EEL 3701C – Digital Logic & Computer System

LAB 5 – Datapath and Control Path

Due April 03rd 2020

The prelab should consist of answers to prelab questions, all requested VHDL, and any diagrams/tables you used in your solution. The simulation can be completed during the lab. Final report is due a week after the lab.

Problem 1: ALU Design (12 pts)

Figure 1 illustrates the architecture of a 4-bit ALU that must be designed for a simplify processor called the G-CPU processor.

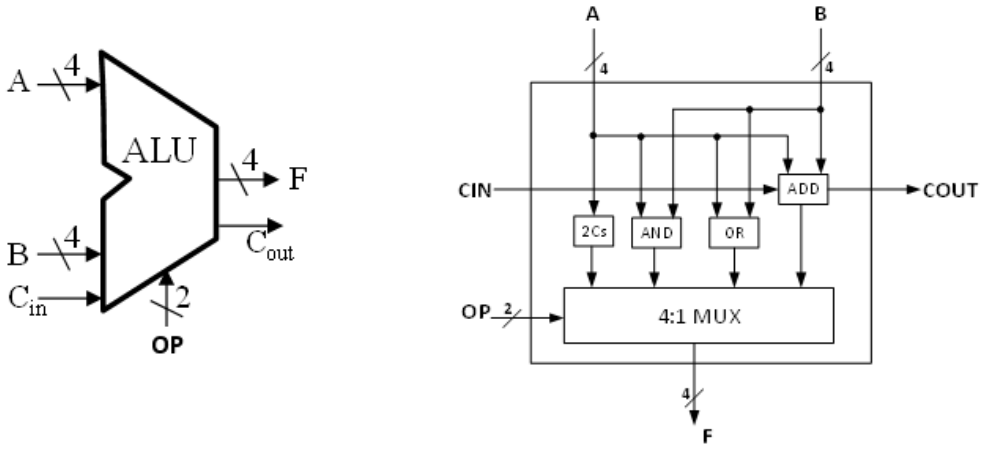

Figure 1: Schematic ALU representation

OP	Operation	Meaning
00	2Cs	F = 2's Complement of A
01	AND	F = A AND B
10	OR	F = A OR B
11	ADD	F = A + B; C_{out} = Carry of sum

Table 1: ALU Operations

1. Provide the VHDL implementation of the ALU. Recall that multiplexers are case differentiations that can be represented with "if then/else," "case/when," or "when/else" statements. You may use existing VHDL operators, including arithmetic and logic ones. **(7 pts)**
2. Simulate your design for functional correctness. Annotate important parts of the simulation with arrows and text. **(5 pts)**

Problem 2: G-CPU Datapath Design (17 pts)

In the second part of this lab, the ALU designed in Problem 1 will be used to build a datapath. The datapath is connected to an input bus to read externally inputted operands, and an output bus to send results out.

