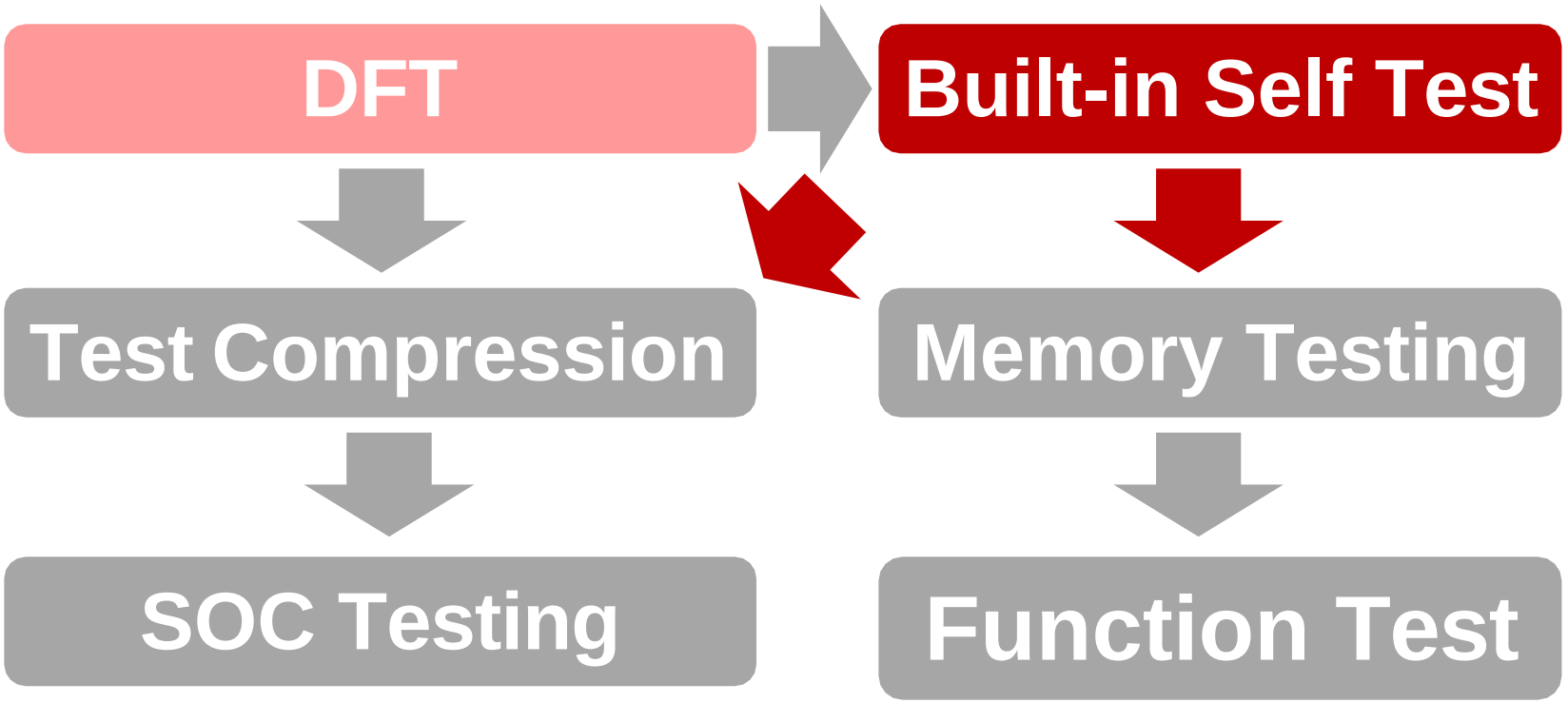
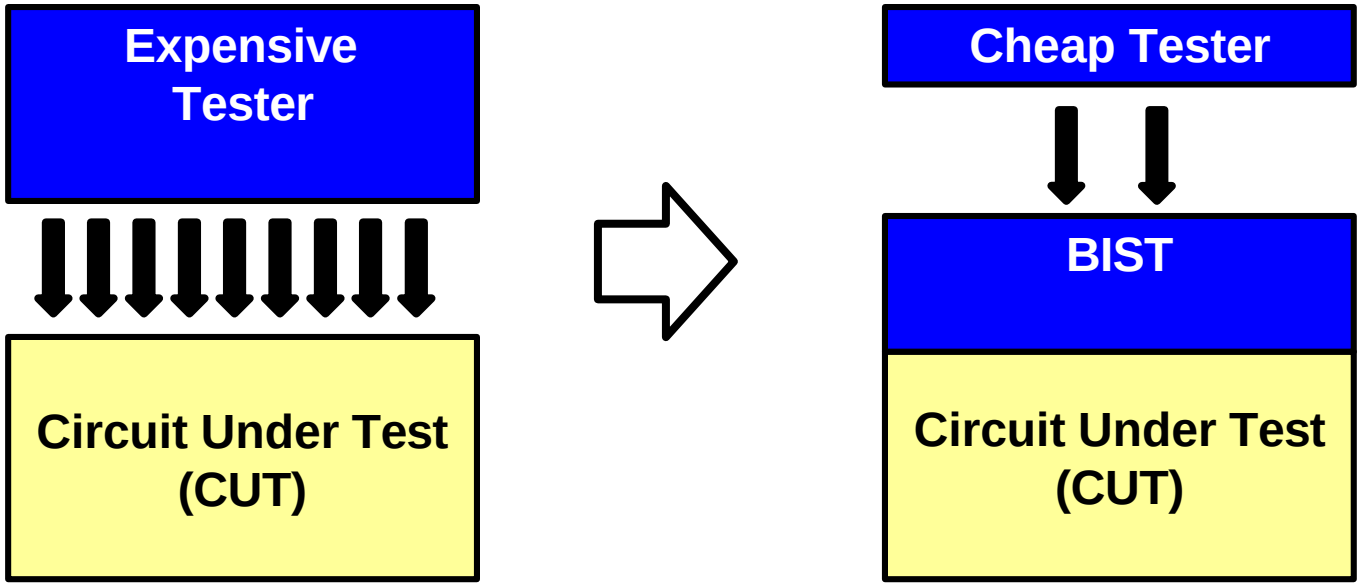
Logic Built-In Self Test (BIST)

Course Roadmap (Design Topics)



Motivating Problem

- Your manger complains tester is too expensive.
- Please design a circuit to test the chip itself!

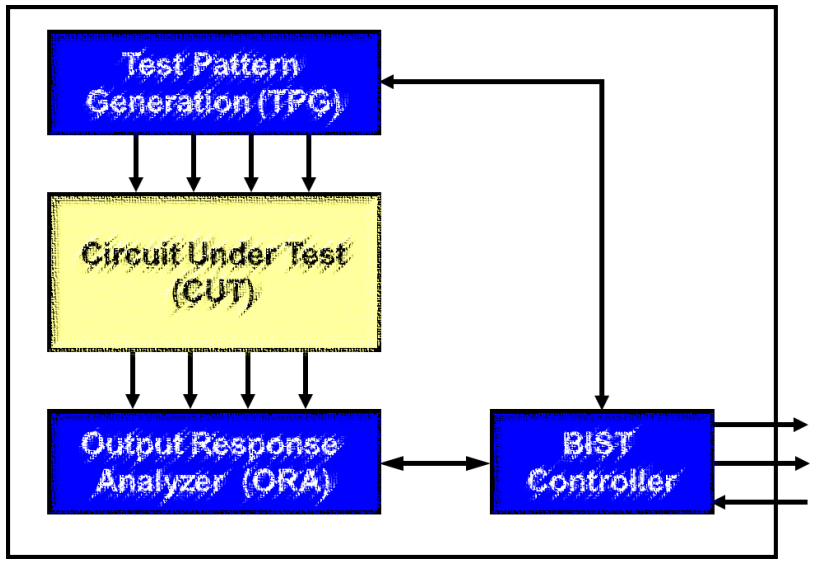


Why Am I Learning This?

