Department of Computer Engineering
COE 571
Testing of Digital Circuits

Course Project – Term 072
Due date: June 4, 2008

Suppose that you are a test engineer working at a company and that you are requested to evaluate different testing methodologies and provide recommendations for the company on the testing strategy to be used. You need to clearly justify your decision by supporting arguments and results.
In order to do this study, you are going to perform your analysis on circuit **s5378.bench**. This circuit has **35 inputs, 49 outputs and 179 D-type flip-flops**. There are four testing methodologies that you are going to evaluate: **No-scan methodology**, **full-scan methodology**, **partial-scan methodology** and **built-in self test methodology.** When you evaluate these schemes, determine the testing cost of each method. This includes the **test application time** which determines the number of clock cycles required to test the circuit using each **methodology**. Also, it includes the test generation cost process and the impact on area and delay.
The company requires that the **fault coverage achieved must be at least 99%**. For each evaluated test methodology, **minimize the number of test vectors** needed as much as possible using **test compaction**. The details of the steps that are to be performed in the evaluation of each step are detailed below.

1. **No Scan Methodology**

   In this methodology, no modifications will be made to the circuit. Use a combination of Random ATPG and the sequential ATPG HITEC to maximize the fault coverage. After you obtain your test sequence, compact it using the test sequence compaction tool that will be given to you.

2. **Full Scan Methodology**

   In this testing methodology, all the flip-flops in the circuit are fully scanned (i.e. stitched into a shift register in test mode). This allows full controllability and full observability to the flip-flops. This makes the circuit like a combinational circuit and a combinational ATPG can be then used to test it. To evaluate this methodology, do the following:
   - Convert the circuit into full scan by replacing all flip-flops in the **.bench* file to primary inputs and primary outputs. You can do this using the OPUS tool with the command **fullscan s5378**. The created circuit will be saved in the file s5378f.bench.
   - Then, use HITEC to generate test patterns for the circuit s5378f.bench.
   - Compact the generated test patterns using the test compaction tool that will be given to you.
   - Next, convert the circuit s5378 by inserting a scan chain. This can be done by placing all the FF numbers to be scanned in the file s5378.scan. Put the FF numbers in the same order as given in the file s5378.name. Then, run the

command **addmux –scan s5378**. This will generate to you the file s5378_scan.bench.

- Run HITEC on the scanned circuit s5378_scan.bench and report the fault coverage.
- Reformat the compacted test set that you obtained for the circuit s5378f in the format applicable for the circuit s5378_scan.bench. Fault simulate the formatted test and report the fault coverage. Comment on the difference between the fault coverage you obtained in this step and by using HITEC.
- One way to improve the test application time is to use multiple scan chains. Based on the compacted test set you have, make the appropriate analysis to determine the number of scan chains needed to make the test application time of this methodology comparable to that of the No-scan methodology and the partial scan methodology. Make the analysis for each case separately.

3. Partial Scan Methodology

In the partial scan methodology, only a subset of the FFs is scanned. Using opus tool, explore with the different options given by the tool to determine the minimum number of FFs required to be scanned to achieve the desired fault coverage. Clearly illustrate the strategy you used to achieve your solution and show all your results. Based on your experimentation, suggest a partial scan strategy and justify your strategy. Then, run HITEC on the partial scan design and demonstrate that you achieved the desired fault coverage. Compact the generated test sequence using test compaction tool and compute the test application time. Compare the test application time obtained with other testing strategies.

4. Built-In Self Test (BIST) Methodology

In this strategy, the circuit will be designed to test itself by generating test patterns based on on-chip pseudo random generators (based on LFSRs) and on-chip test response compactors (based on MISRs). Based on the different BIST techniques studied in class, pick the one that you think is more suitable for your design and adopt it. Justify your choice. Then, add the required design-for-testability to make your circuit BIST-Ready. You do not need to insert LFSRs and MISRs in the circuit. Generate the patterns that will be generated by LFSR by emulating the LFSR and store the patterns in a file. Fault simulate the generated patterns and determine the fault coverage that will be obtained. Assume that the effect of masking is negligible and what you obtain from the fault simulator is the actual fault coverage obtained. Based on your experiments, determine the suggested number of patterns that will be generated by LFSR.
Determine the remaining undetected faults which are not detected by the LFSR. Use HITEC to generate test sets for these faults. Compact the generated test set. Determine the achieved fault coverage based on the LFSR patterns and the deterministic patterns. Compute the test application time of the combined patterns.

Write a professional report showing all the details of your analysis, evaluation, and recommendations. You will be asked to present your findings in a 10-minute presentation.