CPE/EE 422/522
Spring 2004
Chapter 10 - Hardware Testing and Design for Testability

Dr. Rhonda Kay Gaede

UAH

Motivation for Testing

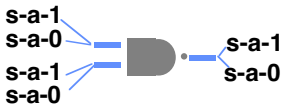
- Testing during design process
  - —use VHDL test benches to verify that the _____ used are correct
  - —verify _____ after synthesis
- Post-fabrication testing
  - —when a digital system is manufactured, test to verify that it is free from _____ _____
  - —today, cost of testing is _____ component of the manufacturing cost
  - —efficient techniques are needed to test and design digital systems so that they are easy to test

10.1 Testing Combinational Logic

- Common types of errors
  - _____
  - _____
- If the input to a gate is shorted to ground, the input acts as if it is stuck at _____
  - s-a-0 (stuck-at-0) faults
- If the input to a gate is shorted to positive supply voltage, the input acts as if it is stuck at _____
  - s-a-1 (stuck-at-1) faults

10.1 Testing Combinational Logic - Stuck-at Faults

- How many single stuck-at faults
  - _____ where n is the number of inputs

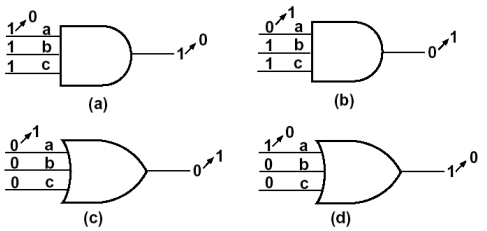


- We will assume
  - that there is _____ active at a time in the whole circuit
  - SSF single stuck-at fault

10.1 Testing Combinational Logic - Stuck-at Faults for AND and OR gates

Test *a* for s-a-0

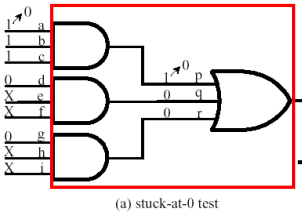
Test *a* for s-a-1



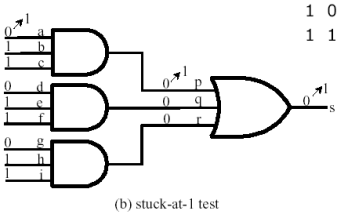
Test *a* for s-a-1

Test *a* for s-a-0

10.1 Testing Combinational Logic - Testing an AND-OR Network



(a) stuck-at-0 test

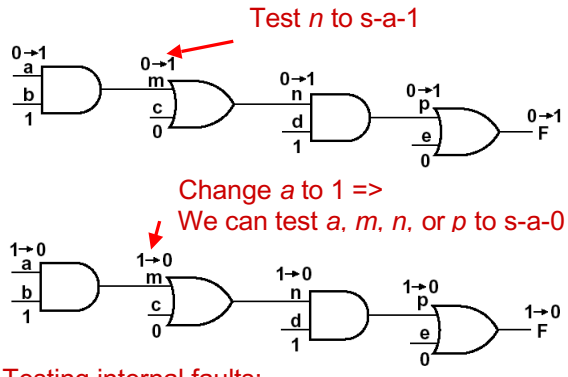


(b) stuck-at-1 test

a	b	c	d	e	f	g	h	i	Faults Tested
1	1	1	0	X	X	0	X	X	a0, b0, c0, p0
0	X	X	1	1	1	0	X	X	d0, e0, f0, q0
0	X	X	0	X	X	1	1	1	g0, h0, i0, r0
0	1	1	0	1	1	0	1	1	a1, d1, g1, p1, q1, r1
1	0	1	1	0	1	1	0	1	b1, e1, h1, p1, q1, r1
1	1	0	1	1	0	1	1	0	c1, f1, i1, p1, q1, r1

BRUTE-FORCE testing:
apply 2^9=512 different input combinations and check the output

