UNIVERSITY OF ALBERTA

DEPT. OF ELECTRICAL AND COMPUTER ENGINEERING

ECE 512 – Digital System Testing and Design for Testability

Final Examination Solutions

Instructors: B. F. Cockburn
Exam date: December 7, 2006
Exam duration: 60 minutes
Aids permitted: Hardcopies of the course overheads.
Copies of the assignment model solutions.

Instructions:
1. Fill out your printed name, signature and I.D. number on this page.
2. Verify that this booklet contains 8 pages.
3. Neatly enter your answers in the spaces provided.
4. Use the reverse sides of the pages for rough work.

Student name: ___________________________________

Signature: _________________________________

Student I.D.: ______________________________

Question	Time	Worth	Mark	Subject
1.	12	20		MATS+ Memory Test
2.	12	20		Delay Fault Testing
3.	12	20		Scan Chains
4.	12	20		BIST Trade-offs
5.	12	20		LFSRs for Test Pattern Generation
Total	**60 mins**	**100**		--

Question #1 (MATS+ Memory Test)

The MATS+ memory test is specified by the following sequence of march elements:

$$\Uparrow\Downarrow(w0);\ \Uparrow(r0, w1);\ \Downarrow(r1, w0);$$

(a) Present a brief argument that demonstrates that all cell stuck-at faults and cell transition faults are detected by MATS+.

The test contains three march elements. All stuck-at-0 cells will be detected by the r1 operations in the third element. Similarly, all stuck-at-1 faults will be detected by the r0 operations in the second element.

All 0-to-1 transition faults are detected because the second element first verifies the presence of a 0 and then writes a 1 into every cell; the third element contains a r1 operation that verifies that every cell did indeed change to a 1.

All 1-to-0 transition faults are not guaranteed to be detected. Cells that start out containing a 1 will be tested for these transition faults because the first element will produce the required 1-to-0 transition, and the r0 in the second element will detect the 1-to-0 transition fault. However, cells that happen to power up containing a 0 already will not be tested since there is not guaranteed to be a 1-to-0 transition in the first element.

(b) Give an example of an idempotent coupling fault that is successfully detected by MATS+. Give numerical addresses so that the address order is clear between the aggressor cell and the victim cell. Briefly indicate which march elements are responsible for triggering the appearance of the error, then later detecting the cell error.

Consider an idempotent coupling fault in which a 0-to-1 transition in the aggressor cell i causes a 0-to-1 transition in a victim cell j, where the address of j appears *after* the address of i in the "up" address direction. (So, i could have value 4 and j could have value 8.) The first element ensures that the 0-to-1 transition is written in cell i; just before this transition, cell j should contain a 0 as a result of the w0 operation in the first element. The coupling fault is detected later by the r0 to cell j in the second element.

Question #1 (MATS+ Memory Test, cont'd)

(c) Now give an example of an idempotent coupling fault that is *not* detected by MATS+. Once again, give the numerical address of the aggressor cell and victim cell in your example fault. Briefly explain why MATS+ fails to detect the fault. Suggest a way in which MATS+ could be enhanced to be able to detect the fault.

```
Consider an idempotent coupling fault in which a 0-to-1
transition in the aggressor cell i causes a 0-to-1
transition in a victim cell j, where the address of j
appears *before* the address of i in the "up" address
direction.  (So, i could have value 8 and j could have
value 4.) The first element ensures that the 0-to-1
transition is written in cell i; however, just before
this transition, cell j should already contain a 1 as a
result of the w1 operation to cell j in the first
element.  Thus there will be no difference between a
good memory and a bad memory.  Therefore the coupling
fault cannot be detected later by the r0 to cell j in
the second element or by the r1 in the third element.
```

Question #2 (Delay Fault Testing)

(a) Briefly define what is meant by a "path delay fault" and a "transition fault". Briefly comment on the relative difficulty of testing a combinational circuit for all possible path delay faults and all possible transition faults.

A path delay fault is a fault in which the propagation delay along a signal path (which may pass through several gates and along several segments of inter-connect) is longer than expected. The extra delay could cause errors in the logical values produced at the output of a block of combinational logic. The extra delay is not necessarily lumped on any one wire segment or at any one gate; it can be distributed over many circuit elements.

A transition fault is a fault in which the output transition for one particular gate is overly slow-to-rise (from 0 to 1) or overly slow-to-fall (from 1 to 0). The fault can be viewed as being lumped at the one gate; not other circuit elements (wires or gates) are affected by the fault. The transition fault delay must be longer than the delay between subsequent circuit input vectors; otherwise, the fault would not be large enough to be observable at the circuit block outputs. A stuck-at fault affecting a gate output can be viewed as a very slow transition fault.

(b) Briefly explain what is meant by a *non-robust* path delay test and a *robust* path delay test. What is the main advantage of using robust path delay tests instead of non-robust path delay tests?

A robust path delay test is a test that detects the presence of a delay fault, regardless of the presence of any other delay faults in the circuit. Thus the delay fault being tested is in no danger of escaping detection if there happed to be other unusual delays present.

A non-robust path delay test is a test that is guaranteed to detect one delay fault in a circuit only if there are no other delay faults present. (It is possible for a non-robust path delay test to miss a fault as a result of other delays in the circuit. Unfortunately, there exist delay faults for which robust tests do not exist, and so we must use non-robust tests instead).

