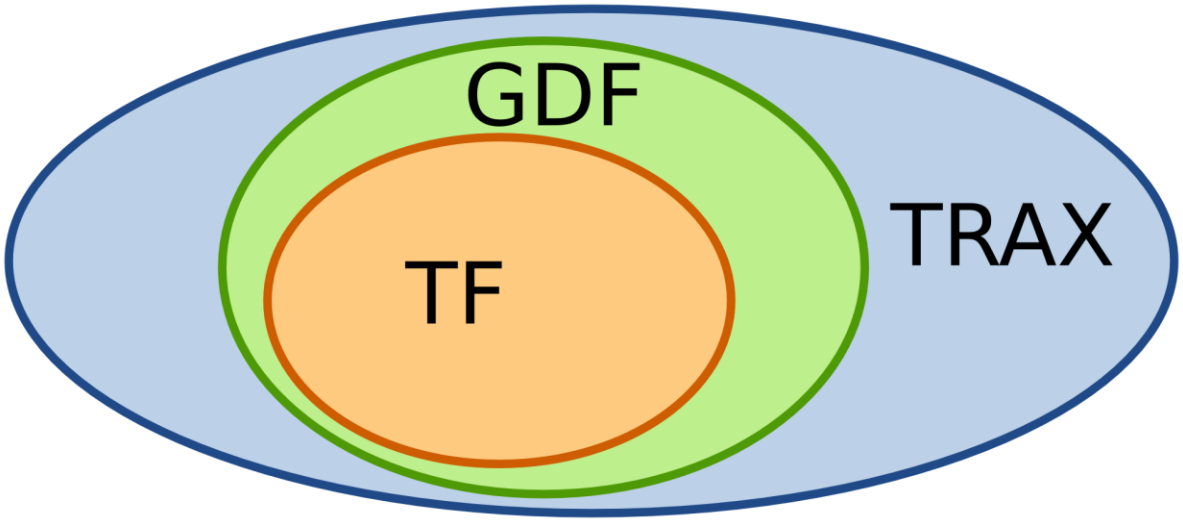
Extensions to improve and extend TRAX model

2. Multiple Faults
   - Most paths are “nearly critical” paths in modern design
   - Aging effects likely show spatial correlation
   - How does TRAX perform in presence of multiple faults?
   - If poorly, can we enhance model and diagnosis?

