



VLSI Testing

積體電路測試

Test Compression

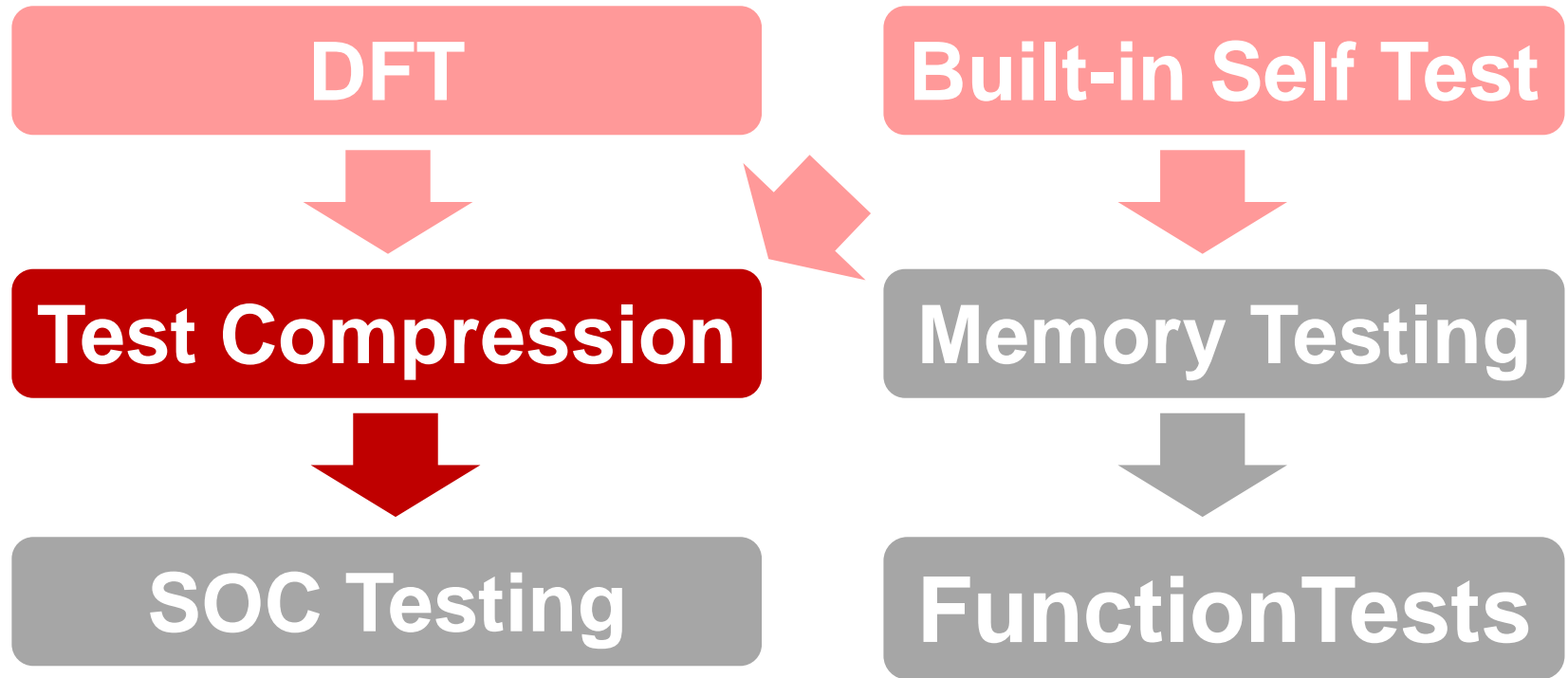
Professor James Chien-Mo Li 李建模

Lab. of Dependable Systems

Graduate Institute of Electronics Engineering

National Taiwan University

Course Roadmap (Design Topics)



Motivating Problem

- You generate 100Gb of test patterns
 - ♦ but ATE has only 80Gb
- Your manager asks you to reduce 25% test patterns
 - ♦ but maintain same fault coverage

	f_1	f_2	f_3	f_4	f_5
t_1		X			
t_2			X		X
t_3		X		X	X
t_4	X			X	

Why Am I Learning This?

