Project 2 Simulation

Ben Heard

Abstract

Verification of any design is a requirement to ensure that the system performs as expected. One form of verification is when you compare your design results with those of a known good implementation. This lab is intended to help you with your project implementation by working through a testbench of your own for the encrypted solution.

Rather than complete a fully detailed testbench you will leverage the ECE310 library tb_player module to stimulate the encrypted DUT using a stimulus data file that you create. The intent is you give you practice creating data files so that you may test your own design.

1 Getting the Content

This has been covered a number of times in the previous labs. Please refer to them for reference. Once you pull the latest from your repository you will have a new folder called lab_008 and, in it, there will be a modelsim.ini file, this PDF, and the project 2 encrypted solution and testbench.

2 Hello Testbench

The testbench provided as a reference in the lab 8 materials is the Hello testbench that stimulates the DUT with a size of 5 and data that represents the ASCII for Hello. If you open the data file Hello_sim_tb.dat you can see how the stimulus progresses each clock cycle. The listing below has been annotated with the line numbers of the file for reference

```
01: 0_00000000000000000000000000000000_0_00000000
02: 0_00000000000000000000000000000000_0_00000000
```

```
03: 0_00000000000000000000000000000000_0_00000000
04: 0_00000000000000000000000000000000_0_00000000
05: 1_00000000000000000000000000000101_0_00000000
06: 0_00000000000000000000000000000000_0_00000000
07: 0_00000000000000000000000000000000_0_00000000
08: 0_00000000000000000000000000000000_0_00000000
09: 0_00000000000000000000000000000000_0_00000000
10: 0_00000000000000000000000000000000_0_00000000
11: 0_00000000000000000000000000000000_1_01001000
12: 0_00000000000000000000000000000000_0_00000000
13: 0_00000000000000000000000000000000_0_00000000
14: 0_00000000000000000000000000000000_1_01100101
15: 0_00000000000000000000000000000000_1_01101100
16: 0_00000000000000000000000000000000_1_01101100
17: 0_00000000000000000000000000000000_0_00000000
18: 0_00000000000000000000000000000000_1_01101111
19: 0_00000000000000000000000000000000_0_00000000
20: 0_00000000000000000000000000000000_0_00000000
21: 0_00000000000000000000000000000000_0_00000000
22: 0_00000000000000000000000000000000_0_00000000
23: 0_00000000000000000000000000000000_0_00000000
```

The bits of stimulus are separated by _ characters. This is supported by Verilog and is there to make the file easier to read. The first bit is the size_valid pulse. The next 32 bits are the size. The next bit is the data_valid pulse and the final 8 bits make up the data.

Line 5 is where size_valid is asserted with the binary 3'b101 value (decimal 5) as the size. Later, data_valid is asserted at various times (11, 14, 15, 16, and 18). Each time it is asserted a new character is provided. The character is the ASCII equivalent of character, specifically for 'H', 'e', 'l', and 'o' with the l (el) being send twice.

3 ASCII

There are numerous websites that will show you the ASCII (American Standard Code for Information Interchange) table. One that I have used is

`www.asciitable.com`

4 Adler32

There are also a number of websites that will perform the Adler32 computation for you. Again, one that I use is

`https://md5calc.com/hash/adler32`

You may place any string in these calculators and have it compute the Adler32 checksum.

5 Lab exercises

5.1 Simulate Hello

The first part of the lab is to just simulate the existing Hello testbench to see how it works and to inspect the results. Perform the following steps.

```
% cd path/to/lab_008
% module load modelsim
% export MODELSIM=modelsim.ini
```

At this point you've set up ModelSim. The next steps create a work library, compile the design, compile the testbench, and run the testbench.

```
% vlib work
% vmap work work
% vlog *.vp
% vlog Hello_sim_tb.v
% vsim -c -voptargs="+acc" -L ece310_lib Hello_sim_tb

VSIM> log -r *
VSIM> run -all
VSIM> quit
```

Now that the design has been compiled and simulated you may open the results in the ModelSim GUI and inspect the waveforms.

```
% vsim vsim.wlf
```

5.2 Create your own data set

Consider some string of your own that you would like to verify with the encrypted solution. Using an online calculator, generate the Adler32 checksum for the string. Then, complete the lab_008_tb.dat file with the listing to stimulate the DUT with your string.

Please include the string that you are testing as a comment in the .dat file. Note that you will need to determine the string length for the size and convert the string characters into ASCII.

5.2.1 ECE310

For example, and don't use this one, you could create the string 'ECE310'. In ASCII, this is

```
E = hex 45
C = hex 43
E = hex 45
3 = hex 33
1 = hex 31
0 = hex 30
```

Putting this into the Adler32 calculator yields 32'h05320162. The following is an example listing for the string 'ECE310'.

```
0_00000000000000000000000000000000_0_00000000
0_00000000000000000000000000000000_0_00000000
0_00000000000000000000000000000000_0_00000000
1_00000000000000000000000000000110_0_00000000
0_00000000000000000000000000000000_0_00000000
0_00000000000000000000000000000000_1_01000101
0_00000000000000000000000000000000_1_01000011
0_00000000000000000000000000000000_1_01000101
0_00000000000000000000000000000000_1_00110011
0_00000000000000000000000000000000_1_00110001
0_00000000000000000000000000000000_1_00110000
0_00000000000000000000000000000000_0_00000000
0_00000000000000000000000000000000_0_00000000
0_00000000000000000000000000000000_0_00000000
```

5.3 Simulate your data set

Once you have your own data set and have completed the first exercise (simulating Hello) you can simulate with

```
% vlog lab_008_tb.v
% vsim -c -voptargs="+acc" -L ece310_lib lab_008_tb

VSIM> log -r *
VSIM> run -all
VSIM> quit
```

Now that the design has been compiled and simulated you may open the results in the ModelSim GUI and inspect the waveforms.

```
% vsim vsim.wlf
```

5.4 What to turn in

Your lab_008_tb.dat file with the string that you're encoding as a comment at the top of the file.