Testability of VLSI

Lecture 2: Fault Modelling

By Dr. Sanjay Vidhyadharan

Defects, Errors, and Faults

Defect. *A* defect *in an electronic system is the unintended difference between the implemented hardware and its intended design.*

Example : unwanted wire (short to ground)

1. Process Defects – missing contact windows, parasitic transistors, oxide breakdown, etc.
2. Material Defects – bulk defects (cracks, crystal imperfections), surface impurities, etc.
3. Package Defects – contact degradation, seal leaks, etc
4. Age Defects – dielectric breakdown, electromigration, etc.

Fault. *A representation of a "defect" at the abstracted function level is called a* fault

Example: Stuck to Zero Fault

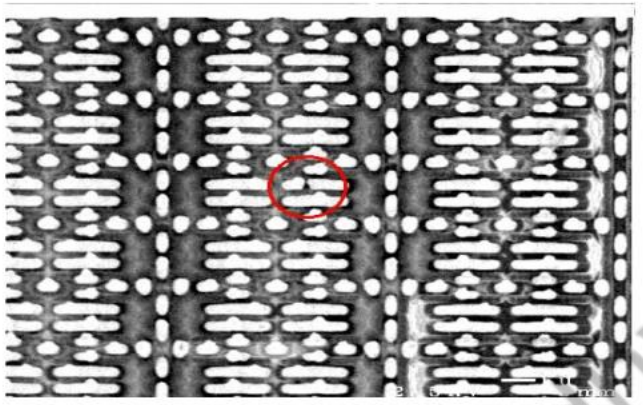
Error. *A wrong output signal produced by a defective system is called an* error. *An error is an "effect" whose cause is some "defect.*

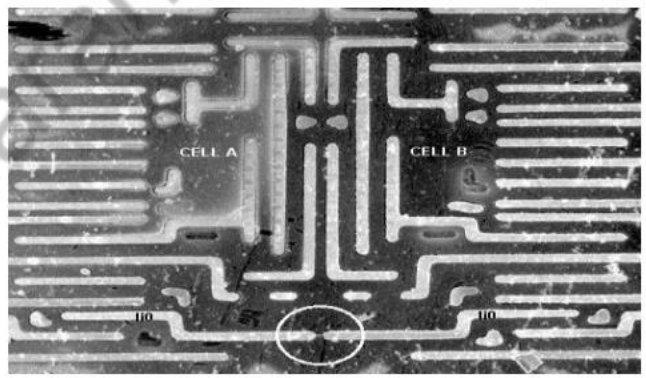
Example: output = 0, when a=b=1 for a AND gate

Failure : *Deviation from expected behavior* **Example: computer crash**

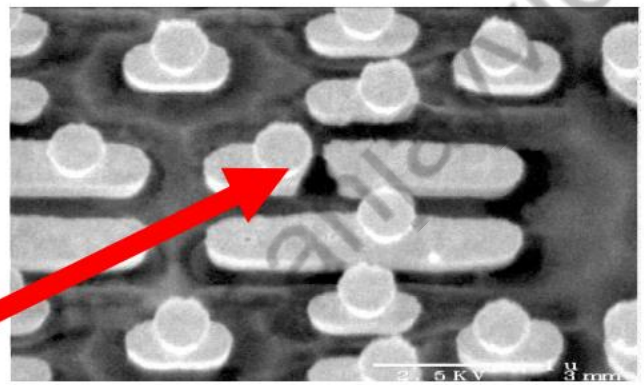
Defect → Fault → Error → Failure

Fabrication Faults

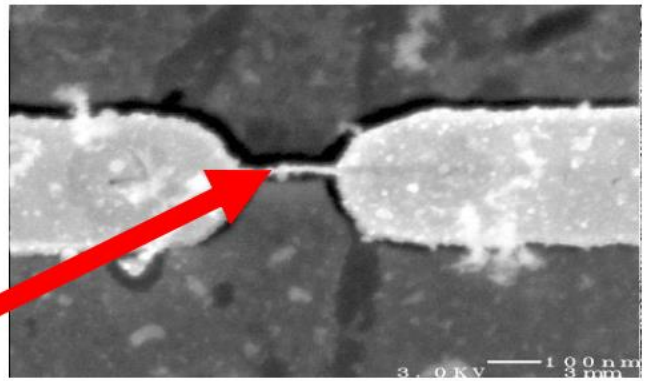




OPEN

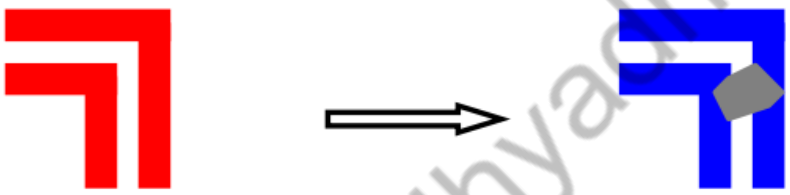


Short

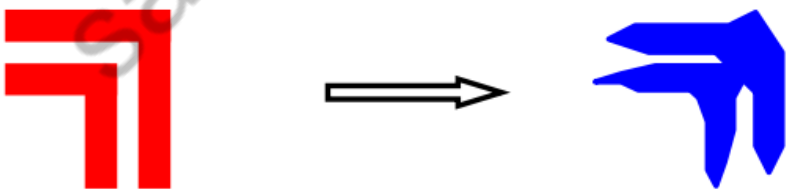


Fabrication Faults

- *Random defects*
  - Caused by random factors such as particles, scratches, …
  - No correlation across wafers, dies



- *Systematic defects*
  - Caused by deterministic factors such as mask, lithography, …
  - Strong correlation across wafers, dies



Fault Models

Why Fault Modelling?

1. **Defects are hard to handle**
   - **How many possible defects in a circuit ? Way too many**
   - **Number of faults can be easily calculated in a circuit**

2. **Fault models makes test automation possible**
   - **Automatic test pattern generation (ATPG) generate test patterns**
   - **Fault simulation**
   - **Evaluate test quality**
   - **Automatic diagnosis**
   - **Locate defects**

Fault Models

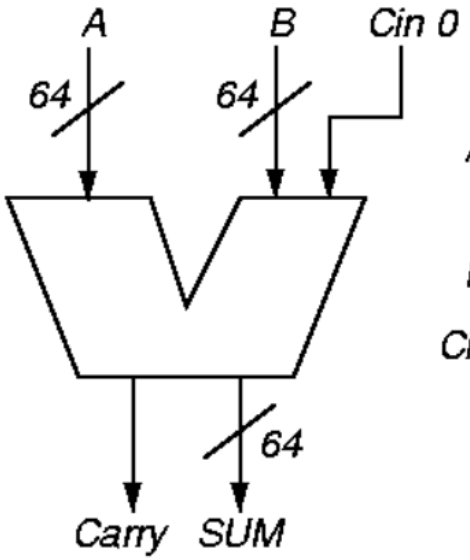
1. Assertion Fault: An assertion expresses a property of a high-level function in the form: "antecedent consequent," where antecedent and consequent can be simple predicates like "line L takes symbolic value v" or conjunctions of simple predicates.

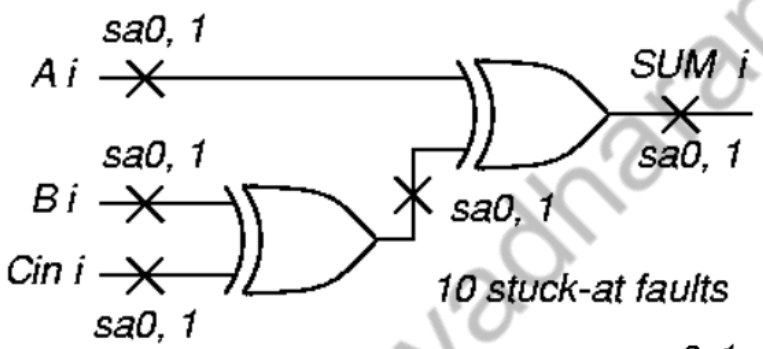
2. Behavioural Faults (*Functional* or *High level*): When the behavior of an electronic system is described in computer-readable form, it is generally written in a programming language (such as C) or some other hardware description language that resembles a programming language.

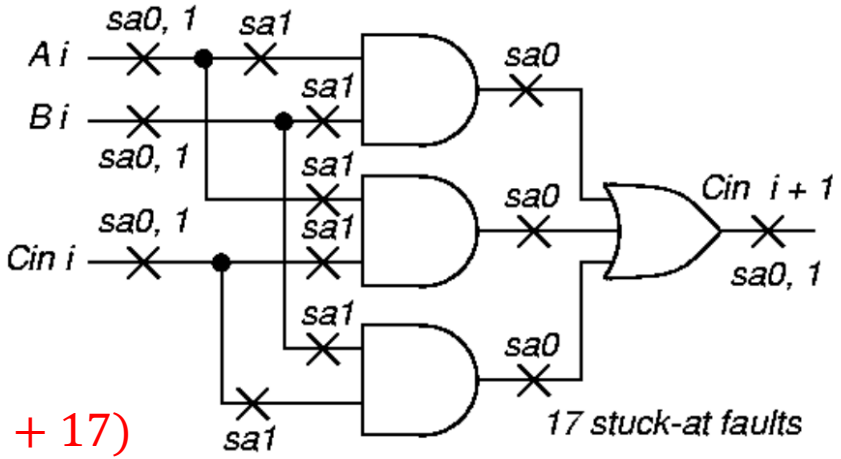
3. Structural Faults: The structure of a circuit may refer to its topology or to physical geometry. Examples of structural faults are single stuck-at faults and bridging faults.
Focus is on manufacturing defects not functional aspect of DUT.

