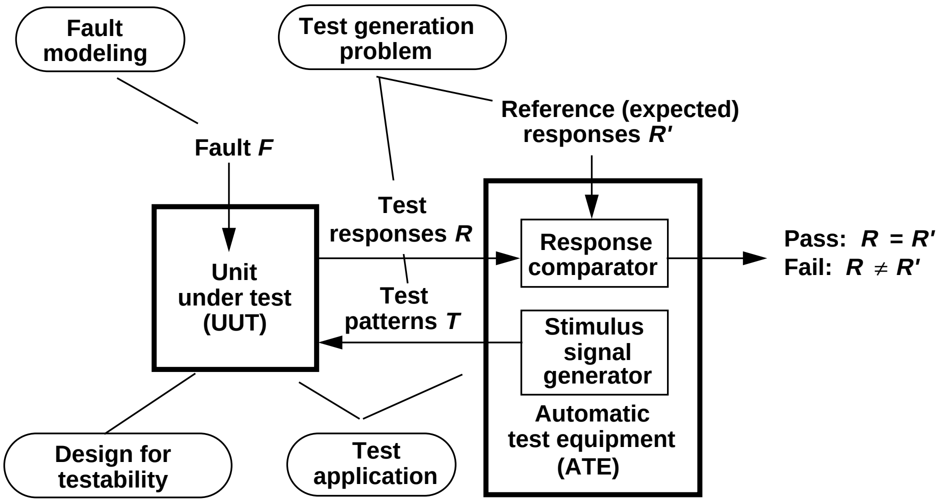
EECS 579 Fall 2001

Recap

- **Text (new):** *Essentials of Electronic Testing* by M. Bushnell & V. Agrawal, Kluwer, Boston, 2000.
- **Class Home Page**: http://www.eecs.umich.edu/courses/eecs579
  - Lecture notes and other materials
  - Homework assignments and solutions

Assignments (tentative)	Grade
Midterm exam: Tue. October 30	20%
Homework assignments (about 6)	20%
Term project	35%
Final Exam: Fri. December 21	25%

What is Testing?



Fault modeling

Test generation problem

Reference (expected) responses *R′*

Fault *F*

Test responses *R*

Unit under test (UUT)

Test patterns *T*

Response comparator

Stimulus signal generator

Automatic test equipment (ATE)

Pass: $R = R'$

Fail: $R \neq R'$

Design for testability

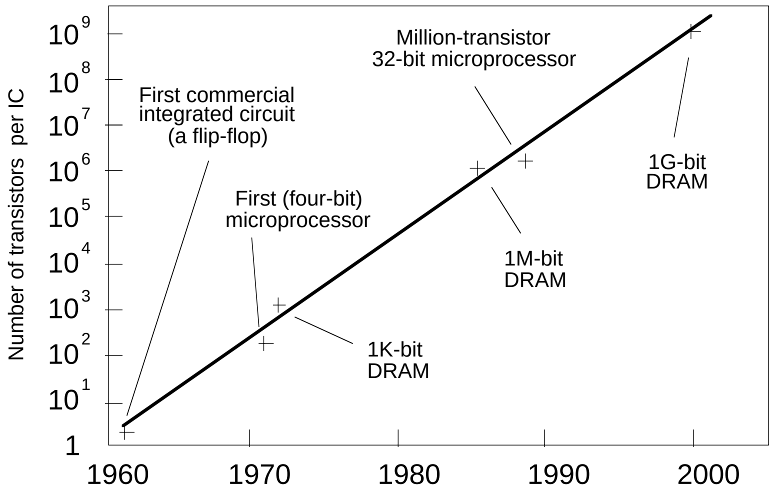
Test application

Why is Testing Important?

(Why do we need a class in testing?)

- Faults cannot be eliminated entirely
- Safety and reliability
    - Its usually not OK to sell faulty products
    - Digital systems are the “brains” of embedded systems
    - In many applications, undetected failures are dangerous
- Testing is inherently a hard problem
    - Good progress has been made, but systems keep getting more complex
- Testing is very expensive
    - ATE for IC production costs millions of dollars
    - Test development affects time to market
    - Adding circuits to improve testability can be costly

Why Testing is Hard



- IC technology is a moving target
- Clock rates and power consumption are soaring too

Why Testing is Hard: SOCs

- SOCs incorporate multiple complex devices and/or technologies on a single IC
    - Processors
    - Memories
    - Communication circuits
    - Application-specific circuits
- In the future:
    - FPGAs
    - MEMS

