Carnegie Mellon University
Department of Electrical and Computer Engineering

18-765: Digital Systems Testing and Testable Design

Fall 2023

Project 5: Design for Testability (DFT)

Start Date: **November 13, 2023** End Date: **December 8, 2023**

Total Points: 100 + 30 (extra credit)

---

1. Background

In this project, you take the role of a DFT engineer. You are given a sequential circuit and must make it "testable" using various DFT techniques that may include scan (lecture 21 and chapter 14), BIST (lecture 23 and chapter 15), test/control points (lecture 21 and chapter 15), and 1149.1 (*i.e.,* boundary scan - see lecture 22 notes and chapter 16 of the textbook). In addition, you are encouraged to explore and research other techniques from textbooks or papers.
This project will be completed in teams of two, and project deliverables will include your design, your test patterns, a written report, and an 8 minute presentation.

2. Project Details

The circuit you'll be working with is the ISACS89 benchmark circuit s9234. It has 5597 gates, 36 inputs, 39 outputs, and 211 D-type flip-flops. It is entirely up to you to decide what DFT should be added to the existing circuit. The goal of this project is to achieve maximum SSL fault coverage with minimum overall cost.

There are three main objectives for this project, revolving around the key objectives and requirements for general DFT implementation:

1. **Test Quality:** The DFT and test patterns you provide should produce a high level of test quality for the design. This is quantified through your achieved stuck-at fault coverage of the signals in the original circuit, and measured by performing fault simulation of your provided test sequence on the provided design for a targeted fault list.
2. **Test Cost**: At its core, DFT and test is economics: how does it cost for you to achieve high quality? For this project, we measure test cost through a combined metric which represents DFT area overhead, test time, and tester complexity.
3. **Functional Correctness:** Test-specific logic must change the behavior of the original circuit in its intended functional (non-test) mode. To ensure this, we will simulate your circuit against the original circuit functionality, providing various inputs to PIs and confirming proper outputs from POs. To facilitate this, you will provide instructions for configuring the circuit in the (ex: `test_enable=0`). **In the functional mode, your modified version of s9234 should have the same I/O behavior as the original circuit.**

3. Cost Metric

As we've discussed over the course of the semester, both in course materials and in guest lectures, there are multiple factors that contribute to the overall cost of test. In this project, we represent the below three factors within a combined metric that will be used to quantify the cost of your solution.

1. **Test Time:** The Automated Test Equipment is expensive, and every second we take to test our chip counts toward the bottom line. In this project, test time is measured through the number of clock cycles needed to apply your entire test sequence multiplied by the number of tester channels used. The number of standard clock cycles to apply the provided tests will be counted, and the number of BIST clock cycles will be divided by a factor of 1,000.
2. **DFT Circuit Overhead:** Adding extra circuitry to an existing design brings significant cost. Increasing overall chip area results in fewer chips to fit on a wafer, and increases the likelihood of a given chip being defective. For this project, we measure circuit overhead by the number of gates added, which is divided into newly-added combinational logic gates and enhancements to existing sequential gates. To measure circuit overhead for extra combinational logic, we will count the number of "gate inputs" you add to the design. For example, if you add 300 2-input gates, it means your gate overhead is 600. To measure circuit overhead for sequential gates, two types of flip-flops can be used in your design. The primitives for these flip-flops are included in primitives.v in the provided files. The first primitive, `dff`, is a simple D flip-flop without a reset pin. The second, `dff_r`, includes the reset pin. As is shown in the following section, each use of `dff_r` within your circuit will cost more than `dff`. Each non-reset flip-flop you add to the design will be equated to a 10-input gate and each reset flip-flop you add to the design will be equated to a 20-input gate.
3. **Tester Complexity:** The more complex your chip is to connect to, the more complex the tester must be to test it. If your tester is required to control/observe many inputs and outputs, then your tester channel cost will increase. The number of tester channels is tabulated by the number of pins that are controlled by the tester during your test sequence.