➢ **Please refer to other types of faults in the textbook**

Functional Versus Structural Testing







Functional Test Vectots $2^{129} = 6.8 * 10^{38}$

Required time with Clock of $1\ GHz \approx 22\ years$

Max Structural Test vectors required $64 * (10 + 17)$

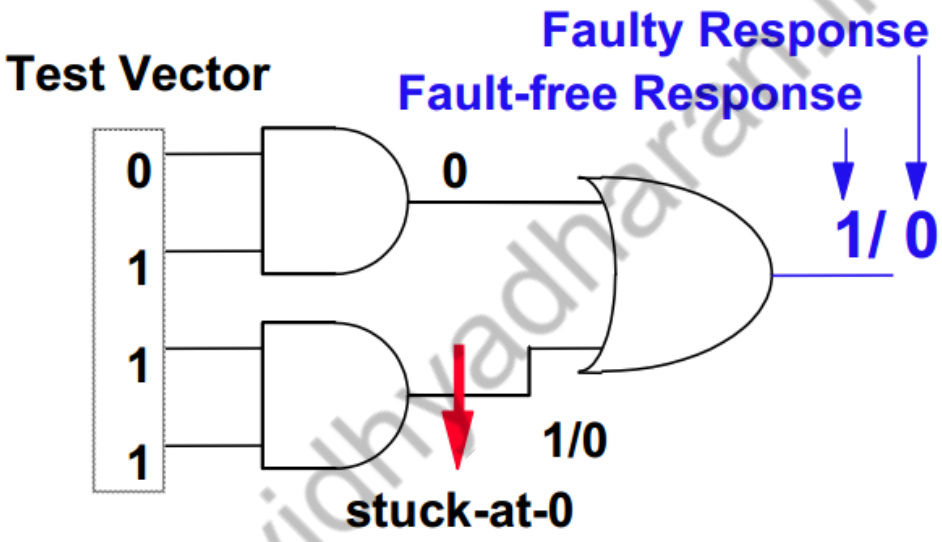
Actaul Structural Test vectors required could be smuch lesseras multiple sa faults gets detected with single vector

Common Structural Fault Models

- Single stuck-at faults
- Transistor open and short faults
- Bridging Faults
- Delay faults (transition, path)
- Analog faults

- **Please refer to other types of faults in the textbook**

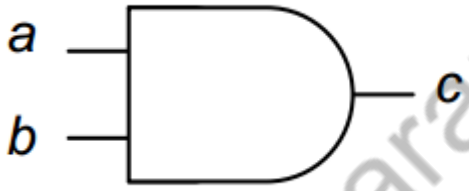
Single Stuck-at faults



Assumptions:

- Only one line is faulty.
- Faulty line permanently set to 0 or 1.
- Fault can be at an input or output of a gate.

Single Stuck-at faults



How many Fault Sites ? 3

How many Fault ? **Total six faults: a/0, a/1, b/0, b/1, c/0, c/1**

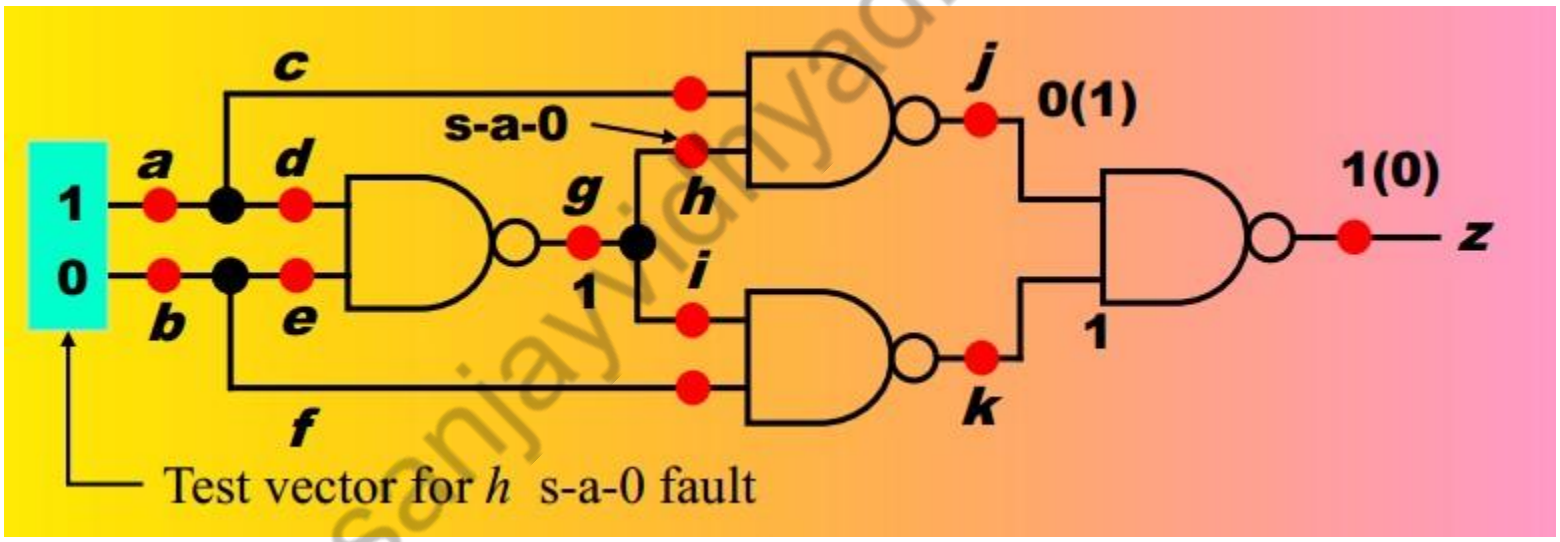
Minimum test length for 100% SSF fault coverage ? **3**

Input	Fault-free	Faulty Output Value with SSF					
a b	Output	*a*/0	*a*/1	*b*/0	*b*/1	*c*/0	*c*/1
0 0	0	0	0	0	0	0	**1**
0 1	0	0	**1**	0	0	0	**1**
1 1	1	**0**	1	**0**	1	**0**	1
1 0	0	0	0	0	**1**	0	**1**

Single Stuck-at faults

Properties of single stuck-at fault

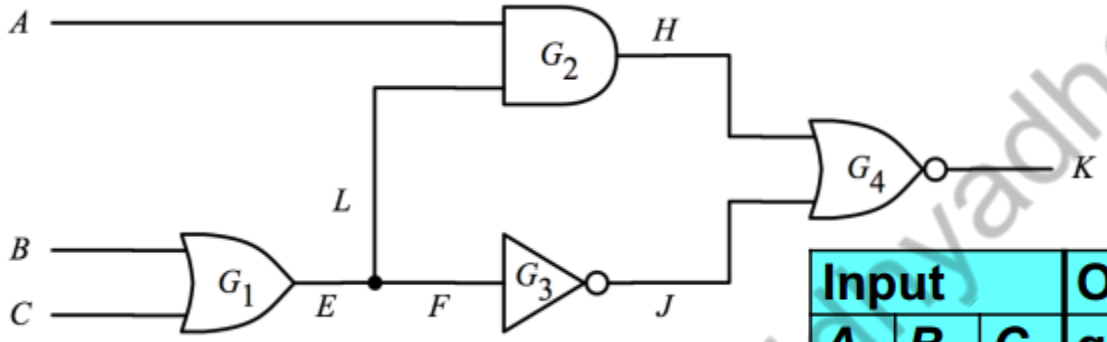
- Only one line is faulty
- The faulty line is permanently set to 0 or 1
- The fault can be at an input or output of a gate



XOR circuit has 12 fault sites (●) and 24 single stuck-at faults

Single Stuck-at faults

SSF on fanout wires *not equivalent* to SSF on fanout branches
Faults on stems and faults on branches are counted separately
Example: *E* is fanout stem; *L,F* are fanout branches



$$K = \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC$$

$$K = \bar{A}(\bar{B}C + B\bar{C} + BC)$$

$$K = \bar{A}(B + C)$$

$$K = \overline{A + \overline{(B + C)}}$$

Input			Output						
A	B	C	good	E/0	F/0	L/0	E/1	F/1	L/1
0	0	0	0	0	0	0	1	1	0
0	0	1	1	0	0	1	1	1	1
0	1	0	1	0	0	1	1	1	1
0	1	1	1	0	0	1	1	1	1
1	0	0	0	0	0	0	0	1	0
1	0	1	0	0	0	1	0	0	0
1	1	0	0	0	0	1	0	0	0
1	1	1	0	0	0	1	0	0	0

