ECE 538: VLSI System Testing

Krish Chakrabarty

Solutions to Homework 3

Problem 1

Note: the solution of problem 1 is prepared by Yang Zhao

(a) Testing time for a full-scan circuit with *n* flip-flops, *m* scan chains, and *p* test patterns. The clock frequency is *f*.

$$T = \{(\left\lceil \frac{n}{m} \right\rceil + 1) \bullet p + \left\lceil \frac{n}{m} \right\rceil\} \bullet \frac{1}{f}$$

$$= \{(\left\lceil \frac{100,000}{512} \right\rceil + 1) \bullet 10^6 + \left\lceil \frac{100,000}{512} \right\rceil\} \bullet \frac{1}{10^8} = 1970.00196 \text{ ms}$$

(b) The test application scheme can be selected as Launch-off-shift (LOS) or Launch-off-capture (LOC). Testing time for a set of *p* 2-pattern transition fault vector pairs is as following. The scan shift clock frequency is f_{shift}. The response capture clock frequency is f_{capture}.

$$T = \{(\left\lceil \frac{n}{m} \right\rceil + 1) \bullet p + \left\lceil \frac{n}{m} \right\rceil\} \bullet \frac{1}{f_{shift}} + p \bullet \frac{1}{f_{capture}}$$

$$= \{(\left\lceil \frac{100,000}{512} \right\rceil + 1) \bullet 10^6 + \left\lceil \frac{100,000}{512} \right\rceil\} \bullet \frac{1}{10^8} + 10^6 \bullet \frac{1}{10^9} = 1971.00196 \text{ ms}$$

(c) One possible solution is to use the scan-based built-in self-test (BIST) method. We partition the single long scan chain into multiple scan chains with moderate lengths. The *scan_in* pins of these scan chains can be connected to the outputs of the pseudorandom test pattern generator (PRTG) in the BIST architecture. The primary input pins of the circuit are also connected to the outputs of PRTG. The *scan-out* pins of these scan chains and the primary outputs of the circuit are connected to the on-chip response compaction component (e.g., multiple input signature register). Since the BIST architecture is constructed on-chip, it avoids adding additional scan pins to the circuit design. The testing time for using multiple scan chains is significantly reduced compared to the single scan-chain scheme. The fault coverage of using the BIST method may be slightly lower than that of using the deterministic test-pattern set, since there are random-resistant faults that cannot be covered by pseudorandom test patterns.

Problem 2

15.4 Standard LFSR

Pattern #	X_7	X_6	X_5	X_4	X_3	X_2	X_1	X_0
1.	0	0	0	0	0	0	0	1
2.	1	0	0	0	0	0	0	0
3.	1	1	0	0	0	0	0	0
4.	1	1	1	0	0	0	0	0
5.	1	1	1	1	0	0	0	0
6.	1	1	1	1	1	0	0	0
7.	1	1	1	1	1	1	0	0
8.	0	1	1	1	1	1	1	0

15.8 Weighted random pattern generator

Use an LFSR of a sufficient size such that the inputs for the logic generating each output are coming from separate FFs of the LFSR. In this case, at least 5 bits are required. We can use a primitive polynomial of degree 5 or greater to design the LSFR, for instance, x5+x2+1. Refer to textbook Page 503 for design guidelines. The required outputs are generated using the LFSR outputs X_0 … X_4 and combinational logic. The logical expressions for these outputs are:

$Z_{1/2}=X_0$
$Z_{1/4}=X_1X_0$
$Z_{11/32}=X_4X_3X_2+X_1X_0$
$Z_{1/16}=X_3X_2X_1X_0$

Problem 3

15.14 Aliasing analysis

$$Z = Y(B \oplus C) \oplus B$$

Results of circuit simulation are as follows:

A	B	C	Y	Z	Good machine	Failing machine, e sa0
					$R_1R_2R_3$	$R_1R_2R_3$
0	0	1	D	D	000	000
1	0	0	D	0	011	000
0	1	0	D	$\overline{D}$	011	000
1	0	1	D	D	011	001
1	1	0	1	0	010	100
1	1	1	1	1	111	000
0	1	1	0	1	000	011
0	0	1	D	D	001	000
1	0	0	D	0	$\boxed{111}$	$\boxed{000}$

For output Y, the fault effect is XORed four times, while the fault effect is XORed into Z three times, during the first 7 clock periods. Repeating the first LFSR pattern during the 8th clock period XORs the fault effect in one additional time frame on each output.

