5. Fault Modeling

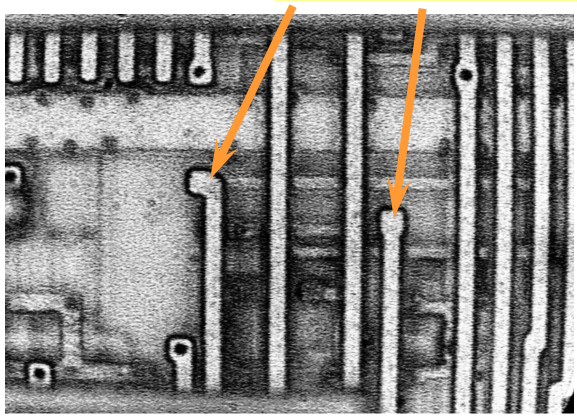
Fernando Gonçalves ©

Outline

- Defects
- Realistic faults
- *Stuck-at* fault model
- Fault detection
  - Path sensitization
  - Fault interaction
  - Undetectable faults
  - Redundancy
- Fault equivalence
- Fault collapsing
- Fault dominance

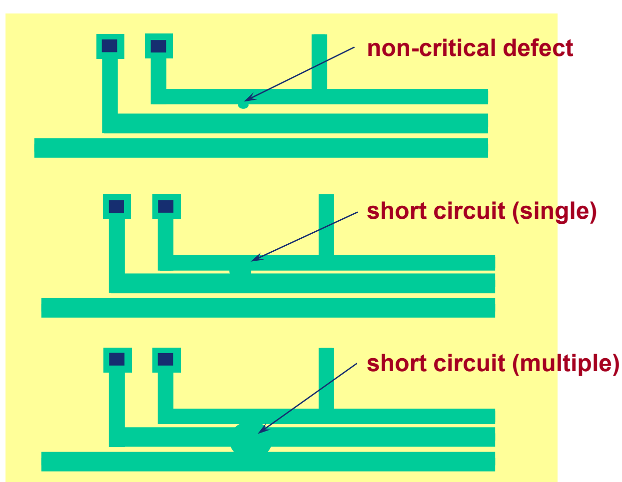
Examples of Manufacturing Defects

Defective contacts ⇒
faults of type "open circuit"



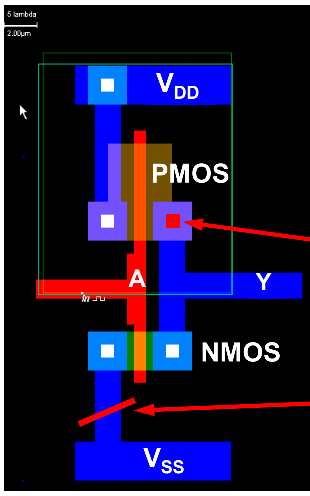
3

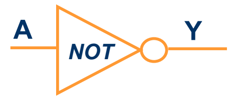
Critical Defects vs. Non-Critical Defects



4

Examples of Realistic Faults: *Open Circuits*



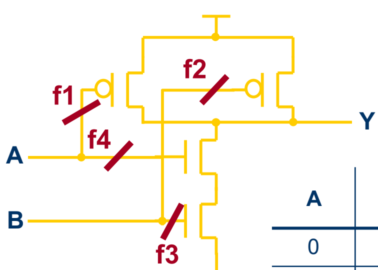


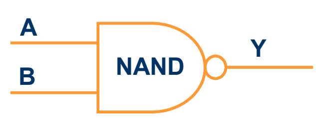
Open contact between metal and diffusion
≈ **Y s-a-0**

Broken metal line
≈ **Y s-a-1**

5

Examples of Realistic Faults: *Open Circuits*



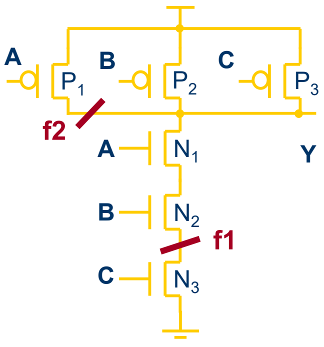


		Y			
A	B	without faults	with **f1**	with **f2**	with **f3** or **f4**
0	0	1	1	1	1
0	1	1	previous state	1	1
1	0	1	1	previous state	1
1	1	0	0	0	previous state