Testing Costs

- Manufacturing test equipment
    - Capital cost of automatic test equipment (ATE)
    - Operating cost of test facility
- Test software development
    - Automatic test pattern generation (ATPG) code
    - Fault simulation and other debugging code
- Design for testability (DFT)
    - Chip area overhead (implying yield loss)
    - Performance overhead

Testing Costs: ATE

- Example of Cost Estimation

  1.0 GHz 1000-pin production IC tester

  Purchase price: \$1. 0M + 1,000 x \$3,000 = \$4.0M
- Annual operating cost

  Depreciation (4-year) + Maintenance + Operation

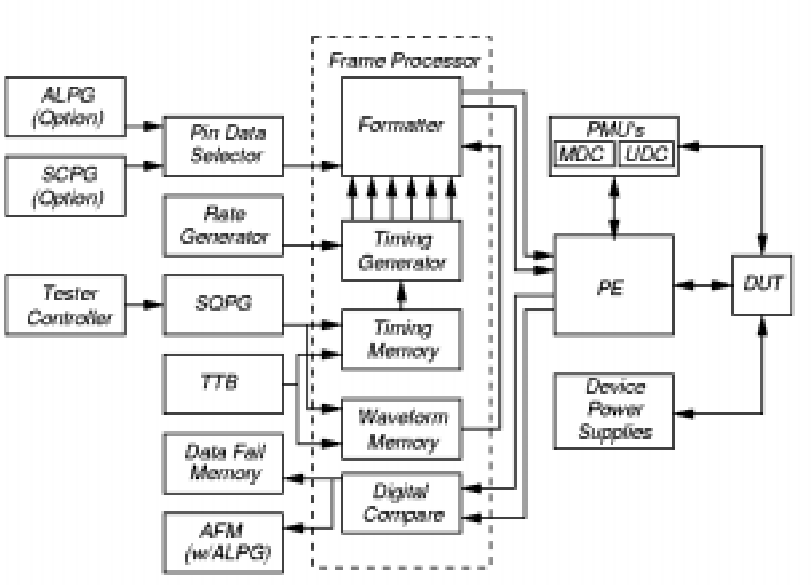
  \$1.0M + \$0.1M + \$0.4M = \$1.5M/year
- Test cost (assuming continuous use)

  \$1.5M/(365 x 24 x 3,600) ≈ 5 cents/sec

Automatic Test Equipment

Advantest T6682

Automatic Test Equipment



Advantest T6682

Testing Costs: DFT

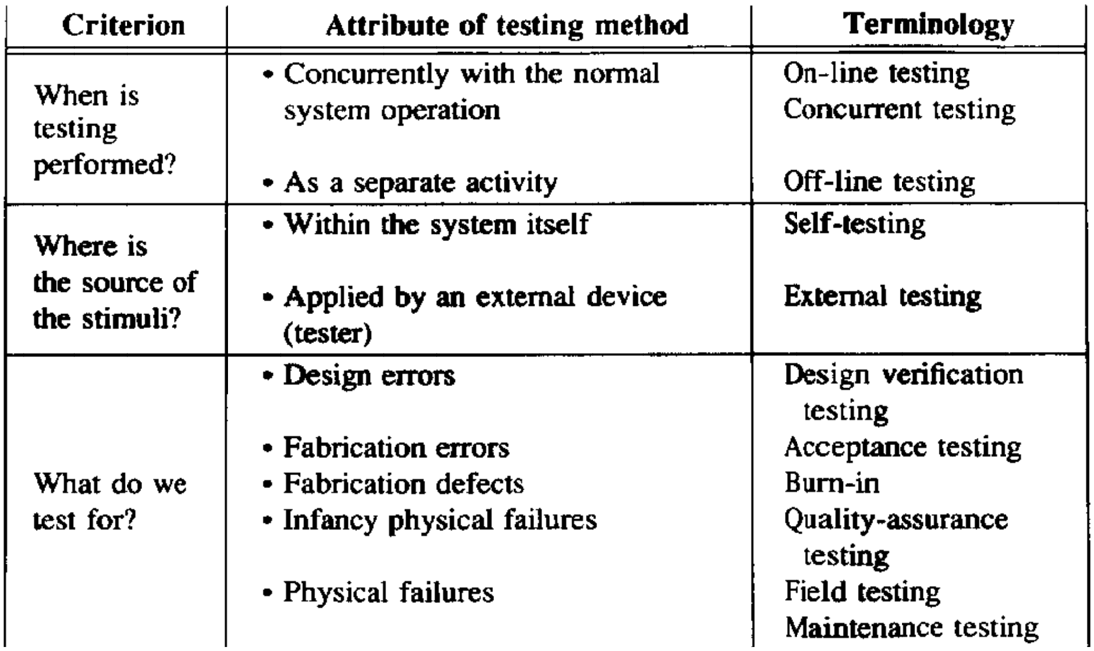
Intel Pentium Microprocessor

- Data from Keynote Address, International Test Conference 1995
- Cost impact of BIST logic that increases area by 1 or 15%

