CME341 - October 2020 Midterm
Material up to end of Assignment 5

`Date: Wednesday, October 14, 2020, 12:30PM (Saskatoon time)`
`Time = 1.00 hours + 0.50 hours to upload files`
`Solution must be submitted by 2:00PM`

Instructions

Exam-Dependent Seed Setup

Before starting the exam modify the second "initial" procedure in the test-bench so that `exam_dependent_seed` is assigned to the last two digits in your student number, treated as a HEXADECIMAL number. For example, if your student number is 11263632, you should set the `exam_dependent_seed` to `8'h32`. You will use this same seed for all of the questions on the exam.

Academic Honesty

In answering the following questions, you may refer to any of the materials on the course website (notes, slides, lectures, etc.). You MAY NOT discuss any of the questions with anyone during the exam. By submitting your exam, you certify that you have not discussed the questions with anyone or used any other resources (aside from those on the course website) to assist with your solution. The instructors and the college may use various methods to detect academic dishonesty and the punishments for such dishonesty can be severe.

Exam Submission

There are a total of five questions, all having equal value. Please answer all of the questions.

Your exam submission must be uploaded to Canvas by the deadline listed above. It should be a single archive consisting of the following:

1. A plain text file (.txt) consisting of your answer to each question, which is the value of `accumulator_output`, when `counter_full_bar` is low for the `exam_dependent_seed`. Please add a header at the top of the file listing your name, NSID, student number, and the value used for your `exam_dependent_seed`.

2. For each question you solved, the SystemVerilog code for the modified student circuit. Based on the guidelines given in class, this would typically be a file called `student_circuit_q#.sv`. **Note that you MUST include the source file in order to get credit for the question.**

Good luck!

(2)

1. Re-design the student circuit to implement a combinational logic circuit that detects whether the input signal (`cct_input`) is between 8'd7 and 8'd11 (inclusive). When the input falls within that range, the output (`cct_output`) should be equal to the input. Otherwise, the output should be equal to zero. The clear input should be ignored for this question.

   The accumulator output, i.e. `accumulator_output`, when `counter_full_bar` is low for `seed = 8'HAA` should be `16'H7868`.
   Once `accumulator_output` is correct for `seed = 8'HAA` report it for the seed specified in the instructions.

(2) 2. Re-design prototype `student_circuit` to make it describe a sequential logic circuit that operates as a counter. When the clear input is high, the output (`cct_output`) should be synchronously set to 0. Otherwise, the output should increase by 3 on each positive edge of the clock. The input (`cct_input`) should be ignored for this question.

   For this question, the correct result for `seed = 8'HAA` is not specified. Please debug your circuit using Modelsim and the provided testbench until you are confident it is working according to the specifications above, then report the accumulator output, i.e. `accumulator_output`, when `counter_full_bar` is low for the seed specified in the instructions.

(2) 3. Re-design prototype `student_circuit` to make it describe a sequential logic circuit whose output is equal to the two's complement of its input (`cct_input`), delayed by two clock cycles (two positive edges of the clock). The clear input signal should be used as an active-high asychronous reset, which clears the output and any intermediate flip-flops contained within the circuit.

   The accumulator output, i.e. `accumulator_output`, when `counter_full_bar` is low for `seed = 8'HAA` should be `16'H8FBD`.
   Once `accumulator_output` is correct for `seed = 8'HAA` report it for the seed specified in the instructions.

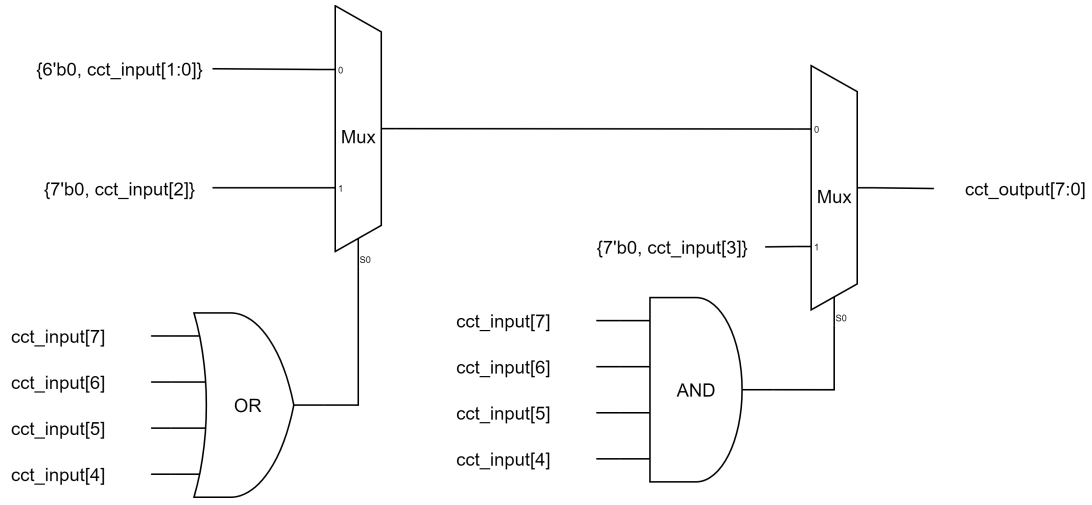

Figure 1: Circuit for Question 4

(2) 4. Re-design `student_circuit` to implement the circuit shown in Figure 4.

For this question, the correct result for `seed = 8'HAA` is not specified. Please debug your circuit using Modelsim and the provided testbench until you are confident it is working according to the specifications above, then report the accumulator output, i.e. `accumulator_output`, when `counter_full_bar` is low for the seed specified in the instructions.

(2) 5. Re-design prototype `student_circuit` to implement a combinational circuit according to the following specifications:

The circuit treats its input (`cct_input`) as two separate unsigned numbers: one represented by its four most significant bits ([7:4]), and one represented by its four least significant bits ([3:0]).

- If both numbers are odd, all of the bits in the output (`cct_output`) should be set to 1.
- If both numbers are even, all of the bits in the output should be set to 0.
- If one of the numbers is odd and the other is even, output bits [4], [5], and [6] should be set to 1 and all other bits should be set to 0.

For the purposes of this question, the clear input should be ignored and zero shall be considered as an even number.

The accumulator output, i.e. `accumulator_output`, when `counter_full_bar` is low for `seed = 8'HAA` should be `16'HF967`.
Once `accumulator_output` is correct for `seed = 8'HAA` report it for the seed specified in the instructions.