If you achieve a high fault coverage without corrupting the normal functionality of s9234 AND your cost *C* is reasonable, you will receive full credit for the technical portion of the project (*i.e.,* 80pts out of 100pts). The overhead cost *C* of your design will be computed using the following equation:

$$C = \left\{\left(\left[\frac{B}{1000}\right] + S \times \frac{T_S}{4}\right)\right\} + (P \times 1000) + (T \times 100) + \left[D_N + \left(2 \times D_R\right)\right] + \frac{G}{10}$$

- *B* is the number of clock cycles for executing any built-in self test
- *S* is the total number of clock cycles to perform test reduced by *B* (*i.e., S=Q-B,* where *Q* is the total number of clock cycles simulated)
- T_S is the number of active tester channels in the test session
- *P*, D_N, D_R, and *G* is the number of extra chip input and outputs, normal flip-flops, reset flip-flops, and gate inputs, respectively, added to the design
- *T* is the number of tester channels. In other words, *T* is the number of inputs and outputs where values are applied or observed in your file of test vectors.

Consider the following example data to illustrate the cost metric. Specifically, let's say your design has the characteristics shown in the table below. The cost *C* of your design will be as shown.

Example design characteristics and associated cost.

Design characteristic	Cost
BIST that operates for 10,000 cycles. **B = 10,000**	10
Total number of cycles that does not include the BIST cycles is 4,000. This means that your test sequence test file would contain 14,000 vectors (i.e., tests). **S = 4,000 and Q = 14,000** Furthermore, seven tester channels are used during the 4,000 test vectors. T_S**=7**	4,000 × 7/4=7000
Suppose four additional PIs and two additional POs are used for test. **P = 6**	6,000
The BIST, boundary scan, internal scan, etc. that you added to the design consisted of 300 3-input gates, 200 2-input gates, and 50 flip flops. **G = 1300,** D_N **= 50,** D_R **= 10.**	130 + 50 + 20 = 200
Suppose one of the original PIs for internal scan. The number of tester channels is then added PIs + added P0s + shared PIs. **T=7**	700
Total cost	13,910

4. Boundary Scan Implementation

For 10 points of extra credit, you will have the opportunity to implement boundary scan circuitry following the below criteria:

1. You must implement the 16-state TAP controller exactly as specified in the lecture notes and the textbook. Test data is to be shifted into and out of s9234 under the control of the TAP controller only. This simply means the TAP must initiate and terminate any test operations within your design.
2. You must implement a 75-bit boundary scan register as specified in the lecture notes and the textbook. The length is dictated by the 36 inputs and 39 outputs of s9234.
3. You must implement a 1-bit bypass register.
4. You must implement an instruction register as specified in the lecture notes. The size of the instruction register depends on what other DFT features, if any, you decide to implement. If, for example, you decide to implement internal scan, then a 2-bit instruction register will suffice since you will need one instruction for using the boundary scan register, one instruction for the bypass register, and one instruction for your internal scan register.

5. Project Tooling

Siemens Tessent
Tessent is the most widely used ATPG and Diagnosis tool in the industry. It has a wide array of capabilities, but here we'll specifically use its combinational circuit ATPG engine to create patterns that can be translated to match the DFT of your circuit.
Test patterns read/written by Tessent will use ASCII format, which is a pattern-based format fully describing the input and output values applied to a combinational circuit.

Siemens KaleidoScope
While standard ATPG tools are capable of simulating scan patterns, we won't be using that capability in this project. KaleidoScope is a functional fault simulator, capable of taking a cycle-level definition of a test on each input/output pin, then grading that test against a list of faults. In this project, you will create the cycle-based tests for KaleidoScope. KaleidoScope will be used to apply this test to your design and determine how many of the faults in the original sequential circuit are successfully detected.
Tests will be read by KaleidoScope using the Value Change Dump (VCD) format, which is a cycle-based format defining the values of each input and output pin.

Note: *As of 11/13, we are waiting for license help from Siemens to enable KaleidoScope. We expect this to be resolved in the next few days, and update when this has been resolved.*

Verilog Simulator
Although Siemens tools are capable of simulating test vectors for your circuit, it does so primarily in performing fault simulation and calculating fault coverage. You will likely wish to use an additional Verilog simulation tool when building and debugging the different parts of your circuit. This will prove to be a much faster and easier way to ensure that your circuit is properly working. Once you have confirmed correct behavior, you can then move to Siemens KaleidoScope to perform fault simulation.