	Nominal Pentium die	1% Die size increase	15% Die size increase
Wafer cost	\$1,460	\$1,460	\$1,460
Die size	160.2mm^2	161.8mm^2	184.2mm^2
Die cost	\$84.06	\$85.33	\$102.55
Added annual cost	—	\$63.5M	\$961M
Dies required/week	1M	1M	1M
Chips fabricated/week	498.1K	482.9K	337.5K

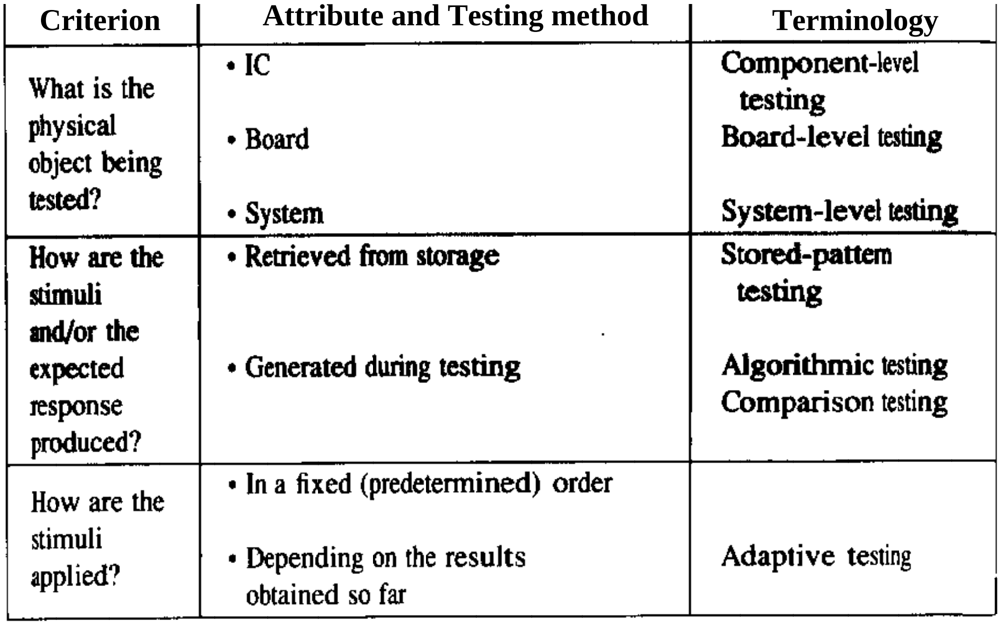
Testing Attributes

Ref: Abramovici et al. [1990 p. 4–5]



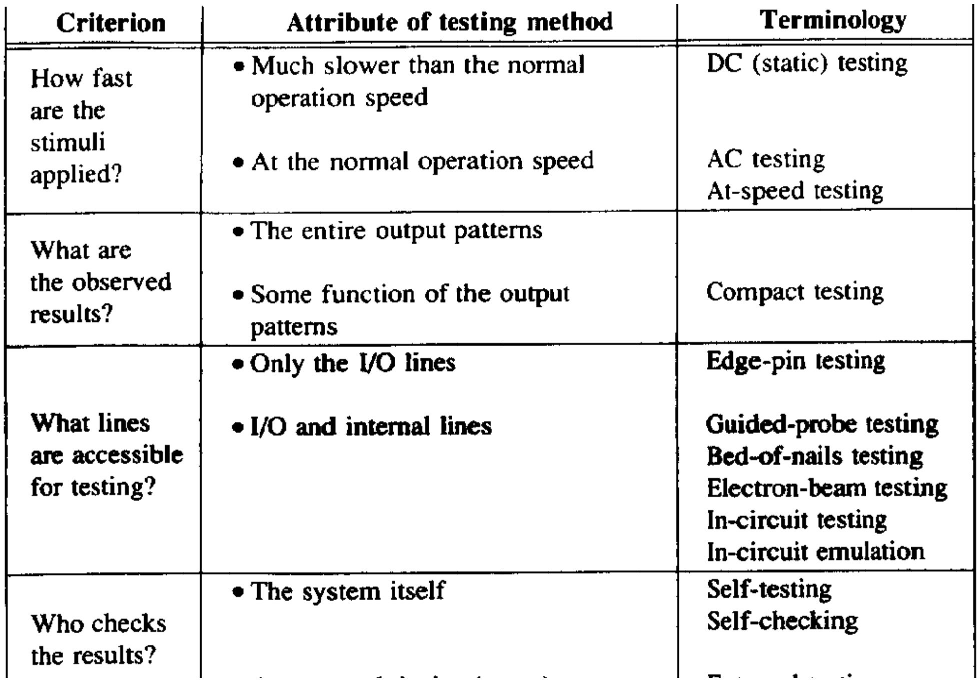
Criterion	Attribute of testing method	Terminology
When is testing performed?	• Concurrently with the normal system operation	On-line testing Concurrent testing
	• As a separate activity	Off-line testing
Where is the source of the stimuli?	• Within the system itself	Self-testing
	• Applied by an external device (tester)	External testing
What do we test for?	• Design errors	Design verification testing
	• Fabrication errors	Acceptance testing
	• Fabrication defects	Burn-in
	• Infancy physical failures	Quality-assurance testing
	• Physical failures	Field testing Maintenance testing

