Sequential ATPG

Virendra Singh
Associate Professor
Computer Architecture and Dependable Systems Lab
Dept. of Electrical Engineering
Indian Institute of Technology Bombay
viren@ee.iitb.ac.in

EE 709: Testing & Verification of VLSI Circuits

Lecture – 16 (Feb 07, 2012)

Static and Dynamic Compaction of Sequences

- **Static compaction**
  - **ATPG should leave unassigned inputs as X**
  - **Two patterns *compatible* – if no conflicting values for any PI**
  - **Combine two tests t_a and t_b into one test $t_{ab} = t_a \cap t_b$ using D-intersection**
  - **Detects union of faults detected by t_a & t_b**
- **Dynamic compaction**
  - **Process every partially-done ATPG vector immediately**
  - **Assign 0 or 1 to PIs to test additional faults**

Compaction Example

- t_1 = 0 1 X t_2 = 0 X 1
  t_3 = 0 X 0 t_4 = X 0 1
- **Combine t_1 and t_3, then t_2 and t_4**
- **Obtain:**
  - t_{13} = 0 1 0 t_{24} = 0 0 1
- ***Test Length* shortened from 4 to 2**

Sequential Circuits

- A sequential circuit has memory in addition to combinational logic
- Test for a fault in a sequential circuit is a sequence of vectors, which
  - Initializes the circuit to a known state
  - Activates the fault, and
  - Propagates the fault effect to a PO
- Methods of sequential circuit ATPG
  - Time-frame expansion methods
  - Simulation-based methods

