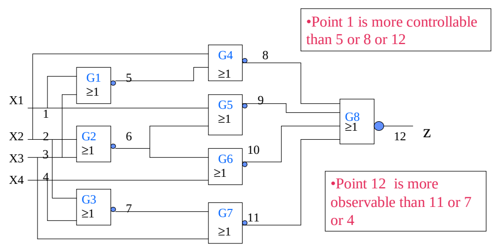
2B1423 ASIC Design using HDLs

F8 : Design for Testability

What is Testability?

- The extent to which a design can be tested for the presence of a variety of faults.
  - This feature of the circuit can be improved at design time
- Measure of Testability
  - Controllability: feature which allows controlling inputs of circuit elements from external ports or pins.
  - Observability: feature which allows observing the output of various elements from I/O ports.

Controllability and Observability: Example



Design for Testability (DFT)

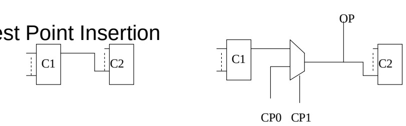
- To take into account the testing aspects during the design process so that testing is simpler and faster.
  - Reduce test effort
  - Reduce test time
  - Reduce cost of test equipment
  - Increase product quality
- Limitations
  - Hardware overhead (5-30%)
  - Performance degradation
  - Increase in design complexity

Fault-oriented Test Pattern Generation

- Combinational circuit with n inputs
  - 2^n possible patterns (exhaustive test)
  - Find minimum number of necessary input patterns
- Problem: Some nodes may not be testable because of circuit structure

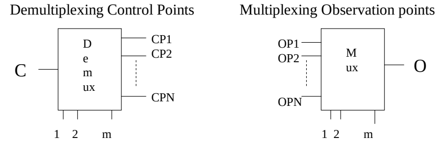
Ad hoc DFT Techniques

- Test Point Insertion



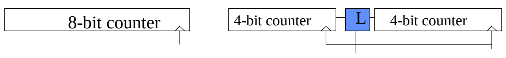
Demultiplexing Control Points

Multiplexing Observation points

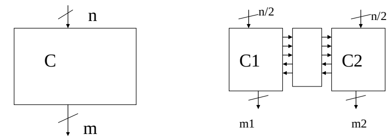


Ad hoc Techniques

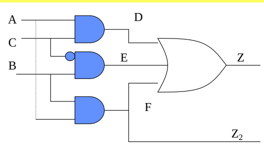
- Intialization
  - Design circuit so that it can be easily brought to known state
    - Master Reset
- External Clock
  - Mechanism to bypass internal clock and provide external clock
- Partitioning of counters and registers



Partitioning of Large Combinational Blocks



Undetectable faults' fix

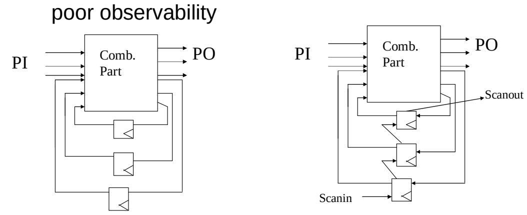


Solution:

Add internal values to propagate an internal value to an output

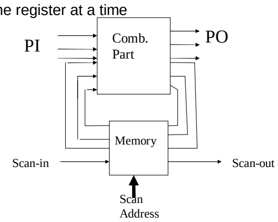
Scan Techniques

- Sequential Circuits have poor controllability and poor observability



Random Access Scan

- To achieve controllability and observability of all registers
  - Access one register at a time



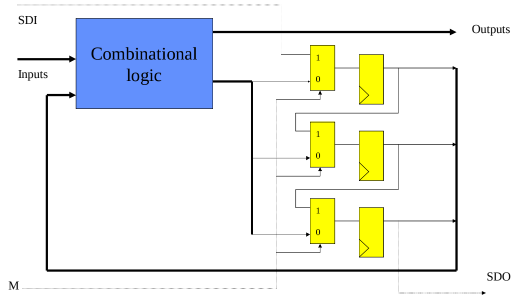
Testing Sequential Systems

- Structured design for test
  - Connect all flip-flops in the design together in a long serial scan-chain
  - Use a mux to select between Test-mode (M=1) and Normal mode (M=0)

Scan-path arrangement

- Full Scan
  - All flip-flops arranged in a single scan-path
- Partial Scan
  - Some flip-flops included in the scan path, some easily observable ones excluded
- Multi Scan paths
  - Flip-flops arranged in smaller chains to allow for parallel testing and thus reduced testing time

