VLSI Test Technology and Reliability (ET4076)

Lecture 6

Sequential Circuit Test Generation

(Chapter 8)

Said Hamdioui

Computer Engineering Lab
Delft University of Technology
2009-2010

Learning aims of today's lecture

- Be able to:
  - Describe the seq circuit testing problem
  - Describe the different methods used for seq ATPG
  - Apply TFEM to generate test vectors for seq circuits

Contents

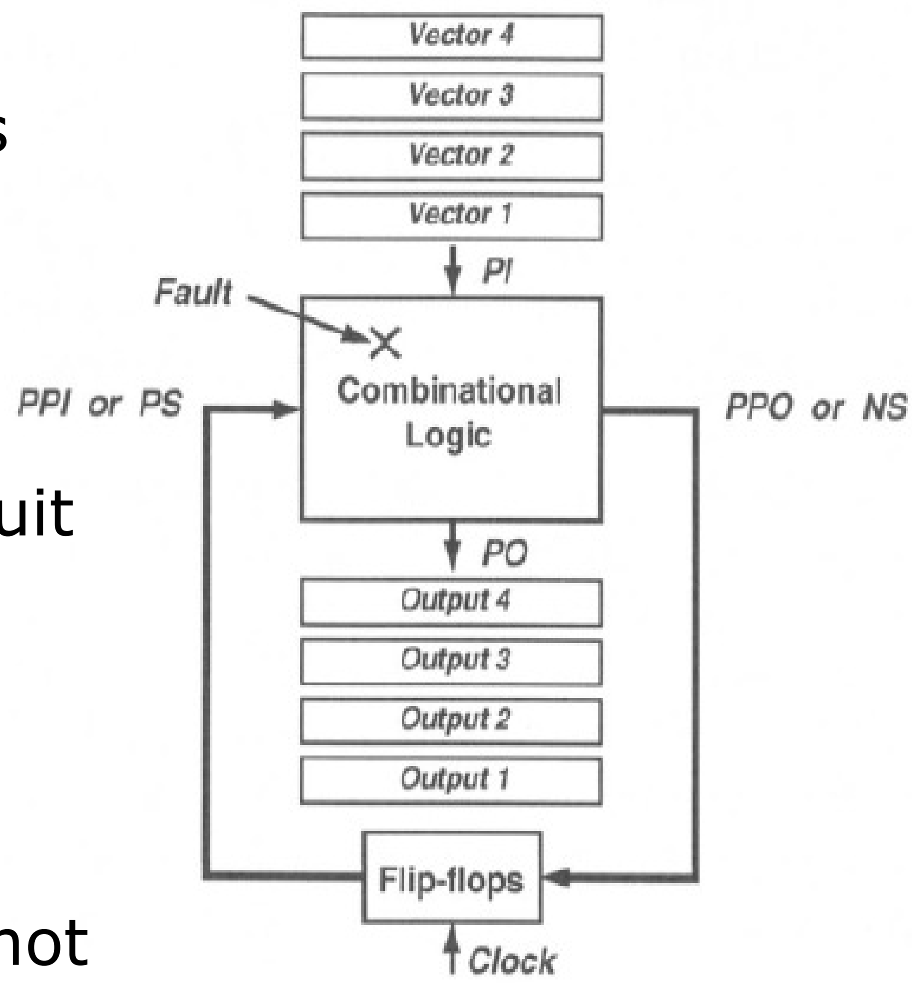
- Problem of sequential circuit ATPG
- Methods for sequential circuit ATPG
- Time-frame expansion
  - Concept
  - Use of nine-valued logic
  - Drivability
  - ATPG implementation
  - ATPG Complexity
  - Cycle-free and cyclic circuits
  - Asynchronous circuits
  - Summary
- Simulation-based sequential circuit ATPG*

Problem of sequential circuit testing (1)

- 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 primary output
- Testing sequential circuits is more complex than combinational testing
  - Unknown internal states
  - Long test sequences
- Two basic differences between Comb. and Seq testing
  - A test for a fault in seq circuit may consist of several vectors
  - The five-value (1,0,D,D*,X) is insufficient for seq circuits

Problem of sequential circuit testing (2)

- Sequential circuit
  - Comb. logic + Flip Flops



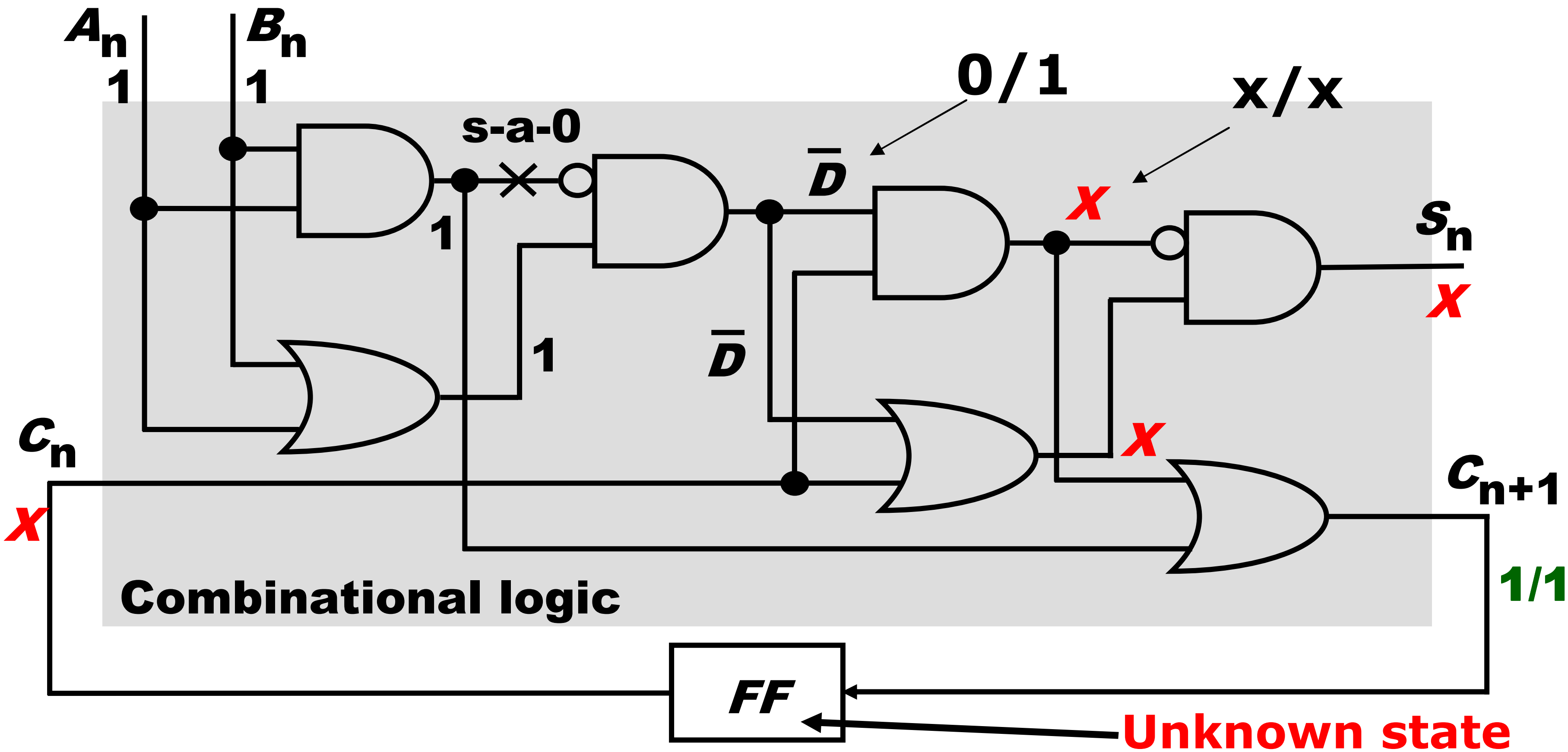
- Simplified problem
  - SAF model for logic circuit
  - FFs are treated as ideal memory elements
  - No fault in the clock is modeled
  - Internal faults of FF as not modeled
    - SAF at inputs and outputs