Accompanying Scripts
As in real-world applications, there are multiple tools used in this project. When developing a full solution like the one in this project, you may need to develop other functions/scripts to perform tasks that translate, manipulate, or massage input and output files. This is an expected component of completing this project.
For example, while you are provided a KaleidoScope-compatible fault list of the original faults in the circuit, these pin names will likely change as you modify the circuit. You will need to modify the pin names in your eventual submitted KaleidoScope fault list to ensure all faults are found in your design.
Additionally, the cycle-based VCD file format used by KaleidoScope may not be easy to create manually. Developing automation to write your desired VCD file will likely be an important piece of this project.

To aid in your work, we have provided a sample set of scripts. These scripts and their usage are detailed further in the *documents/* subdirectory of the Project 5 directory.
Also included is the tutorial *project5_s27_tutorial.pdf*, which is designed to show the general process for using Tessent, KaleidoScope, and the aforementioned scripts. This tutorial uses a scan-based DFT design, which may be different from the implementation you choose to implement.

Note: *The provided scripts are provided purely as a potential assistance – they are not intended to enable all possible DFT approaches/configurations. If the scripts don't work for your particular use case, you may need to modify them or develop new scripts to perform the desired task.*

Important: Siemens tools are not capable of simulating procedural Verilog (i.e. `always` blocks). It is capable of understanding the module hierarchy, but at the lowest level your modules must consist only of the primitives present in the Verilog standard and those provided to you. This includes the standard gates (`or`, `and`, `not`, etc.) as well as the modules provided in `primitives.v`, namely, `dff_r` and `dff`.

6. Deliverables

Report: You must generate a typewritten report (four pages maximum) that describes your project. Specifically, describe what you did, why you did it, and detailed results about the achieved level of fault coverage and the incurred costs. Tell us which of the DFT strategies you chose and how they were implemented. We are interested in hearing about all the DFT features you investigated. Be sure to include in your report the operation requirements for each test mode of operation.

Final Report is due December 8, 2023.

Testing files: You must hand-in one VCD file that we will use to fault simulate the circuit. We will use this file in Siemens KaleidoScope to confirm your reported fault coverage. Note that if you are not using any of the 36 PIs of s9234 as part of your tester channels, you must set them to "X" (*i.e.*, don't care) during the entire test session. Similarly, any unused outputs should be set to "X".

Design-for-testability Verilog circuit: All of the Verilog files containing the modules that make up your circuit must be included.

Project presentation: You must present your project to the rest of the class. We will provide you with a PowerPoint template for creating the presentation.

Presentations will take place during the live session on December 8, 2023.

Submission Format: All pieces of your submission should be compiled in a *zip* file. Within the *zip* file there should be a subdirectory entitled sourcefiles. This directory will hold all of the files required for creating and testing your design (Verilog files, pin assignment file, test files). Only your report and presentation slides should be outside of this subdirectory. The *zip* file should be submitted to Gradescope.

7. Grading

- Report 15 points
- Presentation 5 points
- Test Quality 30 points
- Test Cost 30 points
- Normal Functionality 20 points
- Credit 30 points

Test Quality: 30 points are awarded for the test quality of your design and test set, quantified by your achieved stuck-at fault coverage. While the final grade will be flexible based on the fault coverages achieved by the course, you can expect full credit if you achieve a target of 93.5%.

Test Cost: 30 points are awarded for the test cost achieved by your design and test set, quantified by the metric described above. While the final grade will be flexible based on the costs achieved by the course, you can expect full credit if your cost is under 20,000.

Test/Normal Functionality: Up to 20 points will be deducted for incorrect normal, bypass, or boundary-scan behavior.

Extra Credit:

1. The design that achieves the highest fault coverage will receive 10 points of extra credit.
2. The design that has the lowest cost among those submitted while achieving 93.5% fault coverage will receive 10 points of extra credit.
3. Adding a Boundary Scan implementation and corresponding test vector will earn 10 points of extra credit.