- **Built-in Self Test (BIST) can**
  - **Remove large expensive testers**
  - **Test chips at high speed**
  - **Test chips online**

BIST Outline

- Part 1
  - Introduction
  - Test Pattern Generation
- Part 2
  - Output Response Analysis
  - BIST Architecture
  - Problems and solutions
  - Conclusions



Built-in Self Test

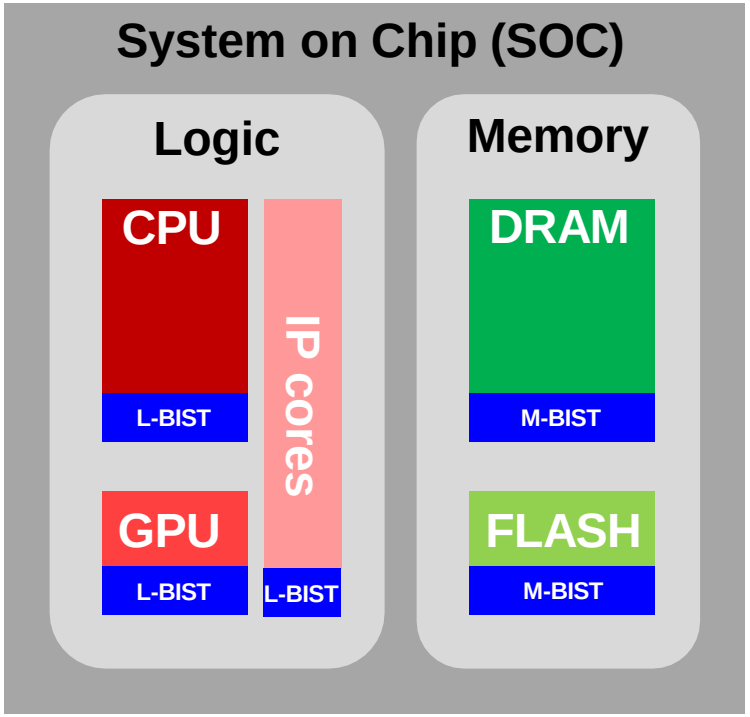
- Definition
  - Capability of hardware/software to carry out explicit test of itself
- Levels of BIST
  - System-level self test
    - system self test of mainframe computer
  - Board-level self test
    - so we can replace a bad board in a system
  - Chip-level self test
    - focus of this lecture

123rf.com

IBM Blue Gene

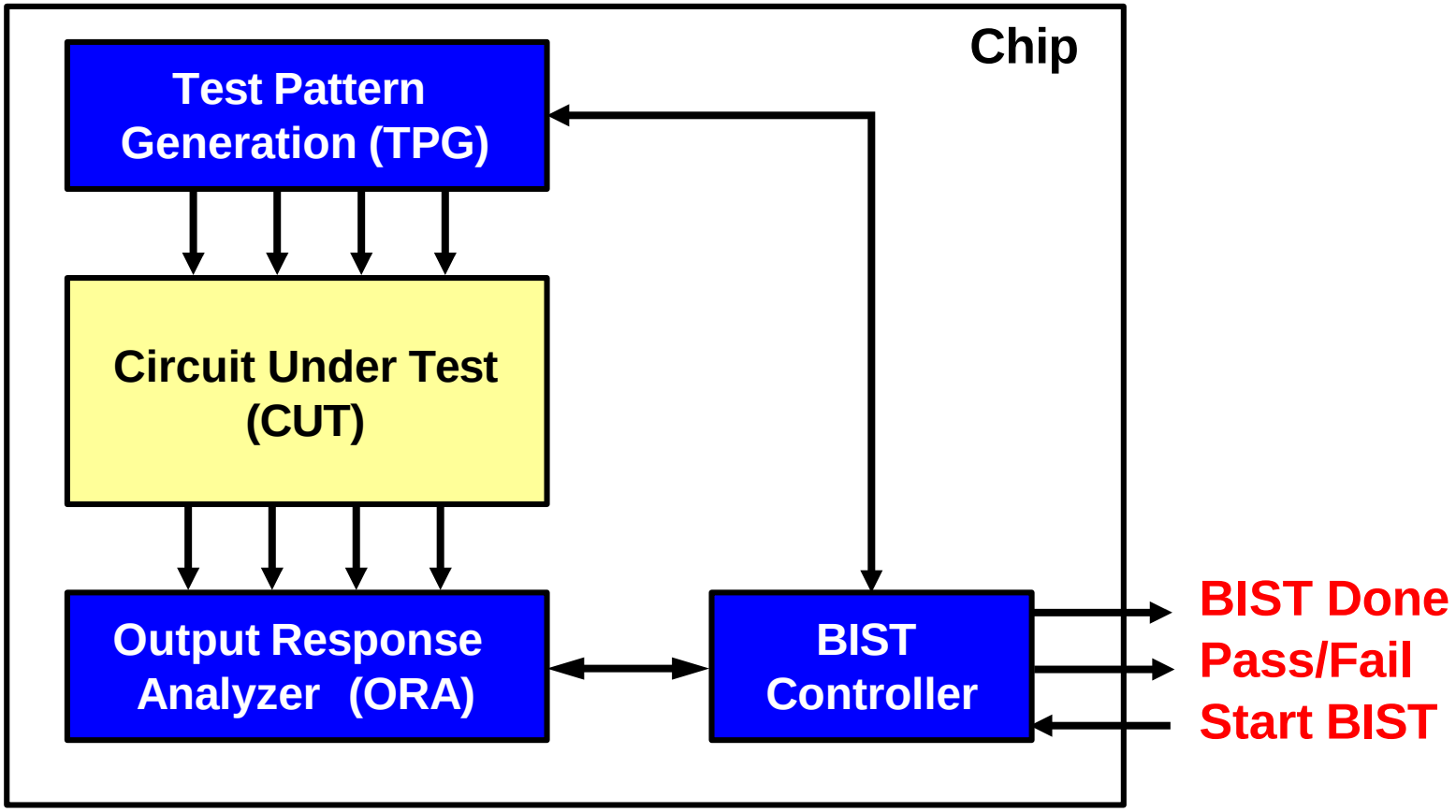
Categories of Chip-level BIST

- Based time of test
  - *On-line BIST*
    - On-line testing while chip in normal operation
    - e.g. error detection and correction for RAM
  - *Off-line BIST*
    - Off-line testing while chip not in normal operation
    - Focus of this talk
- Based on CUT
  - Logic BIST
  - Memory BIST