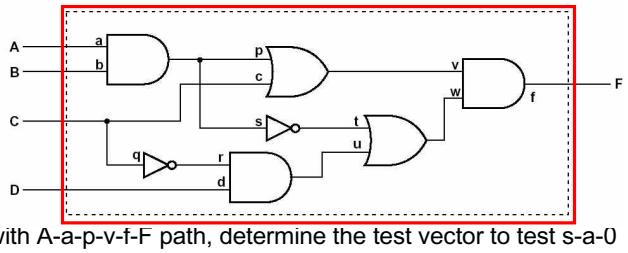
10.1 Testing Combinational Logic - Path Sensitization Example



Testing internal faults:
choose a set of inputs that will excite the fault and then propagate the fault to the network output

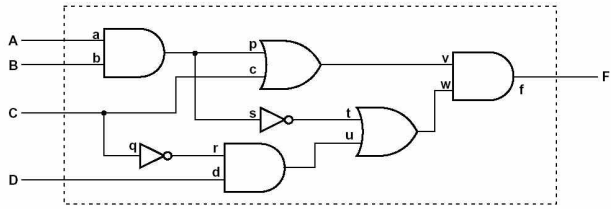
10.1 Testing Combinational Logic - Obtaining a Test Set

- What is a minimum set of test vectors to test the network below for all stuck-at-1 and stuck-at-0 faults?



Start with A-a-p-v-f-F path, determine the test vector to test s-a-0
determine the list of faults covered
select an untested fault, determine the required ABCD inputs
determine the additional faults tested
repeat the process until all faults are covered

10.1 Testing Combinational Logic - Obtaining a Test Set (continued)



	0	1
a	+	&
b	+	*
c	*	&
d	+	%
p	+	*
q	#	+
r	+	#
s	#	*
t	*	#
u	+	#
v	+	&
w	+	#
f	+	#

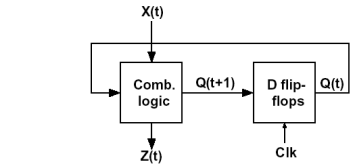
- Step 1: A-a-p-v-f-F, s-a-0
  - ABCD: 1101 (+)
- Step 2: s-a-0 for *c*
  - C=1, p=0, w=1 => ABCD=1011 (*)
- Step 3: s-a-0 for *q*
  - C=1, D=1, t=0, s=1 => ABCD=1111 (#)
- Step 4: s-a-1 for *a*
  - A=0, B=1, C=0, D=1 => ABCD=0101 (&)
- Step 5: s-a-1 for d (%)
  - D=0, C =0, t=1 => ABCD = 1100

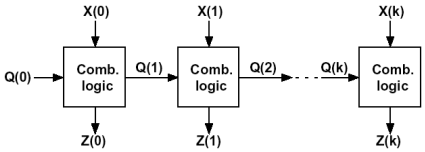
10.2 Testing Sequential Logic

- In general, much more difficult than testing combinational logic since we must use _____
  - typically we can observe inputs and outputs, not the _____
  - assume the _____ input, so we can reset the network to the _____
- Test procedure
  - reset the network to the initial state
  - apply a test sequence and observe the output sequence
  - if the output is correct, repeat the test for another sequence
- How many test sequences do we have?
  - how do we test that the initial state of the network under test is equivalent to the initial state of the correct network?
  - what is the sequence length?

10.2 Testing Sequential Logic - Magnitude of the Problem

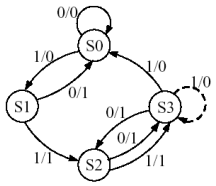
- In practice, if the network has N or fewer states, then apply only input sequences of length less than or equal 2N-1
- Example
  - consider a network which includes 5 inputs, 1 output, and 4 states
  - total number of test sequences: $(2^5)^7 = 2^{35}$ => infeasible (!)
  - derive a small set of test sequences that will adequately test a SN





10.2 Testing Sequential Logic - Distinguishing States

- Consider input sequence
  - X = 0 1 0 1 1 0 0 1 1
  - Output sequence Z = 0 0 1 0 1 1 1 1 0
  - If we change the network S3->S0 => S3->S3, the output sequence will be the same
- Find distinguishing sequence
  - an input sequence that will distinguish each state from the other states