In the presence of a fault, a **combinational circuit** can be transformed in a **sequential circuit**

6

Examples of Realistic Faults: *Open Circuits*



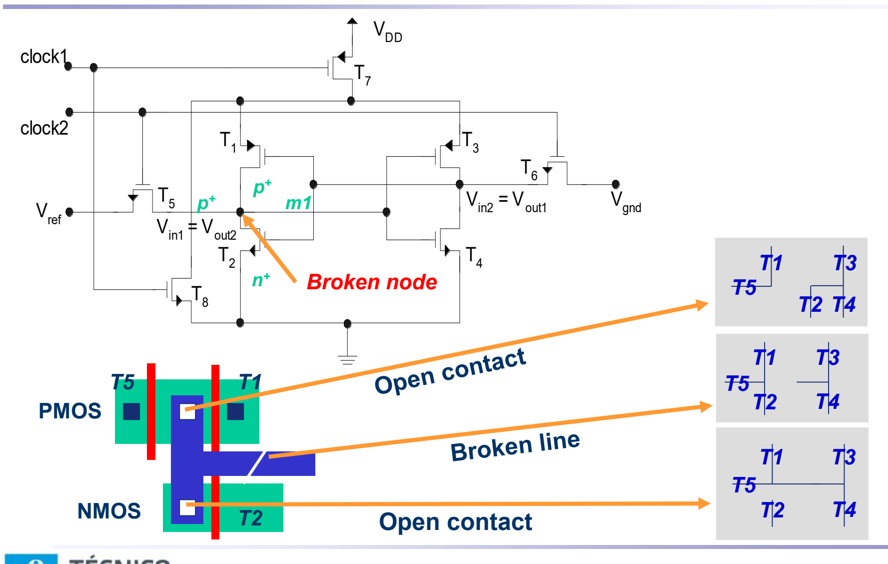
Q1: Determine a test vector that makes the fault **f1** observable in Y.

Q2: How many test vectors detect **f1**?

Q3: How many test vectors are needed to detect **f2**? Determine the sequence of test vectors.

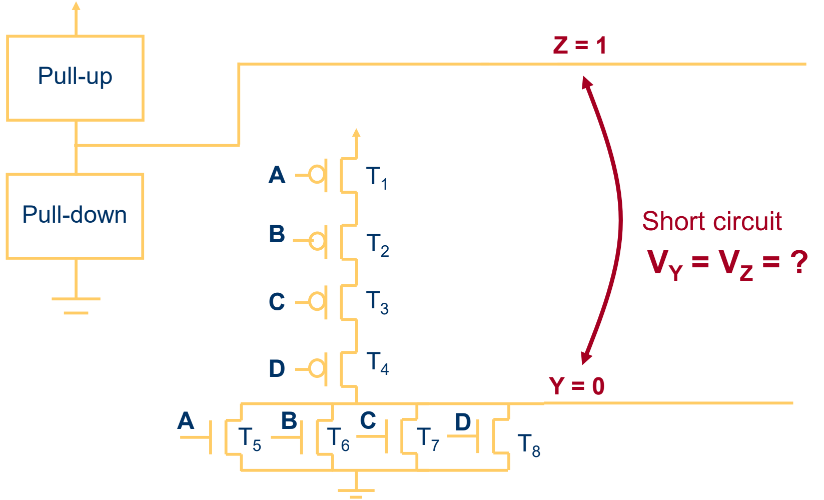
7

Examples of Realistic Faults: *Open Circuits*



8

Examples of Realistic Faults: *Short Circuits*



9

The Single Stuck-At Fault Model (SSF)

- Is technology independent
- The tests that detect SSFs also detect many physical defects
- The number of SSFs in a circuit is relatively small

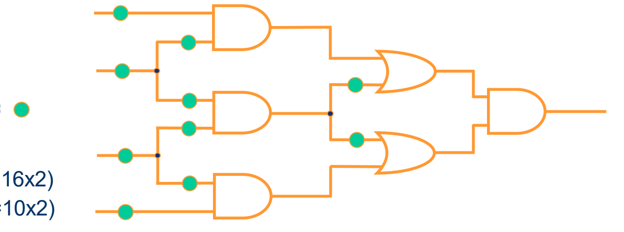
10

Which SSFs to test?

