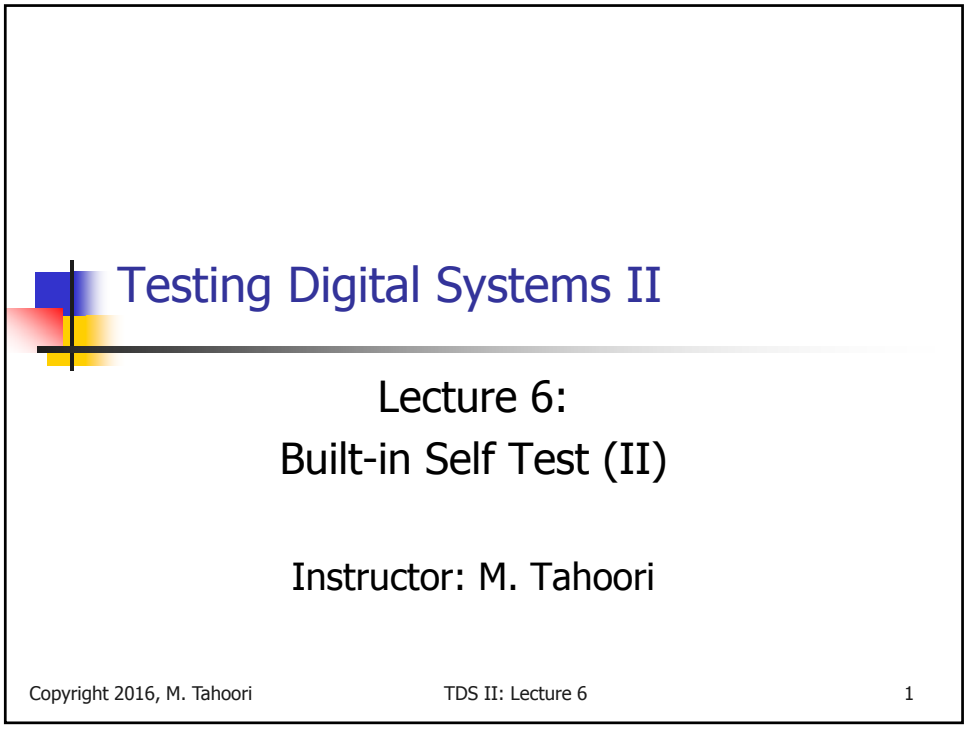
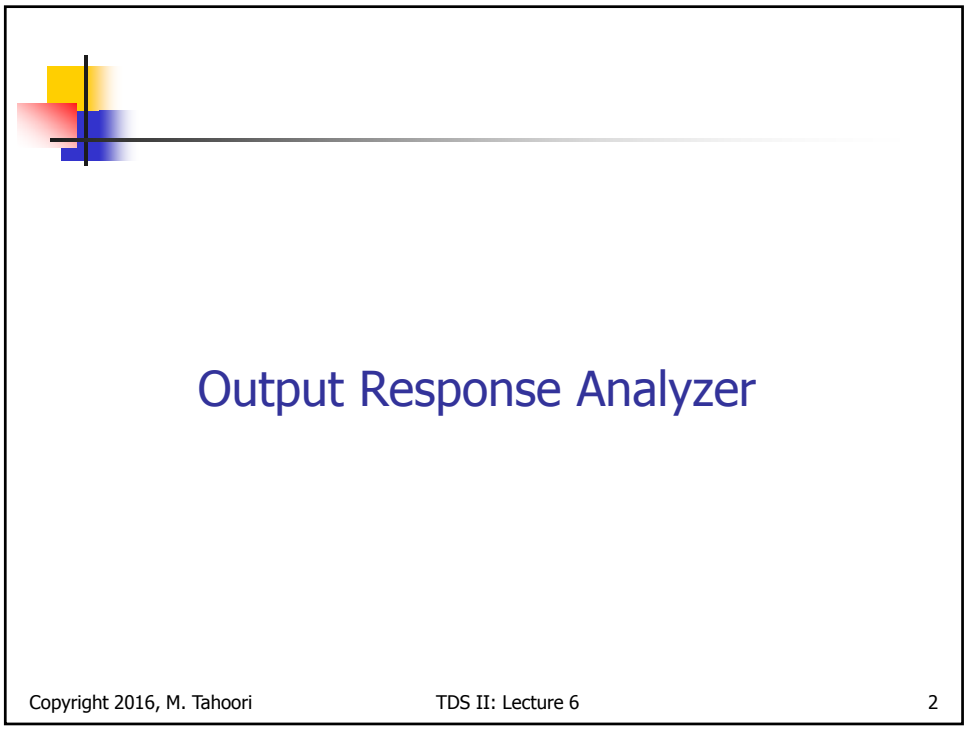