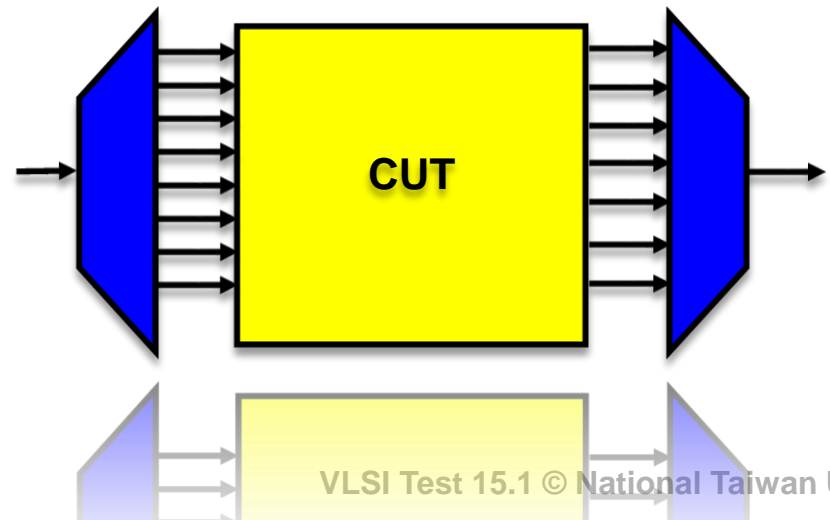
- Test compression reduces test data
 - ◆ Reduces ATE cost, test cost and package cost
 - ◆ It is essential for modern complex designs

“The Simplest Answer Is the Best.”

(Occam’s razor)

Test Compression

- Introduction
- Software Techniques
 - ♦ Dynamic Test Compression
 - ♦ Static Test Compression
- Hardware Techniques
 - ♦ Test Stimulus Compression
 - ♦ Test Response Compaction
- Industry Practices* (not in exam)
- Conclusion



Introduction

- What is test compression?
 - ♦ Reduces test data, keeps same test quality
- Why test compression?
 - ♦ Reduce **test data** (ATE cost ↓)
 - ♦ Reduce **test time** (test cost ↓)
 - ♦ Reduce **DFT pins** (Package/ATE cost ↓)
- Why can we compress test data?
 - ♦ Test stimulus: ATPG patterns have many **don't care** bits
 - ♦ Test responses: **Not every bit** needs to be observed

Compression v.s. compaction

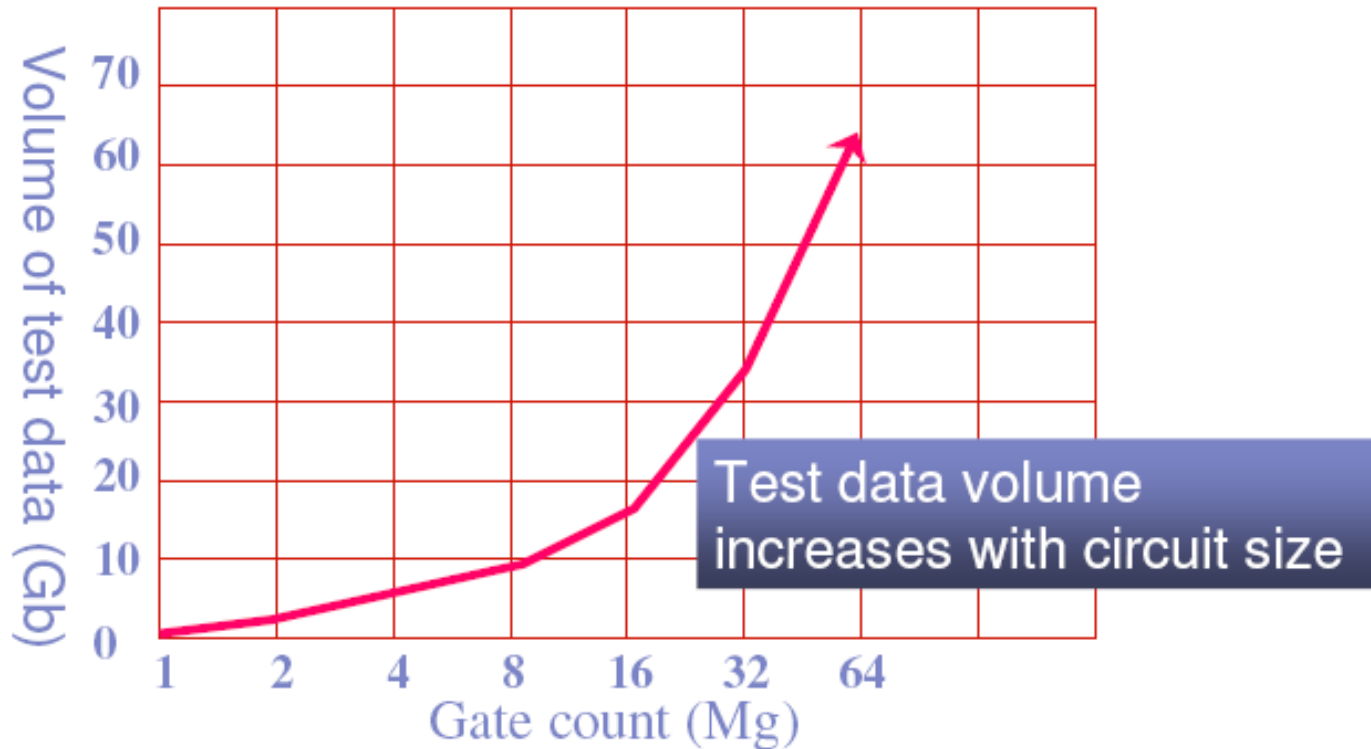
Test stimulus *compression* is **lossless**

