# Sequential Automatic Test Pattern Generation (ATPG)



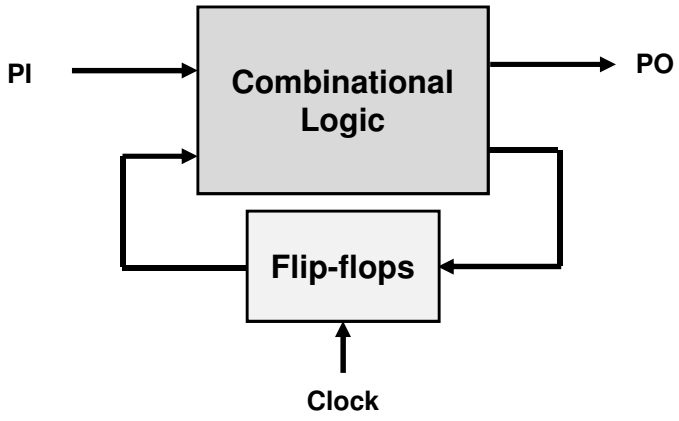
## Introduction

- Almost all practical digital systems are sequential circuits.
- Their testing is more complex than that of combinational circuits, due to two reasons:
  1. Internal memory states
     - State not known at the beginning of test.
     - The test must initialize the circuit to a known state.
  2. Long test sequences

2. Long test sequences
   - A test for a fault in a sequential circuit essentially consists of three parts:
     - Initialization of the internal memory.
     - Combinational test to activate the fault, and bring its effect to the boundary of the combinational logic.
     - If the fault is in the memory elements, observation of the faulty state in one of the primary outputs.
   - Thus the test for a fault may be a sequence of several vectors that must be applied in the specified order.

3

## Model of a Synchronous Sequential Circuit




4

## Assumptions

- We consider synchronous sequential circuits.
  - All memory elements are under the control of a clock signal.
  - Vectors at the PIs are synchronized with the clock.
    - A new vector is applied just after the active edge of the clock.
    - Outputs reach their steady-state values just before the next active edge of the clock.
  - Signal propagation time through the combinational logic does not exceed the clock period.

5

## A Simplified Model

- The combinational part is modeled at the gate level.
  - All single stuck-at faults are considered in it.
- Flip-flops are treated as ideal memory elements.
  - Clock signal is not explicitly represented, and no faults in the clock signals are modeled.
  - Internal faults in flip-flops are not modeled.
  - Input/output faults on flip-flops are modeled as faults on output and input signals of the combinational logic.

6

## Test Generation Methods

- Can be classified into two categories:
  1. Time frame expansion
     - A model of the circuit is created such that tests can be generated by a combinational ATPG tool.
     - Efficient for circuits described at the gate level.
     - Efficiency degrades significantly with cyclic structure, multiple-clocks, or asynchronous logic.

7

  2. Simulation-based methods
     - A fault simulator and a test vector generator are used to derive tests.
     - Circuits modeled at other levels (RTL, transistor, etc.) can be treated.