Multiple Stuck-at faults

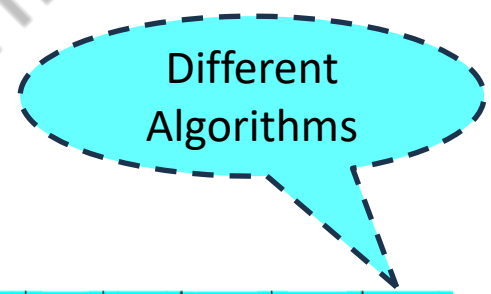
- **Several stuck-at faults occur at the same time**
  - **Important in high density circuits**

If there are N possible fault locations in a circuit
Total possibilities is 3^N *as a line be be* $sa0$, $sa1$ *or good*
One possibility of circuit being good
$2N$ *possibilities of single fault*
Possibilities of multiple fault 3^N *- 1- 2N*

MSF Not Considered in Practice

Multiple Stuck-at faults

- How effective is SSF test sets for multiple stuck-at faults?
  - 14-input ALU 74LS181 circuit
  - 400 single stuck-at faults
  - 79,600 double stuck-at faults
  - 16 different test sets

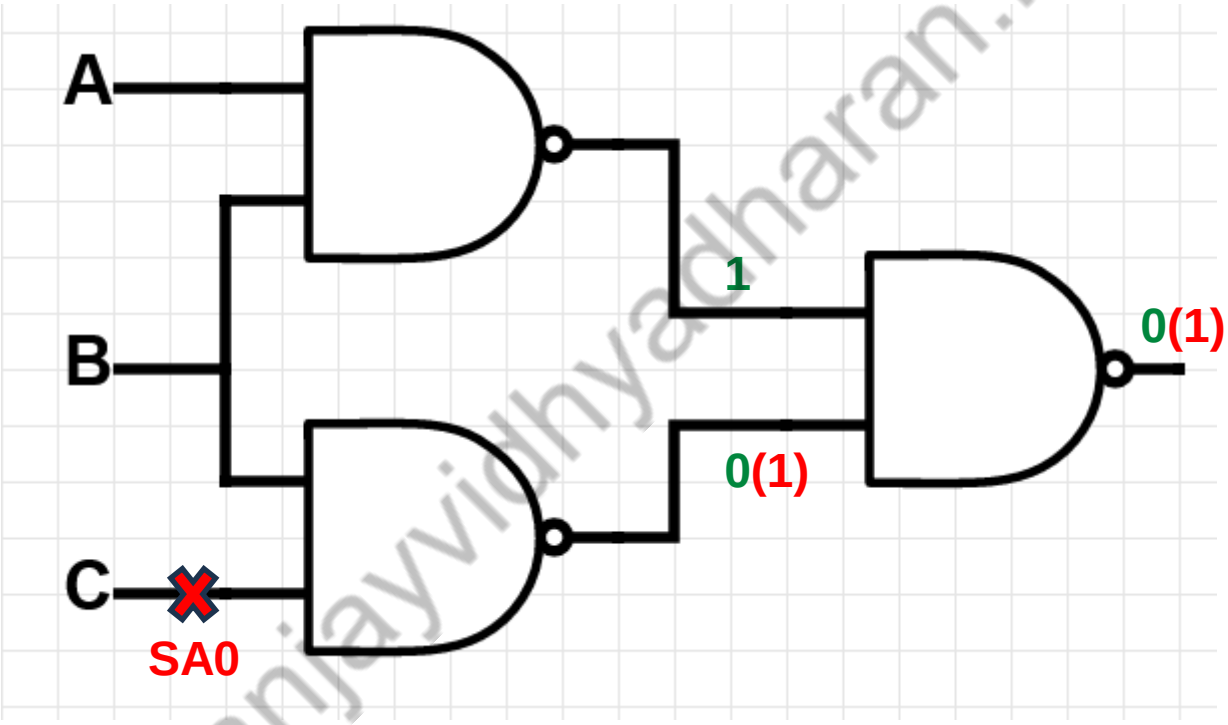


Test Set	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Length	12	12	12	12	12	12	12	14	14	14	14	17	35	124	135	352
Undetected Double-stuck faults	9	8	1	9	28	13	19	4	14	11	3	30	0	0	0	0

- Observation #1: Some shorter tests are better than longer tests
  - Smart test generation/selection is important
- Observation #2: Most double SA faults are detected by SSF test sets

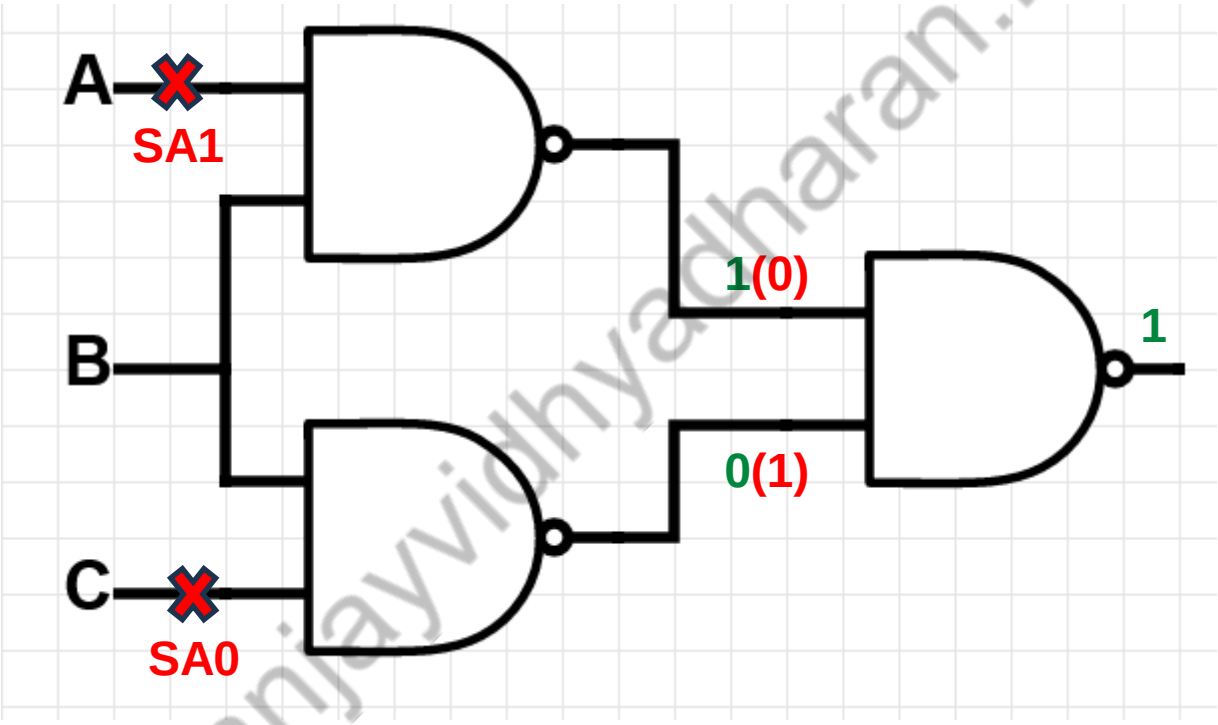
Hughes, J.L.A., and E.J. McCluskey, "An Analysis of the Multiple Fault Detection Capabilities of Single Stuck-at Fault Test Sets," Proc. of Int'l Test Conf, Philadelphia, PA, Oct. 1984, pp. 52–58.

Multiple Stuck-at faults



What should be the test vector ? **011**

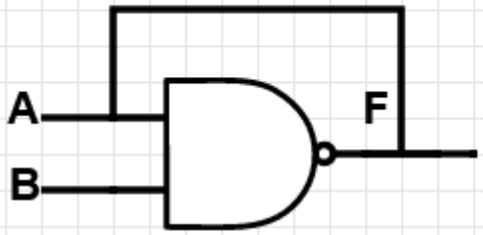
Multiple Stuck-at faults



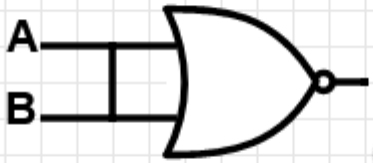
What should be the test vector ? **011** **SA1 is Masking SA0**

What should be the test vector ? **010 detects the MSF {c SA0, a SA1}.**

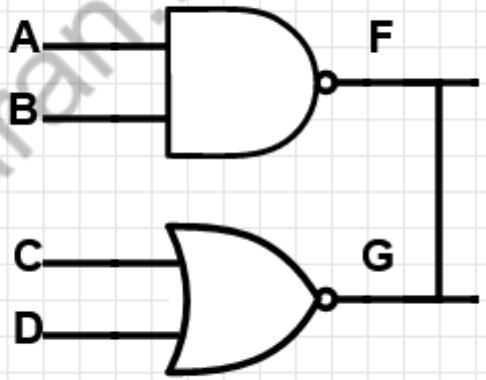
Bridging faults



Input to Output Short



Input to Input Short



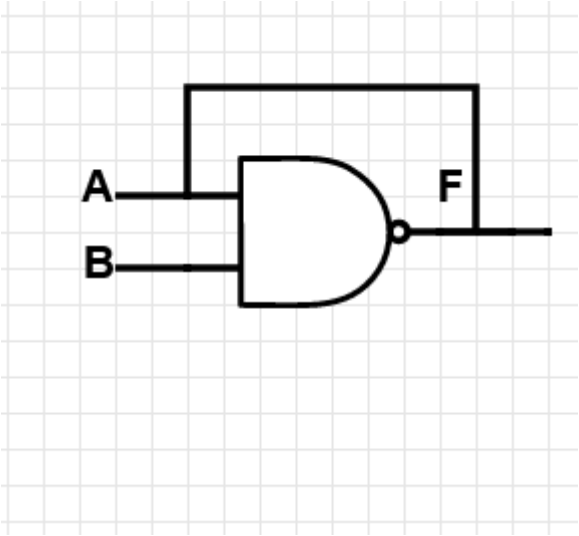
Output to Output Short

- Improper masking or etching
- Loose or excess bare wires
- Defective printed circuit boards
- Shorting of pins of a chip