Proposed Work No. 1 – TRAX Improvements

Extensions to improve and extend TRAX model

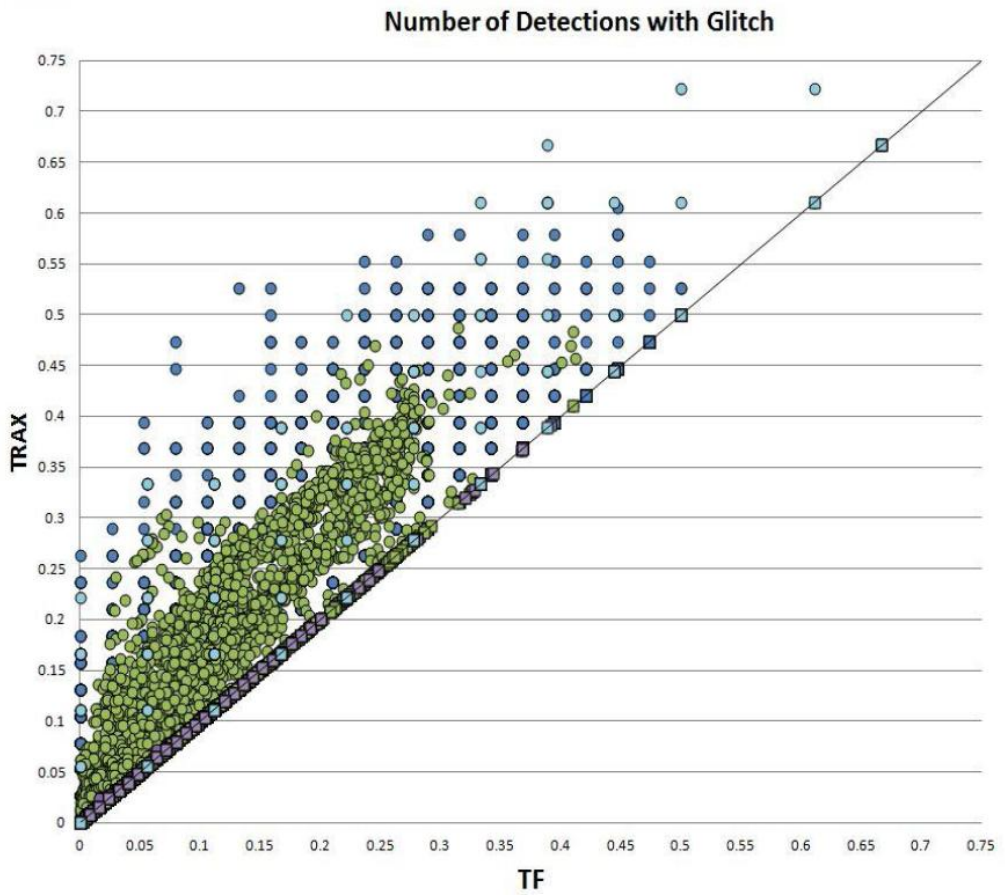
3. Comparison of TRAX with other delay-fault models
   - TRAX is a very conservative (general-purpose) fault model
   - Predicted faulty behavior should be a superset of others'
   - Propose empirical as well as theoretical analysis