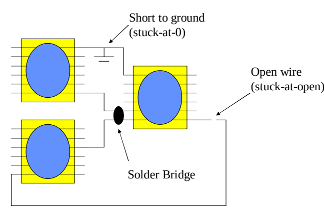
SISO Principle



SISO Principle

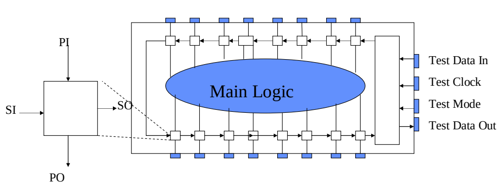
- Test sequential logic
  - Set M=1.
    - Test Flip-flops as shift-register.
    - Test with Pattern 00001100… (length the same as number of flip-flops). Tests transitions and stable flip-flops.
- Test combinational logic
  - Set M=1.
    - Shift in predetermined state (=test pattern) into the n flip-flops
  - Set M=0.
    - Apply a single clock cycle. Latch result into flipflops
  - Set M=1.
    - Shift out result and compare with simulations
    - At the same time, shift in a new pattern and repeat test

Circuit Board Faults



Board Level DFT: Boundary Scan

- Every input and output port of a chip is included in a special scan chain



Chip Architecture with Boundary Scan

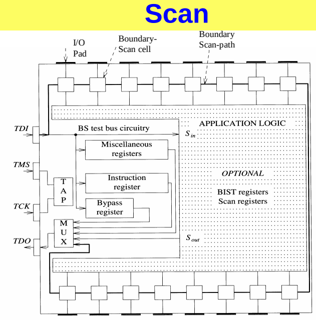
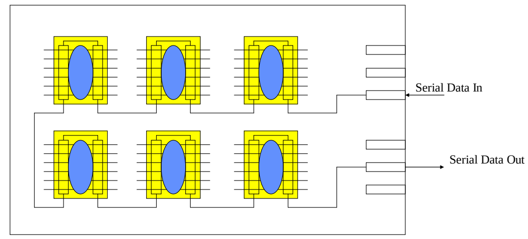


Figure 9.45 Chip architecture for IEEE 1149.1

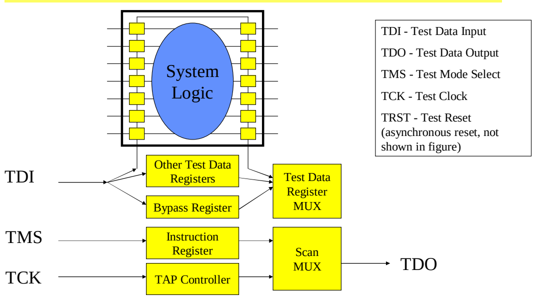
Boundary Scan (IEEE 1149.1)

- Used for testing PCBs
- Older “bed-of-nails” measuring no longer possible
  - Mounted ICs not testable
  - PCBs nowadays have more than 20 Layers => Impossible to measure intermediate layers
  - Component density is increasing. Multi-chip modules (MCMs) have unpackaged integrated circuits mounted directly on a silicon substrate.

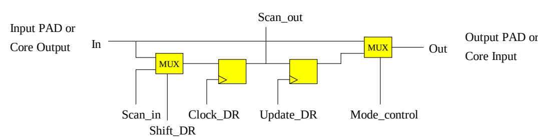
PCB with Boundary Scan



The JTAG (Joint Test Action Group) Standard



IEEE Standard 1149.1 - JTAG



Four Modes of operation:

1) Normal Mode - In to Out

2) Scan Mode - Select Scan_in; Capture Data (Clock_DR gated by TAP controller)

3) Capture Mode - Select In; Capture snapshot of System by pulsing Clock_DR

4) Update Mode - Clock captured data into Out_register (Update_DR gated by TAP) controller. TAP controller sets Mode_control according to instruction in Instruction Register.

VETENSKAP OCH KONST

KUNGL TEKNISKA HÖGSKOLAN

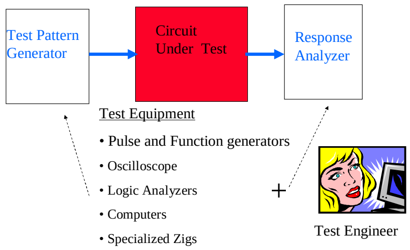
2B1423 ASIC Design using HDLs

Built in Self Test (BIST)

