VLSI Testing

Tsung-Chu Huang

Department of Electronic Engineering
National Changhua University of Education
Email: tch@cc.ncue.edu.tw

2016/05/31-6/7

Syllabus & Chapter Precedence

Introduction
Modeling
Logic Simulation
Fault Modeling
Fault Simulation
Testing for Single Stuck Faults
Design for Testability
Test Compression
Built-In Self-Test

What's Testing

To tell whether a system is good or bad

Related fields

- **Verification:** To verify the correctness of a design
- **Diagnosis:** To tell the faulty site
- **Fault-tolerance:** To work normally even faults exist

Automatic Test Equipment (ATE)

Why Test?

1. **Why not ship without test?**
2. **Why not final product test only?**

Rule of Tens

3. **Why not functional test only?**

- **Without test at stage k**
- **Cost wasted:** $(1-Y)(P_{k+1}-P_k)$

From Defect to Failure

System Level	Failure	Not work
Function Level	Error	Different state
Logic Level	Fault	Different logic
Physical Level	Defect	Different LRC

Fault Models

1. For Logic Circuits:
   1. Transistor Level:
      - Stuck-Open Fault
      - Bridging Fault
      - Transistor Stuck On/Off Fault
   2. Gate Level:
      - Gate-Output Bridging Fault
      - Delay Fault: {path, segment, gate} delay fault
        Why not cone delay fault?
      - Gate-Output Stuck-At Fault
        - Single Stuck-At (SSA) Fault
        - Multiple Stuck-At (MSA) Fault
2. For Memory:
   - SSA, MSA, Coupling Fault, Transition Fault, …
3. State Machine:
   - State Transition Fault, …

Fault modeling

- **The effects of physical defects**
- **Most commonly used fault model: Single stuck-at fault**

- **Other fault models:**
  - Break faults, Bridging faults, Transistor stuck-open faults, Transistor stuck-on faults, Delay faults

Fault coverage (FC)

$$FC = \frac{\text{\# faults detected}}{\text{\# faults in fault list}}$$

Example:

6 stuck-at faults
(a_0,a_1,b_0,b_1,c_0,c_1)

Test	faults detected	FC
{(0,0)}	c_1	16.67%
{(0,1)}	a_1,c_1	33.33%
{(1,1)}	a_0,b_0,c_0	50.00%
{(0,0),(1,1)}	a_0,b_0,c_0,c_1	66.67%
{(1,0),(0,1),(1,1)}	all	100.00%

Bathtub Curve

Fault Detection of Combinational Circuits

Review Fault Collapsing & Dropping

Illustrating by Fault Dictionary

Fault \ Pattern	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	∞
f_1		1			1			1				
f_2												
f_3			1	1		1				1		
f_4			1	1		1				1		
f_5												?
f_6		1		1	1		1		1		1	
f_7		1		1			1				1	
f_8		1		1			1				1	
f_9	1											

Redundant Fault
← Equivalent Fault
$f_3=f_4$
Abandoned Fault
← Dominant Fault
← Equivalent Fault
$f_6>f_7=f_8$

Review Fault Collapsing & Dropping

Illustrating by Fault Dictionary

↑Fault Collapsing (folding)

Fault \ Pattern	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	?
f_1		1			1			1				
f_2												
f_3			1	1		1				1		
f_4			1	1		1				1		
f_5												?
f_6		1		1	1		1		1		1	
f_7		1		1			1				1	
f_8		1		1			1				1	
f_9	1											

Review Fault Collapsing & Dropping

Illustrating by Fault Dictionary

Fault Dropping↘

Fault \ Pattern	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	?
f_1		1			1			1				
f_2												
f_3			1	1		1				1		
f_5												?
f_6		1		1	1		1		1		1	
f_9	1											

Review Fault Collapsing & Dropping

Illustrating by Fault Dictionary

Fault Dropping↘

Fault \ Pattern	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	?
f_1		1			1			1				
f_3			1	1		1				1		
f_6		1		1	1		1		1		1	

Essential Fault

Fault	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	?
f_9	1											

Fault Grading: $g(f_9)>g(f_1)$ and $g(f_6)>g(f_7)=g(f_8)$ if $f_6>f_7=f_8$

Serial Fault Simulations

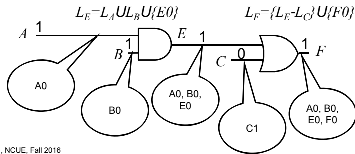
1. Simplest and able to handle of any type of faults.

