EECS 579 Homework No.3 Solutions

Problem 1 (20 points) *Boolean difference*

(a) We treat e as a "primary" input, and calculate the output function $z(A,B,C,D,E,e)$ as follows:

$$z = (A + E)k = (A + E)(B + g)(D + g)' = (A + E)(B + g)D'g' = (A + E)BD'g'$$
$$= (A + E)BD'de = (A + E)BCD'e$$

Now the tests for e stuck-at-1 are denoted by the Boolean function T = $(e').dz/de$, where dz/de is the Boolean difference of z with respect to e.

$$dz/de = = (A + E)BCD'.1 \oplus (A + E)BCD'.0 = (A + E)BCD'$$

Now $e' = (C \oplus D)' = CD + C'D'$, so T = $(e').dz/de = (CD + C'D')\,(A + E)BCD' = 0$

which means that the test set for the given fault is empty, i.e., the fault is undetectable. The SOP expression in this rather uninteresting case is trivially "T = 0".

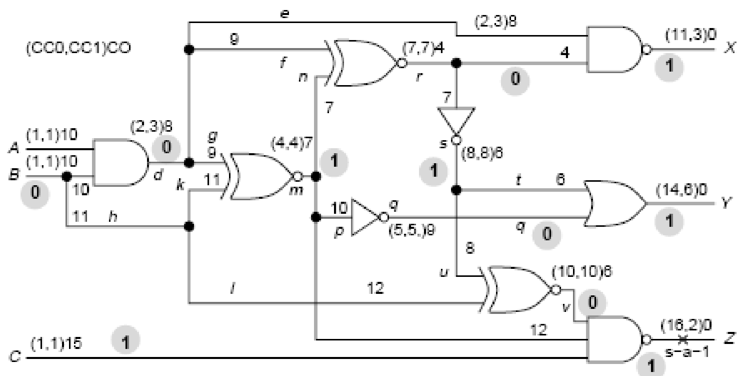
(b) The new "primary" output line k changes the test-set function for e stuck-at-1 from T = $(e').dz/de$ to

$$T^* = (e').dz/de + (e').dk/de$$

where $k(A,B,C,D,E,e)$ denotes the function appearing on line k. However, as computed above $k = BCD'e$, so $T^* = (e').dk/de = 0$, and again the fault is undetectable.

Problem 2 (10 points) *PODEM.* Text, Page 209, Problem 7.20.

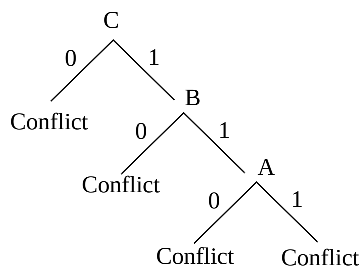
The given circuit (Figure 7.24, page 190) is shown below, along with its SCOAP testability measures (not requested in this problem). The target fault is Z s-a-1.



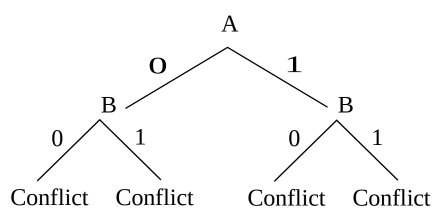
	Decision	Implications	Explanatory Comments
1	C=1	None	Start with ABC = XXX. Set initial objective = (Z,0) to sensitize fault. Backtrace to C using heuristic #1 "Go to closest primary input," giving the result (C,1).
2	B=0	d=g=e=0, x=1, m=1, r=0, t=u=1, q=0, y=1, v=0, z=1 Objective cannot be met.	Objective is still (Z,0). Backtrace to B using heuristic #2 for XNOR "Select 0 before 1" to get (B,0).
3	B=1	No implications from B=1	Backtrack. (Note that previous implications are reset to X.)

	Decision	Implications	Explanatory Comments
4	A=0	d=g=e=0, x=1, z=1, ... Objective cannot be met.	Objective is still (Z,0). Backtrace to A to get (A,0).
5	A=1	d=g=e=1, m=0, z=1, ... Objective cannot be met.	Backtrack.
6	C=0	z=1. Objective cannot be met	Backtrack. All possibilities have been tried so fault z/1 is undetectable.

Solution 1: Solutions will vary depending on the tie-breaking and other heuristics used in backtracing. The solution given above corresponds to the following decision tree:



Solution 2: Another representative solution corresponds to the following decision sequence: A=0, B=0, B=1, A=1, B=0, B=1.



Problem 3 (20 points) *FAN* Text, Page 209, Problem 7.22.

