

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

**ECE 553: Testing and Testable Design of Digital Systems
Fall 2009**

ASSIGNMENT #4

Solution

Date: Tuesday, October 20, 2009

Due date : Tuesday, November 3, 2009

1. (Bushnell and Agrawal) Problem 8.2

It requires just one vector to initialize the circuit. If the initial state is unknown, i.e., $C_n = X$, the vector $A_n = B_n = 1$ initializes the state to 1, irrespective of the presence of any fault at the output S_n . Given this state, detection of any output fault at the output reduces to a combinational ATPG problem of setting the output to the opposite value. This can be done by a single vector: $(A_n = 0, B_n = 0)$ will set the output to 1 or $(A_n = 0, B_n = 1)$ will set it to 0. Thus, just two vectors, an initialization vector 11 followed by an appropriate vector to set the output, will detect the output fault in the circuit of Figure 8.3 (see page 215 of the book.)

2. (Bushnell and Agrawal) Problem 8.5

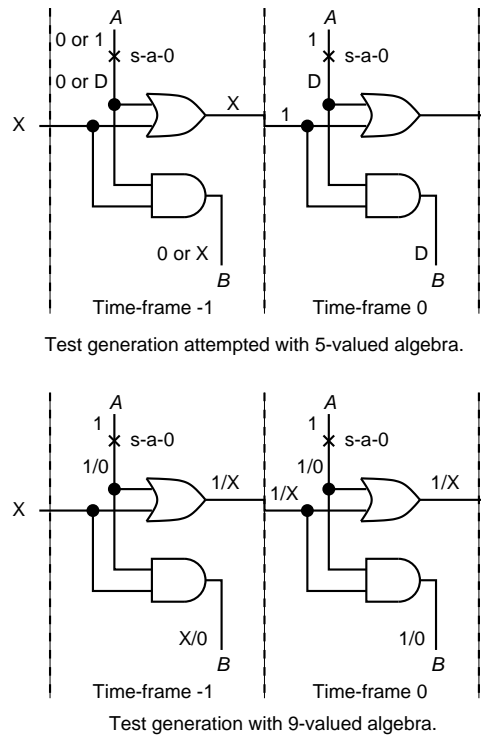
For test generation with the five-valued algebra, we use the following steps (also see the illustration):

Step 1: Place a D at the output B in time-frame 0.

Step 2: This can only be justified by either DD or D1 input to the AND gate in time-frame 0. DD is not possible due to the state input being X in the time-frame -1. We place D1 by applying $A = 1$ and assuming that a state 1 can be justified.

Step 3: Any input, 0 or 1, as shown in the figure, produces a state output X from time-frame -1. Thus, the faulty circuit cannot be initialized to any known state, including the 1 needed for the test. **Hence, it is impossible to find a test by the 5-valued algebra.**

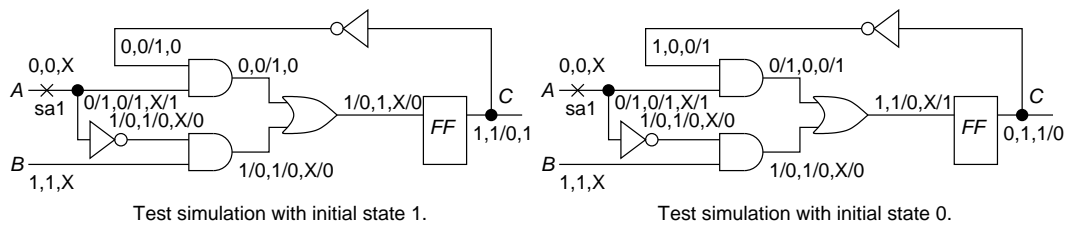
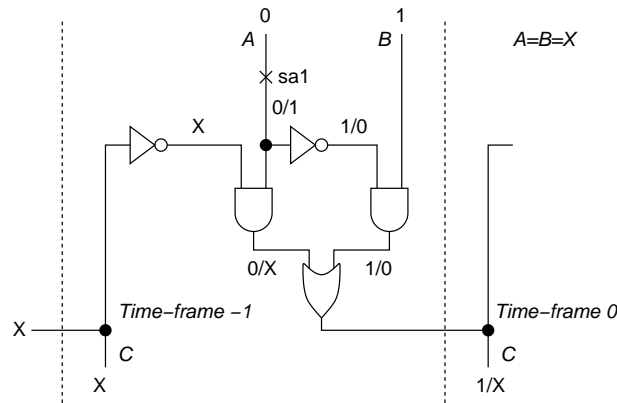
Following similar steps with the nine-valued algebra (see illustration below), we find that two 1's at A detect the fault at B as 1/0 in time-frame 0. Notice that the fault is detected although the faulty circuit is never initialized.