Testing Attributes (cont'd)



Criterion	Attribute and Testing method	Terminology
What is the physical object being tested?	• IC	Component-level testing
	• Board	Board-level testing
	• System	System-level testing
How are the stimuli and/or the expected response produced?	• Retrieved from storage	Stored-pattern testing
	• Generated during testing	Algorithmic testing Comparison testing
How are the stimuli applied?	• In a fixed (predetermined) order	
	• Depending on the results obtained so far	Adaptive testing

Testing Attributes (contd)



Criterion	Attribute of testing method	Terminology
How fast are the stimuli applied?	• Much slower than the normal operation speed • At the normal operation speed	DC (static) testing AC testing At-speed testing
What are the observed results?	• The entire output patterns • Some function of the output patterns	Compact testing
What lines are accessible for testing?	• Only the I/O lines • I/O and internal lines	Edge-pin testing Guided-probe testing Bed-of-nails testing Electron-beam testing In-circuit testing In-circuit emulation
Who checks the results?	• The system itself	Self-testing Self-checking

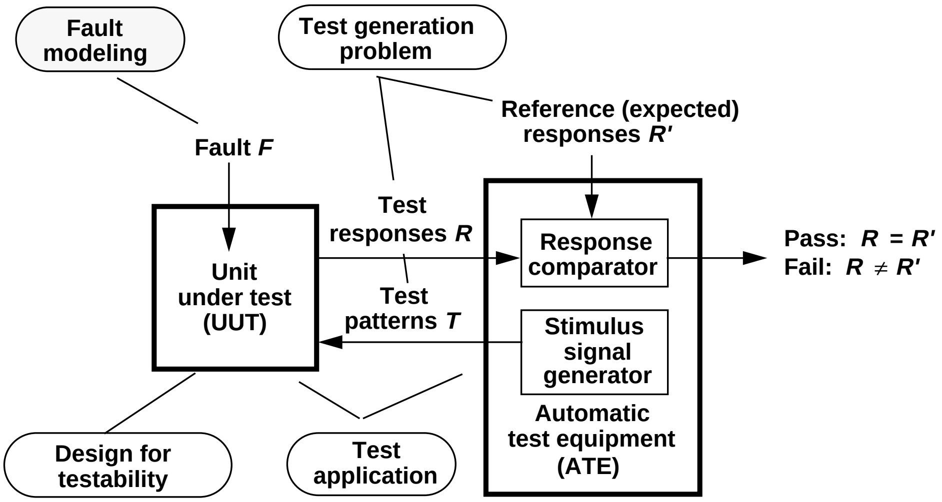
Signal Detection

- Voltage testing
- Current testing: I_{DDQ} testing

Possible Testing Goals

- Complete detection of all modeled faults = high fault coverage
- Fault diagnosis or location to the smallest replaceable component = high fault resolution
- Efficient test generation procedures
- Short test application and response comparison times
- Built-in self-testing or BIST

What is Testing?

