EC500

Design of Secure and Reliable Hardware

Lecture 3

Mark Karpovsky
January 24th, 2013

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

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

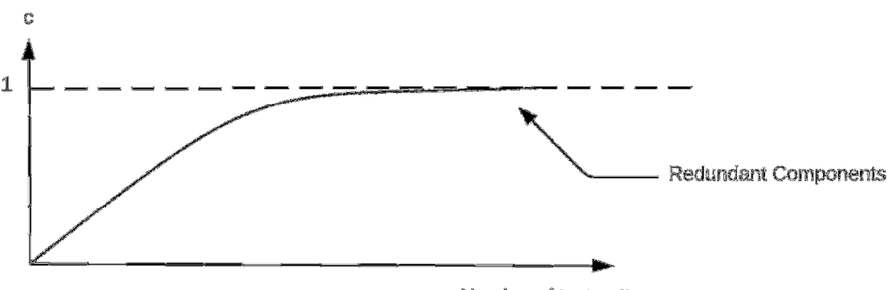
Stuck-at faults are fetected by off-line and by on-line testing (error detection)

Fault coverage – percentage of faults which are detected
Example: For SSF and μ processors, 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. μ processors) about 2% of SSF are undetectable



Estimation of fault-coverage by simulation software for fault simulation is expensive. Up to 100,000 SSF's can be simulated in a reasonable CPU time.

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

$$\sum_{i=1}^{l} 2^i \binom{L}{i}, \qquad \binom{L}{i} = \frac{L!}{i!\,(L-i)!}, \qquad i = 1,2,3,\ldots$$

L is the number of lines in a network

Example: $L = 1000, l = 2 \rightarrow 2 \times 10^6$ 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

$$\sum_{i=1}^{l} \binom{L}{i}$$

Example: $L = 1000, l = 2 \rightarrow 500500$ faults

I. **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

$$\sum_{i=1}^{l} 2^i \binom{m+k}{l}$$

Example: $m = k = 16, l = 2 \rightarrow 2048$ faults (symmetrical)

II. **Intermittent (transient, soft) faults:**

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

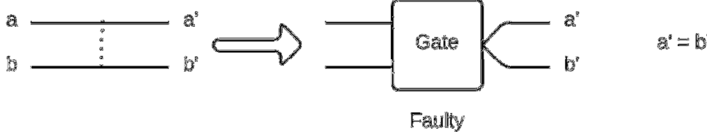
Single intermittent faults

($\propto$-particales) vs repeating faults

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

III. **Bridging(BF) (short circuits)**

(high density of gates for VLSI)



AND and OR type bridgings:

AND: $(a, b)_*$

OR: $(a, b)_+$

a	b	AND	OR
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

Relation between stucks and bridgings:

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

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

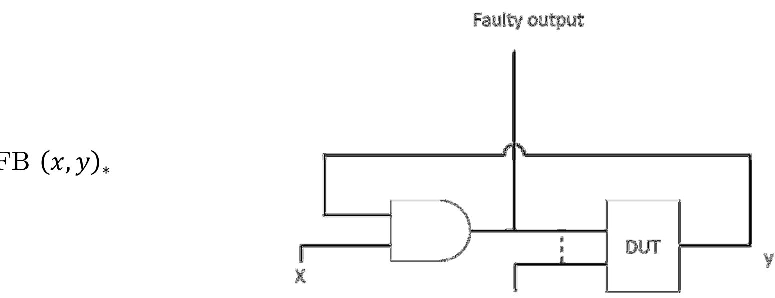
Number of bridings:

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

Feedback (FB) and Non-Feedback (NFB) Bridgings

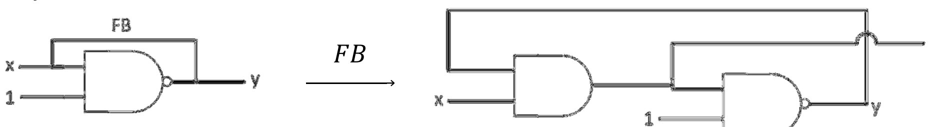
m —/— DUT —/— k → m-k FB bridgings between inputs and outputs

