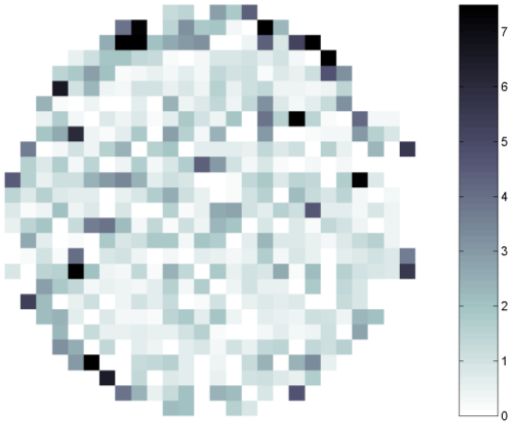
Advanced Topics: ATPG

- Introduction
- Defect-based Testing
- Advanced ATPG
  - N-detect ATPG (Stanford 1995)
    - Introduction
    - ATPG
    - Experimental Results
  - Cell-aware ATPG (Mentor 2009)
  - Timing-aware ATPG (Mentor 2006)
  - Power-aware ATPG (KIT 2005)
- Conclusion



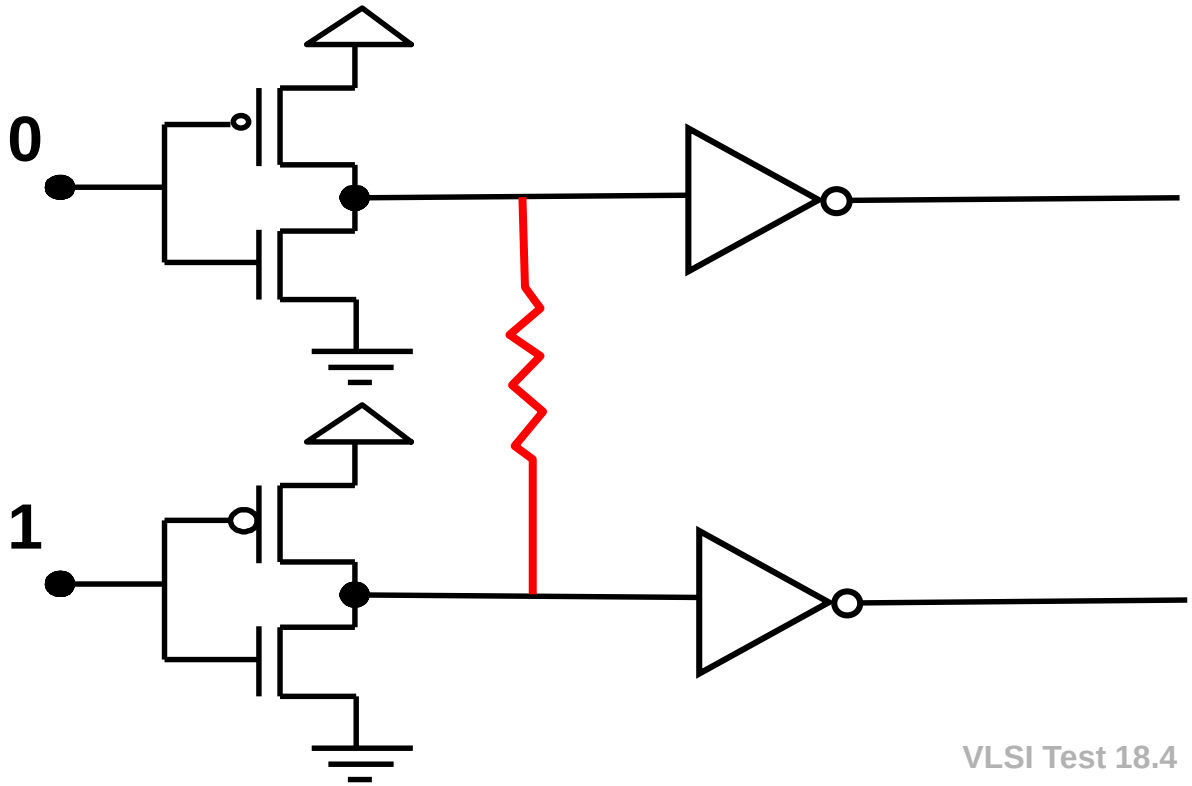
N-detect Test Set

- What is *N-detect* test set?
  - Detect each fault at least *N* times, if possible
  - Every detecting pattern can be different (but not required)
  - Can be applied to any fault model e.g. SSF, TDF
- Discovered by IBM and other companies
  - but first published by Stanford Murphy experiment [Ma 95]
- Why *N*-detect test set useful?
  - Improve DPM
    - Increase diversity of test patterns
    - Increase probability to detect un-modeled defects

$$\textbf{Fault Coverage}_{N-det} = \frac{number\ of\ faults\ \textbf{detected at least N times}}{number\ of\ total\ faults} \times 100\%$$

