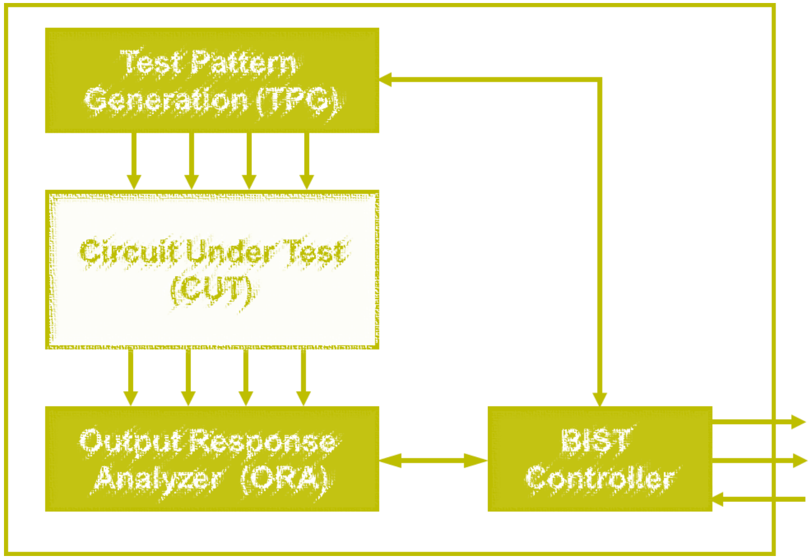
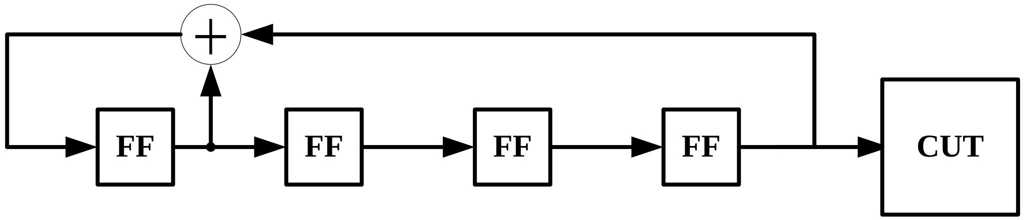
BIST Part 2

- **Output Response Analysis**
  - Simple ORA
  - **LFSR-based ORA**
    - **Serial : compress one bit at a time**
      - **CRC Theory**
      - **PAL analysis**
      - **How to design LFSR as ORA?**
    - Parallel : compress multiple bits at a time
- BIST Architecture
- Issues with BIST
- Conclusions

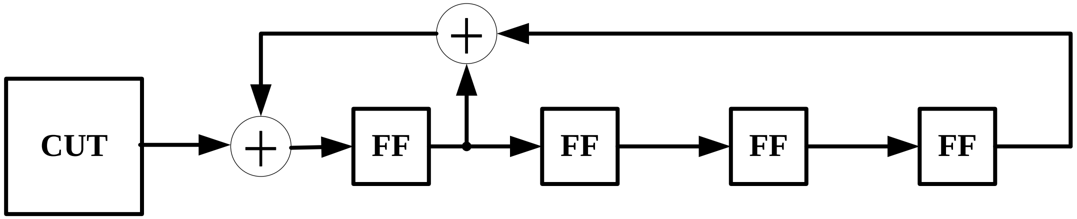


LFSR (Review)