2. (#Faults + 1) times.

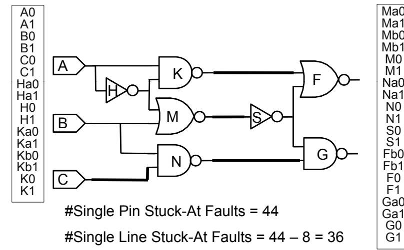
Deductive Fault Simulation

1. Fault List at wire i: the set of all faults that cause the values of I in N and N_f to be different at the current simulated time.
2. Memory for a G-wire circuit with F faults: $G(F+1) \sim O(G^2)$.
3. For a gate Z with controlling value c and inversion i:

4. A simple example of Fault-List Propagation:



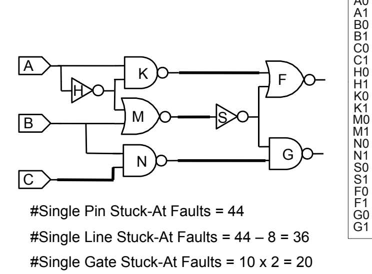
Example of TPG-FS-FD (1) SAF List



#Single Pin Stuck-At Faults = 44

#Single Line Stuck-At Faults = 44 – 8 = 36

Example of TPG-FS-FD (2) SGSAF List

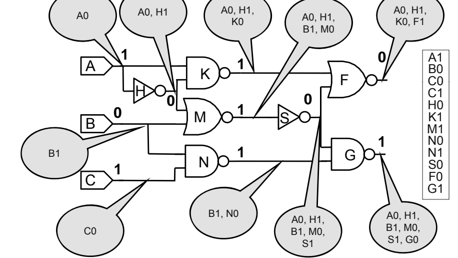


#Single Pin Stuck-At Faults = 44

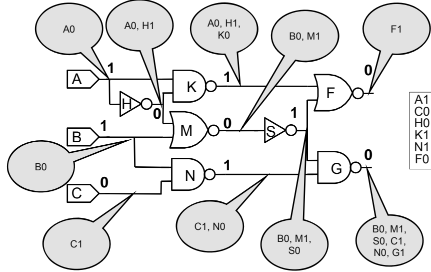
#Single Line Stuck-At Faults = 44 – 8 = 36

#Single Gate Stuck-At Faults = 10 x 2 = 20

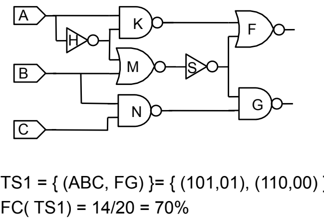
Example of TPG-FS-FD (3) Random Test



Example of TPG-FS-FD (4) RT-FS



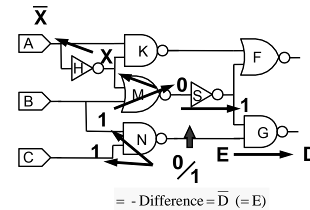
Ex. of TPG-FS-FD (5) Fault Coverage



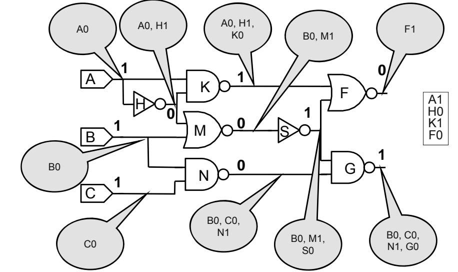
TS1 = { (ABC, FG) }= { (101,01), (110,00) }

