Final Exam: Graduate Course – Testing of Ultra Large Scale Integrated Circuits, 16:332:576
Spring 2002

May 15, 2002

Problem 1: Fault Modeling and Simulation (11 Points)

For the circuit of Figure 1, do the following:

(i) Count the total number of single fault sites. (2 points)

(ii) Show that the three single faults, $H$ s-a-1, $J$ s-a-1, and $K$ s-a-1, are equivalent. (4 points)

(iii) Using the parallel fault simulation algorithm (assuming a four-bit machine word), or any alternative fault simulation algorithm, determine which of the three single faults, $F$ s-a-1, $H$ s-a-0, and $L$ s-a-1, are detected by the input vector $A = 1, B = 0$. (5 points)

![Circuit Diagram]

Figure 1: Circuit for fault modeling and simulation problem.
Solution to Problem 1

(1) Total fault sites = \#PI + \#gates + \#fanout_branches = 2 + 4 + 6 = 12

(ii) Three faulty functions are:

\[
\begin{align*}
Z(H \ s - a - 1) &= \overline{A.B} = A + B \\
Z(J \ s - a - 1) &= \overline{A.AB.B} = A.\overline{(A + B)B} \\
&= \overline{A(A + B)} = \overline{A}B = A + B \\
Z(K \ s - a - 1) &= \overline{A.AB.\overline{B}} = A(A + \overline{B}).\overline{B} \\
&= (A + \overline{B})\overline{B} = \overline{A}B = A + B
\end{align*}
\]

Since the faulty functions are identical, the three faults are equivalent.

(iii) Parallel fault simulation using a four-bit computer word is shown in the figure below. The fault-free value of a line is represented by bit 0, the value corresponding to fault \( F \ s-a-1 \) by bit 1, the value corresponding to fault \( H \ s-a-0 \) by bit 2, and the value corresponding to fault \( L \ s-a-1 \) by bit 3.

Simulated value of the output \( Z \) indicates that faults \( H \ s-a-0 \) and \( L \ s-a-1 \) are detected and \( F \ s-a-1 \) is not detected.
**Problem 2: Combinational ATPG (11 Points)**

Consider the circuit under test (CUT) with two outputs, $C$ and $S$, shown in Figure 2. We desire a test that will detect a target fault at any one output but not at both outputs. Figure 2 gives an ATPG model circuit in which the two outputs of the CUT are combined into an exclusive-OR gate (shown shaded) to produce an output $Z$.

![Circuit for combinational ATPG problem.](image)

(i) Show that when a fault is detected at $Z$ in the ATPG model, it must be exclusively detected either at $C$ or at $S$. (4 points)

(ii) Using either the five-valued logic or the nine-valued logic, obtain a test to detect $B$ s-a-1 exclusively at one of the outputs of the CUT. (4 points)

(iii) Can such a test be found for $H$ s-a-0. (3 points)

**Solution to Problem 2**

(i) The following table shows how the fault effect ($D$ or $\overline{D}$) propagates through an exclusive-OR gate.

<table>
<thead>
<tr>
<th>One input</th>
<th>Other input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D$</td>
<td>$D$</td>
<td>0</td>
</tr>
<tr>
<td>$\overline{D}$</td>
<td>$\overline{D}$</td>
<td>0</td>
</tr>
<tr>
<td>$D$</td>
<td>$\overline{D}$</td>
<td>1</td>
</tr>
<tr>
<td>$D$</td>
<td>0</td>
<td>$D$</td>
</tr>
<tr>
<td>$\overline{D}$</td>
<td>0</td>
<td>$D$</td>
</tr>
<tr>
<td>$D$</td>
<td>1</td>
<td>$\overline{D}$</td>
</tr>
<tr>
<td>$\overline{D}$</td>
<td>1</td>
<td>$D$</td>
</tr>
</tbody>
</table>

That is, for a fault to affect the value of $Z$, it should affect the value of $C$, or that of $S$, but not those of both. Therefore, when a test detects a fault at $Z$ the fault must be detected (observable) exclusively at $C$ or $S$. 

*Final Exam Solution: Testing course 16:332:376 – May 15, 2002*
(ii) Using the five-valued logic a test $A = 0$, $B = 0$ is found to detect the fault $B_{s-a-1}$ at $Z$, as shown in the following figure. This test detects the fault at $S$, but does not detect it at $C$. \\

![Circuit diagram](image)

Test generation for $B_{s-a-1}$ using five-valued logic.

