# Introduction to Design-For-Test (DFT) and Built-in Self Test (BIST)

## Motivation:

- Design of complex digital systems requires that considerable attention be devoted to assuring that the system can be tested.
- In the past, when systems were not as complex, testing was treated as an afterthought (addressed only after the design was complete).
- Today, such a strategy will likely result in an untestable (and possibly faulty) network. We must therefore "design for test" as we conceive the network.
- Furthermore, the cost and time associated with thorough testing of complex systems requires that we include some "built-in self test" capabilities.

## Outline:

- Stuck-at Fault Model
- Test Generation time
- The Cost of Testing
- "Testability" "Controllability" "Observability"
- Ad Hoc DFT (Partitioning, Test Points, Bus Architecture, Signature Analysis)
- Structured DFT (LSSD, Scan Path, Scan/Set Logic, Random Access Scan)
- Built-in Self Test using BILBO

## Stuck-At Fault Model:

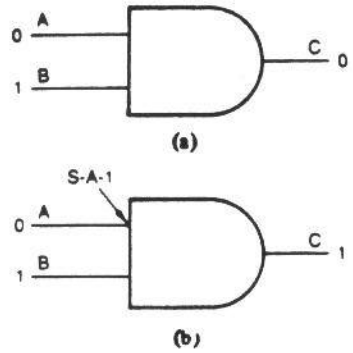



**Fig. 1. Test for input stuck at fault. (a) Fault-free AND gate (good machine). (b) Faulty AND gate (faulty machine).**

S-A-1 => Stuck-at One
S-A-0 => Stuck-at Zero

For combinational logic, the process of developing test patterns is fairly well automated (ATPG).

Time for test generation is proportional to $N^3$ where N is the number of gates.

For sequential logic, test pattern generation is much more complex (the best approach is to break feedback paths during test to reduce the network to combinational logic).

The cost of test application: increases by an order of magnitude (or more) for each level of system integration. i.e. \$0.30 at chip-level => \$3.00 at the board level => \$30 at the system level.

# Ad Hoc DFT

## Partitioning

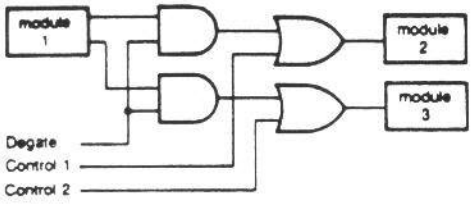



Fig. 2. Use of degating logic for logical partioning.

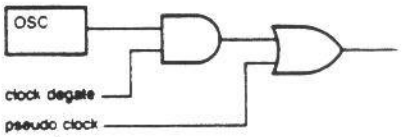



Fig. 3. Degating lines for oscillator.

## Test Point Insertion

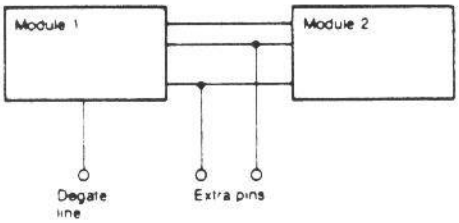



Fig. 4. Test points used as both inputs and outputs.

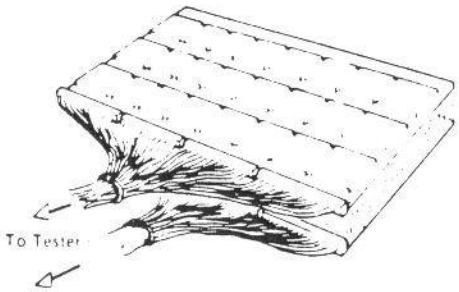



Fig. 5. "Bed of Nails" test.