8

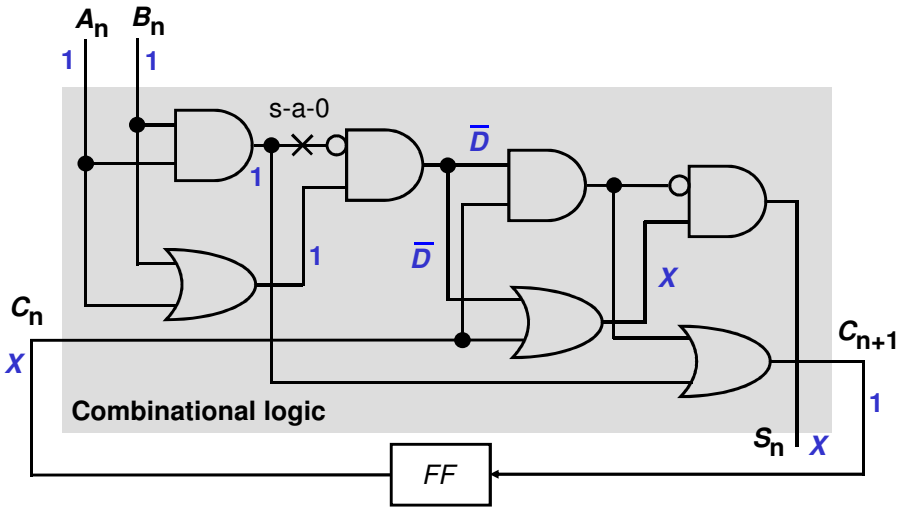
## Time Frame Expansion Method



## Point to Note

- Two basic differences between combinational and sequential circuits.
  1. A test for a fault in a sequential circuit may consist of several vectors.
     - A combinational ATPG is capable of generating only a single vector for a target fault.
  2. Presence of uninitialized states of the sequential circuit.
     - A combinational ATPG can deal with unknown (X) signal states.
     - 5-valued logic, usually effective for combinational circuits, is insufficient for sequential circuits.

## Example: A Serial Adder

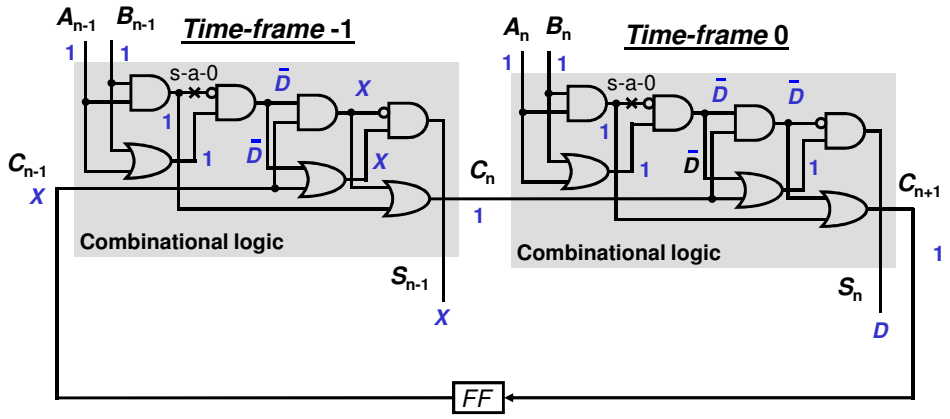



11

- How the adder works?
  - Initialization: say, by applying 00 input.
  - Serial addition: applied bit by bit.
- To apply combinational ATPG procedures:
  - We can "unroll" the sequential circuit into a larger combinational circuit.
  - Called time frame expansion.
- For the adder example:
  - The fault cannot be propagated to $S_n$.
  - We repeat the combinational logic twice to generate a 2-vector test.

12

# Time-Frame Expansion

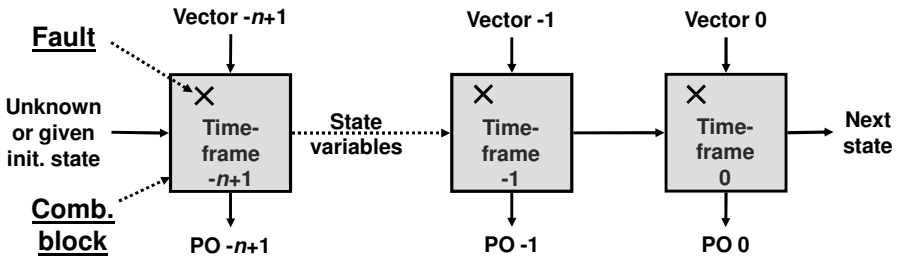





- The leftmost block has its state input in the ‘X’ state.
- Since the fault is present in all frames, it is modeled as a multiple fault.
- It is possible to propagate a ‘D’ to the output.
  - All four input bits justified to be 1’s.
  - Thus the test is an initialization vector 11 followed by another 11 vector that produces a ‘D’ at the output.

## Concept of Time-Frames

- If the test sequence for a single stuck-at fault contains *n* vectors,
  - Replicate combinational logic block *n* times.
  - Place fault in each block.
  - Generate a test for the multiple stuck-at fault using combinational ATPG with 9-valued logic.




