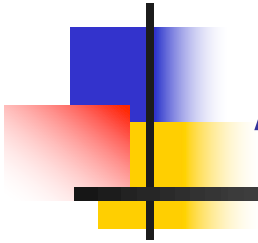
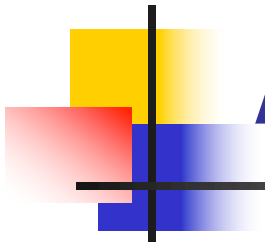


Chapter 2
Fault Modelling

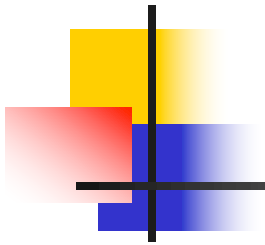
Arnaud Virazel

virazel@lirmm.fr



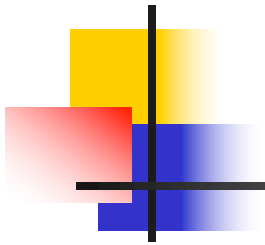
Agenda

- **Introduction**
- Defect characterisation
- Defect modelling
  - Stuck-at
  - Bridging
  - Delay
- Fault equivalence
- Example/Exercise



How to get a test sequence?

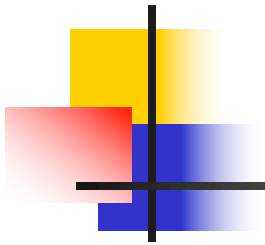
- Use a **functional sequence** produced to verify the DUT by simulation during the design phase
  - easy to find
  - more difficult to validate
  - even more difficult to improve



How to get a test sequence?

- Apply an **exhaustive sequence**
  - Much too long
  - Example
    - A 64 inputs circuit
    - $2^{64} = 18^{18}$ vectors
    - ATE frequency: 1 GHz
    - Test application time: about 585 years

$\Rightarrow$ Structural Test – Use of fault model

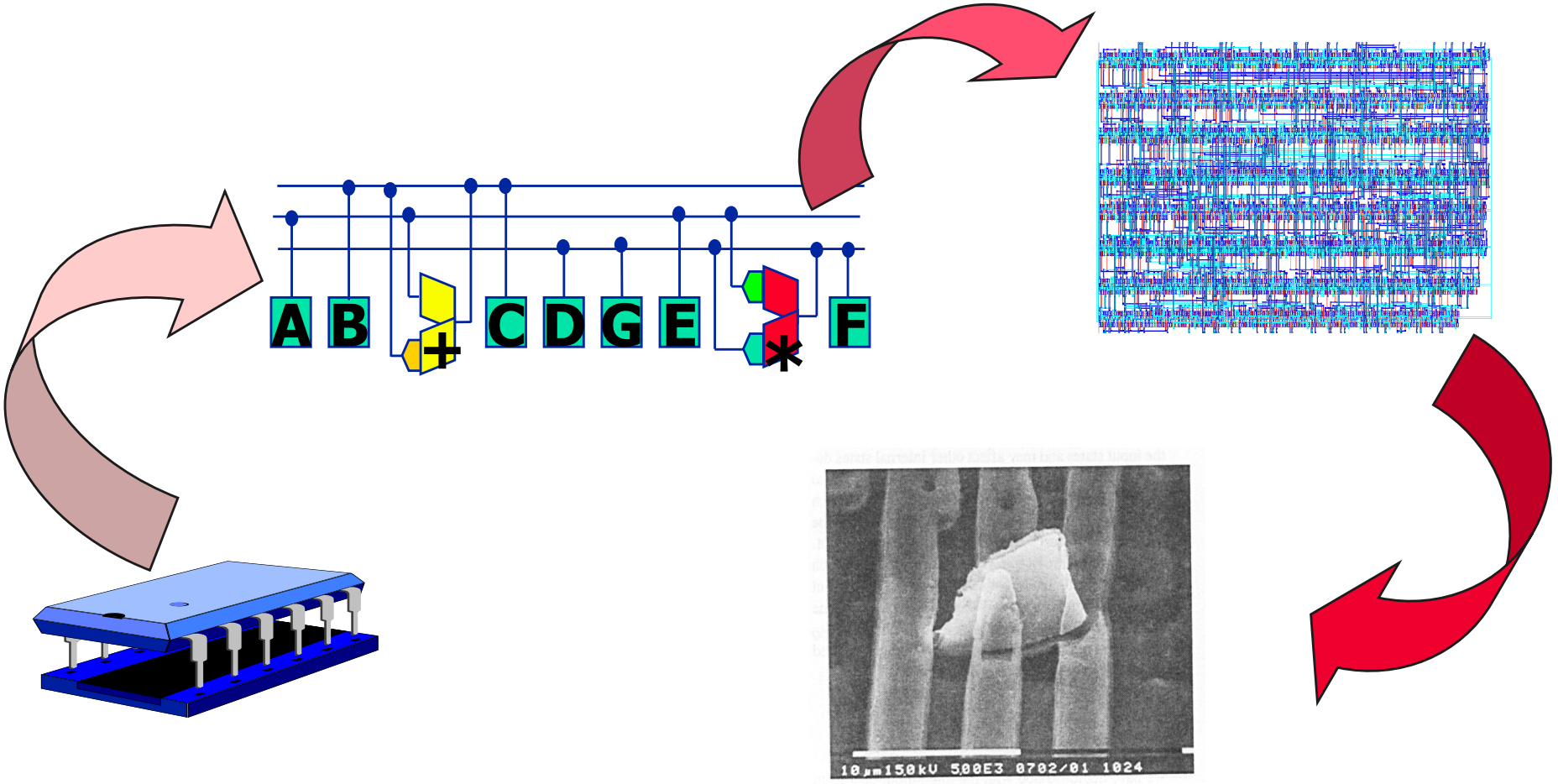


How to get a test sequence?