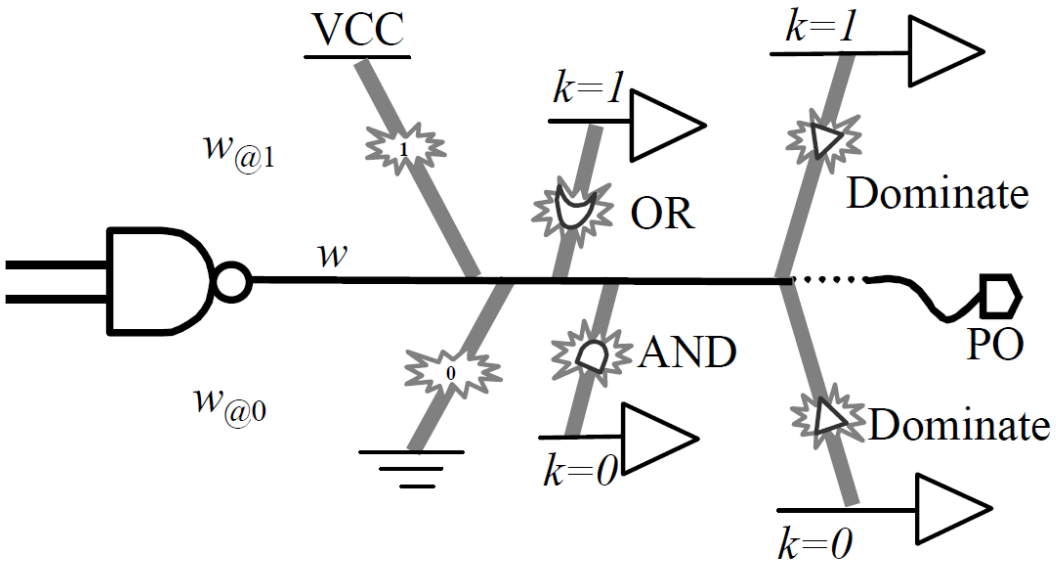
Example 1

- {00, 11} single-detect SSF test set
  - But does not detect bridging fault
- {00, 11, 01, 10 } 2-detect SSF test set
  - More likely to detect bridging fault



Example 2

- **Diversified test patterns help to detect different bridging faults**



[Benware 03]

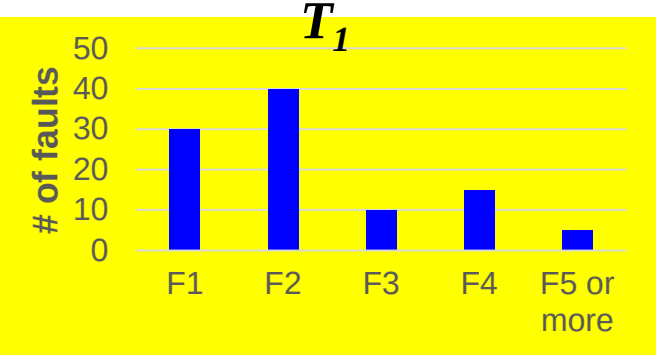
	w	k	OR	AND	D	V_{cc}	V_{ss}
$w_{@1}$	0	0	0	0	0	1	0
$w_{@1}$	0	1	1	0	1	1	0
$w_{@0}$	1	0	1	0	0	1	0
$w_{@0}$	1	1	1	1	1	1	0

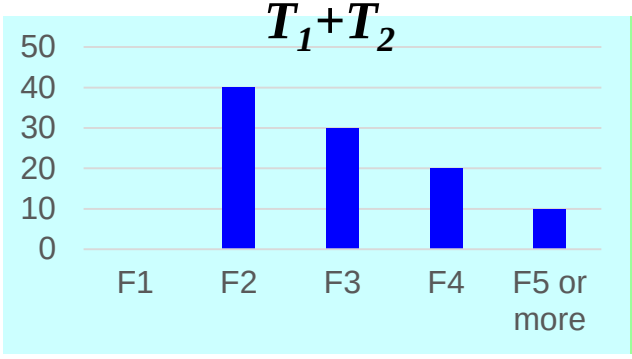
Detection condition highlighted

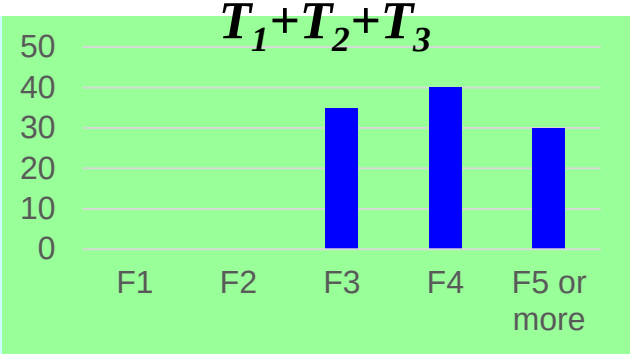
N-detect ATPG Algorithm [Benware 03]

