CIS 4930/EEL 6706 VLSI Testing Homework Discussion

Hao Zheng
Practice 1
1. Use the parallel fault simulation to simulate the circuit shown in Figure 5.10 with input vector \textbf{abcde} = 11110. Show simulation results in each step in terms of values found for each wire level-by-level. Determine the set of faults that are detected at the end of simulation. Consider faults \( F = \{ a_0, a_1, b_1, c_0, c_1, d_1, e_0, g_0, h_0, h_1 \} \).

**Figure 5.10**
\textbf{abcde} = \textbf{1110}. \quad F = \{a_0, a_1, b_1, c_0, c_1, d_1, e_0, g_0, h_0, h_1\}.

\begin{align*}
a & : 01111111111 \\
b & : 11111111111 \\
c & : 11101111111 \\
d & : 11111111111 \\
e & : 00000000000
\end{align*}

\textbf{Figure 5.10}
\( \text{abcde} = 11110 \). \( F = \{ a_0, a_1, b_1, c_0, c_1, d_1, e_0, g_0, h_0, h_1 \} \).

\[
\begin{align*}
\text{a:} & \quad 01111111111 \\
\text{b:} & \quad 11111111111 \\
\text{c:} & \quad 11101111111 \\
\text{d:} & \quad 11111111111 \\
\text{e:} & \quad 00000000000 \\
\text{f:} & \quad 01111111111
\end{align*}
\]

F and other vectors are computed by following the formula on page 136 in the book.
abcde = 11110. \( F = \{ a_0, a_1, b_1, c_0, c_1, d_1, e_0, g_0, h_0, h_1 \} \).

c: 11101111111  \quad e: 000000000000  \quad g: 11101110111

d: 11111111111  \quad f: 01111111111  \quad h: 111011110111

Figure 5.10
abcde = 11110. \( F = \{ a_0, a_1, b_1, c_0, c_1, d_1, e_0, g_0, h_0, h_1\}. \\
d: 11111111111 \quad f: 01111111111 \\
e: 00000000000 \quad g: 11101110111 \\
h: 11101111011 \\
j: 00000000000 \\
i: 11101111011

Figure 5.10
$\text{abcde} = 11110$. $F = \{ a_0, a_1, b_1, c_0, c_1, d_1, e_0, g_0, h_0, h_1 \}$.

$e: 000000000000 \quad j: 000000000000 \quad m: 111011110111$

$i: 111011110111 \quad k: 111011110111$

11110 detects $\{c_0, h_0\}$.
5.3 For the circuit and the fault set used in Example 5.1, determine the faults detected by \( abcde=11010 \) by the deductive simulation.
$abcde = 11010,$ \hspace{1cm} $F = \{ a_0, a_1, b_1, c_0, c_1, d_1, e_0, g_0, h_0, h_1 \}$.

$L_a = \{a_0\}, \hspace{1cm} L_b = \emptyset, \hspace{1cm} L_c = \{c_1\}, \hspace{1cm} L_d = \emptyset, \hspace{1cm} L_e = \emptyset$
\text{abcde=11010,} \quad F = \{ a_0, a_1, b_1, c_0, c_1, d_1, e_0, g_0, h_0, h_1 \}.

L_a = \{ a_0 \}, \quad L_b = \emptyset, \quad L_c = \{ c_1 \}, \quad L_d = \emptyset, \quad L_e = \emptyset

L_f = \{ a_0 \}, \quad L_g = \{ c_1 \}, \quad L_h = \{ c_1, h_1 \}, \quad L_i = \{ a_1, h_1 \}, \quad L_j = \{ c_1, h_1 \}, \quad L_k = \{ c_0, h_1 \}, \quad L_m = \emptyset
\( \text{abcde}=11010, \quad F = \{ a_0, a_1, b_1, c_0, c_1, d_1, e_0, g_0, h_0, h_1 \}. \)

\[ L_a = \{ a_0 \}, \quad L_b = \emptyset, \quad L_c = \{ c_1 \}, \quad L_d = \emptyset, \quad L_e = \emptyset \]

\[ L_f = \{ a_0 \}, \quad L_g = \{ c_1 \}, \quad L_h = \{ c_1, h_1 \}, \]

\[ L_j = \{ a_0 \}, \quad L_i = \{ c_1, h_0 \}, \]

\[ L_k = \{ c_1, h_1 \}, \]

\[ L_m = \{ a_0, c_1, h_1 \}, \]
5.7 For the circuit and the fault set used in Example 5.1, determine the faults detected by \( abcde=00110 \) by the concurrent simulation
$abcde=00110,$  
$F = \{ a_0, a_1, b_1, c_0, c_1, d_1, e_0, g_0, h_0, h_1 \}.$