Methods for sequential circuit ATPG

- **Time-frame expansion methods (TFEM)**
  - The circuit is modeled in such a way that combinational ATPG can be used
  - Very efficient for circuits described at logic-level
  - Not efficient for asynchronous circuits, multiple clocks, cyclic structures
- **Simulation-based methods***
  - Fault simulator and vector generator used to derive tests
  - Test can be generated for any circuit that can be simulated
  - Memory and time consuming
  - Lower FC unless extra observation points added

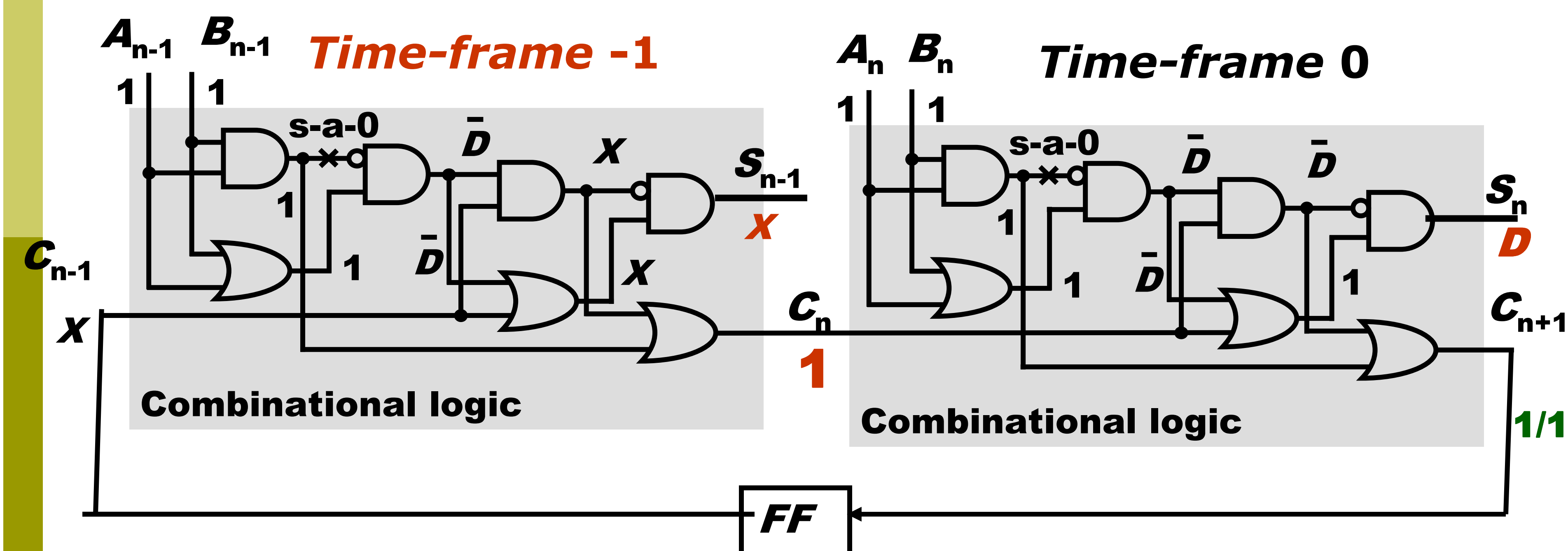
Time-frame expansion (TFE) method….Example(1)

- Example: **Serial** Adder, s-a-0
  - Use D Algorithm: $A_n=B_n=1$
  - Fault cannot be propagated (C_n=X)
  - To propagate the fault, C_n has to be 1
  - Hence, precede the test vector 11 with an <u>initialization vector</u>



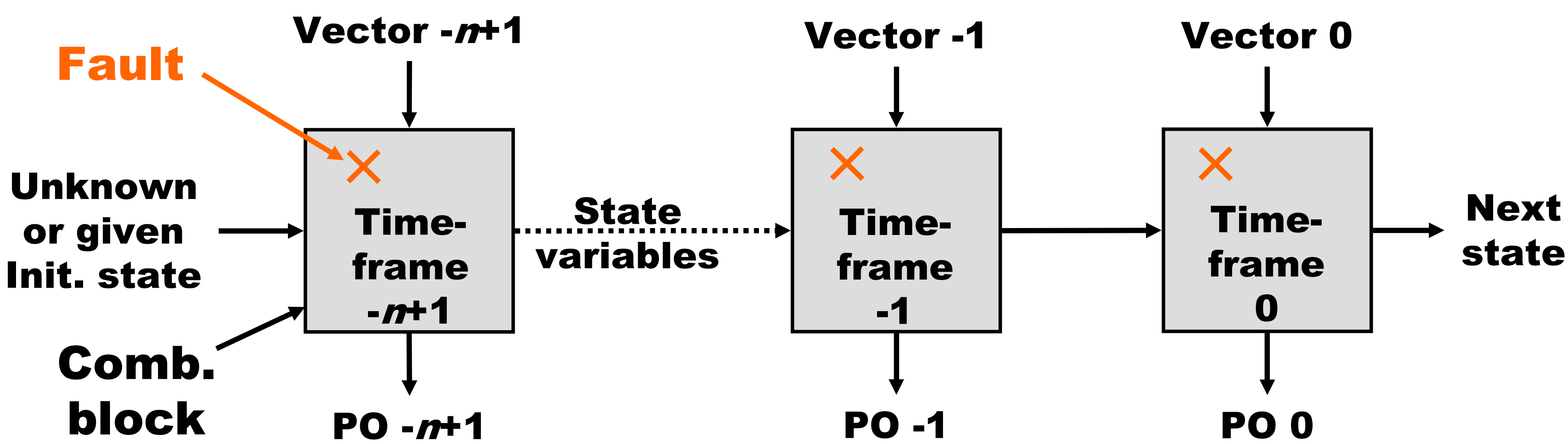
TFE method……….. Example(2)