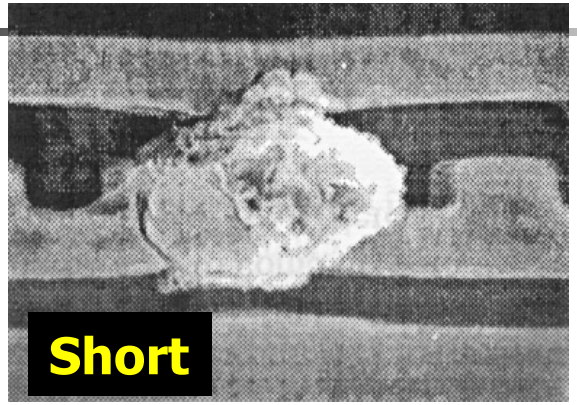
- **Structural Test**
  - Fault models
    - Coverage rate → "defect level"
  - EDA tools
    - Fault simulator
    - ATPG – Automatic Test Pattern Generator

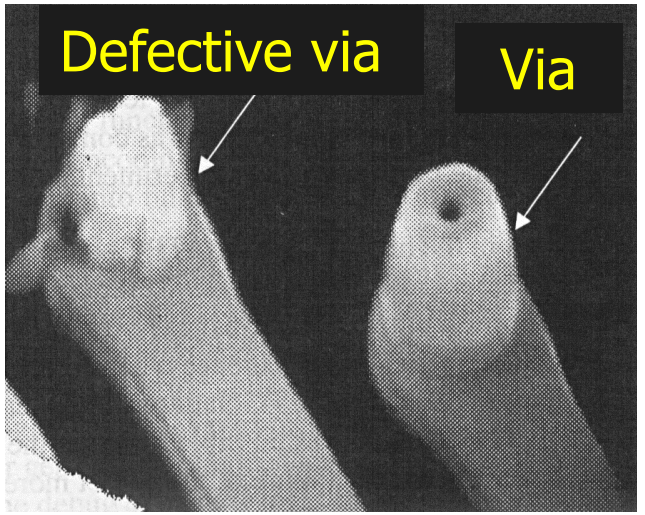
DUT levels

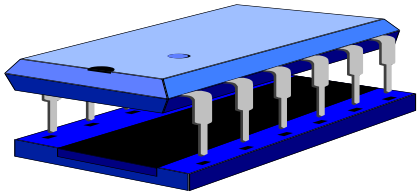
- Failures in the manufacturing process

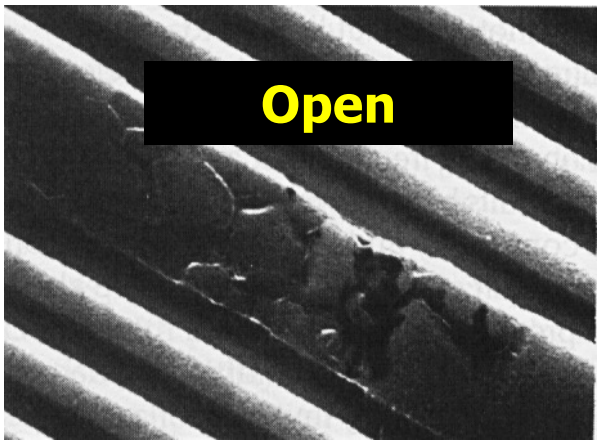


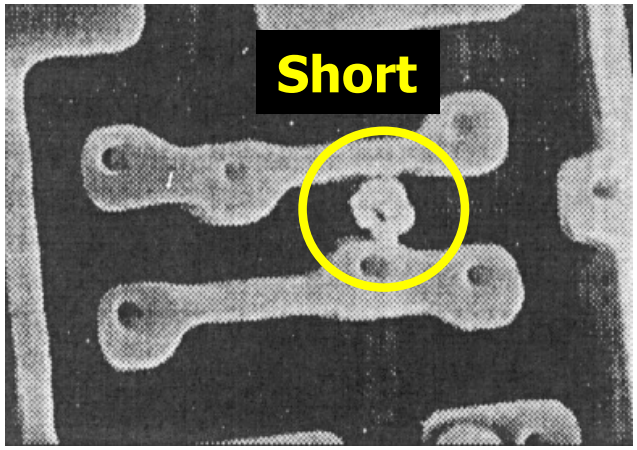
Some Defect Examples

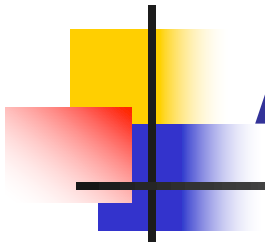










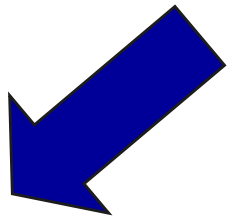


Agenda

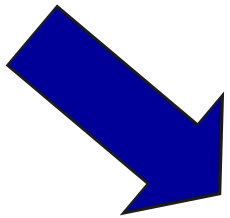
- Introduction
- **Defect characterisation**
- Defect modelling
  - Stuck-at
  - Bridging
  - Delay
- Fault equivalence
- Example/Exercise

Failure Mechanisms

- Wafer defects
- Human interactions with the manufacturing process
- Equipment failure
- Impact of the environment
- Technological process variations



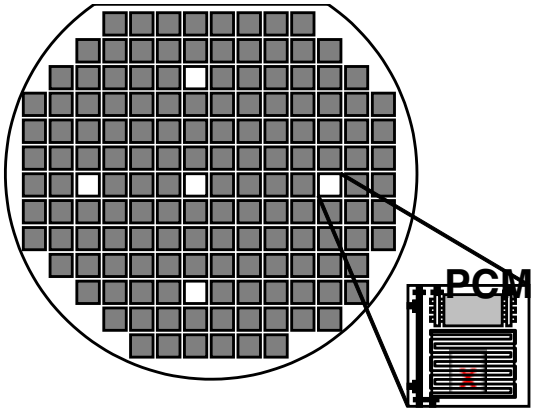
GLOBAL DEFECTS



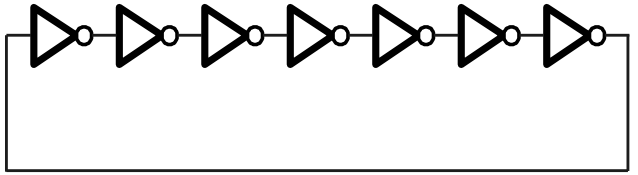
LOCAL DEFECTS

Monitoring of Global Defects

- PCM - "Process Control Monitor"
  - Composed of basic structures (transistors, metal lines, via chains, …)
  - Distributed on the wafer (placed on the cutting lines)