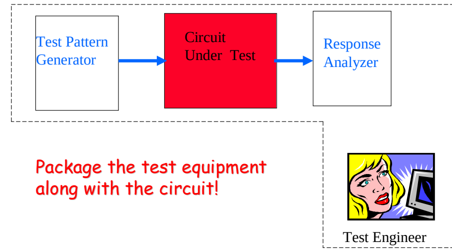
OUTLINE

- BIST Concepts
- Requirements for BIST
- Compression Techniques
  - One’s Count Compression
  - Transition Count Compression
  - Syndrome Testing
  - Linear Feedback Register as Signature Analyzer
- Test Generation for BIST
- Specific BIST Architectures
- Testing Sequential Circuits
  - Checking Experiment

Testing a Circuit



Self-Test a Circuit



BIST

- Simple to apply all possible combinations
  - Use a counter with the same number of flip-flops as the scan-path
  - Problem: Counters are complicated structures, and scan-path may be large.
  - Problem: number of answers large, requiring large pattern ROM for comparisons

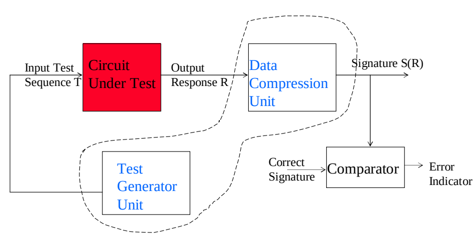
BIST: Requirements

- Generation of Test patterns for testing
  - DFT is very important
  - Algorithm has to be very simple and easy to implement in hardware
- Comparing Response
  - It is not possible to store the response in memory: area overhead
  - Related to Test Pattern Generation
- Compression of Test Patterns and Response is essential for BIST design

BIST : Basic Concepts

- BIST is a design technique in which some parts of the circuit are used for testing the circuit itself.
- Online BIST: Testing occurs during normal functional operating conditions.
  - Concurrent Online BIST: System doing normal functions
  - Non-Concurrent online BIST: System is idling
- Off-line BIST: System brought to a test mode.
  - Functional Off-line BIST: uses a high level functional model of the system
  - Structural Off-Line BIST: Uses structural model of system and detects structural faults

Test Generation and Response Compression



General Aspects of Compression

- A simple hardware implementation
- It should not slow down normal operations
- Good Compression
  - The signatures of good and faulty circuit should be different
  - Small size of signature. Signature size should be log of data size.
- Issues:
  - How to generate signatures for good circuit?

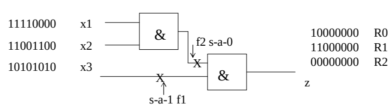
BIST - Implementation Alternatives

- Solution:
  - Find a structure that goes through all combinations (in any order) that is small and easy to implement.
  - Pseudo-random generators are easy to implement
  - Use signature analysis with small probability for errors instead of large ROMs (=high error probability) for pattern analysis

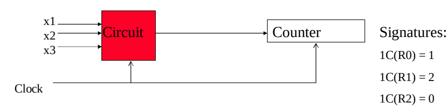
Ones-Count Compression

Assume a single output circuit C. Let the response of C to a test sequence be R = $r_1, r_2, ..., r_m$ .In ones counting the signature 1C(R) is the number of 1s appearing in R, i.e.

$$1C(R) = \sum_i r_i \quad \text{where } 0 \le 1C(R) \le m.$$



Ones Count Compression: Analysis



Masking Probability

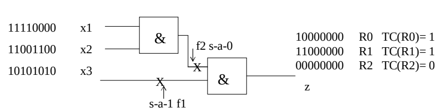
The probability of an erroneous output sequence having the same number of 1's as the correct sequence.

Theorem: The masking probability for ones-count compression for a combinational network circuit asymptotically approaches $(\pi m)^{-1/2}$.

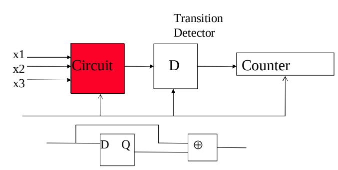
Transition-Count Compression

The Transition Count signature is the number of 0-1 and 1-0 transitions in the output response sequence R. Let the response of C to a test sequence be R = $r_1, r_2, ..., r_m$. In transition counting the signature TC(R) is

$$TC(R) = \sum_{i=1} (r_i \oplus r_{i+1})$$



Transition Count Compression: Analysis

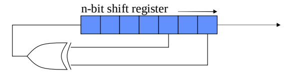


Theorem: The masking probability for transition count compression for a combinational network circuit asymptotically approaches $(\pi m)^{-1/2}$.