15

## Necessity of Nine-Valued Logic

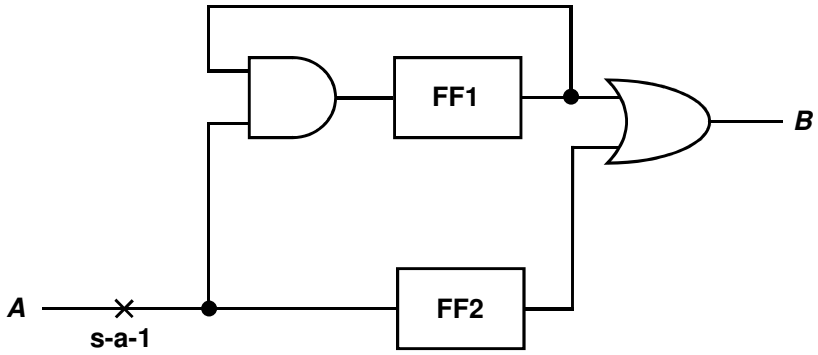
- Five valued system

  0, 1, D, D′, X
- Nine valued system

  0, 1, 1/0, 0/1, 1/X, 0/X, X/0, X/1, X
- We show an example to illustrate the advantage of the nine-valued system.
  - A s-a-1 fault cannot be propagated using 5-valued logic.
  - Can be propagated using 9-valued logic.