The error vector is set to 1 on an output when it differs from a good machine. Here are the other error vectors:

	1	2	3	4	5	6	7	8
Y	1	1	1	1	0	0	0	1
Z	1	0	1	1	0	0	0	1

Even with the repeated pattern, the cumulative # of 1's in the error vector remains odd. This is why aliasing does not occur. If the total # of 1's in the error vector becomes even, then aliasing might occur.

15.15 Fault detection

ABC	Good	A s-a-0		A s-a-1		$B-e$ s-a-0	
	$R_1R_2R_3$	YZ	Bad $R_1R_2R_3$	YZ	Bad $R_1R_2R_3$	YZ	Bad $R_1R_2R_3$
001	000	11	000	11	000	11	000
100	011	10	011	10	011	10	011
010	011	10	011	10	011	10	011
101	011	11	011	11	011	11	011
110	010	10	010	10	010	10	010
111	111	01	111	11	111	11	111
011	000	01	010	11	000	11	000
001	001	11	100	11	011	11	011
	$\boxed{111}$		$\boxed{001}$ Yes		$\boxed{010}$ Yes		$\boxed{010}$ Yes

ABC	Good	$B-e$ s-a-1		$C-e$ s-a-0		$C-e$ s-a-1	
	$R_1R_2R_3$	YZ	Bad $R_1R_2R_3$	YZ	Bad $R_1R_2R_3$	YZ	Bad $R_1R_2R_3$
001	000	00	000	11	000	11	000
100	011	10	000	10	011	10	011
010	011	10	010	10	011	01	010
101	011	00	111	11	011	11	100
110	010	10	011	10	010	10	001
111	111	11	011	11	111	11	110
011	000	01	010	11	000	01	100
001	001	00	100	11	011	00	011
	$\boxed{111}$		$\boxed{010}$ Yes		$\boxed{010}$ Yes		$\boxed{001}$ Yes

