ECE 745: ASIC VERIFICATION
LAB 1

Introduction:
In this laboratory exercise, you are going to complete the Questa tutorial at NCSU and get yourself acquainted with some of the fundamental components (interface, program body and top-level integration) of a testbench. You will also learn to recognize a few basic data-types and their usage in a testbench environment. The basic idea is to a) Get the feel for a typical testbench using SystemVerilog and b) to get comfortable with the Questa verification environment c) To get acquainted with the DUT that will be used in all the Laboratory exercises henceforth.

Lab1 Requirements:
Step1:
To enable an understanding of the Questa tool and the complexities associated with it, please complete the Questa Tutorial at:
http://www.eda.ncsu.edu/wiki/Tutorial:Questa_SystemVerilog_Tutorial

Please note that a basic description of the Execute block that will be used as the DUT and the valid commands for it can be found at
http://www.ece.ncsu.edu/asic/asic_verification/shared/System_Spec/System_Spec.pdf

Lab1 Questions:
Answer the following questions:

a. What does `repeat(5) @(Execute.cb)` do?
b. Describe the functionality and the timing characteristics of the reset task
c. There are multiple tasks defined in the testbench provided. Based on the concepts conveyed in Lecture 1, please state the layers (in a layered testbench) that the tasks would belong to.
d. What lines would you modify to make the testbench run 5 times with 10 instructions being sent into the DUT each time?
e. What set of inputs would you send in (relevant input signals only need to be stated) for the following to be executed
    a. Signed Addition: `aluout = -45 + 60` with `immp_regn_op = 1`;
    b. Arithmetic Left Shift : `aluout = -23 left shift by 5 where 5` is an immediate value.
    c. Write `32'hbaad` to memory (i.e. `mem_data_write_out`)
    d. Load `32'hbaad_face` from **memory/src2** but with the following variations
        i. Load sign-extended byte [7:0]
        ii. Load sign-extended word [15:0]
        iii. Load entire 32 bits [31:0]
        iv. Load word [15:0]to most significant word of output

f. Send in each of the inputs from the previous 2 questions into the DUT by modifying the testbench program. Also, determine whether the outputs are along expected lines. If not, state the errors.

Lab1 Submission Requirements:

To show that the required inputs are sent in, you need to capture the resulting outputs from the DUT using `$display` statements of the form shown below:

```
$display($time, "ns:  Outputs from DUT: aluout = %h,      carry = %b,
mem_write_en = %b,     mem_data_write_out = %h", Execute.cb.aluout,
Execute.cb.carry, Execute.cb.mem_write_en,
Execute.cb.mem_data_write_out);
```

Make sure that these results are displayed by running your program just as stated in the tutorial. The same file names (`Execute.tb.sv`, `Execute.test_top.sv`, `Execute.if.sv`) need to be used.

Follow the following steps for submissions (Solaris/Linux only please)

- `mkdir Lab1` (creates the directory Lab1)
- copy all the SystemVerilog files into the Lab1 directory
- Answer all the above questions in a file called `Lab1.doc` and copy this into the Lab1 directory
- Zip the file using the command > `zip Lab1.zip Lab1/*`
- Submit the zip using the submit utility on the course webpage.