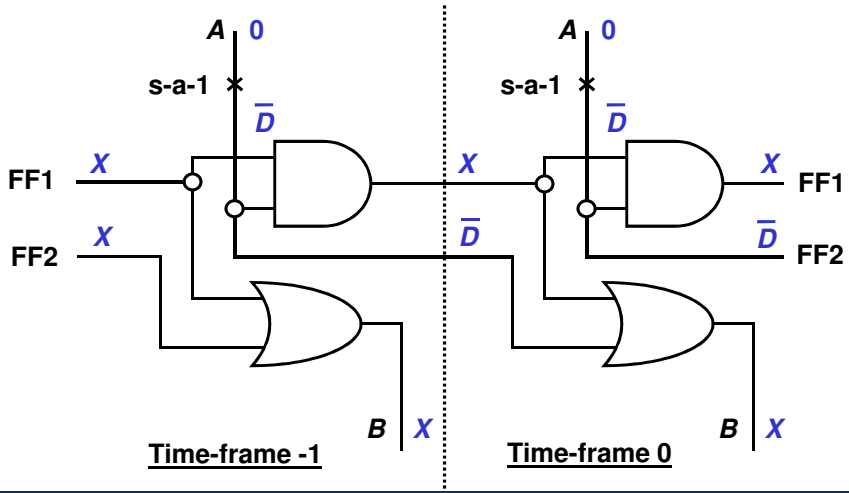
16

## Example for Logic Systems



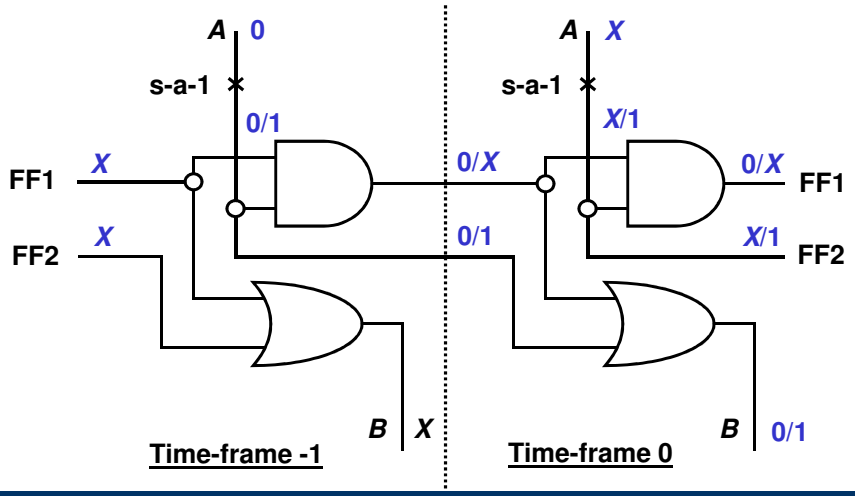

17

## Five-Valued Logic (Roth)
## 0,1, D, D′, X




18

## Nine-Valued Logic (Muth)
## 0,1, 1/0, 0/1, 1/X, 0/X, X/0, X/1, X



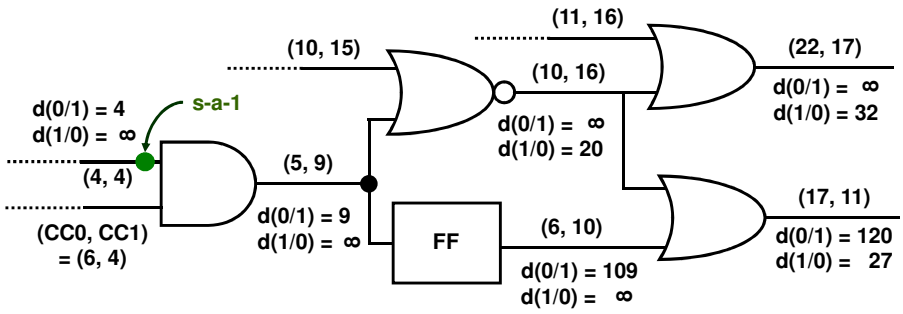

19

## Implementation of ATPG

- Select a PO for fault detection based on drivability analysis.
- Place a logic value, 1/0 or 0/1, depending on fault type and number of inversions.
- Justify the output value from PIs, considering all necessary paths and adding backward time-frames.
- If justification is impossible, then use drivability to select another PO and repeat justification.
- If the procedure fails for all reachable POs, then the fault is ***untestable***.
- If 1/0 or 0/1 cannot be justified at any PO, but 1/X or 0/X can be justified, then the fault is ***potentially detectable***.

20

# Drivability Example



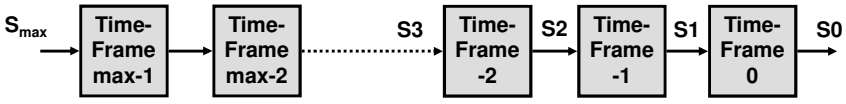

CC0 and CC1 are SCOAP combinational controllabilities

d(0/1) and d(1/0) of a line are effort measures for driving a specific fault effect to that line



# Complexity of ATPG

- Synchronous circuit -- All flip-flops controlled by clocks; PI and PO synchronized with clock:
  - Cycle-free circuit – No feedback among flip-flops: Test generation for a fault needs no more than $d_{seq}$ + 1 time-frames, where $d_{seq}$ is the sequential depth.
  - Cyclic circuit – Contains feedback among flip-flops: May need $9^{Nff}$ time-frames, where *Nff* is the number of flip-flops.
- Asynchronous circuit – Higher complexity!



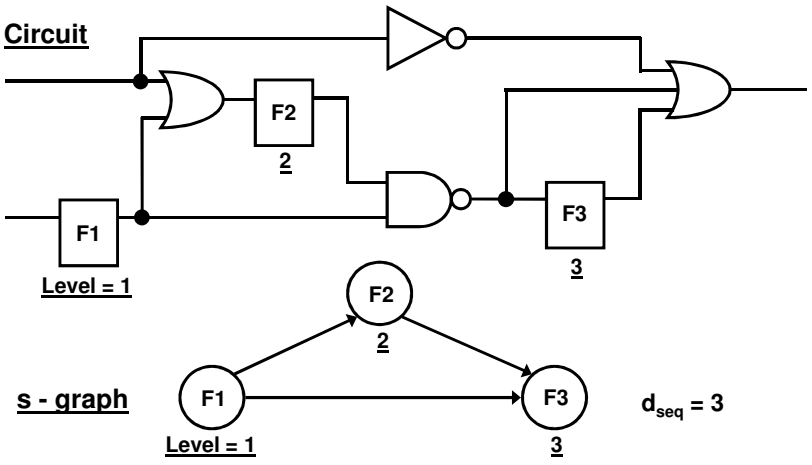

*max* = Number of distinct vectors with 9-valued elements = $9^{Nff}$

## Cycle-Free Circuits

- Characterized by absence of cycles among flip-flops and a sequential depth, $d_{seq}$.
- $d_{seq}$ is the maximum number of flip-flops on any path between PI and PO.
- Both good and faulty circuits are initializable.
- Test sequence length for a fault is bounded by ($d_{seq}$ + 1).