# Ad Hoc DFT

## Bus Architecture

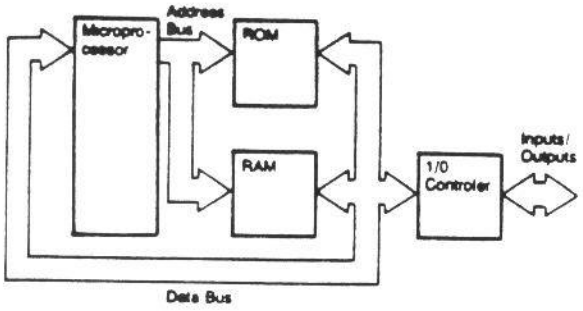



**Fig. 6. Bus structured microcomputer.**

## Signature Analysis

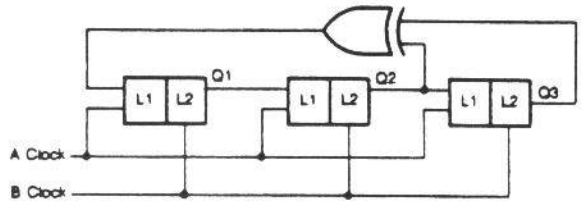



**Fig. 7. Counting capabilities of a linear feedback shift register.**

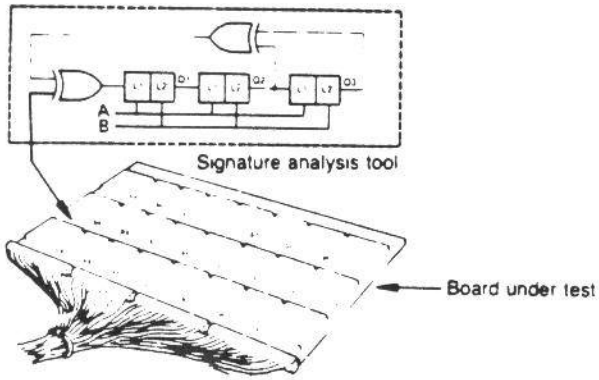



**Fig. 8. Use of signature analysis tool.**

# Structured DFT

The classical model of a sequential network with scan:

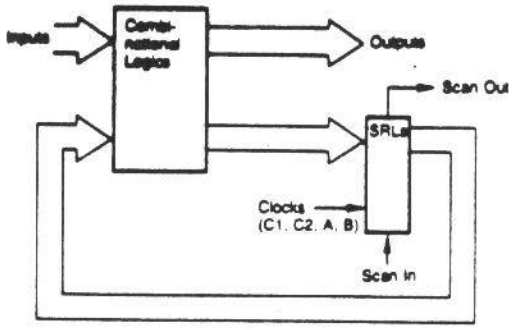



**Fig. 9. Classical model of a sequential network utilizing a shift register for storage.**

Level-Sensitive Scan Design (LSSD):