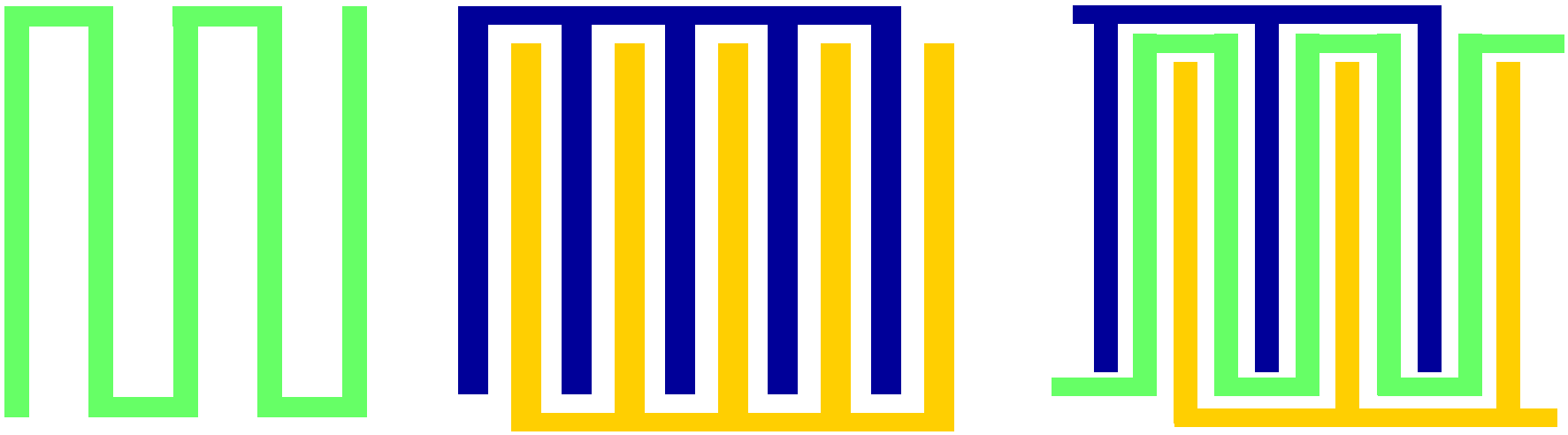


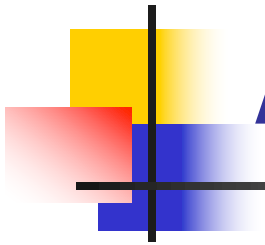
- Ring oscillator
  - Monitoring of high-level parameters
  - Oscillation frequency v.s. low level parameters of the technological process



Monitoring of Local Defects

- Online monitoring at different stages of the manufacturing process
- Gate oxide monitors
- Interconnect monitors



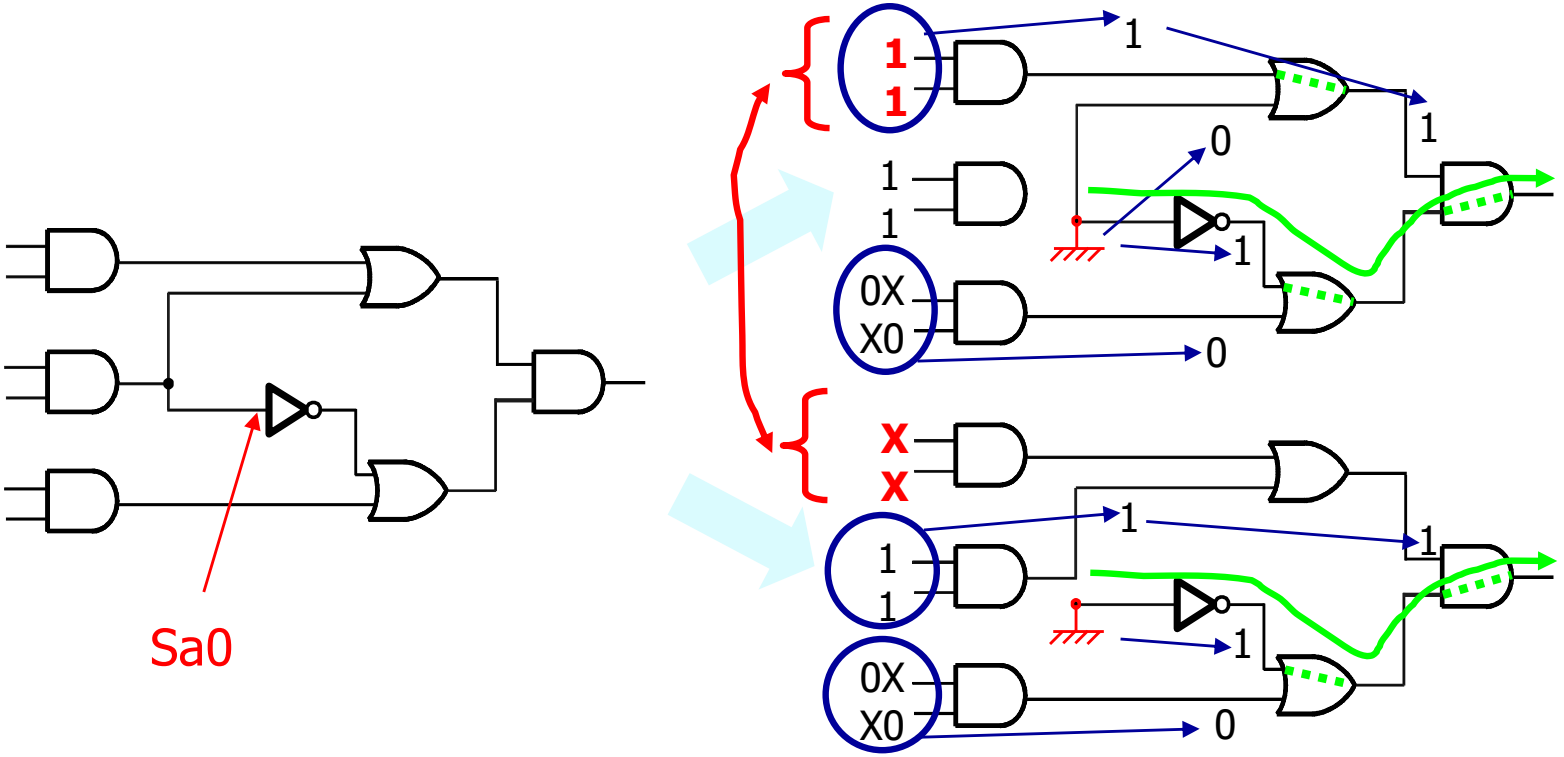


Agenda

- Introduction
- Defect characterisation
- **Defect modelling**
  - Stuck-at
  - Bridging
  - Delay
- Fault equivalence
- Example/Exercise

Stuck-at Fault Principle

- One and only one (single fault assumption) line of the DUT is stuck-at logic '0' or logic '1'