8. Project 5 Checklist

- 4-page report that includes all test-mode op-codes and the calculation for your design's total cost
- PowerPoint presentation emailed to the instructor by 12pm on the day of presentations
- All Verilog files included in your final DFT circuit design.
- Necessary KaleidoScope files and command required to validate your reported fault coverage
- README file along with the submission that details
  - Instructions for applying the provided test to your circuit
  - Instructions for operating the circuit in functional (non-test) mode to enable functional validation
- *zip* file containing all deliverables, with sourcefiles subdirectory and proper file organization

9. Project Checkpoints

To ensure you have the support you need on this project, we plan to hold brief weekly check-ins with each group. In Week 1, we ask for a written update submitted via email. From Week 2 onward, we will hold brief, directed discussions on your project status and next steps at the timeslots noted below. You should schedule a 10 minute timeslot during these windows at the start of each week using the Google Sheet here, and prepare a brief update based on the guiding questions below.

Below are the planned schedule and topics for each of these check-in discussions:

Project Update 1: Initial Ideas – *written update submitted via email by Friday, 11/17*

- What are your initial ideas?
- Review overall project plan

Project Meeting 2: Design-Space Exploration - *Tuesday, 11/21 10am-12pm*

- What have you looked into?
- What are you hoping to explore in the next week?
- Has anything been implemented and tested?

Project Meeting 3: Made Decisions - *Tuesday, 11/28 10-11am; Thursday, 11/30 2-3pm*

- What DFT has been implemented?
- What fine-tuning/finalizing still needs to happen?

Project Meeting 4: Final Design - *Tuesday, 12/5 10am-12pm*

- What were your results?
- How is the presentation going?

10. Qualcomm Design-for-Test Contest

For the first time this semester, this project will also be a contest, in which the groups with the selected designs will be awarded with a cash prize. DFT experts from Qualcomm will be judging your projects through your reports and final presentations, and some will be attending the live session to ask questions about your design choices. Please use this opportunity to interact and learn from industry experts, and demonstrate understanding and creativity in your design.

Judges will be scoring you on the below criteria:

1. **Quality:** How effective is this design in achieving high test quality?
2. **Cost:** How effective is this design in achieving low test cost?
3. **Diagnosability:** How well would this design allow for diagnosis and debug of failures?
4. **Exploration/Analysis:** How well was the DFT space explored in arriving at the final design?
5. **Innovation/Creativity:** How well did the design analysis extend beyond standard DFT approaches?

The cash prizes awarded to each winning team (split between team members) are:

First Place: $3,000
Second Place: $1,500
Third Place: $500

11.Strategy

The process for completing the project can vary based on the DFT structures you choose to implement. There are many possible approaches — a couple of suggestions are described below, but these approaches are not necessarily sufficient in achieving the quality/cost requirements

Sequential Test Generation + Boundary Scan

You may choose to use boundary scan as a primary means for testing your circuit, then you can test the original logic of s9234 by "serializing" a test sequence for s9234. Siemens Tessent can be used to perform sequential automatic test pattern generation for s9234 circuit. Each clock cycle of the sequential test can be serialized and applied through the TAP (test access port) of the boundary scan implementation. By alternating the complete loading/unloading of boundary scan registers with functional clocks, sequential test patterns can be applied.

While this approach may seem reasonable, the number of clock cycles required for this approach will likely be very high, and it is possible that sequential ATPG will not be able to satisfy the fault coverage requirements.

Full Scan

A second approach centers on adding full scan. Specifically, s9234 can be converted to a combinational circuit by implementing one or more internal scan chains using the 211 flip-flops. Siemens Tessent can be used to generate high coverage test patterns on the combinational circuit, which can then be serialized and applied to load the scan flops of the DFT-inserted circuit. This will reduce the test set for s9234, as the circuit being tested is now virtually combinational. However, depending on the manner of implementation, there could be many cycles required to load/unload all scan flops on a single scan chain, or many tester channels required to load/unload parallel scan chains.

Full Scan + Boundary Scan Flops

Techniques can be combined by using full scan with boundary scan flops but choosing not to use TAP for test application. The use of boundary scan flops will reduce the number of PI/POs that require control from the tester by controlling/observing the combinational logic on the boundary of the chip, while choosing not to use the TAP may simplify test application and enable greater flexibility with other DFT techniques (compression, BIST, etc.).

While these more simple approaches may satisfy the project criteria, this project is intended to promote creativity in DFT implementation approaches. Additional alternate techniques for reducing cost and improving coverage should be explored such as BIST, test point insertion, and scan compression/decompression. Sections 15.2.5-15.2.8 of the textbook can help provide details and ideas into possible techniques, and some industrial approaches will be discussed in the course lecture/recitation.