N-detect ATPG (*N*) // *N* = number of detections

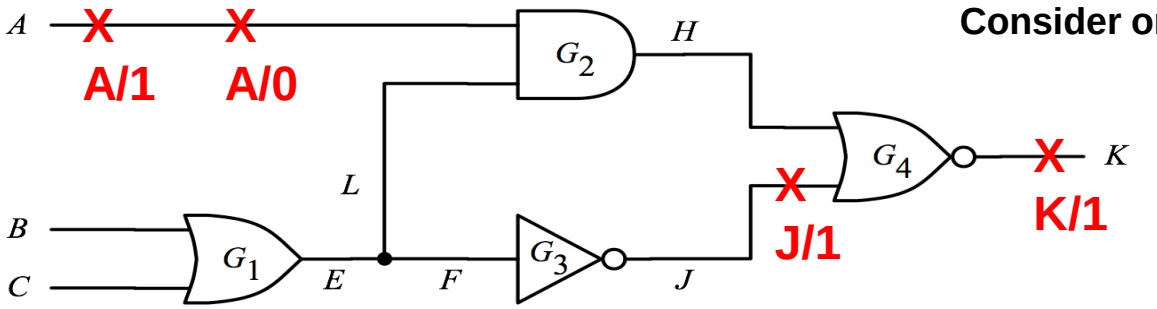
1 Perform single-detect ATPG obtain test pattern set T_1
2 TF_{MD} = all faults detected by single-detect fault simulation with T_1
3 **for** i = 1 to N-1
4 Perform multiple-detect fault sim with T_1 to T_i for TF_{MD} faults
5 Save faults detected exactly i times to F_i
6 Target faults F_i and perform single-detect ATPG
7 Save patterns to T_{i+1}
8 Perform multiple-detect fault simulation with T_1 to T_N for all faults to obtain multiple-detect fault coverage profile







Example (1/2)



Consider only 4 faults

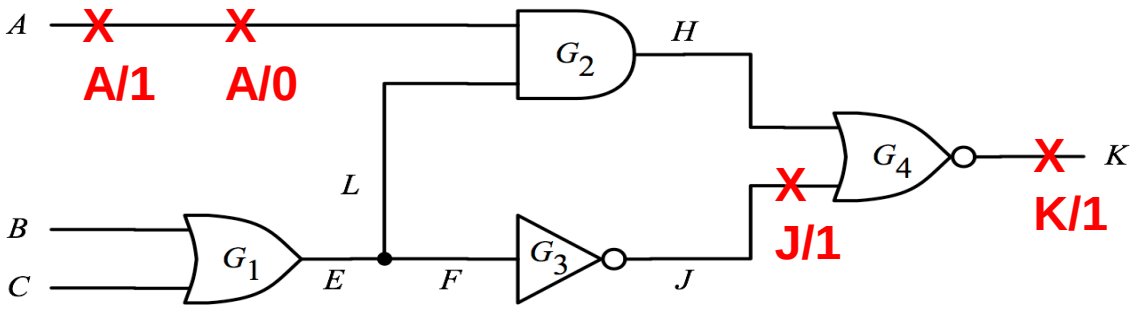
T_1	A	B	C	E	H	J	K	Detected faults
P_1	0	1	0	1	0	0	1	A/1, J/1
P_2	0	0	0	0	0	1	0	K/1
P_3	1	1	1	1	1	0	0	A/0, K/1

F_1 ={A/1, A/0, J/1} F_2 ={K/1} F_3 ={}

i = 1, target fault list = F_1

T_2	A	B	C	E	H	J	K	Detected faults
P_4	0	0	1	1	0	0	1	J/1, A/1
P_5	1	1	0	1	1	0	0	A/0, K/1

Example (2/2)



T_2	A	B	C	E	H	J	K	Detected faults
P_4	0	0	1	1	0	0	1	J/1, A/1
P_5	1	1	0	1	1	0	0	A/0, K/1

F_1 ={} F_2 ={J/1, A/1, A/0} F_3 ={K/1}

i = 2, target fault list = F_2

T_3	A	B	C	E	H	J	K	Detected faults
P_6	0	1	0	1	0	0	1	J/1, A/1
P_7	1	0	1	1	1	0	0	A/0, K/1

7 Patterns, $FC_{3\text{-det}}$=100%

Stanford Experiments

- **Murphy experiment: 0.7 μm**
  - **Total 5.5K chips tested**
  - **116 defective chips**

minimum number of detects	SSF Tool 2			
	test length	characterized speed	slow speed (1/3)	very slow speed (1/30)
1	313	3	5	7
2	671	2	5	7
3	981	0	3	5
4	1,292	0	5	5
5	1,605	0	3	5
7	2,203	1	5	5
10	3,022	0	2	6
12	3,578	0	2	5
15	4,396	0	2	5

Number of test escapes [McCluskey 00]

- **ELF experiment: 0.35 μm**
  - **Total 10K chips tested 324 defective chips**

Tools			1C
SSF	N-Detect	15	0
		10	1
		5	2
		3	2
		2	3
	Fault	1.00	3
		0.99	2
		0.95	8
		0.90	9
		0.80	18
		0.50	91

Number of test escapes [McCluskey 04]

LSI/Mentor Experiments [Benware 03]