Question #3 (Scan Chains)

(a) A scan chain is sometimes described as an effective way of converting a sequential circuit into a combinational circuit for the purposes of testing. Briefly justify this interpretation.

The scan chain allows the tester to examine the signals at all flip-flop inputs, which are the internal Primary Outputs of the combinational blocks (the external Primary Outputs of the circuit are already directly observable). Also, using the scan chain, arbitrary values can be applied to the flip-flop outputs, which become Primary Inputs to the combinational blocks (along with the externally accessible Primary Inputs). So we can generate the test vectors for the comb. block as if we can completely control all PIs and observe all POs.

(b) What are the major trade-offs when one considers increasing the number of parallel scan chains in a sequential circuit. What would be the advantages and disadvantages of increasing the number of scan chains?

Increasing the number of parallel scan chains directly reduces the time required to load and unload them serially. This reduces the test time and the test costs. But adding more scan chains implies a little extra cost for the multiplexers that will be required to change the extra input pins into extra serial test inputs. Also, there may be additional cost in the test equipment since more tester pin channels will need to be able to supply serial test data, or to store it. Finally, there may be some additional design cost to ensure that all of the scan chains are roughly (or exactly) the same length.

Question #4 (BIST Trade-offs)

Built-in self-test (BIST) has gained much greater popularity in recent years, and its importance is expected to increase in future years. What are the major advantages of BIST, the advantages that you would cite if you were to attempting to convince your superiors to spend extra silicon area to include BIST in a proposed integrated circuit? What would be the major potential disadvantages of including BIST? How would you go about showing that the advantages of BIST outweigh the disadvantages?

The main advantages of BIST are as follows:

(a) The BIST circuitry can test the design at the design's full speed, limited only by the process technology. The test equipment, which is made from older technology than the circuit under test, does not have to operate as fast. Storage is not required for the test vectors. The test can be allowed to run for a very long time, without test vector storage constraints.

(b) The BIST circuit can be used again and again at the different stages of manufacture, system assembly, and system maintenance at customer premises. The BIST circuit is especially useful to allow field maintenance personnel to diagnose the location of a problem down to a "field-replaceable" unit.

(c) The BIST circuits on multiple devices in a system can be executed in parallel, thus reducing the overall system test time.

The major potential disadvantages of BIST are:

(a) Reduced fault coverage, since pseudorandom test vectors are applied instead of test vectors that target particular faults.

(b) Extra manufacturing cost because the BIST circuitry increases the silicon area of each chip, and thus reduces the manufacturing yield.

(c) The BIST circuit requires engineering time to design and to verify, and this adds to the total cost of the circuit.

The best way of justifying BIST would be to use economic arguments that compare (1) the costs of adding BIST to (b) the benefits from increased testing (such as increased product quality and reduced need for expensive test equipment) and fault diagnosability that would result from having BIST.

Question #5 (LFSRs for Test Pattern Generation)

(a) Compute the first six patterns produced by a "standard" (external XOR) LFSR with the characteristic polynomial $f(x) = x^3 + x + 1$ starting in state "010". Draw a circuit diagram to show the structure of the LFSR.

```
The circuit contains three flip-flops in a row (which we
can label A, B and C going from left to right).  The
four wires connecting the flip-flips are labeled x^3,
x^2, x^1 = x, and x^0 = 1 (going from left to right).
There is one external XOR gate, whose inputs are taken
from the x^1 = x and x^0 = 1 wires.  The output this XOR
gate drives the x^3 wire.

The next state equations for the bits of the LFSR are as
follows:  A+ = XOR(B,C), B+ = A, C+ = B

The first six patterns produced by the LFSR starting
from state 010 are just the six states of the LFSR,
which can be computed using a state transition table as
follows;

A  B  C     A+ B+ C+
=======     ========
0  1  0     1  0  1
1  0  1     1  1  0
1  1  0     1  1  1
1  1  1     0  1  1
0  1  1     0  0  1
0  0  1     1  0  0
1  0  0     0  1  0
```

Question #5 (LFSRs for Test Pattern Generation, cont'd)

(b) Compute the first six patterns produced by a "modular" (internal XOR) LFSR with the characteristic polynomial $f(x) = x^5 + x^2 + 1$ starting in state "11010". Draw a circuit diagram to show the structure of the LFSR.

```
The circuit contains four flip-flops in a row (which we
can label A, B, C and D going from left to right).  The
four wires connecting the flip-flips are labeled x^0 =
1, x^1 = x, x^2, x^3, x^4 and x^5 (going from left to
right).  The x^2 term in the polynomial indicates that
there is an internal XOR gate between flip-flops B and
C.

The next state equations for the bits of the LFSR are as
follows:  A+ = E, B+ = A, C+ = XOR(B,E), D+ = C, E+ = D

The first six patterns produced by the LFSR starting
from state 11010 are just the six states of the LFSR,
which can be computed using a state transition table as
follows;

A   B   C   D   E     A+  B+  C+  D+  E+
=============         ==============
1   1   0   1   0     0   1   1   0   1
0   1   1   0   1     1   0   0   1   0
1   0   0   1   0     0   1   0   0   1
0   1   0   0   1     1   0   0   0   0
1   0   0   0   0     0   1   0   0   0
0   1   0   0   0     0   0   1   0   0
```