

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

ECE 553: Testing and Testable Design of Digital Systems
Fall 2008-2009

Midterm Examination

CLOSED BOOK

Kewal K. Saluja

Date: October 30, 2008
Place: Room 1152 Mechanical Engineering (In Class)
Time: 11:00 - 12:15 PM
Duration: 75 minutes

PROBLEM	TOPIC	POINTS	SCORE
1	General Questions	16	
2	Test Economics	15	
3	Modeling	10	
4	Fault Simulation - Parallel	15	
5	SCOAP Computation	15	
6	Test Generation - PODEM	15	
7	Checking Seq	14	
TOTAL		100	

Show your work carefully for both full and partial credit.
You will be given credit only for what appears on your exam.

Last Name (Please print): **SOLUTION**

First Name: _____

ID Number: _____

1. (16 points) General Questions

Answer the following in brief and to the point. You must not use more than two to three lines of explanation where an an explanation is needed.

- (a) **(1.5 points)** If a fault a is equivalent to fault b, does that mean that fault b is also equivalent to fault a? yes or no
Yes
- (b) **(1.5 points)** If a fault a dominates fault b, does that mean that fault b also dominates fault a? Yes or no.
No
- (c) **(1.5 points)** If a fault x dominates fault y, which of the two faults can be deleted to reduce the fault list for fault detection.
Fault x
- (d) **(2 points)** If fault a dominates fault c and fault b also dominates fault c, what relation can you deduce between faults a and b?
Nothing
- (e) **(1.5 points)** In a gate level fanout-free realization of a circuit we only need to consider the faults at primary inputs and all other faults need not be considered from fault detection point of view. True or false?
True
- (f) **(2 points)** Should redundancy always be removed from a circuit? Explain.
No. Sometimes it is intentional for performance or design reasons. Such as use of buffers, use of extra logic to avoid hazards.
- (g) **(2 points)** Name two techniques that can be used to reduce the number of test patterns for a circuit while still providing the same or better fault coverage.
Reverse order simulation, merging of test vectors, filling x's
- (h) **(2 points)** What is meaning by “flattening” of a hierarchical design of a circuit? Each instantiation of the circuit in the hierarchy is replaced by its netlist and the resulting circuit contains all the gates and interactions information. This is “flattening at gate level” for this course.
- (i) **(2 points)** Give two reasons as to why a “good circuit” may fail during testing when being tested by a test set generated to detect stuck-at faults.
The models used for test generation man not be good - for example lack of modeling three-state may give tests that may be quite a test needed to test the circuit.
Test environment may be reflective of the environment for test generation.
The test mode may cause failures that only happen in the the test mode and not the normal mode of operation.

2. (15 points) Test Economics

A manufacturer of boards uses two types of IC in its boards. The information about these devices is given in the table below.

Device Type	Number on Board	Fault Cov (T)	Clustering β	Area X density (Af)	Yield (Y)
Type-1	10	89%	0.11	3.05	–
Type-2	25	95%	10^8	–	0.80

- (a) **(10 points)** Determine the Defect Level (DL) of the above devices using the appropriate Yield equation in each case. You must show your work and write the value of defect level in parts per million?

For type-1 device,

$$\begin{aligned} DL(T) &= 1 - \left(\frac{\beta + T Af}{\beta + Af} \right)^\beta \\ &= 1 - \left(\frac{0.11 + 0.89 \times 3.05}{0.11 + 3.05} \right)^{0.11} \\ &= 0.012275 \text{ or } \mathbf{12,275 \text{ parts per million}} \end{aligned}$$

For type-2 device,

$$\begin{aligned} DL(T) &= 1 - Y^{1-T} \\ &= 1 - (0.80)^{(1-0.95)} \\ &= 0.011095 \text{ or } \mathbf{11,095 \text{ parts per million}} \end{aligned}$$

- (b) **(5 points)** Although the boards are tested thoroughly before the ICs are placed on them by the board manufacturer, yet the manufacturer has found that the defect level for the bare boards is 0.2%. Determine the percentage of boards that will be found to be faulty after the ICs are placed on the boards. Again, you must show your work.

$$\begin{aligned} Prob(\text{good board}) &= (1 - DL_{\text{board}}) \times (1 - DL_1)^{10} \times (1 - DL_2)^{25} \\ &= (1 - 0.002) \times (1 - 0.012275)^{10} \times (1 - 0.011095)^{25} \\ &= 0.6674 \end{aligned}$$

$$\begin{aligned} Prob(\text{faulty board}) &= 1 - Prob(\text{good board}) \\ &= 0.3326 \text{ or } \mathbf{33.26\%} \end{aligned}$$