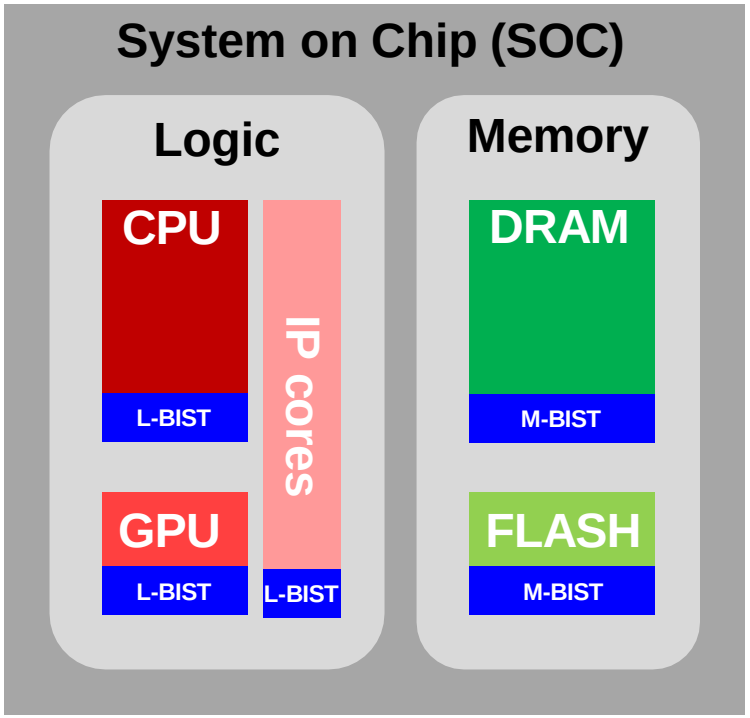
Architecture of BIST

- Three components: ***BIST Controller, TPG, ORA***
- Three I/O Pins: ***Start BIST, BIST Done, Pass/Fail***



Why BIST?

- 1. Save ATE cost
  - Smaller test pattern storage
  - Fewer DFT pins
  - Slower tester speed
- 2. Better IC quality
  - Test speed higher than ATE
- 3. Easier integration of tests
  - *Intellectual Property* (IP) cores
- 4. Easier test access
  - Test embedded memory in SOC
- 5. Enable on-line testing
  - Ensure reliability



BIST Has Many Unique Advantages

ATE Cost w/wo BIST

- Consider a 1 GHz μP with 1000 pins.

- Test w/o BIST
  - 1 GHz ATE: $3,000/pin×1,000pins = $3,000,000 USD
  - Huge initial capital cost
- Test w/ BIST
  - 20 MHz ATE: $400/pin×1,000pins = $400,000 USD
    - Test I/O pad contact
    - Provide test commands, and read out test results

BIST Reduced ATE Cost

Disadvantages of BIST

- 1. Area Overhead
  - Yield loss due to BIST circuitry
- 2. Performance degradation
  - Extra hardware
- 3. Extra design effort
  - Test point insertion, BIST insertion, verification …
- 4. Lack information for debug and diagnosis
  - Need to bypass BIST when diagnosis
- 5. Long test length but fault coverage may not good enough
  - Random test patterns not as good as ATPG patterns
  - Mixed solution (BIST + ATE) is often needed

BIST can NOT Solve All Problems

Quiz

Q: Which of the following is NOT true for BIST?

A. BIST reduces test cost because of good fault coverage

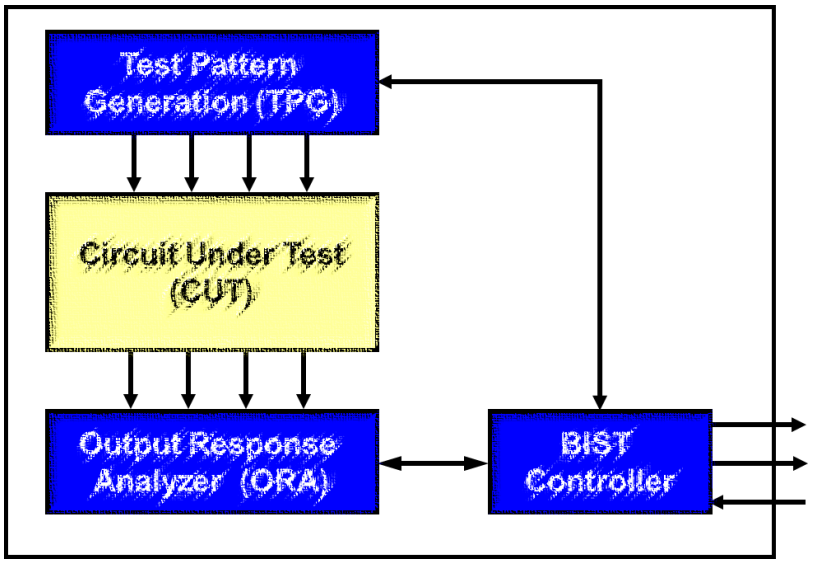
B. BIST reduces DFT pins required

C. BIST improve test quality because test speed is higher

ANS:

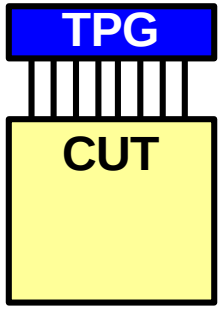
BIST Part1 - TPG

- Introduction
- **Test Pattern Generation (TPG)**
  - **Deterministic: ROM, Algorithm, Counter**
  - Pseudo Random:
    - Linear Feedback Shift Register (LFSR)
    - Cellular Automata (CA)



ROM/Algorithm as TPG

- **ROM as TPG**
  - **Store test patterns in ROM**
  - **Very expensive for chip-level BIST**
  - **Maybe doable for system-level BIST**
    - **e.g. self test program in BIOS (Basic Input/Output System)**



- **Algorithm as TPG**
  - **Test pattern generation based on certain mathematical rule**
  - **Suitable for regular structure like memory, FPGA**
    - **Not very useful for random logic**

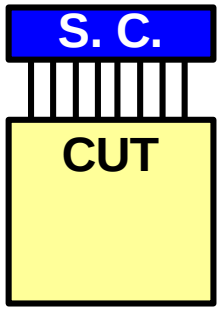
No Good for Logic BIST

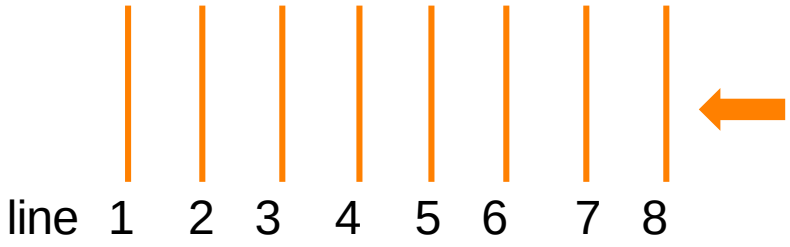
Shift Counters as TPG

- Generates regular test sequences
  - Such as walking sequence for interconnect testing
- Advantage: Linear test time
- Disadvantage: Too regular, not useful for random logic

Walking Sequence

cycle								
1	1	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0
4	0	0	0	1	0	0	0	0
5	0	0	0	0	1	0	0	0
6	0	0	0	0	0	1	0	0
7	0	0	0	0	0	0	1	0
8	0	0	0	0	0	0	0	1