Theorem: In a fanout free circuit, the detection of all SSFs in the primary inputs ⇒ detection of all SSFs

Checkpoints: Primary inputs and *fanout* branches

Theorem: A set of test vectors that detects all SSFs in all checkpoints of a combinational circuit ⇒ detection of all SSFs



\# possible faults = 32 (=16x2)
\# of checkpoints = 20 (=10x2)

11

The Multiple Stuck-At Fault Model (MSF)

How many faults?

assuming **n** = nbr. of possible fault locations

⇒ **2n** SSFs

⇒ **3^n-1** MSFs (any multiplicity)

The multiplicity until **k** is given by: $\sum_{i=1}^{k} \binom{n}{i} 2^i$

Example:

n = 1.000, k = 2 ⇒ approximately 2.000.000 double faults

12

Should we Consider the MSF Model?

- The probability that a physical defect produces a MSF is low
- Experimental results for one circuit shown that a complete set of tests for SSFs also detects more than 99.9% of double, triple and quadruple faults

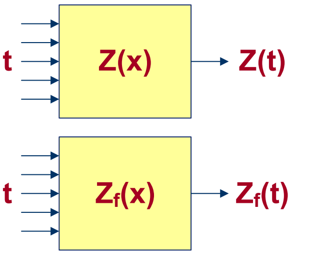
⇒ so, in general MSFs are not used !

Fault Detection

If a test **t detects a fault f**, then $Z(t) \neq Z_f(t)$

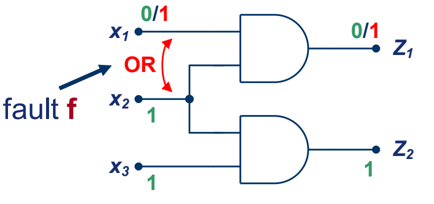
For a circuit with a single output, the set of tests that detect the fault **f** are the solutions of the following equation:

$$Z(x) \oplus Z_f(x) = 1$$



Fault Detection

	Without faults	With fault **f**
Z_1	$x_1 x_2$	$x_1 + x_2$
Z_2	$x_2 x_3$	$(x_1 + x_2)\, x_3$



All tests that detect **f** in $\mathbf{Z_1}$

$$x_1 x_2 \oplus (x_1 + x_2) = 1 \;\Rightarrow\; \boldsymbol{x_1 \oplus x_2 = 1}$$

All tests that detect **f** in $\mathbf{Z_2}$

$$x_2 x_3 \oplus (x_1 + x_2)\, x_3 = 1 \;\Rightarrow\; \boldsymbol{x_1 \overline{x_2} x_3 = 1}$$

Line Sensitization

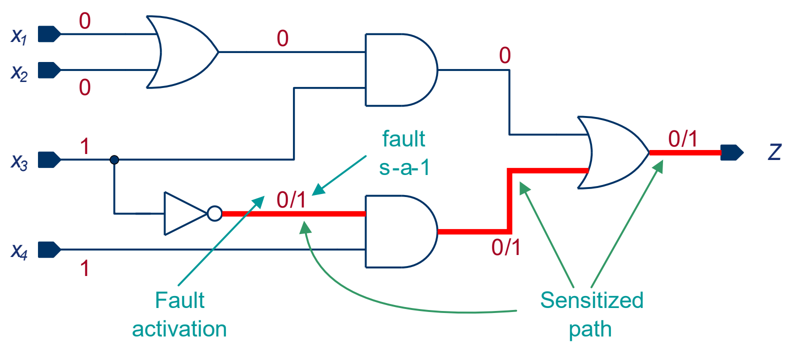
- A line is sensitive to a fault when the logical value is different in the presence of the fault
- A sensitive line highlights the effect of a fault
- The values in the fault-free and faulty circuits are denoted according to the following notation:

