CSE435 Introduction to EDA & Testing - Spring 2022

Homework Assignment #5

Shao-Hsuan Chu - B073040018

1. (20%) A circuit has the truth table of Table 1. When there is a fault (faults) on the circuit, the faulty truth table becomes Table 2. Try to derive tests to detect the fault (faults).

CD \ AB	00	01	11	10
00	1			
01		1	1	
11				1
10			1	1

Table 1

CD \ AB	00	01	11	10
00				
01		1	1	
11				
10	1		1	1

Table 2

> **Solution:** Compare two truth tables, we can tell the circuit has stuck-at-0 fault at output when input (A, B, C, D) equals (0, 0, 0, 0) or (1, 0, 1, 1). The circuit also has stuck-at-1 fault at output when the input equals (0, 0, 1, 0).
>
> **Answers:** {(0, 0, 0, 0), (1, 0, 1, 1), (0, 0, 1, 0)}

2. (80%) Generate a test for the fault f-sa1 in Figure 1 by the following FOUR methods. Be sure to give the **key steps to show the features of every algorithm**, and also **draw the decision trees** for each case.

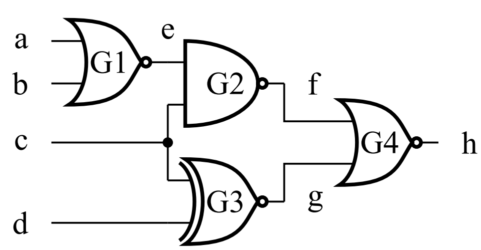

Figure 1

(a) (20%) Use the **Boolean difference method** to derive all the test patterns to detect the fault f-sa1.

> **Solution:** To test the stuck-at-1 fault at f, f must equal 0 to activate the fault. In addition, the fault has to be observable at the output, meaning that the boolean difference of the logic function F w.r.t. f should be 1, i.e., $F_f(0) \oplus F_f(1) = 1$.

$$
\begin{aligned}
f &= 0 \\
f' &= 1 \\
F_f(0) \oplus F_f(1) &= 1 \\
(f')(F_f(0) \oplus F_f(1)) &= 1 \\
(f')(0 + (c \oplus d)')' \oplus (1 + (c \oplus d)')' &= 1 \\
(f')(c \oplus d \oplus 0) &= 1 \\
(f')(c \oplus d) &= 1 \\
((a+b)'c)''(c \oplus d) &= 1 \\
((a+b)'c)(c \oplus d) &= 1 \\
a'b'c(c \oplus d) &= 1 \\
a'b'c(cd' + c'd) &= 1 \\
a'b'ccd' + a'b'cc'd &= 1 \\
a'b'cd' &= 1
\end{aligned}
$$

Answer: {(a, b, c, d) | a'b'cd' = 1} = {(0, 0, 1, 0)}.

(b) (20%) Generate a test for the fault f-sa1 by using **D-algorithm**.

Solution: We'll use some acronyms, explained in Table 3, in the following D-algorithm procedure, shown in Table 4.

Acronym	Meaning	Comments
TC	Test Cube	TC(n) is the test cube at APTG step n.
PDCF	Primitive D-Cube for a Fault	Minimal input condition to activate the fault.
PDC	Propagation D-Cube	Minimal input condition to propagate the fault through a gate.
SC	Single Cover	Valid input/output combination for a gate.

Table 3: Acronym for D-algorithm procedure

Step	Comments	Cube a	b	c	d	e	f	g	h
1	Activate f-sa1. TC(0) = PDCF			1		1	D'		
2	Backward implication: SC_{G1}	**0**	**0**			1			
	TC(1) = TC(0) $\cap$ SC_{G1}	0	0	1		1	D'		
3	Propagate fault. D-drive = G4. PDC_{G4}						D'	0	D
	TC(2) = TC(1) $\cap$ PDC_{G4}	0	0	1		1	D'	0	D
4	Backward implication: SC_{G3}			1	**0**			0	
	TC(3) = TC(2) $\cap$ SC_{G3}	0	0	1	0	1	D'	0	D
5	No justification needed. Test generated, (a, b, c, d) = (0, 0, 1, 0)								

Table 4: The procedure for D-algorithm

Answer: (a, b, c, d) = (0, 0, 1, 0).

(c) (20%) Generate a test for the fault f-sa1 by using **9-V Algorithm**.

Solution: The acronyms in Problem 2(b) are also used in the 9-V algorithm procedure, shown in Table 5.

Step	Comments	Cube a	b	c	d	e	f	g	h
1	Activate f-sa1. TC(0) = PDCF			1		1	D'		
2	Backward implication: SC_{G1}	**0**	**0**			1			
	Forward implication: SC_{G4}						D'		**0/D**
	TC(1) = TC(0) $\cap$ SC_{G1} $\cap$ SC_{G4}	0	0	1		1	D'		0/D
3	Propagate fault. D-drive = G4. PDC_{G4}						D'	0	D
	TC(2) = TC(1) $\cap$ PDC_{G4}	0	0	1		1	D'	0	D
4	Backward implication: SC_{G3}			1	**0**			0	
	TC(3) = TC(2) $\cap$ SC_{G3}	0	0	1	0	1	D'	0	D
5	No justification needed. Test generated, (a, b, c, d) = (0, 0, 1, 0)								

Table 5: The procedure for 9-V algorithm

Answer: (a, b, c, d) = (0, 0, 1, 0).

(d) (20%) Generate a test for the fault f-sa1 by using **PODEM algorithm**.

Solution:

1. Activate f-sa1. Initial objective: f = 0.
2. Since G2 is an imply gate, backtrace to the hardest PI, a. With an even inversion parity, a = 0. Simulation, objective not achieved.

3. Backtrace to the next hardest PI, b. With an even inversion parity, b = 0, e = 1. Simulation, objective not achieved.
4. Backtrace to the next hardest PI, c. With an odd inversion parity, c = 1. Simulation, objective achieved.
5. Propagate the fault. Objective: g = 0.
6. Backtrace to the remaining PI, d. With an odd inversion parity, d = 1. Simulation, $g = (c \oplus d)' = 0$. Conflict with the objective. Flip d to 0. Simulation, objective achieved.
7. Test generated, (a, b, c, d) = (0, 0, 1, 0).

Answer: (a, b, c, d) = (0, 0, 1, 0).