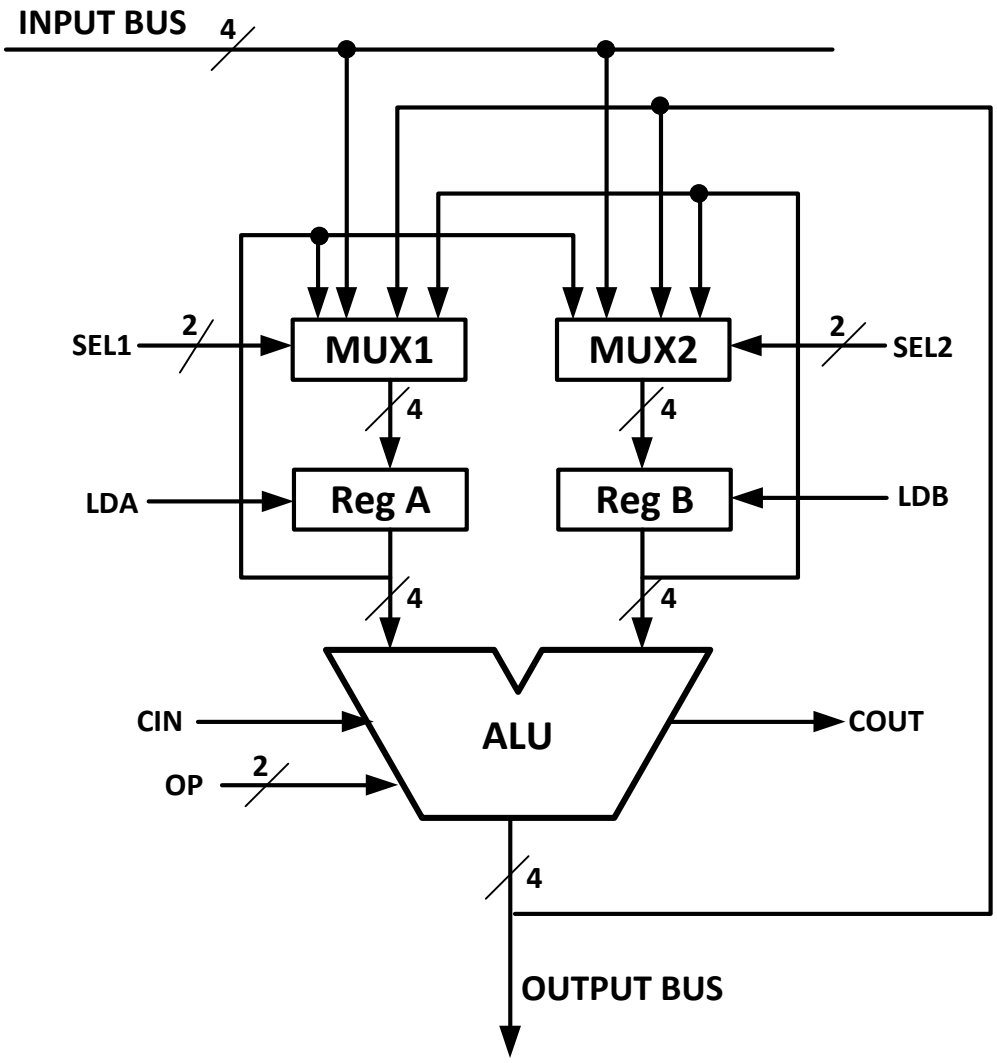

Figure 2: G-CPU Datapath

1. Enumerate and explain all possible computations that can be performed with this datapath. **(5 pts)**
    a. A computation is an operation on data in registers.
    b. All ALU operations are performed only on registers. This means that data must first be stored in those registers with load instructions before the operation can take place.
    c. The result of a computation can directly be placed on the output bus or copied back into a register.

2. How many clock cycles does it take to execute a 1-operand instruction (Ex. 2s complement of A)? How many cycles is necessary to execute a 2-operand instruction (ex. f+g). **(2 pts)**
3. Design the G-CPU datapath in VHDL and simulate your design for functional correctness. Annotate important parts of the simulation with arrows and text. **(10 pts)**

Problem 3: G-CPU Controller Design (23 pts)

Identify all control and status signals of the G-CPU for the design of a G-CPU controller. The purpose of the controller is to take an instruction as an input and guide the datapath to execute that instruction. The reading of the operands from INPUT and the storage of the results into registers must all be controlled by your controller.

As shown on Figure 3 Your controller should have **two** inputs: START, and the instruction (Remember: clock inputs are implied). It should have **one** output: DONE, to indicate that the requested instruction has completed.

