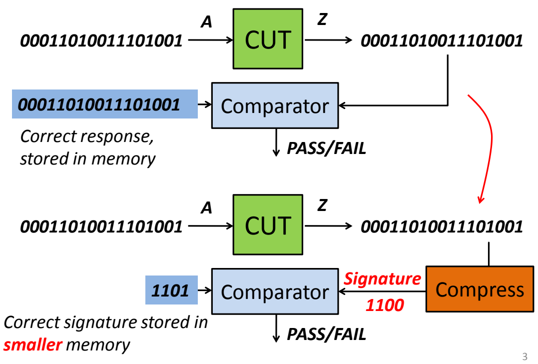
ECE-470 Digital Design II
Compression Techniques

Overview

- Motivation
- Basics
- Ones-Count compression
- Transition-Count compression
- Parity checking
- Syndrome checking
- Signature analysis
- Summary

Motivation



Key Points

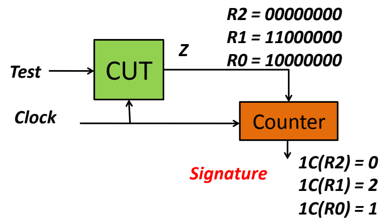
- Memory savings should overcome the extra hardware to implement the compression unit
- The compression unit could be implemented together with the CUT, achieving built-in self-test (BIST): to be discussed in a later lecture

Compression Unit Requirements

- Should not introduce big signal delays
- Length of signature should be a logarithmic factor of the length of the output response length
- If response of faulty CUT is different from correct response, the faulty signature should also be different from the good signature. No ***error masking*** (otherwise faulty response is an ***alias*** of the correct output response)

1. **Ones-Count** Compression

- Signature: $1C(R) = \sum r_i$
- Compressor: simple counter
- Degree of compression: $\lfloor log_2(m+1) \rfloor$



Ones-Count Compression

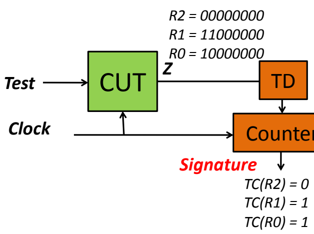
- Separate counter for each PO for multi-output circuits, or
- Parallel-to-serial converter first, then a single counter
- ***Theorem 1****: Masking probability for ones-count compression for a combinational circuit approaches* $(\pi m)^{-1/2}$

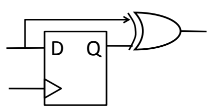
Ones-Count Compression

- ***Theorem 2****: When applying ones counting and the test set T'(C) to C, no error masking occurs for any fault in F*
- Where:
  - *C*: combinational circuit, with *F* faults of interest
  - *T = {T0, T1}* set of *m* test vectors that detect *F*
  - *T'(C)*: test set with one copy of every pattern in *T0* and *|T0| + 1* copies of every pattern in *T1*
  - For all tests in *T0*, the fault free response is 0; for all tests in *T1*, the fault free response is 1

2. Transition-Count Compression

- Signature: $TC(R) = \sum (r_i \text{ XOR } r_{i+1})$
- Compressor: transition detector (TD) and a counter with $\lceil \log_2 m \rceil$ stages
- Degree of compression: $\lceil \log_2 (m-1) \rceil$





Transition Detector (TD)

Transition-Count Compression

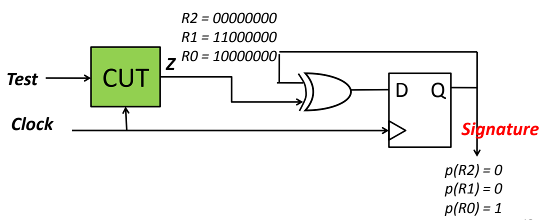
- Sensitive to the order of bits
- Does not guarantee detecting all single-bit errors
- ***Theorem 1****: In an arbitrary m-bit sequence, the probability of a single-bit error being masked is (m-2)/2m*
- ***Theorem 2****: Masking probability for transition-count compression for a combinational circuit approaches* $(\pi m)^{-1/2}$

Transition-Count Compression

- ***Theorem 3****: Let T be single-fault test for irredundant single-out circuit C. Let T0(T1) be all tests in T producing output 0(1). Construct test sequence X = t(1) t(2)… t(p):*
  - *X contains every element in T*
  - *X is alternating sequence of tests from T0 and T1. If |T0| ≥ |T1|, t(1) ϵ T0, otherwise t(1) ϵ T1. If t(i) ϵ Td, select t(i+1) ϵ* $\overline{Td}$ *for 1 ≤ i ≤ p -1*
  - *Resulting sequence is a single-fault transition-count (TC) test for C*