- **Test effectiveness increase with *N***

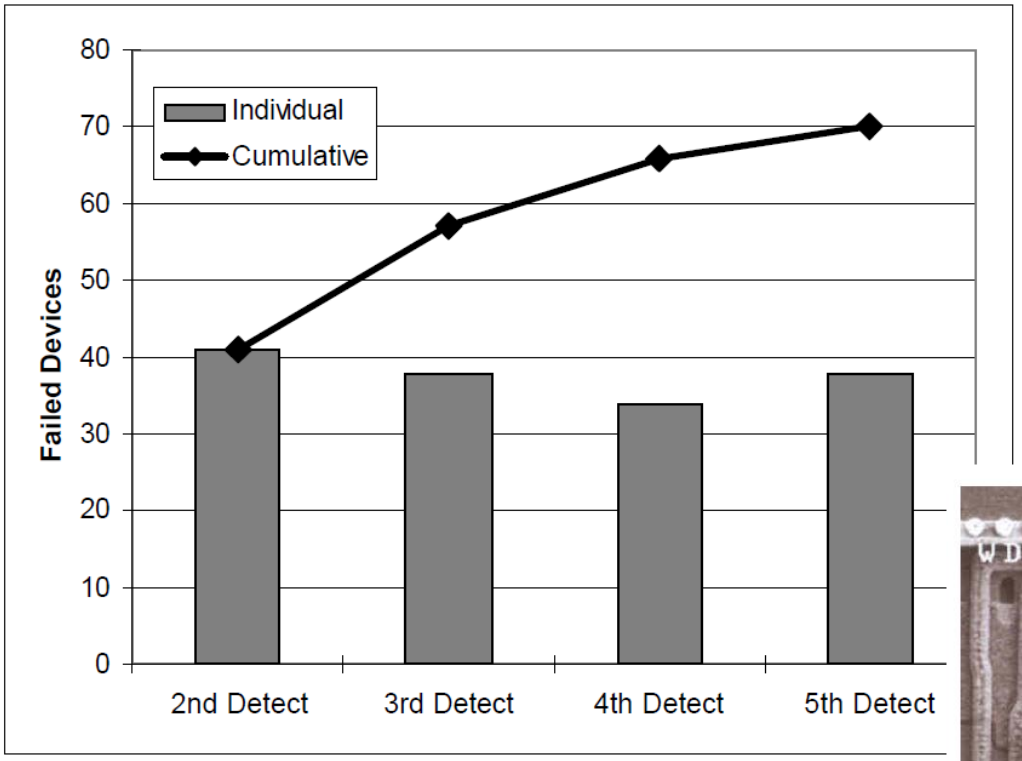
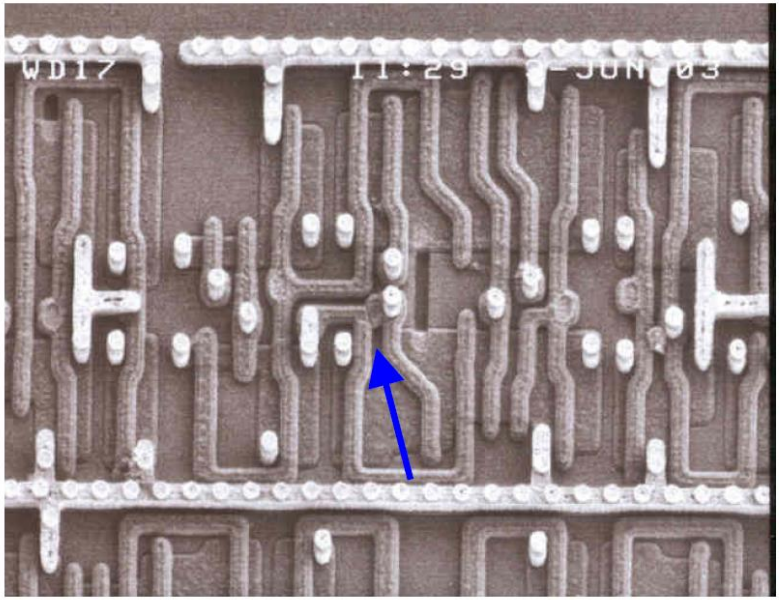


Figure 5 Individual and cumulative fallout observe
the multiple detect patterns on ASIC1

PFA photo of chip failed N-detect test

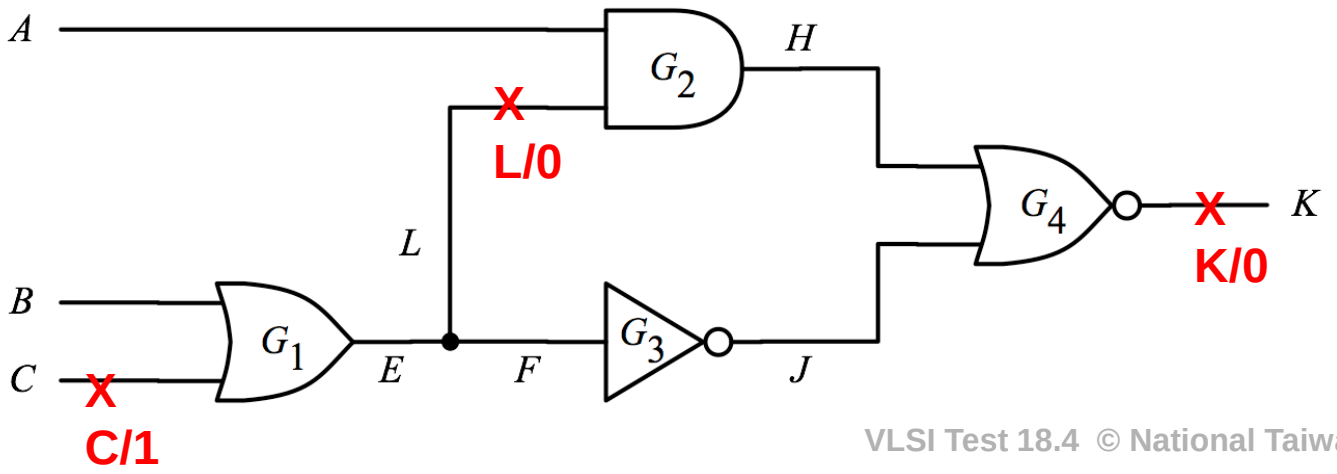


Quiz

Q: Consider 3 SSF, what is F_1, F_2, F_3 of this test set?
What is 2-detect SSF fault coverage?