<value for fault-free circuit> / <value for faulty circuit>

Example: **0/1** $\Rightarrow$ '0' in the fault-free circuit
'1' in the faulty circuit

Path Sensitization

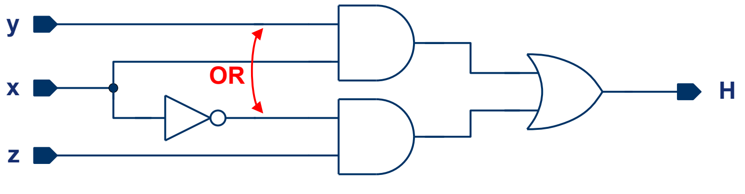
A sensitized path has different logical values in the presence of a fault



Fault Detection

If **f is detectable,** then there is at least a test that detects **f**

Otherwise, **f is undetectable** $\Leftrightarrow$ **H = H_f**



$H = \boldsymbol{x\, y + \bar{x}\, z}$

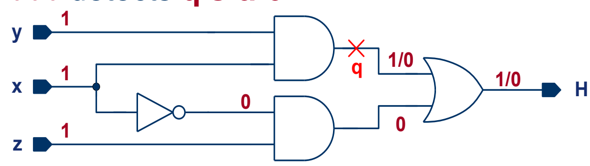
$H_f = x\,(\bar{x} + y) + (\bar{x} + y)\, z = x\, y + \bar{x}\, z + y\, z = \boldsymbol{x\, y + \bar{x}\, z}$

H = H_f $\Rightarrow$ f is undetectable

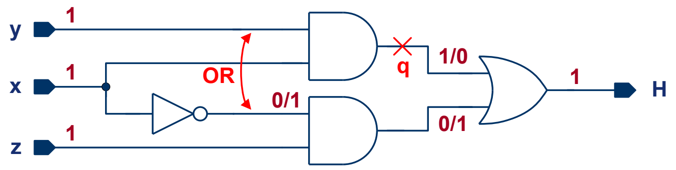
Are there any problems with undetectable faults?

Fault Interaction

The test xyz = 111 detects **q s-a-0**



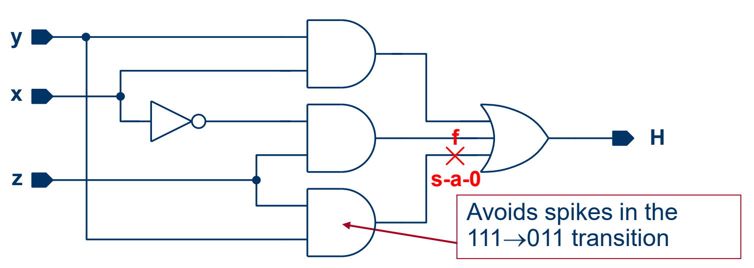
But xyz = 111 does not detect **q s-a-0** in the presence of another undetectable fault ⇒ **fault masking** might exist, even when one of the faults is undetectable



Redundancy and Detection

$H = x\,y + \bar{x}\,z + y\,z = \boldsymbol{x\,y + \bar{x}\,z}$

$H_f = \boldsymbol{x\,y + \bar{x}\,z}$ ⇒ f is undetectable



All combinational circuits with undetectable stuck-at faults can be simplified

In an irredundant circuit, all stuck-at fault are detectable

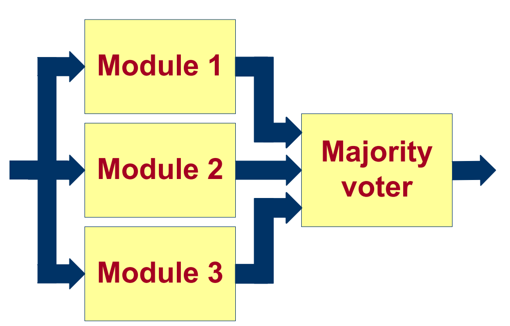
Disadvantages of Redundancy

- Testability ↓
- Complexity of Automatic Test Pattern Generation (ATPG) ↑↑
- Fault coverage evaluation ↑
- Circuit area ↑
- Propagation delays ↑
- Power consumption ↑
- Manufacturing yield ↓

Is redundancy always disadvantageous?

