An-Najah National University
Computer Engineering Department

VLSI Design Verification
Final assignment
Functional Coverage

Constraint-random testing (CRT) is very dominant in modern test bench and verification approaches. For CRT to be effective, it is usually combined with functional coverage to clearly quantify the how good is the CRT stimulus as a first step. Advanced functional coverage goes beyond identifying how some variables are driven by the TB or the DUT, but also identifying complex stimulus/output combinations and how well the TB drives these complex scenarios.

In this exercise, the goal is to experiment further with functional coverage.

Instructions:

1. Download the file coverage_class.sv:

   https://www.asic-world.com/code/systemverilog/coverage_class.sv

2. Update the TB in the example to achieve 100% functional coverage. Your solution should include the following features:
   a. Must use classes with **rand** variables and constraints on those random variables.
   b. Then use a randomized transaction object for driving the values to the DUT.

3. Verify that the DUT works as expected: include a checker class which monitors the signals going to/from the DUT and issue an error when the expected vale is not identical with the copy kept at the TB. Hint: there should be on bugs in the downloaded example, the goal is to know how to verify this.

4. Avoid driving specific values to the DUT. You should always use values generated by the randomize method.

Good Luck

Note1: It is sufficient to display the get_coverage function to display all coverage for all covergroups.

Note2: It is possible to generate the full coverage report using EDA Playground. For this you need to use run.bash option and write a script to generate the coverage report. Try to search for an example on EDA playground performing this task.

Note3: Exercise is based on ASIC-World functional coverage pages: https://www.asic-world.com/systemverilog/coverage1.html