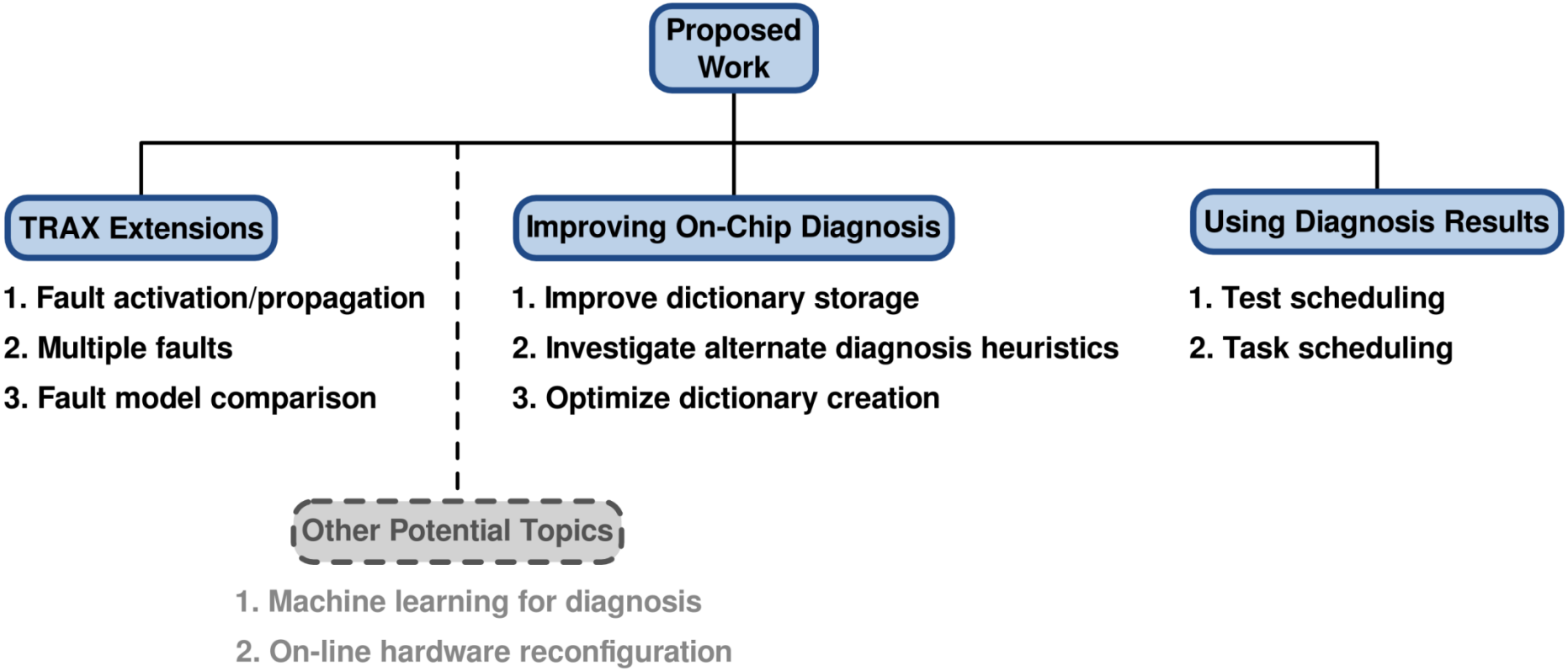
Proposed Work No. 1 – TRAX Improvements

Extensions to improve and extend TRAX model

3. Comparison of TRAX with other delay-fault models
   - Initial results comparing TRAX and TF, for multiple circuits

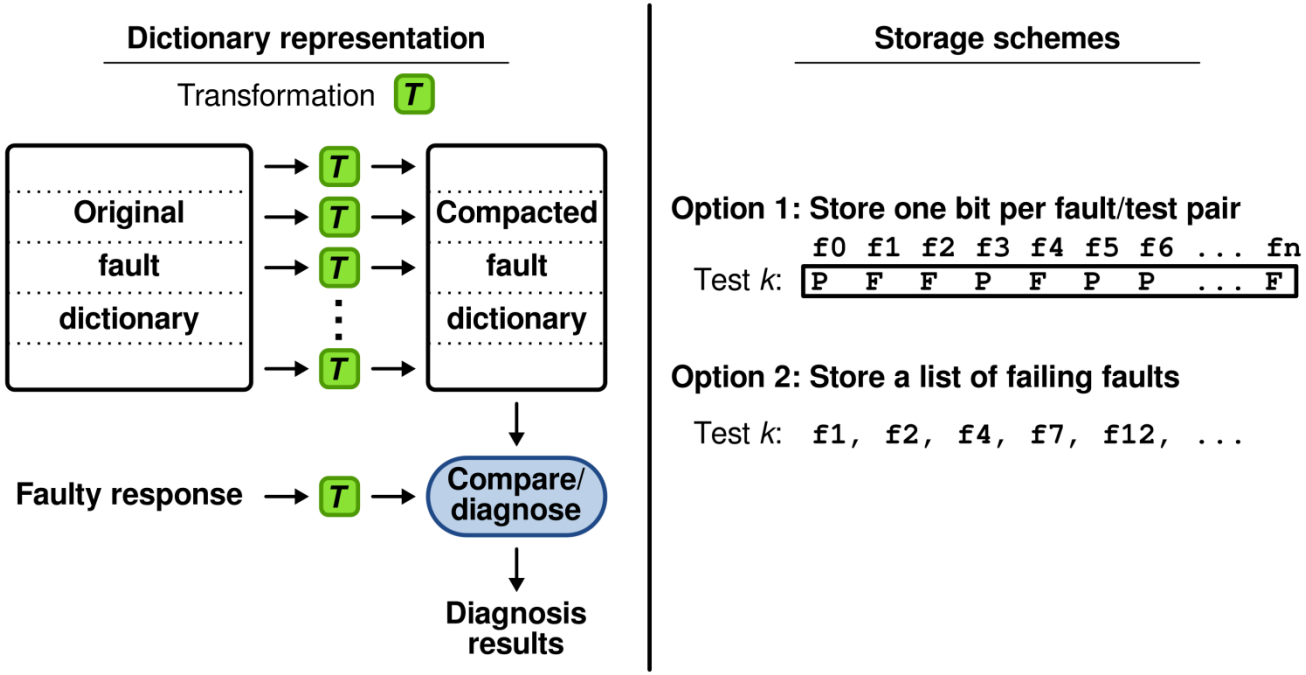


Proposed Work



Proposed Work No. 2 – Improving On-Chip Diagnosis

1. Fault Dictionary Storage