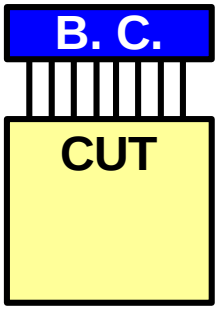




Bridging faults between lines can be detected

Binary Counters as TPG

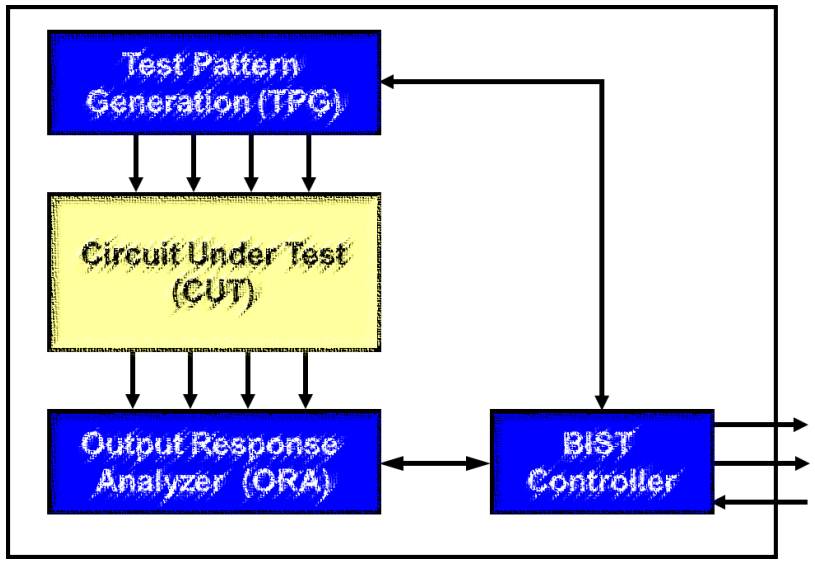
- N-stage binary counters
  - Generate test patterns in sequence
    - from 0 to 2^N-1, or from 2^N-1 to 0
- Advantage
  - Simple design
  - Exhaustive test is high quality
- Disadvantages
  - Exhaustive test length is very long
    - ⇒ No randomness, deterministic pattern sequence
    - ⇒ e.g. Need 2^{N-1} test length to reach a 1 at MSB
  - Large area overhead (compared to LFSR)



Need Random Counter for Logic BIST

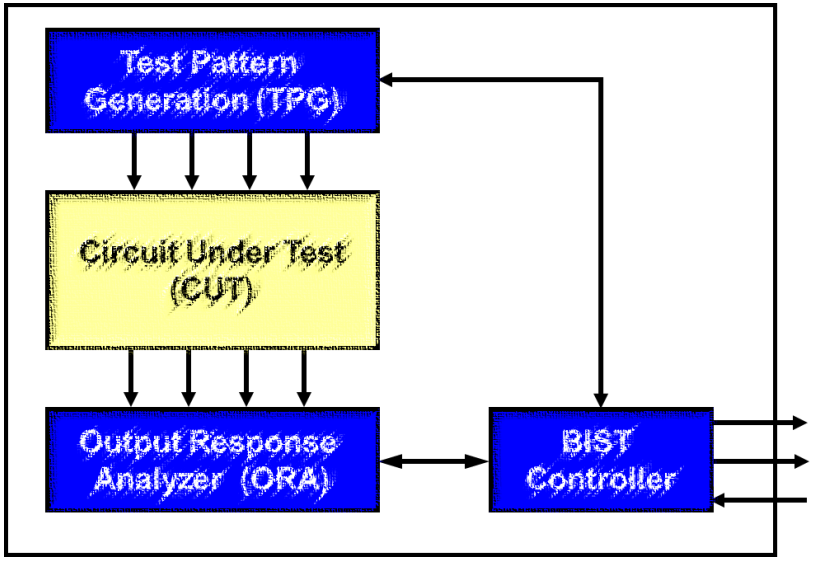
Summary

- Introduction
  - BSIT components: TPG, ORA, and controller
  - Pros: reduced pins, tester cost, on-line testing
  - Cons: area overhead, performance degradation, lower FC
- Test Pattern Generation
  - Deterministic: ROM, Algorithm, Counter



BIST Part1 - TPG

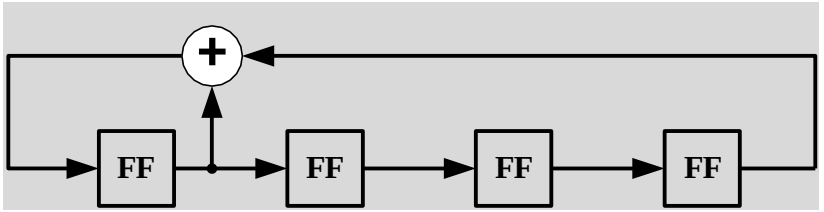
- Introduction
- Test Pattern Generation (TPG)
  - Deterministic: ROM Algorithm Counter
  - **Pseudo Random**
    - ***Linear Feedback Shift Register, LFSR* (1977)**
      - **Two types of LFSR**
      - Design of LFSR
    - Cellular Automata, CA (1984)



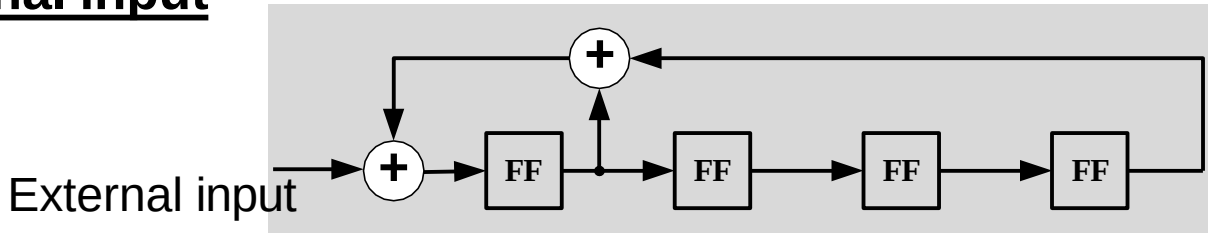
Linear Feedback Shift Register [Frohwerk 77]

- LFSR consist of unit delays (flip-flops, FF) and feedback (XOR)
- Two applications of LFSR:

1. **LFSR without external input**
   - Used for TPG
   - aka *Autonomous LFSR*



2. **LFSR with external input**
   - Used for ORA



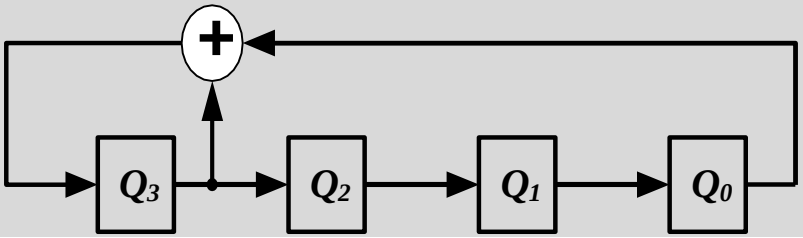
⊕ =XOR

- *"Linear"* because XOR is *mod-2 addition*

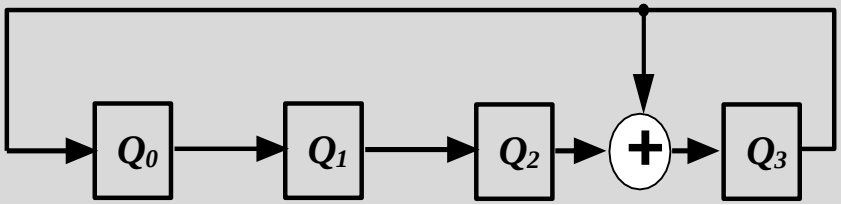
Two Types of Autonomous LFSR

- **Autonomous LFSR is *Modular Counter***
  - **Very small area, generate *pseudo random* outputs**
- **Two structures:**
  - ***Type 1: Standard Form* (aka *external XOR*) LFSR**
  - ***Type 2: Modular Form* (aka *internal XOR*) LFSR**