FC(TS1) = 14/20 = 70%

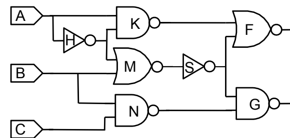
Example of TPG-FS-FD (6) TPG



Example of TPG-FS-FD (7) TG-FS



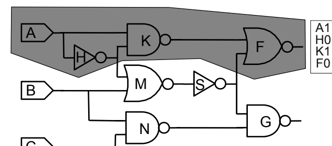
Ex. of TPG-FS-FD (8) Fault Coverage



TS2 = { (ABC, FG) }= { (101,01), (110,00), (111,10) }

FC(TS2) = 16/20 = 80%

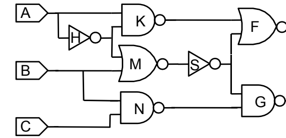
Example of TPG-FS-FD (9) Redundant



Redundant Circuit ➔ Redundant Faults

➔ Undetectable Faults

Ex. of TPG-FS-FD (10) Test Coverage



TS1 = { (ABC, FG) }= { (101,01), (110,00) }

FC(TS1) = 14/20 = 70%

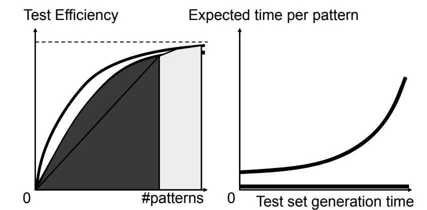
TC(TS1) = 14/16 = 87.5%

TS2 = { (ABC, FG) }= { (101,01), (110,00), (111,10) }

FC(TS2) = 16/20 = 80%

TC(TS2) = 16/16 = 100%

Test Generation: Random vs. Deterministic

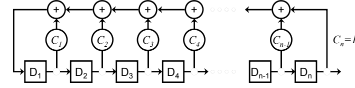


Linear Logic Circuit

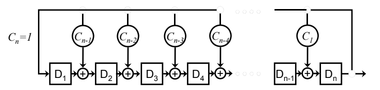
- A linear (logic) circuit preserved the principle of superposition and is constructed from:
  - Delay Flipflops $x \rightarrow$ D $\rightarrow y = Dx$
  - Modulo-2 Adder $x \rightarrow \oplus \rightarrow x+y$ (with input y)
  - Modulo-2 Scalar Multiplier $x \rightarrow \otimes \rightarrow cx$ (with input c)

Linear Feedback Shift Register, **LFSR**

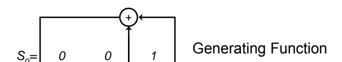
- An LFSR can be mapped into a feedback linear circuit with DFFs & XORs in a modulo-2 domain.
- Canonical Forms:
  - Type I (Direct Form, External Form):



  - Type II (Indirect Form, Internal Form):

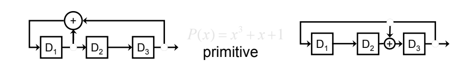


Example



	D_1	D_2	D_3
S_1=	1	0	0
S_2=	0	1	0
S_3=	1	0	1
S_4=	1	1	0
S_5=	1	1	1
S_6=	0	1	1
S_7=	0	0	1
S_0=	1	0	0

Examples of Primitive Polynomials



$x^5 + x^2 + 1$

$x^8 + x^6 + x^5 + x + 1$

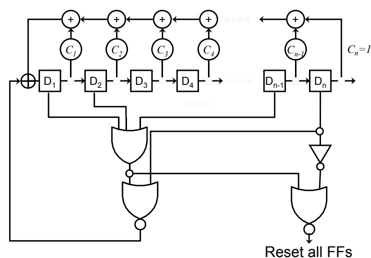
$x^{16} + x^5 + x^3 + x^2 + 1$

$x^{32} + x^{28} + x^{27} + x + 1$

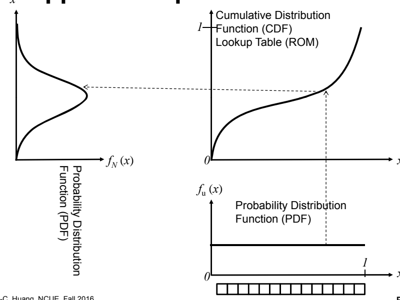
$x^{36} + x^{11} + 1$

Note that $P^*(x)$ must not be primitive even if $P(x)$ primitive.

