Fault Models

Fault (failure) is a physical event

Error is a manifestation of a fault

Logic-level faults

Fault model is necessary for efficient test generation

Statistical analysis of faults to construct fault models.

Fault model depends on implementation and environment.

Standard Fault Models

I. Stuck-at faults - a line Z is stuck-at-0 or stuck-at-1 (Z/0, Z/1 or Z≡0, Z≡1)

Single (SSF) and multiple (MSF) faults

SSF is the most popular model

Stuck-at faults are detected by off-line and by on-line testing

Fault coverage - percentage of faults which are detected)

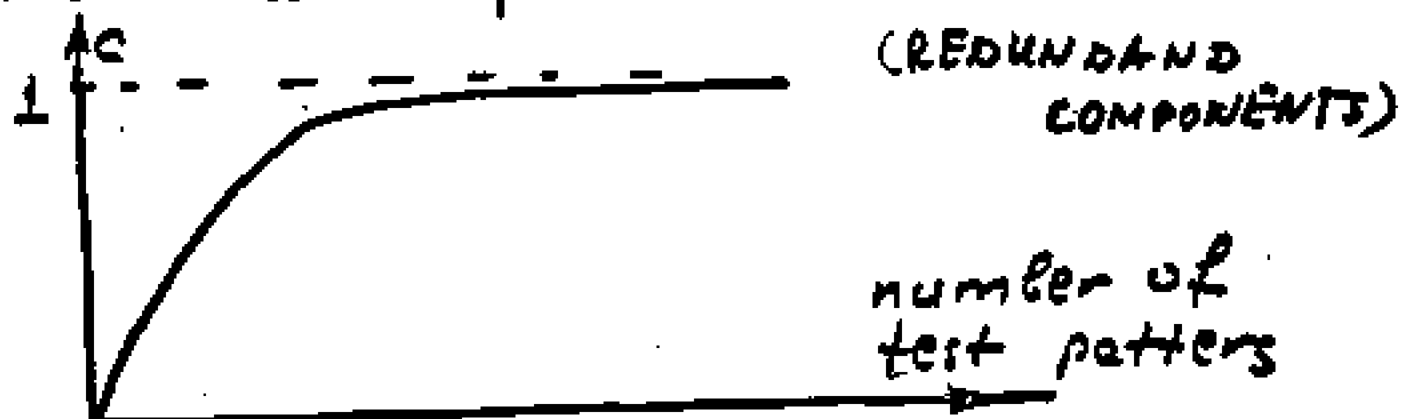
(Example: For SSF and μprocessors 98% is a reasonable fault coverage)

Fault-Coverage for SSF

Fault Coverage C is a percentage of SSF which are detected

SSF - SINGLE STUCK-AT FAULTS

For complex devices (ex. μprocessors) about 2% of SSF are undetectable (REDUNDAND COMPONENTS)



Estimation of fault-coverage by simulation

Software for fault simulation is expensive

Up to 20,000 SSF's can be simulated in a reasonable CPU time

Number of stuck-at faults with a multiplicity at most ℓ

$$\sum_{i=1}^{\ell} 2^i \binom{L}{i} \; ; \quad \binom{L}{i} = \frac{L!}{i!\,(L-i)!} \; ; \quad a! = 1 \cdot 2 \cdot 3 \cdot \ldots \cdot a$$

L is a number of lines in a network

Example $L = 1{,}000$, $\ell = 2$

$2 \cdot 10^6$ faults

Special Classes of Stuck-at Faults

Bursts of length b (for discs or tapes)

Number of Bursts: $N-b+1$ N is a number of binary cells

Symmetric, asymmetric and unidirectional stuck-at faults

Number of unidirectional faults with a multiplicity at most l: $\sum_{i=1}^{l} \binom{L}{i}$ Ex: $L=1{,}000$; $l=2 \Rightarrow 500{,}500$ faults

Input / Output (Terminal or Pin) stuck-at faults at interconnections between chips

Number of I/O stuck-at faults for a chip with m inputs and k outputs: $\sum_{i=1}^{l} 2^i \binom{m+k}{i}$; Ex: $m=k=16$; $l=2 \Rightarrow 2{,}048$ faults (symmetrical)

Fault Models (Cont.)

