Introduction

In this lab, you will design and implement your own CAD/EDA DFT Tool to add BIST capabilities to any given digital circuit. Please note that you will do this lab individually.

As your first show case, you will add BIST circuitry to the **Arithmetic Unit** (CUT) from Lab 1.

This BIST circuit, as shown in Figure 1, will have three major components:

a. a BIST controller,
b. a Test Pattern Generator (TPG), and
c. an Output Response Analyzer (ORA).

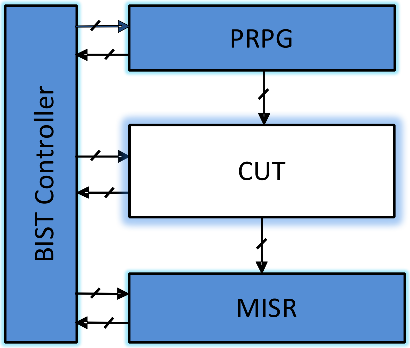

Figure 1: BIST Architecture

BIST controller initiates the testing by setting TEST signal '1' to enable the test mode in AU (Arithmetic Unit) (i.e. data inputs will be coming from the PRPG rather than normal operational input), and then set START signal '1' to have PRPG produce vectors at every clock.

For each input vector (including data and control signals), AU will produce an output, with which MISR will be accordingly updated.

Upon the completion of the vectors, PRPG will set COMPLETE signal '1'.

The controller will then compare MISR content with the expected (golden) signature that has been already calculated and stored at a register. If it matches, FAULT-FREE signal will be set '1', otherwise will be reset '0'.

Tasks expected from this Lab

1. You will design and implement a **PRPG** as the Test Pattern Generator**.** This PRPG will be as wide as necessary, i.e. it should include all input and control data, necessary for the AU to work properly.
2. Design and implement a **MISR** as the Output Response Analyzer**.** This will be wide enough to produce a signature at the end of the testing cycle. You will need to determine the final (golden) signature for a fault-free circuit, which can be stored in some register for comparison at the end.

3. Design and implement a **BIST controller** to coordinate the BIST logic. This controller will a) initiate ALU to go under the test (select proper mode in AU), b) initiate PRPG, and c) at end of the cycle, compare MISR content with the expected signature, d) and produce the proper outcome.
4. Design a testbench to **infuse specific faults** in the design to test the BIST circuit and show your simulation results for several faults. Please note that you can infuse specific faults into your design by modifying your source code for fault-free AU.
5. Demonstrate your **synthesized design** using Synopsys Design Compiler. This should include the complete circuit with the CUT, PRPG, MISR, and the Controller.
6. Compare and contrast your **ATPG-approach** (from Lab 1) and **BIST-approach** (from Lab 2) in terms of fault coverage, cost, time, efficiency, etc. For the fault coverage, you don't need to show the reports, just analyze which method gives a higher fault coverage, and explain why.

Your typed report shall include the following

1. Block diagram and explanation of your complete design with CUT and BIST Logic. Please include your selection criteria of BIST components and also how you calculated the golden (expected) signature.
2. VHDL/Verilog implementation of each of the three major BIST components.
3. Test benches for each component and simulation waveforms.
4. Simulation waveforms for test mode with FAULT-FREE (PASS) and FAULTY (FAIL) cases, and normal mode with some operations.
5. DC compiler script for synthesis and synthesis results.
6. Please elaborate also on what you've learnt and how your approach can be extended to create a complete DFT EDA software.