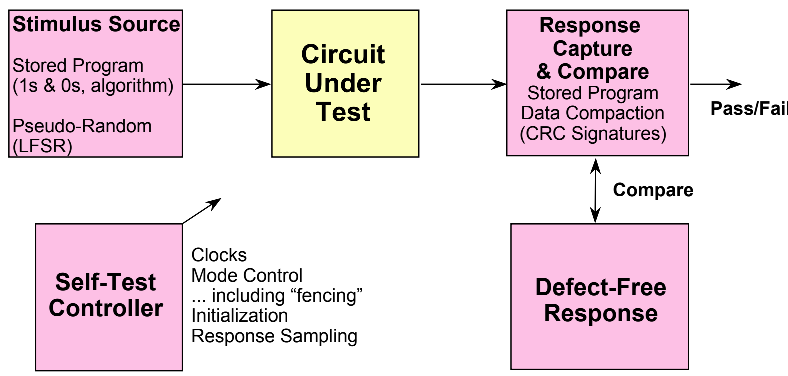
Built-In Self Test

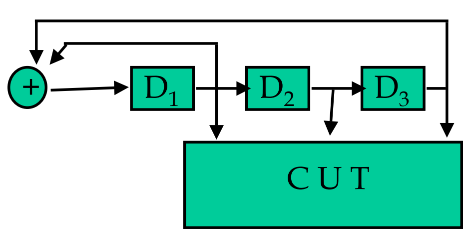


Test Pattern Generator for BIST

(a) Exhaustive test: use a counter and apply all possible patterns (2^n patterns) to the circuit under test.

(b) Random test: Use linear-feedback shift register (LFSR) to apply random patterns to CUT.

Ex: TPG of random testing



initial value (seed) →

D_1	D_2	D_3
1	0	0
1	1	0
1	1	1
0	1	1
1	0	1
0	1	0
0	0	1
1	0	0

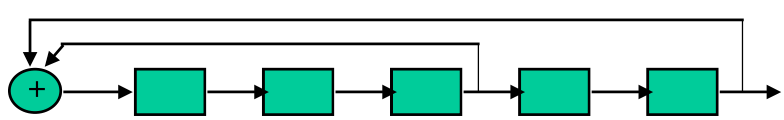
- This 3-stage LFSR can generate test sequence of length 2^3-1=7

- An m-stage LFSR can generate test seq. of length 2^m-1
  - Such sequence are called maximal length sequence.
  - Such an LFSR is called a maximum-length LFSR.
- When only a fraction of the 2^m-1 can be applied, (because m is too large), LFSR is better than counters.

Counter	LFSR
0 0 0	1 0 0
0 0 1	1 1 0
0 1 0	1 1 1
0 1 1	0 1 1
1 0 0	1 0 1
1 0 1	
1 1 0	0 1 0
1 1 1	0 0 1

- Sequences of LFSR is more random
  - Every bit is random

An LFSR can be expressed by its Characteristic Polynomial f(x)

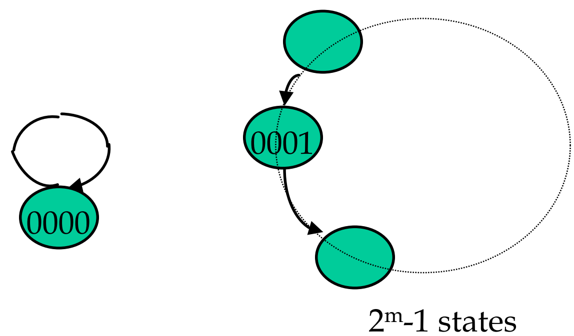


$f(x)=x^5+x^3+1$ $\qquad a_n=a_{n-3}+a_{n-5}$

- The characteristic polynomial of a maximum-length LFSR is called primitive polynomial.
- Several listings of such polynomials (up to degree 300) exist in the literature.
  - Given a CUT with *m* inputs, pick a primitive polynomial of degree *m* and construct the corresponding LFSR as a TPG.
  - Ref: "Built-In Test for VLSI", Paul H. Bardell et al., John Wiley & Sons, 1987

Characteristics of Maximun-Length LFSR

- The state diagram contains two components: one contains the all-zero state, the other contains the other 2^m-1 states.



- For every bit, the number of 1's differs from the number of 0's by one.
- The number of transitions between 1 and 0 that an *m*-sequence makes in one period is (m+1)/2.
- Autocorrelation between different bits:
  - Autocorrelation function is defined as:

$$C(i,j) = \frac{1}{2^m - 1} \sum_{n=1}^{2^m - 1} b_i(n) b_j(n)$$

$$where \begin{cases} b_i(t) = 1 & where\ \ a_i(t) = 0 \\ b_i(t) = -1 & where\ \ a_i(t) = 1 \end{cases}$$

Ex: $k=3$

D_1	D_2	D_3
1	0	0
1	1	0
1	1	1
0	1	1
1	0	1
0	1	0
0	0	1
1	0	0

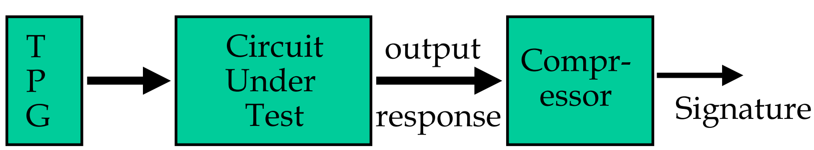
$C(1,2)=-1/7$

$C(1,3)=-1/7$

$C(2,3)=-1/7$

- The autocorrelation function of every maximum-length LFSR of period $p=2^m-1$ is:
  - $C(i,i)=1$
  - $C(i,j)=-1/p \quad i \neq j$

Test Response Compression Techniques



- The signature & its collection algorithm should meet the following guideline:

1. The algorithm must be simple enough to be implemented as part of the built-in test circuitry.
2. The implementation must be fast enough to remove it as a limiting factor in test time.
3. The compression method should minimize the loss of information. Specifically it should minimize the loss of evidence of a fault indicated by a wrong response from the circuit under test.