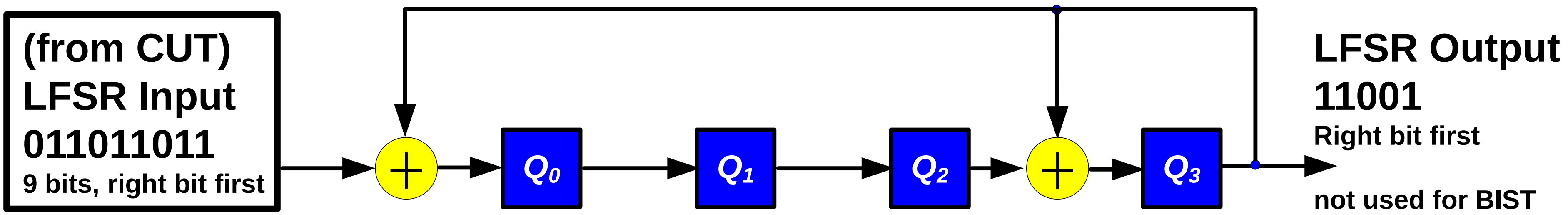
- LFSR consist of FF and feedback XOR
- Two applications of LFSR:
  - 1. LFSR without external input
    - Used for TPG



  - 2. LFSR with external input
    - Used for ORA



LFSR as ORA



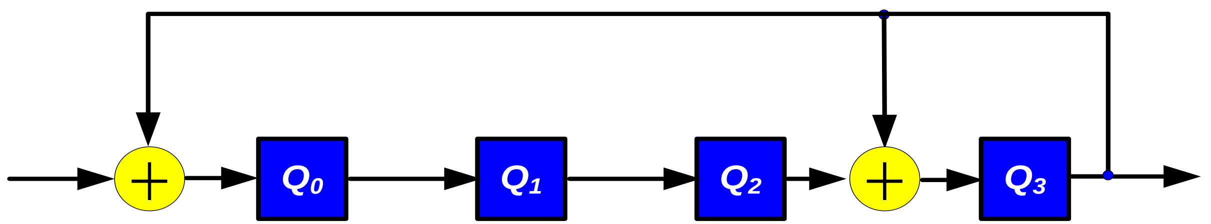
cycle	LFSR input	Q_0	Q_1	Q_2	Q_3	LFSR output
0	011011011	0	0	0	0	
1-3	...					
4	01101	1	0	1	1	
5	0110	0	1	0	0	1
6	011	0	0	1	0	01
7	01	1	0	0	1	001
8	0	0	1	0	1	1001
9		1	0	1	1	11001

$\underbrace{Q_0\ Q_1\ Q_2\ Q_3}_{\text{signature}}$ $\underbrace{\text{LFSR output}}_{\text{not used}}$

Quiz

Q: Suppose CUT output is '001001'. (right bit first)
What is the signature after 6 cycles?
ANS:

LFSR Input
(from CUT)
001001



cycle	LFSR input	Q_0	Q_1	Q_2	Q_3	LFSR output
0	001001	0	0	0	0	
1-3	…					
4						
5						
6						

Too Slow. Any Better Method?

Cyclic Redundancy Code (CRC) Theory

- **Represent bit streams by polynomials**
  - x **is dummy variable**
  - **Exponent represents delay**
  - **bits are coefficients**

$$M(x) = \sum_i b_i x^i$$

- **Example:** 011011011 → $x + x^2 + x^4 + x^5 + x^7 + x^8$
  - **Left bits = LSB, right bits =MSB**

For more details, see reference book (BA)
or textbooks in *finite field*

Évariste Galois
1811-1832

Modular-2 Arithmetic

- *Modulo-2:* **Addition (=subtraction) is XOR, Multiplication is AND**
  - **0+0=0, 0+1=1, 1+0=1, 1+1=0**
  - **0x0=0, 0x1=0, 1x0=0, 1x1=1**
  - ***aka. Galois Field 2, GF(2)***
- **GF(2) Multiplication** **GF(2) Division**

GF(2) Multiplication:

$$(x^3+x^2+x+1)\times(x^2+x+1)$$

$$\begin{array}{r}
x^3+x^2+x+1 \\
\underline{x^2+x+1} \\
x^3+x^2+x+1 \\
x^4+x^3+x^2+x \quad\; \\
\underline{x^5+x^4+x^3+x^2} \qquad\qquad\;\, \\
x^5+0+x^3+x^2+0+1
\end{array}$$