Testing Digital Systems II
Lecture 6:
Built-in Self Test (II)
Instructor: M. Tahoori
Copyright 2016, M. Tahoori
TDS II: Lecture 6
1

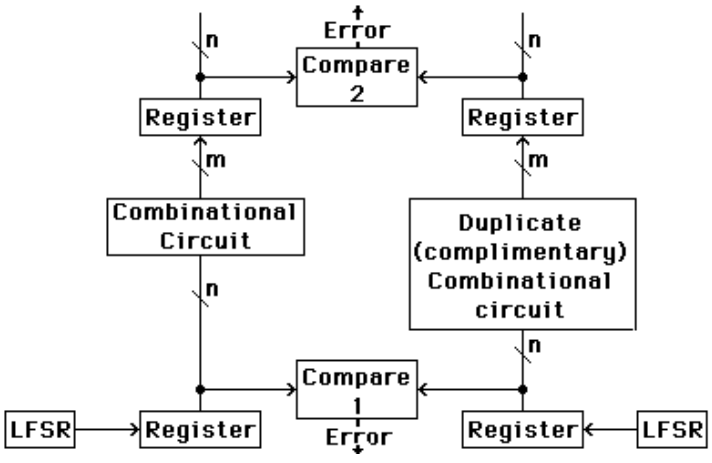

Output Response Analyzer
Copyright 2016, M. Tahoori
TDS II: Lecture 6
2

## Output Response Analysis

- Duplication
- Response Compaction
  - Parity, Transition, Ones Counting
  - Syndrome Analysis, Checksums
  - Walsh Coefficients — Spectral Techniques
  - Signature Analysis — Galois Division using an LFSR

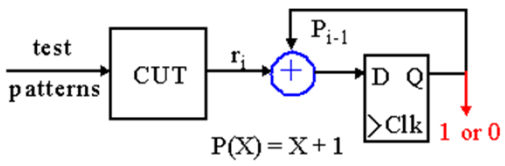


## Duplication



## Parity

- Parity testing
  - Detects all single bit and multiple bit errors of odd cardinality




$$P = \sum_{i=1}^{L} r_i$$

where L is the length of the test.



## Ones Count Testing

Assume the CUT has one output and the output contains a stream of *L* bits. Let the fault-free output response be

$$\{r_0, r_1, r_2 \cdots r_{L-1}\}$$

Ones count testing will need a counter to count the number of 1s in the bit stream.

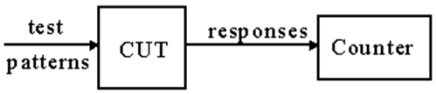
***Aliasing probability*** *[Savir 1985]*

$$P_{OC}(m) = (C(L,m) - 1)/(2^L - 1)$$

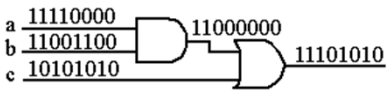
L bit output sequence containing m ones

## Ones Counting

- One Counting :
  - # of 1's in the response stream is compared with fault free value




- For the fault-free circuit, the 1-count is 5.
  - The faults a/0 and a/1 would be detected since the counts would be 4 and 6.






## Transition Count Testing

Transition count testing is similar to that for ones count testing, except the signature is defined as the number of *1-to-0* and *0-to-1 transitions.*

*[Hayes 1976]*

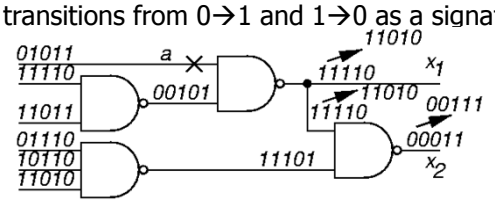
***Aliasing probability***

$$P_{TC}(m) = (2C(L-1,m)-1)/(2^L-1)$$

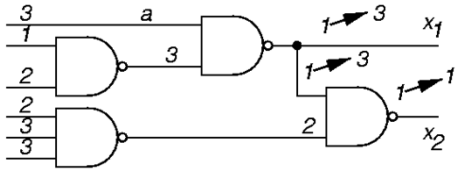
L bit output sequence containing m transitions

## Transition Counting

- *Transition Count Response Compaction*
  - Count # transitions from 0→1 and 1→0 as a signature




(a) Logic simulation of good machine and fault a stuck-at-1.