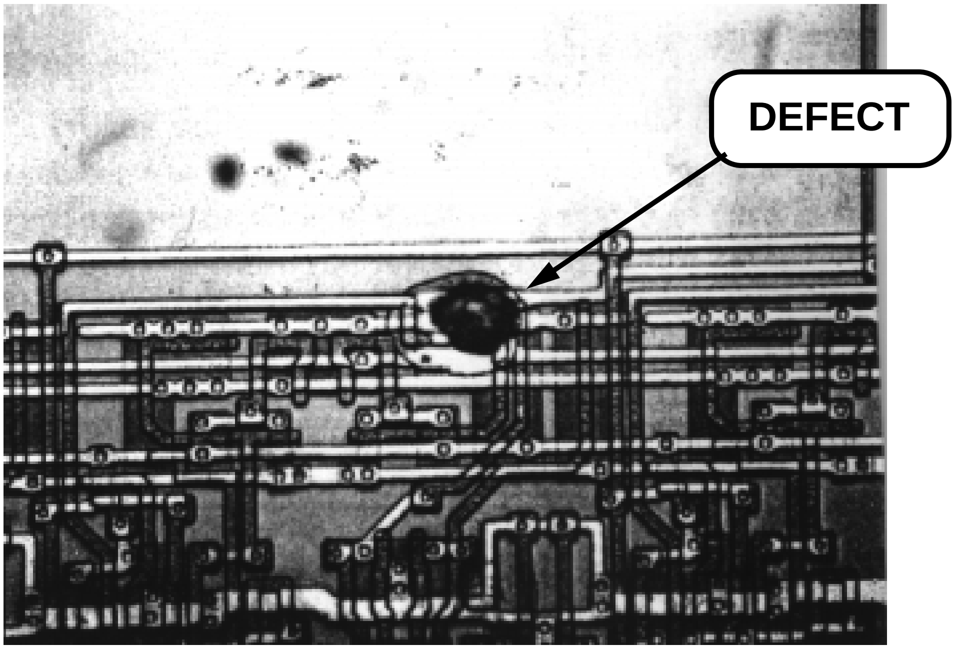


Modeling Levels

Name	*Major components*	*Signals*
Functional (architecture)	Processors, memories, switches, input/output equipment	Data blocks
Register-transfer (RTL)	Registers, combinational circuits, sequential circuits (FSMs)	Words
Gate	Gates, flip-flops	0,1 (bits)
Switch	Transistors as on-off switches	0,1,U,Z
Electric	Transistors, resistors, capacitors,	Analog (V, I,…)

These levels form a hierarchy of “circuits” or “systems”

Physical Faults



Fault Types

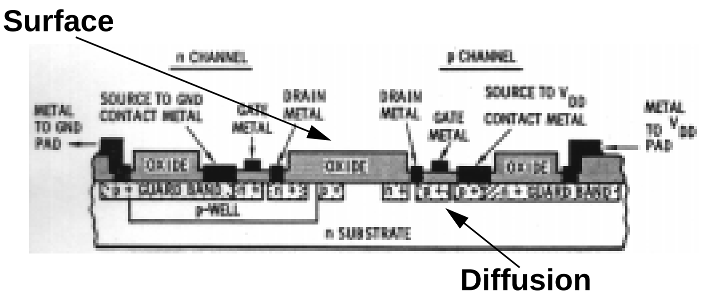
- Permanent: Continuous, stable, irreversible hardware change
- Intermittent: Only occasionally present due to unstable hardware
- Transient: Temporary, caused by environmental conditions

Data:

Sun-2 file server data [Siewiorek and Swartz 1992]

Source of Errors	Number of occurrences	Mean time to occurrence (hrs)
Permanent fault	29	6552
Intermittent fault	610	58
Transient fault	446	354
Failure (system crash)	298	689