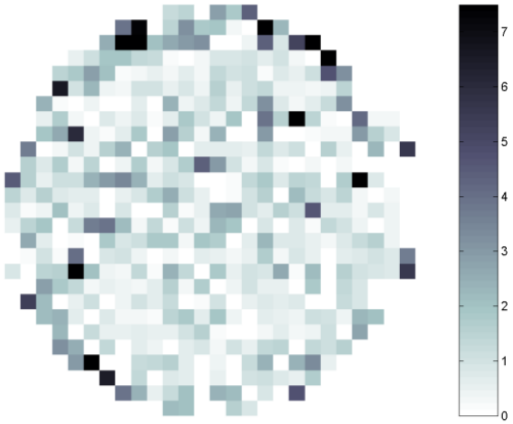
A: F_1={ }, F_2={ }, F_3={ }
$FC_{2\text{-det}}$=

	A	B	C	E	H	J	K	Detected faults
P1	1	1	0	1	1	0	0	
P2	0	0	0	0	0	1	0	
P3	1	0	1	1	1	0	0	
P4	0	0	1	1	0	0	1	



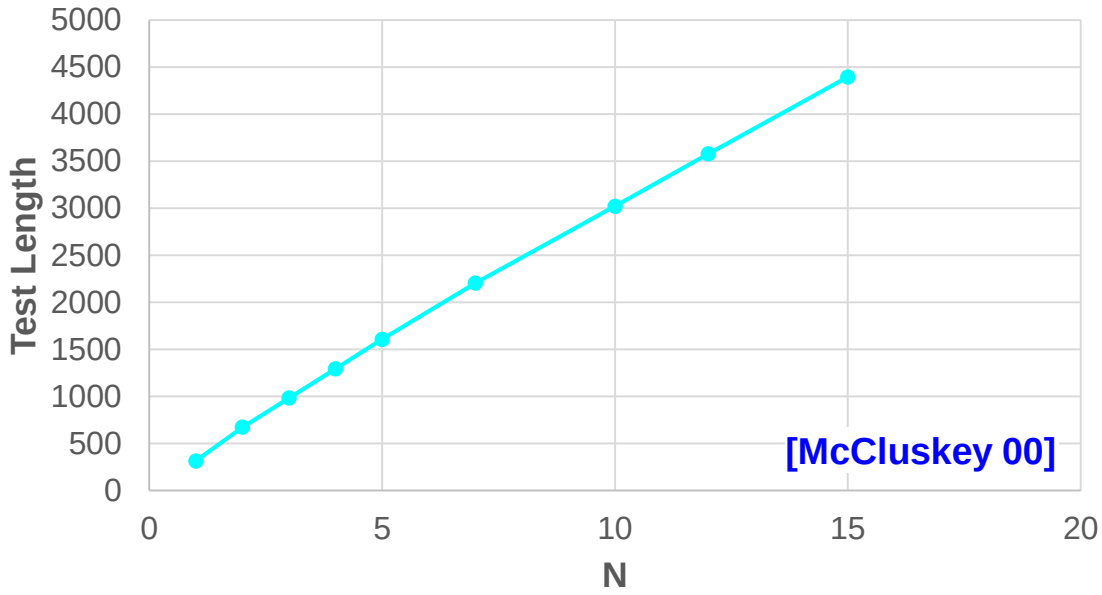
Advanced Topics: ATPG

- Introduction
- Defect-based Testing
- **Advanced ATPG**
  - N-detect (Stanford 1995)
  - **Cell-aware (Mentor 2009)**
    - **Introduction**
    - **Experimental Results**
  - Timing-aware (Mentor 2006)
  - Power-aware (KIT 2005)
- Conclusion



Problems with N-detect

- Test length grows (almost linearly) with *N*
  - Too long for large *N*
- Do not know reasons why and which defects are detected
  - Some patterns are more useful than others
  - Hard to Compress

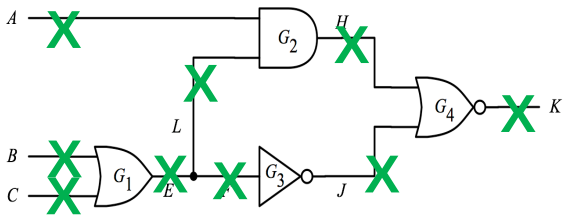


TL Too Long for Large *N*

Gate-level Fault Model Not Enough

Fault Models for Logic Circuits

"High" level
or Functional level
or RT-level



Gate level
(inter-cell)