3. (10 points) Modeling

- (a) (3 points) K-map of a combinational building block that realizes a single output z and is function of four variable (a, b, c, d) is shown below.

		c d			
		00	01	11	10
a b	0 0		1		
	0 1	1	1		
	1 1	1	1		
	1 0			1	1

Figure 1: A combinational building block with four inputs

For the above circuit, part of the singular cover is listed below (note that the order of inputs is a,b,c,d and the output is z):

$$x10x/1 \quad 0x01/1 \quad 101x/1 \quad x11x/0 \quad 0x1x/0 \quad 00x0/0$$

Now complete the above singular cover by listing the remaining two cubes.

$$x000/0 \\ 100x/0$$

- (b) (3 points) For the above block write two propagation D-cubes such that each must have exactly one D or \overline{D} as an input.

There are a whole lot of these, for example

$$x1\overline{D}x/D, 01\overline{D}x/D, 0D00/D, 1D0x/D, 01\overline{D}1/D$$

- (c) (**4 points**) In the presence of a fault the above building block realizes the following function:

$$f(a, b, c, d) = (a + \bar{c})(b + c + d)(\bar{b} + c + d)(\bar{a} + c)$$

Write two primitive D-cubes of failures for this fault.

You can first list singular cover of the faulty circuit consisting of β_1 and β_0 . Which are:

$$\begin{array}{cccc} 0x01/1 & 1x1x/1 & & \\ xx00/0 & 1x0x/0 & 0x1x/0 & 0xx0/0 \end{array}$$

Now you can write the primitive D-cubes of failures, which are:

$$x100/D, 111x/\bar{D}$$

4. (15 points) **Fault Simulation - Parallel Pattern and Parallel Fault**

The circuit of Fig 2 is to be simulated using a combined parallel pattern and parallel fault simulation method for the following two input patterns:

pattern 1 A B C = 1 0 1
 pattern 2 A B C = 0 1 0

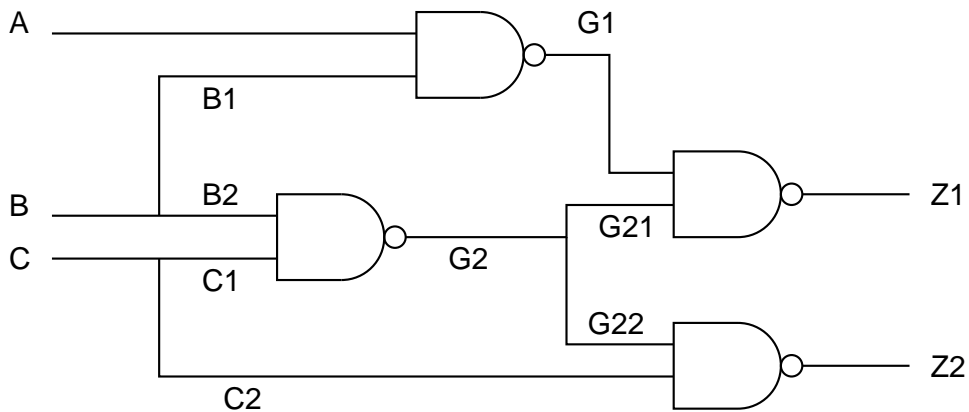


Figure 2: Circuit for parallel pattern parallel fault simulation

Assuming that the word size of a computer is 6 bits and the bits are numbered from 0 to 5 starting from the right most bit as shown in the Fig 3.

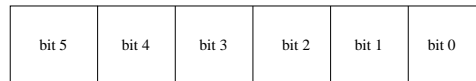


Figure 3: Bits and Faults to be Injected

You are to use bits 0-2 for simulation of pattern 1, and bits 3-5 for simulation of pattern 2. Further, bits 0 and 3 are to be used to simulate the fault free circuit. Bits 1 and 4 are to be used to simulate the fault C1 stuck-at 1, and finally bits 2 and 5 are to be used to simulate fault G2 stuck-at 0.

- (a) (**11 points**) Use the table below to show values on each line for each test for the circuit without and with two faults of interest.