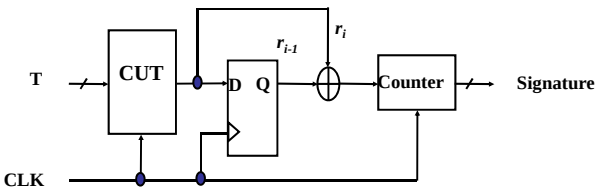



(b) Transition counts of good and failing machines.



## Transition Counting Details

- Transition count:
- $C(R) = \sum_{i=1}^{m} (r_i \oplus r_{i-1})$ for all $m$ primary outputs
- To maximize fault coverage:
  - Make $C(R_0)$ – good machine transition count – as large or as small as possible

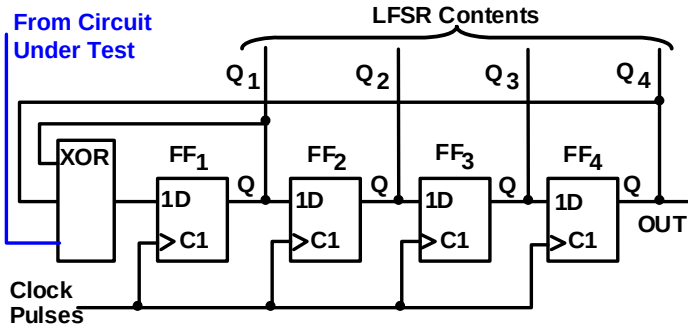


## Signature Analysis

- Definition:
  - *Signature* is Remainder in LFSR after Galois Division of Test Output Bit Stream
- Aliasing Definition and Coverage Loss Demonstration
- Division Examples
- Linearity Demonstration
- Aliasing Analysis and Design Guidelines
- Multiple-Output DUT
  - Multiplexed Outputs
  - Parallel Signature Analysis (aka MISR)
  - Masking in Parallel Signature Analyzer

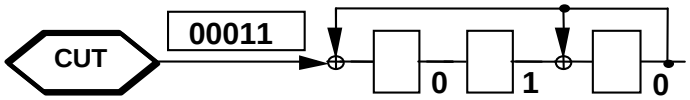


## Serial Signature Analyzer



## Signature Analysis

- Definition:
  - *Signature* is Remainder in LFSR after Galois Division of Test Output Bit Stream



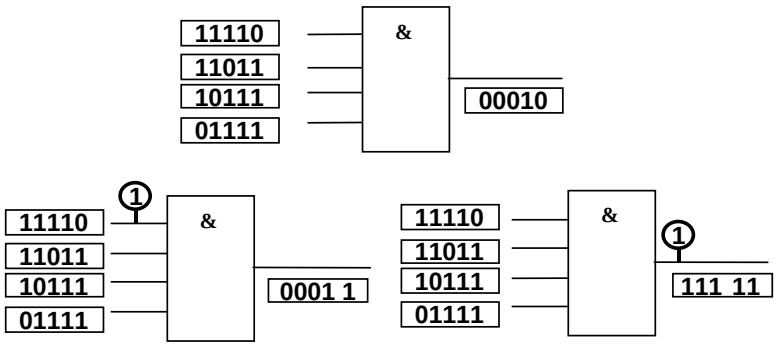

  - LFSR Polynomial: $1+X^2+X^3$
  - Test Output Data = 00011 = $X^3+X^4$
  - Signature = 010 = X (remainder of $X^3+X^4 \div 1+X^2+X^3$ )
    - (Assumes 000 Initial Contents)
- *Serial Signature Analysis*
  - One Test Output Data Stream