Test response *compaction* is **lossy**

Sometimes people use them **interchangeably**

Test Data Volume Skyrockets

- Suppose ATE system has 500 pins, each has 64Mb memory
 - ♦ Total ATE memory available = **32Gb**
- 1M FF x 60K test patterns = **60Gb** > 32Gb



(Source: Blyler, Wireless System Design, 2001)

More & More Compression Needed

- Required compression ratio keep increasing
- More **1,000x** test compression needed by 2020!

$$\text{Compression Ratio} = \frac{\text{Original Data}}{\text{Compressed Data}}$$

Year of Production	2013	2014	2015	2016	2017	2018	2019	2020
Worst Case (Flat) Data Volume (Gb)								
MPU-HP - High Performance MPU (Server)	1458	1984	2699	3673	4998	6138	7537	9256
MPU-CP - Consumer MPU (Laptop/Desktop)	853	1160	1579	2149	2924	3591	4409	5415
SOC-CP - Consumer SOC (Consumer SOC, APU, Mobile Proces	1122	1526	2077	2826	3846	4723	5800	7122
Best-Case Test Data Volume (Hierarchal & Compression) (Gb)								
MPU-HP - High Performance MPU (Server)	4.7	5.1	5.7	6.4	7.2	7.3	7.4	7.5
MPU-CP - Consumer MPU (Laptop/Desktop)	3.7	4.1	4.6	5.1	5.7	5.7	5.8	5.8
SOC-CP - Consumer SOC (Consumer SOC, APU, Mobile Proces	6.9	7.9	8.8	10.2	11.6	12.2	12.6	12.5
Best-Case Compression Ratio (Hierarchal & Compression)								
MPU-HP - High Performance MPU (Server)	312	389	471	572	694	842	1022	1242
MPU-CP - Consumer MPU (Laptop/Desktop)	231	280	342	425	516	625	758	926
SOC-CP - Consumer SOC (Consumer SOC, APU, Mobile Proces	162	192	236	278	330	388	461	568

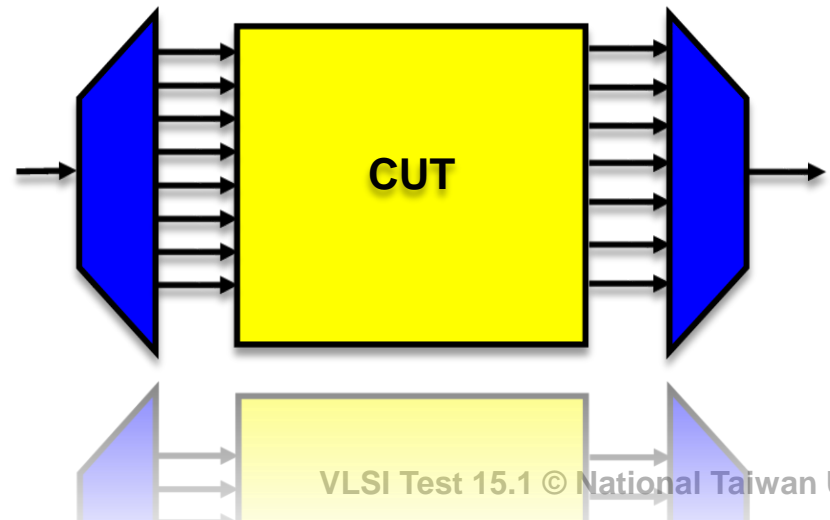
Source: Int'l Technology Roadmap for Semiconductor (ITRS) 2013