GF(2) Division:

$$\begin{array}{r}
x^3+x^2+x+1 \\
x^2+x^1+1\overline{)\,x^5+0+x^3+x^2+0+1} \\
\underline{x^5+x^4+x^3} \qquad\qquad\qquad \\
x^4+0+x^2 \qquad\quad\;\; \\
\underline{x^4+x^3+x^2} \qquad\quad\;\; \\
x^3+0+0 \qquad \\
\underline{x^3+x^2+x} \qquad \\
x^2+x+1 \\
\underline{x^2+x+1} \\
\underline{0}
\end{array}$$

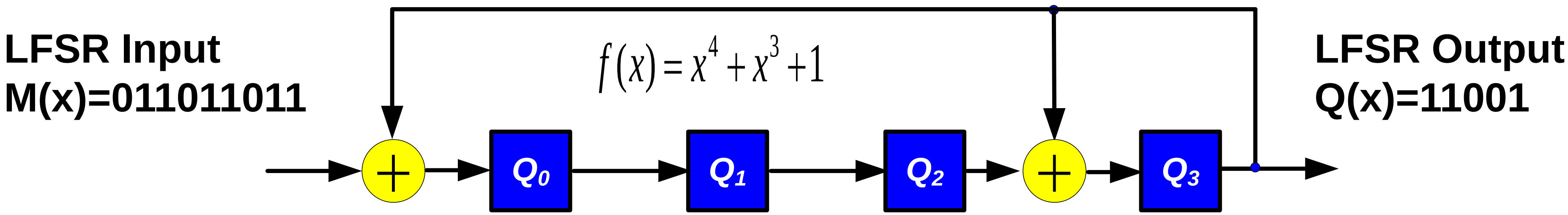
Congruent

- $M(x) \div f(x) = Q(x) \;\ldots.\; R(x)$
- If $M_1(x)$ and $M_2(x)$ have same remainders when divided by $f(x)$
  - $M_1(x)$ and $M_2(x)$ are *congruent*
  - $M_1(x) \equiv M_2(x)$ mod $f(x)$
- If $M(x) \equiv 0$ mod $f(x)$
  - $M(x)$ is *divisible* by $f(x)$

$$
\begin{array}{r}
x^3 + x^2 + x + 1 \\
x^2 + x^1 + 1 \overline{\big) x^5 + 0 + x^3 + x^2 + 0 + 1} \\
\underline{x^5 + x^4 + x^3} \qquad\qquad\quad \\
x^4 + 0 + x^2 \qquad\quad \\
\underline{x^4 + x^3 + x^2} \qquad\quad \\
x^3 + 0 + 0 \quad\; \\
\underline{x^3 + x^2 + x} \quad\; \\
x^2 + x + 1 \\
\underline{x^2 + x + 1} \\
\underline{\quad 0 \quad}
\end{array}
$$

Congruent $M_1(x) \equiv M_2(x)$

LFSR is GF(2) Divider



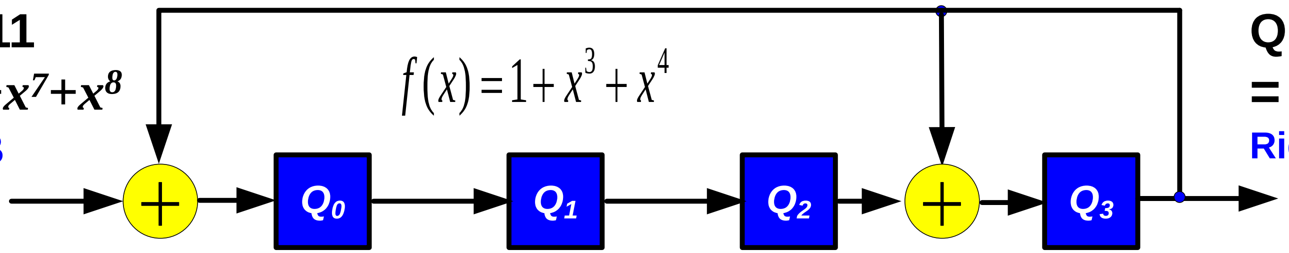
- **Assume initial LFSR content = 0000, then**
- $M(x) \div f(x) = Q(x) \ldots\ R(x)$
  - **LFSR Input bit stream = dividend** $M(x)$
  - **LFSR characteristic polynomial = divisor** $f(x)$
  - **LFSR output bit stream = quotient** $Q(x)$
  - **Signature = Remainder** $R(x)$
  - $R(x) \equiv M(x)$ **mod** $f(x)$