   - Dictionary compaction techniques (P-F, etc)
   - Dictionary storage schema (e.g., list of faults vs table)



Proposed Work No. 2 – Improving On-Chip Diagnosis

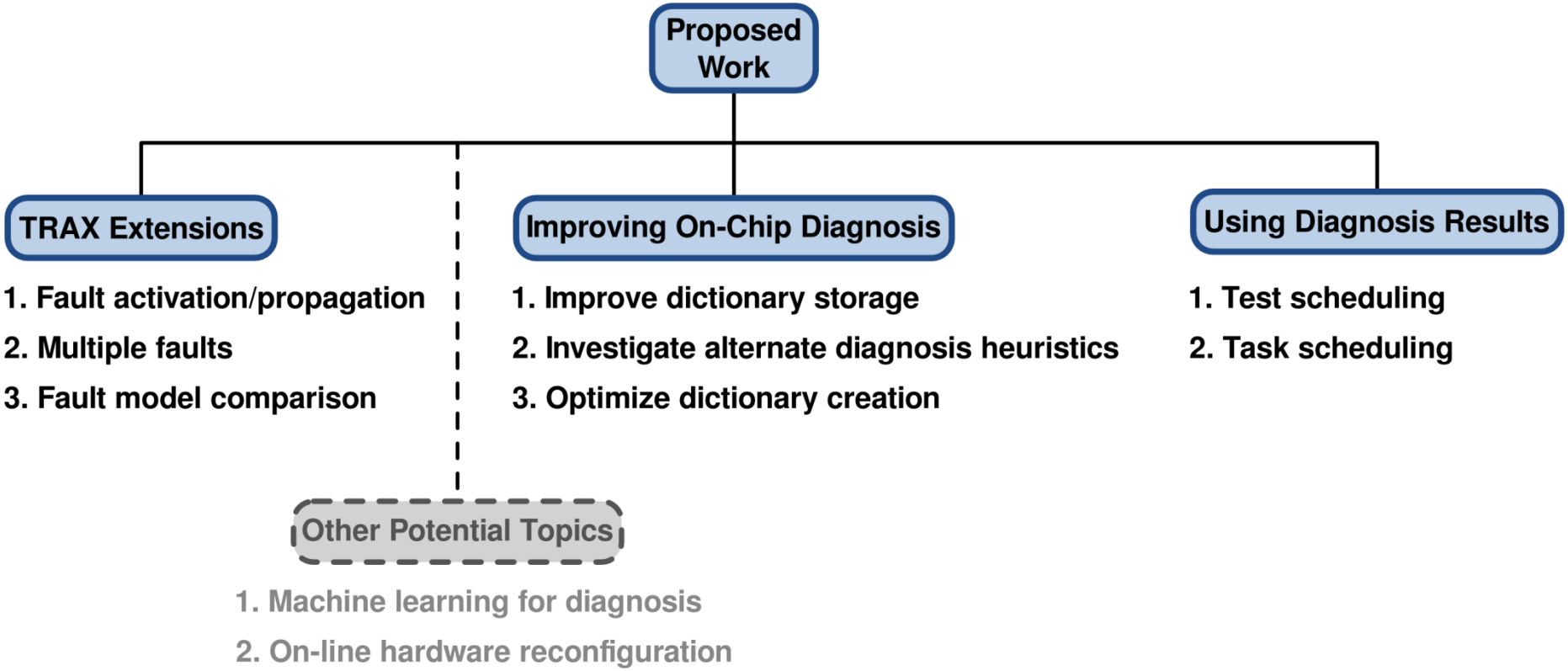
2. Alternate Diagnosis Heuristics
   - Currently, only TFSP used to eliminate faults
   - Perhaps track all four below and pass to OS
   - Other potential heuristics