Type-1 Standard LFSR



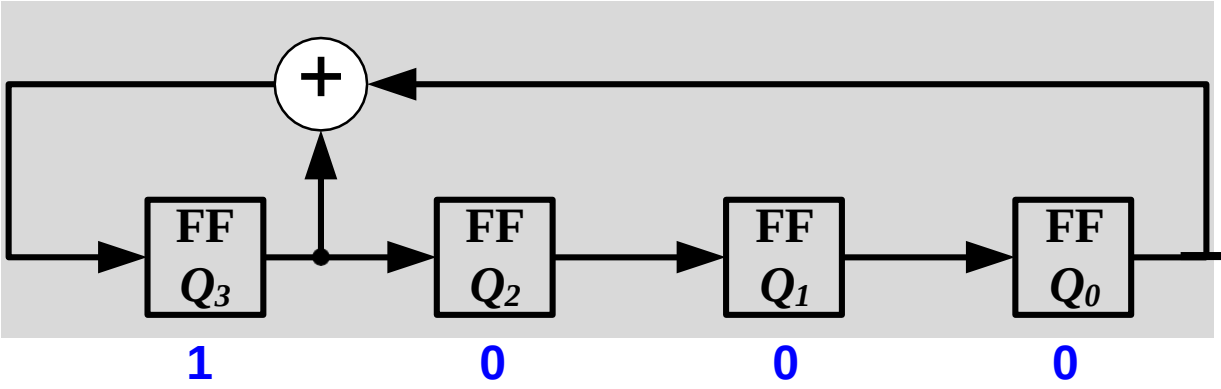
Type-2 Modular LFSR



LFSR is Good for TPG

Type-1, Standard Form LFSR

- **Three ways to describe LFSR**
  1. ***Next state equation*:** $Q_3^+ = Q_3 \oplus Q_0$
     - Q_3^+ means next state of FF Q_3
  2. ***Recurrence equation*:** $q_m = q_{m-1} \oplus q_{m-4}$
  3. ***Characteristic polynomial*:** $f(x) = x^4 + x^3 + 1$
     - **Most popular.**



output bit stream (q_0 is first)

$q_0\ q_1\ q_2\ q_3\ q_4\ q_5\ q_6\ q_7$

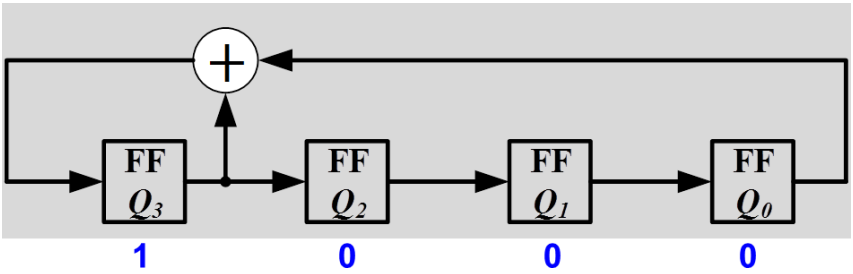
0 0 0 1 1 1 1 0

$q_4 = q_3 \oplus q_0$
$q_5 = q_4 \oplus q_1$
...
$q_m = q_{m-1} \oplus q_{m-4}$

- **NOTE: Modular-2 arithmetic: ⊕ + - are same**
  - **1⊕1=0; 0⊕1=1; 1⊕0=1; 0⊕0=0**

State Sequence of x^4+x^3+1 LFSR

- *Seed* = Initial state of LFSR
  - must be non-zero
- Total $2^4-1 = 15$ distinct states
  - All-zero state not included
- Periodical.
  - Cycle length $L_c = 15$

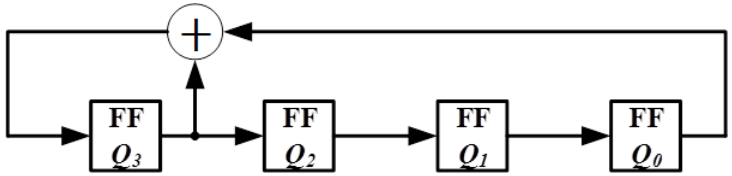


state	Q_3	Q_2	Q_1	Q_0
0	1	0	0	0
1	1	1	0	0
2	1	1	1	0
3	1	1	1	1
4	0	1	1	1
5	1	0	1	1
6	0	1	0	1
7	1	0	1	0
8	1	1	0	1
9	0	1	1	0
10	0	0	1	1
11	1	0	0	1
12	0	1	0	0
13	0	0	1	0
14	0	0	0	1
15 (=0)	1	0	0	0