GF(2) Divider is Simple

M(x)=011011011

= $x+x^2+x^4+x^5+x^7+x^8$

Right bit is MSB



Q(x)=11001

= $1+x+x^4$

Right bit is MSB

cycle	LFSR input	Q_0	Q_1	Q_2	Q_3	LFSR output
0	011011011	0	0	0	0	
1-3	…					
4	01101	1	0	1	1	
5	0110	0	1	0	0	1
6	011	0	0	1	0	01
7	01	1	0	0	1	001
8	0	0	1	0	1	1001
9		1	0	1	1	11001

remainder=R(x)=$1+x^2+x^3$ **quotient=Q(x)=**$1+x+x^4$

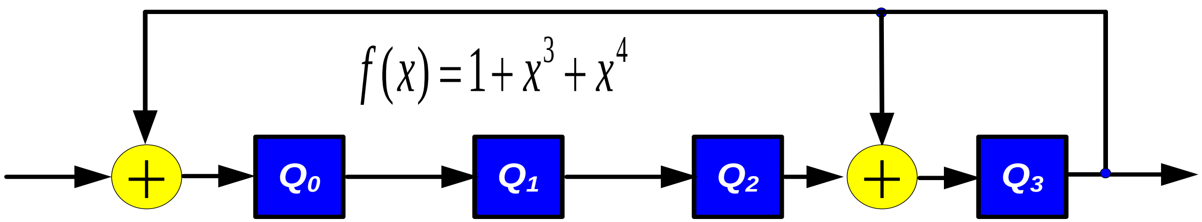
$$(x+x^2+x^4+x^5+x^7+x^8) \div (1+x^3+x^4) = 1+x+x^4 \;\ldots\ldots\; 1+x^2+x^3$$

$$M(x) \quad \div \quad f(x) \quad = \quad Q(x) \quad \ldots\ldots \quad R(x)$$

Quiz

Q: Suppose CUT output is '001001'. (right bit first)
Use GF(2) division to find quotient and remainder
ANS:

LFSR Input (from CUT) 001001

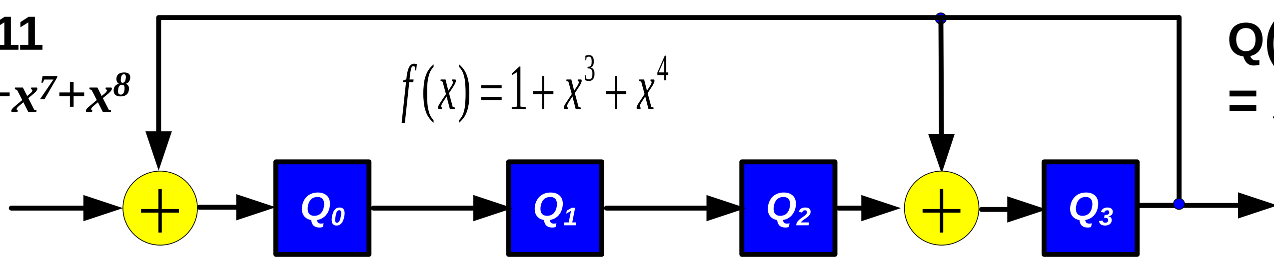


cycle	LFSR input	Q_0	Q_1	Q_2	Q_3	LFSR output
0	001001	0	0	0	0	
1-3	...					
4	00	1	0	0	1	
5	0	1	1	0	1	1
6		1	1	1	1	11

Why LFSR = Divider?