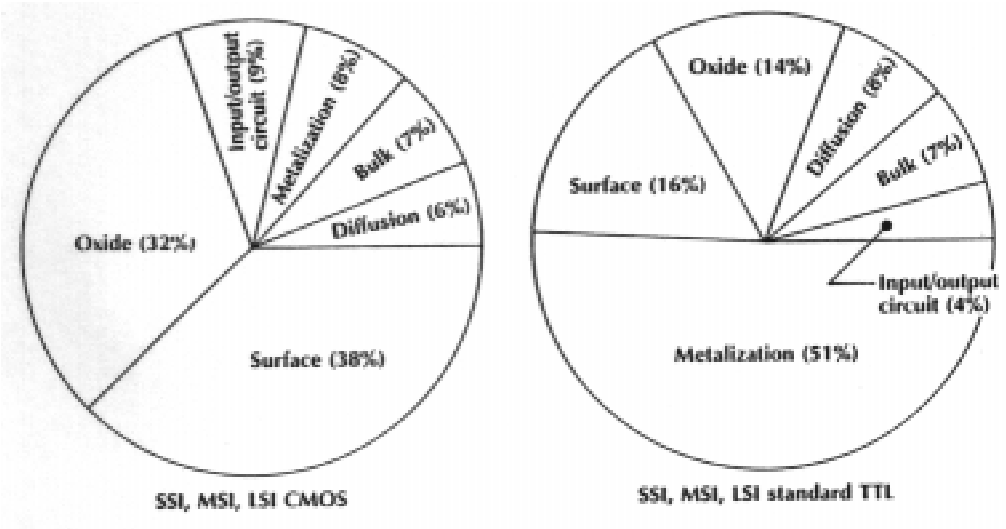
IC Fault Sources



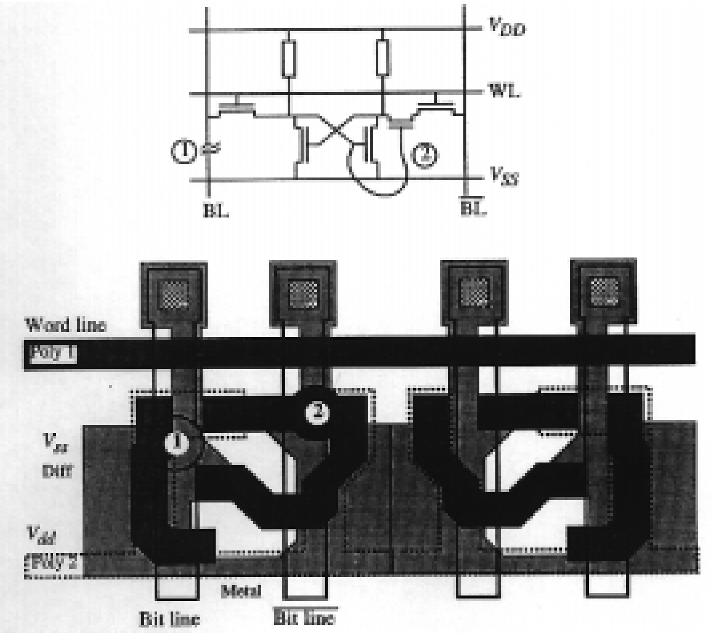
- Surface
- Oxide defects
- Metallization defects
- Die defects
- Bulk (substrate) defects
- Diffusion defects
- Bond defects
- Input/Output circuit defects

<u>Source</u>: Siewiorek and Swartz 1999 and G. R. Case: “Analysis of Actual Fault Mechanisms in CMOS Logic Gates”, *Proc. DAC*, 1976, pp. 265–270.

Defect Distributions



Inductive Fault Analysis



Inductive Fault Analysis

- Model faults as spots of various sizes on layers in layout model—statistical distribution [Shen and Ferguson 1986]
- Abstract (inductively) resulting defects to faults at electrical level and finally logic level

Comparison of fault types using IFA

Fault types	Number of defects	Percentage of faults
Line stuck faults	132	28
Transistor stuck faults	70	15
Open (floating) line faults	101	21
Bridging faults	144	30
Miscellaneous faults	29	6
Total	476	100

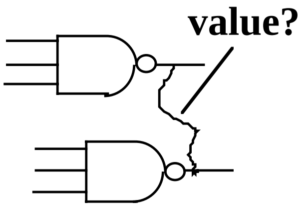
Fault Model Requirements

Requirements

Faults must match the circuit level of abstraction in terms of

Component types
Signal values
Time units

Example: Logic (gate) level



Component types: lines, gates, flip-flops
Signal values: 0, 1
Time units: gate delays

"Meaningless" concepts at this level
Voltage change
Short circuit
Lost message

Common Fault Models

Fault Model	*Definition*
Single stuck-line (SSL)	Any logic line *x* stuck at 0 or 1
Multiple stuck-line (MSL)	Several lines stuck at 0 or 1 simultaneously
Bridging fault	Signals on *x*,*y* become *AND*(*x*,*y*) or *OR*(*x*,*y*)
Delay fault	Delay of signal path changed
Coupling fault	Signals on *x* and *y* become *F*(*x*,*y*)
Stuck-open fault	Signal *x* stuck in some previous state
Pattern interference	Signals interact in space or time

Key Questions

- Does the model adequately represent actual faults?
- Is the model well-behaved?
- Is the model simple enough to use in practice?