CDA 4203L Spring 2025
Computer System Design Lab
Lab 5 – GCD Design on FPGA
Lab time: March 25
Zip file Report Due Date: April 4

Note: This lab does not need your FPGA boards. It is a simulation-only lab with ISE. No demonstration is needed to the TAs. It is a team-based lab.

Objective: To implement RTL Design of Greatest Common Denominator (GCD) of numbers.

Description: Design GCD architecture control/data paths for two 4-bit numbers **(in your lecture notes, we have already done this)**. It will output the binary value of the greatest common divisor of those two 4-bit numbers.

1. **Datapath**
   The datapath top-level module can be constructed by the instantiated individual components. You can create these individual behavioral components (Register (DFF), not-equal, less-than, subtractor, MUX, etc.) for each element in the datapath. Some examples are given below.
   Registers are the only clocked component in datapath (Example 1 below). Registers in datapath should have enable signals coming from FSM module (to load X, Y, etc. to DFFs in your datapath), and separate reset signals. All non-register components must be purely combinational (Example 2 next page shows some but not all such components).

Example 1, one way to implement your registers:

D Flip Flop

```
module dff(clk, d, q, enable, reset);

input clk;
input [3:0] d;
input enable, reset;
output [3:0] q;

reg [3:0] q;

initial begin q = 0; end


always @(posedge clk) begin

        if (reset && !enable) begin  q = 0; end

        else if (enable && !reset) begin        q = d; end

        else q = q;
end


endmodule
```

Example 2, two of the components in the datapath but you need more:

Not Equal

```
module NEQ(X, Y, x_neq_y);

        input [3:0] X;
        input [3:0] Y;

        output x_neq_y;

        assign x_neq_y = (X != Y) ? 1'b1:1'b0;


endmodule
```

Less Than

```
module less_than(X, Y, x_lt_y);

        input [3:0] X;
        input [3:0] Y;

        output x_lt_y;

        assign x_lt_y = (X < Y) ? 1'b1:1'b0;

endmodule
```

2. Control path with FSM

The FSM top-level module, according to the current state, output signals that the datapath top-level module use to perform arithmetic and to load or output from registers. The datapath top-level module sends signals back to the FSM to use. The controller is run by the FSM, and it could run on an opposite clock of the datapath (or you could run the clock for datapath with different speed, or use any trick to make sure the "load" inputs are picked up by datapath registers correctly). Develop a Moore Verilog code for FSM. FSM templates were given in Lab 4, use them here too.

In your FSM module, you can use parameters for the states, below is just one example you might want to use in your Verilog (the states and their names below match those of the GCM example):

```
localparam
        state_1 = 4'b0000,
        state_2 = 4'b0001,
        state_2J = 4'b0010,
        state_3 = 4'b0011,
        state_4 = 4'b0100,
        state_5 = 4'b0101,
        state_6 = 4'b0110,
        state_7 = 4'b0111,
        state_8 = 4'b1000,
        state_6J = 4'b1001,
        state_5J = 4'b1010,
        state_9 = 4'b1011,
        state_1J = 4'b1100;
```

Points:
(50 pts.) Write structural Verilog for datapath and behavioral Verilog for controller (FSM). Use the design we discussed in the class. The FSM Verilog code should strictly follow the FSM template.

Your top_level entity (let us call it GCD_top) would instantiate the datapath and controlpath modules. Connect the inputs/outputs of these two modules correctly. Is/Os of FSM are mostly Os/Is of datapath and vice versa.

Deliverables (Only one .zip file per group) A zipped file (.zip) which includes these two items:
(a) Your group design files (Verilog Models and test benches). (b) A concise PDF group report (that includes your Verilog code and simulation results) needs to be included in the submitted .zip file.

Who submits? Only the group lead. Do not submit the same .zip file for each student.

PDF Report Organization to be included in your ZIP submission (A template is provided on Canvas):

- □ Cover sheet
- □ Verilog Code, Test Bench, and Simulation Results (Waveforms)
- □ Feedback: Hours spent, Exercise difficulty (Easy, Medium, Hard)

- □ You need to submit your group report on Canvas in ZIP format (only one .zip file).