- **Modular-form (Type-2) LFSR**
  - **shift-and-add = shift-and-subtract = mod f(x) divider**

M(x)=011011011
= $x+x^2+x^4+x^5+x^7+x^8$



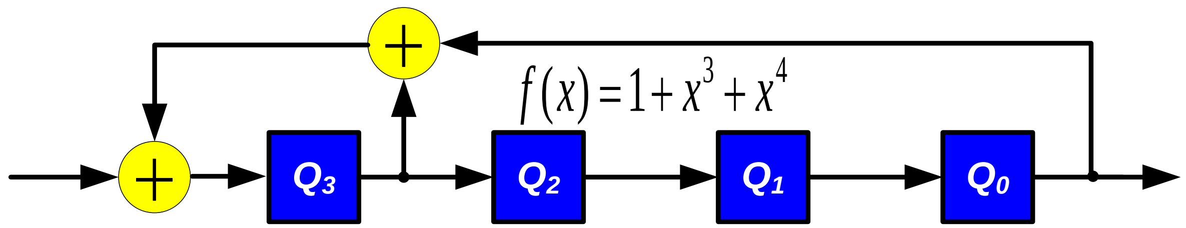
Q(x)=11001
= $1+x+x^4$

$$
\begin{array}{r}
x^4 \\
1+x^3+x^4 \,\big)\, \overline{x+x^2+x^4+x^5+x^7+x^8} \\
x^4 \qquad +x^7+x^8 \\
\hline
x+x^2+ \qquad +x^5 \\
\ldots
\end{array}
$$

cycle	LFSR input	Q_0	Q_1	Q_2	Q_3	LFSR output
0	011011011	0	0	0	0	
1-3	…					
4	01101	1	0	1	1	
5	0110	0	1	0	0	1

So We Call It "Modular-form" LFSR

How about Standard-form LFSR?



cycle	LFSR input	Q_3	Q_2	Q_1	Q_0	LFSR output
0	011011011	0	0	0	0	
1-3	...					
4	01101	1	0	0	1	
5	0110	1	1	0	0	1
6	011	1	1	1	0	01
7	01	0	1	1	1	001
8	0	0	0	1	1	1001
9		1	0	0	1	11001