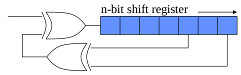
BIST Structures

- LFSR
  - Pseudo-random sequences of length 2^n-1 (the combination all 0's not covered)
- SISR
  - After n cycles, the register holds the residue from a modulo-2 division => stream compression, produces good signature
- MISR
  - Small, compresses multiple inputs into a signature.
  - Probability that a faulty circuit produces correct signature (tends to 2^{-n} for n-stage register and long test-sequences)

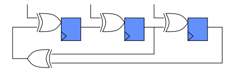
BIST Structures



Linear Feedback Shift Register (LFSR)

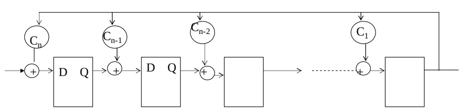


Single Input Signature Register (SISR)



Multiple Input Signature Register (MISR)

LFSR as a signature analyzer



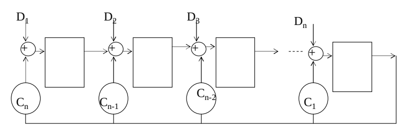
⊕ : Modulo-2 addition (an ex-or gate)

C_1 : If C_1=1 then there is a connection otherwise no connection

This Circuit can be used to compress the response with masking probability = 2^{-n} .

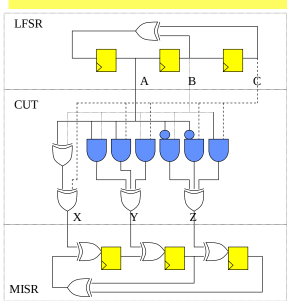
Multiple-Input Signature Registers

- Method 1: Time Multiplex the signature analyzer for each output. That is, repeat the test sequence for each output. This will require long test time.
- Method 2: Use Multiple Input Signature Register (MISR).



Error Masking Probability = 2^{-n}

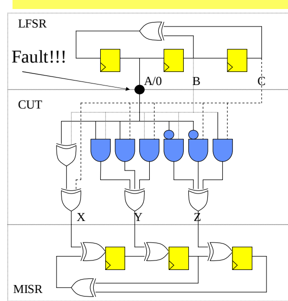
Example



LFSR output *abc*	CUT Output *xyz*	MISR
111	111	111
011	011	100
001	101	001
100	100	001
010	101	000
101	010	101
110	010	100
111	111	**000**

Correct Signature

Example

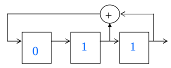


LFSR output *abc*	CUT Output *xyz*	MISR
111	011	111
011	011	000
001	101	011
100	000	100
010	101	010
101	101	000
110	101	101
111	011	**011**

Faulty Signature!!!

NB! *Aliasing* Problem:
A/1 => 000 (Error probability 2^{-3})

Signature Analysis

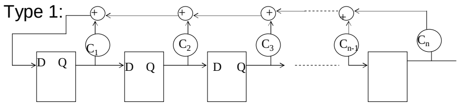


S_0	0	1	1
S_1	0	0	1
S_2	1	0	0
S_3	0	1	0
S_4	1	0	1
S_5	1	1	0
S_6	1	1	1
$S_7 = S_0$	0	1	1

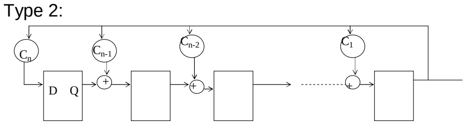
- Period of Linear Feed back shift register depends upon
  - Feed back connections
  - Intial value of register
- Masking probability is inversely related to the period

Types of Linear Feed Back Shift Register

- Type 1:



- Type 2:



Test Pattern Generation for BIST

Summary

Exhaustive Testing

§ Exhaustive Test Pattern Generators

Pseudo-Random testing

§ Weighted test generator

§ Adaptive Test Generator

Pseudo-Exhaustive testing

§ Constant Weight Counter

§ Combined LFSR and Shift Register

§ Combined LFSR and XOR

§ Cyclic LFSR

Exhaustive Testing

- Apply all possible 2^n test patterns to a n input circuit.
- A binary counter can be used
- Or a maximum length autonomous LFSR can be used (add special circuit to generate all zeros).
- Normally applied when number of inputs are less than 25.

Pseudorandom Testing

- Patterns are generated deterministically but have characteristics of random patterns.
- Pseudo-random patterns without replacement can be generated using autonomous LFSRs.
- Test length can be chosen to achieve the desired fault coverage. This can be determined using fault simulation.
- Limitation
  - LFSR tend to produce patterns with equal numbers of 1's and 0's
  - For some circuit it is better to bias the distribution of 0's and 1's of the pattern generator. For example testing of AND requires more 1's in the test pattern.

