1. (7 points) (Bushnell and Agrawal) Problem 5.25

Since the size of fault population \( N_p = 10^5 \) is very large compared to the sample size \( N_s = 4,000 \), we use the approximation of Equation 5.5 (page 123 in the book.)

Sample coverage, \( x = \frac{3,900}{4,000} = 0.975 \)

Using Equation 5.8 (see page 123 in the book), we get

\[
3\sigma \text{ coverage estimate} = x \pm \frac{4.5}{N_s}\sqrt{1 + 0.44N_sx(1-x)}
\]

\[
= 0.975 \pm \frac{4.5}{4,000}\sqrt{1 + 0.44 \times 4,000 \times 0.975 \times 0.025}
\]

\[
= 0.975 \pm 0.0075 \text{ or } 97.50 \pm 0.75 \text{ percent}
\]

2. (7 points) For the circuit shown in the Figure 1, perform Parallel pattern single fault simulation targeting the fault h stuck-at-1. You are given 4 vectors to simulate and the vectors are encoded as 11 for 1, 00 for 0 and 01 for X. Make sure to exchange the bits when inversion occurs.

From the figure, we can see that vectors 1 and 2 produce different outputs from the fault free circuit at the output and hence the fault is detectable by these two vectors. However vector 3 fails to distinguish between the faulty and fault-free circuits at the PO and hence the fault is undetectable by vector 3. Vector 4 is a special case in that it produces an X at the primary output and hence the fault is said to be potentially detectable by vector 4.
3. (8 points) For the circuit in Figure 2, which is the same circuit as problem 1 but with different data, perform parallel fault simulation for the vector abc = 101 and three faults: e stuck-at-1, h stuck-at-1 and n stuck-at-1 to determine which faults will be detected at the primary output.

From the figure, e s-a 1 and h s-a 1 are detectable by the input, but n s-a 1 is not.

4. (8 points) For the same circuit as Problem 1, perform deductive fault simulation to determine all faults that will be detected at the primary output ‘o’ for the test vector a=1 b=0 c=1.

Solution:

$L_a = \{a/0\}$
$L_b = \{b/1\}$
$L_c = \{c/0\}$
$L_d = \{b/1, d/1\}$
$L_e = \{b/1, e/1\}$
$L_f = L_d \cap L_a \cup L_f = \{b/1, d/1, f/1\}$
$L_g = L_e \cap L_c \cup L_g = \{b/1, e/1, g/0\}$
$L_h = L_f \cup L_h = \{b/1, d/1, f/1, h/1\}$
$L_i = L_f \cup L_i = \{b/1, d/1, f/1, i/1\}$
$L_j = L_g \cup L_j = \{b/1, e/1, g/0, j/0\}$
Thus the faults in the final list d/1, e/1, f/1, g/0, h/1, k/0, m/0, n/0, o/1 can be detected with the given vector.

5. (10 points) (Bushnell and Agrawal) Problem 7.13 In this problem and in the next problem, use the following table type to explain the steps of your process. You may not need all the columns, therefore, you can leave the columns not used as blank columns.

<table>
<thead>
<tr>
<th>Step No.</th>
<th>Objective</th>
<th>Action</th>
<th>Imp. stack</th>
<th>Implied signal values</th>
<th>D front.</th>
<th>X path</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>abc....</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In the table below I show only the columns needed for this problem.
<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
<th>Impl. stack</th>
<th>Forward implications</th>
<th>D-frontier</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Fault act.</td>
<td>(h = 0)</td>
<td>(h = 0, h1 = \overline{D}, i2 = 0)</td>
<td>(i1)</td>
</tr>
<tr>
<td>2</td>
<td>D-prop.</td>
<td>(g1 = 1, h = 0)</td>
<td>(g1 = 1, h = 0, h1 = \overline{D}) (i1 = \overline{D}, i2 = 0)</td>
<td>PO</td>
</tr>
<tr>
<td>3</td>
<td>Justify</td>
<td>(e1 = 1, g1 = 1) (h = 0)</td>
<td>(e1 = 1, g1 = 1, h = 0) (h1 = \overline{D}, i1 = \overline{D}, i2 = 0)</td>
<td>PO</td>
</tr>
<tr>
<td>4</td>
<td>Justify</td>
<td>(a = 1, b = 1) (e1 = 1, g1 = 1) (h = 0)</td>
<td>(a = 1, b = 1, e1 = 1, g1 = 1) (e2 = 1, g1 = 1, g2 = 1) (h = 0, h1 = \overline{D}, i1 = \overline{D}) (i2 = 0)</td>
<td>PO</td>
</tr>
</tbody>
</table>

Test found: \((a, b, c, d, h, k) = (1, 1, X, X, 0, X); i1 = \overline{D}\)

The above figure shows the circuit and the signal values specified by \(D\)-algorithm.

![Figure 3: Figure for Problem 5](image-url)
6. (15 points) (Bushnell and Agrawal) Problem 7.8

Note: While you are performing the PODEM algorithm, follow the rules given below.

- Order: Try to excite the fault first then propagate.
- Backtrace: Follow a path from the objective to a primary input while always following the alphabetical order (e.g. if a gate has input A and B, backtrace on that gate goes to line A first).
- PI assignment: Always assign 0 first, then assign 1 in case of backtrack.
- Choice of D or D_bar: Always try to propagate a D or D_bar from the D-frontier which has the shortest path to the primary output. In the case of tie, follow the alphabetical order.