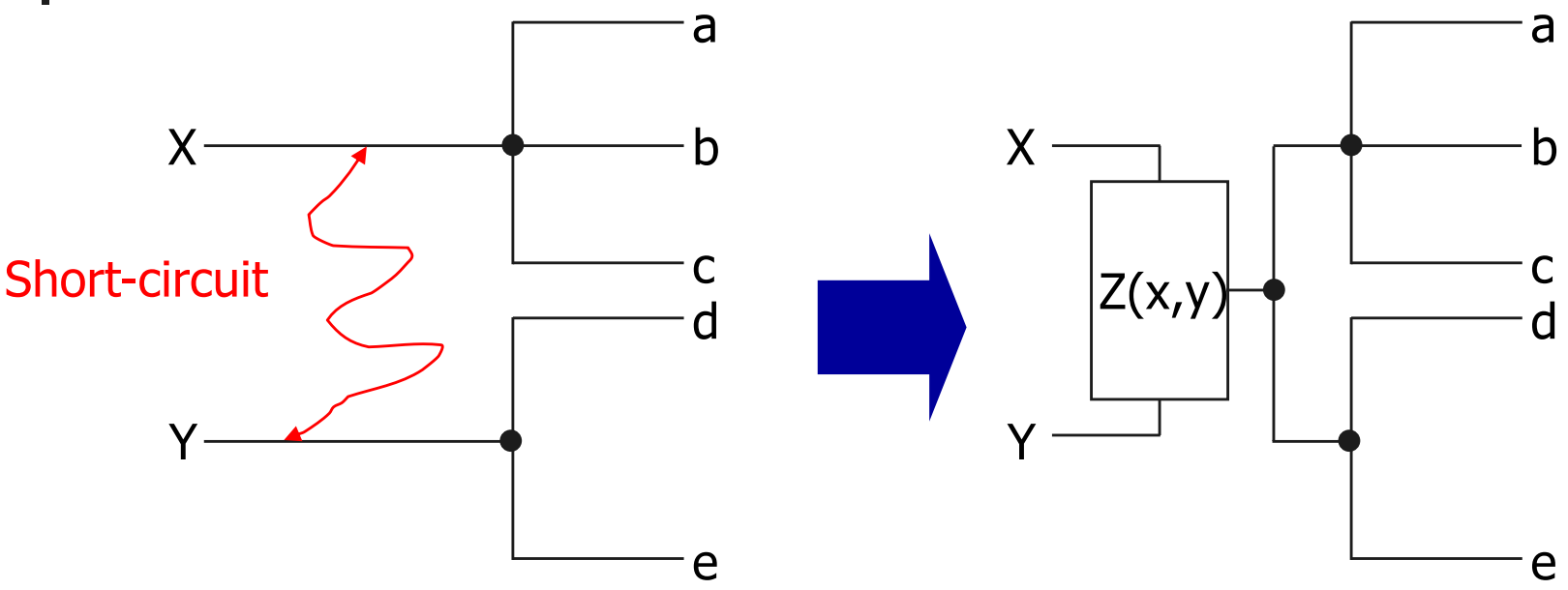
Stuck-at Fault Features

- It allows to represent many different physical defects
- It is technology independent
- It allows the usage of the Boolean algebra to find the test vectors
- Test vectors targeting SaF also detect other defects
- The set of faults obtained is limited
- The Test Coverage (TC) is an accepted metric between suppliers and customers