Historical Review

- 1980's: **software techniques**
 - ◆ Performed in ATPG, no extra DFT hardware
 - ◆ Reduce number of test patterns
- 1990's: **hardware techniques**
 - ◆ Test stimulus compression
 - ◆ Reduce number of DFT pins
- 2000's: **hardware techniques**
 - ◆ Test response compaction
 - ◆ Tolerate unknown outputs
- after 2010: **hardware techniques**
 - ◆ Test point insertion
 - ◆ Reduce number of test patterns

Test Compression

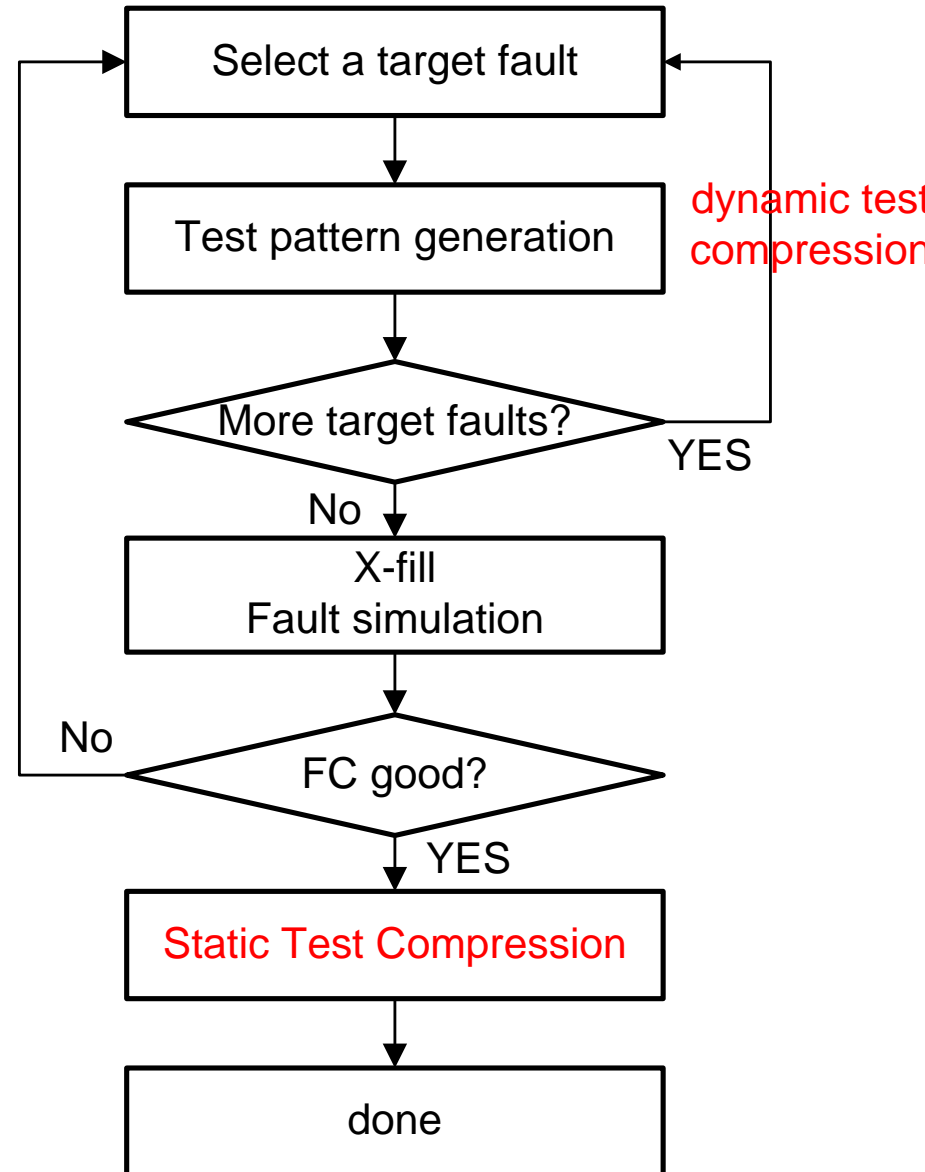
- Introduction
- **Software Techniques**
 - ♦ **Dynamic test compression, DTC**
 - ♦ Static test compression, STC
- Hardware Techniques
- Industry Practice
- Conclusion