- Use the **expended circuit model**
  - Combinational circuit is repeated twice
  - Generate a combinational test that detect **multiple faults**
  - Example:
    - Generate vector A_n B_n =11 for Time-frame 0
    - Generate vector $A_{n-1}B_{n-1}$=11 for time-frame 1



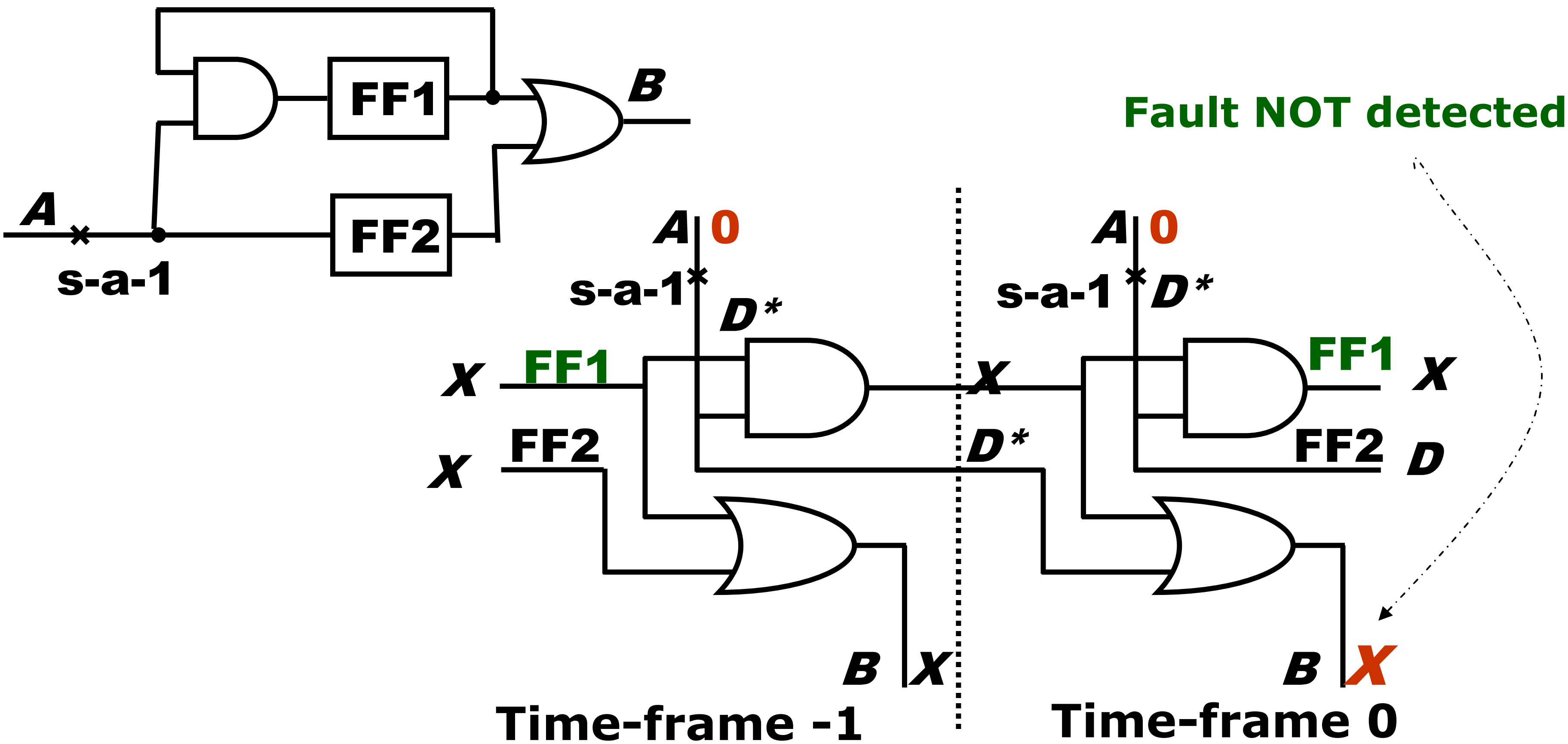
TFE method ……… Concept of Time-Frames

- 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**



TFE method…… Use of nine-valued logic(1)

- If A is s-a-1, B always 1: Fault will be detected
- Test generation with **5 logic values (1,0,D,D*,X)**
  - Fault does **not allow initialization** of FF1
  - 0/X is regarded as X in 5 logic values



