

**Department of Electrical and Computer Engineering  
University of Wisconsin–Madison**

**ECE 553: Testing and Testable Design of Digital Systems  
Fall 2011**

**ASSIGNMENT #5  
Solution**

1. For this problem, you need to use a sequential circuit available in

`~ece553/TESTCAD/nets/hw5-circuits/hw5_circuit_1`

- (a) Derive a combinational circuit by replacing each flip-flop (LATCH) to a buffer (BUF). This is known as the pseudo-combinational transformation, which can be applied to any cycle-free clocked sequential circuit.

Answer: The circuit is obtained by replacing flip-flop (LATCH) to a buffer (BUF) as follows:

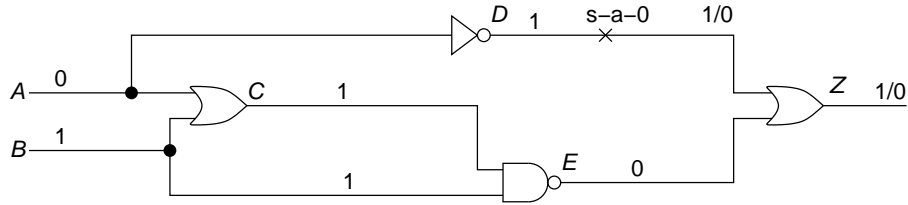
```
1 PI 3 7 ;
2 PI 4 ;
3 OR 5 ;
4 BUF 3 6 9 ;
5 BUF 6 ;
6 NAND 8 10 11 ;
7 NOT 10 ;
8 BUF 10 ;
9 BUF 11 ;
10 OR 12 ;
11 OR 13 ;
12 PO ;
13 PO ;
```

- (b) Derive a test for the circuit obtained after (a) for the fault “7 0 0” and “9 0 1”. (You can do this either by hand or by PODEM in the testcad toolset).

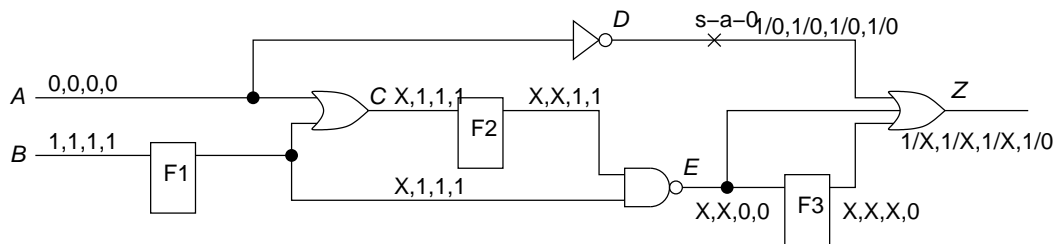
For the fault 7 0 0 : Input = 1/0 2/1.

For the fault 9 0 1 : Test does not exist.

- (c) Verify that the fault “7 0 0” can be detected in the original sequential circuit by repeating the generated vector for three clock cycles.



Pseudo-combinational circuit for the sequential circuit of Figure 8.9..



Test simulation in sequential circuit.

- (d) At this point you found that “9 0 1” is redundant in the pseudo-combinational circuit. Now, try to use the sequential test generator (fastest in testcad toolset) on the original circuit to generate a test vectors for “9 0 1”. Verify if the generated test sequence indeed detects the fault by hand simulation. You’ll find “9 0 1” is not redundant in the original circuit.

Answer:

The test sequence found for “9 0 1” is 00, 11, 00, and Using similar simulation method given in (c), the fault must be detected.

- (e) Explain what causes such a difference between using combinational ATPG on the pseudo-combinational circuit and using sequential test generator on the original sequential circuit.

(Submission credit)

2. (Sequential test generation using TESTCAD) For this problem, you are required to use ‘FASTEST’, the sequential test generator in TESTCAD toolset.

The sequential circuit under test can be found at the location given below:

/pong/usr0/e/ece553/TESTCAD/nets/hw5-circuits/hw5\_circuit\_2

Use the testcad tools to answer the following questions.

- a) Write the fault-list using equivalent fault collapsing method. Note that you cannot use 'listfaults' for sequential circuit. You are required to use 'faultgen' instead.