LFSR with the All-Zero Pattern

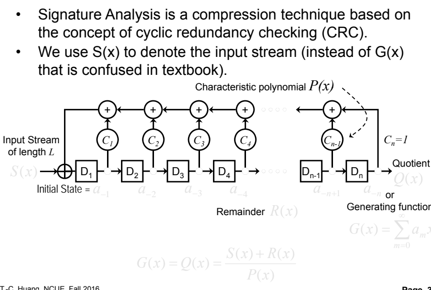


Approx. of Specific Distribution



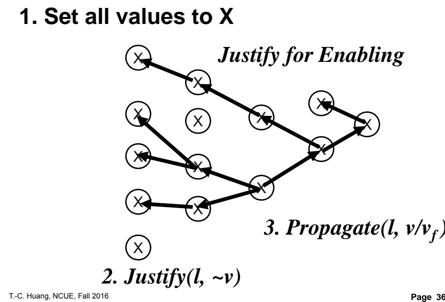
Signature Analyzer (SA)

- Signature Analysis is a compression technique based on the concept of cyclic redundancy checking (CRC).
- We use S(x) to denote the input stream (instead of G(x) that is confused in textbook).

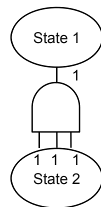


Test Generation for Fanout-Free Tree

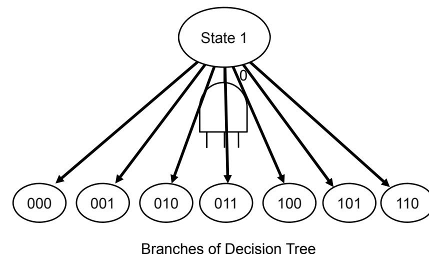
1. Set all values to X



Decision Process in Justification

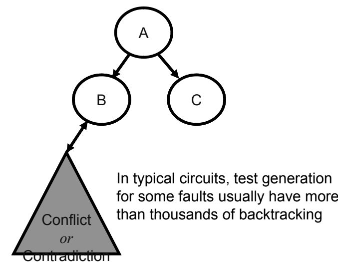


Decision Process in Justification



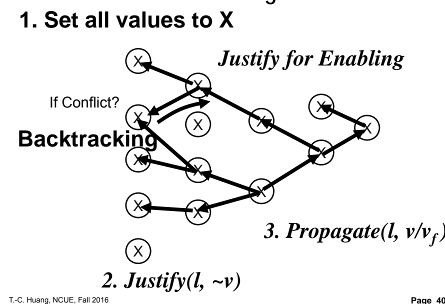
Branches of Decision Tree

Backtracking in Decision Tree



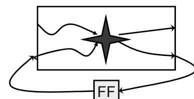
Test Generation for Fanout-Free Tree

1. Set all values to X



Usual Causes to Untestability in SSF Model

1. Combinational Circuits:
   - Reconvergent Fanout and Self-masking
   - Redundant Circuits
2. Sequential Circuits:
   - Uncontrollable at PPI (Pseudo Primary Input)
   - Unobservable at PPO (Pseudo Primary Output)

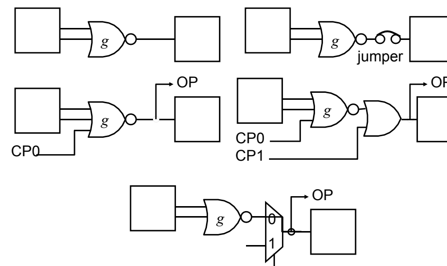


Ad Hoc for Testability Techniques

頭痛醫頭，腳痛醫腳

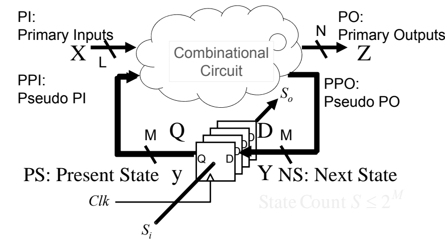
1. Test Points
2. Initialization (Reset)
3. Monostable Multivibrators (1-shots)
4. Oscillators and Clocks
5. Counters/Shift Registers
6. Partitioning Large Circuits
7. Logical Redundancy
8. Breaking Global Feedback Paths

