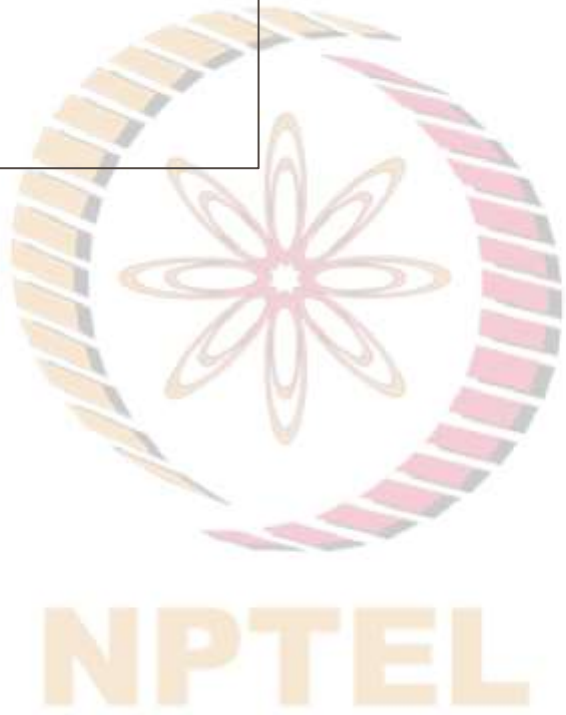
VLSI DESIGN FLOW: RTL TO GDS

Lecture 33
Automatic Test Pattern Generation (ATPG)

Sneh Saurabh
Electronics and Communications Engineering
IIIT Delhi

Lecture Plan

- Terminologies
- General approach to ATPG
- Redundant Faults



Talking to a kitten …

"… they always purr. 'If they would only purr for "yes" and mew for "no," or any rule of that sort' she had said, 'so that one could keep up a conversation!

But how can you talk with a person if they always say the same thing?'

On this occasion the kitten only purred: and it was impossible to guess whether it meant 'yes' or 'no'."

—Lewis Carroll, *Through the Looking-Glass*, Chapter 12, 1871

Source: https://commons.wikimedia.org/wiki/File:LewisCarrollSelfPhoto.jpg Lewis Carroll, Public domain, via Wikimedia Commons

ATPG: Objective, Challenges and Practical Solution

ATPG: Automatic Test Pattern Generator

- **Objective:** Generate test patterns (test vectors) that can be used to detect faults
- **Desired:** small set of test vectors that detect all faults considered for that circuit