<table>
<thead>
<tr>
<th></th>
<th>$a$</th>
<th>$b$</th>
<th>$f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a0$</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$a1$</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$b1$</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
$abcde=00110,$

$F = \{ a_0, a_1, b_1, c_0, c_1, d_1, e_0, g_0, h_0, h_1 \}.$

\[
\begin{array}{|c|c|c|}
\hline
 & f & g & j \\
\hline
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 \\
\hline
\end{array}
\]

Composed event: $j:0, (g0, 1)$
abcde=00110,  
F = \{ a_0, a_1, b_1, c_0, c_1, d_1, 
e_0, g_0, h_0, h_1 \}.  

Composed event: \( i:1, (h0, 0) \)
**abcde=00110,**
\[ F = \{ a_0, a_1, b_1, c_0, c_1, d_1, \\
    e_0, g_0, h_0, h_1 \}. \]

\[
\begin{array}{|c|c|c|c|}
\hline
& i & e & k \\
\hline
h0 & 1 & 0 & 1 \\
\hline
\end{array}
\]

Composed event: \( k:1, (h0, 0) \)
\[ abcd e = 00110, \]
\[ F = \{ a_0, a_1, b_1, c_0, c_1, d_1, e_0, g_0, h_0, h_1 \}. \]
HW 3
1. Refer to Figure 4.30, answer the following questions about faults $f: a \ s-a-0$ and $g: b \ s-a-0$. Given sufficient explanations to your answers.

a) Are $f$ and $g$ functionally equivalent?

b) Are $f$ and $g$ functionally equivalent under some tests $T$? If so, show $T$.

![Figure 4.30](image-url)
2. Refer to Figure 4.14. Show why \( f : z_2 s-a-0 \) does not dominate \( g : y_1 s-a-1 \) under test \( T_g = \{10\} \).

![Figure 4.14](image-url)
3. Refer to Figure 4.17. Explain whether \( f : c \text{-s-a-1} \) and \( g : d \text{-s-a-1} \) are structurally equivalent?

![Figure 4.17](image-url)
4.11 Prove that in a combinational circuit, if two faults dominate each other, then they are functionally equivalent.

**Definition 4.8:** Let $T_g$ be the set of all tests that detect a fault $g$. We say that a fault $f$ dominates the fault $g$ iff $f$ and $g$ are functionally equivalent under $T_g$.

**Definition 4.4:** Two faults $f$ and $g$ are said to be functionally equivalent iff $Z_f(x) = Z_g(x)$. 
4.15

a. For the circuit and the test in Figure 4.22, show that \( j \) s-a-0 is detected, but \( k \) s-a-0 and \( m \) s-a-0 are not.

![Figure 4.22](image-url)
4.15

b. For the circuit and the test in Figure 4.23, show that both $x_1$ s-a-0 and $x_2$ s-a-0 are detected, but $x$ s-a-0 is not.

Figure 4.23
4.17 Consider the circuit of Figure 4.31. Let $f$ be the fault $b$ s-a-0 and $g$ be a s-a-1.

a. Does $f$ mask $g$ under the test 0110? Does $f$ mask $g$ under the test 0111?

b. Are the faults $f$ and $\{f, g\}$ distinguishable?

\[ \text{Figure 4.31} \]
HW 2
Problems

3.1 Consider a function \( f(x_1, x_2, x_3) \) defined by the following primitive cubes: \( x10 | 0, 11x | 0, x0x | 1, \) and \( 011 | 1 \). Determine the value of \( f \) for the following input vectors: \( 10u, 01u, u1u, \) and \( u01 \).

If an input vector is not compatible with any cubes, its output is \( u \).

Input vector \( 10u, 01u, u01 \) are compatible with \( x0x \), their output is 1.