(iii) The following figure shows that the fault $H_{s-a-0}$ cannot be detected at $Z$. Only on input vector, $A = B = 1$, can activate the fault but it propagates the fault to both outputs of the CUT. An exclusive test for this fault is not possible. \\

![Circuit diagram](image)

No test can detect $H_{s-a-0}$ at $Z$. 

---

*Final Exam Solution: Testing course 16:332:576 – May 15, 2002*
Problem 3: Testability Measures (11 Points)

For the circuit of Figure 3, compute the combinational and sequential SCOAP testability measures (both controllability and observability, and including the \textit{CLOCK} and \textit{synchronous \texttt{RESET}} signals.)

![Circuit Diagram](image)

Figure 3: Circuit for Problem 3.

Solution to Problem 3

The steps of calculation for SCOAP testability measures are shown in the three figures that follow. Combinational measures are shown as \((CC0, CC1)CO\) and sequential measures as \([SC0, SC1]SO\).

![Circuit Diagram](image)

Circuit of Problem 3: PI and PO initialization and first controllability pass.
Circuit of Problem 3: Converged controllability values.

Circuit of Problem 3: All controllability and observability values.
**Problem 4: Delay Test**

(i) Specify a single input change (SIC) test for the critical path \( \uparrow a - z \) (shown in bold lines) in the circuit of Figure 4(i). Is this a robust test? (5 points)

(ii) The circuit of Figure 4(i) is redesigned in Figure 4(ii) to reduce the delay. Will the SIC test obtained above still test the longest delay path shown in bold lines? If not, what is the minimum modification required in the test? (6 points)

![Original circuit](image1)

(i) Original circuit.

![Redesigned circuit](image2)

(ii) Redesigned circuit.

Figure 4: Circuits for delay test problem.

**Solution to Problem 4**

(i) A SIC test is found by statically sensitizing the path \( a - z \) and applying a rising transition to \( a \). Thus, the test is: \( a = R1, b = S0, c = S1, d = S0, e = S1, f = S0, g = S1, h = S0, i = S1 \). This is a robust test.

(ii) When the above test is applied to the redesigned circuit, the OR gate at the output receives a rising transition (\( R1 \)) at its upper input and a \( S0 \) at the lower input. Although \( R1 \) is produced at \( Z \), this transition does not arrive through the longest path (shown in bold). Thus, the upper path, which is 5-gate long,
is tested but the 6-gate path (shown as critical path) is not tested. We find that no input vector can sensitize this path when \( a = 1 \) and the PDF \( \uparrow a - z \) for the critical path is untestable.

*This part not required: When \( a = 0 \), setting \( h = S1 \) and leaving all other inputs as before, we can sensitize the critical path. Thus, the PDF \( \downarrow a - z \) for the critical path can be tested by applying \( a = F0 \), \( h = S1 \) and leaving all other inputs as before. We also note that the s-a-1 fault at the output of the NOT gate is redundant. If that fault is removed, then the 6-gate critical path will also be removed. Then, the 5-gate path becomes the critical path and it is robustly testable by the test derived in part (i).*

**Problem 5: Memory Test (11 Points)**

Rigorously prove that the MARCH C— test detects all inversion coupling faults \(<\uparrow;\downarrow>\). Indicate the testing time complexity for MARCH C— in terms of \( n \), the number of bits in the memory.

| MARCH C—          | \{ \( \uparrow (w0) \); \( \uparrow (r0, w1) \); \( \uparrow (r1, w0) \); \( \downarrow (r0, w1) \); \( \downarrow (r1, w0) \); \( \uparrow (r0) \) \} |

**Solution to Problem 5**

We rigorously prove that the MARCH C— test detects all inversion coupling faults (CFin).

The MARCH C— test is,

\[
\{ M0 : \uparrow (w0); M1 : \uparrow (r0,w1); M2 : \uparrow (r1,w0); \\
M3 : \downarrow (r0, w1); M4 : \downarrow (r1,w0); M5 : \uparrow (r0) \}
\]

and the inversion coupling faults are \(<\uparrow;\downarrow>\) and \(<\downarrow;\uparrow>\).

**Necessary condition:** For all cells that are coupled, each should be read after series of possible CFins may have occurred, and the number of coupled cell transitions must be odd.