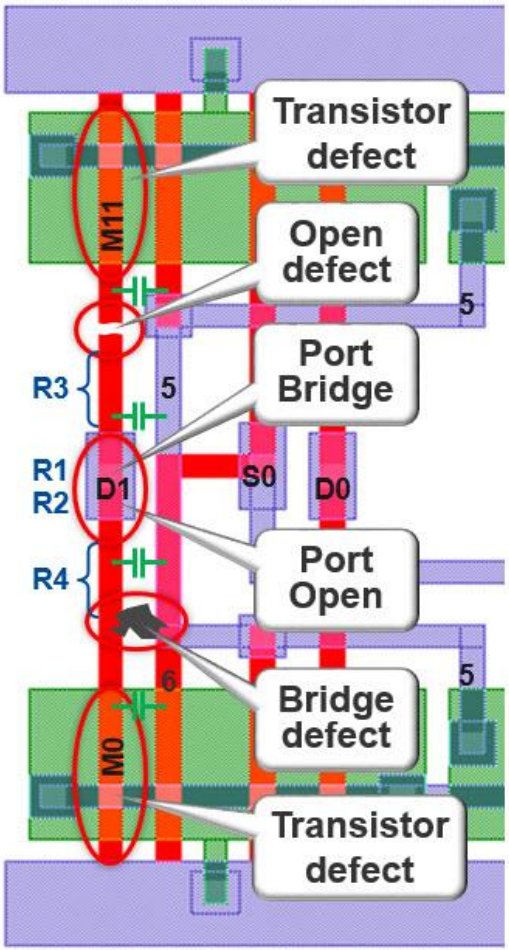
Stuck-at
Bridging
Transition Delay
Path delay

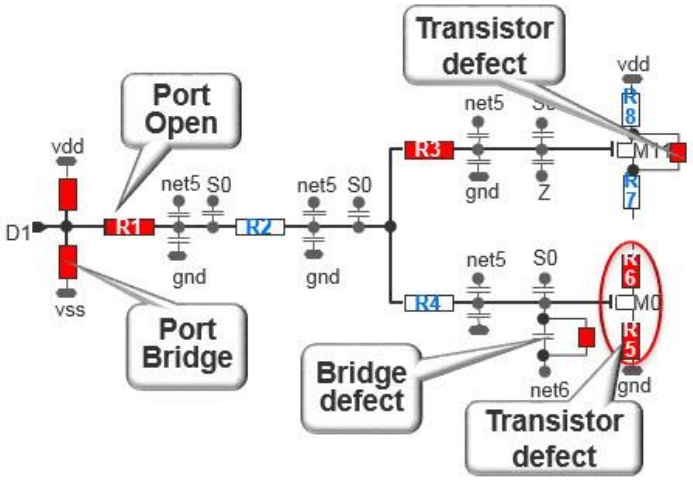
Transistor level
(intra-cell)

Stuck-open
Stuck-on
Cell-aware

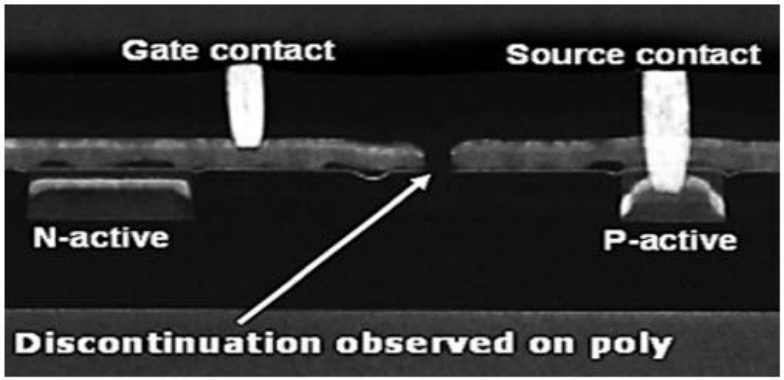
Cell-aware Test (CAT) [Hapke 09]

- Consider different defects types inside cell: open, bridge, transistor
- Need *layout extraction* and *analog fault simulation* (SPICE)





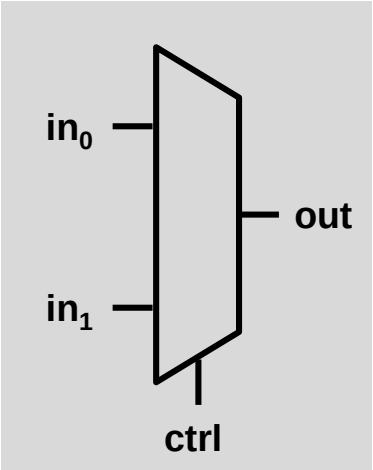
PFA proven CAT only detected defect



[Mentor website]

Why CAT more Effective?

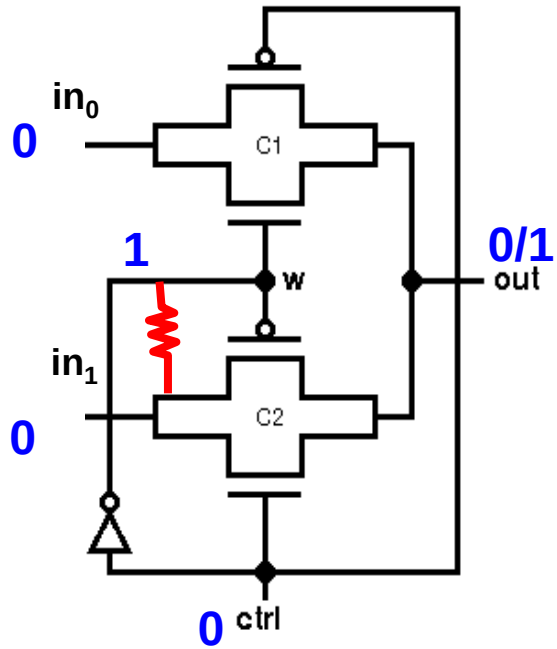
- 4 test patterns detect 8 SSF at MUX I/O pins
  - 100% SSF coverage



in_0	in_1	ctrl	out	detected SSF
0	1	0	0	ctrl SA1, out SA1, in_0 SA1
1	0	0	1	in_1 SA1
1	0	1	0	ctrl SA0, out SA0, in_0 SA0
1	1	1	1	in_1 SA0

