SV Lab 3: SV & Questa for CRV and Code Coverage

EEL 4712 – Fall 2025

Objective:

The objectives of this lab are to study the use of Questa for ***Code Coverage***, and continue the use of SystemVerilog (SV) and Questa for ***Constrained Random Verification (CRV)***. In **Part 1**, you will explore and made use of Questa's features for code coverage for design verification. In **Part 2** you will modify a provided testbench to implement and demonstrate the various SV and Questa capabilities for CRV.

Required tools and parts:

Intel Quartus Prime Lite, Questa-Intel FPGA (full version on linux.ece.ufl.edu)

Helpful resources:

- Tutorial for using Questa for code coverage

Pre-lab Requirements:

Part 1: Code Coverage in SV and Questa

You are provided with hlamgcd.sv (SV design file for the GCD module) and a testbench, hlamgcd_tb.sv.

- For **hlam**gcd.sv, you are to rename the file to **yourname**gcd.sv and the module to **yourname**gcd.
- For hlamgcd_tb.sv, you are to rename the file to **yourname**gcd_tb.sv and the module to **yourname**gcd_tb.
- Also made any minor change to the code as necessary because of the name changes.

You are also provided with the slide set, Code verification lecture slides.

(1a) Follow the instructions on Slide 7 to optimize and instrument **yourname**gcd_tb.sv for code coverage, except use only the **s** (statement), **b** (branch), and **f** (fsm) flags.

(1b) Follow the instructions on Slide 8 to set up the required windows to collect the code coverage information listed below (c) ... (f).

After you have performed (a) and (b) above, collect the code coverage information for (c) ... (f).

(1c) Produce the code coverage summary table (command: coverage report -summary).

(1d) Statement (s) flag: Produce a slide that is equivalent to Slide 11 of the provided slide set. Replace the contents of Slide 11 with the information you obtained from running the testbench, **yourname**gcd_tb. Include a short paragraph, describing the information you collected on that slide.

(1e) Branch (b) flag: Produce a slide that is equivalent to Slide 13 of the provided slide set. Replace the contents of Slide 13 with the information you obtained from running the testbench, **yourname**gcd_tb. Include a short paragraph, describing the slide.

(1f) FSM (f) flag: Produce a slide that is equivalent to Slide 18 of the provided slide set. Replace the contents of Slide 18 with the information you obtained from running the testbench, **yourname**gcd_tb. Include a short paragraph, describing the slide.

Part 1 Pre-lab submission instructions (on Canvas)

yournamegcd.sv, **yourname**gcd_tb.sv, **entire** Transcript window, and the deliverables described in (c), (d), (e), and (f).

Part 2: Constrained Random Verification (CRV)

You are provided with a testbench (**tb_part2_template.sv**). I suggest that you run the provided template code, as it is (without the added constraints, etc.) with your old Lab2 design code first. That would confirm that the provided testbench work properly with your Lab 2 design code. Then if there are errors later, then you would know that they are caused by the new modifications to the testbench (e.g., constraints) that you made.

In Part 2, you are to modify the testbench to add the specified constraints or modify the testbench code. The details are described below. Note that for each added constraint or modification, there is a corresponding comment in the provided testbench to guide you.

(2a) **Cross coverage**: Define a cross coverage for operands a and b. Note where it is specified in comment (2a) in the provided testbench.
Deliverable: Copy and paste two screens, as shown in the Cross_appendix.

(2b) **Constraint 1**: Define a constraint named **unique_operands** to generate different a and b values at each iteration. Again, see the corresponding comment (2b) in the provided testbench.
Deliverable: On one page, copy/paste of a part of the Transcript window, and one or two sentences describing how this part of the Transcript window shows (or at least does not contradict) this constraint.

(2c) **Constraint 2**: Using the **implication** operator **->**, define a constraint named **small_operands** which will constrain, "If a and b are both small (<25), then the opcode is changed to ADD".
Deliverable: On one page, copy/paste of a part of the Transcript window, and one or two sentences describing how this part of the Transcript window shows (or at least does not contradict) this constraint.

(2d) **Constraint 3**: Write a **soft** constraint named **soft_b_greater_a** in which b > a, but can be overridden by using an inline constraint to make it a > b (see the randomize() function in 2e).

(2e) **randomize () inline constraint overridding soft constraint:** Find the corresponding comment in the provided testbench (Comment (2e)) which invoke the randomize() function Modify the statement to override the soft constraint (2d) and make a > b.

Deliverable for 2d and 2e: On one page, copy/paste of a part of the Transcript window, and one or two sentences describing how this part of the Transcript window shows (or at least does not contradict) the results of 2d and 2e.

(2f) post-randomize() function: Insert code in the post-randomized() function where indicated by the Comment (2f), which will add 1 to both operands a and b.

Deliverable: On one page, copy/paste of a part of the Transcript window, and one or two sentences describing how this part of the Transcript window shows (or at least does not contradict) this constraint.

Part 2 Pre-lab submission instructions (on Canvas)

In addition of the deliverables stated in (2a) ... (2f), turn in your modified testbench.

In-lab

Demonstrate to your TA the Questa simulation of Part 1 and Part 2. Be prepared to answer questions or to make simple extensions that your TA may request. If you have done the pre-lab exercises, these questions/extensions should not be difficult.

Grade Breakdown

Criteria	Points
----- Pre-lab -----	
Part 1	
Deliverable (1c)	5 pts
Deliverable (1d)	5 pts
Deliverable (1e)	5 pts
Deliverable (1f)	5 pts
Part 2	
Modified tb_part.sv	25 pts
Deliverable (2a)	5 pts
Deliverable (2b)	5 pts
Deliverable (2c)	5 pts
Deliverable (2d, 2e)	5 pts
Deliverable (2f)	5 pts
----- In-Lab -----	
Demonstrate Part 1 simulation and answer questions	10 pts
Demonstrate Part 2 simulation and answer questions	20 pts