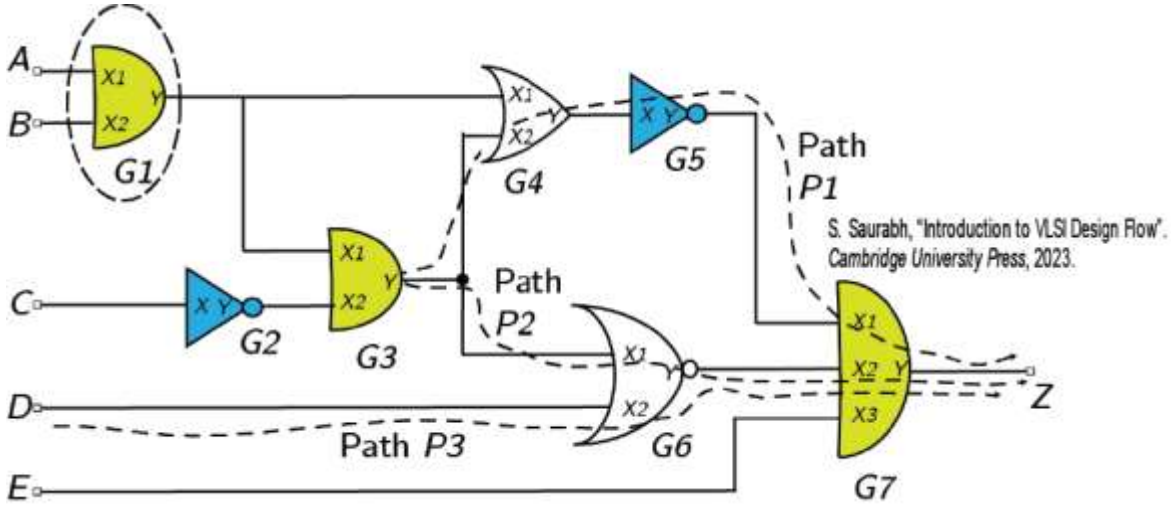
Challenges

- For generalized sequential circuits, generating test vectors/sequences is very difficult
  - Reaching certain states may take exponential clock cycles
- Finding the test sequence that controls or observes an internal point in the circuit is runtime intensive

Practical Solution

- Sequential ATPG problem is transformed to Combinational ATPG problem by Scan Design Flow
- Combinational ATPG problem, though NP-complete, has efficient algorithms

ATPG: Terminologies



- **Path:** sequence of pins in topological order
- **On-path input** or **on-input:** for a given path, the input pins lying on that path
- **Side path inputs** or **side-inputs:** The input pins other than the on-inputs of the instances lying on that path

ATPG: Controlling/Non-controlling values

Controlling value of a multi-input gate

- Value that can be assigned to any input of the gate such that the output is known irrespective of values on other inputs.

Non-controlling value of a multi-input gate

- Value that can be assigned to any input of the gate such that the output is decided by the value on other inputs

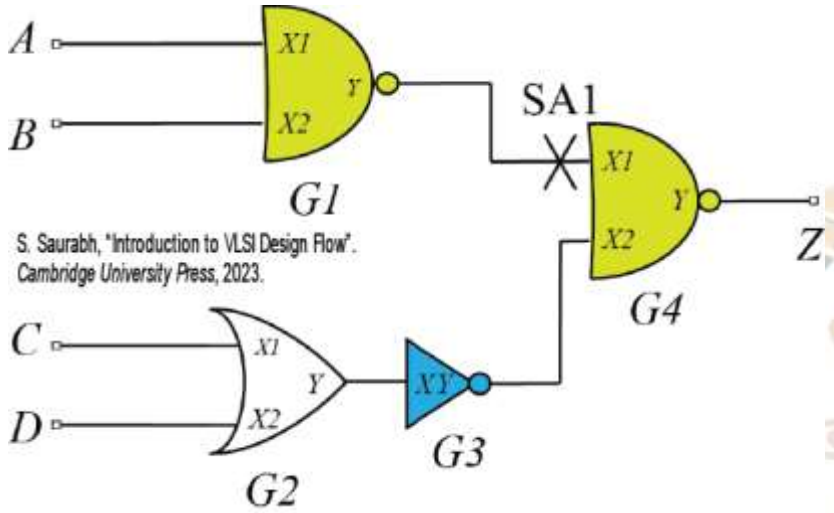
Type of Gate	Controlling Value	Non-controlling value
AND	0	1
NAND	0	1
OR	1	0
NOR	1	0
XOR	Not defined	Any value