STC vs. DTC

- **Dynamic test compression**
 - ♦ performed **during** TPG
 - ♦ more CPU time
 - ♦ more effective
- **Static test compression**
 - ♦ performed **after** TPG
 - ♦ less CPU time
 - ♦ less effective

**TC Increases CPU Time
but Reduces Test Cost**

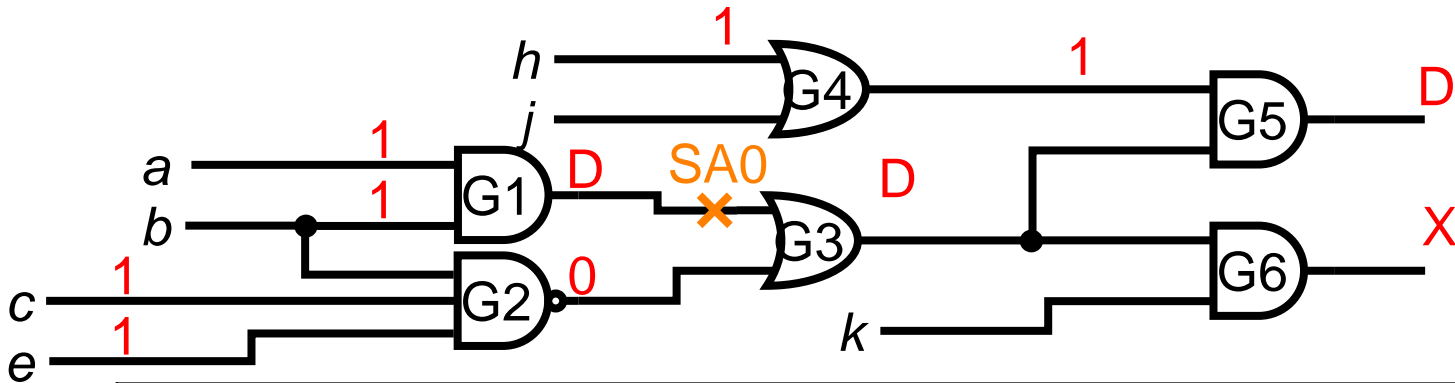


PODEM-X [Goel 81]

- Step1: Select a **primary fault, f_1**
 - ◆ generate test cube, t_1
 - ◆ if fail, remove f_1 from fault list (don't try this fault again)
- Step 2: If f_1 succeed, backtrace from a output =X
 - ◆ select one undetected fault
 - * **secondary fault, f_2**
 - ◆ generate a test cube based on t_1
 - ◆ if fail, try next secondary fault, f_3
 - * Still keep f_2 in fault list (this fault can be tried later)
 - ◆ repeat step 2 until
 - * time out
 - * many continuous failures
 - * no more output =X

PODEM-X Example

- primary fault: **G1 output SA0**



Initial objective: G1 output =1

Backtrace to PI: $b = 1$. simulation, objective not achieved

Backtrace to PI: $a = 1$. simulation, objective achieved

Objective: G2 output = 0 (propagate through G3)