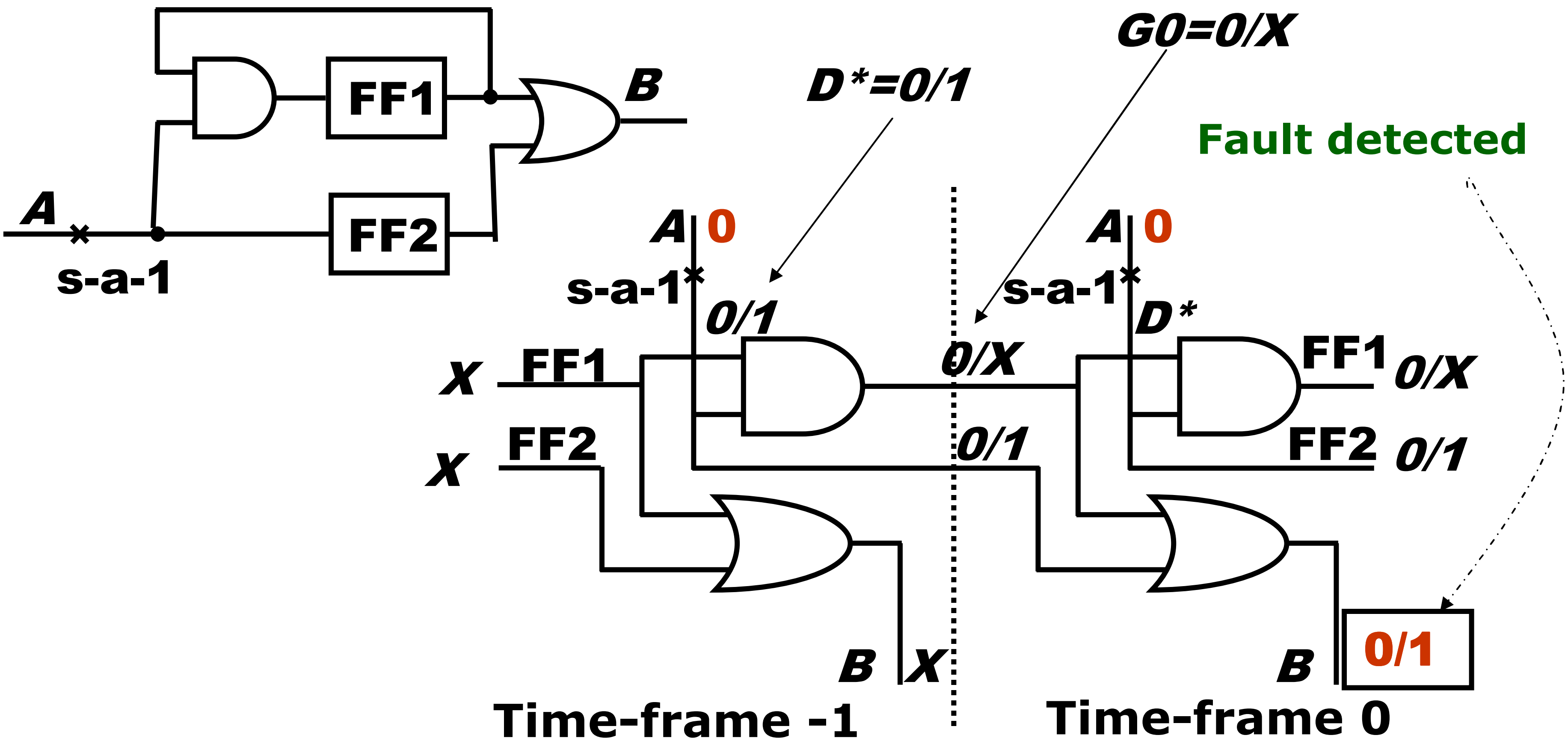
TFE method……Use of nine-valued logic(2)

- Roth's 5-Valued and Muth's 9-Valued

Symbol	Meaning	Good machine	Failing machine	
D	1/0	1	0	Roth's Algebra [5 values]
D*	0/1	0	1	
0	0/0	0	0	
1	1/1	1	1	
X	X/X	X	X	
G0	0/X	0	X	Muth's additions [Extended unknowns]
G1	1/X	1	X	
F0	X/0	X	0	
F1	X/1	X	1	

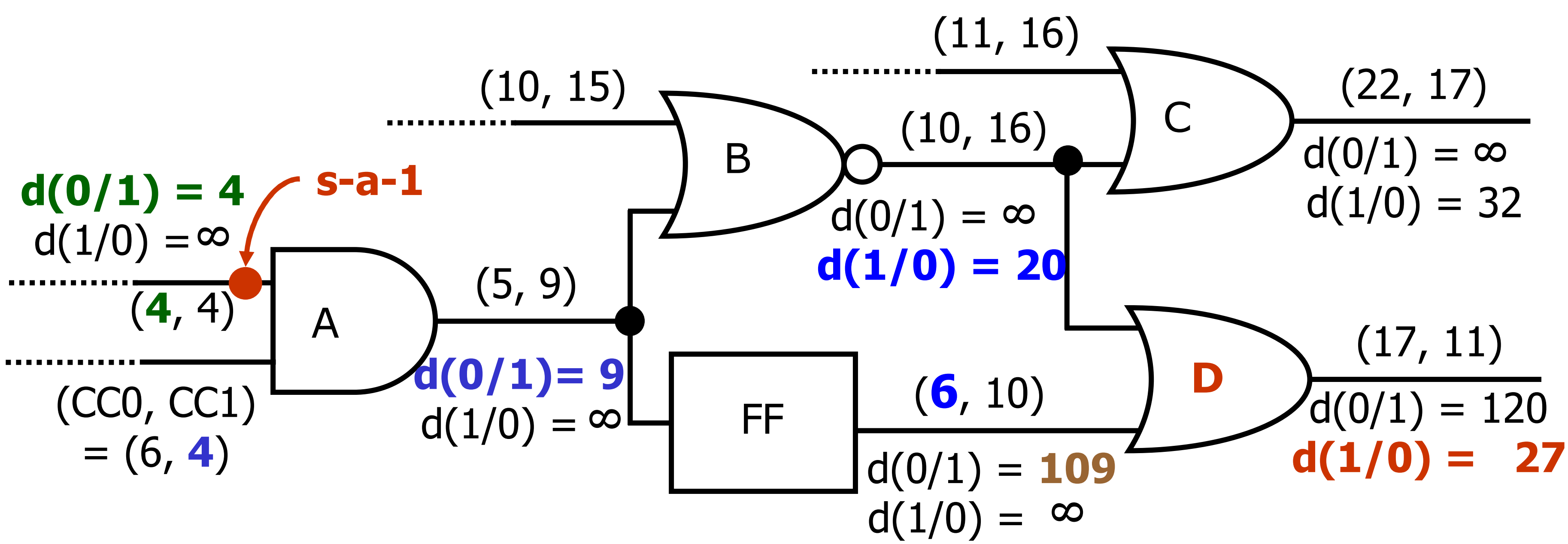
TFE method……Use of nine-valued logic(2)

- If A is s-a-1, B always 1: Fault will be detected
- Test generation with **9 logic values (Muth)**
  - **0, 1, 0/1, 1/0, 1/X, X/1, 0/X, X/0, X**
  - Fault detected with A=0 and after one clock



TFE method……… Drivability

- **Drivability**: Testability measure of the effort of driving the fault effect from the fault site to PO (use SCOAP measures)
- Example: CC0 and CC1 are SCOAP combinational controllabilities
  - Depth of the circuit is assumed to be **100** (100 times frames)