Weighted Test Generation

- Generate Test Patterns with the desired distribution of 1's and 0's.
- Such a generator can be constructed using a LFSR and a combinational circuit.
- Different parts of the circuit may require different distribution of 0's and 1's.

Adaptive Test Generation

- Also employs weighted test pattern generation.
- Fault simulation is used to determine weights for various faults.
- Different distributions are used for different class of faults.
- A Test Pattern Generator (TPG) is designed to produce the required distributions.
- Advantage: Small test lengths
- Disadvantage: Costly TPG hardware

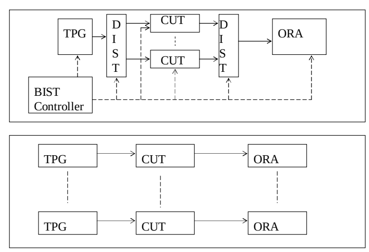
Pseudo-exhaustive Testing

- Requires fewer tests but has the advantage of exhaustive testing.
- Segments the circuit into parts and each segment is exhaustively tested.
- Logical Segmentation
  - Cone Segmentation
  - Sensitized path segmentation
- Physical Segmentation

Generic Off-line BIST Architectures

- Classification
  - Centralized or Distributed BIST circuitry
  - Embedded or Separate BIST
- Key Elements of BIST
  - Test pattern Generator (TPG)
  - Out-put Response Analyzers (ORA)
  - Circuit Under Test (CUT)
  - A Distribution system for transmitting data from TPG to CUT and from CUT to ORAs.
  - A BIST Controller

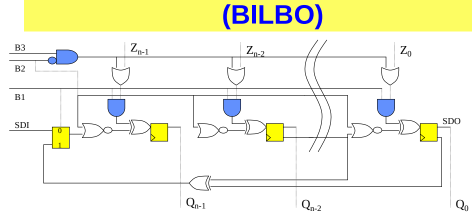
Generic BIST Architectures



Build-In Logic Block Observation (BILBO)

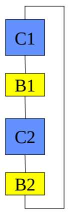
- Motivation: It is difficult to self test a circuit which has a large number of (Inputs+Outputs+Storage Cells).
  - You require Pseudo-Random Signal Generation (PRSG) for inputs, MISR for outputs PRSG, and MISR for storage cells.
  - Solution: Clustering of storage cells into registers and make these registers carry out multiple jobs.
- BILBO is one such scheme.
  - Registers can be configured for :
    - **Normal Operation**
    - **As PRSG**
    - **As MISR**
    - **As Scan Register**

Built-In Logic Block Observation (BILBO)



B1	B2	B3	Mode
1	1	-	Normal
0	1	-	Reset
1	0	0	Signature Analysis MISR
1	0	1	Test Pattern generation LFSR
0	0	-	Scan

BILBO - Circuit partitioning for self-test



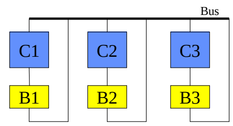
Serial Partitioning

Replace registers with BILBO's.

To test C1, set B1 to MISR and B2 to LSFR.

To test C2, set B2 to MIST and B1 to LSFR

BILBO - Circuit partitioning for self-test



Parallel Partitioning

Replace registers with BILBO's.

To test C1, set B1 to MISR and B2 to LSFR.

To test C2, set B2 to MISR and B3 to LSFR.

etc.

Conclusions

- BIST of a system requires techniques for
  - Data Compression
  - Test Pattern Generation
- Linear Feedback Shift Registers are good for doing both
- Important concerns
  - Masking Probability
  - Fault coverage
  - Overhead
- Data Compression and Test Generation Functions can be combined in one register

How do we know that the Test Circuitry is OK?

- We don't, otherwise we would need test circuitry for the test circuitry and so on, *ad infinitum*
  - Initialize the scan Flip-flops using the Reset function
  - Use the scan path to test the Flip-flops
  - Test the rest of the circuit

Further Reading

- M. Abramovici, M.A. Breuer and A.D. Friedman (1990)
  Digital System Testing and Testable Design
  IEEE Press
- A. Miczo (1987)
  Digital Logic Testing and Simulation
  John Wiley & Sons, New York.
- B.R. Wilkins (1986)
  Testing Digital Circuits
  Van Nostrand Reinhold (UK).

Further Reading

- FPGA manufacturers' Web sites
- HP offers an online BSDL (Boundary Scan VHDL)
  - IEEE Standard 1149.1 - 1994
  - To be honest, the BSDL standard contains the only example I've ever seen using the linkage mode on VHDL ports (a VDD and a GND port)