**ECE 6213 – Design of VLSI Circuits**

**Fall 2022 – The George Washington University – Dr. Jerry Wu**

**Osama Yousuf – HW 1**

**Problem 1 – 40 points – Design a testbench**

* **Ch-5 – Problem 13: Verify the functionality of *Universal\_Shift\_Reg* in Example 5.45.**

Verilog code: *universal\_shift\_reg.v*

Testbench: *universal\_shift\_reg\_tb.v*

Output waveform from my testbench:

A screenshot of a computer

Description automatically generated with medium confidence

Corresponding waveform from the book – the input and output waveforms align one-to-one.

A sheet of music

Description automatically generated with low confidence

* **Ch5 - Problem 14: Write a Verilog description of the circuit shown in Figure P5-15 and verify that the circuit's output, , is asserted if successive samples of have an odd number of Is.**

Verilog code: *d\_flipflop.v, p5-15.v*

Testbench: *p5-15\_tb.v*

My testbench tests for the following cases:

* When successful samples of have an even number of 1s, the output remains 0
* When successful samples of have an odd number of 1s, the output remains 1
* When is maintained at 1, alternates between 1 and 0 as the count of 1s in alternates between odd and even, respectively.
* The reset pin resets the count of the module correctly, and the clk pin of the flip flop operates correctly.

Waveform:

Graphical user interface, application

Description automatically generated

**Problem 2 – 40 points – 8 bit counter**

* **Ch-5 – Problem 15: Develop and verify a Verilog model of a 4-bit binary synchronous counter with the following specifications: negative edge-triggered synchronization, synchronous load and reset, parallel load of data, active-low enabled counting.**

Verilog code: *counter\_4bit.v*

Testbench: *counter\_4bit\_tb.v*

My testbench tests for the following cases:

* Reset pin works correctly, output is unknown when the counter has not been reset
* Counting is done synchronously with the clock pin, on negative edge
* The counter up-counts only when the enable pin is 0, i.e. that the counter is active-low enabled
* When the load pin is 1, the counter synchronously loads in parallel whatever 4-bit input is present on its data in port (D in my results)

Text format output:

*A picture containing table

Description automatically generated*

Waveform level verification:

Graphical user interface

Description automatically generated

* **Ch-5 – Problem 16: Modify the counter of the previous problem to have an additional output (ripple carry output [RCO]) that asserts while the counter is at 1111)2. Cascade two such counters and demonstrate tbat the unit now works as an 8-bit counter.**

Verilog code:

* + *counter\_4bit\_rco.v* – This is a modified version of my implementation with *counter\_4bit.v*. The load and reset are asynchronous for simplicity.
  + *counter\_8bit.v –* This is the 8-bit counter. It instantiates two 4 bit rco counters, and the cascading is done such that the rco output of the first counter acts as the clock of the second counter.

Testbench:

* + *counter\_8bit\_tb.v*

My testbench tests for the following cases:

* Reset pin works correctly, output is unknown when the counter has not been reset
* 8-bit counting is done synchronously with the clock pin, on negative edge
* The counter up-counts only when the enable pin is 0, i.e. that the counter is active-low enabled
* When the load pin is 1, the counter asynchronously loads in parallel whatever 8-bit input is present on its data in port (D in my results)

Text format output:

Table

Description automatically generated with medium confidence

The complete text format output is not shown, but the following highlights the scenario when the rco of the first 4-bit counter is 1, and triggers the negative edge of the second counter (indicated by the black arrow):

A picture containing table

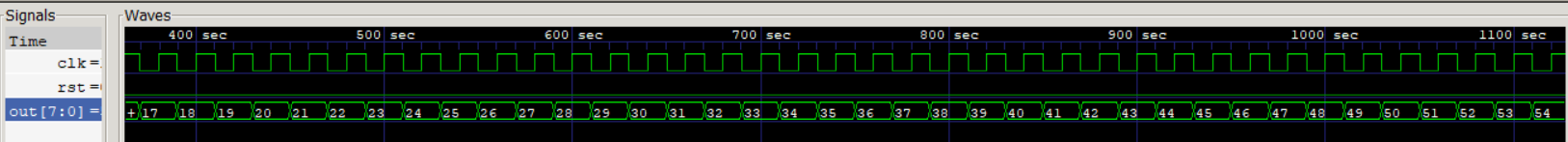
Description automatically generated

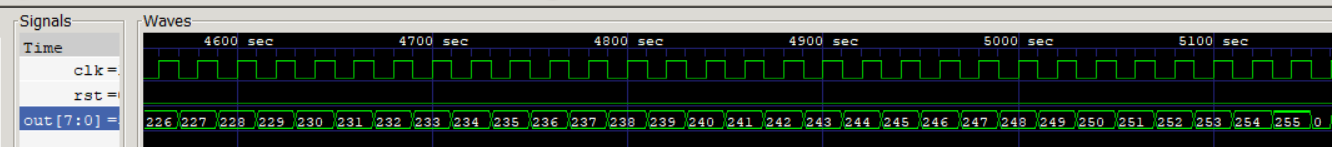
Waveform output, focusing on that all 8-bits are working:

The total digit that the counter can represent is 28 – 1 = 255.

The following snippets show the counting operation works as intended for all 8 bits. The output cycles back to 0 after the maximum possible digit – 255 – has been counted.

Note: Waveforms are in decimal format.





The following snippet shows that the counter can asynchronously load input data directly, and resume counting from the loaded data. It also shows that the reset pin can reset the counter from its initial state i.e. 0.

A screenshot of a computer

Description automatically generated