Line Name	Sim value	Line Name	Sim value
A	0 0 0 1 1 1	G21	0 0 1 0 1 1
B	1 1 1 0 0 0	G22	0 0 1 0 1 1
C	0 0 0 1 1 1	Z1	1 1 0 1 0 0
B1	1 1 1 0 0 0	Z2	1 1 1 1 0 0
B2	1 1 1 0 0 0		
C1	0 1 0 1 1 1		
C2	0 0 0 1 1 1		
G1	1 1 1 1 1 1		
G2	0 0 1 0 1 1		

- (b) (**4 points**) Now, indicate which of the faults will be detected and at which output and by which test. Thus list clearly for each fault if the fault at C1 (G1) detected (not detected) by pattern 1 (pattern 2) at output Z1 (output Z2).

Output Z1 will detect: fault G2 stuck-at-0 by pattern 1, fault C1 stuck-at-1 by pattern 2, fault G2 stuck-at-0 by pattern 2

Output Z2 will detect: fault G2 stuck-at-0 by pattern 1

5. (15 points) SCOAP Computation

Consider the circuit shown in Figure 4 for SCOAP based testability analysis. You

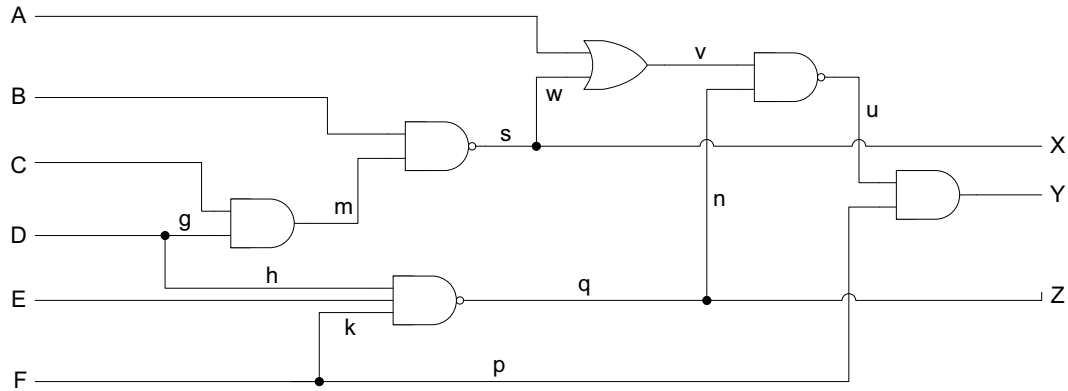


Figure 4: Circuit for SCOAP Computations

are to compute the SCOAP observability values for each line in this circuit. The controllability values (CC0 and CC1) for each line is already computed and is given in the table below. You may notice that input controllability values are not 1 and 1 - that is because this circuit is part of some other circuit. Also, the following additional modification was used while computing CC0 and CC1 values.

You may recall while computing the SCOAP value at the output of a gate we compute a function (such as sum or min) of the SCOAP values at the inputs and then add one to it. Instead of adding one, while preparing the table below we added a constant equal to the number of inputs to the gate.

For example, to compute the CC0 value at the output of a three input NAND gate we used the following equation:

$$CC0(\text{NAND3}) = CC1(\text{input1}) + CC1(\text{input2}) + CC1(\text{input3}) + 3$$

While computing the observability values, instead of using the constant 1, you are to use similar values as we used while computing the controllability values.

Line	Controllability		Observability	Line	Controllability		Observability
	CC0	CC1	CO		CC0	CC1	CO
A	10	5	84	n	56	10	49
B	6	6	36	p	9	30	54
C	2	9	33	q	56	10	23
D	7	11	31	s	30	6	12
E	25	12	67	u	19	44	40
F	9	30	49	v	42	7	52
g	7	11	31	w	30	6	64
h	7	11	68	X	30	6	12
k	9	30	49	Y	11	76	8
m	4	22	20	Z	56	10	23

6. (15 points) **Combinational Test Generation - PODEM**

A PODEM like test generator is used to generate a test for the fault line v s-a-1 in the circuit shown in Figure 6. Note that this circuit is same as the one used for SCOAP value calculations. However, for this problem, you can use the controllability information shown in the table for the SCOAP problem.