**Fault \(<\uparrow;\downarrow>\): Address of coupled cell \( i \) > address of coupling cell \( j \).** Cell \( j \) initialized to 0 by \( M0 \), \( j \) is made to \( \uparrow \) by \( M1 \), coupled cell \( i \) set to 0 by \( M0 \), unexpected inversion detected by \( M1 \), number of coupled cell inversions = 1.

**Address of coupled cell \( i \) < address of coupling cell \( j \).** Cell \( j \) initialized to 0 by \( M2 \), \( j \) made to \( \uparrow \) by \( M3 \), coupled cell \( i \) set to 0 by \( M2 \), unexpected inversion detected by \( M3 \), number of coupled cell inversions = 1.

**That completes the proof.**

The test complexity is \( O(10n) \).
Problem 6: Analog Test (11 Points)

a. Unit test period. For an analog circuit, the test waveform frequency \( F_t = 2010 \) Hz and the sampling frequency in the DSP ATE is 8000 s/s. Compute the minimum unit test period and the corresponding primitive frequency.

b. Unit test period. A CODEC is to be tested on a DSP ATE with \( F_s = 8000 \) s/s. Originally, \( P = 40 \) ms, but that does not allow \( N \geq 400 \). Select a test waveform frequency as close to 2000 Hz as possible that still generates \( N \geq 400 \) unique samples by adjusting \( \Delta \). How many test waveform cycles \( (M) \) will there be in the primitive period?

Solution to Problem 6

a. 

\[
\frac{F_t}{F_s} = \frac{M}{N} = \frac{2,010 \text{Hz}}{8,000 \text{Hz}} = \frac{201}{800}
\]

That is, \( M = 201 \) and \( N = 800 \). Unit test period is obtained as,

\[
\text{UTP} = \frac{M}{F_t} = \frac{201}{2,010 \text{Hz}} = 0.1 \text{ sec}
\]

Primitive frequency, \( \Delta = \frac{1}{\text{UTP}} = 10\text{Hz} \)

b. 

\[
\Delta = \frac{1}{p} = \frac{1}{40 \text{ msec}} = 25\text{Hz}
\]

\[
N = \frac{F_s}{\Delta} = \frac{8,000 \text{s/s}}{25\text{Hz}} = 320
\]

We must change \( \Delta \) and \( p \) to get \( N = 400 \).

\[
\Delta = \frac{8,000 \text{s/s}}{400} = 20\text{Hz}
\]

If \( F_t = 2,000\text{Hz} \), \( M = \frac{F_t}{\Delta} = \frac{2,000\text{Hz}}{20\text{Hz}} = 100 \), and \( \frac{M}{N} = \frac{100}{400} = \frac{1}{4} \), which gives only 4 samples. So, choose either \( M = 99 \) or \( M = 101 \).

For \( M = 99 \), \( F_t = M \times \Delta = 99 \times 20\text{Hz} = 1,980\text{Hz} \), and

For \( M = 101 \), \( F_t = M \times \Delta = 101 \times 20\text{Hz} = 2,020\text{Hz} \).

In both cases, we get 400 unique samples.
**Problem 7: DFT (11 Points)**

The circuit in Figure 5 is a sequence detector. A sequence 111 in the *INPUT* bit-stream locks the output $Z$ to 1. The state of the circuit can be set to 000 with output $Z = 0$ by applying *CLEAR* = 1.

![Circuit Diagram](image)

Figure 5: Circuit for DFT Problem 7.

(i) Redesign the circuit using minimum extra hardware to conform to the scan design rule, “clock must not be gated by a combinational signal.” Neatly sketch the redesigned circuit. (5 points)

(ii) Sketch the schematic of the full-scan circuit using the multiplexer type of scan flip-flops (SFFs). Show the complete wiring of the *SCANIN*, *SCANOUT* and test control (*TC*) signals. (6 points)

**Solution to Problem 7**

(i) To conform to the “clock rule,” we use the clock gating signal $Z$ to inhibit the data input. This is economically done by inserting an OR gate (shown shaded) in the following schematic. $Z = 0$ allows the normal *INPUT* to be applied to the circuit. $Z = 1$, which occurs only when a 111 stream is detected, forces a constant 1 input.

![Redesigned Circuit](image)

Redesign to avoid clock rule violation.

