UNIVERSITY OF ALBERTA

DEPT. OF ELECTRICAL AND COMPUTER ENGINEERING

ECE 512 – Digital System Testing and Design for Testability

Midterm Examination

Instructors: B. F. Cockburn
Exam date: October 31, 2008
Exam duration: 50 minutes
Aids permitted: Hardcopies of the course overheads.
Electronic calculators are permitted.

Instructions:

1. Fill out your printed name, signature and I.D. number on this page.
2. Verify that this booklet contains 6 pages.
3. Neatly enter your answers in the spaces provided.
4. Use the reverse sides of the pages for rough work.

Student name: ______Model Solutions__________

Signature: ___________________________________

Student I.D.: _______________________________

Question	Time	Worth	Mark	Subject
1.	10	20		Test Vectors
2.	10	20		Fault Modeling
3.	10	20		Fault Simulation Methods
4.	10	20		ATPG Algorithms
5.	10	20		Sequential Circuit Testing
Total	**50 mins**	**100**		--

Question #1 (Test Vectors)

(a) Briefly describe the similarities and differences between (1) the test vectors that are used by integrated circuit designers for design verification and (2) the test vectors that are used by test engineers for production testing.

Design verification vectors are used to find design errors. They target errors in the design (e.g., incorrect assumptions, bad/incomplete initialization, rare cases not correctly handled, some cases not anticipated, errors in extreme cases that are still within specification, etc). Test vectors are used for finding manufacturing defects in fabricated chips. The design of the chip should be nearly perfect, but errors may be introduced in manufacturing (e.g. shorts, stuck lines, open transistors, etc.).

(b) What would be the advantages and disadvantages of using design verification vectors for production testing?

DV vectors may be readily available from the designers, which is an advantage. They should verify the basic functionality of the design. But DV vectors are not likely to have very good fault coverage because they were not designed to target faults (which model manufacturing defects). This is the disadvantage of using DV vectors in production testing: their fault coverage is unlikely to be adequate.

(c) Production test vectors can be generated pseudorandomly or deterministically. Why is it common to use test vector sets that include both pseudorandom and deterministic test vectors?

Pseudorandom vectors are easy to generate, and many easy-to-detect faults will actually be detected by them. The remaining faults are harder to detect: tests must be developed for those harder-to-detect faults using an automatic test pattern generation programs like PODEM or the D algorithm. As a final step, one might fault simulates the ATPG vectors first to see which easy faults are detected by them. This may allow some of the pseudorandom vectors to be omitted if they don't detect any faults not already detected by other test vectors. By doing this so-called "reverse fault simulation" you often obtain a more compact test set.

Question #2 (Fault Modeling)

(a) The stuck-at fault model is still the most commonly used fault model in industry, but test engineers have at times found it useful to introduce transistor-oriented faults models, such as the stuck-on and the stuck-open. Briefly define these two transistor fault models, and then (using a CMOS inverter as an example) explain what additional testing challenges are caused by transistor faults.

A transistor stuck-open fault behaves as if there is no connection between the source and drain terminals. It's as if the transistor were entirely missing. A transistor stuck-on fault behaves as if there is a low-resistance permanent connection from the source to drain terminals.

In an inverter, a stuck open fault will cause the output node to be either (1) driven to the one supply that has a good transistor, or (2) left floating in a high impedance state with the functioning transistor in the off state. A stuck on fault will cause either (1) the gate output to be driven to the supply (Vdd or Vss) through the stuck on transistor, or (2) a short circuit path will be set up between Vdd and Vss through the two transistors in series. This last case leads to a possibly illegal output signal midway between Vss and Vdd. It is hard to predict the logical behaviour resulting from stuck on faults since it is difficult to predict how intermediate signals will be interpreted downstream. Stuck open faults introduce memory (floating nodes) into combinational logic, and two-vector tests are usually required to detect them.

(b) What is meant when it is stated that fault models F_1 and F_2 are *equivalent*, and that fault model F_3 *dominates* fault model F_4?