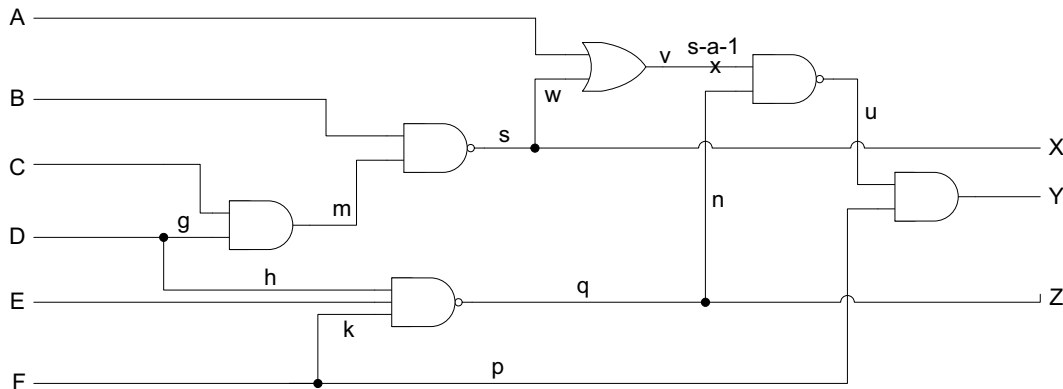


Figure 5: Circuit for Test Generation

- (a) Complete the table below for the test generation process and during the test generation follow the rules given below:

While backtracing, use the easy/hard heuristic.

While assigning a value at an input, always assign a 1 before a 0.

Do not perform x-path check.

In the table fill the necessary entries for each step.

- (b) Construct the decision tree.
- (c) Write the generated test. A B C D E F = **0 1 1 1 0 1**

Step	Objective	Backtrace path	PI assign	D-front	comment
1	v=0	v-w-s-m-g-D	D=1		Objective not satisfied
2	v=0	v-w-s-m-C	C=1		Objective not satisfied
3	v=0	v-w-s-B	B=1		Objective not satisfied
4	v=0	v-A	A=1		failure/backtrack
5	v=0	v-A	A=0	u	fault excited
6	n=1	n-q-F	F=1	u	Objective not satisfied
7	n=1	n-q-E	E=1	u	failure/backtrack
8	n=1	n-q-E	E=0	Z	test found
9					

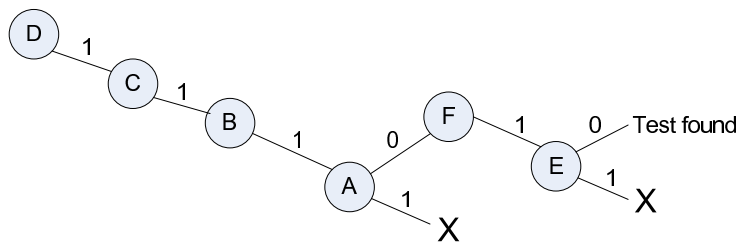


Figure 6: Decision tree

7. (14 points) Checking Experiment

- (a) (8 points) State table of a finite state machine with four states, A, B, C, and D; and an input alphabet consisting of x, and y; is given below.

Table 1: State Machine for Problem 7

	Input	
	x	y
A	B/0	D/0
B	A/0	B/0
C	D/1	A/0
D	D/1	C/0

Find the shortest synchronizing sequence for this circuit if one exists. You **must show your work otherwise no points** will be awarded.

The shortest synchronizing sequence is **xyxyx**

- (b) (6 points) A state machine has two distinguishing sequences which are “a b c” and “a c d”. It also has a synchronizing sequence “d e f a” which is not a distinguishing sequence. Note these may or may not be the shortest sequences. Answer the following regarding this state machine and where necessary you must give a brief explanation:
- Is the sequence “a b c d” a distinguishing sequence for this machine. Answer yes or no.
Yes.
 - Is the sequence “d e f a b c” a distinguishing sequence for this machine. Give a brief explanation for your answer.
Generally speaking the answer is No. But if the synchronizing sequence “d e f a” is such that it also acts like a DS then the sequence “d e f a b c” will also be a DS. Note that after the input sequence “d e f a” all the states are merged to one state and no additional input provides any information about the start state.
 - Is the sequence “a c d e f a” a distinguishing sequence for this machine. Answer yes or no.
Yes.

- iv. Can the sequence “a b c” possibly be a synchronizing sequence for this machine. Give a brief explanation for your answer.

It can possibly be, but there are is one reason it may not be a SS in this case and that is if the sequence “d e f a” is shortest SS than a sequence of length 3 can not be a SS. However, the problem states that “d e f a” may not be shortest SS, in which case the answer is it is possible that “a b c” is a SS but we can not be sure without some more information.

Many of you implied that every DS is a SS by saying that at the end of DS we know the end state. But this argument is not valid because for DS we know the outputs and for SS we ignore the outputs and without looking at the outputs we can tell the end state for DS.