$$TC = \frac{Nbre\ de\ fautes\ détectées}{Nbre\ de\ fautes\ total}$$

- The SaF model can be used to model other types of faults

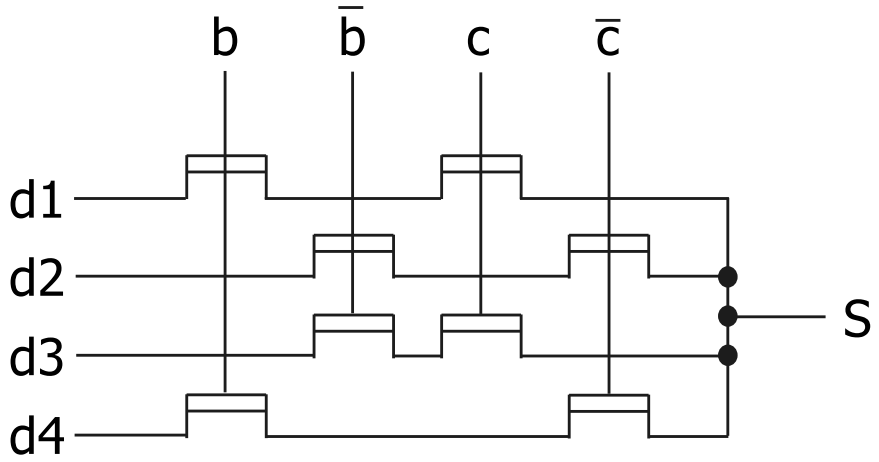
Bridging Fault Principle



- Z(x,y) = z
  - Z(0,1) = 0 → **wire AND**
  - Z(0,1) = 1 → **wire OR**

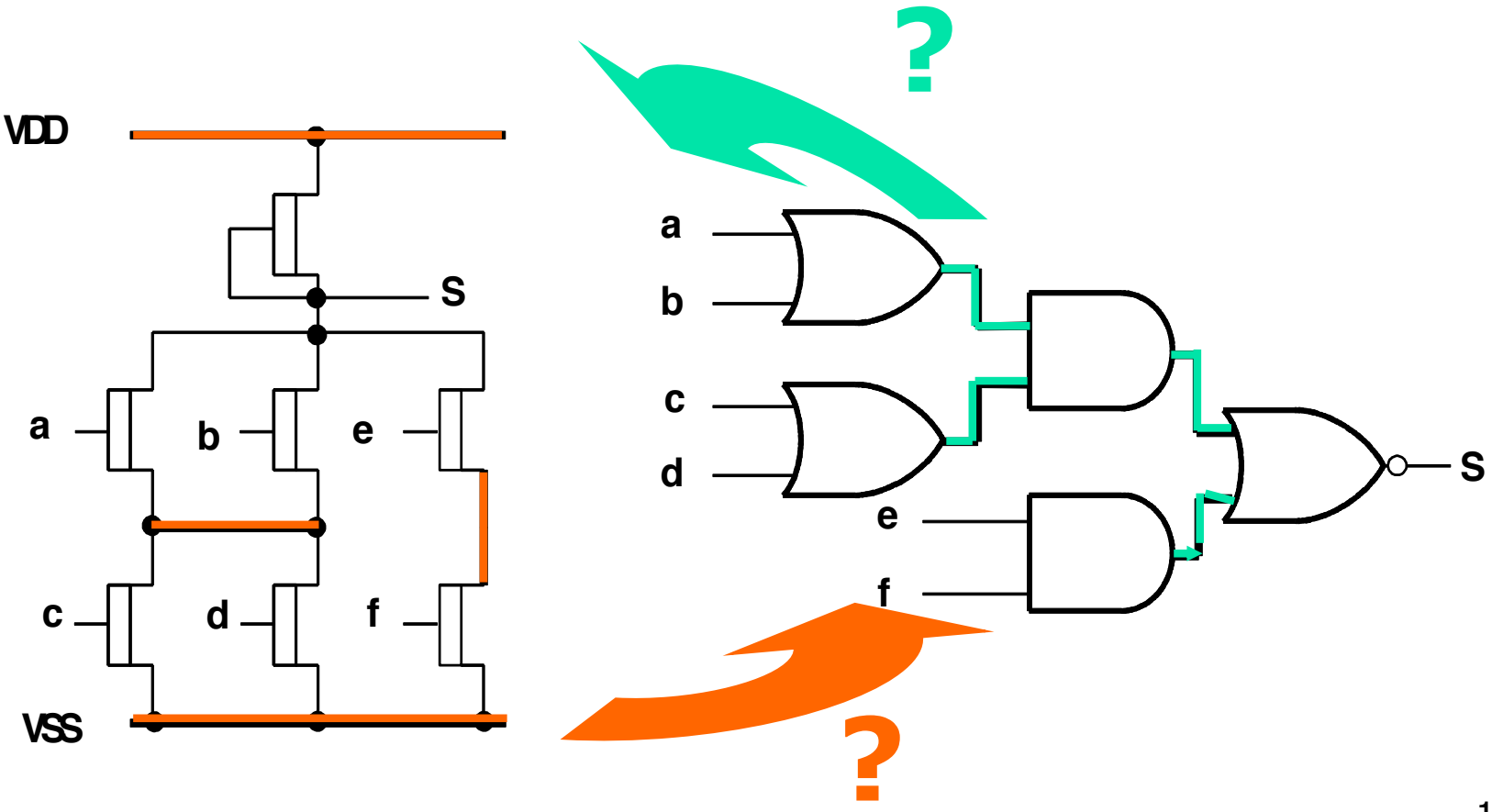
SaF Model Issues

- Gate level modelling
  - Multiplexer structure → LUT



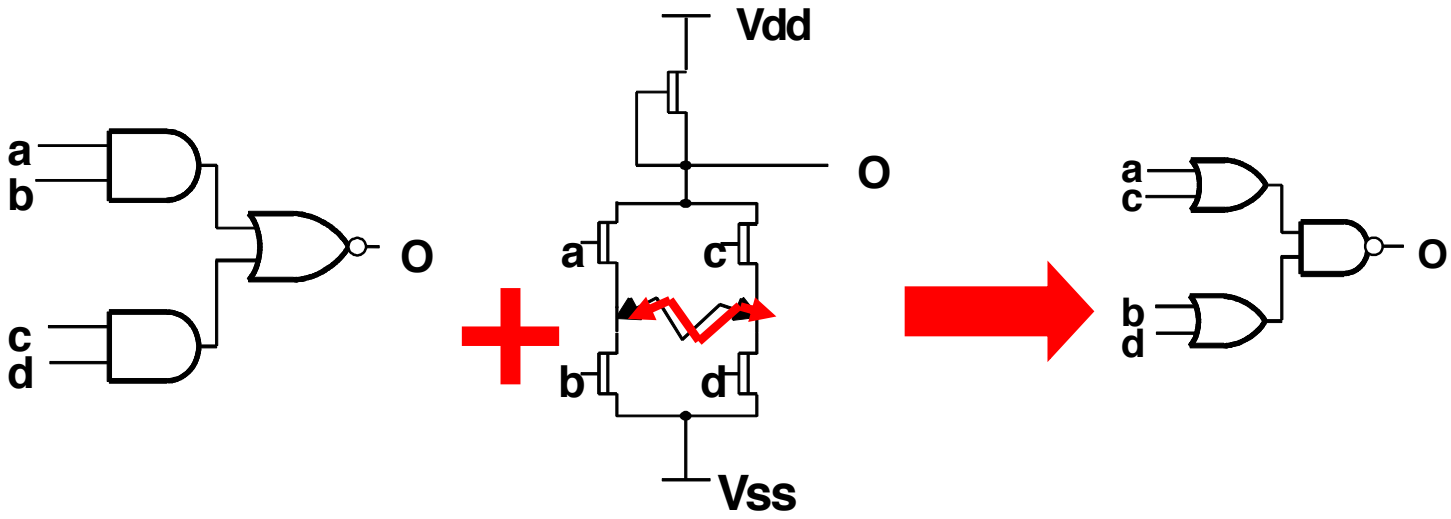
SaF Model Issues

- Transistor level v.s. Gate level



SaF Model Issues

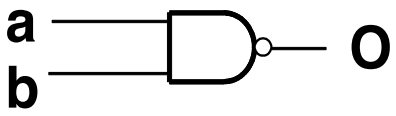
- Modification of the logic function

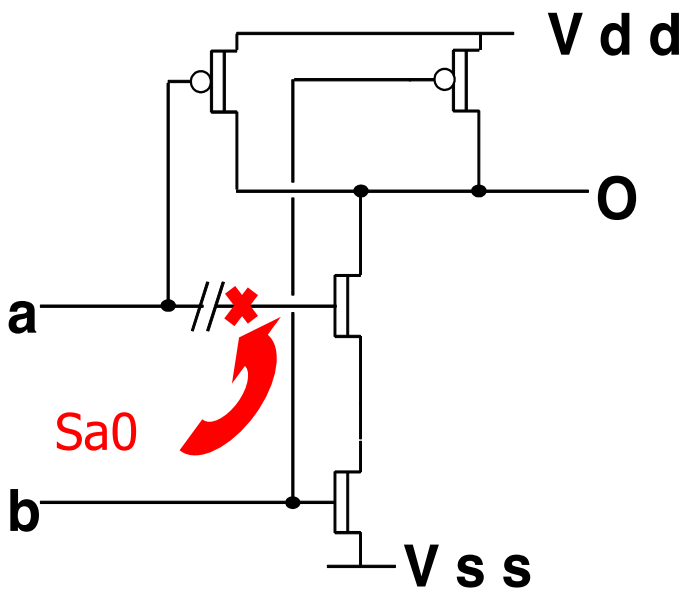


a	b	c	d	O	Detected SaF	O (with defect)
0	1	0	1	1	Sa1(a) Sa1(c)	1
1	0	1	0	1	Sa1(b) Sa1(d)	1
1	1	0	X	0	Sa0(a) Sa0(b)	0
0	X	1	1	0	Sa0(c) Sa0(d)	0