Test Vectors for Input and Output Stuck-at Faults cover Input-to-Output Shorts

Bridging faults

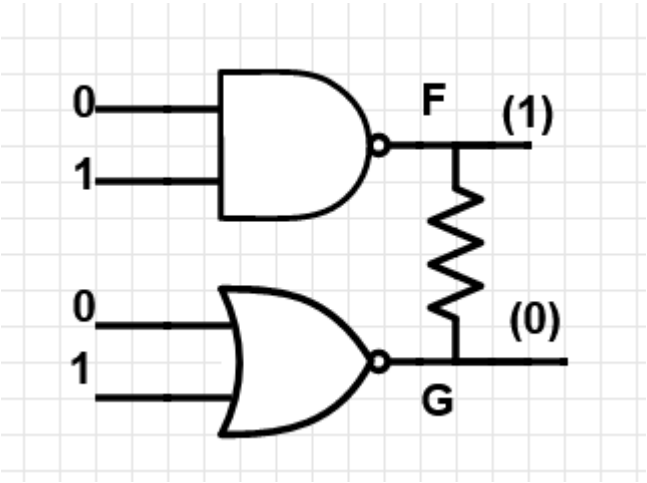
Input to Output Short



1. **Can cause oscillations**
2. **Creates Memory**

Test Vectors for Input and Output Stuck-at Faults cover Input-to-Output Shorts

Bridging faults



Short is Modeled as Low Resistance

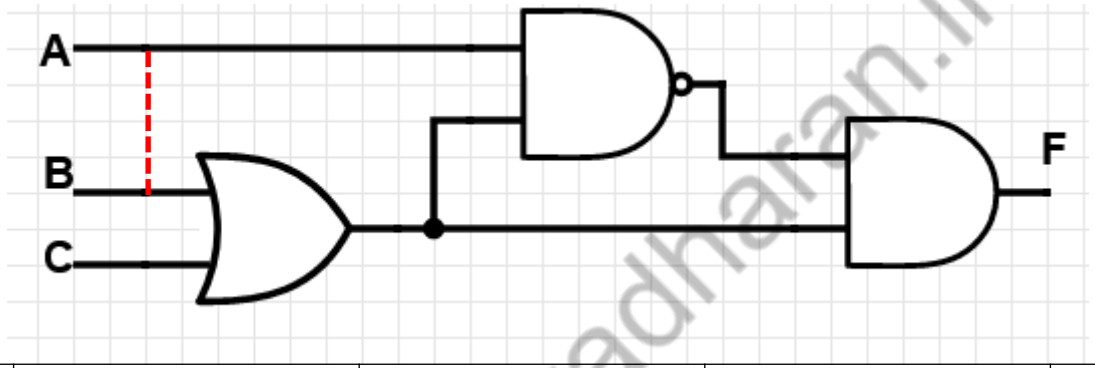
Original Fault-free FG	Wired-OR Faulty FG	Wired-AND Faulty FG	F-dominant Faulty FG
0 0	0 0	0 0	0 0
0 1	**1** 1	0 **0**	0 **0**
1 0	1 **1**	**0** 0	1 **1**
1 1	1 1	1 1	1 1

If F,G = 0,0 can be detected as F s-a-0
If F,G = 1,1 can be detected as G s-a-1

High Resistance Bridges do not affect the logic value, and hence are undetectable by a static logic test.

Bridging faults

Different Models Need Different Patterns



Inputs A B C	Fault-free Output	Wired OR	Wired AND	A dominant
0 0 0	0	0	0	0
0 0 1	1	1	1	1
0 1 0	1	**0**	**0**	**0**
0 1 1	1	**0**	**0**	1
1 0 0	0	0	0	0
1 0 1	0	0	0	0
1 1 0	0	0	0	0
1 1 1	0	0	0	0

Bridging faults

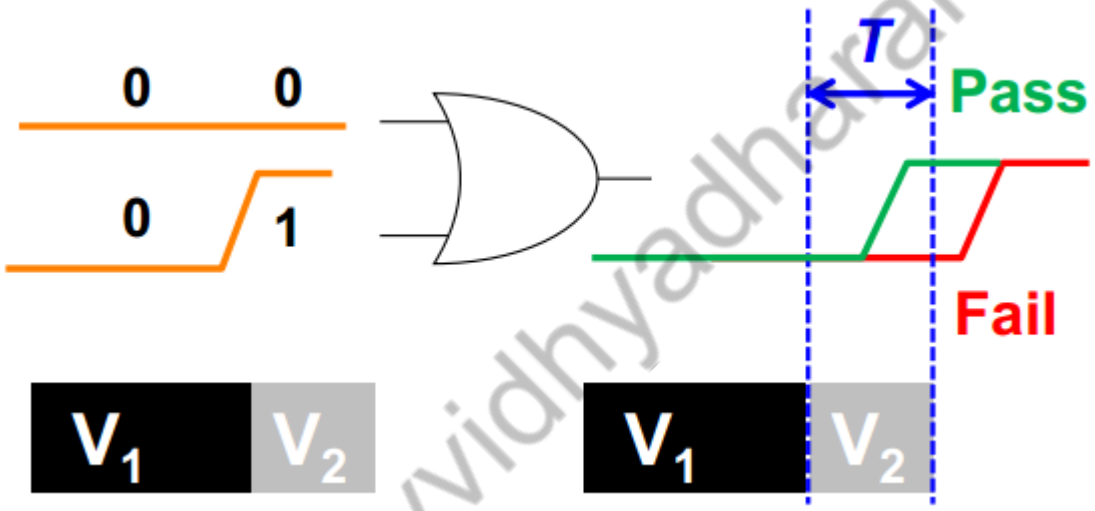
Table 1. Results of 74LS181 simulations.

Test Set	Bry2	Bry6	Goel	Hugh	Krish	McC4	Micz2
Test Length	14	12	35	135	12	124	17
Number of missed AND faults	24	28	4	0	33	0	46
Number of missed OR faults	35	44	5	0	66	1	18
Coverage of AND faults (%)	99.18	99.04	99.86	100.0	98.87	100.0	98.43
Coverage of OR faults (%)	98.79	98.48	99.83	100.0	97.72	99.97	99.38
Total number of each type of fault	2926						

[Millman 88] S.D. Millman, McCluskey, "Detecting bridging faults with stuck-at test sets," ITC 1988.

Delay faults

Delay Fault



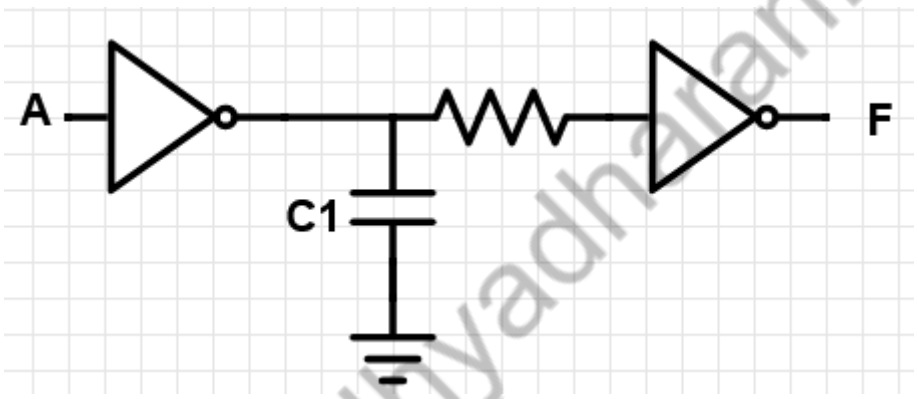
Slow to rise (**STR**), slow to fall Transition (**STF**), faults due to Vt Variation, Doing Variation, Improper contacts etc

No fault detected at static and low frequency operation but glitches can be there at high operating frequencies and cause errors in sequential circuits