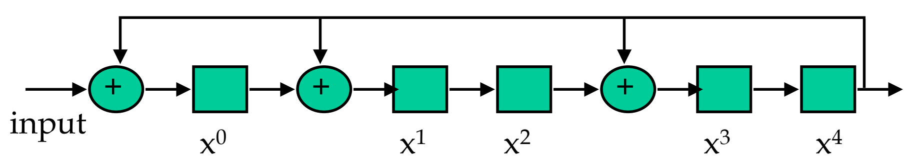
Use of LFSRs for Polynomial Division

- Suppose we are interested in modulo 2 division.
  $P(x)/G(x) = (x^7+x^3+x)/(x^5+x^3+x+1)$
- The longhand division can be conducted in terms of the detached coefficients only:

```
                101       Q(x)=x^2+1
101011 | 10001010
         101011
         --------
         00100110
           101011
         --------
           001101=R      R(x)=x^3+x^2+1
```

- This division process can be mechanized using a LFSR.

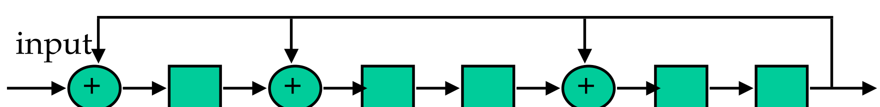
Ex: LFSR implementing division by
$f(X)=x^5+x^3+x+1$



- When a shift occurs, x^5 is replaced by $x^3+x^1+x^0$.
- Thus, whenever a quotient coefficient(the x^5 term) is shifted off the right-most stage, $x^3+x^1+x^0$ is added to the register (or subtracted from the register since addition is the same as subtraction modulo 2). Effectively, the dividend has been divided by $x^5+x^3+ x+1$.

- If the LFSR is initialized to zero & the message word (or dividend) P(x) is serially streamed to the LFSR input, high-order coefficient first, the content of the LFSR after the last message bit is the <u>remainder</u> from the division of the message polynomial by the divisor G(x).

Example: $P(x)=x^7+x^3+x$ input=10001010



time	input	x^0	x^1	x^2	x^3	x^4	x^5
0		0	0	0	0	0	
1	1	1	0	0	0	0	0
2	0	0	1	0	0	0	0
3	0	0	0	1	0	0	0
4	0	0	0	0	1	0	0
5	1	1	0	0	0	1	0
6	0	1	0	0	1	0	1
7	1	1	1	0	0	1	0
8	0	1	0	1	1	0	1

Remainder $R=x^3+x^2+1$

Quotient $Q=x^2+1$

- Any data, such as the test response results from a circuit, can be compressed into a "code word" by an LFSR.
- This code word, the remainder from the division process, is called the "signature" of the input data stream.
- The LFSR itself is called a "signature analyzer".
  - $P(x)=Q(x)G(x) + R(x)$

divisor signature

LFSR polynomial

- If P(x) is the polynomial of the correct data

  => $P'(x)=P(x)+M(x)G(x)$ will have the same signature as P(x) for any M(x).
- Example: $P(x)=x^7+x^3+x$

  $G(x)=x^5+x^3+x+1$

  signature $R(x)=x^3+x^2+1$

  $P'(x)=P(x)+G(x)=x^7+x^5+1$

  $P''(x)=P(x)+x \bullet G(x)=x^7+x^6+x^4+x^3+x^2$

  P'(x) and P''(x) have same signature x^3+x^2+1.
- Aliasing: condition in which a faulty circuit with erroneous response produces the same signature as a good circuit.
- Aliasing probability is usually used to measure the quality of a data compressor.

What is the aliasing probability of using LFSR as a data compressor?

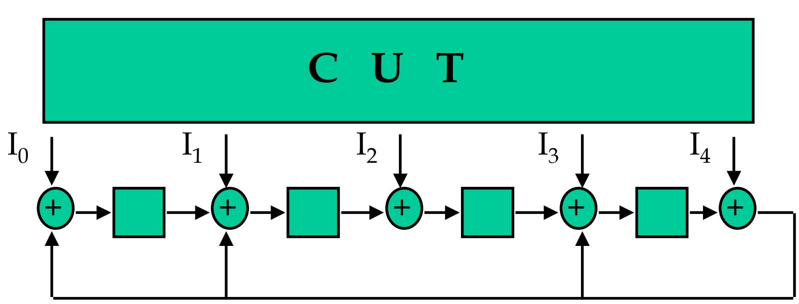
- Suppose the input string is m-bit long. There are 2^m-1 possible wrong bit streams.
- If the divisor polynomial G(x) is of degree r, then it has 2^{m-r}-1 nonzero multiples that result in a polynomial of degree less than m.

  => There are 2^{m-r} -1 wrong m-bit streams that can map into the same signature as the correct bit stream.

  – => Aliasing prob. $P(M)=(2^{m-r}-1)/(2^m-1)$
  – => For large m, $P(M) \cong 1/2^r$.

Multiple-Input Signature Register (MISR)

- For built-in testing of multiple-output circuits, the overhead of a single-input signature analyzer on every output would be high.
- A multiple-input signature register (MISR) is used for multiple-output circuits.



- It can be proved that the aliasing probability of a MISR is:

  $(2^{n-1}-1)/(2^{m+n-1}-1) \cong 1/2^m$

  m: the number of stages in MISR

  n: length of data to be compressed.

Ref: "Built-In Test for VLSI", Chapter 5, Paul H. Bardell et al, 1987.