SaF Model Issues

- Memory effect

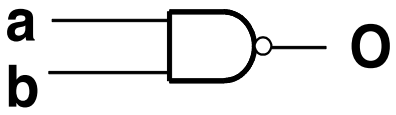


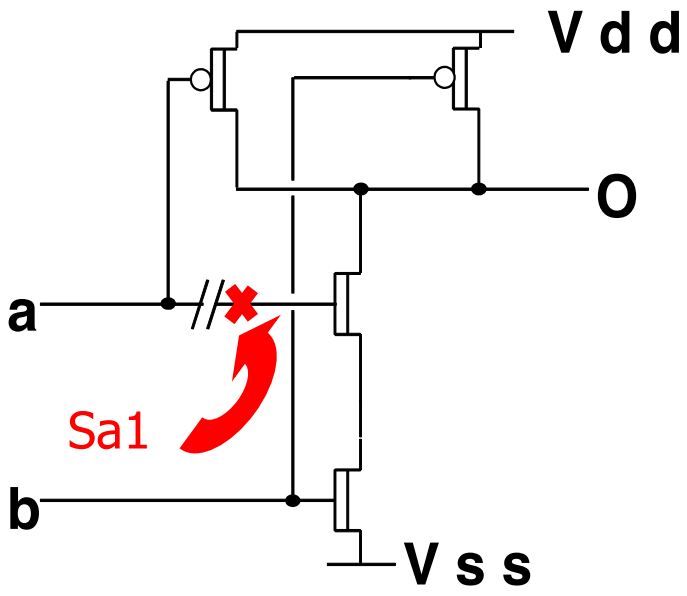


	a	b	O	O (with the fault)
V1	0	0	1	1
V2	0	1	1	1
V3	1	0	1	1
V4	1	1	0	?

SaF Model Issues

- Analog behaviour





	a	b	O	O (with the fault)
V1	0	0	1	1
V2	0	1	1	?
V3	1	0	1	1
V4	1	1	0	0

Unknown value
(Depends on Ron of
active transistors)

SaF Model Issues

Unknown value

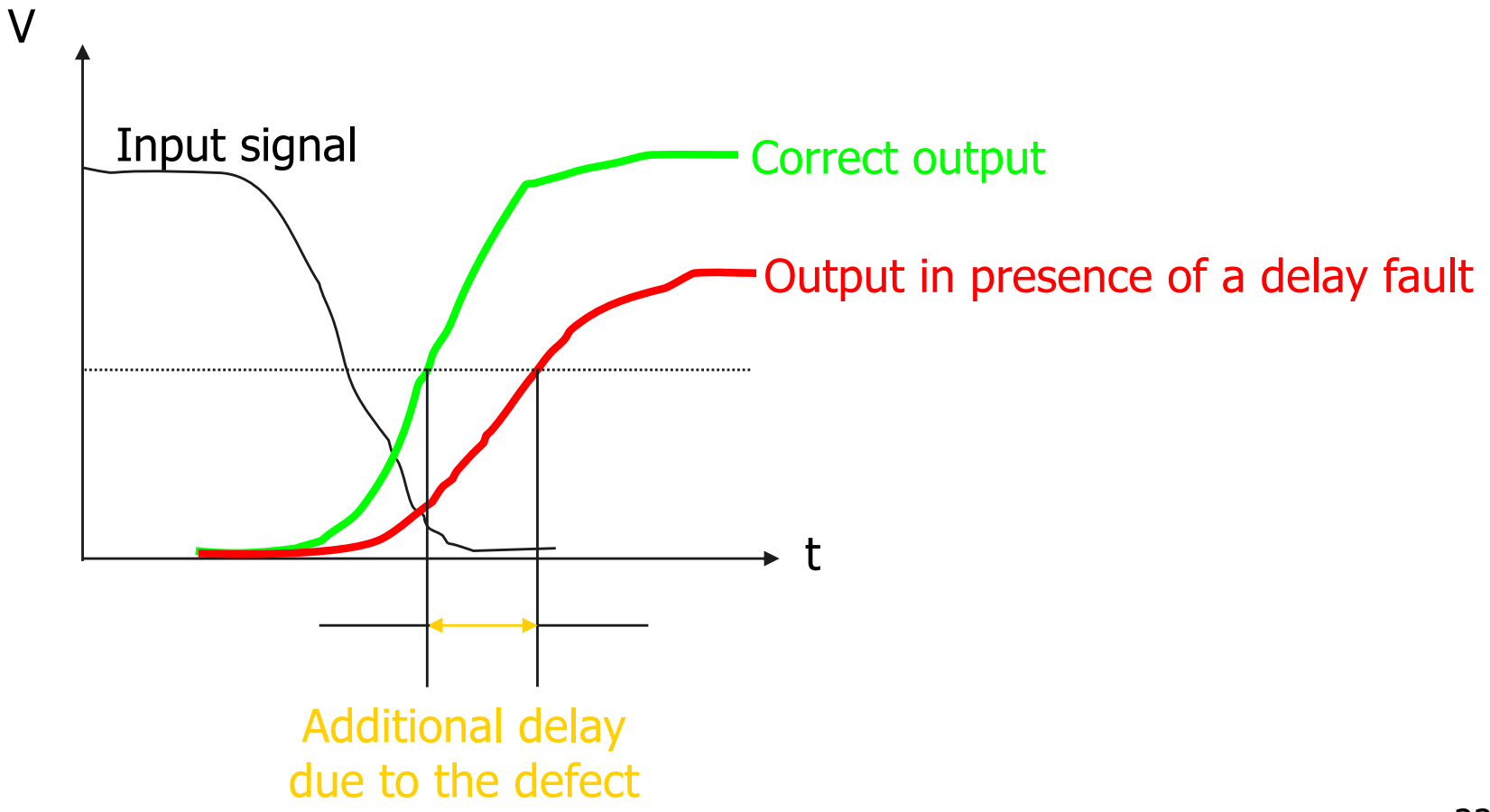
Erroneous logic value

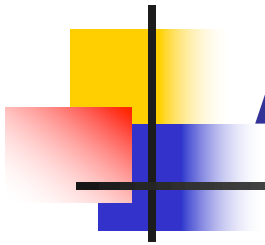
- Possibly detected by a classical logic test
- Augmentation du courant consommé ("IDDQ testing » - Idd quiescent)