ATPG: General Approach

Path Sensitization Method

- Fault Sensitization
- Fault Propagation
- Line Justification

ATPG: Illustration of Path Sensitization Method



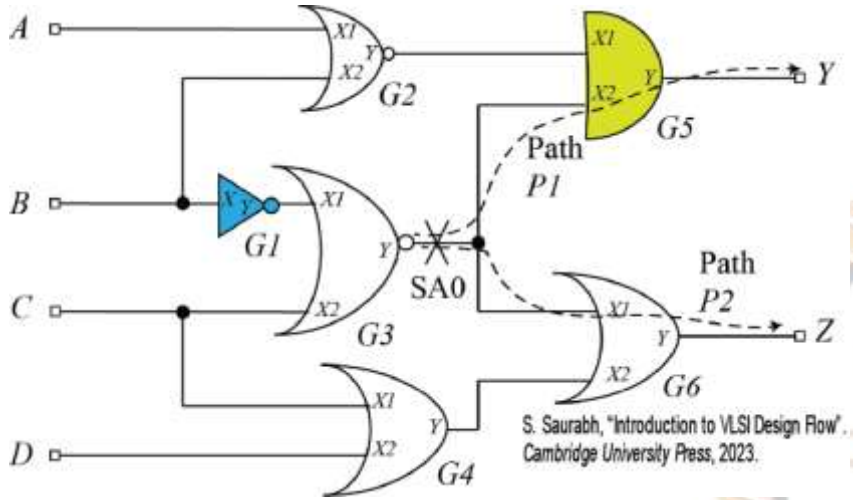
Given: Stuck-at 1 at *G4/X1*

To Find: A test vector (value of A, B, C, D) which will be able to detect the existence of SA1 at *G4/X1*

Test Vector: (1,1,0,0)

- **Fault Sensitization**: Find the value at input ports that will set the value at the fault site **opposite** of the fault value
  - Find A, B, C, D such that *G4/X1*=0 (A=1, B=1)
- **Fault Propagation**: For the path from the fault site to the output port, set the value of side-inputs to non-controlling inputs
  - Set *G4/X2* to non-controlling value (*G4/X2*=1)
- **Line Justification**: Find the value at input ports that will set the side-inputs as found in fault propagation
  - Find A, B, C, D such that *G4/X2*=1 (C=0 D=0)

ATPG: Illustration of Backtracking



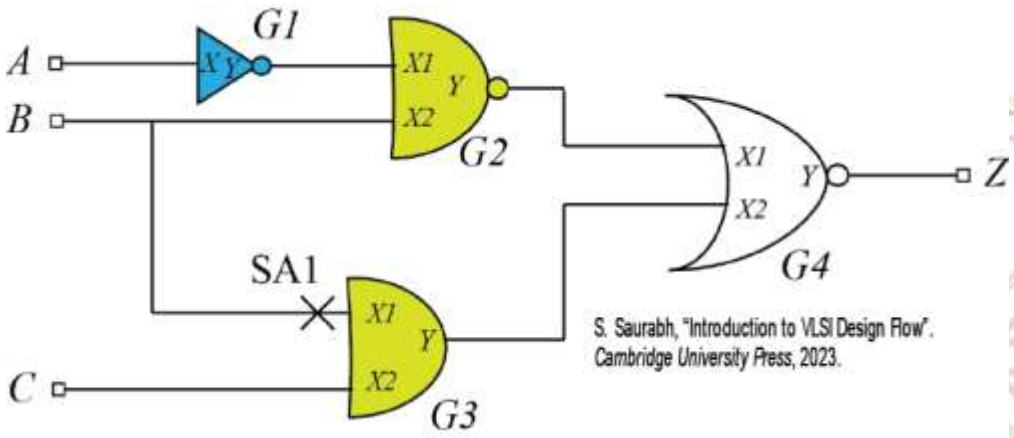
Given: Stuck-at 0 at *G3/Y*

To Find: A test vector to detect this fault.

- Fault Sensitization
  - Find *A, B, C, D* such that *G3/Y=1* (*B*=1, *C*=0)
- Fault Propagation: **Path to Y**
  - Set *G5/X1* to non-controlling value (i.e. 1)
- Line Justification: *A*=0, *B*=0
  - B=0 is in **conflict!**
  - Must be Backtracked
- Fault Propagation: **Path to Z**
  - Set *G6/X2* to non-controlling value (i.e. 0)
- Line Justification: *C*=0 *D*=0

Test Vector: (x,1,0,0)

ATPG: Illustration of Redundant Fault



- Fault Sensitization
  - Find A, B, C such that *G3/X1*=0 (*B*=0)
- Fault Propagation:
  - For Propagation to Z: Set *G3.X2*=1 and *G4/X1*=0
- Line Justification:
  - *B*=1 is in conflict with earlier *B*=0