Fault Simulation	Circuit under test		Conclusion
Pass	Pass	TPSP	Ok
Pass	Fail	TFSP	**Inconsistent: Remove Fault**
Fail	Pass	TPSF	Ok
Fail	Fail	TFSP	Ok

Proposed Work No. 2 – Improving On-Chip Diagnosis

3. Dictionary Creation Optimization
   - Eliminate inefficiencies in FATSIM fault simulator
   - Advanced computational resources:
     - Condor work sharing already in limited use
     - Graphics Processing Unit (GPU) fault simulation
       - Course project in fall 2012 to do TRAX simulation on GPU
       - Achieved 5-12x increased performance w.r.t reference
       - Novel features of GPU fault simulator implementation:
         - » No fault dropping
         - » Complex TRAX fault activation conditions

Proposed Work



Proposed Work No. 3 – Using Diagnosis Results

- Use diagnosis results to improve operation
- These ideas serve to broaden the focus of thesis

1. Test Scheduling
   - Currently, test at fixed intervals or use simple metrics
   - If an accelerated test fails, test this core again soon
   - Could affect other test options, like test set selection
   - Investigations would center around system-level simulation coupled with core/uncore aging models

Proposed Work No. 3 – Using Diagnosis Results

2. Task Scheduling
   - Provide diagnosis results to OS process scheduler
   - Avoid stressed modules showing signs of aging
   - Use system simulator with wear models for individual cores/uncores
   - Metrics:
     - User-perceived performance loss
     - Occurrence of data corruption
     - Number of cores with pending failures

Other Potential Topics

1. **Use of machine learning to improve diagnosis**
   - Conservative nature can result in degraded resolution
   - Train ML classifier on diagnosis and other on-chip data
   - Classifier predicts correct classification
   - Track past predictions until repair/replacement determines the actual faulty module
   - Update classifier on-line with this new data point
   - Investigate HW/SW trade-offs for ML implementation

Other Potential Topics

2. On-Line Hardware Reconfiguration

- Diagnosis must produce action to ensure robustness
- Repair, replacement, avoidance not directly addressed
- One recent methodology uses switches and muxes to select between multiple identical copies of logic block*
- Only used for manufacturing-time yield improvement
- Investigate use of similar techniques for on-line use

* M. Mirza-Aghatabar, M. Breuer, S. Gupta, and S. Nazarian, "Theory of Redundancy for Logic Circuits to Maximize Yield/Area," in *Quality Electronic Design (ISQED) 2012, pp663-671, March 2012.*