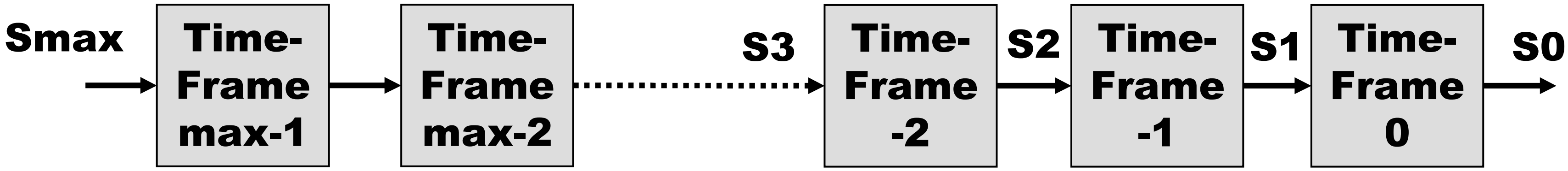
- **d(0/1)** and **d(1/0)** are the drivability measures
- Output of D is the best choice for detecting the fault: d(1/0)=27

TFE method…….ATPG Implementation

- 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.*

TFE method ….. ATPG Complexity

- 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}

TFE method ……..Cycle-Free Circuits(1)

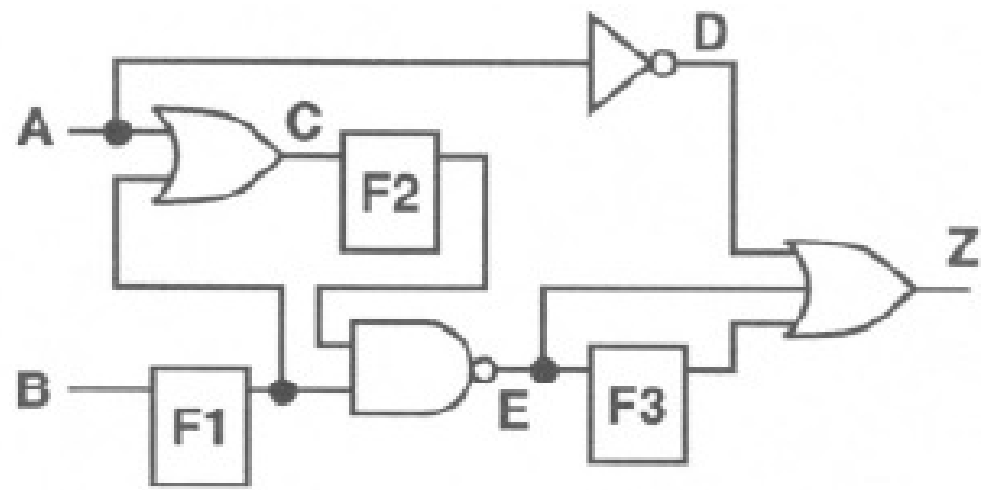
Cycle-Free Circuits

- 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.

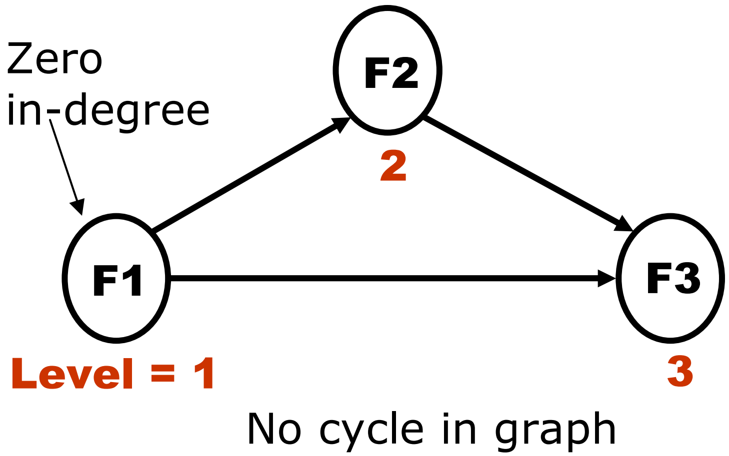
TFE method …….Cycle-Free Circuits(2)