1 $+x^3$ $1+x+x^4$

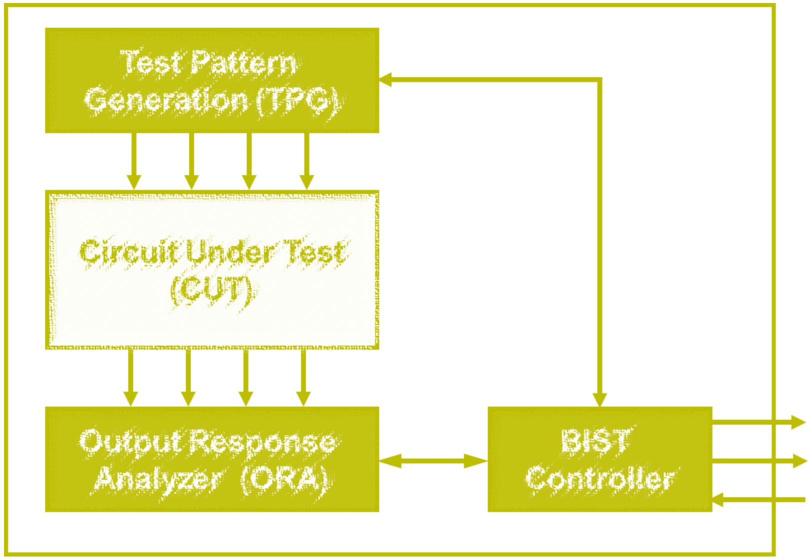
signature≠remainder

quotient is correct

Std-form LFSR ≠ Divider

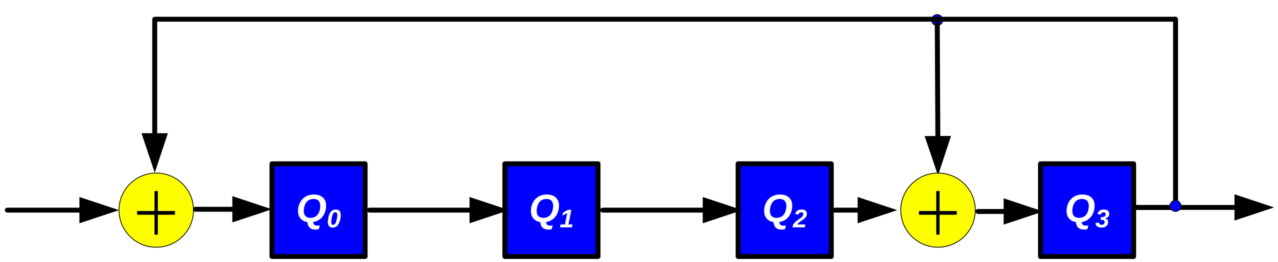
BIST Part 2

- **<u>Output Response Analysis</u>**
  - Simple ORA
  - **<u>LFSR-based ORA</u>**
    - **<u>Serial : compress one bit at a time</u>**
      - **CRC Theory**
      - **<u>PAL analysis</u>**
      - **How to design LFSR as ORA?**
    - Parallel : compress multiple bits at a time
- BIST Architecture
- Issues with BIST
- Conclusions



Linearity of Signature

- $[M_1(x) + M_2(x)]$ mod $f(x) \equiv [M_1(x) \bmod f(x)] + [M_2(x) \bmod f(x)]$ mod $f(x)$
- Example:
  - $M_3(x) = M_1(x) + M_2(x)$
    - $x+x^4+x^7+x^8 = (x+x^2+x^4+x^5+x^7+x^8) + (x^2+x^5)$
  - Then signature $R_3(x) = R_2(x) + R_1(x)$
    - $(x+x^2+x^4+x^5+x^7+x^8) \div (1+x^3+x^4) = 1+x+x^4 \;\ldots\ldots\; 1 \quad +x^2+x^3$
    - $(\quad x^2 \quad +x^5) \quad \div (1+x^3+x^4) = 1+x \quad \ldots\ldots\; 1+x+x^2+x^3$
    - $(x \quad +x^4 \quad +x^7+x^8) \div (1+x^3+x^4) = \quad x^4 \;\ldots\ldots\; \quad x$



Signature of (Σ inputs) ≡ Σ (signature of inputs)

What Is Aliasing?

- $M_{good}(x)$ is good output, $R_{good}(x)$ is gold signature
- $M_{faulty}(x)$ is faulty output, $R_{faulty}(x)$ is faulty signature
- Aliasing occurs when $R_{good}(x) = R_{faulty}(x)$

- Example:
  - $M_{good}(x)$, gold signature = $1+x^2+x^3$
    - $(x+x^2+x^4+x^5+x^7+x^8) \div (1+x^3+x^4) = 1+x+x^4$ …… $1+x^2+x^3$
  - $M_{faulty}(x)$, faulty signature = x no aliasing
    - $(x \quad +x^4 \quad +x^7+x^8) \div (1+x^3+x^4) = \quad x^4$ …… x
  - $M_{faulty2}(x)$, faulty signature = $1+x^2+x^3$ aliasing!
    - $(\quad x^2 \quad +x^7+x^8) \div (1+x^3+x^4) = 1+ \quad x^4$ …… $1+x^2+x^3$