Correct logic value

- No logic error – undetectable by a static test
- Higher power consumption ("IDDQ testing")
- Higher propagation delay – Delay Fault

Delay Fault Principle



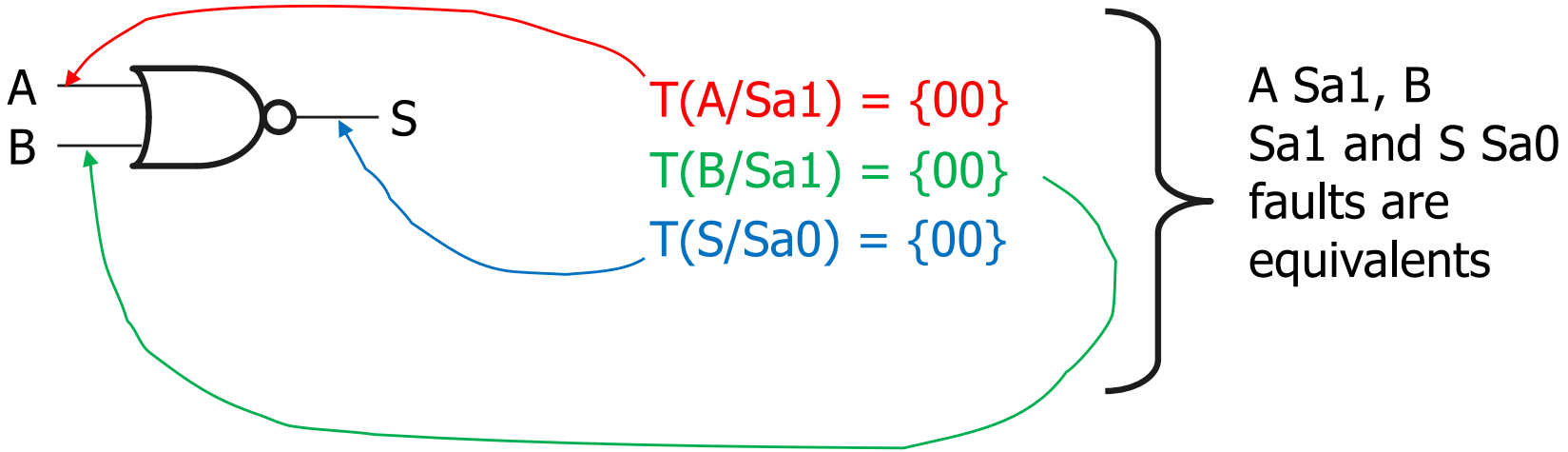


Agenda

- Introduction
- Defect characterisation
- Defect modelling
  - Stuck-at
  - Bridging
  - Delay
- **Fault equivalence**
- Example/Exercise

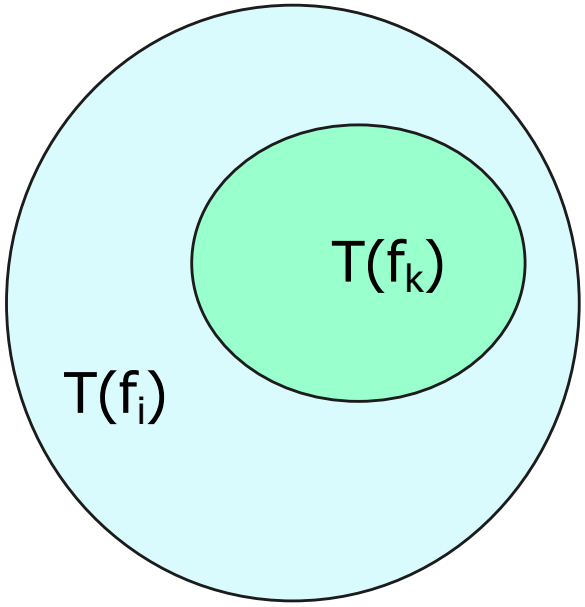
Definition - Equivalence

- $T(f_i)$ is the set of tests that detect the fault f_i
- Two faults f_i et f_j are **equivalents** if and only of $T(f_i) \equiv T(f_j)$
- All test detecting f_i also detects f_j (reciprocally)