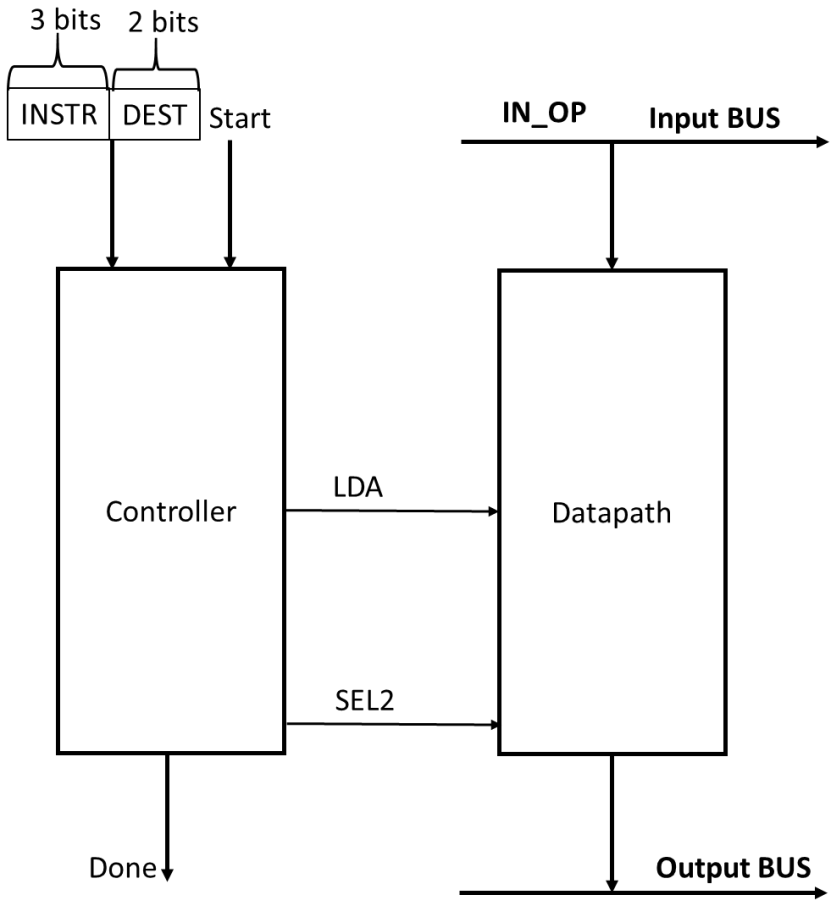

Figure 3: Control and Datapah for the G-CPU

Refer to the following tables for instruction layout, meaning, and the possible instruction (INSTR) values:

INSTR	IN_OP	DEST
What the G-CPU should do	Operand	Destination register
3 bits	4 bits	2 bits
Refer to Table 3	OP should be placed on INPUT	00 = RegA 01 = RegB 1- = Nothing

Table 2: Instruction format

INSTR	Opcode	Meaning
000	LOAD	DEST = IN_OP
001	COPYA	RegB = RegA
010	COPYB	RegA = RegB
011	AND	DEST = RegA AND RegB
100	OR	DEST = RegA OR RegB
101	ADD	DEST = RegA + RegB
110	SUB	DEST = RegB - RegA

Table 3: Possible INSTR values and their meanings

1. Complete the diagram of Figure 3 that shows how the controller interfaces (control, status, and data signals) with the datapath. Clearly label all inputs and outputs of the controller. **(2 pts)**
2. Design your controller as a Moore Automaton (use the template of Figure 5). Assume that your controller's initial state is expecting an instruction from a user and possibly a start signal. Upon receiving the start signal, the controller will then branch in different directions based on the INSTR field of the instruction. **(15 pts)**

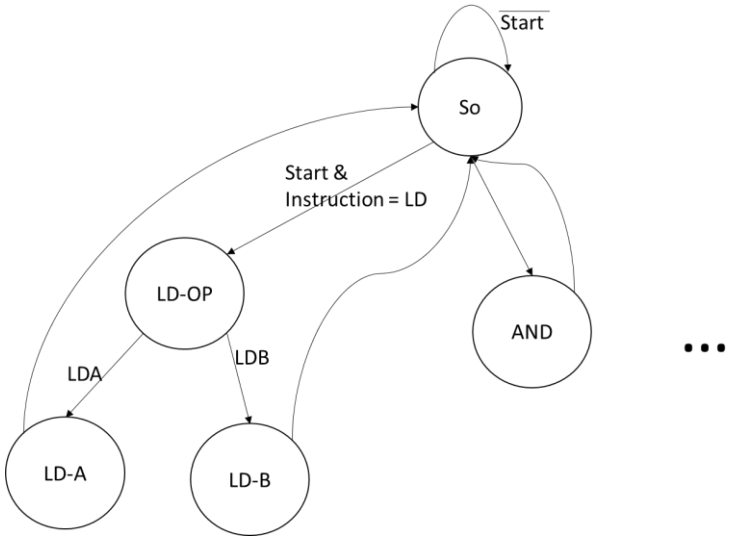

Figure 5 – Template Moore Automaton

3. Implement your controller in VHDL and connect to the datapath to form your G-CPU. Simulate your design with a couple of examples. Annotate important parts of the simulation with arrows and text. **(6 pts)**

Document your design, explain your design choices and provide results of your simulation, including all waveforms and annotations in your final report. Zip all the VHDL files and upload them with your project report.

Bonus Problem: G-CPU Controller Design (10 pts)

Map the input and output bus to the peripherals of your board (ex. SSG or LEDs and switches) to see what computation is being performed. The choice of which peripheral to use and what to show is at your own discretion. However, what you input and display must reflect the operation performed in the G-CPU. You can do the following for instance.

4. Show the value loaded after a load instruction.
5. Show the result of a computation.
6. Show the value of a register.