- CAT adds {000} to detect in_1/w bridging

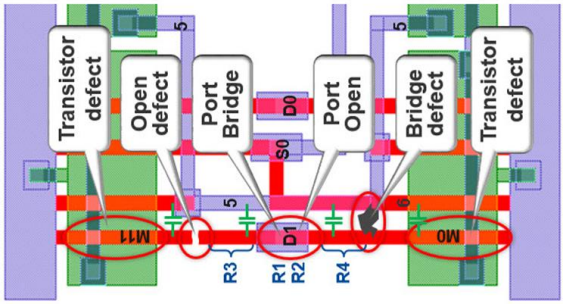
in_0	in_1	ctrl	out
0	1	0	0
1	0	0	1
1	0	1	0
1	1	1	1
0	0	0	0/1



CAT is Effective but Longer

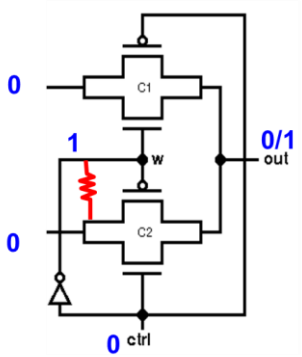
CAT Test Generation

Library cell layout



Layout Extraction

Fault list



in_0	in_1	ctrl	out
0	1	0	0
1	0	0	1
1	0	1	0
1	1	1	1
0	0	0	0/1

SPICE Simulation

Gate input patterns

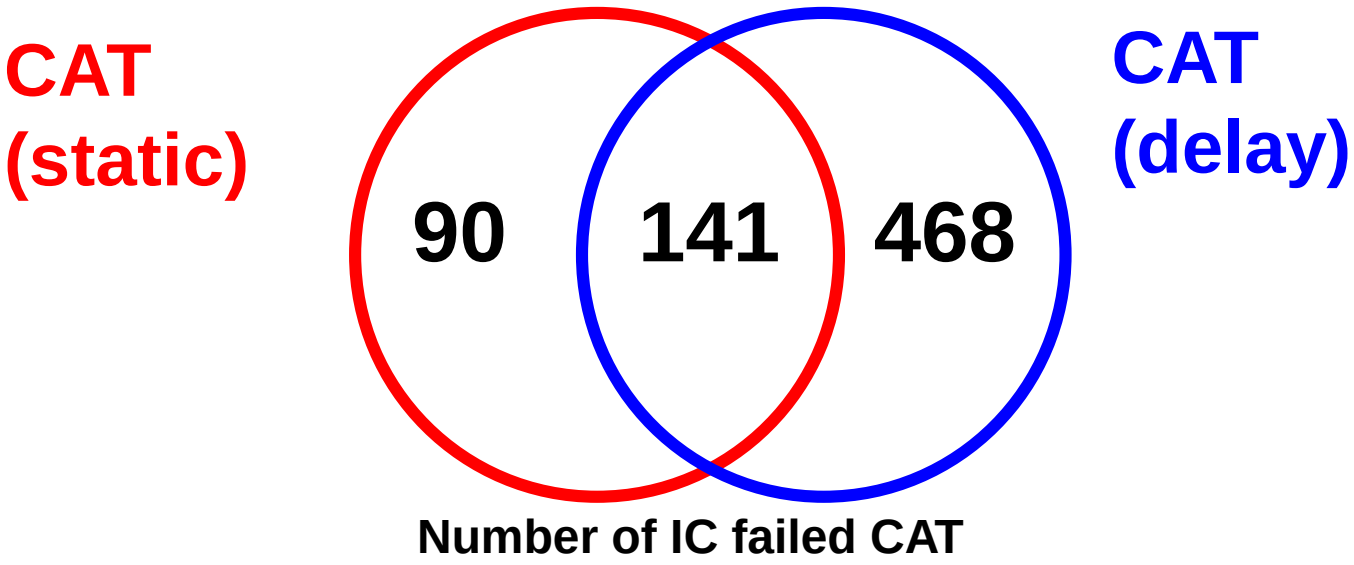
ATPG

CAT Test Patterns

$$\textbf{Fault Coverage}_{CAT} = \frac{number\ of\ faults\ detected\ cell-aware\ faults}{number\ of\ total\ Cell-aware\ faults} \times 100\%$$

AMD Experiments [Hapke 14]

- AMD 32 nm, total 800K chips tested
- 699 chips failed only CAT, passed other tests