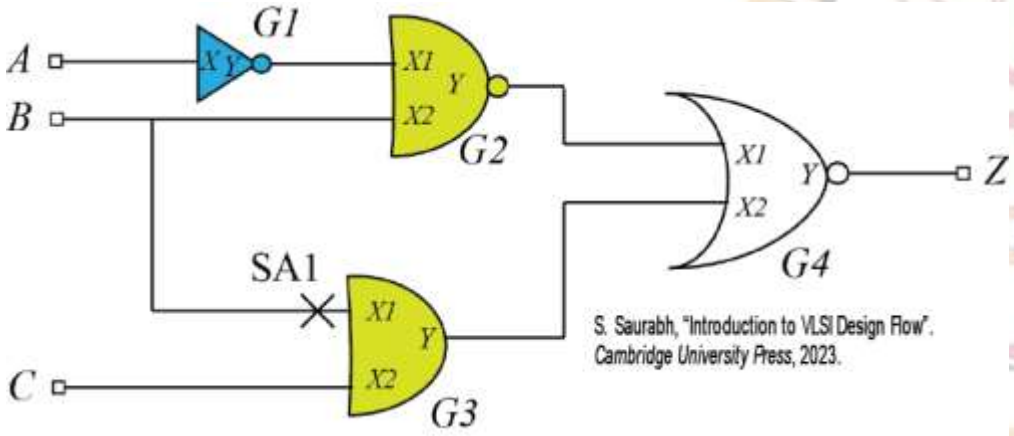
Given: Stuck-at 1 at pin *G3/X1*

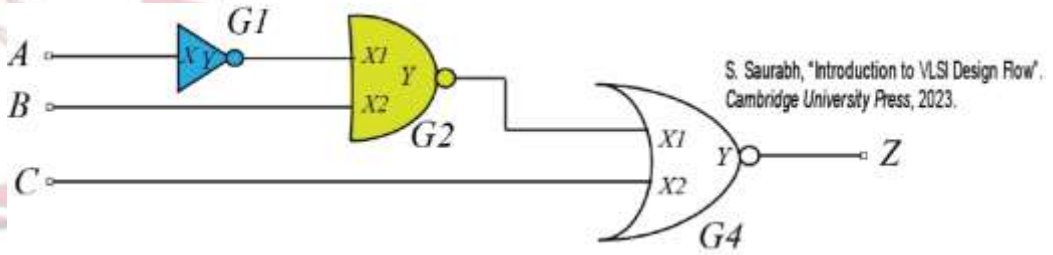
To Find: A test vector to detect this fault

- No test vector exists to detect this fault
- This is known as ***redundant fault***

ATPG: Redundant Fault based Optimization

- Redundant faults can be used to reduce the hardware
- Redundant fault means that the behavior of the circuit with/without fault is the same
- Assume that the fault site has constant 0/1 optimize out gates that are not required





- Assume that SA1 at *G3/X1* is always present.
- *G3/X1*=1 implies that *G3/Y=C*

ATPG: Challenges and Solutions

Complications

- Primarily due to re-convergent fanouts
- Decisions required to be taken at a particular point
- Decision leads to implications
- Some decisions can be wrong (leading to conflicts in decisions)
  - ➢ Backtracking required

Backtracking Limit

- Too many backtrack => possible redundant fault
- User defined Backtrack Limit employed by tool
  - ➢ Example Backtrack Limit = 1000
  - ➢ Abort finding test vector when backtrack limit is hit, i.e. when 1000 times backtracking has been done

Algorithmic Advancements

- D-algorithm: Roth 1966
- PODEM (Path-oriented Decision Making): Prabhu Goel 1981
- FAN (Fanout-oriented) Algorithm: Fujiwara

References

- M. Bushnell and V. Agrawal, “Essentials of electronic testing for digital, memory and mixed-signal VLSI circuits”, *Springer Science & Business Media*, 2004.
- S. Saurabh, “Introduction to VLSI Design Flow”. Cambridge: *Cambridge University Press*, 2023.