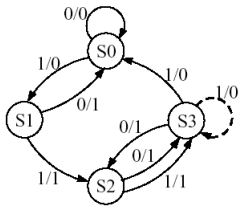


Q1Q2	State	Next State X=0	Next State 1	Output X=0	Output 1
00	S0	S0	S1	0	0
10	S1	S0	S2	1	1
01	S2	S3	S3	1	1
11	S3	S2	S0	1	0

Input sequence: X=11

- S0: Z = 01
- S1: Z = 11
- S2: Z = 10
- S3: Z = 00

10.2 Testing Sequential Logic - Testing Each State Transition



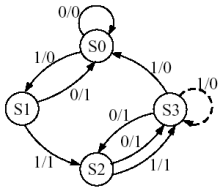
Q1Q2	State	Next State X=0	Next State 1	Output X=0	Output 1
00	S0	S0	S1	0	0
10	S1	S0	S2	1	1
01	S2	S3	S3	1	1
11	S3	S2	S0	1	0

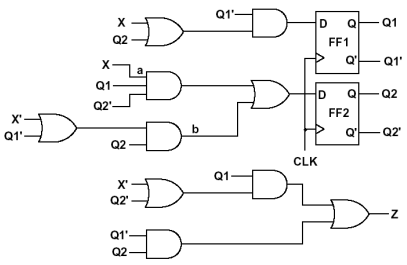
Verify each entry in the table using the following sequences:

Input	Output	Transition Verified
R 0 1 1	0 0 1	(S0 to S0)
R 1 1 1	0 1 1	(S0 to S1)
R 1 0 1 1	0 1 0 1	(S1to S0)
R 1 1 1 1	0 1 1 0	(S1 to S2)
R 1 1 0 1 1	0 1 1 0 0	(S2 to S3)
R 1 1 1 1 1	0 1 1 0 0	(S2 to S3)
R 1 1 0 0 1 1	0 1 1 1 1 0	(S3 to S2)
R 1 1 0 1 1 1	0 1 1 0 1 0	(S3 to S0)

10.2 Testing Sequential Logic - Testing Each State Transition (cont'd)

- Implementation of the FSM
  - S0=00, S1=10, S2=01, S3=11
- Test *a* for s-a-1
  - to do this Q1Q2 must be 10 => go to the state S1 and then set X to 0 (R10)
  - in normal operation, the next state will be S0; if a is s-a-1 then next state is S2
  - distinguish the state (S0 or S2); apply sequence 11
  - Final sequence: R1011
    Normal output: 0101
    Faulty output: 0110



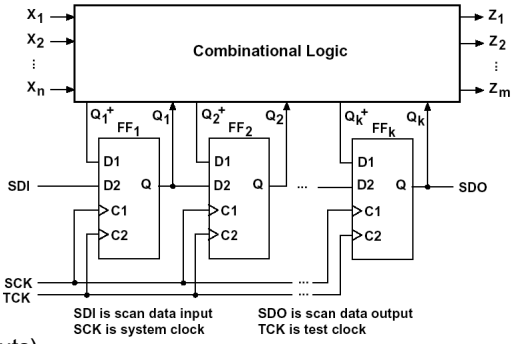


10.3 Scan Testing

- Testing of sequential networks is greatly simplified if we can observe the state of _____ in addition to observing the network outputs
  - —Connect the output of each flip-flop to one of the IC pins?
  - —Arrange flip-flops to form a _____ => shift out the state of flip-flops bit by bit using a single serial output pin => Scan path testing

10.3 Scan Path Testing - Transforming From Sequential to Combinational

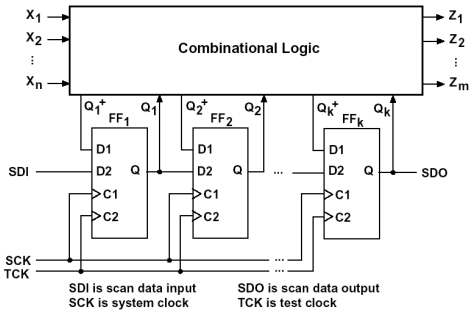
- Sequential network is separated into a combinational logic part and a state register composed of flip-flops