Employing Test Points



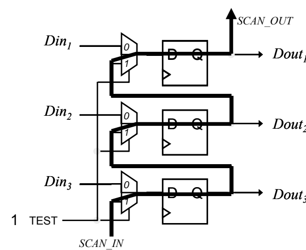
Huffman Model for a Typical Scan Circuit

Single Clock, Synchronous, DFF-based



Basic Scan Cells

MDFF, Multiplexed D-Flipflop



Scan Cell Insertion in HDL

```
module ckt(Clk, input, output);
…
leftmodule  (Clk, D2, …);
DFF  D2(Clk, D1, Q2);
 rightmodules  (Clk, Q2, …);
…
endmodule
```

→

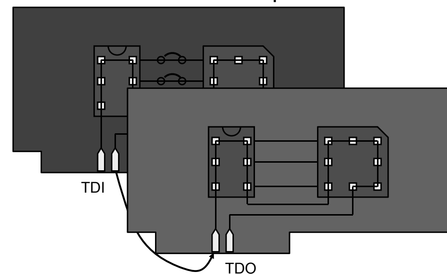
```
module scan_cell (TEST, Clk, Di, Do, Si, So);
…
mux (TEST, Di, Si, Mo);
DFF (Clk, Mo, Do);
assign So=Do;
endmodule

Module scan_ckt(TEST, Clk, input, output, Si, So);
…
assign  Si2=So1;
leftmodule (Clk, D1, …);
scan_cell  SC1(TEST, Clk, D1, Q1, Si2, So2);
 rightmodules (Clk, Q1, …);
…
assign So=Son;
endmodule
```

Test Synthesis

Boundary Scan

Basic Concept



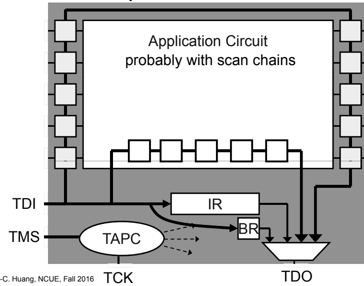
Boundary Scan

Background

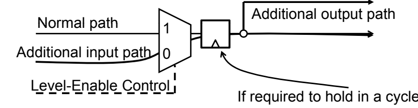
1. Joint Test Action Group (JTAG) Boundary Scan Standard, 1988
2. IEEE P1149.1 Testability Bus Standard (Proposal), 1989
3. Basic Structure:
   - TAP (Test Access Port) Controller
   - Registers: IR (Instruction Register) and BR (Bypass Register)
   - Extra Pins:
     - TMS (Test Mode Singal)
     - TCK (Test Clock)
     - TDI (Test Data Input)
     - TDO (Test Data Output)

Boundary Scan

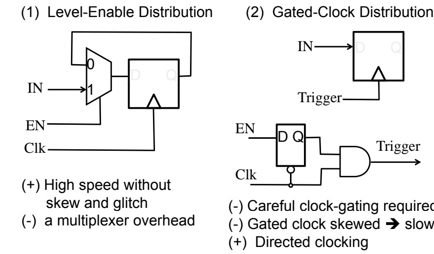
Chip Architecture for BS1149.1



Path-Oriented Scan Cell Design



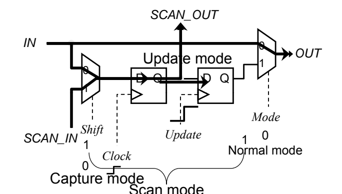
Conditional Activation



(+) High speed without skew and glitch
(-) a multiplexer overhead

(-) Careful clock-gating required
(-) Gated clock skewed ➔ slow
(+) Directed clocking

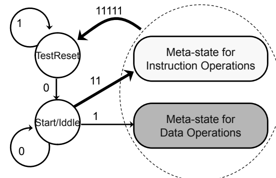
A Basic Boundary Scan Cell



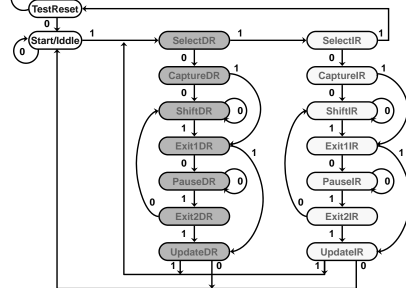
- Can update PIs (capture POs) simultaneously for detecting delay response.
- Can be used both as input and output boundary scan cells.

