

Functional Coverage of the LC3

Guidelines

- 1) The following are the requirements for this project:
 - You need to code the functional coverage constructs for the LC3 from Project 2a based on concepts presented in LAB 5.
 - You need to achieve full functional coverage for the coded coverage constructs.
- 2) Coverage should be comprehensive with regards to analyzing:
 - The complete instruction set of the LC3 architecture (ALU, Control & Memory operations and different fields within the instructions)
 - The sequence of instructions (eg: a branch operation followed by another branch operation)
 - The inputs, outputs and their relationship for each of the blocks constituting the LC3
- 3) Your coverage will be compared with our coverage benchmark. You will be evaluated based on how well you have understood the special cases of the inputs, corner cases and your approach in covering the possibilities of the inputs. Sending 2^{32} instructions to obtain 100% coverage for a 32 bit field is not what we expect in this project. You can change your Scoreboard in the Project 2a, to achieve the coverage.

The DEADLINE for this project is 2nd December 10pm Wednesday 2008.

Submission requirements

1. Copy modelsim.ini in your submitting folder.
2. Make sure that the DUT instance name is “DUT” in your LC3_test_top file.sv file.
Eg : LC3 DUT (.clock(clock)). Also make sure you have reset in the procedure.
3. Submit the .v, .sv, .vp files that are being used in your project (Note: submit all .vp files being used for the LC3 and Cache compilation).
4. Do not submit .wlf files or any other unnecessary large files.
5. **Do not submit the mti_lib folder** or its contents (we will create this when we run your simulation).

6. Submit a single *.do* submission file (Group 33 submits only *grp33submit.do* for example and not submit multiple *.do* files) that runs your entire simulation for as long as it takes for you to achieve the coverage goals you have set for verification. The aim is to be enable us to recreate all of the individual ucdb files that you need for the final merged ucdb file. One of two methods can be adopted:
- If you have all your input constraints specified in your *Packet.sv/Generator.sv* file and you only need to run your simulation once for a long time, say 100000ns, then specify that in the *.do* file.
 - If you are required to “change” the stimulus or randomization constraints for achieving your coverage goal, you will need to create separate *.sv* files for the same. So, *Packet1.sv*, *Packet2.sv*, ..., *Packetn.sv* files will need to be created by you. You will compile each of these stimulus files separately (as shown in the example below) in order to create the individual *.ucdb* files.
7. Please verify that the *.do* file you are submitting gets compiled error free.
8. Make note of the following in the example *.do* file below:
- The design units (*.vp* and *.v* files) need to be compiled only once in the beginning.
 - The *Packetn.sv* files are compiled along with the rest of the testbench files when there is a change in the stimulus for coverage.
 - Please ensure that you provide the *-c*, *-coverage*, *-novopt* options with *vsim*.
 - Do not miss the “run” and “coverage” commands as shown below.
 - Between two simulation runs, you will mostly need a “restart -f” command to ensure that your design resets and you start afresh for your next coverage run.
 - For the example below, say, you have 3 different kinds of stimulus files, you will give the “vcover merge” command.
 - Please save the final coverage report in the format *group[n]coverage.ucdb* – eg *group33coverage.ucdb* in the example below.
9. Please note: Five points will be deducted from your final project score if you do not follow the instructions outlined in this document. If your submitted *.do* file does not compile, 10 points will be deducted.
10. Please submit a **report** on your coverage goals in a PDF format as **grp_{xx}_report.pdf** and include the report in the folder that you are submitting. This document must briefly explain and summarize the coverage plans for the instruction set, internal nets, sequence of instructions and any special cases which you have used for achieving your coverage goals.

Eg: .do file:

```
vlog *.vp TopLevelLC3.v
```

```
//coverage for Packet1
```

```
vlog -sv -mfcu Packet1.sv Driver.sv Receiver.sv Scoreboard.sv Scoreboard_Coverage.sv  
Generator.sv DataMem.sv LC3.tb.sv LC3.test_top.sv LC3.if.sv
```

```
vsim -c -coverage -novopt LC3_test_top
```

```
run 1200
```

```
coverage save ./cov1.ucdb
```

```
restart -f
```

```
//Coverage for Packet2
```

```
vlog -sv -mfcu Packet2.sv Driver.sv Receiver.sv Scoreboard.sv Scoreboard_Coverage.sv  
Generator.sv DataMem.sv LC3.tb.sv LC3.test_top.sv LC3.if.sv
```

```
vsim -c -coverage -novopt LC3_test_top
```

```
run 2000
```

```
coverage save ./cov2.ucdb
```

```
restart -f
```

```
//Coverage for Packet3
```

```
vlog -sv -mfcu Packet3.sv Driver.sv Receiver.sv Scoreboard.sv Scoreboard_Coverage.sv  
Generator.sv DataMem.sv LC3.tb.sv LC3.test_top.sv LC3.if.sv
```

```
vsim -c -coverage -novopt LC3_test_top
```

```
run 8000
```

```
coverage save ./cov3.ucdb
```

```
//merge all the coverages into one ucdb file
```

```
vcover merge grp33coverage.ucdb ./cov1.ucdb ./cov2.ucdb ./cov3.ucdb
```