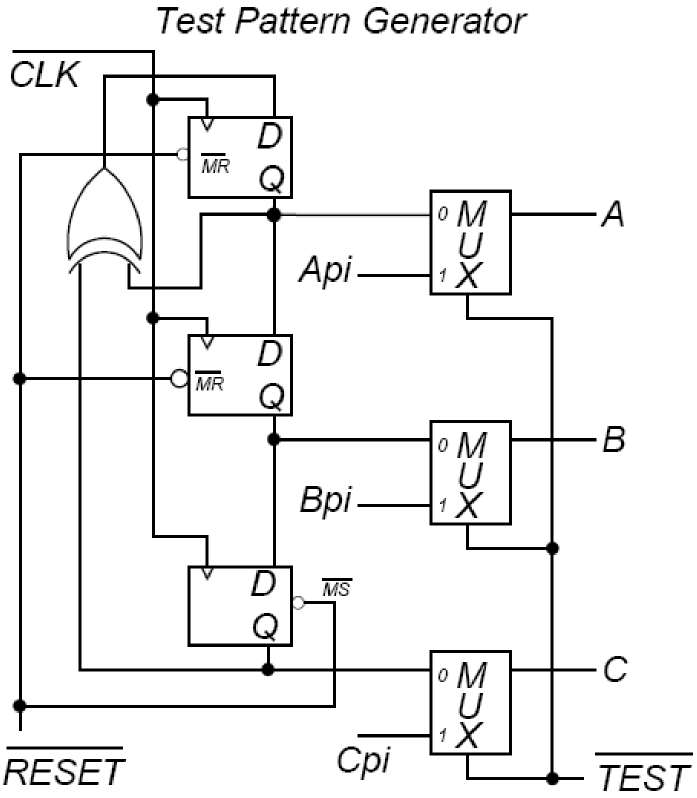
15.16 Fault detection

ABC	Good	B s-a-0		B s-a-1		$B - g$ s-a-0	
	$R_1R_2R_3$	YZ	Bad $R_1R_2R_3$	YZ	Bad $R_1R_2R_3$	YZ	Bad $R_1R_2R_3$
001	000	11	000	01	000	11	000
100	011	10	011	10	001	10	011
010	011	10	011	10	110	11	011
101	011	11	011	11	101	11	010
110	010	10	010	10	101	11	110
111	111	11	111	11	100	10	100
011	000	11	000	01	001	01	000
001	001	11	011	01	101	11	001
	$\boxed{111}$		$\boxed{010}$ Yes		$\boxed{111}$ No		$\boxed{111}$ No

ABC	Good	$B - g$ s-a-1		f s-a-0		f s-a-1	
	$R_1R_2R_3$	YZ	Bad $R_1R_2R_3$	YZ	Bad $R_1R_2R_3$	YZ	Bad $R_1R_2R_3$
001	000	10	000	00	000	11	000
100	011	11	010	00	000	10	011
010	011	10	110	01	000	10	011
101	011	10	101	00	001	11	011
110	010	10	100	01	100	10	010
111	111	11	000	01	011	11	111
011	000	01	011	01	000	11	000
001	001	10	000	00	001	11	011
	$\boxed{111}$		$\boxed{010}$ Yes		$\boxed{100}$ Yes		$\boxed{010}$ Yes

15.19 Signature computation

(a) The hardware is shown in following figure.

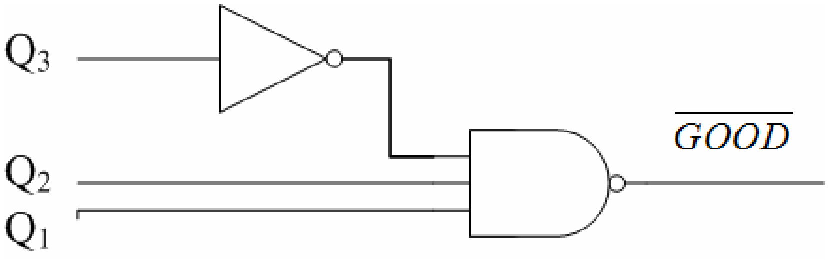


Circuit for Problem 15.19 with BIST pattern generator and input MUX.

(b)

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} (t+1) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} (t)$$

(c) The initial state of the flip-flops is assumed to be $Q_1Q_2Q_3$ = 000. Good signature is $Q_1Q_2Q_3$ = 110. The NAND gate-signature comparator should be designed as following: only $Q_1Q_2Q_3$ = 110 can make $\overline{GOOD}$ = 0, other values have to make $\overline{GOOD}$ = 1.



(d) For q: s-a-0, the bad signature is $Q_1Q_2Q_3$= 010. Since this bad signature is different from the good signature of the circuit, there is no alias for q: s-a-0.

Problem 4

The idea of LOC and LOS is to create a transition on the paths being tested and capture the response at-speed. The pattern for the transition is generated differently in LOC and LOS. In LOC, the pattern is generated by the logic. In LOS, the pattern is shifted in through the scan chains. Not all possible bit combinations of this pattern can be generated as both generation mechanisms put constraints on the solution. In LOC, some patterns cannot be generated because the logic cannot create a certain output vector for any input vector. Some scan elements need to be fixed in both launch and capture cycles which is not always possible with LOS (shift dependency). Because of the different origins of constraints in LOC and LOS, some patterns can only be generated only by using one method and not the other. Therefore, some faults can only be detected by LOC and not LOS and vice versa.

Solution for better fault coverage: generate test patterns for LOC. Run fault simulation using the full fault list. Drop all the faults that can be detected by the generated LOC test pattern set. Run ATPG for the remaining fault list using the LOS scheme. Merge the two pattern sets.
Alternatively, use LOS for nodes that have high controllability values (poor controllability) and LOC for the rest. This way LOS can be limited to a small subset of faults.