Basic Test Access Controller

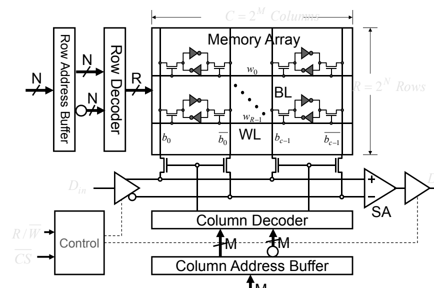
A Synchronous Finite State Machine with 2X8 States



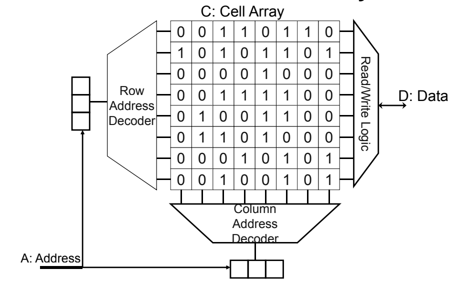
State Diagram of TAPC



Basic SRAM Architecture



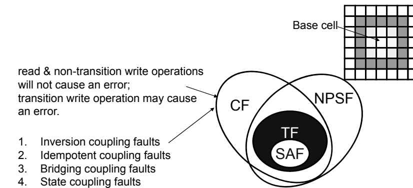
Reduced Functional Memory Model



Reduction of Functional Faults

1. Stuck-At Faults
   - Cell stuck
   - Driver stuck
   - Read/write line stuck
   - Chip-select line stuck
   - Data line stuck
   - Open in data line
2. Transition Faults
   - Cell can be set to 0 but not to 1 or vice versa.
3. Coupling Faults
   - Short between data lines
   - Crosstalk between data lines
4. Neighborhood Pattern Sensitive Faults
   - Pattern sensitive interaction between cells
5. Address-decoder Faults
   - Address line stuck
   - Open in address line
   - Shorts between address lines
   - Open decoder
   - Wrong access
   - Multiple access

Fault Levels and Assumptions



1. Inversion coupling faults
2. Idempotent coupling faults
3. Bridging coupling faults
4. State coupling faults

Brief Introduction to Memory Test

Fault Model:

0	0	1	1	0
1	0	1	0	1
0	1	0	0	0
0	0	1	1	0
0	1	0	0	0

Basics: $\Downarrow w0 \Downarrow r0 \Downarrow w1 \Downarrow r1 \Downarrow (w1r1w0r0)$

Detailed in the *Introduction to IC Test.*

March Test

Suk, 1981

- A march test consists of a finite sequence of march element that is a finite sequence of operations applied to every cell in memory before proceeding to the next cell.
- Notation of March Tests:

$\Uparrow(operations) \equiv$ $for(a=0; a<n; a++)\{$ $operations;$ $\}$

$\Downarrow(operations) \equiv$ $for(a=n-1; a>=0; a--)\{$ $operations;$ $\}$

March Test

Suk, 1981

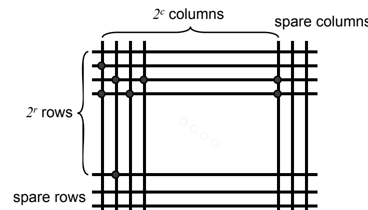
- Operations of March Tests:
  - *w0*: write zero to the cell,
  - *w1*: write one to the cell,
  - *r0*: read and detect whether the result is 0,
  - *r1*: read and detect whether the result is 1.
- Example: the simplest model (non-coupling SAF)
- For the non-coupling SAF, *2n wr*-operations are needed.

Traditional RAM Test

- Zero-One:
  - Not all TF, CF are detected, 4x2a length (a-bit address)
- Checkboard:
  - additionally detects shorts btw adjacent cells.
- GALPAT (Galloping pattern) and Walking 1/0
- Sliding Diagonal
- Butterfly

RAM Self-Repair

- To promote the product yield.



Schmoo Plot

- To show the parametric relations during parameter test.