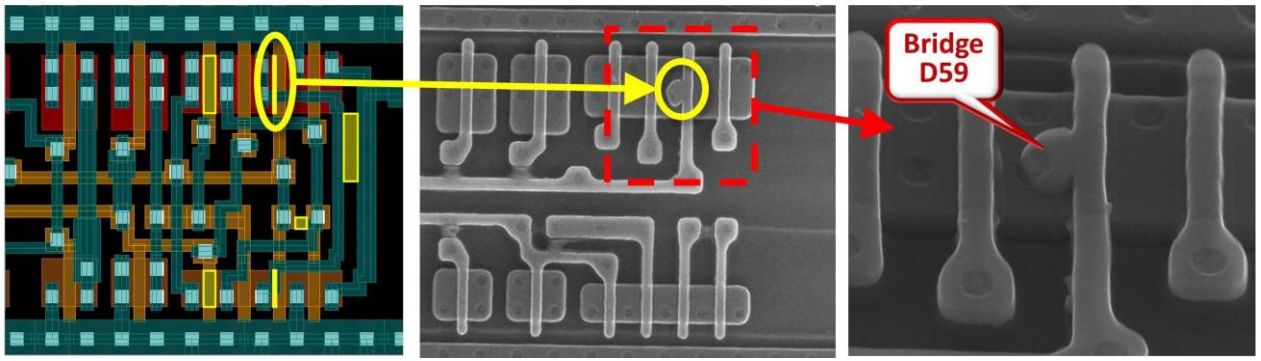


CAT Improves 885 DPM

Pros and Cons of CAT

- ☺ Advantage
  - Reduce DPM effectively
  - Helps to diagnose yield loss due to library cells
- ☹ Disadvantage
  - Test length is longer than traditional test
  - Needs Layout extraction and SPICE simulation for library cells

```
suspect score type   cell=adder
-------------------------------------
1.1     100   CELL   /top/core1/alu
...     ...   ...    ...
1.6     100   BRIDGE D59/MP8:G-MN8:D/1.0
```

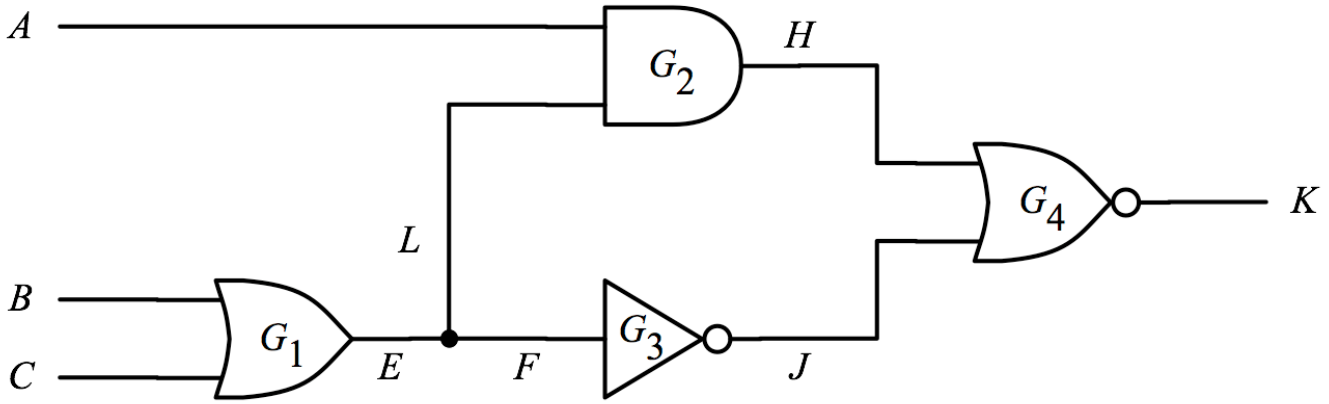


P. Maxwell, et.al, "Cell-Aware Diagnosis: Defective Inmates Exposed in their Cells", European Test Symposium (ETS) 2016

Quiz

Q: Suppose CAT requires 4 gate inputs {00, 01, 10, 11} for OR gate. Please add test pattern to improve the following test set.
A:

	A	B	C	E	H	J	K
P1	1	1	0	1	1	0	0
P2	0	0	0	0	0	1	0
P3	1	0	1	1	1	0	0
P4							



Summary

- *N*-detect test (Stanford 1995)
  - Detects fault at least *N* times to increase diversity of test patterns
  - ☺ Early effective test method to reduce DPM
  - ☹ Very long test length for large *N*
  - ☹ Do not know real reasons for test effectiveness
- Cell-aware test (Mentor 2009)
  - ☺ Clearly know reasons for test effectiveness
    - Useful for diagnosis of yield problems
  - ☹ Also test length is longer
  - ☹ Cell library preparation is difficult
    - Layout extraction and analog fault simulation

Needs More Test Cost to Reduce DPM

References

- [Benware 03] B. Benware , C. Schuermyer , S. Ranganathan , R. Madge , P. Krishnamurthy," Impact of multipledetect test patterns on product quality, "IEEE Int'l Test Conference, 2003.
- [Hapke 09] F. Hapke, et al. "Defect-oriented cell-aware ATPG and fault simulation for industrial cell libraries and designs." IEEE Int'l Test Conference, 2009.
- [Ma 95] Ma, S.C., P. Franco, and E.J. McCluskey, " An Experimental Chip To evaluate Test Techniques Experimental Results," Proc. ITC, pp.663-672, 1995.
- [McCluskey 00] E. J. McCluskey and C. W. Tseng, "Stuck-fault tests vs. actual defects," Int'l Test Conf., 2000.
- [McCluskey 04], EJ McCluskey, JCMLi, et al "ELF-Murphy Data on Defects and Test Sets," VTS 2004.