Aliasing Means $R_{faulty} = R_{good}$

Aliasing Condition

- $M_{good}(x)$ is good output
- $M_{faulty}(x)$ is faulty output
- $M_{error}(x)$ = difference between $M_{faulty}(x)$ and $M_{good}(x)$
  - $M_{faulty}(x) = M_{good}(x) + M_{error}(x)$
- Aliasing means
  - $R_{faulty}(x) = R_{good}(x)$
  - $M_{faulty}(x) \equiv M_{good}(x) \mod f(x)$
- Aliasing condition:
  - $M_{good}(x) + M_{error}(x) \equiv M_{good}(x) \mod f(x)$
  - $M_{error}(x) \equiv 0 \mod f(x)$
  - *i.e.* $M_{error}(x)$ *divisible* by f(x) of LFSR

Aliasing when M_{error} Divisible by f

Quiz

$M_{good}(x)$, gold signature = $1+x^2+x^3$

- $(x+x^2+x^4+x^5+x^7+x^8) \div (1+x^3+x^4) = 1+x+x^4$ …… $1+x^2+x^3$

$M_{faulty2}(x)$, faulty signature = $1+x^2+x^3$ aliasing!

- $(\quad x^2 \quad\quad +x^7+x^8) \div (1+x^3+x^4) = 1+ \quad x^4$ …… $1+x^2+x^3$

Q1: $M_{error}(x)$ = ?

ANS:

Q2: Use long division to verify that $M_{error}(x) \equiv 0$ mod $f(x)$

ANS:

PAL Estimate

- Assume $M(x)$, length m
  - $M_{faulty}(x) = M_{good}(x) + M_{error}(x)$
  - Divisor $f(x)$, degree N
  - Every bit has equal probability to flip
    - Every bit of $M_{error}(x)$ can be 1 with equal probability

$$\begin{array}{r} 11101 \\ \oplus\ 00010 \\ \hline 11111 \end{array} \qquad m = 5$$

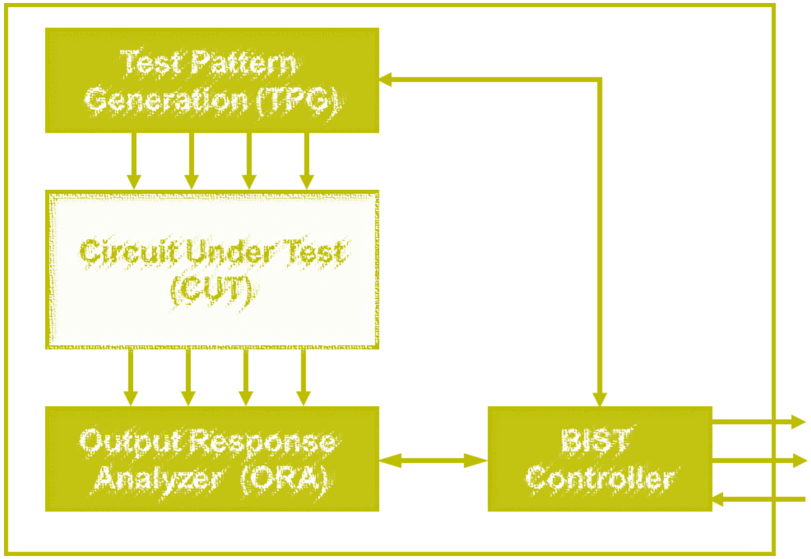
- Total number of errors that can occur
  - = total number of nonzero $M_{error}(x)$ polynomials
  - = 2^m-1
- Number of errors that cause aliasing
  - = number of nonzero $M_{error}(x)$ that are divisible by $f(x)$
  - = 2^{m-N}-1

$$\boxed{PAL = \frac{2^{m-N}-1}{2^m-1} \approx 2^{-N} \ (if\ m >> N)}$$

- 5-degree LFSR PAL=1/32; 6-degree LFSR PAL=1/64
  - LFSR increases 1 bit, PAL decreases 50%

BIST Part 2

- **<u>Output Response Analysis</u>**
  - Simple ORA
  - **<u>LFSR-based ORA</u>**
    - **<u>Serial : compress one bit at a time</u>**
      - **CRC Theory**
      - **PAL analysis**
      - **<u>How to design LFSR as ORA?</u>**
    - Parallel : compress multiple bits at a time
- BIST Architecture
- Issues with BIST
- Conclusions