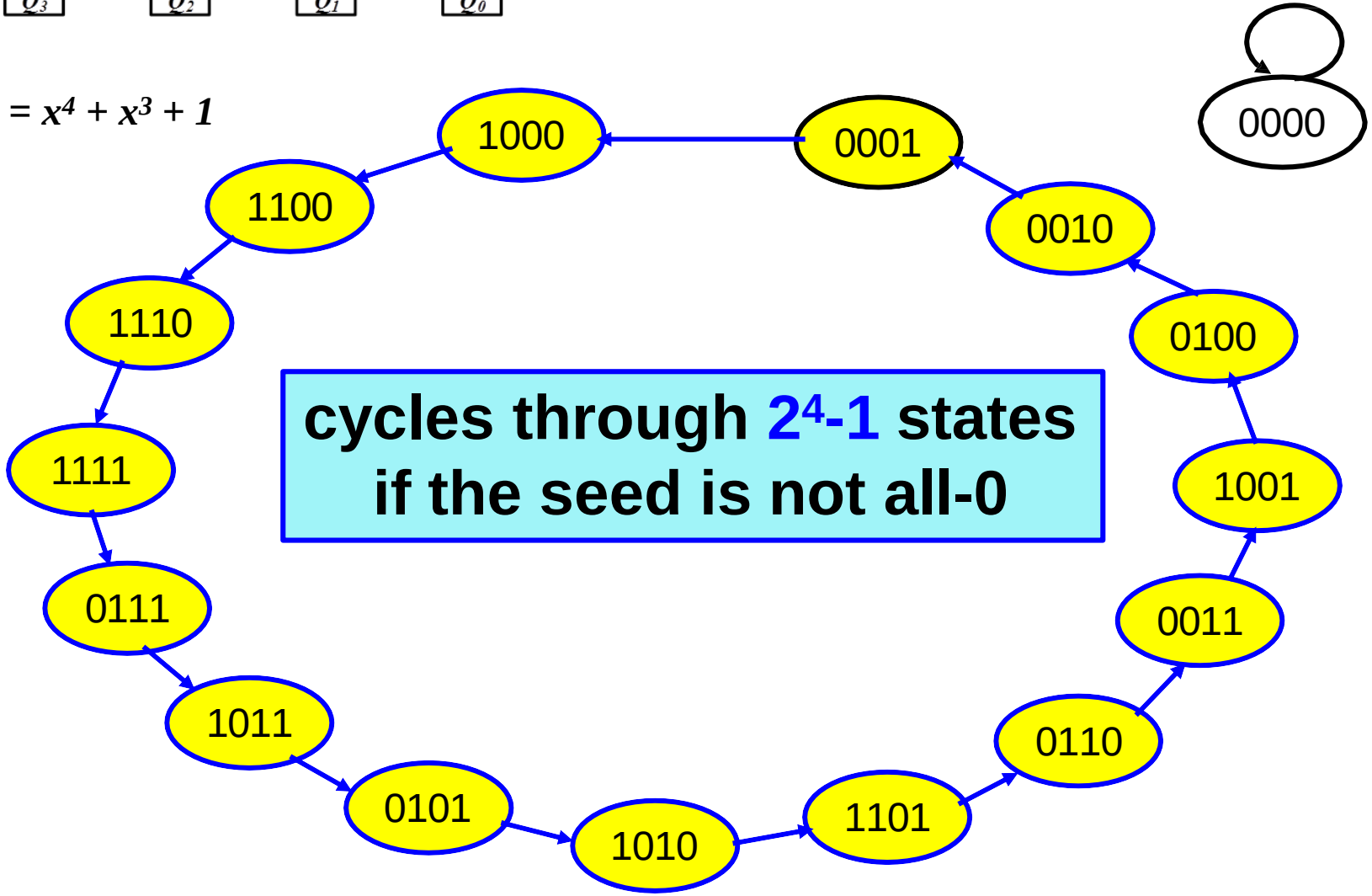
seed

Back to seed after 15 cycles →

State Diagram

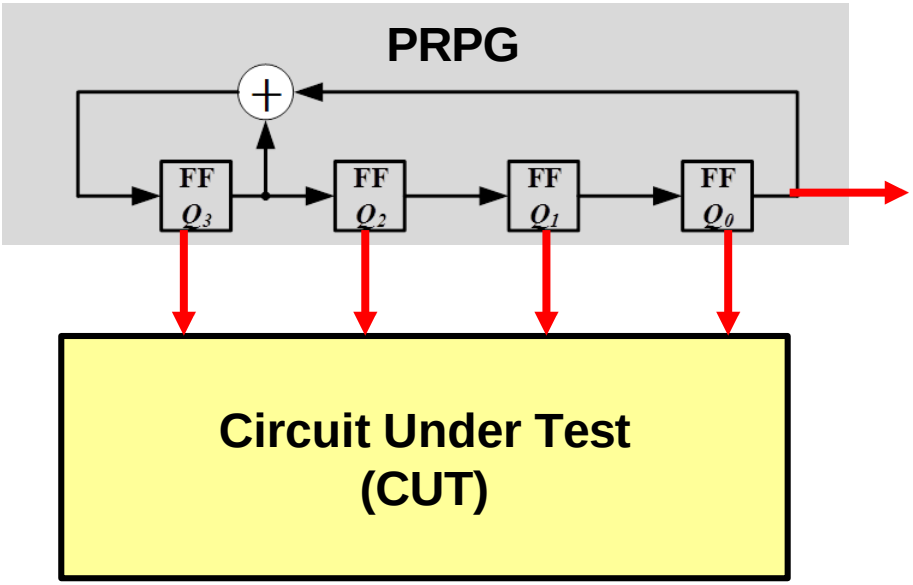


$f(x) = x^4 + x^3 + 1$



Pseudo Random Pattern Generator (PRPG)

- *Pseudo random.* NOT truly random
- Serial PRPG: $q_0, q_1, q_2, q_3, \ldots$
  - Periodical. $L_c = 15$
- Parallel PRPG: $(Q_3\ Q_2\ Q_1\ Q_0)$
  - Each output shifted by one cycle
  - *Phase difference* =1



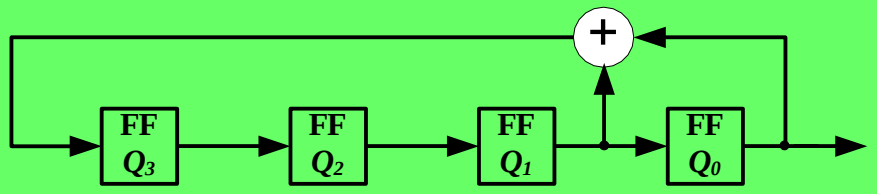
state	Q_3	Q_2	Q_1	Q_0	
0	1	0	0	0	q_0
1	1	1	0	0	q_1
2	1	1	1	0	q_2
3	1	1	1	1	q_3
4	0	1	1	1	q_4
5	1	0	1	1	q_5
6	0	1	0	1	q_6
7	1	0	1	0	q_7
8	1	1	0	1	q_8
9	0	1	1	0	q_9
10	0	0	1	1	q_{10}
11	1	0	0	1	q_{11}
12	0	1	0	0	q_{12}
13	0	0	1	0	q_{13}
14	0	0	0	1	q_{14}
15 (=0)	1	0	0	0	q_{15}

Phase diff. =1

Quiz

Q: Given this new LFSR, show the state sequences with seed [1 0 0 0]

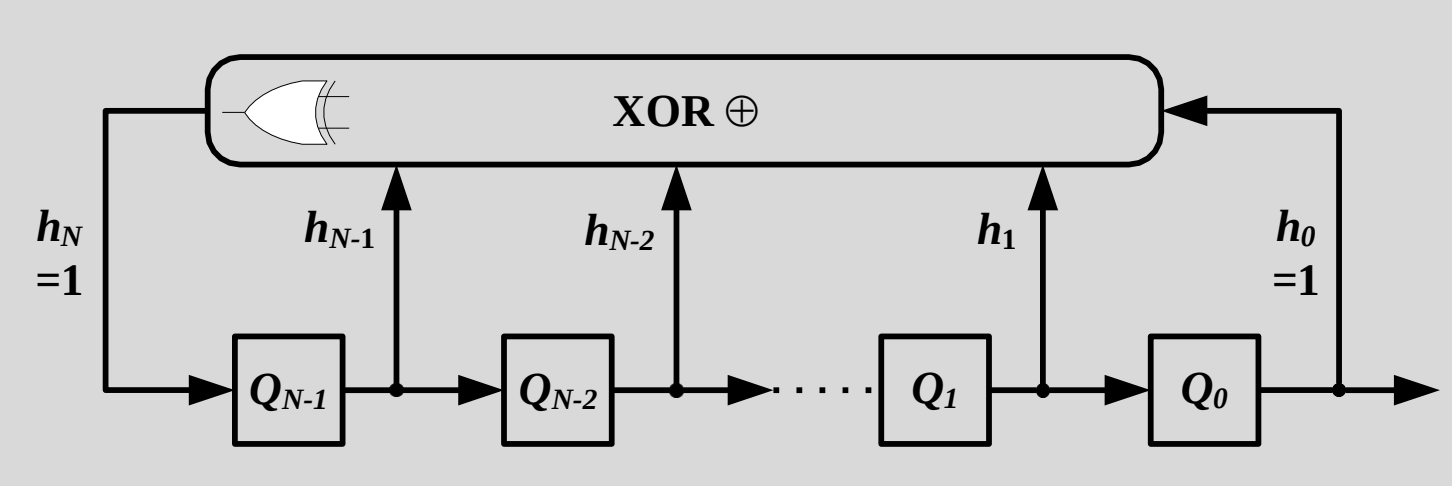
Q2: What is cycle length L_c=?



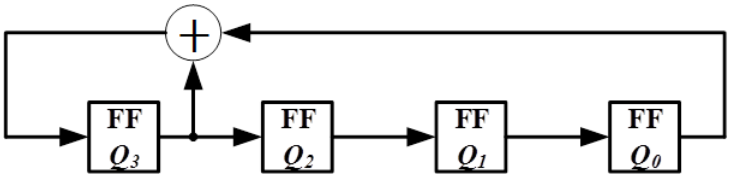
state	Q_3	Q_2	Q_1	Q_0
0	1	0	0	0
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				

N-degree Standard Form LFSR

- ***N = LFSR degree*** **= number of FF = characteristic polynomial degree**



h_i = 1 if feedback exists
h_i = 0 if no feedback
h_N =1, h_0 =1



$f(x) = x^4 + x^3 + 1$

- ***Characteristic polynomial***:
  (details see appendix)

$$f(x) = \sum_{i=0}^{N} h_i x^i$$

***f(x)* Can Be Written from LFSR Structure**

Matrix Representation (Type-1 LFSR)

$$\begin{bmatrix} Q_0^+ \\ Q_1^+ \\ \vdots \\ Q_{N-3}^+ \\ Q_{N-2}^+ \\ Q_{N-1}^+ \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & 0 & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ h_0 & h_1 & h_2 & \cdots & h_{N-2} & h_{N-1} \end{bmatrix} \begin{bmatrix} Q_0 \\ Q_1 \\ \vdots \\ Q_{N-3} \\ Q_{N-2} \\ Q_{N-1} \end{bmatrix}$$