- Two ports FFs (2 D inputs and 2 clock inputs)
  - — D1 is stored in the FF on C1 pulse
  - — D2 is stored in the FF on C2 pulse
  - — Q of each FF is connected to D2 of the next FF to form a shift register

10.3 Scan Path Testing - Modes of Operation

- Normal operation
  - system clock SCK = C1
  - inputs: $X_1X_2...X_N$
  - outputs: $Z_1Z_2...Z_N$
- Testing
  - FFs are set to a specified state using the SDI and TCK
  - test vector is applied $X_1X_2...X_N$
  - outputs $Z_1Z_2...Z_N$ are verified
  - SCK is pulsed to take the network to the next state
  - next state is verified by pulsing the TCK to shift the state code out of the scan register via SDO

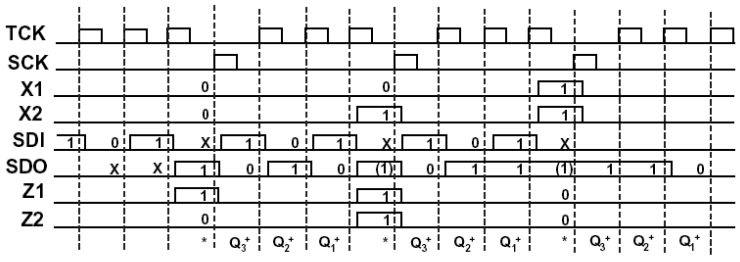


10.3 Scan Path Testing - An Example

- SQ: X_1X_2, $Q_1Q_2Q_3$, Z_1Z_2

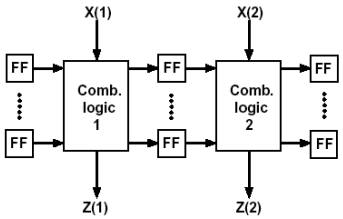
One row of the state transition table:

$Q_1Q_2Q_3$	$Q_1^+Q_2^+Q_3^+$ X_1X_2 = 00	01	11	10	Z_1Z_2 00	01	11	10
101	010	110	011	111	10	11	00	01

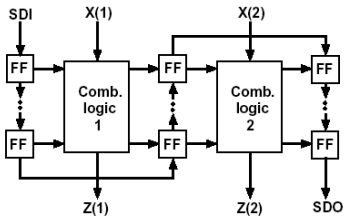


* Read output (output at other times not shown)

10.3 Scan Path Testing - Scan Chain

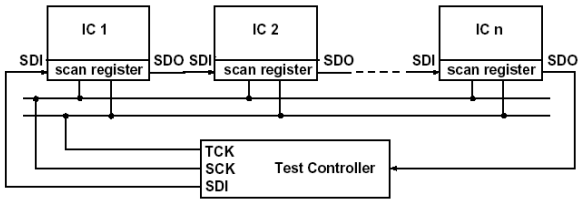


(a) Without scan chain



(b) With scan chain added

10. 3 Scan Path Testing - Connecting Multiple ICs

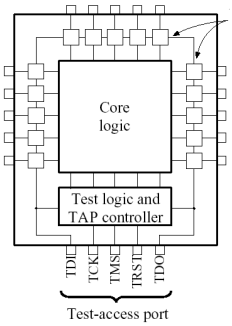


10.4 Boundary Scan - Motivation and Origin

- PCB testing has become more difficult
  - ICs have become more _____, with more and more _____
  - PCBs have become denser with _____ and _____
  - Bed-of-nails testing
    - use sharp probes to contact the traces on the board
    - test data are applied to and read from various ICs
    - => not practical for high-density PCBs with fine traces and complex ICs
- Boundary scan test methodology:
  introduced to facilitate the testing of complex PC boards
  - developed by JTAG (_____)
  - adopted as ANSI/IEEE Standard _____ — Standard Test Access Port and Boundary Scan Architecture
  - IC manufacturers make ICs that _____ to the standard
  - ICs can be _____ on a PCB, so that they can be tested using only _____

10.4 Boundary Scan - Boundary Scan Register

- Boundary Scan Register (BSR) —cells of the BSR are placed between input or output pins and the internal core logic
- Four or five pins of the IC are devoted to the test-access-port (TAP)

