# CDA 4203L Spring 2022

**Computer System Design Lab Lab 5 – GCD Design on FPGA**

***Assigned on Wednesday, 16th Feb***

***Due Date: 11:59 PM*, *Sunday, 6thMarch***

**Objective:** To implement RTL Design (structural datapath and behavioral controller) of Greatest Common Denominator (GCD) of two numbers to be tested on FPGA board.

**Description:** Design a GCD 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. The numbers will be input via dip switches, the result will be output via LEDs, and control is done via push buttons.

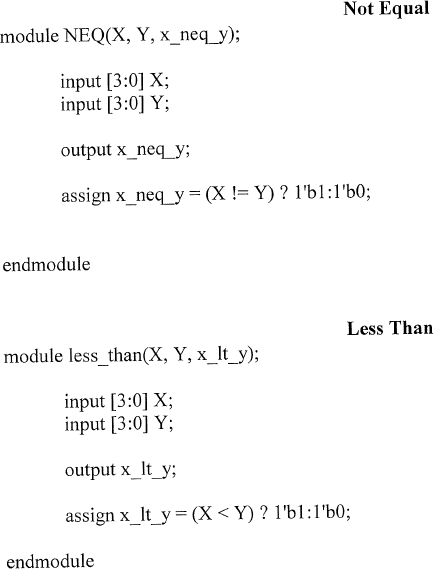
1. Input X via switches SW7-SW0
2. Input Y via switches DIP9-4 and DIP8-1
3. Input START uses pushbutton BTN0 (needs to be debounced)
4. Input RESET uses pushbutton BTN3
5. Output GCD OUT uses LD7-LD0
6. Output DONE uses LD9

## 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 module (Example 1 below) and they should be triggered by the clock on the opposite edge than that of Controller FSM. Registers should have an enable signal for control coming from FSM module (to load), and a separate reset signal. All non-register components must be purely combinational (Example 2 next page).

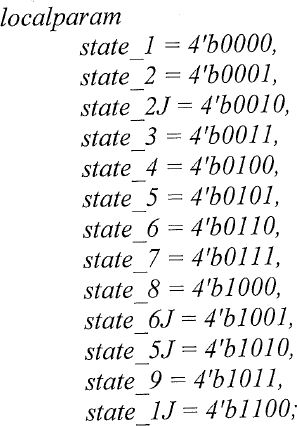
## þÿExample 1:

**Example 2:**

1. **Control path with FSM**

The FSM top-level module, according to the current state, outputs signals that the datapath top- level module uses to perform arithmetic and 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. Templates are given in Lab 4.

In your FSM module, you can use parameters for the states, below is just one example you might want to use in your Verilog:



## Points:

1. (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.
2. (25 pts.) Synthesize the design and verify its functionality on FPGA.

**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 one of group members needs to submit. Choose one student among your group members (it does not matter who) and that person submits. TAs will mark all group members based on that equally. 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
* Demonstration page (available on Canvas have it signed/marked by TA, attach to your report)
* Problem 1: Design details, Verilog Code, Test Bench, and Simulation Results (Waveforms)
* Problem 2: Pin Mapping file and synthesis report.
* Feedback: Hours spent, Exercise difficulty (Easy, Medium, Hard)
* You need to submit your group report on Canvas in ZIP format (only one .zip file).