Cycle-Free Circuit Example

dseq = 3



Directed graph: s - graph

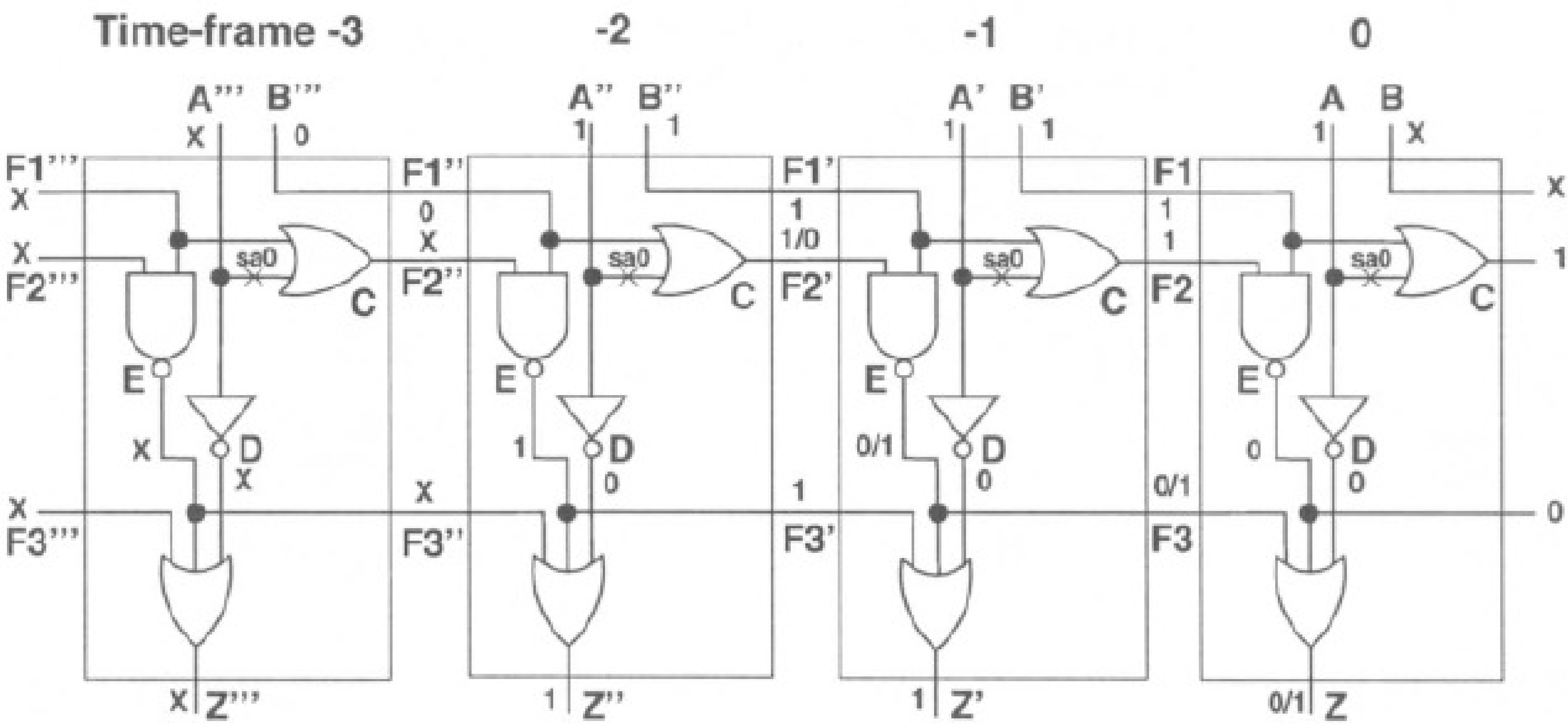


Theorem: A test for non-FF fault in a cycle free circuit can always be found with at most **dseq+1 time-frames** unless the fault in untestable

All faults are testable.
See Example 8.6.

TFE method …….Cycle-Free Circuits(3)

- Example
  - We can generate a test for any fault (all are testable) in the circuit *using at most four time-frames*



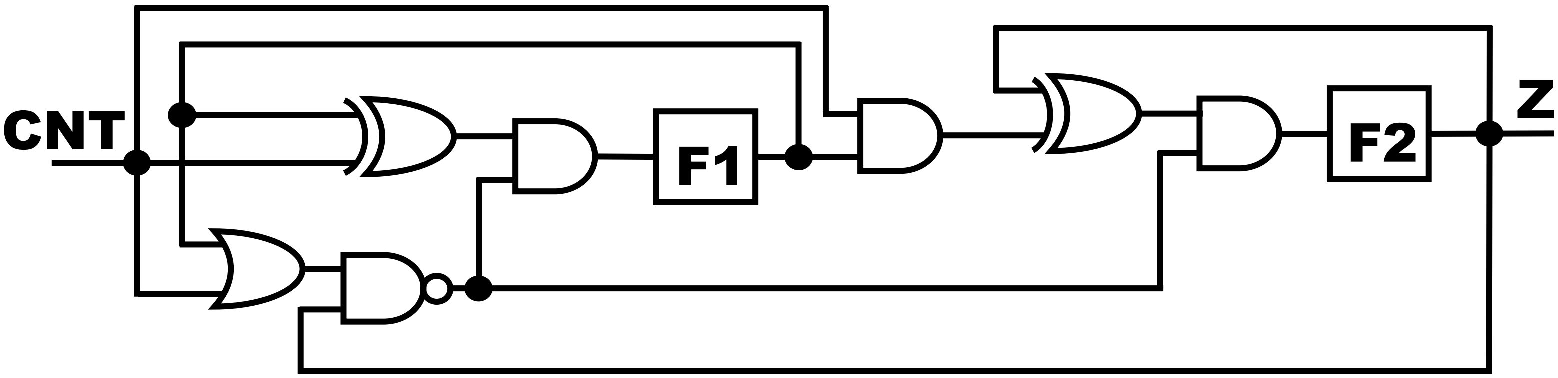
TFE method ………. Cycle Circuits(1)

Cycle Circuits

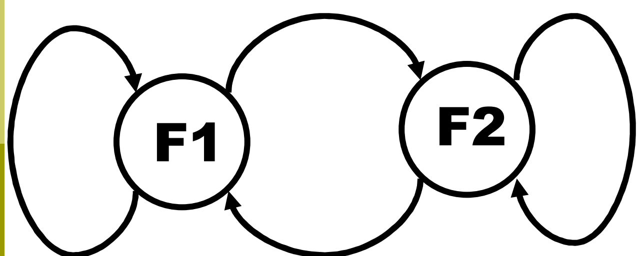
- Cyclic structure
  - Sequential depth is undefined.
- Circuit is not always initializable.
  - No tests can be generated for any stuck-at fault.
- After expanding the circuit to 9^{Nff}, or fewer, time-frames ATPG program calls any given target fault untestable.
- Usually low fault coverage reported by the test generator is related to initialization problems

TFE method ………. Cycle Circuits(2)

Cycle Circuit Example: Modulo-3 counter



Directed graph: s - graph



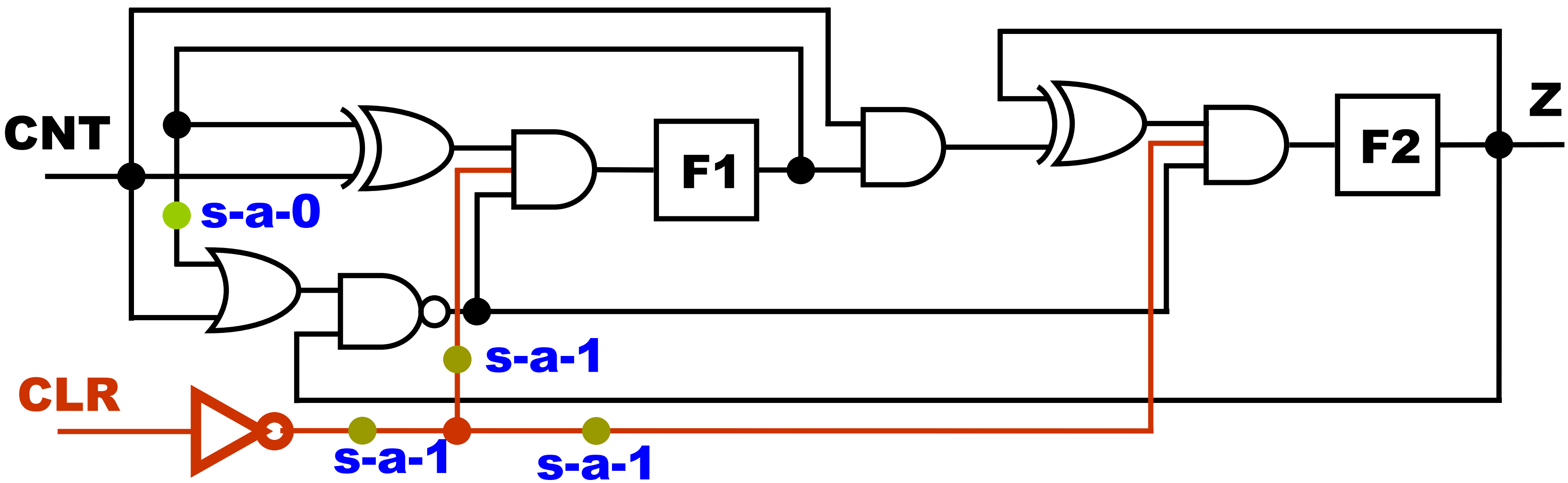
many cycle in graph

- When CNT=1, counter increments its state each time FF clocked 00, 01, 10, 00,..
- When CNT=0, states are hold
- ATPG: faults are untestable
  - State of the fault free circuit cannot be determined

- **Remedy**: Provide an **initialization input**

TFE method .... Cycle Circuits(3)

- **Adding Initializing Hardware: CLR**
  - **Synchronous initialization**



- ATPG:
  - Detects all faults with 9 vectors except 4 shown in the figure
    - Three s-a-1 in CLR* are potentially detectable
    - s-a-0 is partially detectable
      - When circuit powered on: if FF1=FF2=1 (not valid state), then detected

TFE method ……… Benchmark Circuits

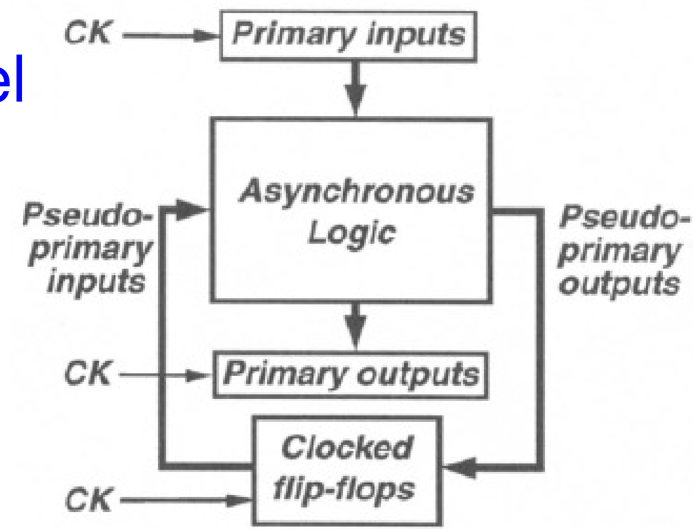
- Obtained at Bell Labs using GENTEST
  - Abandoned fault: if CPU time limit per fault is reached (e.g. 100s, 81s)
  - Fault efficiency = detected faults / (Total faults – Untestable faults)
  - Max. sequence length= longest test sequence for any fault

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
Fault efficiency (%)	100.0	100.0	94.8	93.4
Max. sequence length	3	3	24	28
Total test vectors	313	308	525	559
GENTEST CPU s (Sparc 2)	10	15	19941	19183

TFE method ……… Asynchronous Circuit

- An asynchronous circuit contains unclocked memory often realized by combinational feedback.
  - Some signals can change at any time
  - Some signals may depend on the past inputs
  - Steady state signal values may depends on the circuit delays
- Clock generators, signal synchronizers, flip-flops are typical asynchronous circuits.
- Almost impossible to build, let alone test, a large asynchronous circuit.
- Many large synchronous systems contain small portions of localized asynchronous circuitry.
- Sequential circuit ATPG should be able to generate tests for circuits with limited asynchronous parts, even if it does not detect faults in those parts.

TFE method .. Asynchronous Model

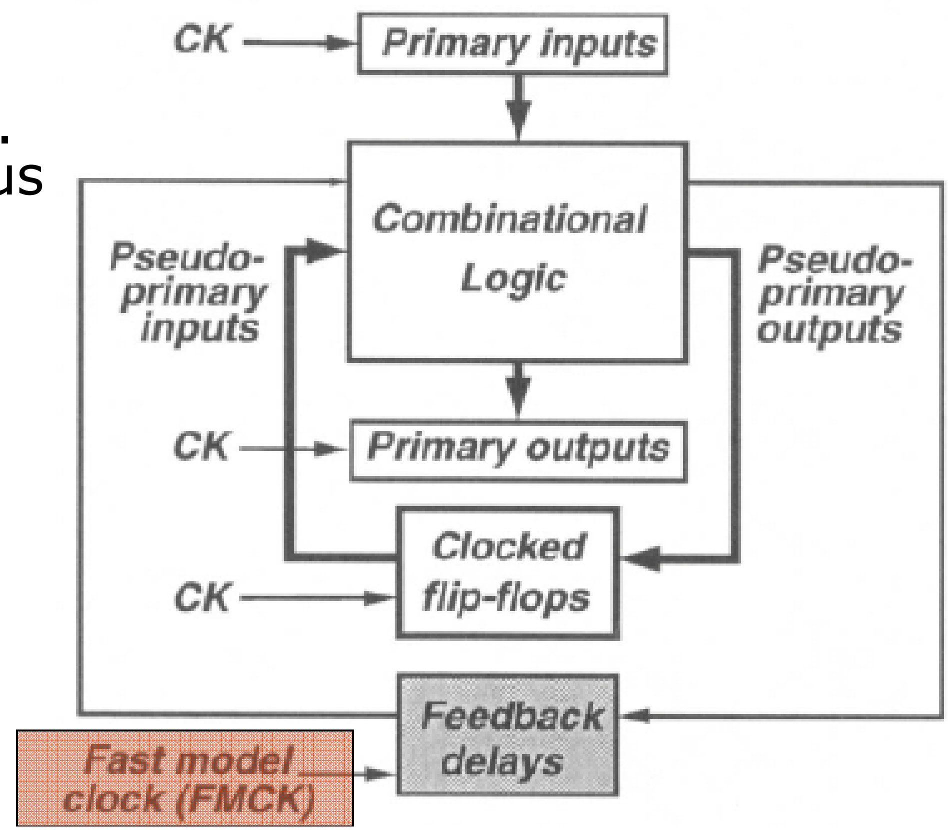


Asynchronous logic is split:

- Feedback **free combinational logic** and
- **Feedback delays** synchronized with **Fast Model Clock FMCK**

