Laboratory 1: Full Adder

1 INTRODUCTION

One of the basic building blocks of a computer is the central processing unit [CPU]. The job of the CPU is to receive input data, perform a mathematical operation upon it, and then generate output data. Shown below are the three main components of the CPU that aid in the execution of the required mathematical operation upon the input data. The
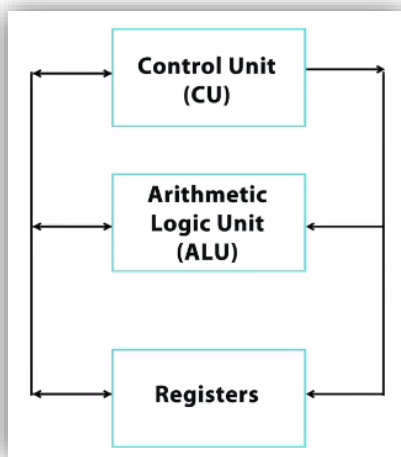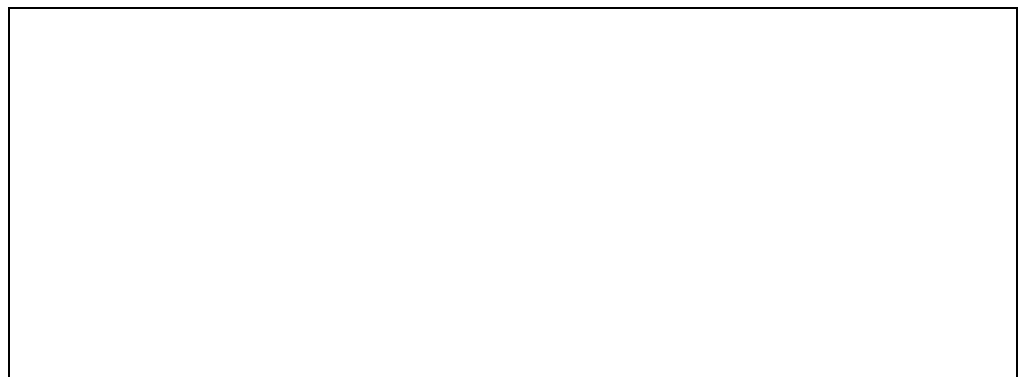

Figure 1. CPU Block Diagram.

Arithmetic Logic Unit [ALU] is essentially a calculator that performs addition, subtraction, multiplication, division, and several logic functions such as 'and', 'or', 'not', etc.. The inputs to the ALU as well as the results from the ALU are stored in the general purpose registers that are connected to the ALU. All of the coordination between the ALU and the registers is handled by the aptly named control unit [CU] which oversees all of the operations of the CPU. Combining the ALU, CU, and registers together creates a very flexible and powerful computing platform called a central processing unit [CPU] or microprocessor [if all of the functions are implemented on a single chip which is usually the case today]. The goal of this lab is to start working on ALU development by creating a basic single

bit addition circuit. This circuit will eventually be updated to perform multiple bit addition and then integrated with subtraction, multiplication, division, and logic circuits to create your very own ALU.

2 PRELAB

Fill in the below truth table for a single bit adder with carry input and carry output. Also draw the basic hardware circuit on the right that implements the truth table. Hint: see hierarchy lecture notes on MyCourses.

a	b	cin	sum	cout

3   BEHAVIORAL SINGLE BIT ADDER IMPLEMENTATION

Code up a behavioral implementation of the single bit adder hardware circuit from the prelab. Make sure to:

- ☐   Include an appropriate header and comments.
- ☐   Remove all tabs from your code.
- ☐   Use version control.
- ☐   Use scripting.
- ☐   Save this design in a folder under lab1 labeled something like adderSingleBitBehavioral.
- ