
DO NOT SHARE
SLIDES AND CLASS MATERIALS
ON ONLINE SITES
Course Hero

Advanced Logic Design
Testing

Mingoo Seok
Columbia University

BV chapters: 11

Testing

- You designed a logic circuit. Then you need to test it working to the logic specification
- The logic circuit is very large and it is not trivial to test them in a short amount of time (longer testing time increases manufacturing cost)
- Testing sequential circuits is even more complex than combinational counterparts due to the state dependencies

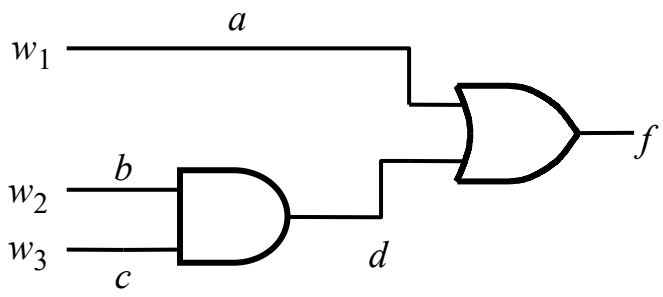
Fault Model

- Stuck-At model
  - Stuck-at-0
  - Stuck-at-1
  - Node w is stuck-at-0: w/0
  - Node w is stuck-at-1: w/1
- Physical meaning
  - Short to power supply voltage or ground
  - Open
  - Drain current is too small

Test Set

- The goal is to create several *tests* that are able to detect the occurrence of one or more faults
- *Test set*: a complete set of tests for a given circuit

Test Set Example



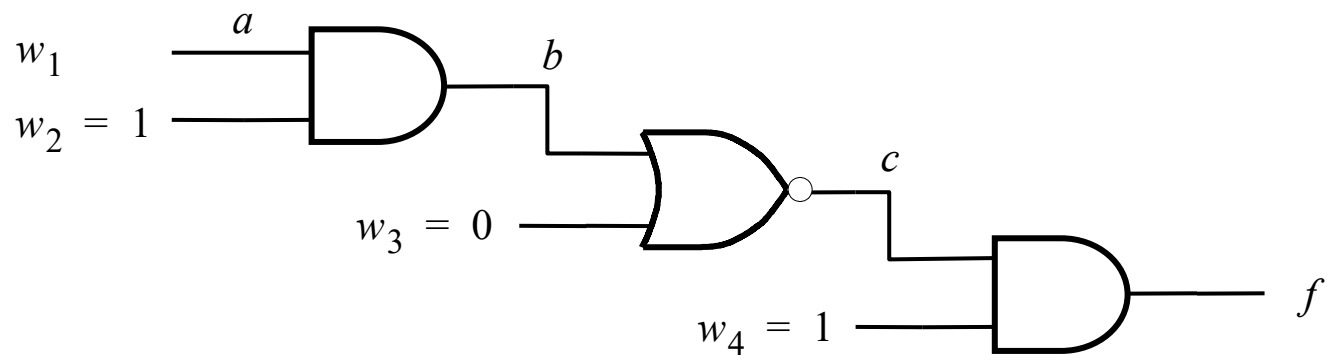
Test $w_1w_2w_3$	Faultdetected									
	a/0	*a*/1	*b*/0	*b*/1	*c*/0	*c*/1	*d*/0	*d*/1	*f* /0	*f* /1
000		√						√		√
001		√		√				√		√
010		√				√		√		√
011			√		√		√		√	
100	√								√	
101	√								√	
110	√								√	
111									√	

Test set = {001, 010, 011, 100}

Test Set

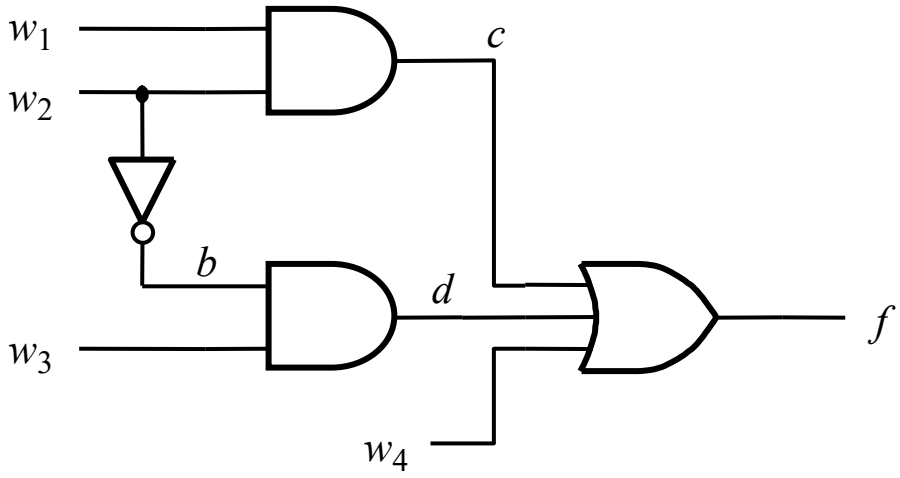
- Instead of 2^n test, a smaller number of tests can detect all the faults of interest (stuck-at-1 and stuck-at-0 faults)
- In the previous example: 4 out of 8
- Still, inspecting all the nodes of a circuit can take too much time due to the large number of nodes
- How can we do it better?

Path Sensitizing



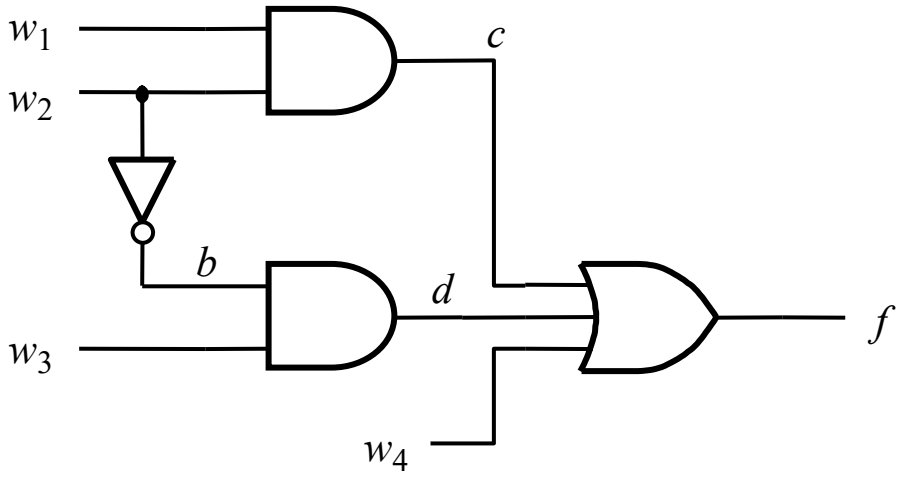
- Rather than nodes, we can inspect paths
- We can activate (sensitize) a path by ensuring that other paths in the circuits do not determine the value of the output f
- By assigning w_2, w_3, and w_4, we can make the path (a→b→c→f) sensitized to input w_1

Path Sensitizing: Example



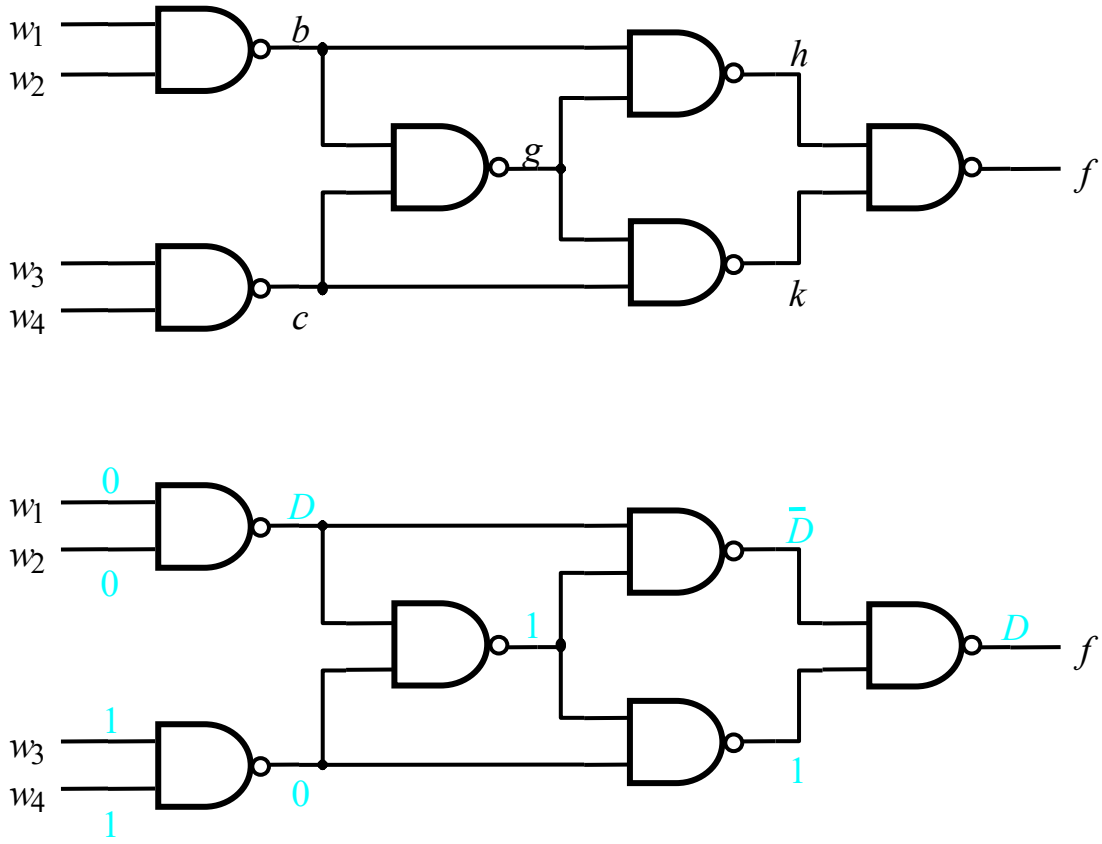
Test set = {0110, 1100, 1000, 0010, 0x00, 0x01}

Detecting a Specific Fault



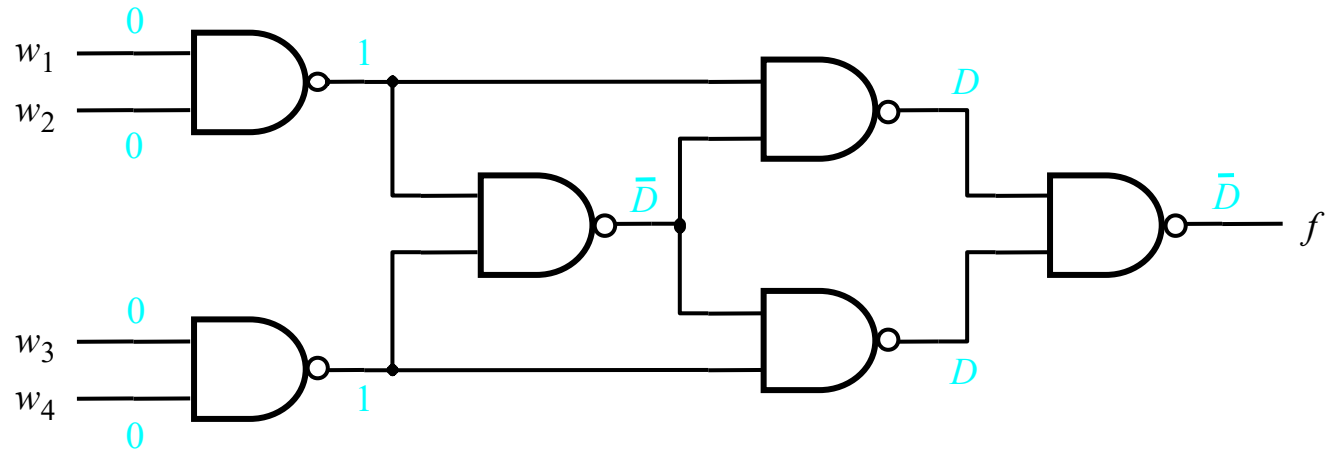
- How to check if the node b is stuck-at-1?
- ***Sensitize*** the path b-d-f
- ***Find the primary input(s)*** that makes the node b to 0
- ***Find the other primary input(s)*** that have not been set
- $w_1w_2w_3w_4$=0110

Propagating a Fault



- Test if b/0
- D: a stuck-at-0
- Consistency check: if we can apply appropriate values to the primary input variables for propagating a fault

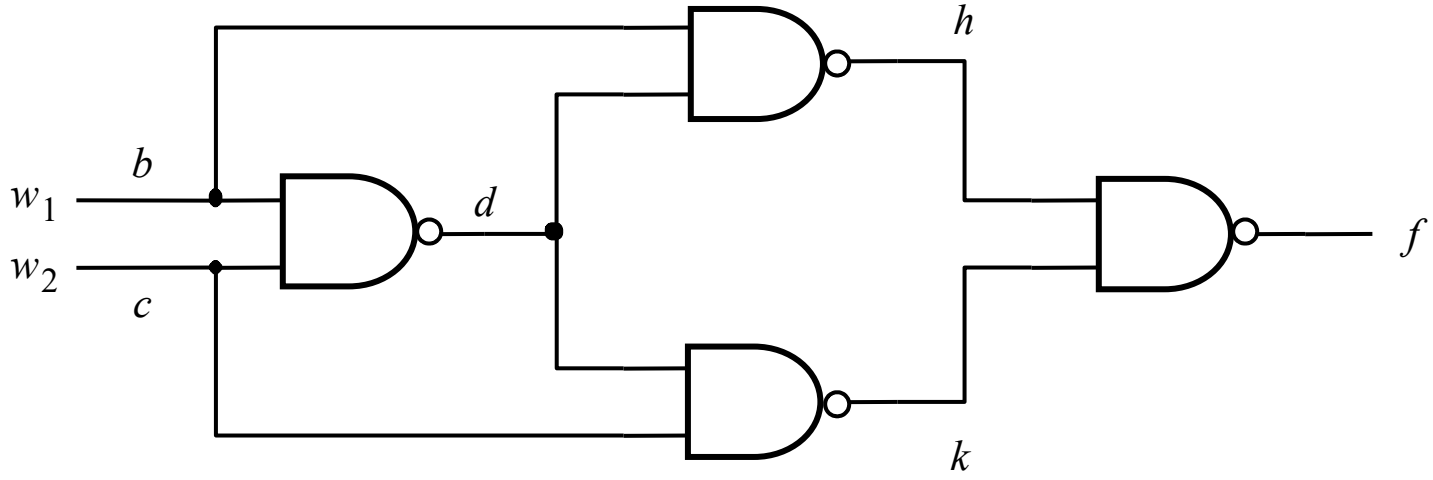
Propagating a Fault



- Let's test if **g/1**
- Propagate via only one NAND will fail the consistency check
- Propagating via both NANDs results in a test vector $w_1w_2w_3w_4$=0000

Random Test

$w_1 w_2$	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
00	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
01	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
10	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
11	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

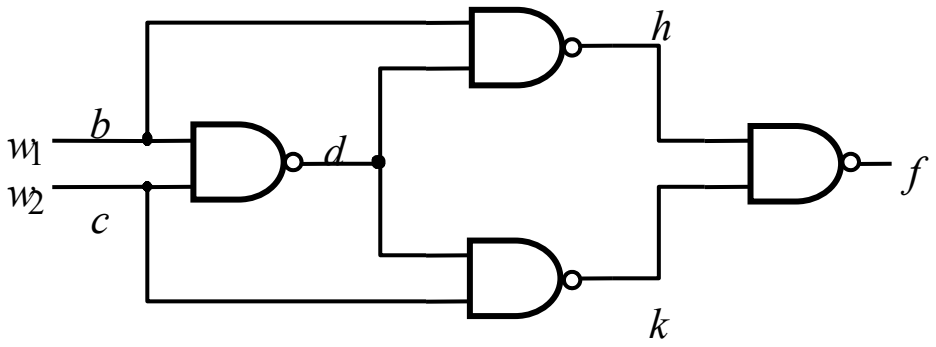


XOR function

Random Test

- For an n-variable input function, there are 2^{2^n} possibilities
- 2 input: 16 functions

Random Test



Fault	Circuit implements
b/0	$f_5 = w_2$
b/1	$f_{10} = \overline{w}_2$
c/0	$f_3 = w_1$
c/1	$f_{12} = \overline{w}_1$
d/0	$f_0 = 0$
d/1	$f_7 = w_1 + w_2$
h/0	$f_{15} = 1$
h/1	$f_4 = \overline{w}_1 w_2$
k/0	$f_{15} = 1$
k/1	$f_2 = w_1 \overline{w}_2$

- With each fault, f becomes one of the ten functions (f_5, f_{10}, …)
- If I use w_1w_2=01, I can detect f_0,f_2,f_3, and f_{10}
- 11 → f_5, f_7 and f_{15}
- 10 → f_4 and f_{12}
- **Looks quite effective!**

Random Test Coverage

w_1w_2	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
00	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
01	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
10	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
11	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

- Each test would rule out the faulty functions whose outputs are different from the correct function
- If I choose 01, the correct circuit produces 1; the faulty eight functions producing 0, i.e., f0, f1, f2, f3, f8, f9, f10, f11, should be ruled out

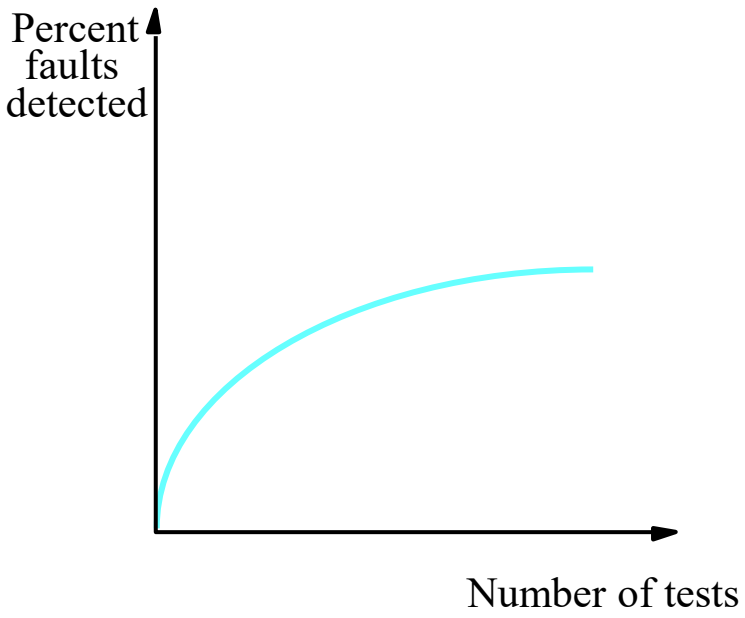
Random Test Coverage

Prob that each faulty circuit can be detected by the first test set

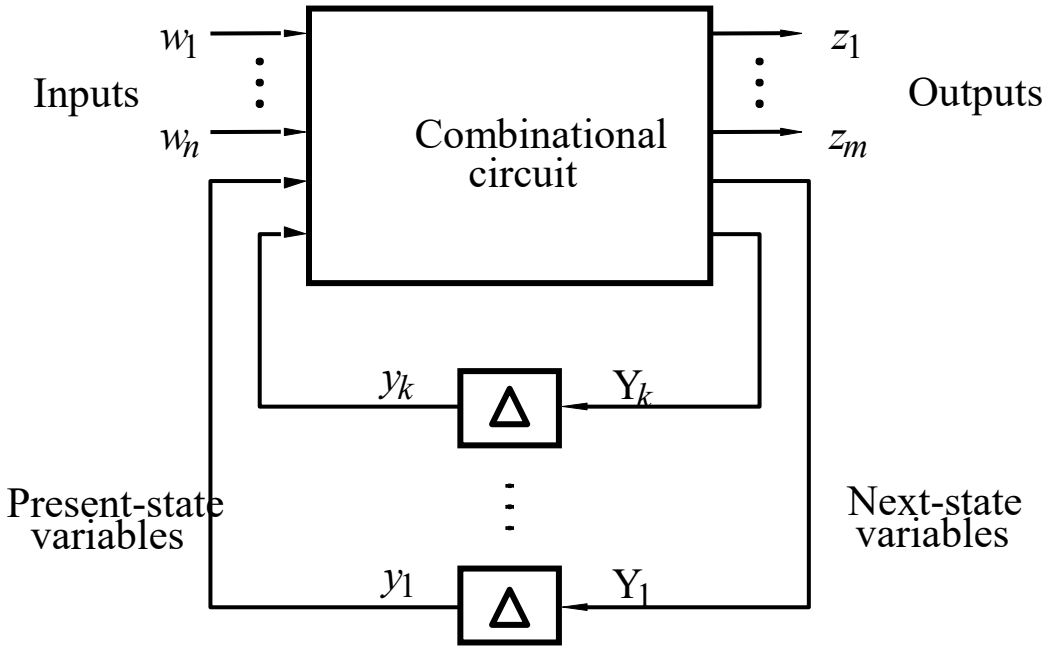
$$P_1 = \frac{1}{2^{2^2}-1} \cdot 2^{2^2-1} = \frac{8}{15} = 0.53$$

Probably that a faulty circuit will be detected by the first m tests

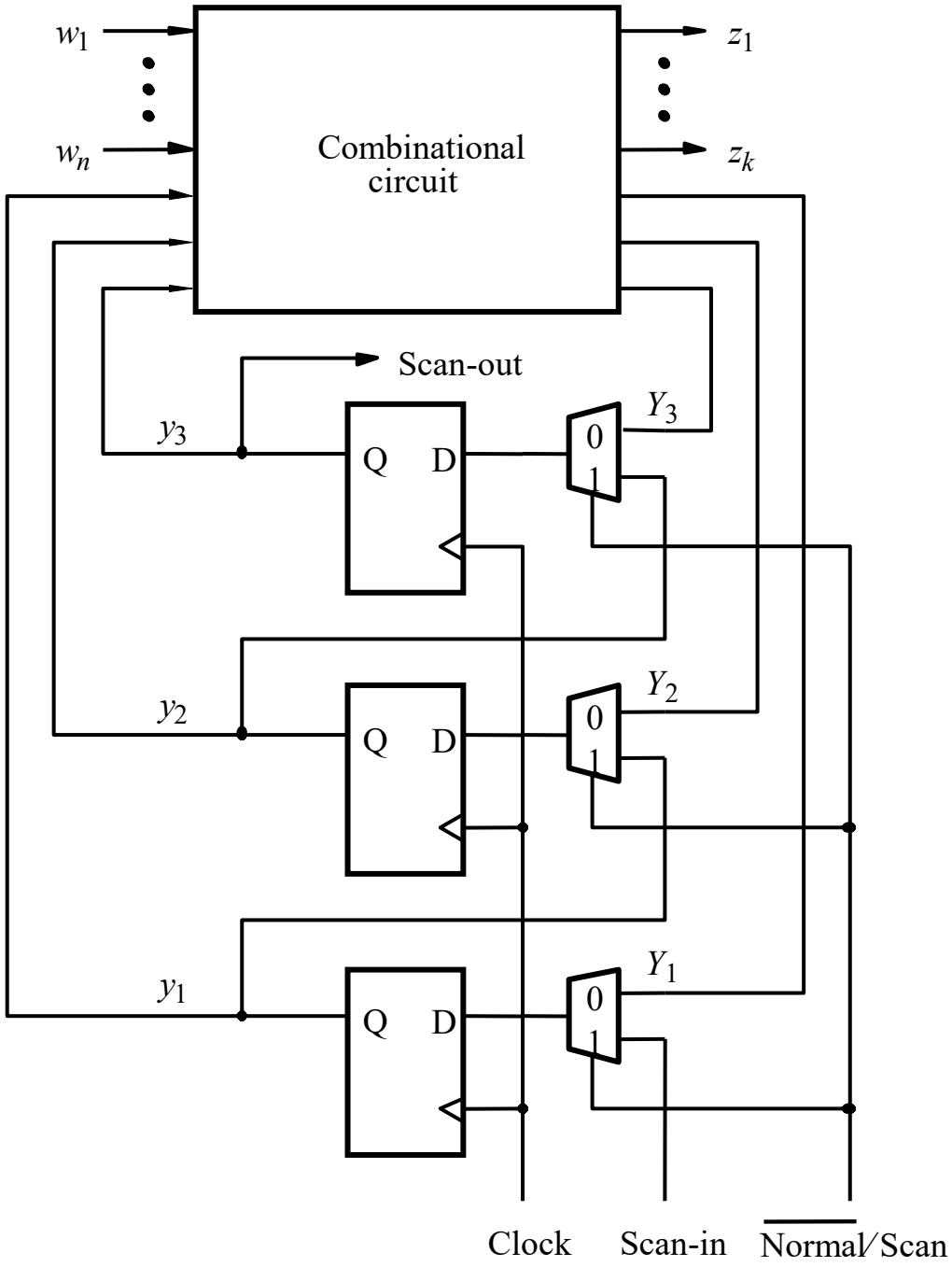
$$P_m = \frac{1}{2^{2^n}-1} \cdot \sum_{i=1}^{m} 2^{2^n-i}$$



Sequential Circuit Testing



- Testing a sequential circuit is much difficult
  - Function is now dependent on states as well
  - It is hard to monitor and assert states because we usually don't have direct access (pins)



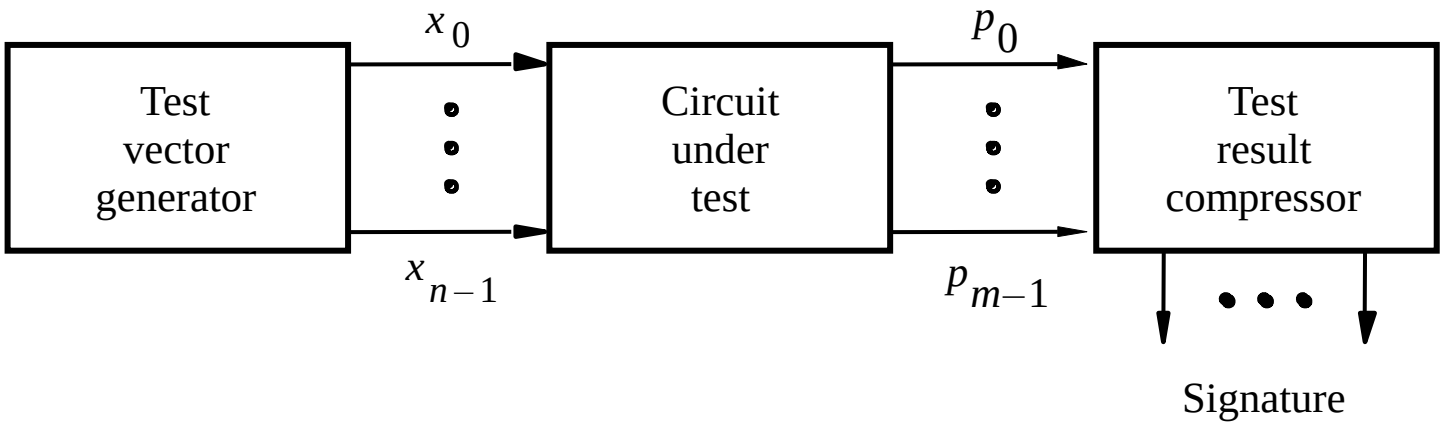
Scan Path

- Add Muxes to the inputs of flip-flops such that the flip-flops can be converted into a shift register

Scan Path

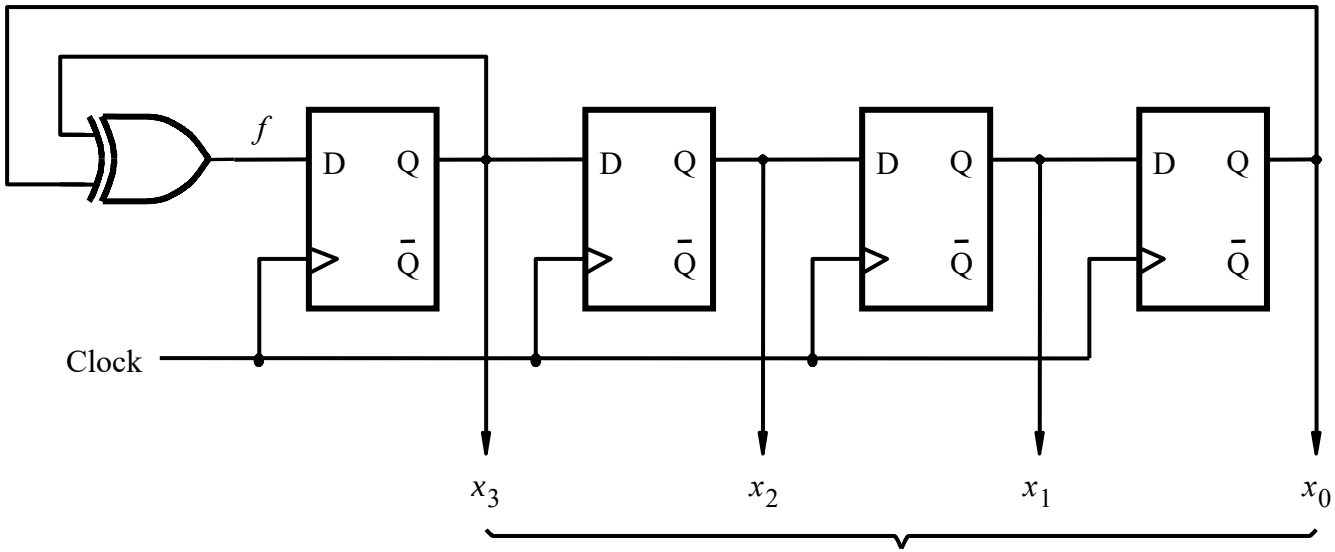
- Typical testing sequences
  1. Operation of flip-flops are tested
  2. Combinational circuit is tested
  3. Set a state and test the state transitions
- The scan path can’t effectively handle if an asynchronous reset/set feature is used during normal operation
- Better to use synchronous set/reset

Built-In Self-Test



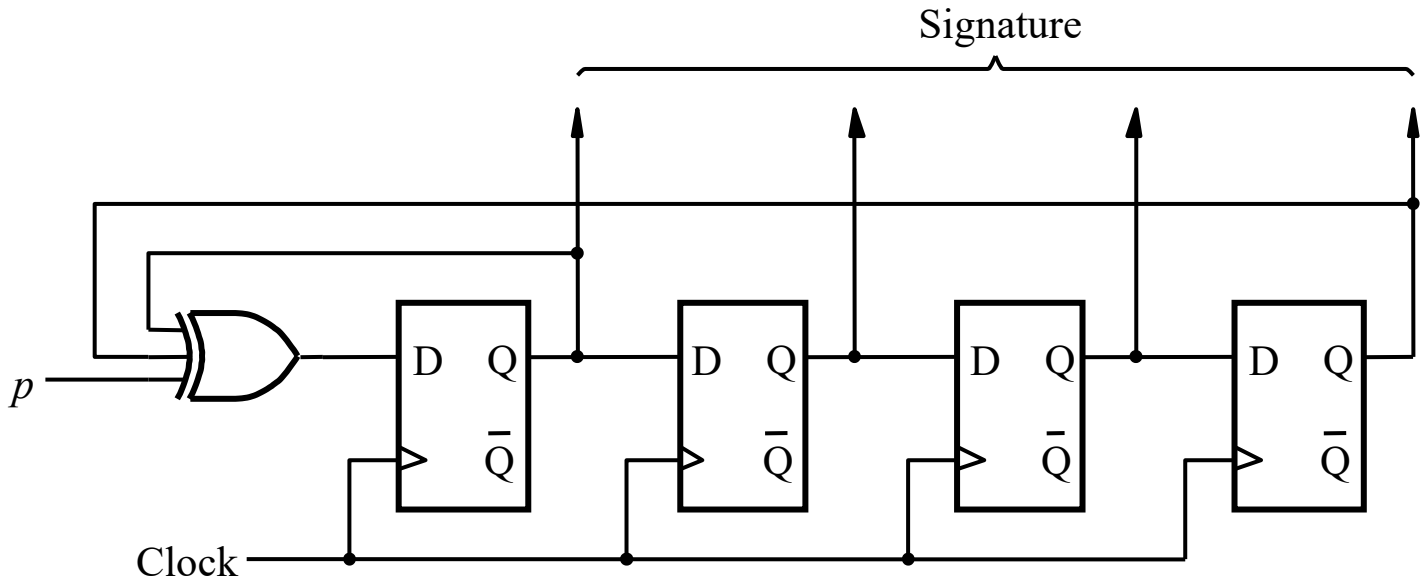
- Incorporate the testing capability within the circuit itself so that no external equipment is needed
- Avoid slow and noisy off-chip communication

PRBSG



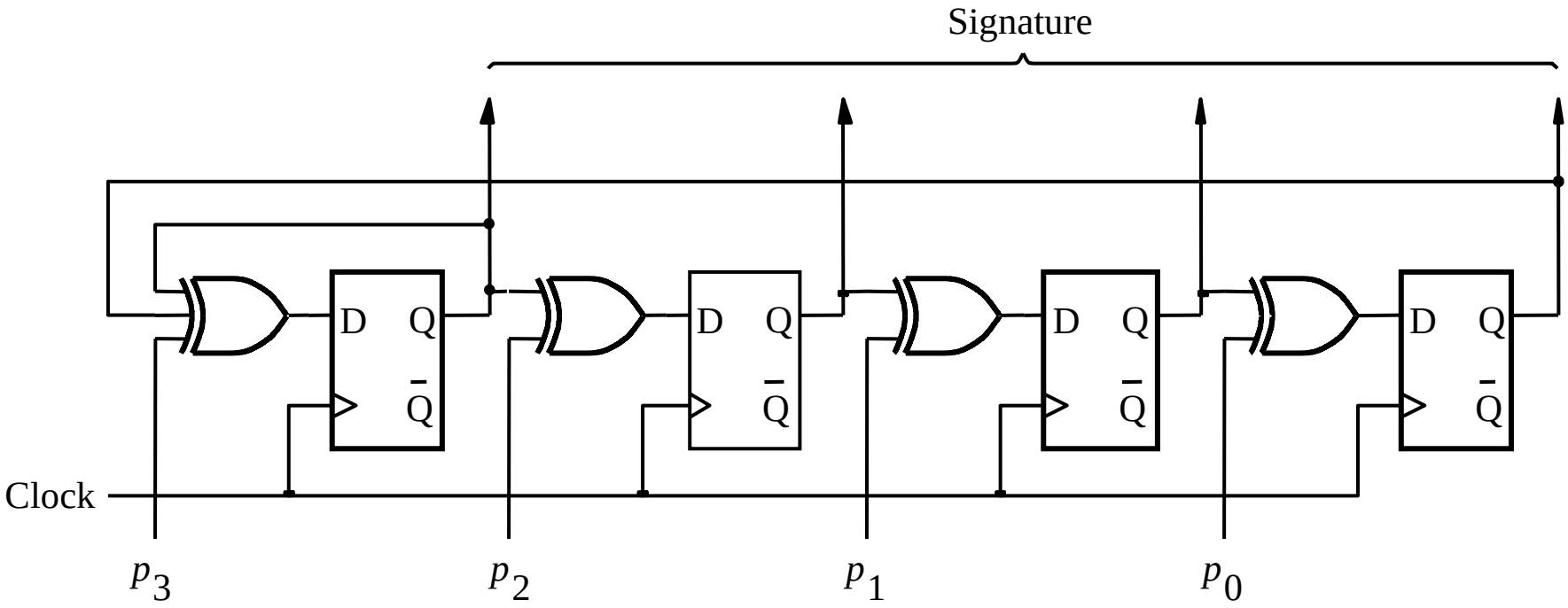
- Pseudorandom binary sequence generator
- Linear feedback shift register: can generate 2^n-1 patterns

SIC



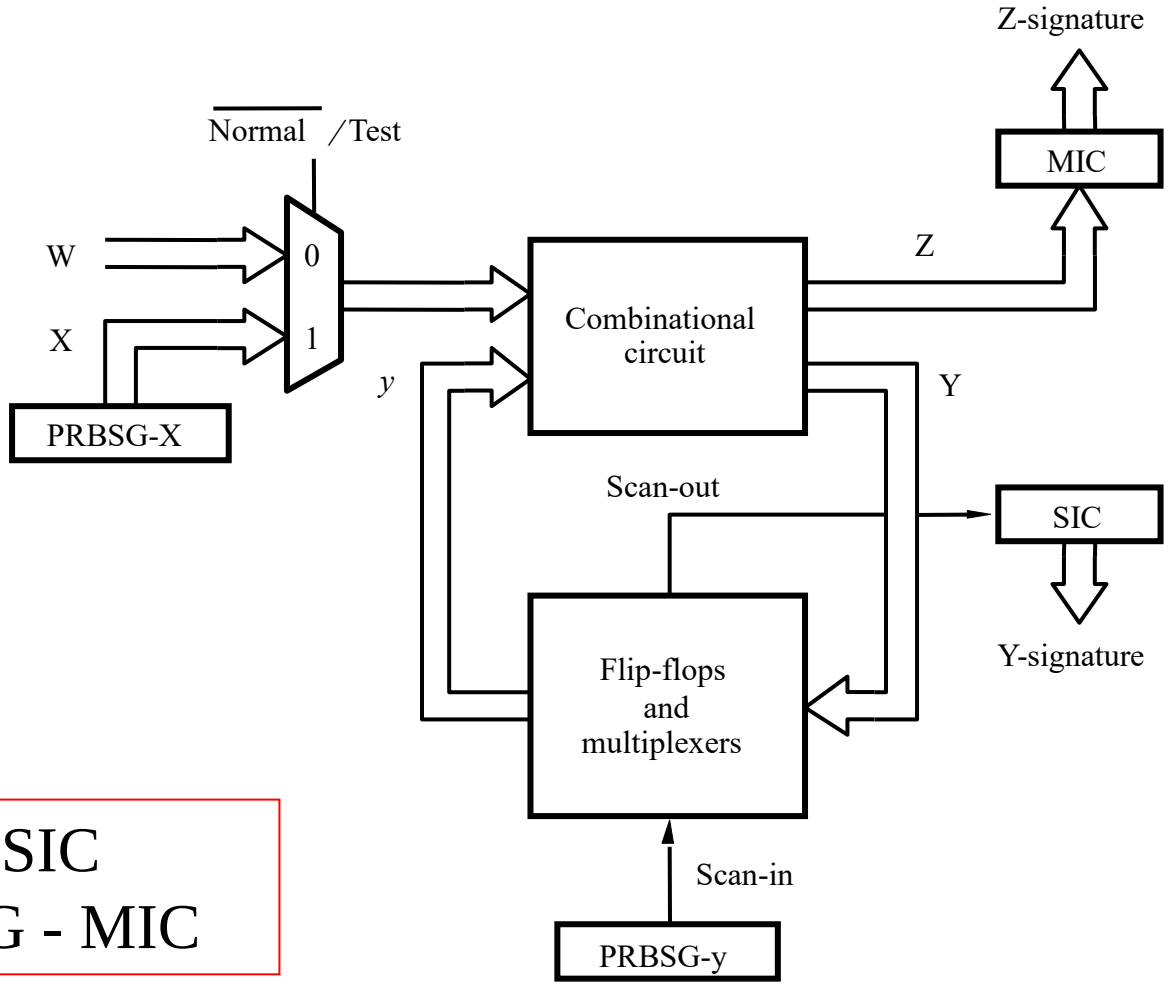
- Single input compressor
- Compress a sequence of patterns into a signature
- A change of one pattern creates different signature

MIC



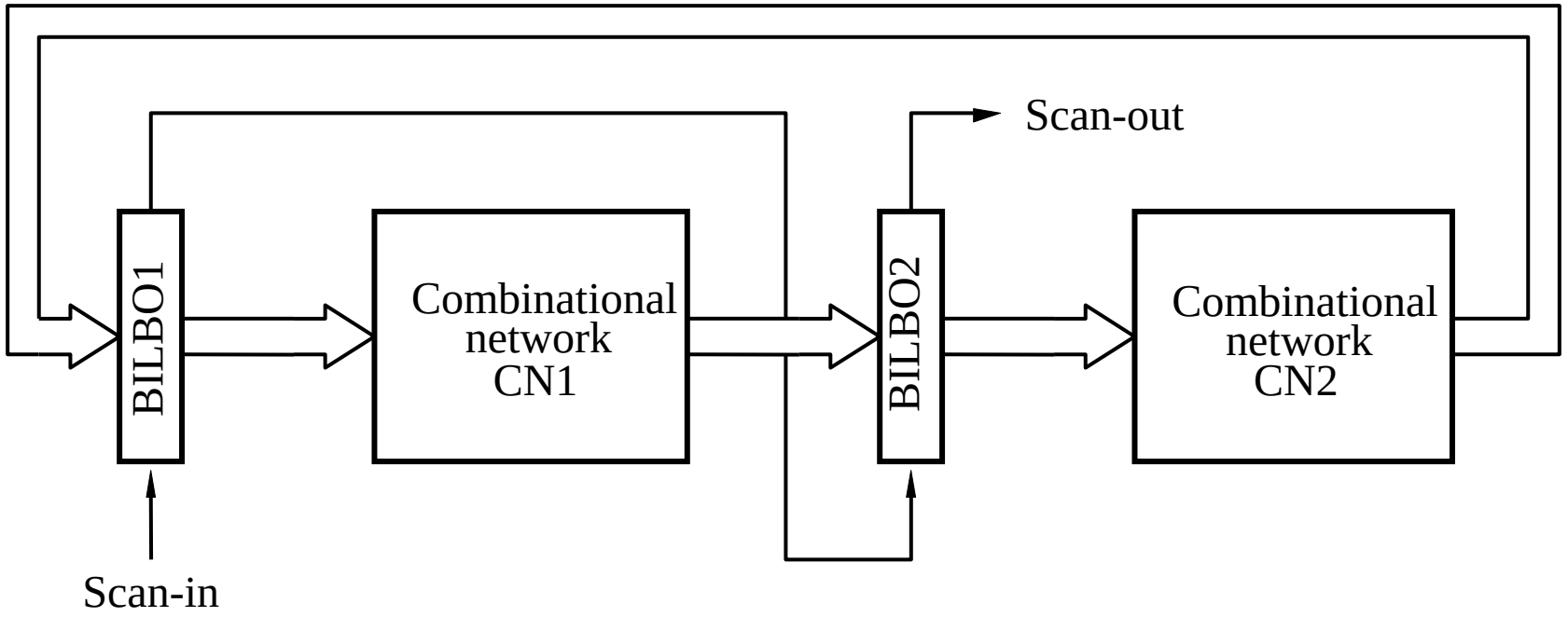
- Multiple input compressor
- Compress a sequence of multi-bit patterns into a signature

BIST Architecture



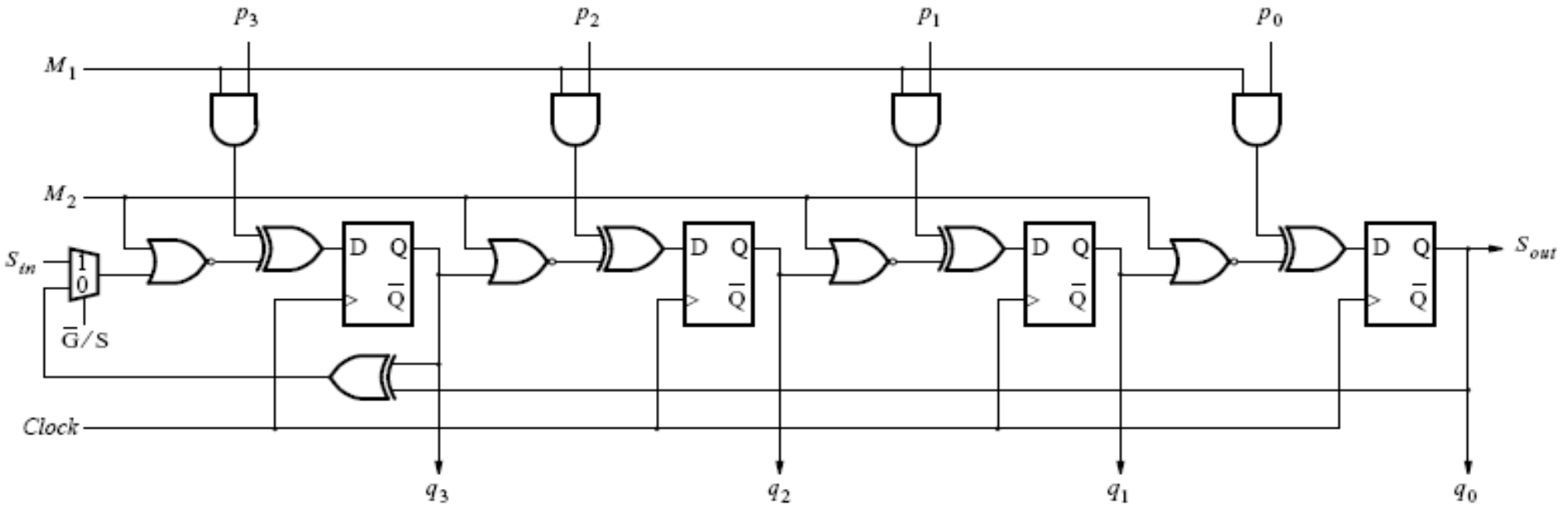
Scan - SIC
PRBSG - MIC

BILBO



- Built-in Logic Block Observer (BILBO)
- BILBO can be configured to PRBSG, MIC, or regular registers

BILBO



- M_1M_2=11: normal
- 00: shift register mode; if G_bar/S=0, it acts like PRBS
- 10: signature mode
- 01: reset mode

Boundary Scan

- Chips are soldered onto a printed circuit board, it often becomes impossible to physically attach test probes to such pins
- The scan path concept is extended so that we can scan-in / -out to the primary inputs and outputs of a chip
- Assume each primary input/output is clocked (i.e., registered)
- IEEE Standard 1149.1 for the protocol

Printed Circuit Board Testing

- Crosstalk
- Power supply noise
- Reflections and terminations

- Power-up
- Functional test
- Timing
- Reliability