Example: A Serial Adder

A_n B_n

1 1

s-a-0

1

$\overline{D}$

1

$\overline{D}$

X

C_n

X

C_{n+1}

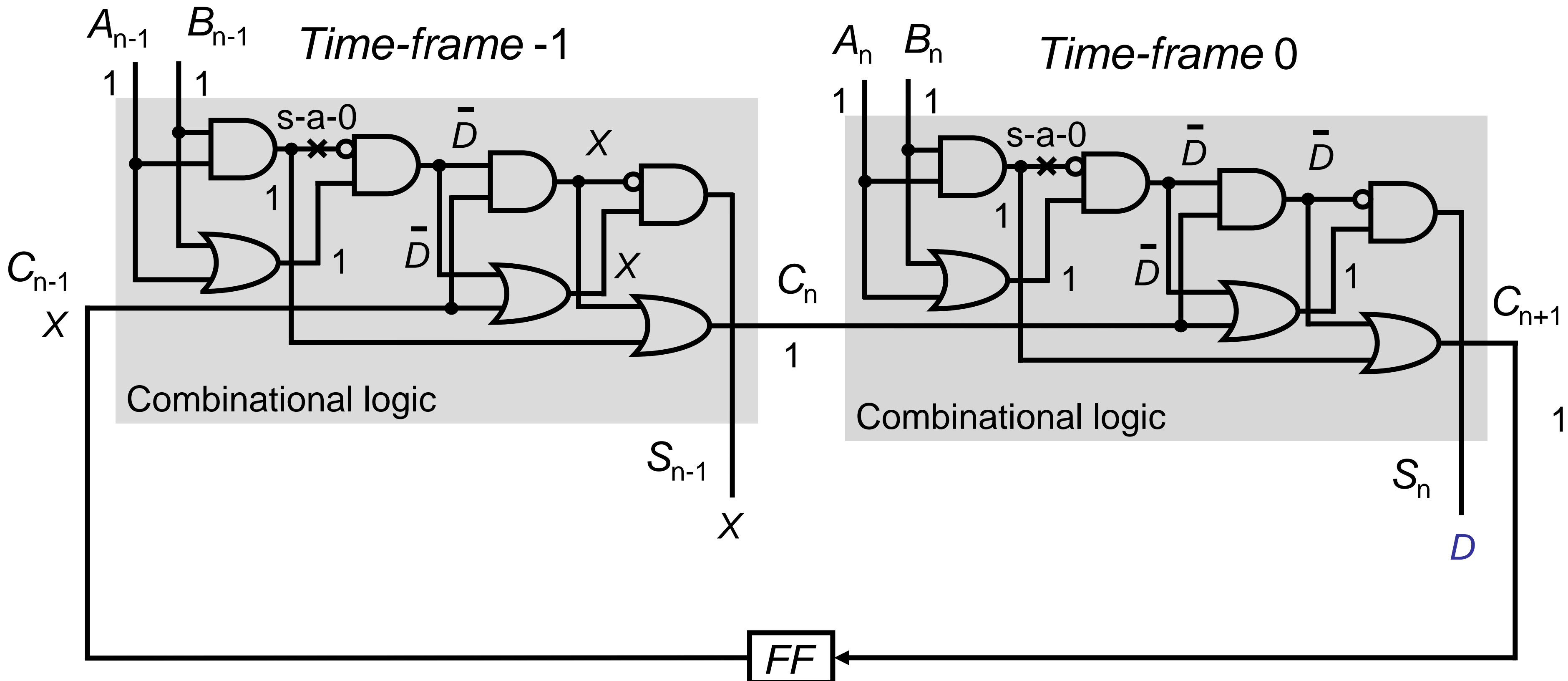
1

Combinational logic

S_n X

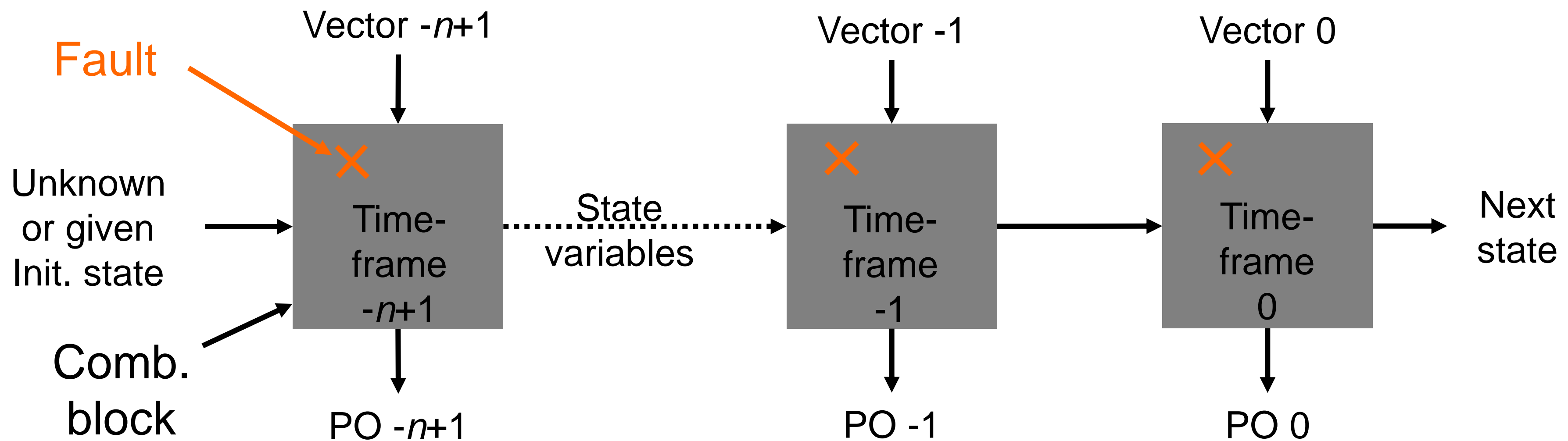
FF

Time-Frame Expansion

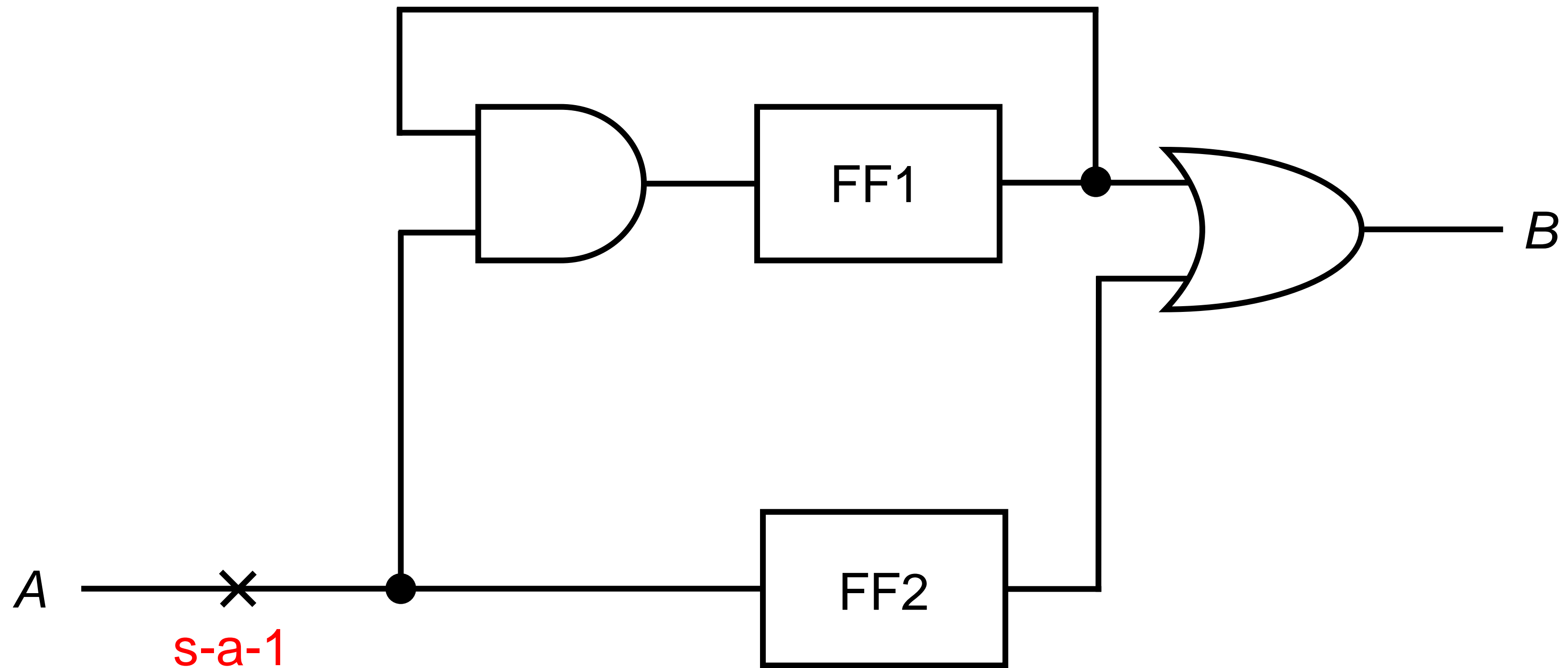


Concept of Time-Frames

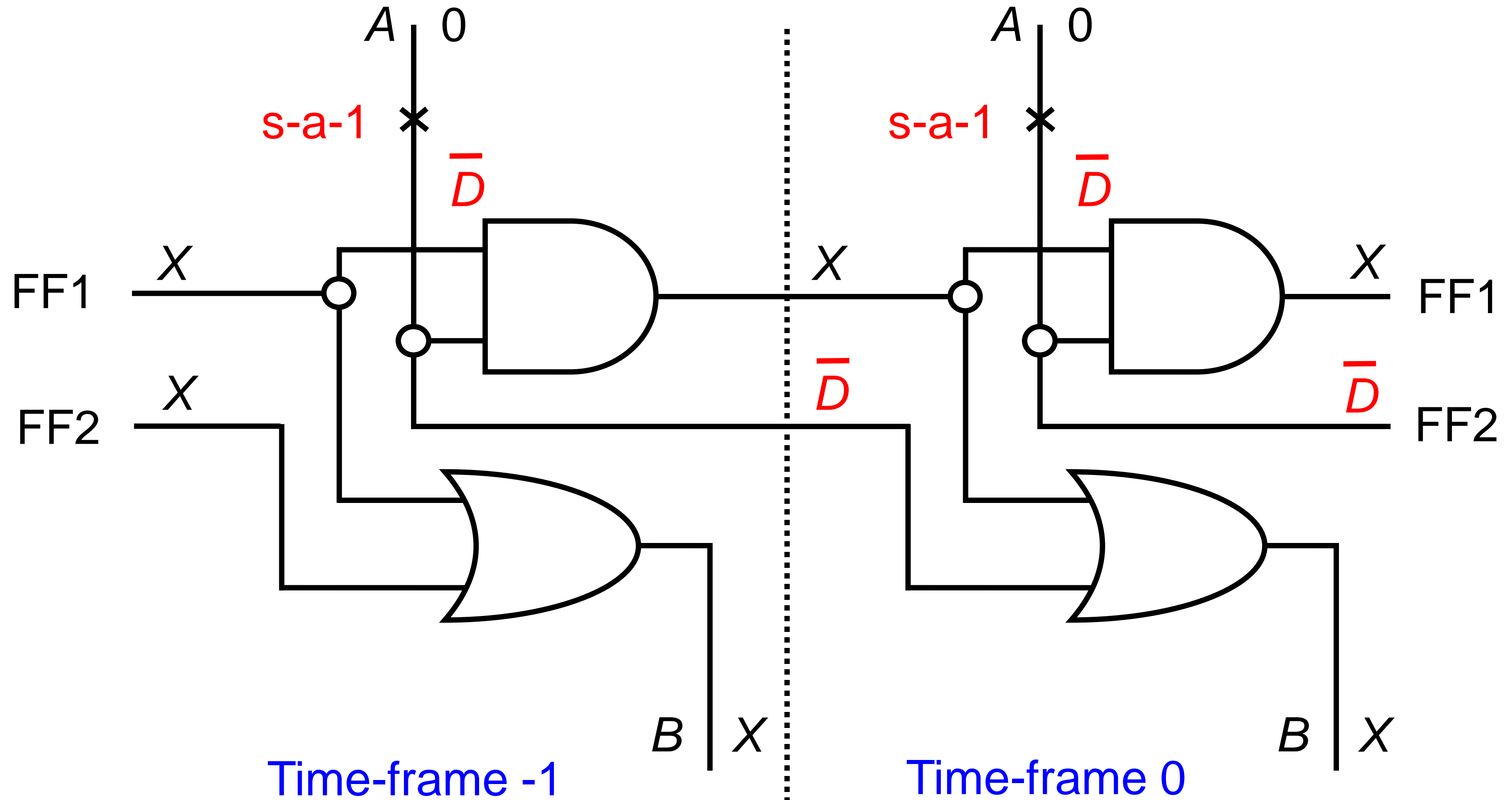
- If the test sequence for a single stuck-at fault contains *n* vectors,
  - Replicate combinational logic block *n* times
  - Place fault in each block
  - Generate a test for the multiple stuck-at fault using combinational ATPG with 9-valued logic



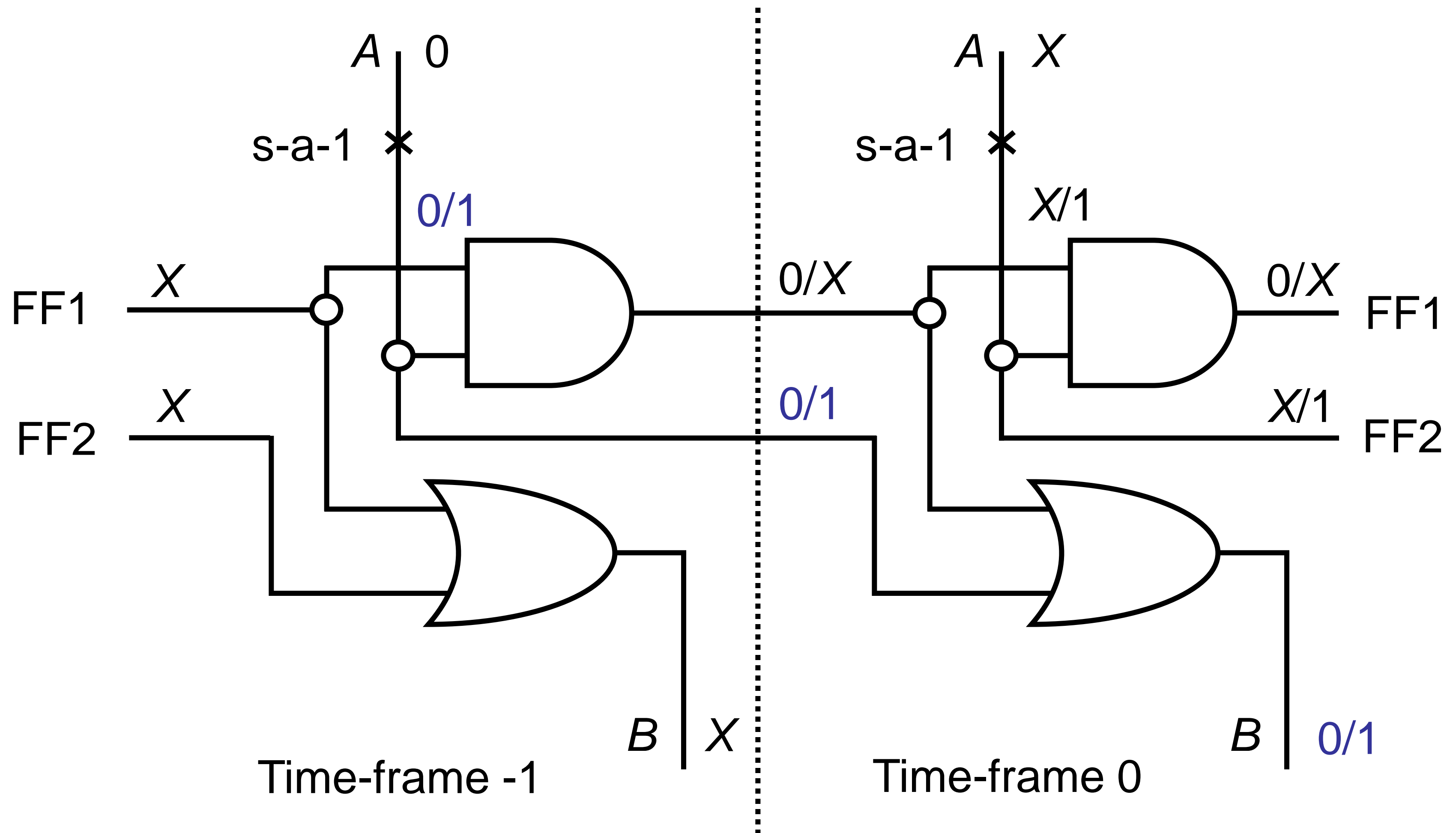
Example for Logic Systems



Five-Valued Logic (Roth)



Nine-Valued Logic (Muth)

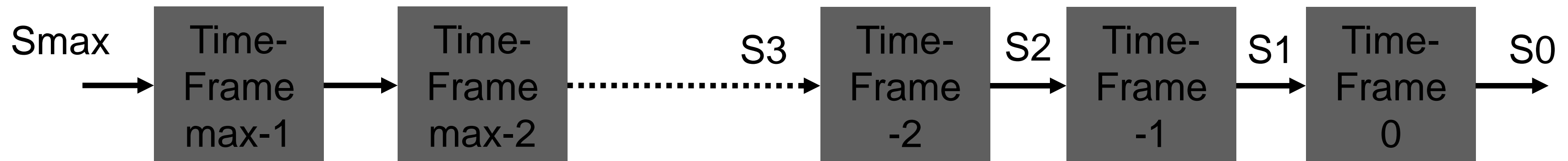


Implementation of ATPG

- 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, select another PO and repeat justification (use drivability).
- 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, the the fault is *potentially detectable.*

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 *dseq* + 1 time-frames, where *dseq* 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!

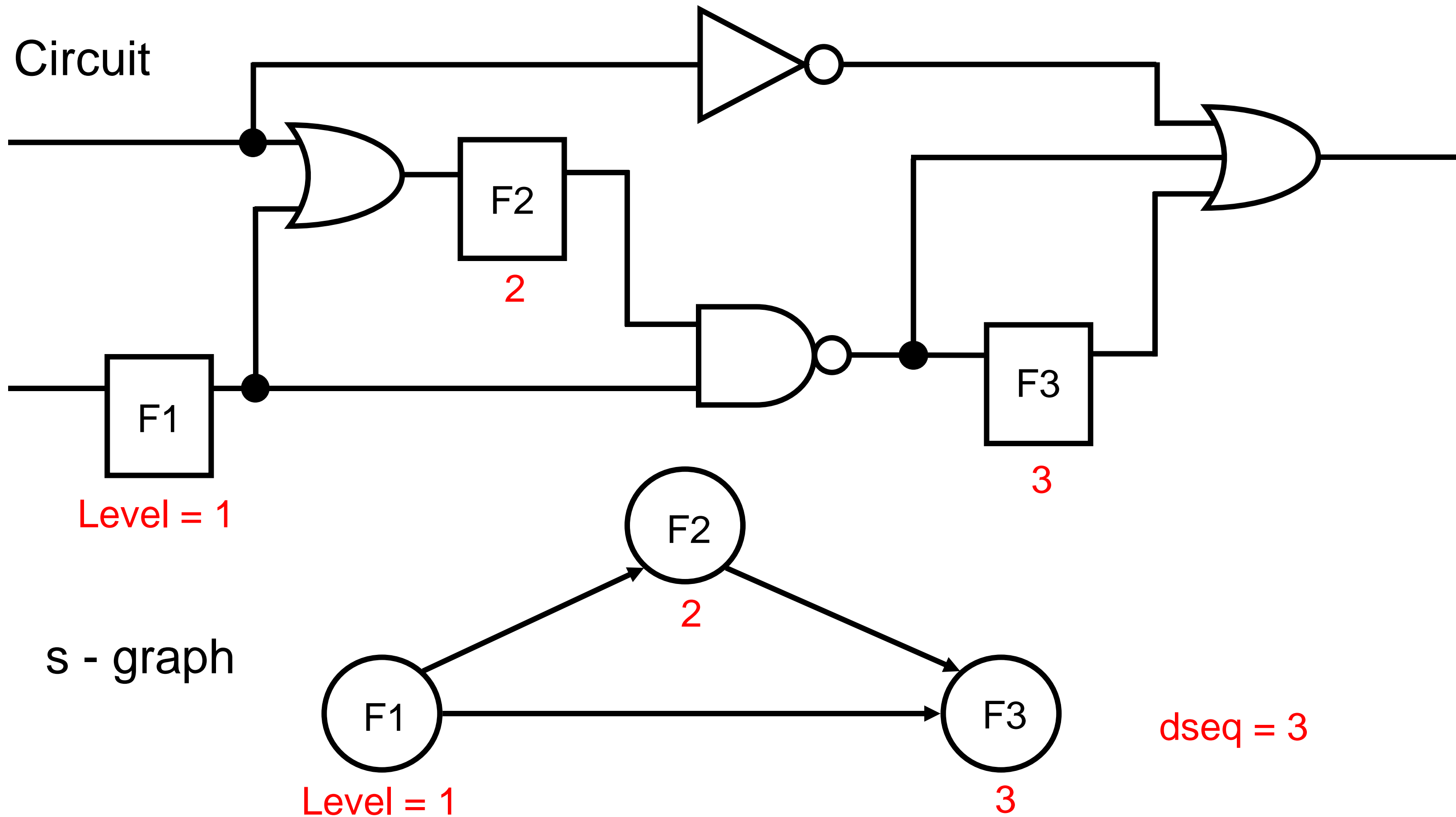


max = Number of distinct vectors with 9-valued elements = 9^{Nff}

Cycle-Free Circuits

- Characterized by absence of cycles among flip-flops and a sequential depth, *dseq*.
- *dseq* 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 *dseq* + 1.

Cycle-Free Example



All faults are testable.

Thank You