Triple Redundancy

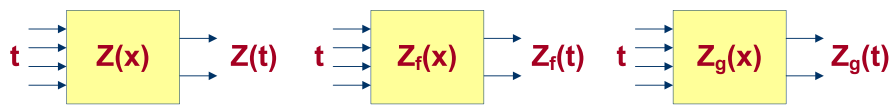
Modules 1, 2 e 3 have identical functionalities, but the implementations are distinct



The triple redundancy …

- is widely used in the design of critical systems
- has a negative impact in the system cost (larger silicon area or larger number of components)
- is an alternative to the on-line testing

Fault Equivalence



f and **g** are functionally equivalent if: (f $\equiv$ g) $\Leftrightarrow$ $\mathbf{Z_f(x) = Z_g(x)}$

t distinguishes between **f** and **g** $\Leftrightarrow$ $\mathbf{Z_f(t) \neq Z_g(t)}$

For a circuit with a single output, the **set of all tests that distinguish between f and g** are the solutions of:

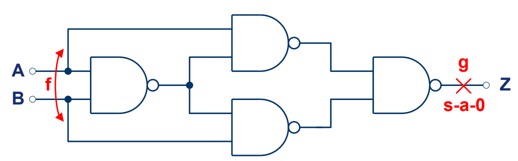
$$\mathbf{Z_f(x) \oplus Z_g(x) = 1}$$

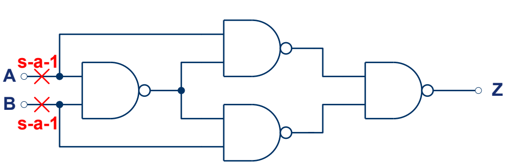
Examples of Fault Equivalence

f: Short circuit between A and B (wired-and)

g: Z s-a-0

$\Rightarrow$ f $\equiv$ g



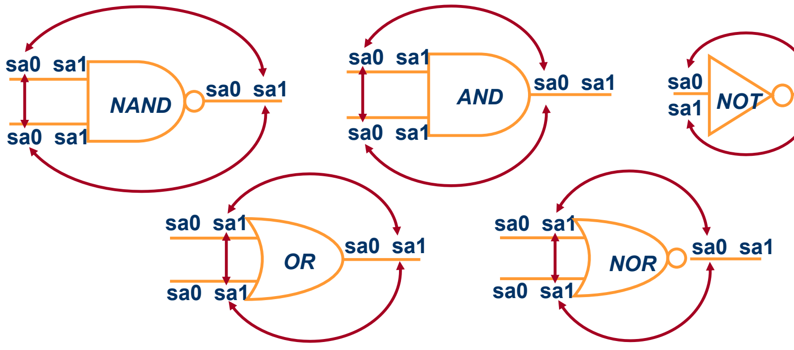


f: A s-a-1

g: B s-a-1

$\Rightarrow$ set of tests that distinguish between f and g: $\mathbf{\overline{A} \oplus \overline{B} = 1}$