## Signature Analysis

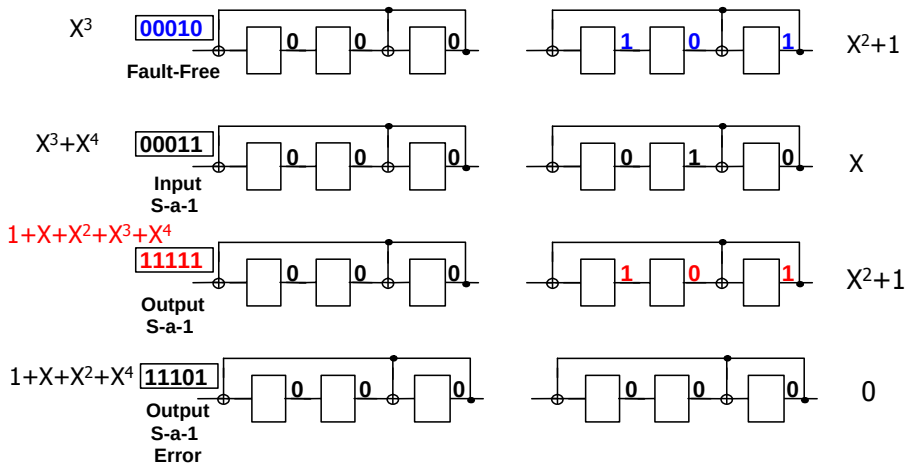
- Why is It Used?
  - Reduce Amount of Stored Test Data
- When is It Used?
  - Usually for Built-In Self Test
- What are Its Drawbacks?
  - Possible Loss of Effective Fault Coverage
    - Aliasing
      - Signature (Fault Present) = Signature (Fault Free)
    - Masking

## Stuck-Fault Error Pattern Example

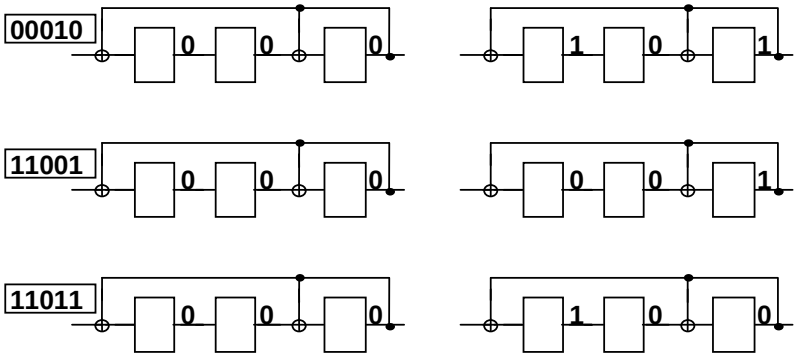





## Alias Example

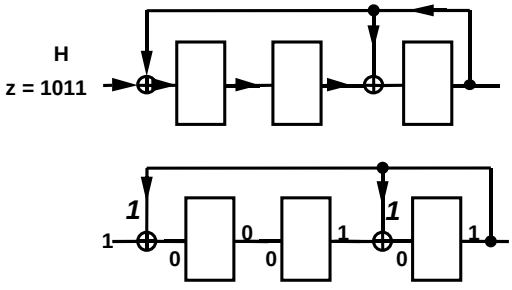


## Linearity Example






## Alias Example




- Signature [H] = 0 (H is LFSR generating polynomial)
- Signature [G × H] = 0 for any polynomial G
- $Z_{Faulty} = Z_{Fault\text{-}Free} + Z_{Error}$
- $Z_{Error} = G \times H \Rightarrow$ Signature $[Z_{Faulty}]$ = Signature $[Z_{Fault\text{-}Free}]$

## Determining Effective Fault Coverage

- Approach 1:
  - Include signature analysis in fault grading simulation
    - Precludes fault dropping in simulation
      - Can be prohibitively expensive
- Approach 2:
  - Retain intermediate signatures during test
    - Allows fault dropping in simulation
      - Requires storage of intermediate signatures
      - Requires interruption of test process
- Both approaches costly, no insight into design



## Design Issues

- What LFSR for Signature Register?
  - How Many Stages?
  - What Feedback Connections?
    - 2 Primitive or Non-Primitive Polynomial

## Alias Probability Bound

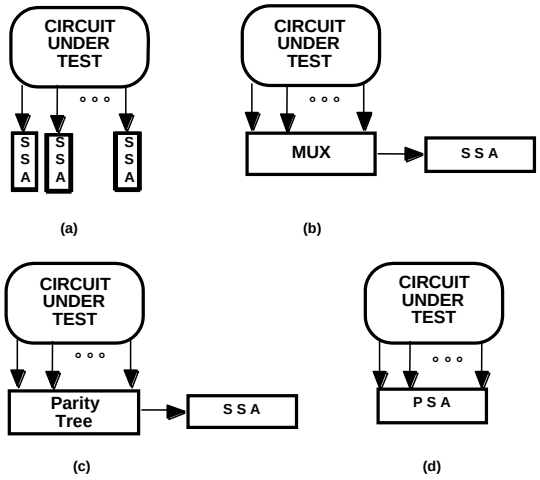
- Serial Signature Analysis
  - $L_c$: # patterns generated by LFSR
  - L: test length
- Alias Probability $\leq (1 + \varepsilon) / L$ $\quad L < L_c$
  - $\varepsilon < 1$, small for large $L$
    - Example: $\varepsilon$ =0.04 for $L$=1000
- Alias Probability $\leq 1$ $\quad L = hL_c$
- Alias Probability $\leq 2/(L_c+1)$ $\quad L > L_c$ , $L \neq hL_c$