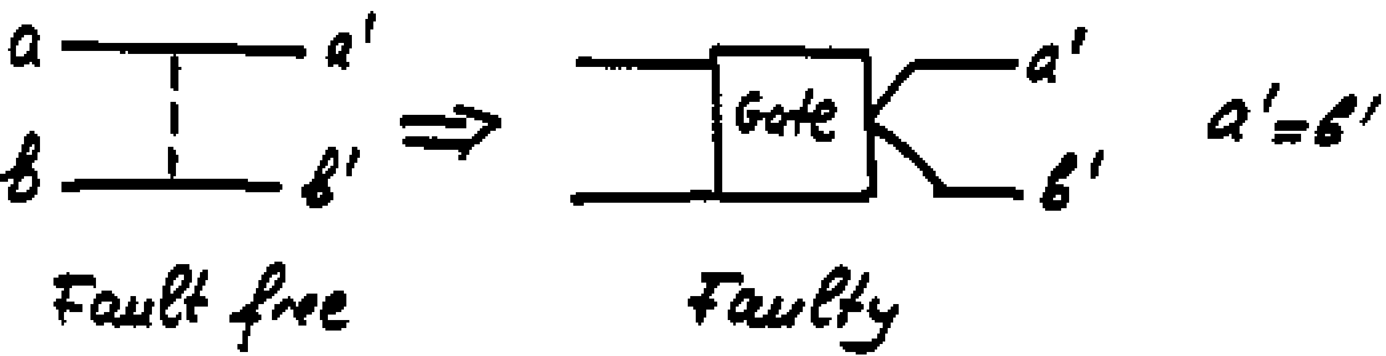
II <u>Intermittent</u> (<u>transient</u>, <u>soft</u>) faults

Detected only <u>by</u> <u>on-line</u> tests
(<u>ex.</u> replication of hardware, parity checks, error-correcting codes, roll-backs of a program, etc.)

<u>Single</u> intermittent faults
(α-particles) vs <u>repeating</u> faults

Up to 80% - 90% of faults are intermittent (Air Force, IBM)

III Bridgings (BF) (short circuits)
(high density of gates for VLSI)



. AND and OR type bridgings

AND: $(a, b)_*$

OR: $(a, b)_+$

a	b	AND $a' = b'$	OR $a' = b'$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

Relation between stucks and bridgings:

$a/0 = (a, 0)_*$

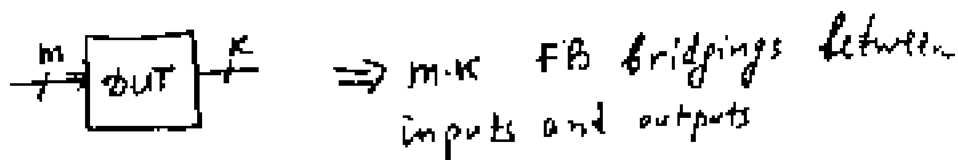
$a/1 = (a, 1)_+$

Number of bridgings:

$$2\binom{L}{2} = L(L-1)$$

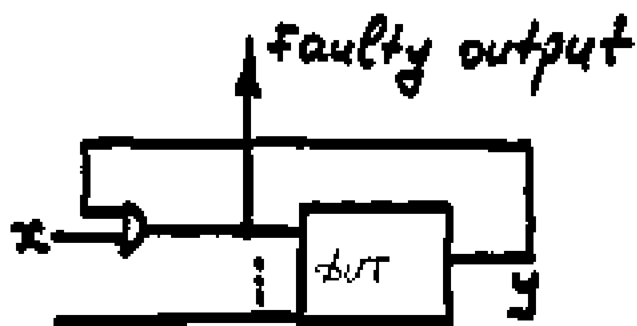
Feedback (FB) and NonFeedback (NFB) 1.8

Bridgings



⇒ m·k FB bridgings between inputs and outputs

FB $(x,y)_*$

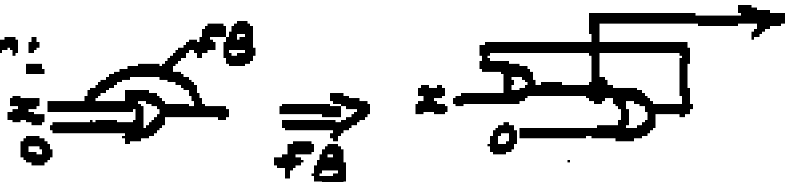


Combinational $\xrightarrow{FB}$ Sequential

Oscillation:
$(x,y)_*$

difficult to observe

Asynchronous Behavior:
$(x,y)_*$



(For any x output stay 0 after x=0 applied)

FOR CMOS CIRCUITS

BRIDGING FAULTS ACCOUNT FOR

FROM 30% to 60% of all

FAULTS

(F. J. Ferguson, J. P. Shen, "Extraction and Simulation of Realistic CMOS Faults ...," Proc. Int. Test Conf. pp. 475-484, 1988)

BFs are detected by current testing (I_{DDQ} testing)

(Rajsuman, IDDQ TESTING FOR CMOS VLSI, ARTECH HOUSE, 1995)

IV Arithmetical Faults (ALU) 1.9

Typical for adders, subtractors counters, multipliers, etc.

$$x \xrightarrow[\text{fault}]{} \hat{x}$$

Multiplicity l: $|x-\hat{x}| = \pm 2^{i_1} \pm \ \ \pm 2^{i_l}$

Example: $x=0111$, $\hat{x}=1000 \Rightarrow l=1$
(For stuck-at fault model $l=4$)