4	0	0 ,4	0	1
3	0	0 ,3	0	1
2	0	0 ,2	0	1
1	0	0 ,1	0	1
16	0	0 ,16	0	1
9	0	0 ,9	0	1
10	0	0 ,10	0	1
10	9	0 ,10	9	1
17	0	0 ,17	0	1
17	16	0 ,17	16	1
11	0	0 ,11	0	1
11	10	0 ,11	10	1
11	16	0 ,11	16	1
12	0	0 ,12	0	1
12	10	0 ,12	10	1
13	0	0 ,13	0	1
15	0	0 ,15	0	1
14	0	0 ,14	0	1
14	9	0 ,14	9	1
14	15	0 ,14	15	1
5	0	0 ,5	0	1
5	14	0 ,5	14	1
5	15	0 ,5	15	1
8	0	0 ,8	0	1
7	0	0 ,7	0	1
7	15	0 ,7	15	1
6	0	0 ,6	0	1
6	14	0 ,6	14	1

- b) Use 'fastest' to generate tests for your fault list obtained in part(a). Report fault coverage, generated test vectors, and statistics report.

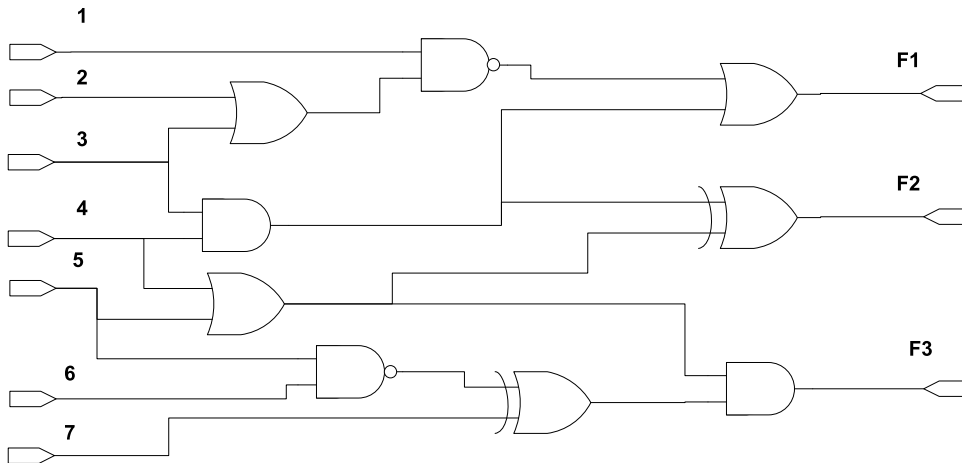
```
Total 45357 implications; 0 hrs 0 sec.
Faults: 56; Detected: 4; Coverage: 0.071429
Total Vectors: 5
Memory used: [138522624/138522624/138522624] 135276K
```

- 1: 0000 XXX;
- 2: 1111 XXX;
- 3: 0011 XXX;
- 4: 0011 XXX;
- 5: 1001 XXX;

c) Repeat part a) and part b) for the circuit in Problem 1. Explain the difference in fault coverage briefly.

Answer : For the circuit in Problem 1, you will find that "fastest" will give 100% fault coverage. That's because the circuit in Problem 1 is a cycle-free circuit but the circuit in this problem is a cyclic circuit.

3. For the following circuit, derive a minimum pseudo-exhaustive test set using only cones of outputs partitions. Mark these partitions and list the test set in a table. Also, for each test vector indicate the outputs of the segment (partition) you are testing. Use as much overlapping as possible to obtain the minimum number of tests. From your results, state some advantages and disadvantages of this method.



**Fig.1**

Figure 1: Figure for Problem 3

1	2	3	4	5	6	7	F1	F2	F3
0	0	0	0	0	0	0	1	0	0
0	0	0	1	0	0	0	1	1	1
0	0	1	0	0	0	1	1	0	0
0	0	1	1	0	0	1	1	0	0
0	1	0	0	1	0	0	1	1	1
0	1	0	1	1	0	0	1	1	1
0	1	1	0	1	0	1	1	1	0
0	1	1	1	1	0	1	1	0	0
1	0	0	0	0	1	0	1		0
1	0	0	1	0	1	0	1		1
1	0	1	0	0	1	1	0		0
1	0	1	1	0	1	1	1		0
1	1	0	0	1	1	0	0		0
1	1	0	1	1	1	0	0		1
1	1	1	0	1	1	1	0		1
1	1	1	1	1	1	1	1		1

4. For the following circuit in Figure 2, derive a minimum pseudo-exhaustive test set with sensitized partition using the partitions shown. List the test set in a table and for each test vector indicate the outputs of the segment (partition) you are testing. (To obtain the minimum number of tests, use as much overlapping as possible)