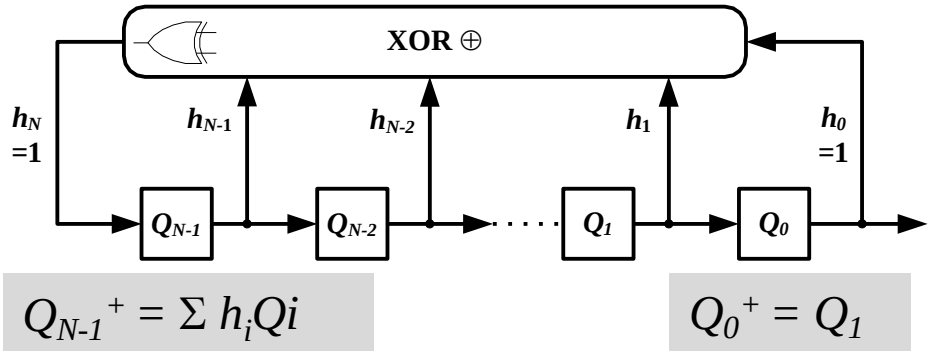
Identity matrix

$Q^+ = T\,Q$ (mod-2)

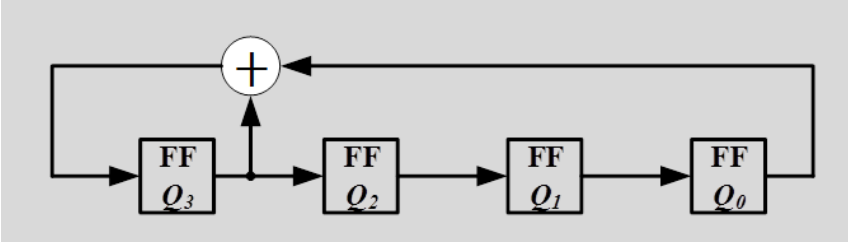
Q = current state

Q^+ = next state

T = companion matrix



Example



state	Q_3	Q_2	Q_1	Q_0
0	1	0	0	0
1	1	1	0	0
2	1	1	1	0
3	1	1	1	1

$$T = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} \qquad T^2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

mod-2 arithmetic:
1x1=1 1x0=0 0x0=0
1+1=0 1+0=1 0+0=0

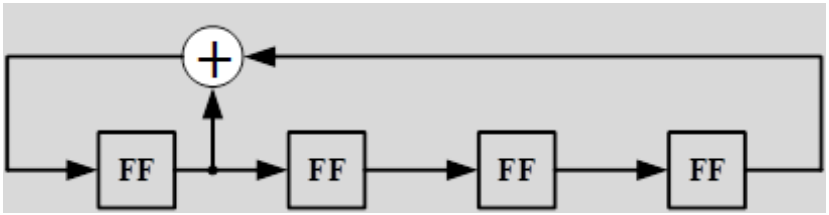
$$Q_{seed} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \qquad TQ_{seed} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \qquad T^2Q_{seed} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$T^3Q_{seed} == T^2TQ_{seed} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Matrix Multiplication Same as Simulation

Linear Relationship between States

- 0_{th} *cycle: LFSR state* = Q_{seed}
- 1_{st} cycle: TQ_{seed}
- 2_{nd} cycle: $TTQ_{seed} = T^2Q_{seed}$
- 4_{th} cycle: $T^4Q_{seed} = (T^2)^2Q_{seed}$
- 16_{th} cycle: $T^{16}Q_{seed} = (T^8)^2Q_{seed} = (((T^2)^2)^2)^2Q_{seed}$
  - only 4 matrix squares needed



- After L_c cycles, LFSR returns to initial state
  - $T^{Lc}Q_{seed} = Q_{seed}$
  - In this case L_c=15. $T^{15} = I$, so $T^{16} = T$

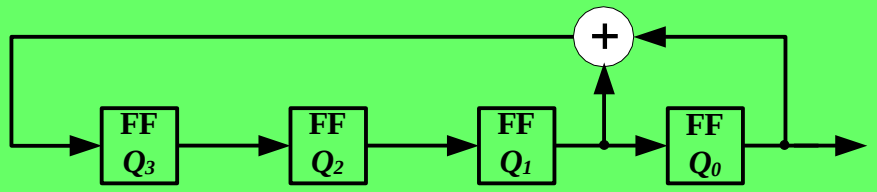
Matrix Faster Than Simulation

Quiz

Q1: Given this LFSR, What is its characteristic polynomial?

Q2: What is its companion matrix *T*?

Q3: Use *T* to derive state after three cycles, starting from seed [1 0 0 0]



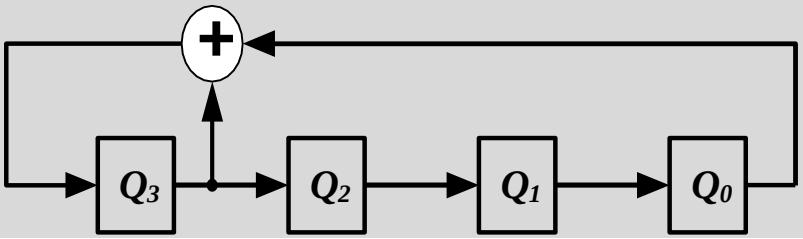
state	Q_3	Q_2	Q_1	Q_0
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	1	0	0	1

ANS

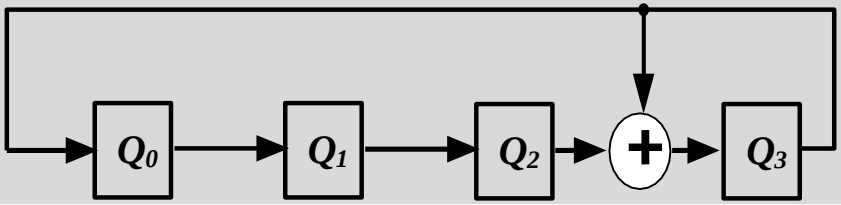
LFSR Types

- *Type 1: Standard Form* (aka *external XOR*) LFSR
- ***Type 2: Modular Form (aka internal XOR) LFSR***
  - **XOR gates are internal to LFSR**
  - **as opposed to standard form LFSR, in which XOR are external**