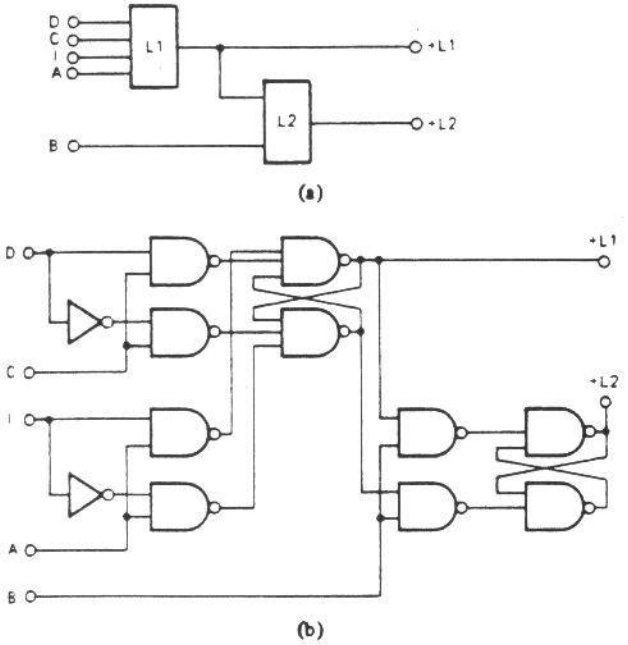



**Fig. 10. Shift register latch (SRL). (a) Symbolic representation. (b) Implementation in AND-INVERT gates.**

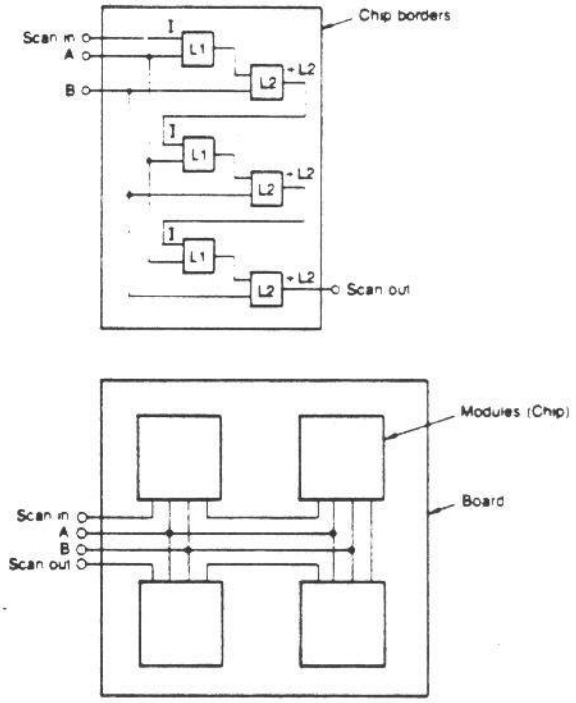



**Fig. 11. Interconnection of SRL's on an integrated circuit and board.**

# Structured DFT

## Scan Path (using a Raceless D-type Flip Flop):

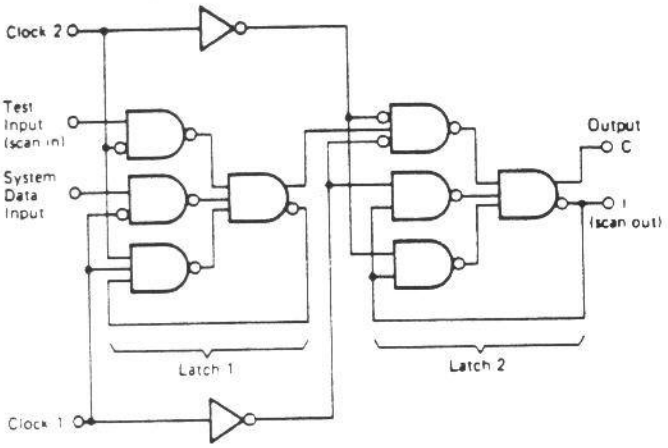



Fig. 13. Raceless D-type flip-flop with Scan Path.

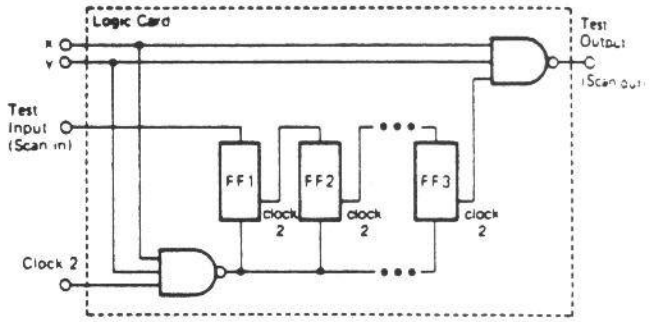



Fig. 14. Configuration of Scan Path on Card.