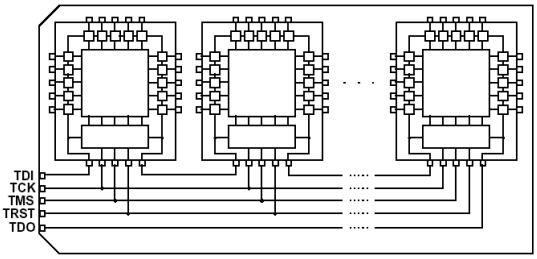


Boundary scan cells

TAP pins

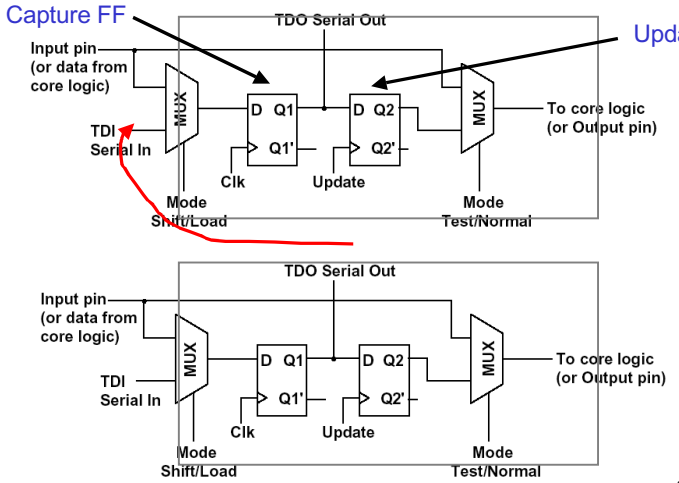
- TDI —
- TCK —
- TMS —
- TDO —
- TRST —

10.4 Boundary Scan - PCB with Boundary Scan ICs

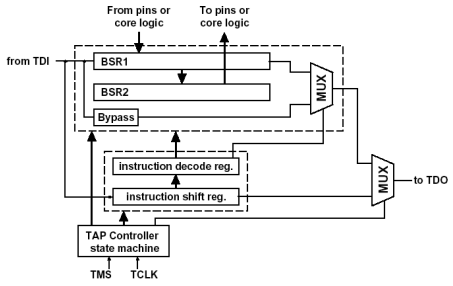


- BSRs in the ICs are linked together serially in a single chain with input TDI and output TDO.
- TCK, TMS, TRST are connected in parallel to all of the ICs.

10.4 Boundary Scan - Boundary Scan Cell

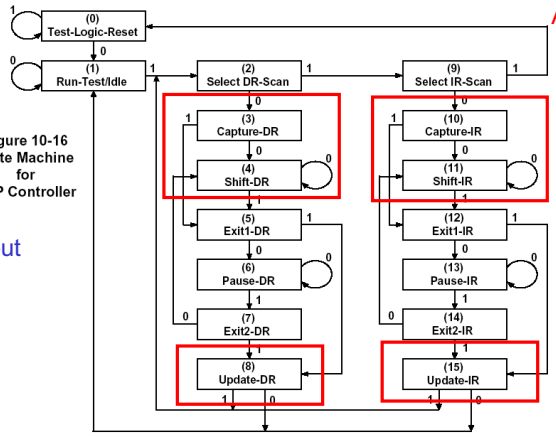


10.4 Boundary Scan - Basic Architecture



- BSR1 —shift register, which consists of the Q1 flip-flops in the boundary scan cells
- BSR2 —represents the Q2 flip-flops; can be parallel loaded from BSR1 when an update signal is received
- TDI can be shifted into the BSR1, through a bypass register, or into the ISR

10.4 Boundary Scan - TAP Controller State Diagram



Affect ASIC core

Figure 10-16 State Machine for TAP Controller

TMS is input

10.4 Boundary Scan - TAP Controller Operation

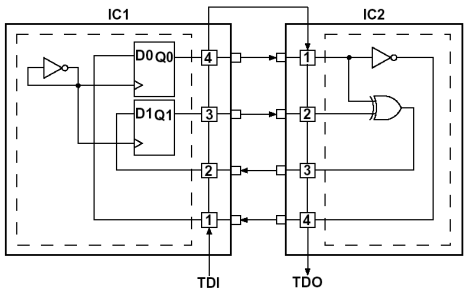
- TAP Controller
  - 16 state FSM
  - Change states depending on TMS and TCK
  - Output: signals to control the test data registers and instruction register (including serial shift clocks and update clocks)
- Test-logic-reset is the initial state; on a low TMS go to Run-Test/Idle state
- TMS: 1100 => Shift-IR
- In Shift-IR command is shifted in through TDI port
-

