The University of Alabama in Huntsville
Electrical & Computer Engineering
CPE 426/526
Spring 2021
Homework #4

Due April 11, 2021

Take the VHDL model of the arbiter provided and instantiate it in a SystemVerilog top module that also contains an interface, 4 instances of the provided request model, 4 test programs (one for each request device) and one test program that randomly generates resets for the system. Create random variables for the (1) number of cycles consumed before a device generates a request, (2) the number of cycles a request is kept high, (3) the number of cycles consumed before a reset is generated, and (4) the number of cycles the reset is asserted.

Use the following constraints:

(1) Range: 101 – 299 (2) Range: 1 – 100 (3) Range: 301 – 499 (4) Range: 4 – 9

Even though you are using different instantiations of a class, the method will give the same results. So, force the system to have different values by changing the number of calls to randomize in each test program.

```
test0
  repeat(200)
    p.randomize();
     …

test1
  repeat(200)
    p.randomize();
    p.randomize();
    …
```

```
test2
  repeat(200)
    p.randomize();
    p.randomize();
    p.randomize();
    …

test3
  repeat(200)
    p.randomize();
    p.randomize();
    p.randomize();
    p.randomize();
    …
```

Compile all your files with all forms of coverage enabled. Optimize using the command vopt top -o opttop +cover=sbecft from the transcript window, where top is your top-level module. Run 200 different test cases for each test program. Generate a coverage report for all types of coverage using Tools -> Coverage Report -> Text. Select Details and all kinds of Coverage.

Turn in your SystemVerilog source files and your coverage report file.