CME341 - October, 2018 Midterm
Material up to end of Assignment 5

`Date: Wednesday, October 10, 2018`
`Time = 1.00 hours`
`Text Books, Notes and Computer Files Only`

Before starting the exam modify the second "initial" procedure in the test bench so that `exam_dependent_seed` is assigned the value on your answer sheet.

(2)

1. Consider the Verilog HDL below:

```
module exam_ques (
input clk,
input [7:0] exam_dependent_seed,
output reg [15:0] Q4
                  );
reg [15:0] Q1, Q2, Q3;
always @ (posedge clk)
begin
    Q1 = {exam_dependent_seed,8'HFF};
    Q2 = Q1 & 16'HFF;
    Q3 = Q1 | Q2;
    Q4 =  (16'hFFFF ^ Q2);
    Q4 = Q1 ^ 16'h0F0F;
end
endmodule
```

Report the 16 bit input to the `Q4` register (For those that may not know, in this case the word "register" means a bank of 16 flip/flops.). For `exam_dependent_seed` = `8'HAA`, `Q4` = 16'HA5F0.

For this question the test bench is not used.

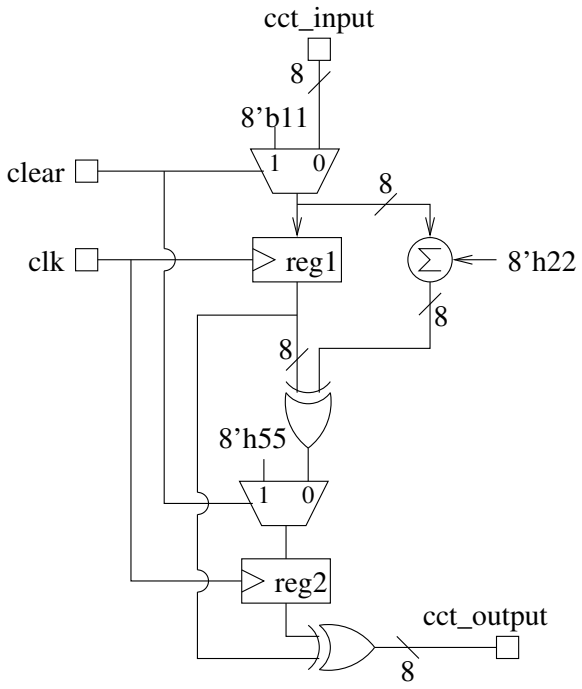

Figure 1: Circuit for Question 3

(2) 2. Re-design prototype `student_circuit` to make it describe a combinational logic circuit that uses the input much like a data selector to do the following:

Sets `cct_output` to zero while `clear` is high. Otherwise

If the input is equal to 8'b10 `cct_output` is to be the ones complement of `cct_input`.

If input is greater than 35 (in decimal) `cct_output` is to be the sum of the least significant 4 bits of the input. This means `cct_output` will be between 8'd0 and 8'd4, inclusive.

Otherwise `cct_output` is to be `cct_input`.

The accumulator output, i.e. `accumulator_output`, when `counter_full_bar` is low for `seed = 8'HAA` should be `16'H005b`.
Once `accumulator_output` is correct for `seed = 8'HAA` report it for the seed on your answer sheet.

(2) 3. Re-design the Verilog HDL prototype `student_circuit` to construct the circuit described in Figure 1.

The accumulator output, i.e. `accumulator_output`, when `counter_full_bar` is low for `seed = 8'HAA` should be `16'H818d`.
Once `accumulator_output` is correct for `seed = 8'HAA` report it for the seed on your answer sheet.

(2)

4. Re-design prototype `student_circuit` to describe a circuit with 8 flip/flops that does the following. The 8 flip/flops are organized into the 8 element vector `goofy_counter[7:0]`. `goofy_counter` is to be synchronously set to `goofy_counter = 8'b1000_0001` when `clear==1'b1`. Otherwise `goofy_counter` is to progress through the sequence given below on each rising edge of input `clk`:

   ```
   8'b1000_0001
   8'b0100_0010
   8'b0010_0100
   8'b0001_1000
   8'b1000_0001
   8'b0100_0010
   8'b0010_0100
   8'b0001_1000
   8'b1000_0001
     etcetera
   ```

   Make `cct_output = goofy_counter`.

   Note this circuit does not use `cct_input`.

   The accumulator output, i.e. `accumulator_output`, when `counter_full_bar` is low for `seed = 8'HAA` should be `16'Hfde3`.
   Once `accumulator_output` is correct for `seed = 8'HAA` report it for the seed on your answer sheet.

(2)

5. Re-design prototype `student_circuit` to make `cct_output` equal to `cct_input` shifted two bits to the right (i.e. toward the least significant bit). The two least significant bits of `cct_input` are not used and the two most significant bits of `cct_output` are to be the same as the most significant bit of `cct_input`.

   Note neither `clk` nor `clear` is used.

   The accumulator output, i.e. `accumulator_output`, when `counter_full_bar` is low for `seed = 8'HAA` should be `16'H9166`.
   Once `accumulator_output` is correct for `seed = 8'HAA` report it for the seed on your answer sheet.