Equivalent faults are indistinguishable with respect to their input-output behaviour (e.g. their truth table). Any test that detects F1 will also detect F2, and vice versa.

If fault F3 dominates fault F4, then any test that detects F4 also detects F3. The opposite statement is not true. It is not necessarily true that a test that detects F3 also detects F4.

Question #3 (Fault Simulation)

Briefly describe the techniques of (1) parallel fault simulation and (2) concurrent fault simulation. Be sure to clearly explain how each attempts to gain efficiencies by exploiting parallelism in the fault simulation calculation.

Parallel fault simulation exploits the multiple bits in each computer word, and the presence of bitwise logical operations (e.g. AND, OR, XOR, NOT). Different bit positions can be used to simulator different variants of the circuit. Each variant could be either a copy of the good circuit, or the circuit with one fault present. So in a 32 bit machine, the good circuit could be simulated with 31 different faults at the same time. This method exploits the parallelism in the simulator machine. It also exploits the fact that the different faulty circuits are, for most of the gates, the same.

In concurrent fault simulation, a single simulation of the good circuit is augmented with replicas of any gates that have either (a) different gate behaviour for the good gate, or (b) have different signals on at least one of the gate terminals. Work is spent on the faulty circuit simulations only where the faulty circuit differs from the good circuit. "Replicas" of the gates are added or deleted as necessary to keep track of the different behaviour of the faulty circuits. We are exploiting the parallelism that the good and faulty circuits are, in most gate positions, the same. We only spend work on the faulty circuit simulations where the signals of the faulty circuit differ from those of the good circuit.

Question #4 (ATPG Algorithms)

The D Algorithm and PODEM are two influential test pattern generation algorithms. PODEM is an improvement on the D Algorithm for most circuits because it is more efficient at searching the space of possible test vectors for a fault. Briefly describe the search strategies of the two algorithms, and then explain where the improved efficiency of PDEM comes from. Why is it that most of the recent test pattern generation algorithms are based on PODEM rather than the D Algorithm?

PODEM gains efficiency by performing a more systematic search of the space of possibly test vectors. The number of variable assignments is restricted to the primary inputs (the D algorithm makes many more choices in its progress, choices between primitive D-cubes, propagations cubes, input variables, etc). PODEM has the structure of a branch-and-bound algorithm, where sections of the search space (i.e. the space of input vectors) can be systematically pruned away if it is found that no test vector can exist in those sections. The D algorithm, because it works from the inside of the circuit to the outside (inputs and outputs), is less systematic in its searching, and the same input combinations can be encountered again and again.

Recent algorithms build on PODEM because they too want to use the branch-and-bound strategy. Heuristics added to PODEM, such as the headline heuristic and multiple path sensitization used in FAN, will only improve on PODEM.

Question #5 (Sequential Circuit Testing)

Briefly explain what is meant by time frame expansion in the context of sequential circuit automatic test pattern generation. How does time frame expansion facilitate the development of algorithms for generating test vectors for sequential circuits? Give a high-level algorithm that shows how time frame expansion would be used.

Time frame expansion is the technique of transforming a sequential circuit (operating over several clock cycles in time) into an equivalent combinational circuit repeated in several concatenated instances in space. The clocked registers are replaced by wires joining the flip-flop outputs of one spatial instance with the flip-flop inputs of the next spatial instance. The primary inputs of the original circuit (which have signals that change in time) are replaced by multiple sets of primary inputs (one to each spatial instance). In effect, the original sequential circuit is converted into a combinational circuit.

Once converted into a combinational circuit in this way, modified combinational test pattern generation algorithms (like PODEM) can be used to find test vectors. If no test vector is found for N instances of the circuit, then the problem can be repeated with N+1 instances. The value of N would start at 1, and then would be increased as necessary until a test vector sequence is found (or a vector count limit is reached). Care must be taken to ensure that one fault in the original sequential circuit appears in each spatial instance of the time frame expanded circuit.