Type-1 Standard LFSR



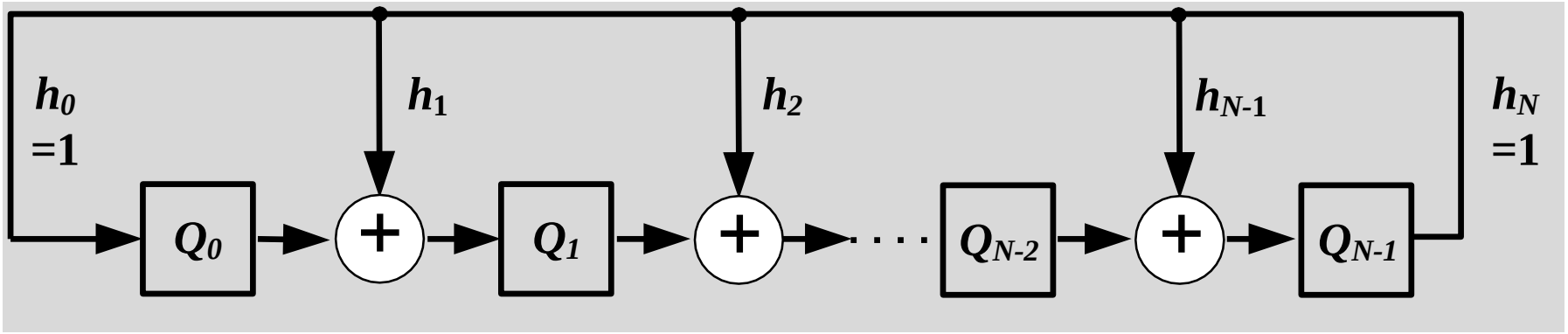
Type-2 Modular LFSR



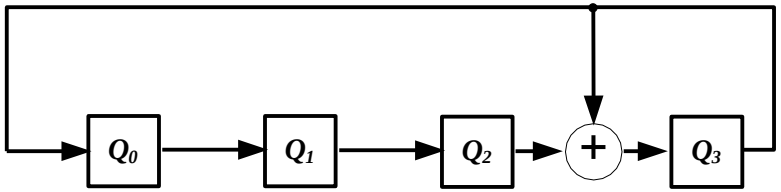
NOTE: Order of FF Different

N-degree Modular Form LFSR

- ***N = LFSR degree*** **= number of FF = characteristic polynomial degree**



h_i = 1 if feedback exists
h_i = 0 if no feedback
h_N =1, h_0 =1



$f(x) = x^4 + x^3 + 1$

- ***Characteristic polynomial*****:** $f(x) = \sum_{i=0}^{N} h_i x^i$

***f(x)* Same as Type-1 LFSR**

(remember correct order of FF !)

Matrix Representation (Type-2 LFSR)

$$\begin{bmatrix} Q_0^+ \\ Q_1^+ \\ \vdots \\ Q_{N-3}^+ \\ Q_{N-2}^+ \\ Q_{N-1}^+ \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & h_0 \\ 1 & 0 & 0 & \dots & 0 & h_1 \\ 0 & 1 & 0 & \dots & 0 & h_2 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 1 & \dots & 0 & h_{N-2} \\ 0 & 0 & 0 & \dots & 1 & h_{N-1} \end{bmatrix} \begin{bmatrix} Q_0 \\ Q_1 \\ \vdots \\ Q_{N-3} \\ Q_{N-2} \\ Q_{N-1} \end{bmatrix}$$

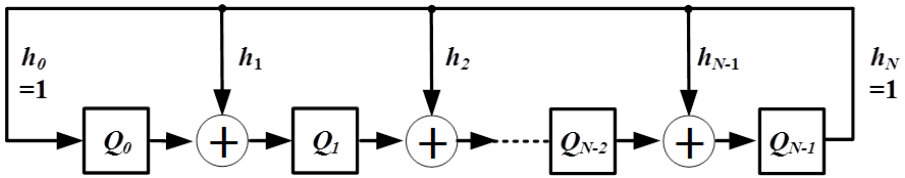
Identity matrix

$Q^+ = T\,Q$

Q = current state

Q^+ = next state

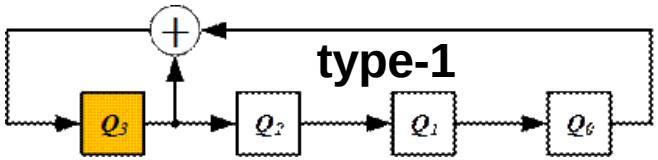
T = Companion matrix

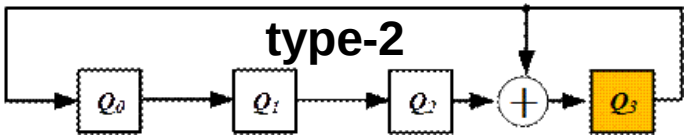


$$Q_1^+ = Q_0 + h_1 Q_{N-1}$$

Quiz

Q: Given two types LFSR of x^4+x^3+1, same seed 1000, fill in table.



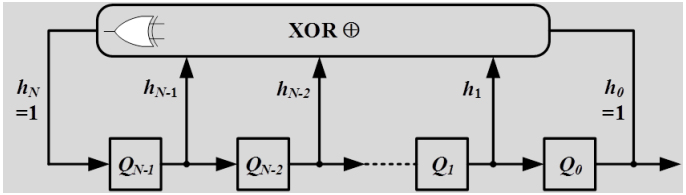
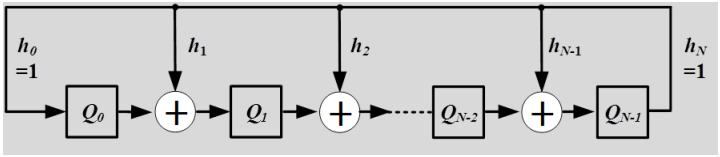


state	Q_3	Q_2	Q_1	Q_0
0	1	0	0	0
1	1	1	0	0
2	1	1	1	0
3	1	1	1	1
4	0	1	1	1
5	1	0	1	1
6	0	1	0	1
7	1	0	1	0
8	1	1	0	1
9	0	1	1	0
10	0	0	1	1
11	1	0	0	1
12	0	1	0	0
13	0	0	1	0
14	0	0	0	1
15 (=0)	1	0	0	0

state	Q_0	Q_1	Q_2	Q_3
0	1	0	0	0
1		1		
2			1	
3				1
4	1			1
5	1	1		1
6	1	1	1	1
7	1	1	1	
8		1	1	1
9	1		1	
10		1		1
11	1		1	1
12	1	1		
13		1	1	
14			1	1
15 (=0)	1			

Summary – LFSR

- **Two types** of LFSR. **Simple** structure, good for **TPG**
- LFSR generate **pseudo random test patterns.** Repeat every L_c cycles
- Linear relationship between states: $Q^+=TQ$
- **Characteristic polynomial** is often used to describe LFSR

	Structure	Matrix
type1 **standard-form**	 index right to left	$\begin{bmatrix} Q_0^+ \\ Q_1^+ \\ \vdots \\ Q_{N-3}^+ \\ Q_{N-2}^+ \\ Q_{N-1}^+ \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & 0 & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \\ h_0 & h_1 & h_2 & \dots & h_{N-2} & h_{N-1} \end{bmatrix} \begin{bmatrix} Q_0 \\ Q_1 \\ \vdots \\ Q_{N-3} \\ Q_{N-2} \\ Q_{N-1} \end{bmatrix}$
type2 **modular-form**	 index left to right	$\begin{bmatrix} Q_0^+ \\ Q_1^+ \\ \vdots \\ Q_{N-3}^+ \\ Q_{N-2}^+ \\ Q_{N-1}^+ \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & h_0 \\ 1 & 0 & 0 & \dots & 0 & h_1 \\ \vdots & 1 & \vdots & & \vdots & h_2 \\ 0 & 0 & 1 & \dots & 0 & \vdots \\ 0 & 0 & 0 & \dots & 0 & h_{N-2} \\ 0 & 0 & 0 & \dots & 1 & h_{N-1} \end{bmatrix} \begin{bmatrix} Q_0 \\ Q_1 \\ \vdots \\ Q_{N-3} \\ Q_{N-2} \\ Q_{N-1} \end{bmatrix}$
Char. Poly	$f(x)=\sum_{i=0}^{N} h_i x^i$	$f(\lambda)=\det(T-\lambda I)$

Reference

- **(BMS 87) P.H. Bardell, W.H. McAnney, J. Savior, *Built-in Test for VLSI: Pseudorandom Techniques*, Wiely Interscience, 1987.**

- **[Könemann 91] B Könemann, "LFSR-coded test patterns for scan designs," European Test Conference, 1991**