3. (Bushnell and Agrawal) Problem 8.6

The following figure illustrates the time-frame expansion procedure of generating a vector, $A = 0$, $B = 1$, which starting from the unknown state detects the fault A s-a-1 as $1/X$. After the application of the input vector, the flip-flop is clocked before the output can be observed. Even if we add more vectors to the test sequence, the faulty circuit output will not become deterministic. This is because the faulty circuit is not initializable. The fault is only potentially detectable.

Note: Some test generators will find the potential detection test of the above type. Others will consider the fault untestable (conservative approach.) Most fault simulators will find the fault potentially detectable. Interestingly, the two test simulation scenarios in the figure show that the fault is definitely detectable, though the detection requires multiple observations. If we assume the initial state to be 1 then the fault is detected as $1/0$ after the application of the first clock. However, this output will be 1 (same as the correct output) if the initial state was 0. In this case, repeating the same vector and clocking once again will produce a $1/0$ output. A conventional fault simulator will not report such detection because it does not enumerate the possible initial state scenarios. For such multiple observation tests see reference [525] of the book.



4. For the finite state machine in Table 1,

a.) This machine is strongly connected.

b.) For the sequence 1000, the state machine has the following transitions, (ABCDE) so 1000 is a synchronizing sequence but is NOT the shortest synchronizing sequence 000. The state transitions for that are (ABCDE) \rightarrow (ABC) \rightarrow (AB) \rightarrow (A)

c.) HS = 101

State Output Final_State

A	001	D
B	000	C
C	111	D
D	000	C
E	011	D

d.) There are six 4-bit distinguishing sequences: 0000,0001,0011,0101,0110,0111.

5. The finite state machine in Fig.1 has a single input, a single output, and 5 states.

(a) Convert the state machine into a state transition table like the one given in Problem 4.

- (b) Find a shortest *synchronizing sequence* for this machine.
- (c) Find a minimum length *distinguishing sequence*. Tabulate the output responses, initial and final states of applying your distinguishing sequence to the machine in each of the 5 starting states.
- (d) Design a *checking sequence* for this machine such that the total length of the sequence is small. Note that you can achieve this by choosing appropriate transfers while designing the checking sequence. You may use SS to denote the synchronizing sequence you found in part (a) and DS to denote distinguishing sequence you found in part (b). Likewise, you may use T_{ij} to denote transfer sequence from state i to state j . However, you have to clearly indicate what the sequences are in terms of inputs, and the states after the application of the sequences. In addition, also indicate the expected outputs whenever the outputs are to be observed.

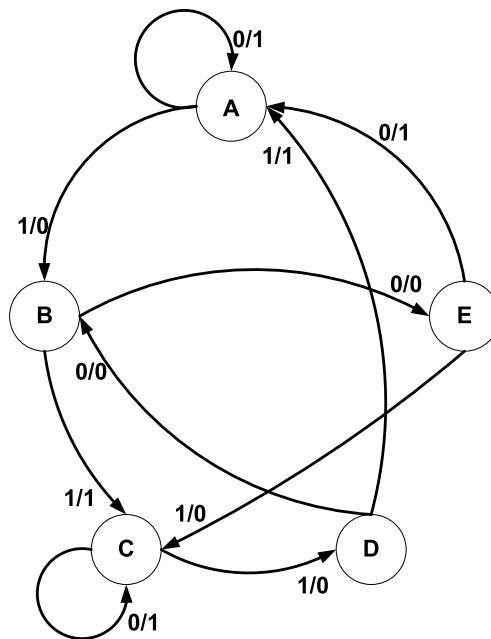


Fig 1. FSM for Prob. 5

a)The state machine is given in the tabular form as follows:

Table 1: State Machine for Problem 5.

	Input	
	0	1
A	A/1	B/0
B	E/0	C/1
C	C/1	D/0
D	B/0	A/1
E	A/1	E/0

b)The shortest SS for the given State Machine is : **0001000**.

c)There are two shortest length Distinguishing Sequences : **10100** and **10101**. The initial and final states can be tabulated as follows:

DS0 = 10100

Initial	Output	Final
A	00011	A
B	11000	E
C	00111	C
D	11001	A
E	01001	A

DS1 = 10101

Initial	Output	Final
A	00010	B
B	11001	C
C	00110	D
D	11000	E
E	01000	E

d)A sample test plan can be given as follows.

Phase I : SS DS1 DS1 DS1 DS1 DS0

A A→B B→C C→D D→E E→A

Phase II : Many different solutions possible.