

Achieving Lower Test Pattern Count Through Deterministic Test Point Insertion

Andrew Ferko

Kenneth Pichamuthu

Prakash Venkitaraman*

IBM Microelectronics

**Now working at AMD Bangalore*

Problem definition - Hard to test

- Increased density and complexity of logic
- Increase in test cost and volume
- Increase in test generation time
- Test quality could suffer in the trade-off

Definition of terms

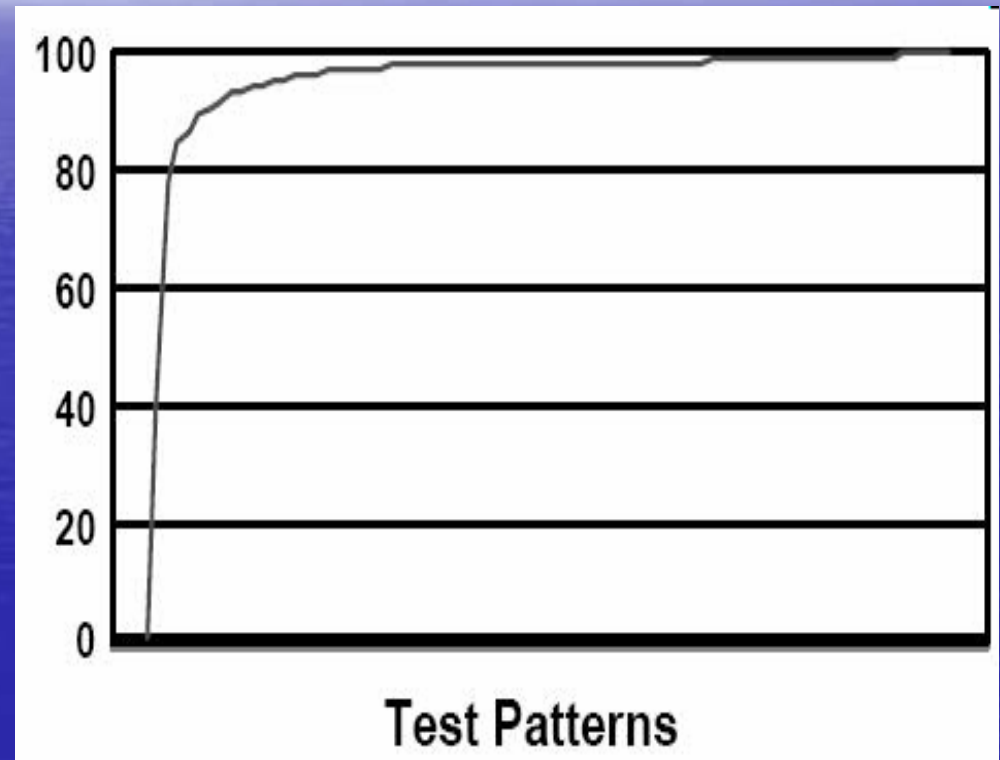
- Focal fault
- Care bits
- Measure bits
- RRFA (Random (pattern) Resistant Fault Analysis)

Current solutions

- Most discussed solutions improve pseudo-random testability => higher coverage
- Better ATPG => Lesser Pattern Count
- Compatibility of the cost function used by the TPI algorithms and the ATPG engine
- Controllability and observability indices when there is reconvergent fanout