3. Parity-Check Compression

- Signature: *p(R) = parity of circuit response*
- Compressor: simple parity checker (LFSR)



Parity-Check Compression

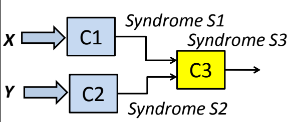
- Initial state of DFF is 0, signature is 0 if parity is even, 1 if parity is odd
- Probability of masking approaches 1/2
- Extension to multiple-output circuits:
  - Use multi-input XOR gate
  - Use separate parity checker for each output

4. Syndrome Testing

- Uses exhaustive testing: applying all 2^n test vectors
- ***Signature (syndrome) S***: normalized number of 1's in the output response stream, $S = K/2^n$
- It is a special case of 1's counting
- Examples:
  - S(3-input AND) = 1/8
  - S(3-input OR) = 7/8
- Syndrome S is a functional property of circuit implementing function f

Syndrome Testing

- It is of interest because of the concept of syndrome testability
- ***Syndrome testability***: any function f can be realized such that all single stuck-at faults are syndrome detectable



Gate type for C3	**Syndrome S3**
OR	S1 + S2 – S1S2
AND	S1S2
NAND	1 – S1S2
NOR	1 – (S1 + S2 + S1S2)
XOR	S1 + S2 – 2S1S2

Syndrome Testing

- **Definition**: A realization C of a function f is said to be syndrome-testable if no single stuck-at fault causes the circuit to have the same syndrome as the fault-free circuit
- **Lemma**: A two level irredundant circuit that realizes a unate function (a function f is unate in xi if there exists a sum of product expression for f where xi appears only in uncomplemented form) in all its variables is syndrome testable
- **Lemma**: Every two-level irredundant combinational circuit can be made syndrome testable by adding control inputs to the AND gates
- **Lemma**: Every fanout-free irredundant combinational circuit composed of AND, OR, NAND, NOR, and NOT gates is syndrome testable

5. Signature Analysis

- Compression technique based on the concept of cyclic redundancy checking (CRC)
- Implemented using linear feedback shift registers (LFSRs), utilized for:
  - Generate pseudorandom sequences
  - Compression of circuit out response, known as ***signature analysis***

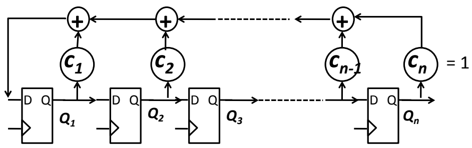
Linear Feedback Shift Registers (LFSR)

- A generating function can be associated with an LFSR's output sequence:

  $G(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_m x^m + \dots$

  $\{ a_m \} = a_0, a_1, a_2, \dots$ output sequence generated by LFSR
- Two types of LFSRs:
  - External XOR
  - Internal XOR

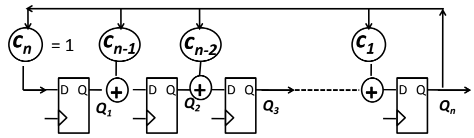
LFSR Type 1: External XOR (autonomous circuit; clock is only input)



Initial		a_{-1}	a_{-2}	a_{-3}	a_{-n-1}	a_{-n}
Current	a_m	a_{m-1}	a_{m-2}	a_{m-3}	a_{m-n-1}	a_{m-n}

19

LFSR Type 2: Internal XOR



Initial	a_{-1}	a_{-2}	a_{-3}	a_{-n-1}	a_{-n}
Current	a_{m-1}	a_{m-2}	a_{m-3}	a_{m-n-1}	a_{m-n}

20

Linear Feedback Shift Registers (LFSR)

- Note that: $a_m = \sum_{i=1}^{n} c_i a_{m-i}$
- Operation of the LSFR can be defined by a recurrence relation:

$$G(x) = \sum_{i=1}^{n} c_i x^i G(x) + \sum_{i=1}^{n} c_i x^i (a_{-i}x^{-i} + \cdots + a_{-1}x^{-1})$$

$$G(x) = \frac{\sum_{i=1}^{n} c_i x^i (a_{-i}x^{-i} + \cdots + a_{-1}x^{-1})}{1 + \sum_{i=1}^{n} c_i x^i}$$