Delay Faults requires two test vectors

Delay faults

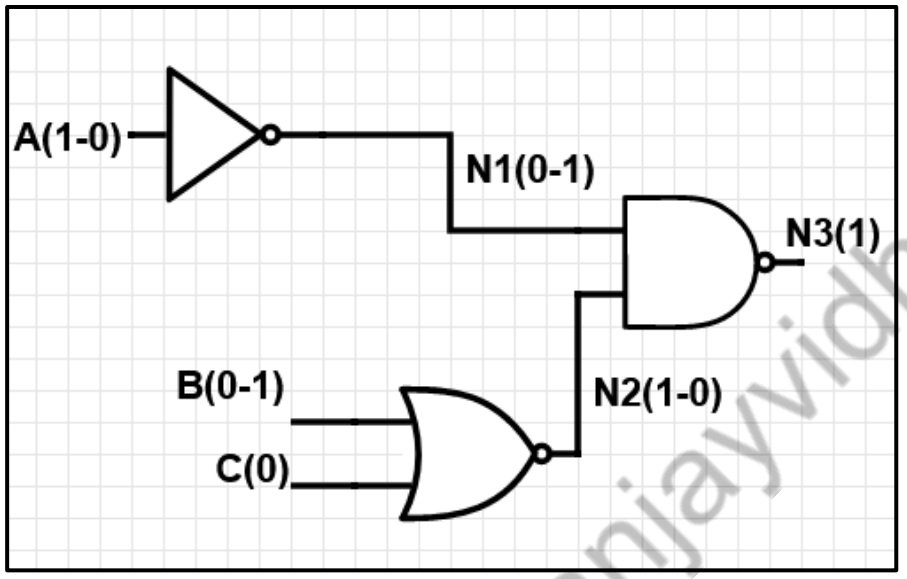
Delay Fault

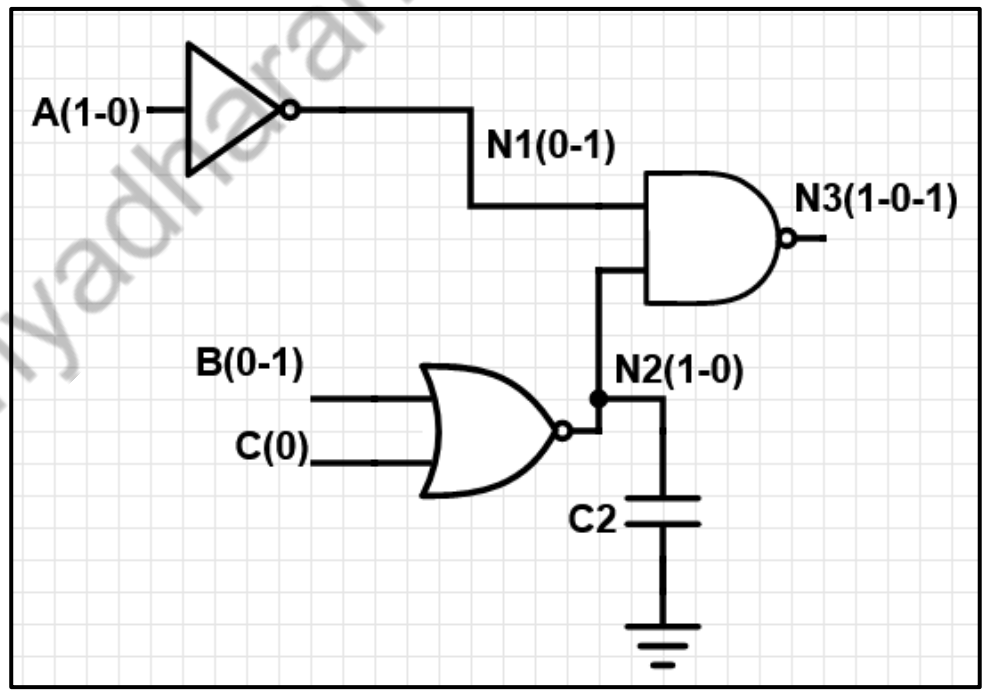


Can be modelled a RC delay but can be because of poor MOSFET being fabricated or nay other fab defects .