Backtrace to PI: $C = 1$. simulation, objective not achieved

Backtrace to PI: $e = 1$. simulation, objective achieved

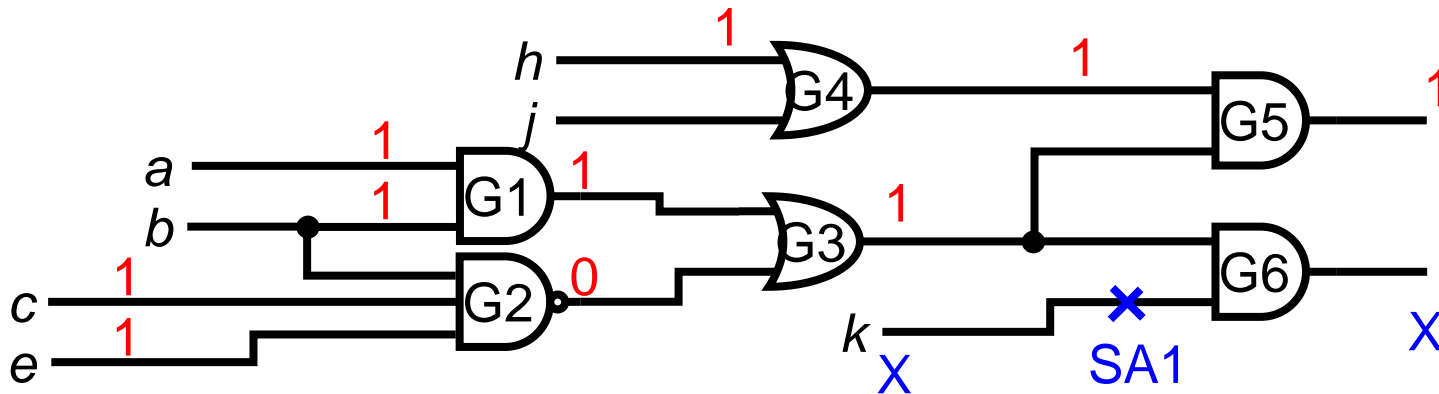
Objective: propagate through G5, G5 output = D

Backtrace to PI: $h = 1$. objective achieved.

Test cube: $abcehjk = 11111xx$

PODEM-X Example (2/3)

- Backtrace from output =X



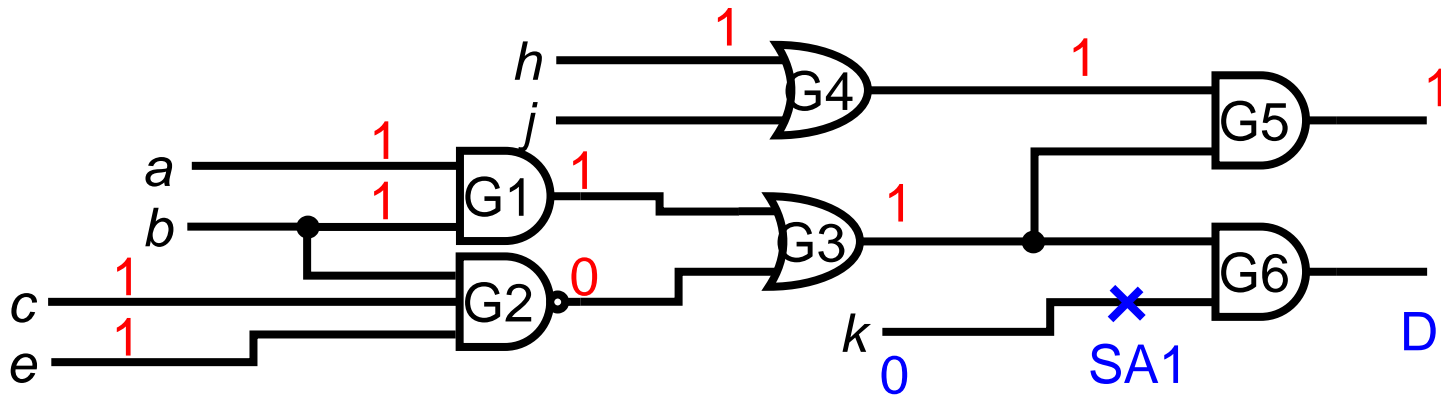
Backtrace from output G_6

Find undetected fault: k SA1

Choose secondary fault = k SA1

PODEM-X Example (3/3)