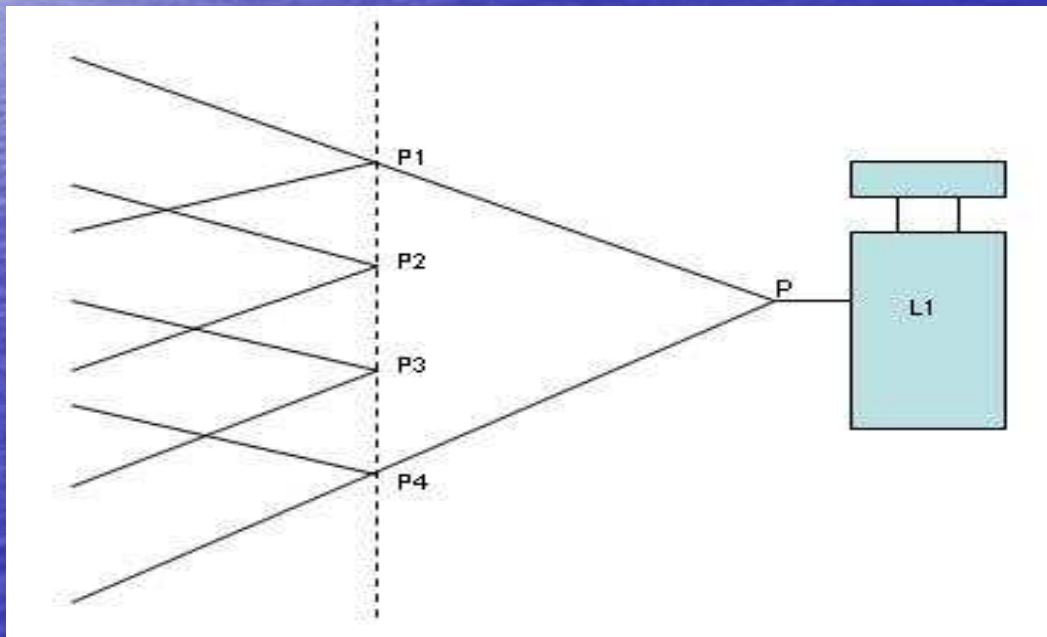
The Test Generation process

- Tests for focal faults merged to create patterns – care bits just 2% of total bits!!!
- Most bits filled with latch-fill – random first and repeat later.



Test Generation data – what we can learn from it?

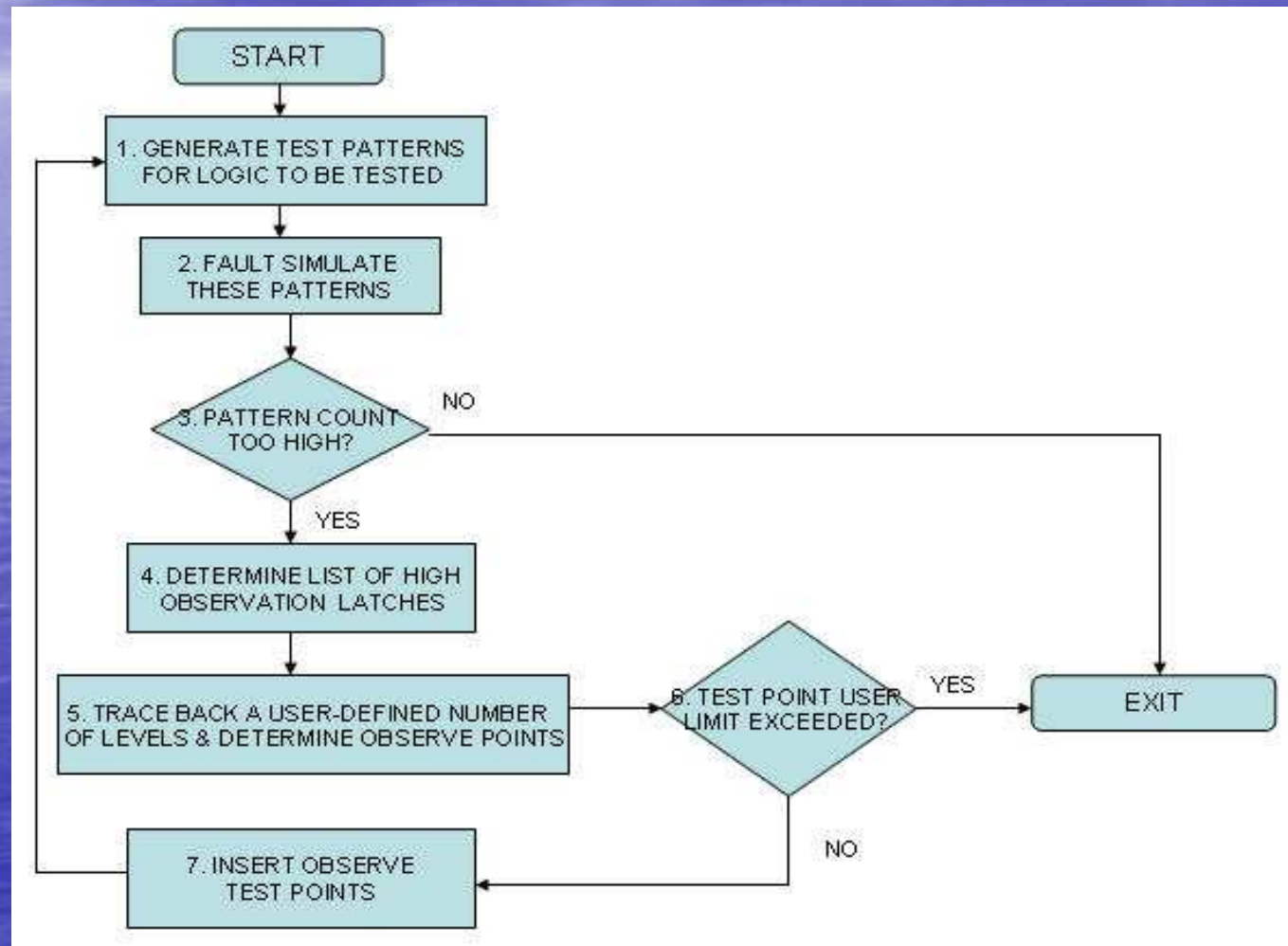
- Measure bits – observability issue?
- Logic cones that need to be broken up?