Delay Faults requires two test vectors

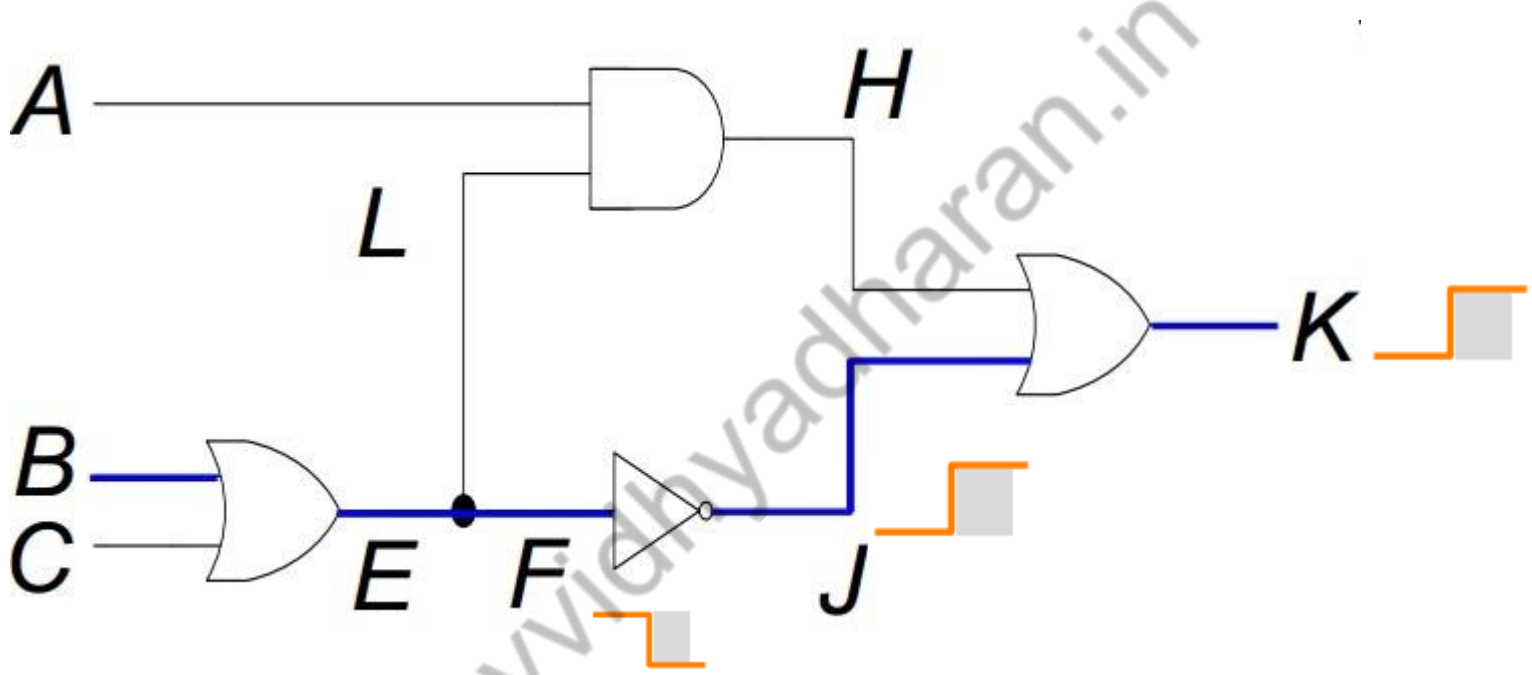
Delay faults





Path Delay Fault

Delay faults

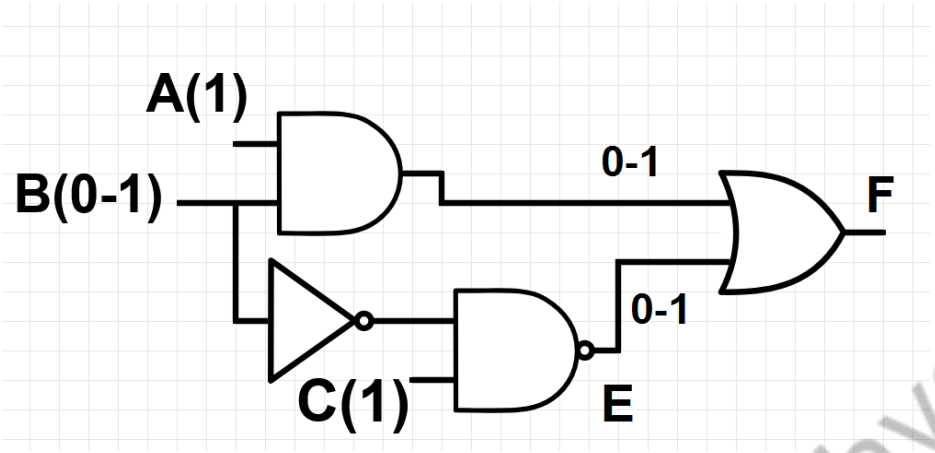


How Many Paths ? 5 paths: {AHK, BELHK, BEFJK, CELHK, CEFJK}

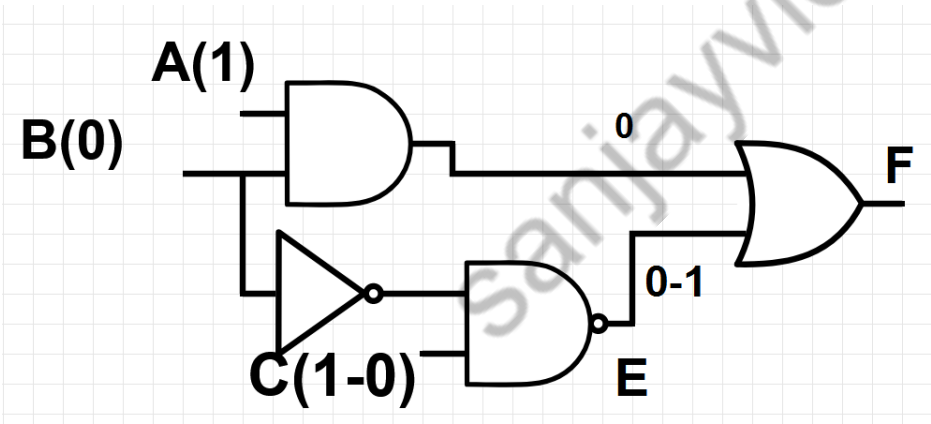
How Many set of test vectors ? 10 sets

Test vector to detect STF fault at F ? 001-000

Delay faults



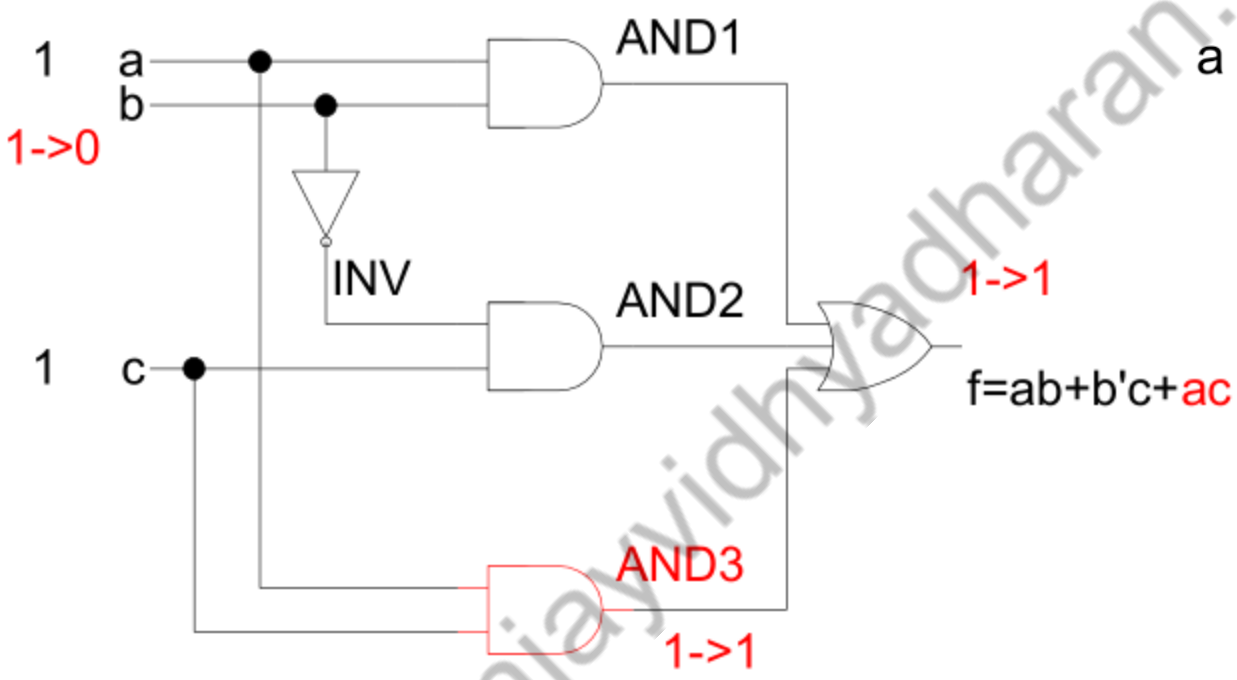
(101,111) : Non-robust test for path STR at E



