CSE 244A Homework Two

November 2, 2019

Only Problem Set Part B will be graded. Turn in only Problem Set Part B which will be due on November 15, 2019 (Friday) at 9:00 am.

1 Problem Set Part A

All questions in this part are from your textbook by Bushnell & Agrawal.

- 8.2, 8.3, 8.4, 8.5, 8.6, 8.7, 8.9, 8.15
- 5.1, 5.5, 5.15, 5.16, 5.17, 5.18, 5.19, 5.21
- 3.4, 3.5, 3.6
- 14.3, 14.8, 14.14
Problem Set Part B

I. (Sequential Machine Test)

In this question we will examine the issue of finite state machine encoding as it bears on the testability of sequential machines. We will work with a Mealy implementation of a recognizer that outputs a 1 upon detecting the string 0100. (This being an overlapping sequence detector an input of 0100100 will result in outputs of 1 upon receiving the 4th and 7th 0, i.e. the fourth 0 will do double duty as the final input of the first sequence and the initial input of the second sequence.)

The FSM for this recognizer is specified as below. Dashed lines denote transitions on 1 input, while straight lines denote transitions on 0 input. Outputs are shown on the transitions following a slash.

(Hint: You may find functional approaches, such as identifying correct and faulty machine behaviors useful in addressing some parts of this question.)

(Parts A-D) The $S_0(00)$, $S_1(01)$, $S_2(11)$, and $S_3(10)$ encoding generates the following Kmap, whose next state equations for $(Q_0, Q_1)$ can be specified as follows:

<table>
<thead>
<tr>
<th></th>
<th>00</th>
<th>01</th>
<th>11</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>01</td>
<td>01</td>
<td>10</td>
<td>01</td>
</tr>
<tr>
<td>1</td>
<td>00</td>
<td>11</td>
<td>00</td>
<td>11</td>
</tr>
</tbody>
</table>

$$Q_0^+ = Q_1(X \oplus Q_0) + Q_0(X \oplus Q_1)$$
$$Q_1^+ = (Q_0 \oplus Q_1) + X'Q_0'$$

The output function $Z$ is specified as follows:

$$Z = X'Q_0Q_1'$$
(Part A) Please specify the s-graph for this sequential circuit.

(Part B) Please identify an initialization sequence if possible. You may find functional reasoning expedient in tackling this part. If you end up using functional reasoning, please explain whether gate-level propagation techniques can resolve the question of initializability.

(Part C) You are asked to determine an approach for identifying a test sequence for the s-a-0 fault at the X input line (the stem fault).
Please specify whether this is a traditional test or a multiple observation test.

(Part D) You are now asked to determine an approach for identifying a test sequence for the s-a-0 fault at the output of the 3-input AND gate that is used to determine the Z output.
Please specify whether this is a traditional test or a multiple observation test.

(Part E-H) You will now work with the alternate encoding that flips the encodings for $S_2$ and $S_3$. The $S_0(00)$, $S_1(01)$, $S_2(10)$, and $S_3(11)$ encoding generates the following Kmap, whose next state equations for $(Q_0, Q_1)$ can be specified as follows:

<table>
<thead>
<tr>
<th></th>
<th>00</th>
<th>01</th>
<th>11</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>11</td>
</tr>
<tr>
<td>1</td>
<td>00</td>
<td>10</td>
<td>10</td>
<td>00</td>
</tr>
</tbody>
</table>

$$Q_0^+ = XQ_1 + X'Q_0Q_1'$$
$$Q_1^+ = X'$$

The output function $Z$ is specified as follows:

$$Z = X'Q_0Q_1$$
(Part E) Please specify the s-graph for this sequential circuit.

(Part F) Please identify an initialization sequence if possible. You may find functional reasoning expedient in tackling this part. If you end up using functional reasoning, please explain whether gate-level propagation techniques can resolve the question of initializability.

(Part G) You are asked to determine an approach for identifying a test sequence for the s-a-0 fault at the X input line (the stem fault).