23

## Cycle-Free Example

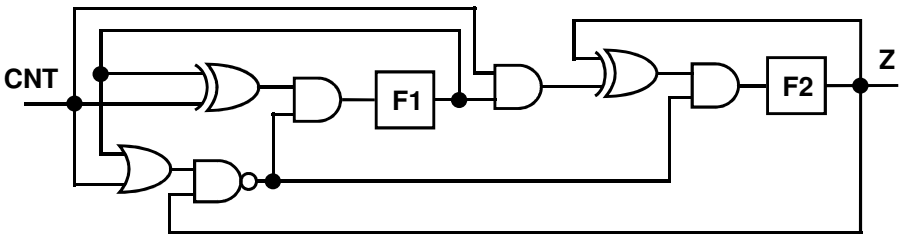



All faults are testable.

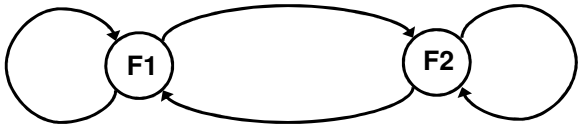
24

## Cyclic Circuit Example

Modulo-3 counter




s - graph




25

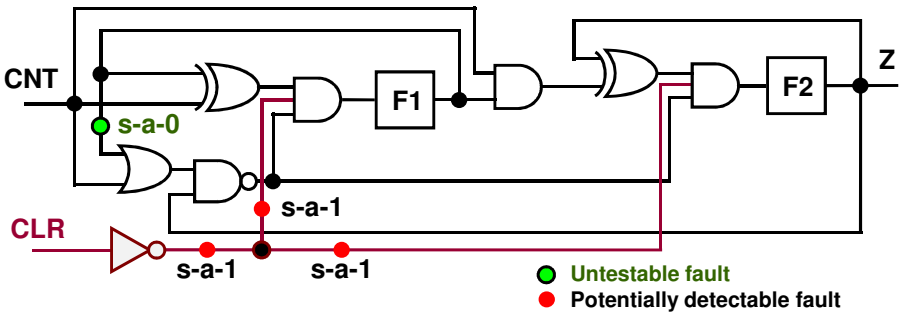
## Modulo-3 Counter

- Cyclic structure
  - Sequential depth is undefined.
- Circuit is not initializable.
  - No tests can be generated for any stuck-at fault.
- After expanding the circuit to $9^{Nff} = 81$, or fewer, time-frames ATPG program calls any given target fault ***untestable***.
- Circuit can only be functionally tested by multiple observations.
- Functional tests, when simulated, give no fault coverage.

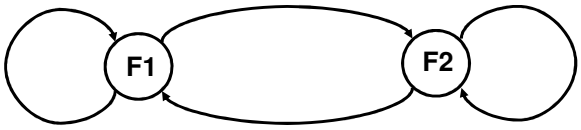
26

# Adding Initializing Hardware

**Initializable modulo-3 counter**




**s - graph**




27

# Benchmark Circuits

| Circuit | s1196 | s1238 | s1488 | s1494 |
|---|---|---|---|---|
| PI | 14 | 14 | 8 | 8 |
| PO | 14 | 14 | 19 | 19 |
| FF | 18 | 18 | 6 | 6 |
| Gates | 529 | 508 | 653 | 647 |
| Structure | Cycle-free | Cycle-free | Cyclic | Cyclic |
| Seq. depth | 4 | 4 | -- | -- |
| Total faults | 1242 | 1355 | 1486 | 1506 |
| Detected faults | 1239 | 1283 | 1384 | 1379 |
| Potentially detected faults | 0 | 0 | 2 | 2 |
| Untestable faults | 3 | 72 | 26 | 30 |
| Abandoned faults | 0 | 0 | 76 | 97 |
| Fault coverage (%) | 99.8 | 94.7 | 93.1 | 91.6 |
| Max. sequence length | 3 | 3 | 24 | 28 |
| Total test vectors | 313 | 308 | 525 | 559 |
| Gentest CPU s (Sparc 2) | 10 | 15 | 19941 | 19183 |

28

## Summary

- Combinational ATPG algorithms are extended:
  - Time-frame expansion unrolls time as combinational array
  - Nine-valued logic system
  - Justification via backward time
- Cycle-free circuits:
  - Require at most dseq time-frames
  - Always initializable
- Cyclic circuits:
  - May need 9Nff time-frames
  - Circuit must be initializable
  - Partial scan can make circuit cycle-free

29