(101,100) : Robust test for path STR at E

Delay faults

Robust Testing may note possible always



bc / a	00	01	11	10
0	0	1	0	0
1	0	1	1	1

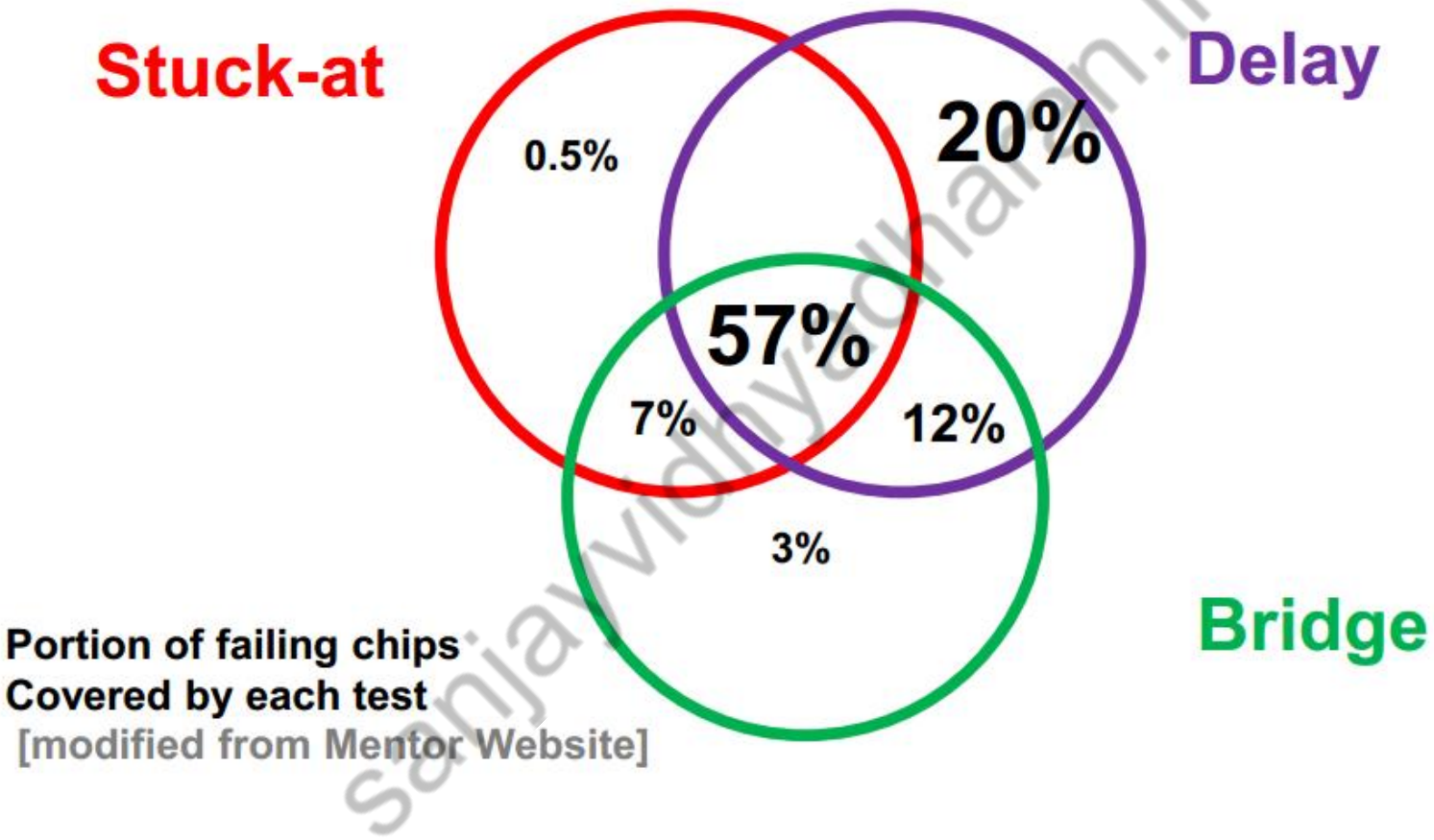
f=ab+b'c + ac

AND1 Check : B=1 , C= 0 **AND2 Check : A=0 , C= 1**

AND3 Check : A=0 , Will render Both AND1 and AND 3 to Low

A=1, B=0 will activate AND2 and B=0 will activate AND 14

Experimental Results

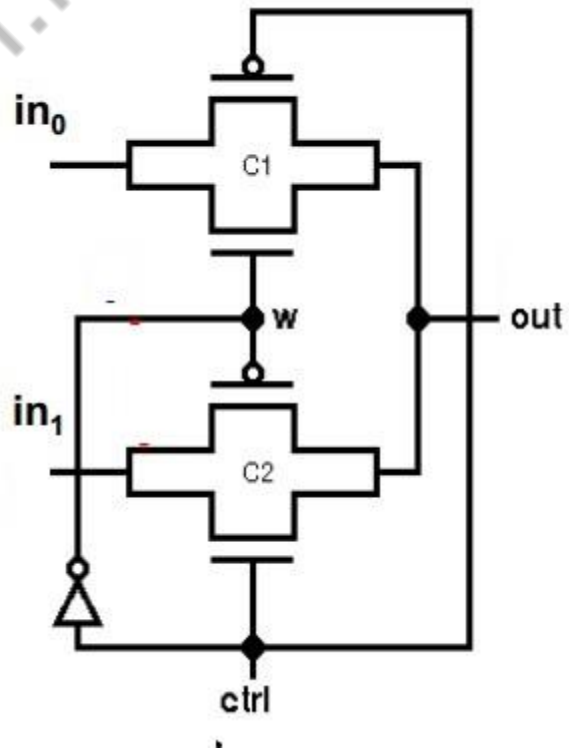


[2]/ Video lectures by Professor James Chien-Mo Li

Fault Models

in_0	in_1	Ctrl	Out	Detected SSF
0	1	0	0	
1	0	0	1	
1	0	1	0	
1	1	1	1	

Note: only four vectors for 3 inputs



Transistor faults

Two types of Transistor Faults

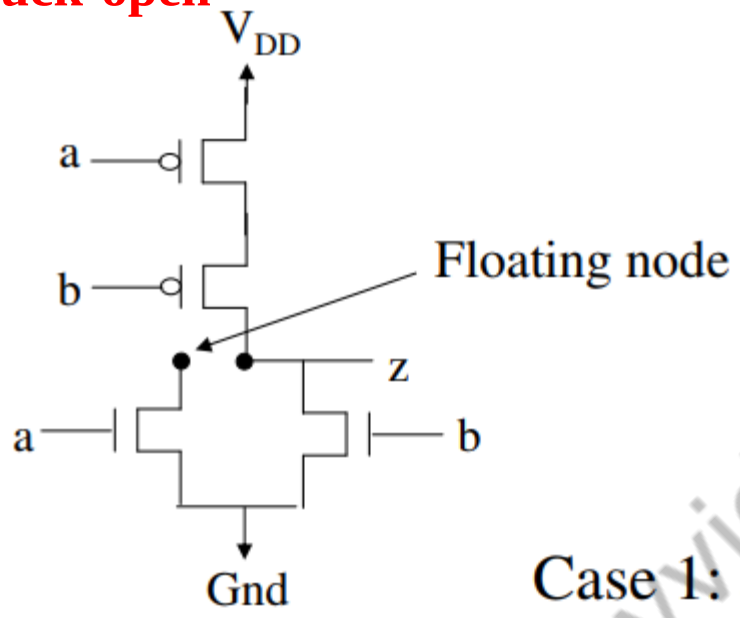
- Stuck-open -- a single transistor is permanently stuck in the open state irrespective of its gate voltage. Single Stuck-open detection requires two test patterns

- Stuck-short -- a single transistor is permanently shorted irrespective of its gate voltage. Detection by quiescent I_{DD}

http://ece-research.unm.edu/jimp/vlsi_test/slides/html/faults2.html

Transistor faults

Stuck-open



Fault-free circuit: $z = \overline{a+b}$

Faulty circuit: $z^f = \overline{a+b} + a\bar{b}\tilde{z}$

$\tilde{z}$: Previous value of z

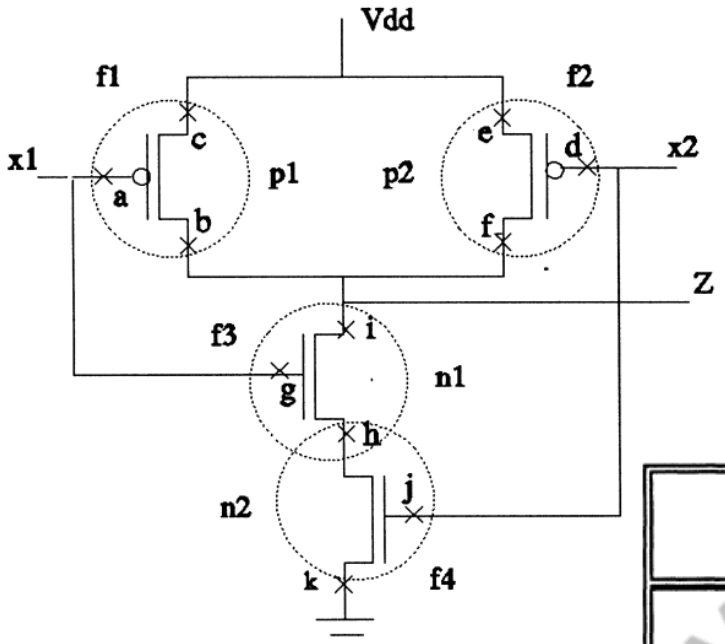
Case 1: a = b = 1, z pulled down to 0

Case 2: a = 1, b =0, z retains previous state

A test for a stuck-open fault requires two patterns

{ab = 00, ab = 10}

Transistor faults



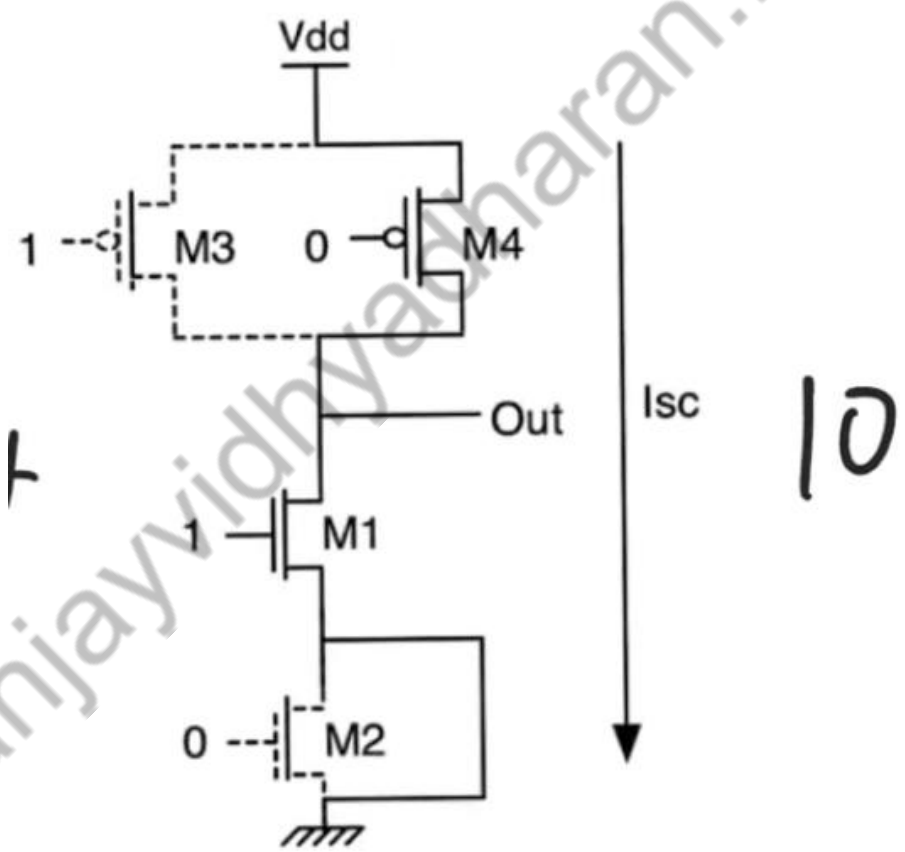
Two-pattern Tests for Stuck-open Faults

Faults	f1 (a,b,c)	f2 (d,e,f)	f3, f4 (g,h,i,j,k)
Initialization Vector (T1)	x1, x2 1 1	x1, x2 1 1	x1, x2 0 0 0 1 1 0
Test Vector (T2)	0 1	1 0	1 1