\( 01u \) is not compatible with any cubes, so its output is \( u \).

\[
\begin{array}{c|ccc}
\text{AND} & 0 & 1 & u \\
0 & 0 & 0 & 0 \\
1 & 0 & 1 & u \\
u & 0 & u & u \\
\end{array}
\]

\[
\begin{array}{c|ccc}
\text{OR} & 0 & 1 & u \\
0 & 0 & 1 & u \\
1 & 1 & 1 & 1 \\
u & u & 1 & u \\
\end{array}
\]

\[
\begin{array}{c|ccc}
\text{NOT} & 0 & 1 & u \\
1 & 0 & u \\
\end{array}
\]

**Figure 3.2** Truth tables for 3-valued logic

\[
\begin{array}{c|ccc}
\cap & 0 & 1 & x & u \\
0 & 0 & \varnothing & 0 & \varnothing \\
1 & \varnothing & 1 & 1 & \varnothing \\
x & 0 & 1 & x & u \\
u & \varnothing & \varnothing & u & u \\
\end{array}
\]

**Figure 3.3** Modified intersection operator
Problems – 3.3

Use compiled simulation to simulate the circuit below with 01, 11. Initial state is u

A=00, B=11

A=11, B=11

<table>
<thead>
<tr>
<th></th>
<th>B</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDA</td>
<td>B=11</td>
<td>E=01</td>
</tr>
<tr>
<td>AND</td>
<td>11 AND 01</td>
<td>11 AND 01</td>
</tr>
<tr>
<td>INV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STA</td>
<td>E=01</td>
<td>E=01</td>
</tr>
<tr>
<td>OR</td>
<td>01 OR 00</td>
<td>01 OR 11</td>
</tr>
<tr>
<td>STA</td>
<td>F=01</td>
<td>F=11</td>
</tr>
<tr>
<td>STA</td>
<td>Q=01</td>
<td>Q=11</td>
</tr>
</tbody>
</table>

0 – 00
1 – 11
u – 01
The set of primitive elements used in most simulation systems includes the basic gates - AND, OR, NAND, and NOR. These gates can be characterized by two parameters, the controlling value $c$ and the inversion $i$.

The value of an input is said to be controlling if it determines the value of the gate output regardless of the values of the other inputs; then the output value is $c(±)i$.

Figure 3.20 shows the general form of the primitive cubes of any gate with three inputs.

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$c x x$</td>
<td>$c \oplus i$</td>
<td>AND</td>
<td>0</td>
</tr>
<tr>
<td>$x c x$</td>
<td>$c \oplus i$</td>
<td>OR</td>
<td>1</td>
</tr>
<tr>
<td>$x x c$</td>
<td>$c \oplus i$</td>
<td>NAND</td>
<td>0</td>
</tr>
<tr>
<td>$\bar{c} \bar{c} \bar{c}$</td>
<td>$\bar{c} \oplus i$</td>
<td>NOR</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 3.20  Primitive cubes for a gate with controlling value $c$ and inversion $i$
Problems

4.1 Find a circuit that has an undetectable stuck fault.

See example 4.3, and figure 4.8 with $Y_{a-0}$

4.2 Is it possible to have a combinational circuit with a signal $S$ and test $t$ such that $t$ detects both $S_{a-0}$ and $S_{a-1}$?

Not possible – same test always produces the same output
Problems

4.6 For the circuit of Figure 4.28

a. Find the set of all tests that detect the fault \( c \ s-a-1 \).
b. Find the set of all tests that detect the fault \( a \ s-a-0 \).
c. Find the set of all tests that detect the multiple fault \( \{c \ s-a-1, a \ s-a-0\} \).

![Figure 4.28](image-url)
Problems

4.8 For the circuit of Figure 4.30

a. Find the set of all tests that detect the fault b s-a-1.
b. Find the set of all tests that distinguish the faults a s-a-0 and c s-a-0.
c. Find the set of all tests that distinguish the multiple faults \{a s-a-0, b s-a-1\} and \{c s-a-0, b s-a-1\}.

![Figure 4.30](image-url)
2.1 Determine whether the following cubes can be cubes of a function $Z(x_1, x_2, x_3, x_4)$.

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>$x$</td>
<td>$x$</td>
<td>0</td>
</tr>
<tr>
<td>$x$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>$x$</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>$x$</td>
<td>1</td>
</tr>
<tr>
<td>$x$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

No. The input vector 0010 covered by the first and the last cubes result in different outputs; input vector 0000 is not defined in any cubes.
2.2 Construct a binary decision diagram for the exclusive-OR function of two variables.
Determine whether $x01x$ and $x0x1$ are compatible or not.
6.1 For the circuit of Figure 6.89, generate a test for the fault \( g \) \( s-a-1 \). Determine all the other faults detected by this test.