Fault Collapsing based on Fault Equivalence

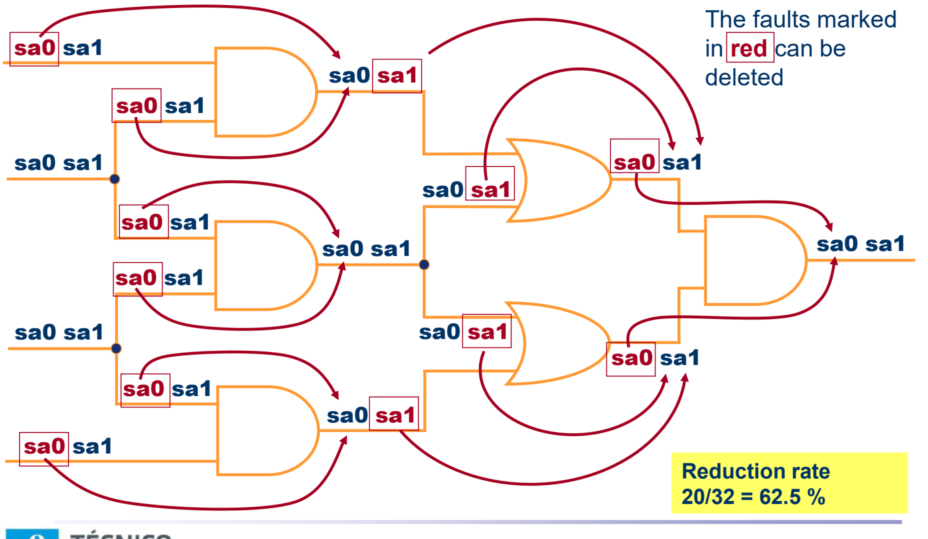


Fault Collapsing: A **representative fault** is selected from each set of equivalent faults. The remaining faults can be deleted.

The weight of the **representative fault** must be equivalent to the weights of the collapsed faults.

25

Example of Fault Collapsing



The faults marked in red can be deleted

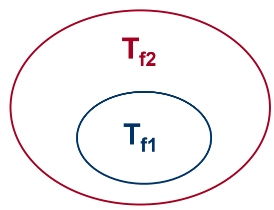
Reduction rate
20/32 = 62.5 %

26

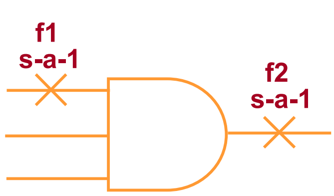
Fault Dominance

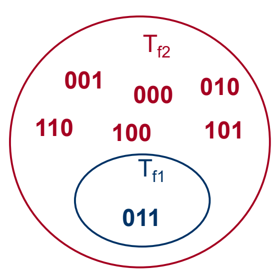
- If all tests that detect f1 also detect f2, then we can state that f2 dominates f1
- If a fault f2 dominates f1, then f2 can be deleted from the fault list
- If two faults dominate mutually, then those faults are equivalent

T_{f1} = {all tests that detect f1}
T_{f2} = {all tests that detect f2}



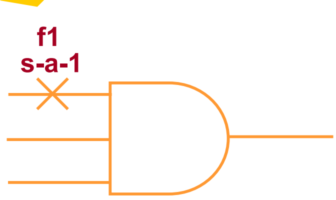
Example of Fault Dominance





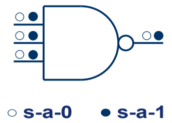
f2 dominates f1

f2 can be deleted



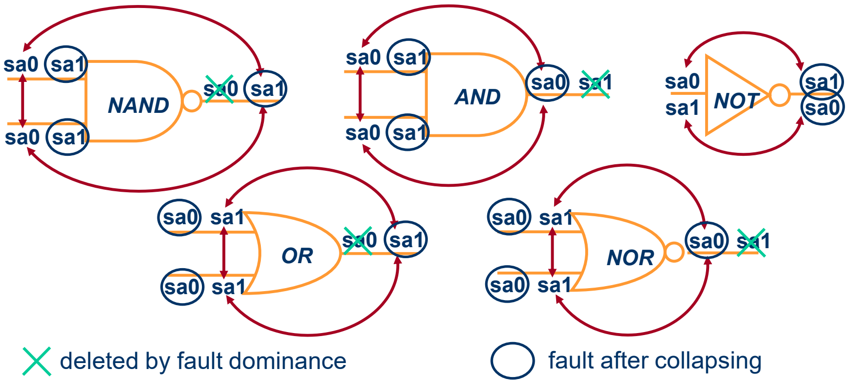
The fault dominance rules can be also used to perform fault collapsing

Fault Collapsing and Dominance: Summary



n inputs ⇒ 2(n + 1) faults (single stuck-at)

but **n + 1** faults can be deleted by fault equivalence or fault dominance



29