Automation available to optimize Stuck-at and Stuk-open Fault

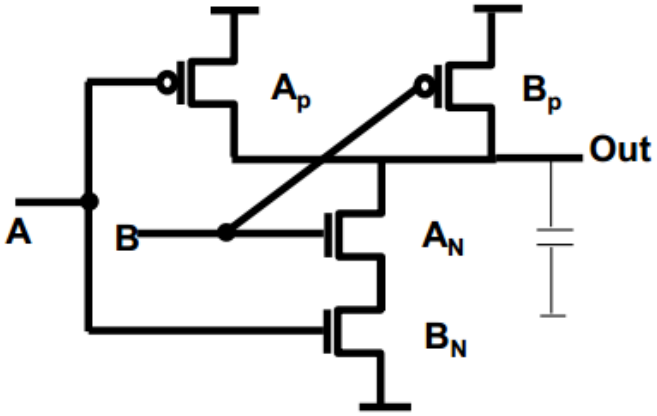
Transistor faults

Stuck-on fault



Transistor faults

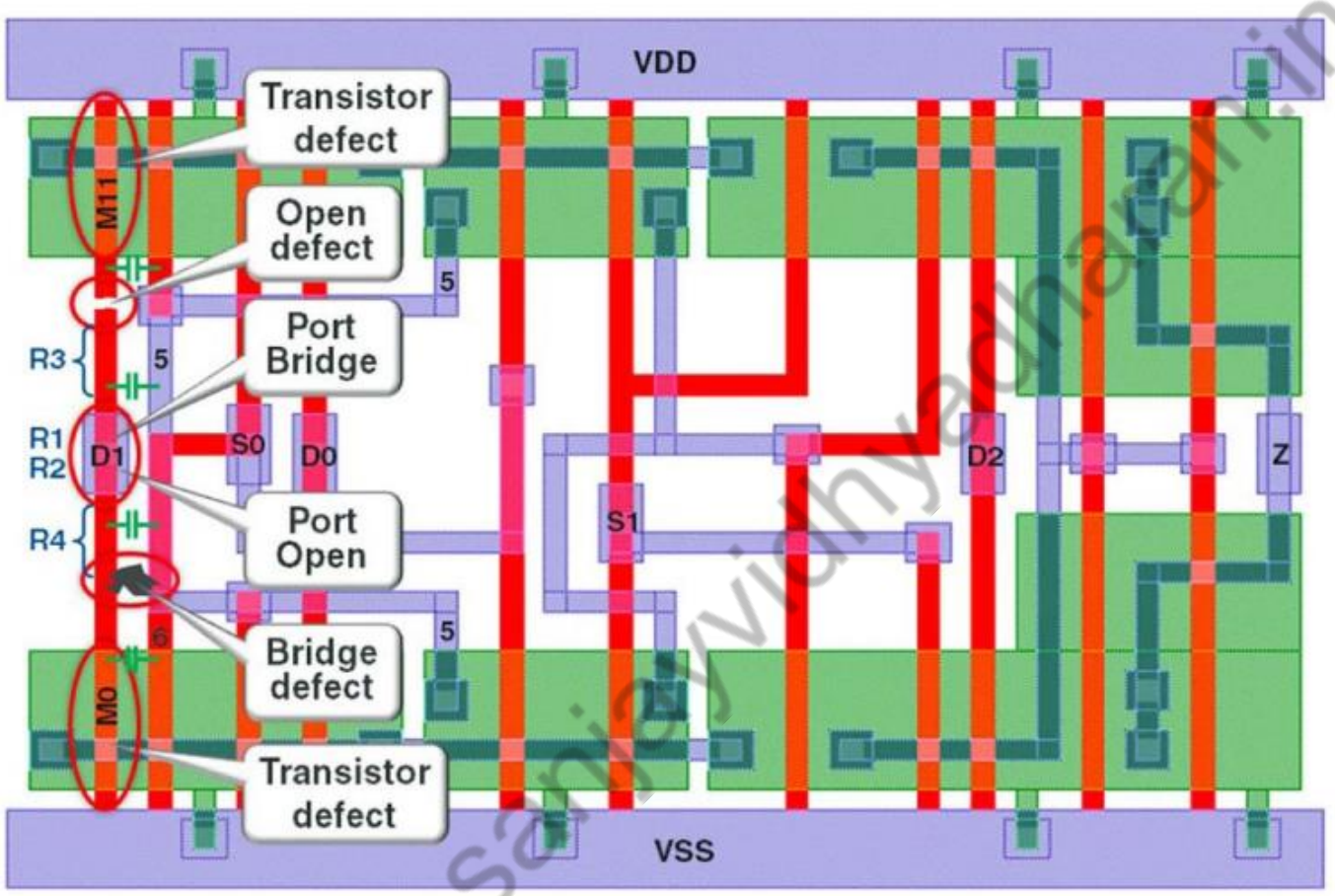
SOP ≠ TDF



$A_1B_1 \rightarrow A_2B_2$	Out_{good}	Detected SOP	Detected TDF
11→01	0→1	A_P	A STF; Out STR
10→01	1→1	-	A STF
11→10	0→1	B_P	B STF; Out STR
01→10	1→1	-	B STF
00→11	1→0	A_N B_N	A STR; B STR Out STF

[2]. Video lectures by Professor James Chien-Mo Li

Cell Aware Fault Model

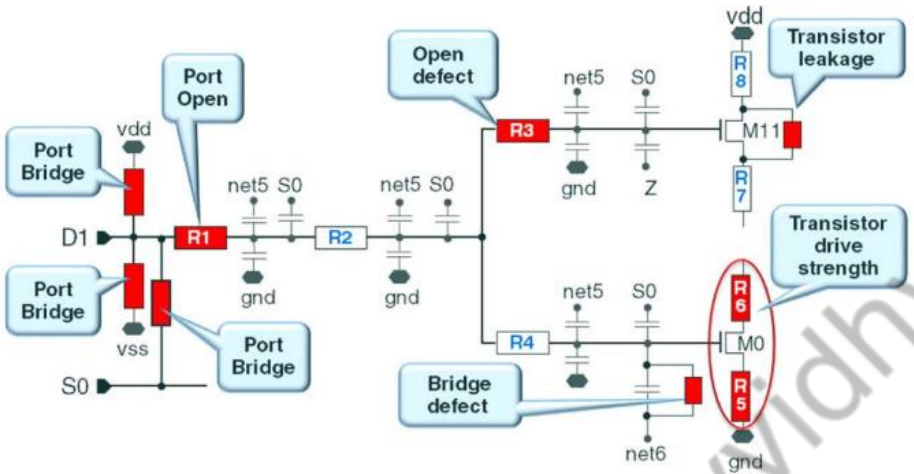


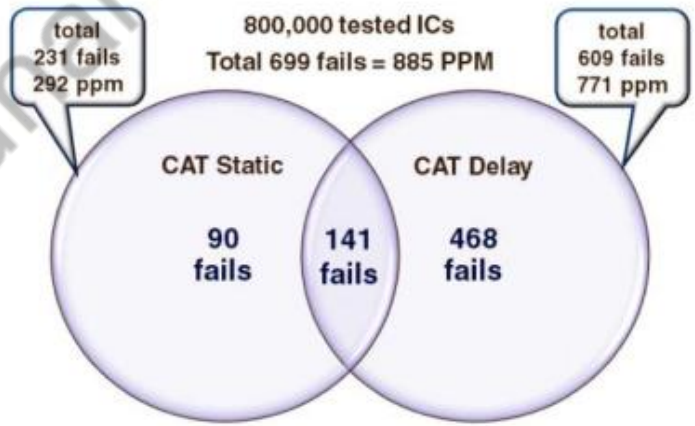
3X1 Mux

Hapke, Friedrich & Redemund, Wilfried & Glowatz, Andreas & Rajski, Janusz & Reese, M. & Hustava, Marek & Keim, Martin & Schlöffel, Jürgen & Fast, Anja. (2014). Cell-Aware Test. Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on. 33. 1396-1409. 10.1109/TCAD.2014.2323216.

Cell Aware Fault Model

3X1 Mux



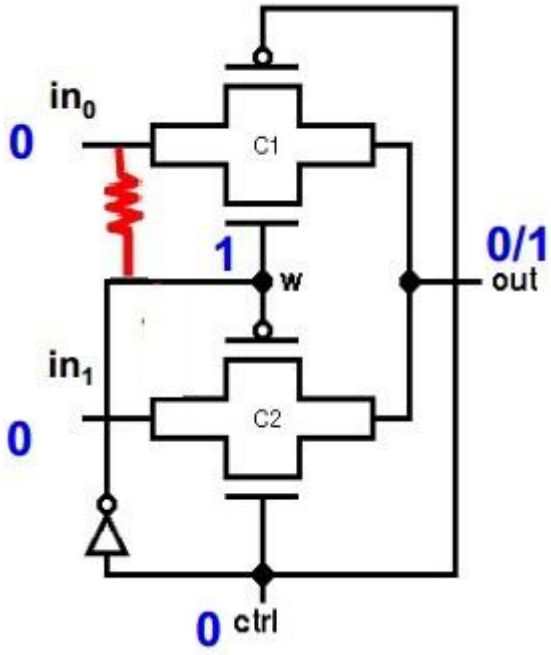


PPM reduction AMD 32 nm notebook processor.

Hapke, Friedrich & Redemund, Wilfried & Glowatz, Andreas & Rajski, Janusz & Reese, M. & Hustava, Marek & Keim, Martin & Schlöffel, Jürgen & Fast, Anja. (2014). Cell-Aware Test. Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on. 33. 1396-1409. 10.1109/TCAD.2014.2323216.

Cell Aware Fault Model

in_0	in_1	Ctrl	Out	Detected SSF
0	1	0	0	In_0 (Sa1), Out (Sa1), Cntrl (Sa1)
1	0	0	1	In_0 (Sa0), Out (Sa0),
1	0	1	0	In_1 (Sa1), Cntrl (Sa0)
1	1	1	1	In_1 (Sa0)
0	0	1	0	Bridging w and in0



Hapke, Friedrich & Redemund, Wilfried & Glowatz, Andreas & Rajski, Janusz & Reese, M. & Hustava, Marek & Keim, Martin & Schlöffel, Jürgen & Fast, Anja. (2014). Cell-Aware Test. Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on. 33. 1396-1409. 10.1109/TCAD.2014.2323216.

References

1. “Essentials of Electronic Testing, for Digital, Memory and Mixed-Signal VLSI Circuits”, Michael L. Bushnell and Vishwani D. Agrawal, – Kluwer Academic Publishers (2000).

2. Video lectures by Professor James Chien-Mo Li
Lab. of Dependable Systems Graduate Institute of Electronics Engineering
National Taiwan University
https://www.youtube.com/watch?v=yfcoKOUV5DM&list=PLvd8d-SyI7hjk_Ci0zpTqImAtpEjdK5JF&index=1

3. http://ece-research.unm.edu/jimp/vlsi_test/slides/html/faults2.html

Thankyou