![Figure 6.89](image-url)
6.3 Use only implications to show that the fault $f_{s-a-0}$ in the circuit of Figure 6.12(a) is undetectable.
6.4 Construct the truth table of an XOR function of two inputs using the five logic values 0, 1, x, D, and D. 

6.5 Can a gate on the $D$-frontier have both a $D$ and a $\bar{D}$ among its input values?

6.6 For the circuit of Figure 6.90, perform all possible implications starting from the given values.
2. Answer the following questions.

(a) Show an example of a gate that belongs to $D$-frontier.
(b) Show an example of a gate that belongs to $J$-frontier.
(c) Briefly explain what reversing incorrect decisions does.
6.8 Consider the circuit and the values shown in Figure 6.92. Let us assume that trying to justify $d=0$ via $a=0$ has lead to an inconsistency. Perform all the implications resulting from reversing the incorrect decision.

Figure 6.92

$D$-frontier = \{ $i$, $j$ \}

$J$-frontier = \{ ..., $d$ \}
HW 5
8.4 Derive a pseudoexhaustive test set for the circuit of Figure 8.8(a) by partitioning it into the following three segments: (1) the subcircuit whose output is \( h \), (2) gate \( y \), and (3) gates \( g \) and \( x \). Compare your results with those obtained in Example 8.4.
9.2  Partition the circuit shown in Figure 9.54 using the partitioning scheme illustrated in Figure 9.11. Attempt to keep the number of inputs to $C_1$ and $C_2$ close to the same, and try to minimize the number of signals between the two partitions.
2. Answer the following questions.

   (a) Concisely describe the differences between SSF testing and functional testing.
   (b) Define controllability and observability.
   (c) Explain briefly how test points improve controllability and observability.
   (d) What is the main limitation of the multiplexed test points?
   (e) What are the benefits of partitioning a large combinational logic circuit? Give an example to illustrate your answer.
   (f) Describe how scan registers work.
   (g) Refer to Figure 9.16 in the book, and explain how scan chain improves testability.

Controllability: ability to establish a specific value at each node in a circuit by setting values on the circuit’ PIs.

Observability: ability to determine the value at any node in a circuit by controlling circuit’ PIs and observing POs.
(e) What are the benefits of partitioning a large combinational logic circuit? Give an example to illustrate your answer.

(f) Describe how scan registers work.

(g) Refer to Figure 9.16 in the book, and explain how scan chain improves testability.
8.5 Show that any pseudoexhaustive test set based on a sensitized partitioning of a combinational circuit $N$ detects all detectable SSFs in $N$. 
9.5 Consider IC1 and IC2, where each IC contains boundary scan registers as shown in Figure 9.18(b). Assume the outputs of IC1 drive the inputs to IC2, and the outputs of IC2 drive the inputs to IC1. Show how the scan registers of these ICs can be interconnected. Describe how the interconnections between these ICs can be tested by means of the boundary scan registers. Explain the control sequencing required for the signal T1 as well as the signal $N/T$ that controls the scan registers.
Backup
5.3 For the circuit and the fault set used in Example 5.1, determine the faults detected by the test 11010 by deductive simulation.

Figure 5.10
5.7 Repeat the simulation carried out in Example 5.1 using the concurrent method. (Show the fault lists after simulating each vector.)
5.9 For the circuit in Figure 5.38, determine the faults detected by the test 111 by

(a) concurrent fault simulation (start with a collapsed set of faults)
(b) critical path tracing.

**Figure 5.38**
5.6 Associate with each line $\alpha$ a list $L^1_\alpha$, called the one-list, where fault $f \in L^1_\alpha$ if and only if line $\alpha$ in the circuit under fault $f$ has the value 1. Note that $L^1_\alpha = L_\alpha$ if line $\alpha$ in the fault-free circuit has the value 0, and $L^1_\alpha = \overline{L_\alpha}$ if the line has the value 1. Show that for an AND (OR) gate with inputs $a$ and $b$ and output $c$, $L^1_c = L^1_a \cap L^1_b \cup \{c \ s-a-1\}$ ($L^1_c = L^1_a \cup L^1_b \cup \{c \ s-a-1\}$). What are the major advantages and disadvantages of carrying out fault analysis using $L^1_\alpha$ rather than $L_\alpha$?