The following table gives the steps of PODEM (see Problem 7.5 for an explanation of X-path, though in this case X-path check is not required):

<table>
<thead>
<tr>
<th>Step No.</th>
<th>Objective</th>
<th>Action</th>
<th>Imp. stack</th>
<th>Implied signal values</th>
<th>D front.</th>
<th>X path</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>g = 0(D)</td>
<td>Backtrace</td>
<td>C = 0</td>
<td>C = 0, h = 0</td>
<td>φ</td>
<td>ok</td>
</tr>
<tr>
<td>2</td>
<td>g = 0(D)</td>
<td>Backtrace</td>
<td>D = 0, C = 0</td>
<td>C = 0, D = 0, g = 0(D)</td>
<td>φ</td>
<td>none</td>
</tr>
<tr>
<td>3</td>
<td>g = 0(D)</td>
<td>Backtrack</td>
<td>D = 1, C = 0</td>
<td>C = 0, D = 1, g = 1, h = 0</td>
<td>φ</td>
<td>none</td>
</tr>
<tr>
<td>4</td>
<td>g = 0(D)</td>
<td>Backtrack</td>
<td>C = 1</td>
<td>C = 1, g = 1, h = 1, m = 1</td>
<td>φ</td>
<td>none</td>
</tr>
<tr>
<td>5</td>
<td>g = 0(D)</td>
<td>Backtrack</td>
<td>Empty</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Algorithm termination: g = 0(D) with X-path impossible; fault g s-a-1 is redundant. 3 backtracks.

7. (15 points) A PODEM like test generator is used to generate a test for the line v s-a-1 in the circuit shown in Figure 5.

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

- For this problem, assume that the order of primary input assignments is D, C, B, A, F, then E.
- 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. $A \ W C \ D \ E \ F = 011101$
<table>
<thead>
<tr>
<th>Step</th>
<th>Objective</th>
<th>Backtrace path</th>
<th>PI assign</th>
<th>D-front</th>
<th>comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>v=0</td>
<td>v-w-s-m-g-D</td>
<td>D=1</td>
<td></td>
<td>Objective not satisfied</td>
</tr>
<tr>
<td>2</td>
<td>v=0</td>
<td>v-w-s-m-C</td>
<td>C=1</td>
<td></td>
<td>Objective not satisfied</td>
</tr>
<tr>
<td>3</td>
<td>v=0</td>
<td>v-w-s-B</td>
<td>B=1</td>
<td></td>
<td>Objective not satisfied</td>
</tr>
<tr>
<td>4</td>
<td>v=0</td>
<td>v-A</td>
<td>A=1</td>
<td></td>
<td>failure/backtrack</td>
</tr>
<tr>
<td>5</td>
<td>v=0</td>
<td>v-A</td>
<td>A=0</td>
<td>u</td>
<td>fault excited</td>
</tr>
<tr>
<td>6</td>
<td>n=1</td>
<td>n-q-F</td>
<td>F=1</td>
<td>u</td>
<td>Objective not satisfied</td>
</tr>
<tr>
<td>7</td>
<td>n=1</td>
<td>n-q-E</td>
<td>E=1</td>
<td>u</td>
<td>failure/backtrack</td>
</tr>
<tr>
<td>8</td>
<td>n=1</td>
<td>n-q-E</td>
<td>E=0</td>
<td>Y</td>
<td>test found</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5: Decision tree
8. (30 points) You are required to work with the circuit n432 and 10 test vectors.

The circuit n432 can be found here:
/userspace/e/ece553/TESTCAD/nets/hw3-circuits/n432

The 10 Test Vectors can be found here:
/userspace/e/ece553/TESTCAD/nets/hw3-circuits/n432.10vec

Use the testcad tools to answer the following questions.

a) Apply the given 10 test vectors to n432 and determine the fault coverage.

*** SFSP (3-valued) Fault Simulation: Net n432 ***
Total Vectors: 10
Total Fault: 524
Detected Fault: 250
Fault Coverage: 47.71 %
CPU Time: 0.000 seconds

b) Use PODEM to generate tests for each fault (without intervening fault simulation). Indicate how many tests are generated and which faults are found to be untestable or aborted.

Result can vary slightly depending on backtrack limit you specify for ‘podem’. Following statistics are obtained with backtrack limit of 1000000.

Total Faults 524
Undetectable faults found 4
Aborted faults 0
Total Time 10 sec.
Time/Fault 0.018384 sec.
Backtracks 2.02574e+06
Implications 4.05777e+06

Undetected fault list :

259 0 1
347 0 1
379 0 1
429 393 1

c) Choose any 10 test vectors from your list of test vectors, then fill in the X’s using ‘randvec’. Determine the fault coverage of your 10 vectors and compare that with the coverage of 10 vectors obtained in part (a).

Any 10 vectors can be chosen for your answer, following result is only one example.
*** SFSP (3-valued) Fault Simulation: Net n432 ***

Total Vectors: 10
Total Fault: 524
Detected Fault: 241
Fault Coverage: 45.99%
CPU Time: 0.000 seconds