Number of faults: $\sum_{i=1}^{l} 2^i \binom{m}{i}$ - symmetrical

m- number of bits $\sum_{i=1}^{l} \binom{m}{i}$ - unidirectional

Symmetric, asymmetric and unidirectional arithmetical faults

Arithmetical codes (modulo A checks)

Example $l=1 \Rightarrow A=3$

V PLA Faults

1.10

1. A product term can grow (cover more minterms)

2. A product term can shrink (cover fewer minterms)

3. A product term can dissappear from a function

4. A product term from one function can appear also in another function

VI Memory faults

5.1. <u>Decoding errors</u>
open decoder ⟹ no addressing
multiple writes, two addresses at a time
replacement of an address by another

5.2. <u>Stuck-at</u> faults in binary cells
5.3. <u>Bridgings</u> between cells
5.4 <u>Bursts</u> on a surface of a disc or tape

<u>For semiconductor memories:</u>

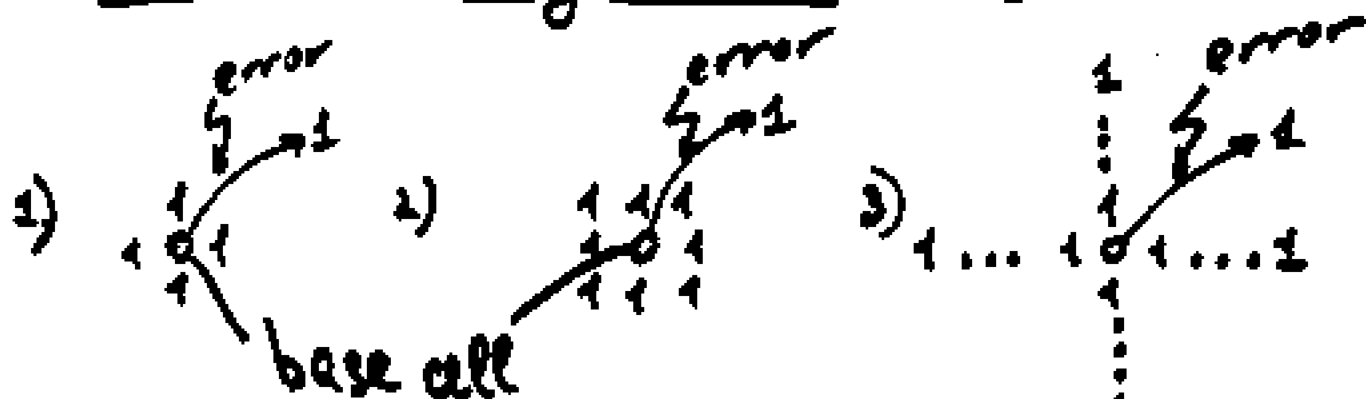
5.5 <u>Hold time</u> for refreshing data ⟹
<u>sleeping sickness</u> - graceful degradation of performance

5.6. <u>Write recovery</u> - not producing data at a given access time if each READ is preceded by WRITE

5.7. Pattern-sensitive faults (crosstalks) –

impossible to write 0 surrounded by 1's

Electrical neighbourhoods a) NPSF



These faults are most difficult to test

b) k-couplings – pattern sensitive faults (crosstalks) between any k cells which may be located anywhere in the memory.

Testing time (test complexity, number of READ and WRITE operations) is proportional to the size of a neighbourhood

VII Functional Faults (μprocessors)

7.1. Replacement of one instruction I_i by another I_j, I_i/I_j

7.2. Replacement of an instruction I_i by no instruction, $I_i/\emptyset$

7.3 Replacement of an instruction I_i by two instructions I_j and I_k, I_i/I_j+I_k

These faults are detected by functional testing (function verification)

EXAMPLE

FUNCTIONAL FAULT [DESIGN ERROR]
IN PENTIUM CHIP (INTEL)
[$6.2 \cdot 10^6$ copies sold in 1994]

FUNCTIONAL FAULT: 1 out of 10^{10} DIVISIONS PRODUCE FAULTY RESULT

FUNCTIONAL TEST:

$$x - (x/y) * y = \begin{cases} 0 & \text{- FAULT-FREE} \\ 256 & \text{- faulty} \end{cases}$$

$x = 4{,}195{,}835$

$y = 3{,}145{,}727$

VIII. Network faults

Faults in distributed systems

8.1. Node faults

Multiplicity of a node fault = number of processors in the system which are faulty at the same time.

8.2. Link faults

Multiplicity of a link fault = number of links between nodes which a faulty at the same time.

∴ Optimal test strategy for a given topology of links

∴ Optimal topology of links for a given class of faults.