10.4 Boundary Scan - Instructions in the IEEE Standard

- BYPASS: allows the TDI serial data to go through 1-bit bypass register on the IC instead of through the BSR1.
- SAMPE/RELOAD: used to scan the BSR without interfering with the normal operation of the core logic. Samples of this data can be taken and scanned out through the BSR. Test data can be shifted into the BSR.
- EXTEST: allows board-level interconnect testing and testing of clusters of components which do not incorporate the boundary scan test features. Test data is shifted into the BSR and then it goes to the output pins. Data from the input pins is captured by the BSR.
- INTEST (optional): this instruction allows testing of the core logic by shifting test data into the boundary-scan register. Data shifted into the BSR takes the place of data from the input pins, and output data from the core logic is loaded into the BSR.
- RUNBIST (optional): this instruction causes special built-in self-test (BIST) logic within the IC to execute.

10.4 Boundary Scan - Interconnection Testing

Test PC board traces between IC1 and IC2



- Test the connections between two ICs.
- IC1: 2 input pins, 2 output pins.
- IC2: 2 input pins, 2 output pins.
- Test data is shifted into the BSRs via TDI.
- Data from the input pins is parallel-loaded into the BSRs and shifted out via TDO.

Assume:
IR on each IC is 3 bits long with
EXTEST coded as 000
SAMPLE/PRELOAD as 001

10.4 Boundary Scan - Interconnection Testing Steps

1. Reset the TAP state machine to the Test-Logic-Reset state by inputting a sequence of five 1's on TMS. The TAP controller is designed so that a sequence of five 1's will always reset it regardless of the present state. Alternatively, TRST could be asserted if it is available.

2. Scan in the SAMPLE/PRELOAD instruction to both ICs using the sequences for TMS and TDI given below.

State:	0	1	2	9	10	11	11	11	11	11	11	12	15	2
TMS:	0	1	1	0	0	0	0	0	0	0	1	1	1	
TDI:	———		—		1	0	0	1	0	0	—	—		

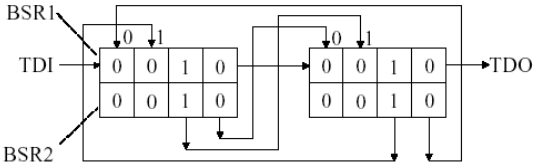
The TMS sequence 01100 takes the TAP controller to the Shift-IR state. In this state, copies of the SAMPLE/PRELOAD instruction (code 001) are shifted into the instruction registers on both ICs. In the Update-IR state, the instructions are loaded into the instruction decode registers. Then the TAP controller goes back to the Select DR-scan state.

10.4 Boundary Scan - Interconnection Testing Steps (cont'd)

¥ **3.** Preload the first set of test data into the ICs using the sequences for TMS and TDI given below.

State: 2 3 4 4 4 4 4 4 4 4 5 8 2
TMS: 0 0 0 0 0 0 0 0 0 1 1 1
TDI: ——0 1 0 0 0 1 0 0 ——

Data is shifted into BSR1 in the Shift-DR state, and it is transferred to BSR2 in the Update-DR state. The result is as follows:



10.4 Boundary Scan - Interconnection Testing Steps (cont'd)

¥ 4. Scan in the EXTEST instruction to both ICs using the following sequences:

State: 2 9 10 11 11 11 11 11 11 12 15 2
TMS: 1 0 0 0 0 0 0 0 1 1 1
TDI: ——— 0 0 0 0 0 0 — —

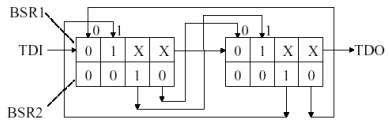
The EXTEST instruction (000) is scanned into the instruction register in state Shift-IR and loaded into the instruction decode register in state Update-IR. At this point, the preloaded test data goes to the output pins, and it is transmitted to the adjacent IC input pins via the printed circuit board traces.