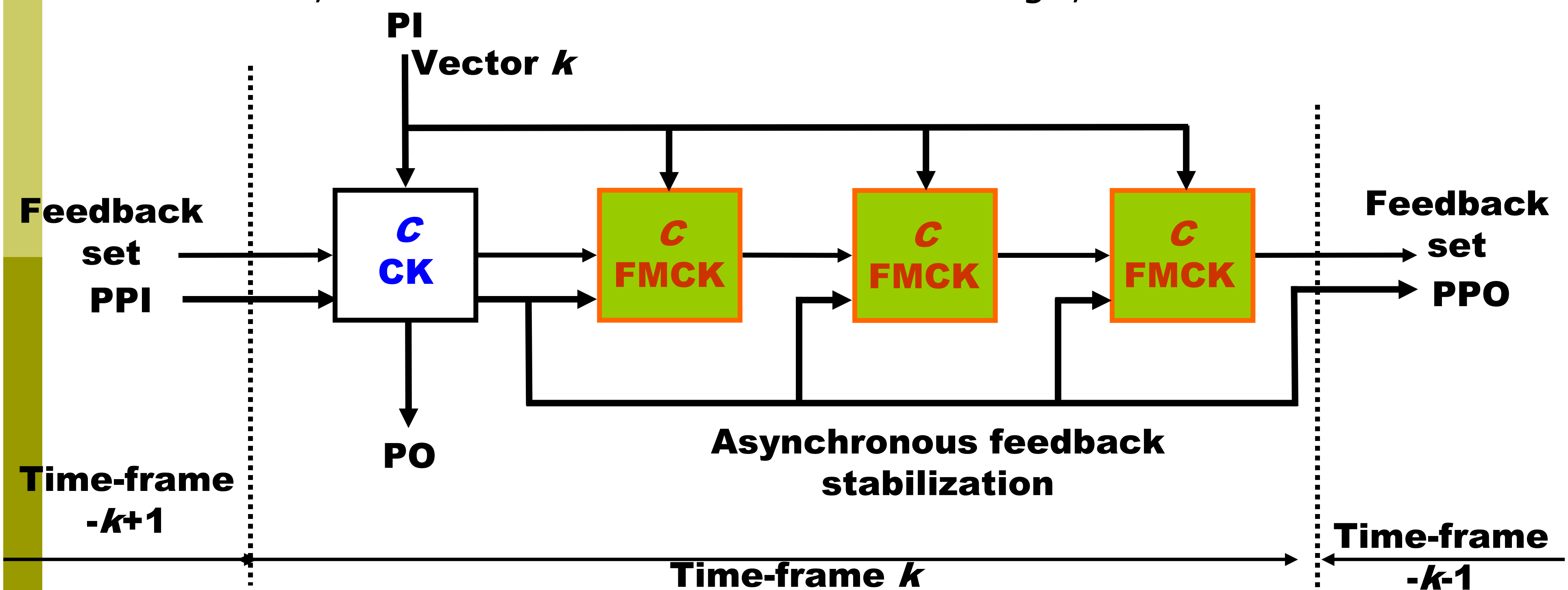
FMCK runs much faster than CK

- Repeatedly evaluate the comb. logic and stabilize asynchronous signals before CK clocks FFs, applies PI or observes PO

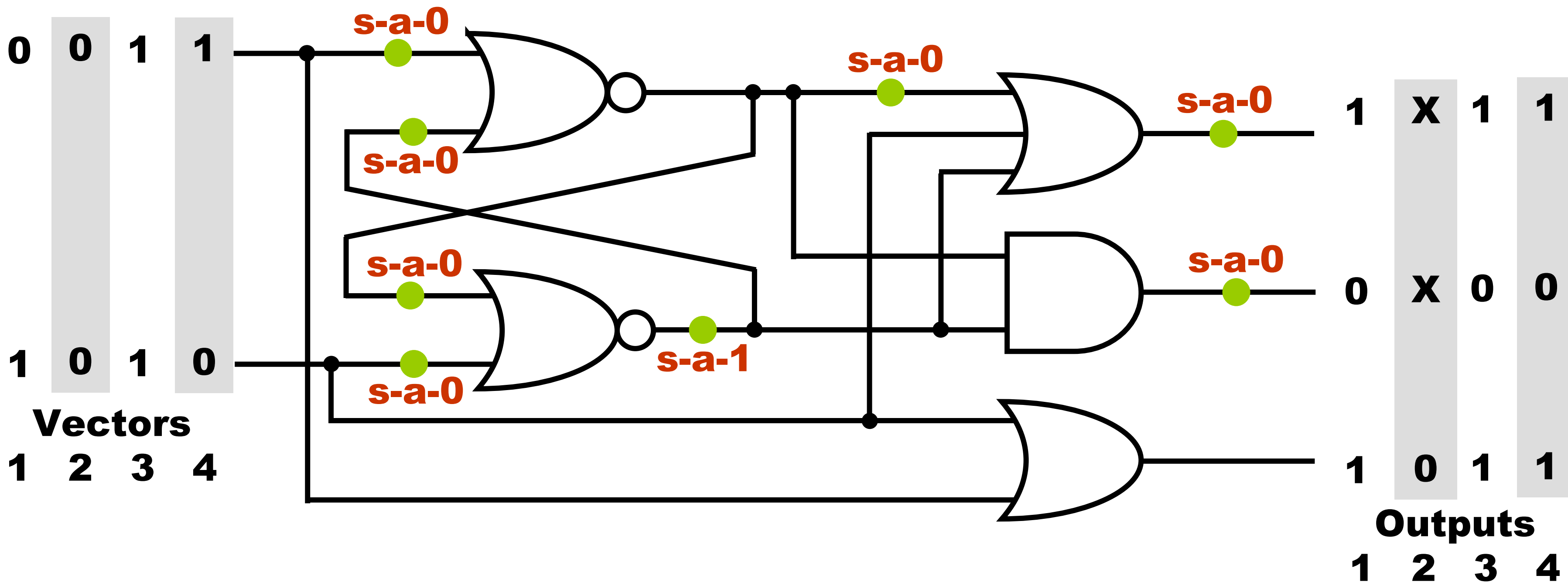


TFE method ……… Time-Frame Expansion

- TME deals with the circuits in two phases:
  - System clock
    - Input vector generated for each period of CK
    - PO are observed once each CK period
  - FMCL
    - Fast time-frames exercise logic signals till they become stable
    - PIs, clocked FFs are held without change; no POs examined



TFE method ….. Asynchronous Example*



- Gentest results:
  - Faults: total 23, detected 15, **untestable 8** (shown in red), Potentially detectable none
  - Vectors: 4
  - Sparc 2 CPU time: test generation 33ms, fault simulation 16ms

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 9^{Nff} time-frames
  - Circuit must be initializable
  - Partial scan can make circuit cycle-free (Chapter 14)
- Asynchronous circuits:
  - **High complexity**
  - **Low coverage and unreliable tests**
  - Simulation-based methods are more useful (Section 8.3)