Design Guideline

- Given a PAL, design an LFSR
  - 1. How many stages, *N* =? (Degrees of LFSR)
    - $N = -\log_2 PAL$
  - 2. Which polynomial?
    - Primitive polynomial
  - 3. Test length, *m* must be greater than *N*

$$PAL = \frac{2^{m-N} - 1}{2^m - 1} \approx 2^{-N} \quad (if \;\; m >> N)$$

- Example: target PAL = 10^{-6}, test length = 1,000
  - $N = 20$
  - PAL = $2^{-20} \approx 10^{-6}$
  - Find a primitive polynomial of degree 20
    - *e.g.* $1+x^3+x^{20}$
  - Test length >> 20
    - Assumption valid

What Polynomial?

- Study shows [Williams 88]
  - PAL of primitive polynomial converge to final steady state value
    - Faster than non-primitive polynomials
  - So it is good to use primitive polynomials

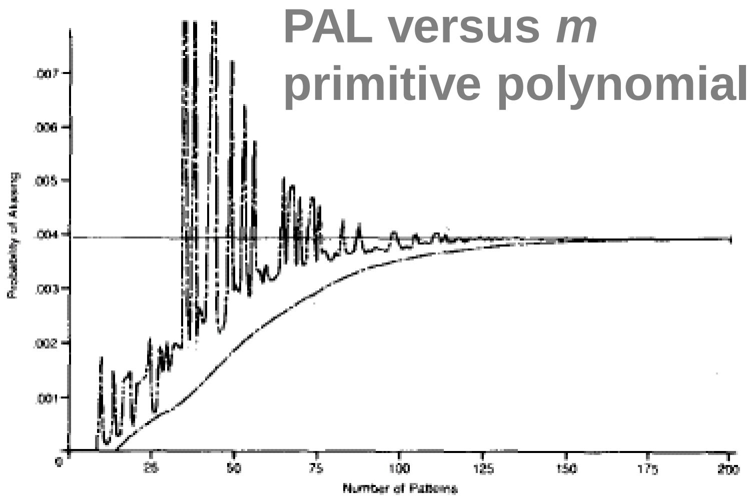


Fig. 15. Aliasing probability as a function of the test length.

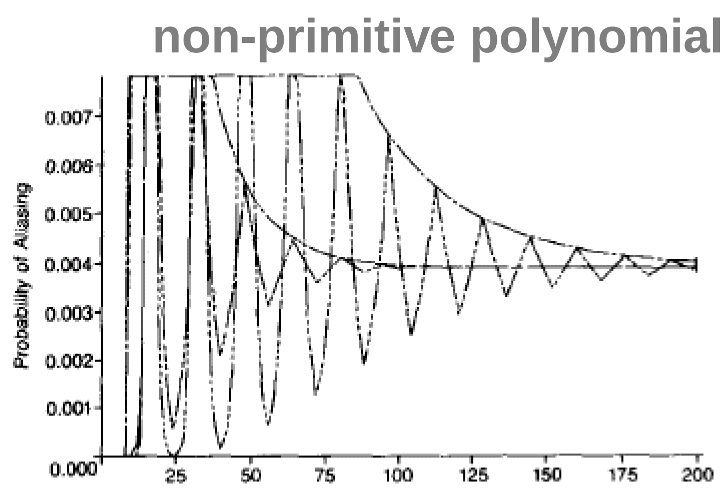


Fig. 16. Aliasing probability as a function of the test length $X^8 + 1$.

Use Primitive Polynomial

Summary

- **<u>LFSR-based ORA</u>**
  - **Type-2 (modular form) LFSR is divider**
  - **Aliasing occurs when M_{error} is divisible by f**
  - **$PAL_{LFSR} = 2^{-N}$ very low**
  - **Use primitive polynomial**