- *G(x)* is a function of the initial state $a_{-1}, a_{-2}, \ldots, a_{1-n}$ and the feedback coefficients $c_1, c_2, \ldots, c_n$

21

Linear Feedback Shift Registers (LFSR)

- The denominator is the **characteristic polynomial** of sequence $\{ a_m \}$ and of the LFSR:

  $P(x) = 1 + c_1 x + c_2 x^2 + \ldots + c_n x^n$
- Note again: characteristic polynomial and the initial states characterize the cyclic nature of the LFSR
- **Reciprocal polynomial:**

  $P^*(x) = c_n + c_{n-1} x + c_{n-1} x^2 + \ldots + c_1 x^{n-1} + x^n$

  $P^*(x) = x^n P(1/x)$

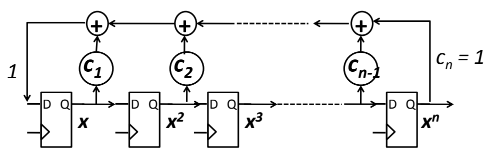
22

Linear Feedback Shift Registers (LFSR)

- Given a characteristic polynomial, it is easy to implement a type 1 LFSR to realize it
- Associate x^i *with* Q_i, the *P(x)* can be read off directly from the diagram of the LFSR
- Associate x^i *with* Q_{n-i} and label input of first FF as Q_0, then *P*(x)* can be read off directly from the diagram of the LFSR

23

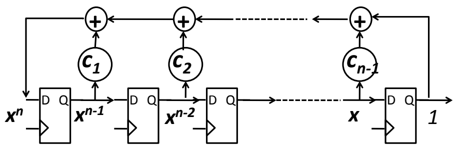
Type 1, external XOR



$P(x) = 1 + c_1 x + c_2 x^2 + \ldots + c_n x^n$

24

Type 1, external XOR



$$P^*(x) = 1 + c_{n-1}\,x + c_{n-1}\,x^2 + \ldots + c_1\,x^{n-1} + x^n$$

Periodicity of LFSR

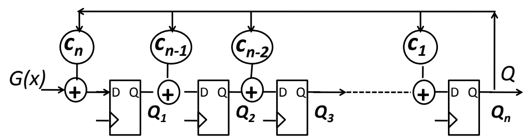
- **Theorem**: If the initial state of an LFSR is zero except $a_{-n} = 1$, then the LFSR sequence $\{a_m\}$ is periodic with a period that is the smallest integer k for which $P(x)$ divides $(1-x^k)$
- If period is 2^n-1 then LFSR generates a maximum length sequence
- In this case, the characteristic polynomial is called **primitive polynomial**

Maximum -length Sequences

- Sequences generated by LFSR with primitive polynomial are called **pseudorandom sequences**
- Any string of 2^n-1 consecutive outputs is called an **m-sequence:**
  - Number of 1's in an m-sequence differs from the number of 0's by one
  - An m-sequence produces an equal number of runs of 1's and 0's
  - One half of the runs have length 1, one forth have length 2, one eighth have length 3, etc.
- **Randomness of LFSR makes them good for generating test sequences in BIST circuits**

5. Single-Input LFSRs as Signature Analyzers

- Signature analyzer is a type 2 single-input LFSR
- Smallest degree of masking makes this approach most popular in practice. The structure of the LFSR distributes all possible input bit streams evenly over all possible signatures.
- *Remainder left in the register (after completion of test) represents the signature*

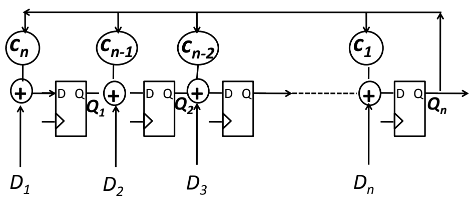


$G(x) = Q(x)P^*(x) + R(x)$

Initial state: $I(x) = 0$; Final state: $R(x)$

Multiple-Input LFSR as Signature Analyzers: MISR

- Used for multi-output circuits



LFSR as Signature Analyzers

- Signature analysis is the most popular method employed for test data compression because produces a small degree of masking
- One can decrease probability of masking by increasing the length of LFSR or changing the characteristic polynomial
- Functional registers are often modified to work as LFSR as well

Summary

- Compression techniques are widely used, esp. because their use in self-testing techniques
- All Boolean functions can be implemented by a circuit that is syndrome testable
- Signature analysis is the most popular test data compression technique due to low error masking probability