10.4 Boundary Scan - Interconnection Testing Steps (cont'd)

¥ 5. Capture the test results from the IC inputs. Scan this data out to TDO and scan the second set of test data in using the following sequences:

State: 2 3 4 4 4 4 4 4 4 4 5 8 2
TMS: 0 0 0 0 0 0 0 0 0 1 1 1
TDI: ——1 0 0 0 1 0 0 0 ——
TDO: ——x x 1 0 x x 1 0 ——

The data from the input pins is loaded into BSR1 in state Capture-DR. At this time, if no faults have been detected, the BSRs should be configured as shown below, where the X's indicate captured data which is not relevant to the test.



The test results are then shifted out of BSR1 in state Shift-DR as the new test data is shifted in. The new data is loaded into BSR2 in the Update-IR state.

10.4 Boundary Scan - Interconnection Testing Steps (cont'd)

¥ 6. Capture the test results from the IC inputs. Scan this data out to TDO and scan all 0's in using the following sequences:

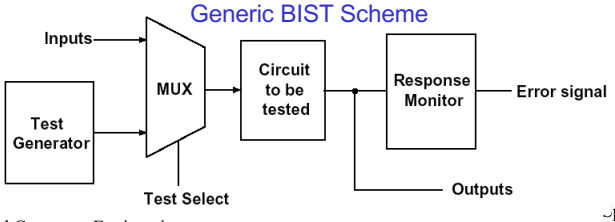
State: 2 3 4 4 4 4 4 4 4 4 5 8 2 9 0
TMS: 0 0 0 0 0 0 0 0 0 1 1 1 1 1
TDI: ——0 0 0 0 0 0 0 0 ————
TDO: ——x x 0 1 x x 0 1 ————

The data from the input pins is loaded into BSR1 in state Capture-DR. Then it is shifted out in state Shift-DR as all 0's are shifted in. The 0's are loaded into BSR2 in the Update-IR state. The controller then returns to the Test-Logic-Reset state and normal operation of the ICs can then occur. The interconnection test passes if the observed TDO sequences match the ones given above.

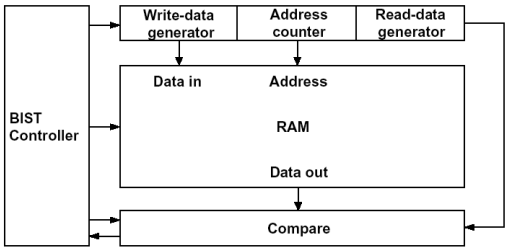
10.5 Built-In Self-Test

- Add logic to the IC so that it can test itself
  - — Built-In Self-Test — BIST
- Using BIST
  - — when test mode is selected by the test-select signal, an on-chip test generator applies test patterns to the circuit under test
  - — the resulting outputs are observed by the response monitor, which produces an error signal if an incorrect output is detected

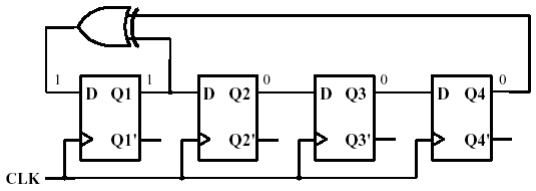
Generic BIST Scheme



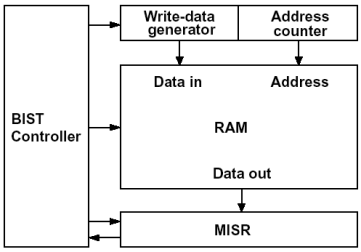
10.5 Built-in Self-Test - Circuit for RAM



10. 5 Built-in Self-Test - Linear Feedback Shift Registers(LFSR)



10.5 Built-in Self-Test - Circuit for RAM with Signature



MISR —Multiple Input Signature Register

E.g. for MISR –form a check-sum by adding up all data bytes stored in the RAM