Step	Objective	Decisions (top) and Implications (below)	Explanatory Comments
1	(d,0) Activate the fault site.	B=0 d=D’	*Fan Heuristic:* Activate (sensitize) the fault directly.
2	(g,D) Propagate error from d to g.	e=1, g=D, f=D	*Fan Heuristic*: Unique sensitization of path d-g-f.
3	(k,D) Propagate error to headline k.	D=0 C=1, h=D’, k=0	Attempt to propagate error to k is found to be impossible.
4	(k,D) Propagate error to headline k.	D=1 C=0, h=0, k=0	Backtrack. Again attempt to propagate error to headline fails.
5	Propagate error from d to g.	B=1 C=0, f=1,d=D’,e=1,g=D, D=1, h=0,k=0	Backtrack. *Fan Heuristics*: Unique sensitization; Immediate implication. Again cannot propagate the error. All cases have been tried so the fault is undetectable.

Other reasonable choices of heuristics were accepted.

Problem 4 (10 points) *Serial adder*. Text, Page 249, Problem 8.4.

The required test sequence exists and can be found using any method such as (simplified) PODEM, DALG, or the state-table method using two time steps:

1. *Fault activation*. Assuming the present state C_n is unknown, we want to set the next state to 1. With C_n = X, backward justication of $C_{n+1} = 1$ in Figure 8.3 (p.215 of the text) gives $A_n = 1$ and $B_n = 1$.

2. *Path sensitization*. For the next vector, the last next state becomes the present state and the fault C_n s-a-0 is sensitized. We can then sensitize a path from C_n to S_n by again setting $A_n = 1$ and $B_n = 1$.

This leads to the test sequence $(A_n,B_n),(A_{n+1},B_{n+1})$ = (1,1), (1,1) which is of length two. In fact, the second input (A_{n+1},B_{n+1}) can be any vector (X,X).

Problem 5 (20 points) *Sequential circuit testing*

Since minimality is required, we use a theoretically exact approach, the state-table method.

(a) The fault-free and faulty machines (with fault $D_2/1$) are as follows:

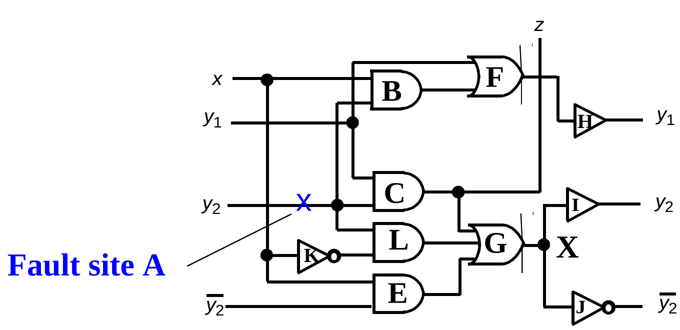
	Good machine M_0		**Faulty machine M_1**	
Present State	**x = 0**	**x = 1**	**x = 0**	**x = 1**
A = 00	A,0	B,0	**<u>B</u>**,0	B,0
B = 01	B,0	C,0	B,0	**<u>D</u>**,0
C = 10	C,0	D,0	**<u>D</u>**,0	D,0
D = 11	D,1	D,1	D,1	D,1

(b) y_1y_2 = 00. The tree-search method discussed in class easily gives the result. We find no sequences of length one or two that distinguish M_0 from M_1. However, there are four such sequences of length three: 010, 010, 110, 111, any one of which is an optimal test for the given fault.

(c) y_1y_2 = XX. Inspection of the state tables shows that if the initial state is D for either machine, then the next state is always D with output 1. (D is a classic “trap” state.) It follows that the given fault becomes undetectable when the initial state is D, so it must also be undetectable when the initial state is unknown.

Problem 6 (20 points) *Sequential PODEM*

(a) Below is the pseudo-combinational (time-frame) model for the given circuit. Since the values of the initial state y_1y_2 = 00 have been decided a priori, start with the implications of those values. Note that y_2



= 0 immediately produces an error value D’ at the fault site *A*.

Objectives	Decisions	Implications	Explanatory Comments
		A = D′, $C = z = 0$, $L = 0$	Time frame 1 y_1y_2 is set to 00. Start with its implications. At this point x = X.
$(x,1)$	$x = 1$	$B = F = y_1$ = D′, $E = G = 1$, $y_2 = 1$, $y'_2 = 0$	Since we now have a D frontier, a suitable first objective is $(x,1)$ to propagate D′ through B. *Backtrace*$(x,1)$ trivially yields $(x,1)$ so PODEM sets $x = 1$.
			Time frame 2 y_1y_2 = D′1 and x = X . When *FF2* outputs $y_2 = 1$, it encounters A/1, so the value of A is 1 not D'. Again do *Imply* first.
		$A = 1$, $E = 0$, $C = z$ = D′	PODEM immediately gets detection at z, so the test is ***x*(1)*x*(2) = 1X**

(a) Intuitively, y_1y_2 = XX allows y_2 to have initial value 1, making the given s-a-1 fault on y_2 undetectable for that case. Repeating the PODEM algorithm above with y_1y_2 = XX (even with 9-valued logic) leads to the same conclusion that the fault is undetectable.