Stim bits

- Controllability issue
- Trace forward, add control points
- Cannot add too many control points

Iterations - flowchart



List of hard-to-test focal faults

- Feed back to the designers
- Simulate patterns that test them early

Results ...

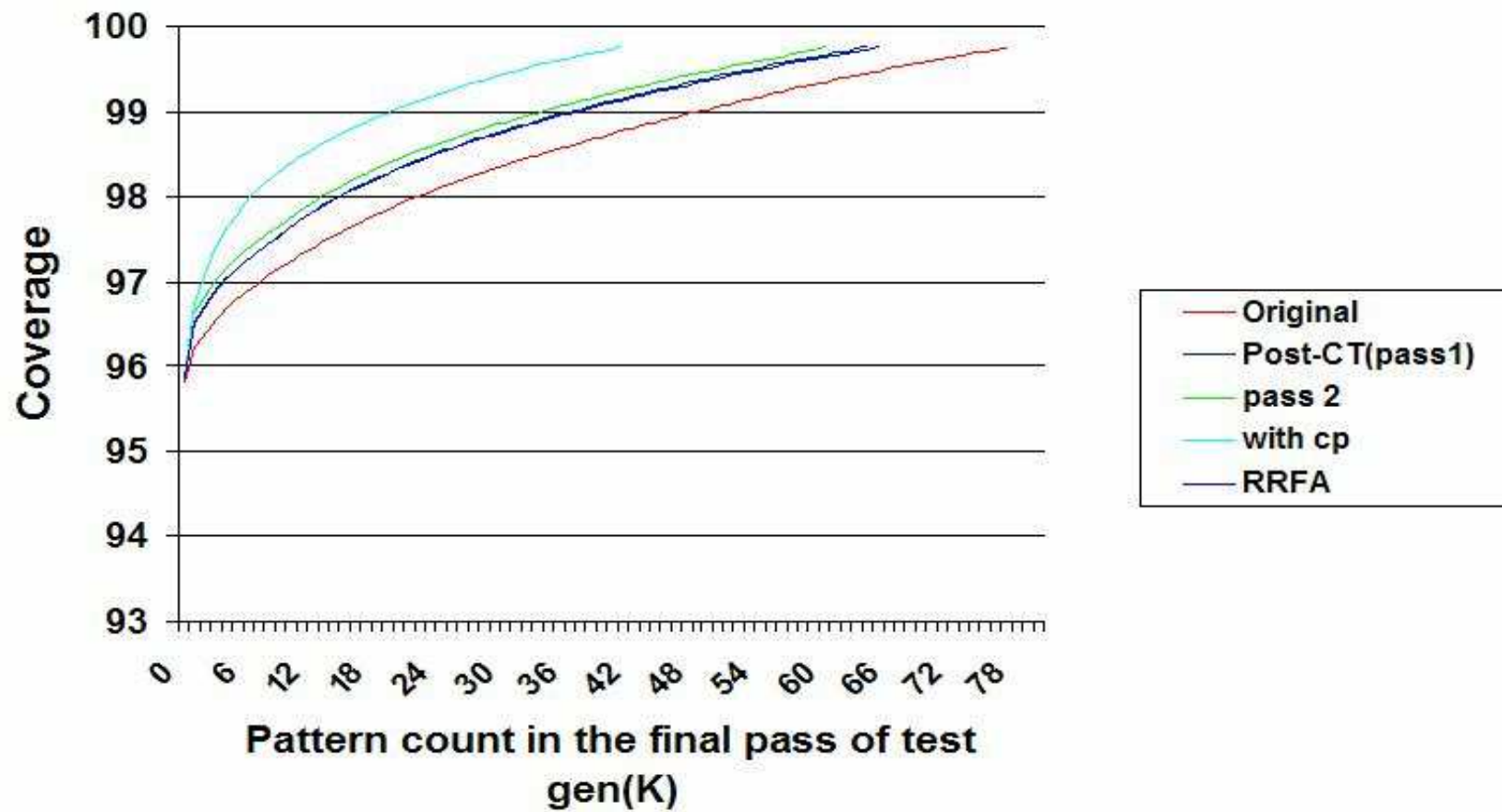
- 2.1 M instances, 115K latches
- 2 passes of observe point insertion
69 test points and then 210
- 1 pass of control point insertion
22 control points
- Comparison with RRFA-only test point insertion

.. Results ..

Experiment	Pattern count	Coverage %	Reduction
Original	77K	99.75	-
69 observe points	64K	99.76	17%
210 observe points	60K	99.76	22%
22 control points	41K	99.76	47%
RRFA alone	65K	99.76	16%

... Results

Caretaker



More results

- 1M instances
- Total of 423 observe points added over 3 passes
- Pattern count went down from 39K to 30K – a reduction of 23%
- No control points added

Future work

- Better method of identifying nodes to insert test points
- Quicker way to determine coverage increase for each test point added