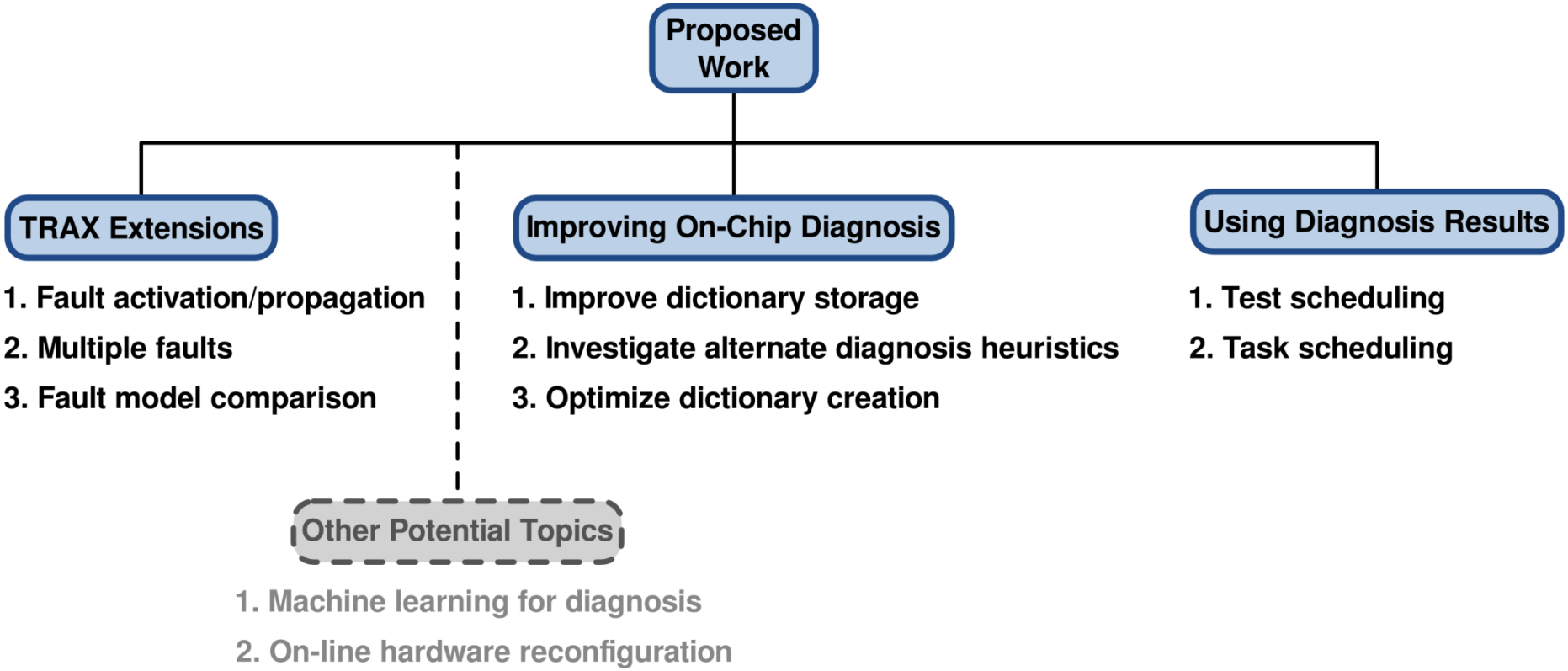
Outline

- Motivation
- Current Work
- Proposed Work
- Summary
  - Timeline
- Discussion

Summary

- Existing work focused around on-chip diagnosis
  - TRAX fault model targets wear-out and aging failures
  - High-level fault dictionary diagnoses to repair level
  - Scalable on-chip hardware for fault diagnosis
- Future work to extend and broaden existing work
  - Enhancements to TRAX fault model
  - Improving the on-chip diagnosis process
  - Use diagnosis results to improve system operation
  - Additional topics as time permits, or in case of trouble

Summary


Proposed Work
TRAX Extensions
1. Fault activation/propagation
2. Multiple faults
3. Fault model comparison
Improving On-Chip Diagnosis
1. Improve dictionary storage
2. Investigate alternate diagnosis heuristics
3. Optimize dictionary creation
Using Diagnosis Results
1. Test scheduling
2. Task scheduling
Other Potential Topics
1. Machine learning for diagnosis
2. On-line hardware reconfiguration

Estimated timeline

TRAnsition-X Fault Model (TRAX)	
Fault Activation and Propagation Conditions	4 weeks
Multiple Faults	4 weeks
Fault Model Comparison	2 weeks
Improving On-Chip Diagnosis	
Fault Dictionary Storage	4 weeks
Alternate Diagnosis Heuristics	4 weeks
Dictionary Creation Optimization	3 weeks
Using Diagnosis Results	
Test Scheduling	6 weeks
Task Scheduling	6 weeks
Thesis Writing	12 weeks
Total	11 months

Thank you for your time!