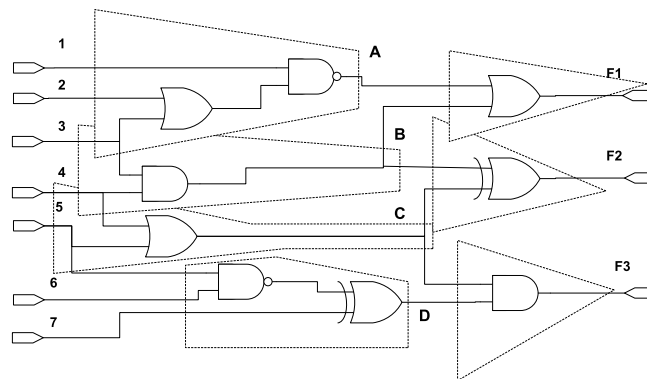


Fig.2

Figure 2: Figure for Problem 4

Solution

In order to test A, B must be 0 so that the output of A can be propagated to an output. For this to happen, 3 or 4 must be 0 to test A. The same applies to D and C, except 4 or 5 must be 1 to test D.

1	2	3	4	5	6	7	A	B	C	D	F1	F2	F3
0	0	0	0	1	0	0	1	0	1	1	1	1	1
0	0	1	0	1	0	1	1	0	1	0	1	1	0
0	1	0	1	0	0	0	1	0	1	1	1	1	1
0	1	1	0	1	1	0	1	0	1	0	1	1	0
1	0	0	1	1	1	1	1	0	1	1	1	1	1
1	0	1	0	0	0	0	0	0	0	1	0	0	0
1	1	0	1	0	0	1	0	0	1	0	0	1	0
1	1	1	0	0	0	1	0	0	0	0	0	0	0
1	1	1	1	0	1	0	0	1	1	1	1	0	1
0	0	1	1	0	1	1	1	1	1	0	1	0	0

Note: F2 cant be tested exhaustively.

5. (a) A *state coupling fault* (SCF)  $\langle i, j \rangle$  is a memory fault where the coupling cell  $i$  entering the state 0 or 1 causes the coupled cell  $j$  to enter the state 0 or 1. These are denoted as  $\langle 0; 0 \rangle$ ,  $\langle 0; 1 \rangle$ ,  $\langle 1; 0 \rangle$  and  $\langle 1; 1 \rangle$ .
- (b) An *inversion coupling fault* (CFin)  $\langle i, j \rangle$  is where the coupling cell  $i$  having a transition causes the coupled cell  $j$  to invert its state. These are denoted as  $\langle \uparrow; \downarrow \rangle$ ,  $\langle \downarrow; \uparrow \rangle$ .
- (c) An *idempotent coupling fault* (CFid)  $\langle i, j \rangle$  is where the coupling cell  $i$  having a transition causes the coupled cell  $j$  to enter a particular state. These are denoted as  $\langle \uparrow; 0 \rangle$ ,  $\langle \uparrow; 1 \rangle$ ,  $\langle \downarrow; 0 \rangle$ , and  $\langle \downarrow; 1 \rangle$ .
- (d) A *dynamic coupling fault* (CFdyn) is where a read or write of a cell in one word forces the contents of a cell in another word to 0 or 1. These are denoted as  $\langle r0|w0; 0 \rangle$ ,  $\langle r0|w0; 1 \rangle$ ,  $\langle r1|w1; 0 \rangle$ , and  $\langle r1|w1; 1 \rangle$ .
- (e) A rising (falling) *transition fault* (TF) in a memory cell that can come up in either the 0 or 1 state, but any attempt to change its state from 0 to 1 (1 to 0) fails. These are denoted as  $\langle \uparrow; 0 \rangle$  and  $\langle \downarrow; 1 \rangle$ .
- (f) An *active neighborhood pattern sensitive fault* (ANSPF) causes the base cell to change due to a pattern and transition in the deleted neighborhood. The base cell can go to 0, 1, or invert.

- (g) A *passive neighborhood pattern sensitive fault* (PNPSF) prevents the base cell from changing when a particular pattern exists in the deleted neighborhood.
- (h) A *static neighborhood pattern sensitive fault* (SNPSF) forces the base cell into a particular state when a particular pattern exists in the deleted neighborhood.
- (i) A *data retention fault* causes a memory cell to forget its content over time, usually due to a damaged SRAM pullup device or a damaged DRAM capacitor.
- (j) An *address decoder fault* in a memory causes, 1) an address  $i$  to instead access location  $j$ , 2) an address  $i$  to access no location, or 3) address  $i$  to simultaneously access multiple locations.