(Clock 2 is high during normal operation)

## Scan/Set Logic:

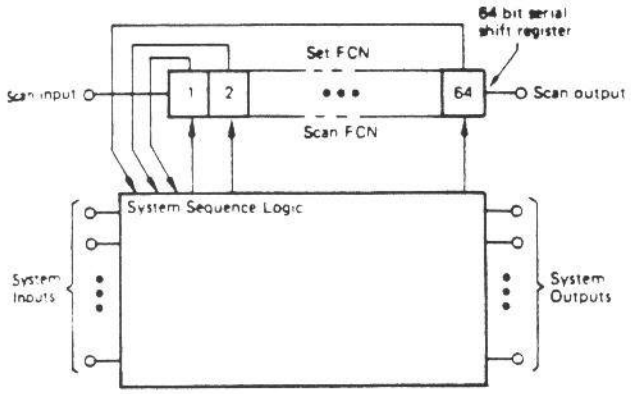
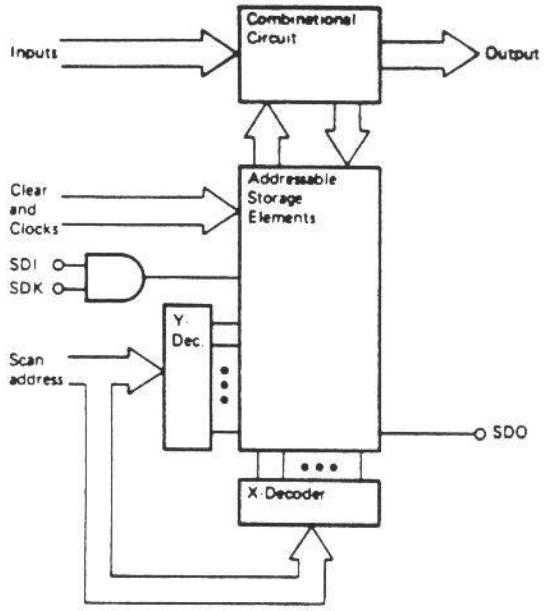



Fig. 15. Scan/Set Logic (bit-serial).

# Structured DFT

## Random-Access Scan:




## Built-in Logic Block Observation (BILBO) for BIST:

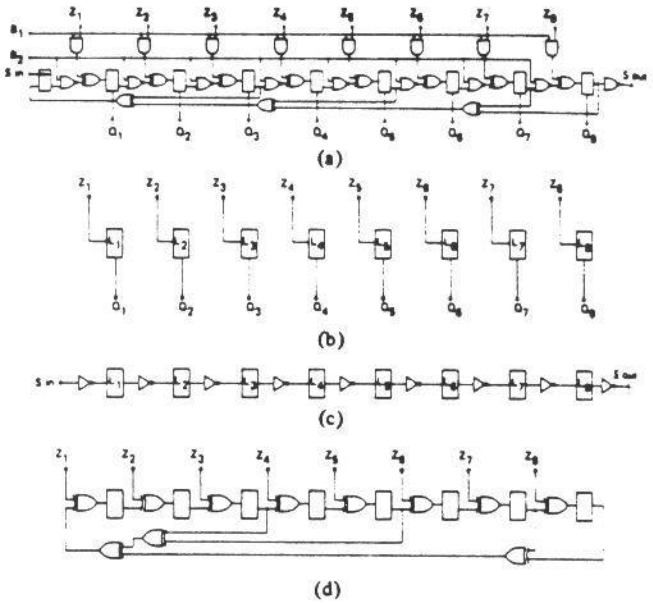



Fig. 19. BILBO and its different modes. (a) General form of BILBO register. (b) $B_1B_2 = 11$ system orientation mode. (c) $B_1B_2 = 00$ linear shift register mode. (d) $B_1B_2 = 10$ signature analysis register with $m$ multiple inputs ($Z_1, Z_2, \cdots, Z_8$).

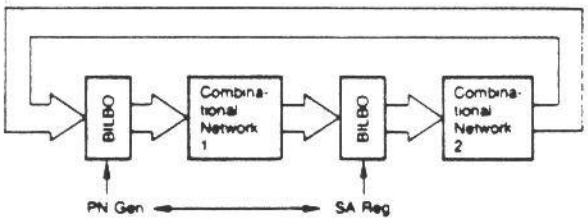



Fig. 20. Use of BILBO registers to test combinational Network 1.