- secondary fault: ***k* SA1**



Backtrace from output *G*₆

Find undetected fault: *k* SA1

Choose secondary fault = *k* SA1

Objective: *k* = 0

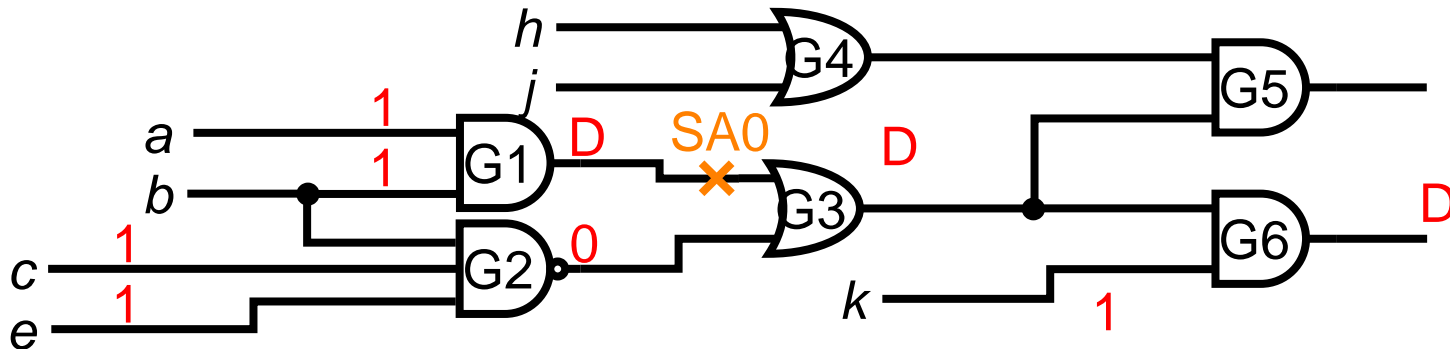
Backtrace to PI: *k* = 0

simulation, output *G*₆ = *D'*, fault detected

new test cube: *abcehjk* = 11111x0

No more secondary fault, DTC ends

Quiz: Redo previous DTC, primary fault: G1 output SA0.
 But this time, propagate to G6 output.
 Please choose secondary fault *h* SA0



Initial objective: G1 output =1

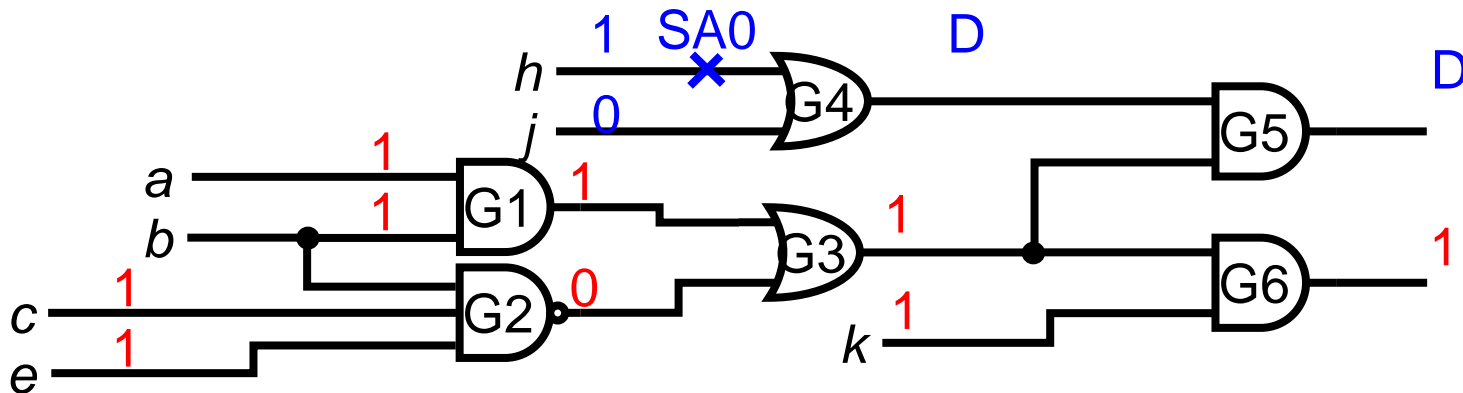
assign *a, b, c, e* = 1

Objective: propagate through **G6**

Backtrace to PI: *k*= 1. objective achieved.

Test cube: *abcehjk* = 1111xx1

Quiz Solution



(cont'd) Test cube: $abcehjk = 1111xx1$

choose secondary fault h SA0

assign $h = 1$, simulate, D generated

choose to propagate through $G4$, objective: $j=0$

assign $j=0$, simulate, D reaches PO

Test cube: $abcehjk = 1111101$

Summary

- Introduction:
 - ◆ Reduce **test data**
 - ◆ Reduce **test time**
 - ◆ Reduce **ATE cost/ DFT pins**
- Software Techniques
 - ◆ Dynamic test compression: **during** TPG, slow but more effective
 - * **PODEM-X**
 - ◆ Static test compression: **after** TPG, fast but less effective