Please specify whether this is a traditional test or a multiple observation test.

(Part H) You are now asked to determine an approach for identifying a test sequence for the s-a-0 fault at the output of the 3-input AND gate that is used to determine the Z output.

Please specify whether this is a traditional test or a multiple observation test.
II. (Simulation-based Verification)

Verification can be effected by applying all possible inputs to a circuit. It can be easily seen that this approach would quickly run into computational explosion considerations. Utilization of even partial knowledge as to how the circuit is implemented may deliver dramatic savings in this regard while retaining full coverage.

(Part A) A favorite implementation of an adder \((X+Y)\) is the carry-ripple implementation composed of single-bit adders connected through a carry chain. The single bit adders implement the functionality given below; actual implementations may vary as you know from your study of adder circuits.

\[
\begin{align*}
\text{sum} &= x \oplus y \oplus c_{in} \\
\text{cout} &= xy + xc_{in} + yc_{in}
\end{align*}
\]

Even though the individual single-bit adder may be implemented in a variety of ways, the component level adder information can be utilized to minimize the number of vectors needed for a 16-bit adder. What is the minimal number of vectors that are needed to simulate (not test) the 16-bit adder? Please enumerate the vectors with their 33 bit inputs (you can obviously abbreviate the input specification in the case of regularity).
(Part B) A favorite implementation of a subtractor \((X - Y)\) is the borrow-ripple implementation composed of single-bit subtractors connected through a borrow chain. The single bit subtractors implement the functionality given below; actual implementations may vary as you know from your study of subtractor circuits.

\[
diff = x \oplus y \oplus b_{in}
\]
\[
b_{out} = x'y + x'b_{in} + yb_{in}
\]

Even though the individual single-bit subtractor may be implemented in a variety of ways, the component level subtractor information can be utilized to minimize the number of vectors needed for a 16-bit subtractor. What is the minimal number of vectors that are needed to simulate (not test) the 16-bit subtractor? Please enumerate the vectors with their 33 bit inputs (you can obviously abbreviate the input specification in the case of regularity).

(Part C) As in the case of the adder, the longest path in the case of the subtractor is the ripple path, in this case, the ripple of the borrow.

Please provide vector(s) and discuss their application approach that will ensure exercising the longest path in the design? Please provide at least one such set.
III. (Fault Simulation)

In this question, you will work yet once again with the 4-NAND implementation of the XOR gate; hopefully, you are familiar with it by now! This question will ask you to show your familiarity with a number of fault simulation techniques we examined in class.

To simplify matters, we have come up with a reduced fault list by taking a fault equivalence pass through the circuit, in the process pruning the fault list by a third. Since equivalent faults do not need to be multiply accounted for, please use the reduced list throughout to speed up your computations. The reduced fault list is:

\[ G1, G0, C1, D1, A0, A1, B0, B1, J1, H1, E1, I1, F1, K1, L0, L1 \]

(Part A) Since you will be working throughout, with NAND gates only, please identify for starters the fault list propagation rules for a NAND gate for a *deductive fault simulator*.

(Part B) Please identify the faults detected by each of the possible 4 input combinations to this XOR circuit by performing *deductive fault simulation*. We are providing you 4 copies of the circuit at the end of the test to enable you to annotate them appropriately. (You may be able to take advantage of certain symmetry aspects to reduce the amount of work needed).
(Part C) In parallel fault simulation, the circuit is modified in order to reflect the effect of the fault (fault injection) while performing parallel fault simulation. Please show the necessary upgrades to the circuit in order to model the $G_0$ and $K_1$ faults in the context of parallel fault simulation.

(Part D) As you can see the list of faults in this circuit is 16. Please use the circuit shown below to identify the values on the circuit when the $\langle 1, 1 \rangle$ vector is applied. Follow this up by showing the values on the 16 faults (in the order defined) as a parallel fault simulator performs its task.

(Part E) Given the test vectors needed to test this circuit, please provide a possible order of application to a concurrent fault simulator that will minimize computational cost. Please justify your answer.