(ii) The following figure shows the scan wiring. All three D-flip-flops are replaced by scan flip-flops (SFFs) shown as shaded blocks. The logic of SFF, which includes a D-flip-flop and a multiplexer, is shown in the second figure.
Problem 8: BIST (12 Points)

1. For the circuit in Figure 6, please design an external-XOR LFSR pattern generator implementing the characteristic polynomial $1 + x^2 + x^3$ and an Input MUX for testing.

2. Express the linear system of matrix equations describing this pattern generator.

3. Now, assume an exhaustive counter-based pattern generator and the given response compacter for the circuit in Figure 7. Compute the good machine signature for the circuit. The output MISR is initialized to “000” before testing.

4. For the same circuit, compute the bad machine signature for the fault $q$ stuck-at-0. Does the test hardware alias for this fault?
Solution to Problem 8

1. The hardware is shown in Figure 8.

2. 

\[
\begin{bmatrix}
    a \\
    b \\
    c
\end{bmatrix}
(t + 1) =
\begin{bmatrix}
    0 & 1 & 0 \\
    0 & 0 & 1 \\
    1 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
    a \\
    b \\
    c
\end{bmatrix}
(t)
\]
3. The table below contains the fault-free outputs of the circuit and the state of the LFSR after every clock. The initial state of the flip-flop is assumed to be $Q_1Q_2Q_3 = 000$. The output equations used for computing the fault-free outputs in the table are:

$$Y = (A \oplus B) + AB \quad \text{and} \quad Z = \overline{B + C} \oplus Y$$

<table>
<thead>
<tr>
<th>$A$</th>
<th>$B$</th>
<th>$C$</th>
<th>$Y$</th>
<th>$Z$</th>
<th>LFSR state</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$Q_1$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Thus the final signature of the good machine is “1 1 0”.

The logic to detect this signature can be implemented by a NAND gate as evident from the following equation.

$$\overline{GOOD} = Q_1Q_2Q_3$$

4. In the case of the fault $q \text{s-a-0}$, the faulty outputs are:

$$Y_f = Y \quad \text{and} \quad Z_f = Y$$

The table below contains the faulty outputs of the circuit and the state of the LFSR after every clock. The initial state of the flip-flops is assumed to be $Q_1Q_2Q_3 = 000$ as before.

<table>
<thead>
<tr>
<th>$A$</th>
<th>$B$</th>
<th>$C$</th>
<th>$Y$</th>
<th>$Z$</th>
<th>LFSR state</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$Q_1$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Thus, the final signature of the faulty circuit will be “0 1 0”, and the test hardware does not alias.
Problem 9: Boundary Scan (11 Points)

Figure 9 shows two boundary scan cells surrounding the on-chip system logic. We test the path from the INPUT boundary scan cell, through the on-chip system logic, and ending at the OUTPUT boundary scan cell. The JTAG commands are: SAMPLE, PRELOAD, EXT TEST, INTEST, RUNBIST, CLAMP, ID CODE, USER CODE, HIGHZ, and BYPASS. Please explain the sequence of these commands used for delay fault testing of this particular path.

![Diagram of Circuit]

Figure 9: Circuit for Problem 9.

Solution to Problem 9

Here is the sequence of JTAG commands used to test the path:

1. SAMPLE – Capture pin signals and functional hardware outputs in the Boundary Scan Register.

2. Apply INTEST – Copy Boundary Scan Register contents into hold latch. Apply as many TCK pulses as are necessary to shift the 1st test pattern in from TDI (this will be the number of pins from TDI up to and including this system input pin.) Then apply a functional system clock.

3. Repeat Step 2 to apply the 1st time frame pattern to the path and shift the 2nd time frame pattern into the Boundary Scan Register (requires many TCK pulses.) At this point, the 1st time frame pattern is applied from the hold registers and the second time frame pattern is in the Boundary Scan Register.

4. Repeat the INTEST instruction and immediately apply a functional system clock. This applies the second time frame pattern to the path-under-test, clocks the functional circuit, and captures its response in the Boundary Scan Register. Then, apply as many TCK pulses as are necessary to shift out the response to the 2nd pattern out through TDO (while simultaneously shifting...
in the next test pattern.) The shifted out response will indicate where a timing fault has occurred.

5. Go back to Step 2 if more paths are to be tested.

This sequence can be preceded by IDCODE and USERCODE instructions, in order to identify which chips, and which part programmings, are in the system. Other chips in the system may be given BYPASS instructions to speed up scan chain shifting by shortening the scan chain.