## Serial Signature Analysis

- Which Polynomial?
  - Primitive Maximizes Lc for Given r
- Exact Aliasing Not Monotonic
- Conjecture: Best Monotonic Bound = 1/L
- Guideline:
  - Use Signature Polynomial with Period > Test Length.
  - Example (Serial Signatures)
    - $L=10^6$, Primitive Polynomial, $r \geq 20$
    - Alias Probability < 0.0001 %

## Multiple Output Circuits

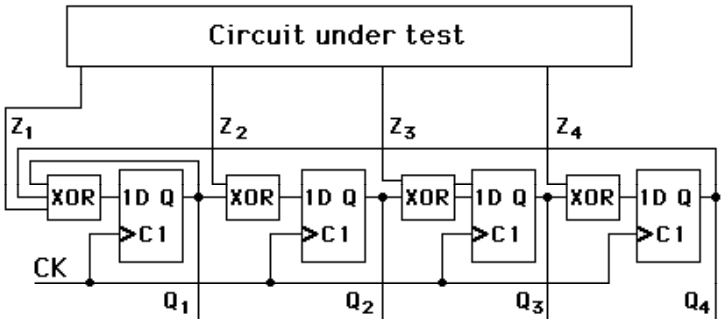



**SSA = Serial Signature Analyzer** **PSA = Parallel Signature Analyzer or MISR**



## Parallel signature Analyzer

- aka MISR or Multiple-Input Signature Register

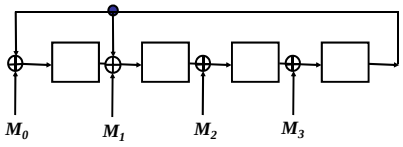


## Multiple-Input Signature Register

- This scheme is equivalent k single input SAs
  - but with the input stream shifted in time
  - $Z(X) = Z_1(X) + XZ_2(X) + \ldots + X^k Z_{k+1}(X)$.
- The error polynomial of the four outputs
  - $E(X) = E_1(X) + XE_2(X) + X^2 E_3(X) + X^3 E_4(X)$



## 4-stage MISR




*A 4-stage MISR*

| | |
|---|---|
| $M_0$ | 1 0 0 1 0 |
| $M_1$ | 0 1 0 1 0 |
| $M_2$ | 1 1 0 0 0 |
| $M_3$ | 1 0 0 1 1 |
| $M$ | 1 0 0 1 1 0 1 1 |

*An equivalent M sequence*

***Aliasing probability***

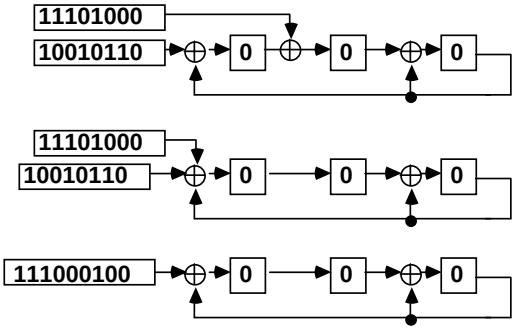
$$P_{PSA}(n) = (2^{(mL-n)} - 1)/(2^{mL} - 1)$$

mL-bit sequences to be compacted in n-stage MISR, $L > n \geq m \geq 2$

For $L >> n$, $P_{PSA}(n) \approx 2^{-n}$

## Masking

- What is Masking?
  - Error Bit in One Output CANCELS
    - Error Bit in Another Output
  - Errors in Zj at time t and in Zj+m at time t + m Cancel One Another






## Signature Analysis

- Serial Signature Analysis
  - Coverage Loss due to Aliasing
    - Use Primitive Polynomial?
    - Use Longer Register?
    - Use Multiple Registers?
- Parallel Signature Analysis (aka MISR)
  - Coverage Loss due to Aliasing and Masking
    - Use Primitive Polynomial?
    - Use Longer Register?
    - Use Multiple Registers?
    - Use Checksum along with Signature?