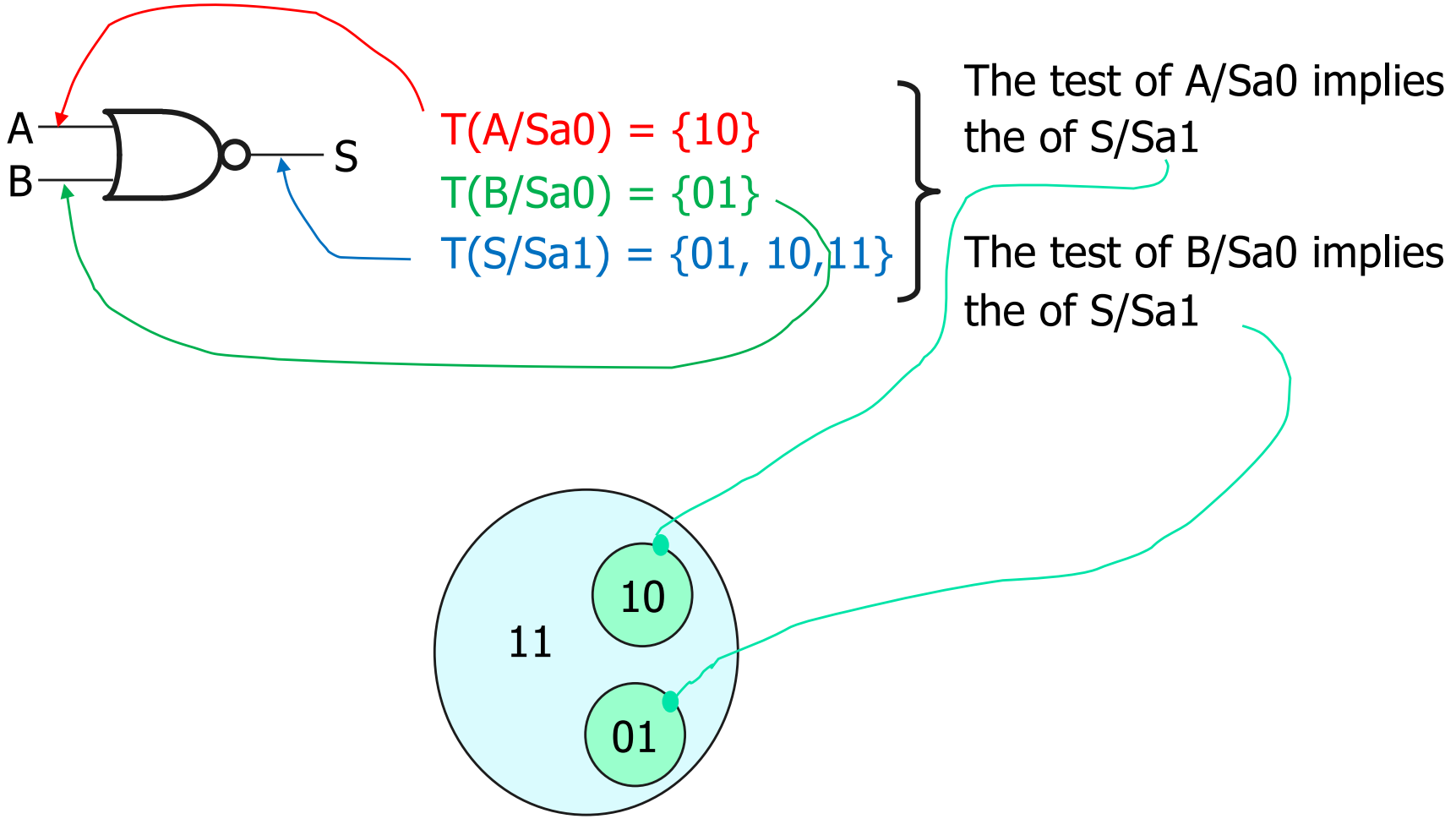


Definition - Implication

- A fault f_i **dominate** the fault f_k if and only if $T(f_k)$ C $T(f_i)$
- All test detecting f_k also detects f_i
- The test of f_k **imply** the of f_i



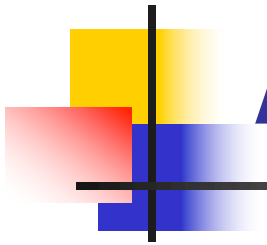
Example


A
B
S
T(A/Sa0) = {10}
T(B/Sa0) = {01}
T(S/Sa1) = {01, 10,11}
The test of A/Sa0 implies the of S/Sa1
The test of B/Sa0 implies the of S/Sa1
10
11
01

Generalisation

IN/OUT			Detected SaF (#)					
A	B	C	Sa0(A)	Sa0(B)	Sa1(A)	Sa1(B)	Sa0(S)	Sa1(S)
0	0	1			#	#	#	
0	1	0		#				#
1	0	0	#					#
1	1	0						#

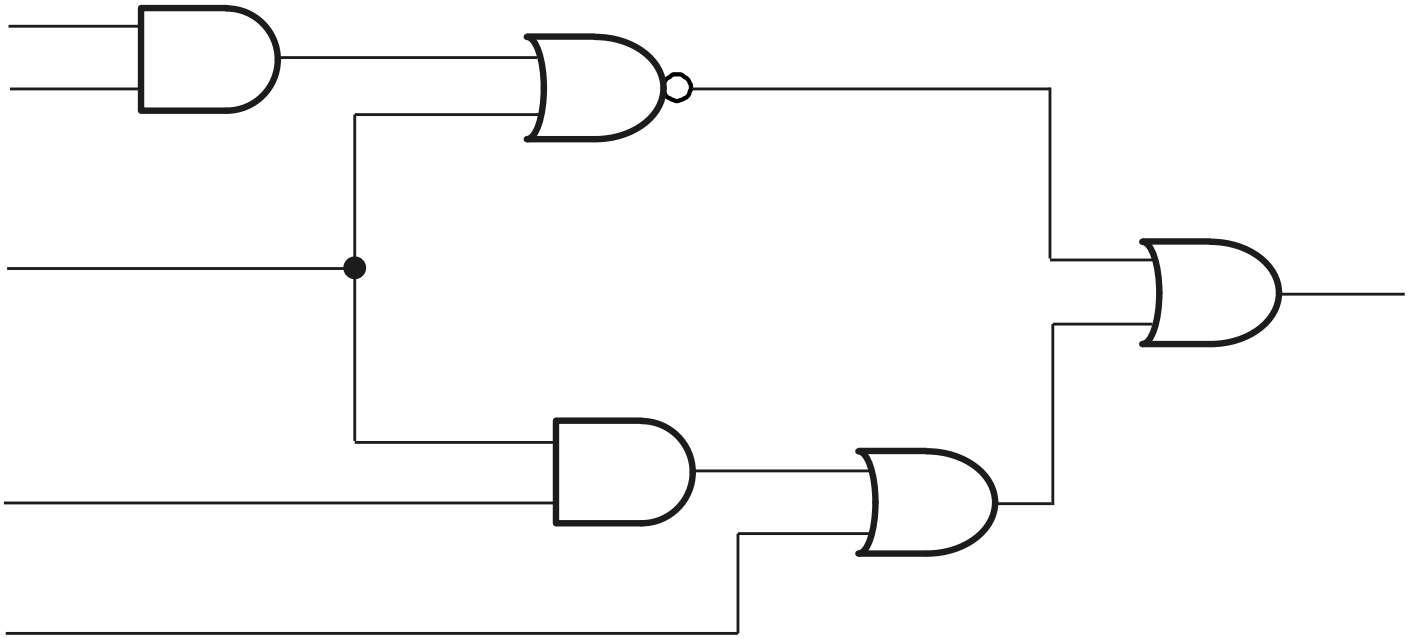
- For a gate with a priority value c and an inversion i, all stuck-at fault at c of an input is equivalent to the stuck-at $c \oplus i$ (equivalence) of the output
- For a gate with a priority value c and an inversion i, all test detecting stuck at fault at $\overline{c}$ of an input implies the test of stuck-at $\overline{c \oplus i}$ (implication) of the output

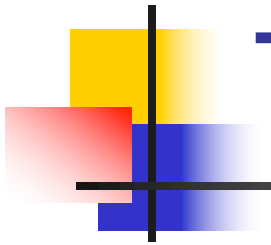


Agenda

- Introduction
- Defect characterisation
- Defect modelling
  - Stuck-at
  - Bridging
  - Delay
- Fault equivalence
- **Example/Exercise**

Exemple





Test Generation Steps

- Fault **sensitization**
  - Apply the logic value to sensitize the fault
    $\Rightarrow$ 0(1) for a Sa1(0)
- Fault effect **propagation**
  - Open a non-masking (non priority values on side inputs) propagation path form the fault site to at least one primary output (an observable point) of the DUT
- **Justification**
  - Justify all logic values fixed during sensitization and propagation steps to primary inputs of the DUT