FB $(x, y)_*$



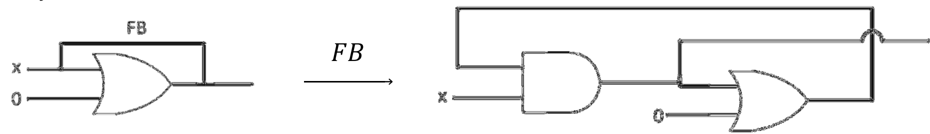
Combinational $\xrightarrow{FB}$ Sequential

<u>Oscillation:</u>
$(x, y)_*$



Difficult to observe

<u>Asynchronous behavior:</u>
$(x, y)_*$



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

Device-Oriented Faults

IV. <u>Arithmetical Faults</u>

Typical for adders, subtractors, counter, multipliers, etc.

$x \xrightarrow{fault} \hat{x}$

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

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

Number of faults:
m number of bits

$$\sum_{i=1}^{l} 2^i \binom{m}{i}$$ - symmetrical

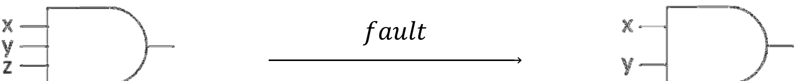
$$\sum_{i=1}^{l} \binom{m}{i}$$ - unidirectional

Symmetric, asymmetric and unidirectional arithmetical faults:
Arithmetical Codes (modulo A checks)

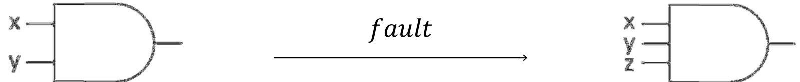
Example: $l = 1 \rightarrow A = 3$ to detect single symmetrical arithmetic errors

V. PLA Faults

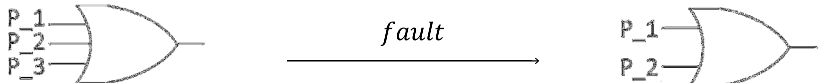
1. A product term can grow (cover more minterms)



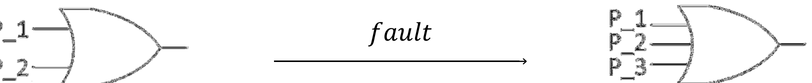
2. A product term can shrink (cover fewer minterms)



3. A product term can disappear from a function



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



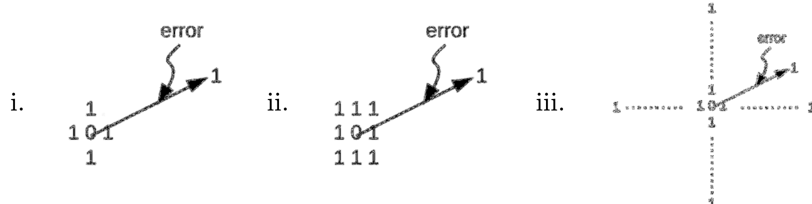
VI. Memory Faults

1. **Decoding errors:**
   open decoder → no addressing
   multiples writes, two address at a time
   replacement of an address by another
2. **Stuck-at** faults in binary cells
3. **Bridgings** between cells
4. **Bursts** on a surface of a disc or tape

For semiconductor memories:

5. **Hold time** for refreshing data → sleeping sickness – graceful degradation of performance
6. **Write recovery** – not producing data at a given access time if each READ is preceded by WRITE
7. **Pattern-sensitive faults** (crosstalkes) – 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 (crosstalkes) between any k cells which may be located anywhere in the memory.

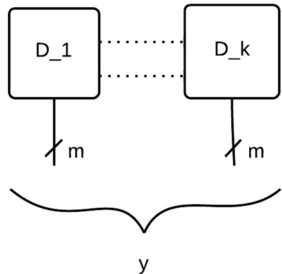
Testing Time (test complexity, number of READ and WRITE operations) is proportional to the size of a neighbourhood

VII. **Functional Faults** (μ-processors)

1. Replacement of one instruction I_i by another I_j — I_i/I_j
2. Replacement of an instruction I_i by no instruction — $I_i/\emptyset$
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)

VIII. **Network Faults** (Faults in Distributed Systems)



y is q-ary k dimentional vector where $q = 2^m, y \in F_q^k$

Node Faults: Multiplicity = numer of nodes which are faulty at the same time $0 \leq l \leq k$

Link Faults: Multiplicty = number of links between nodes which are faulty at the same time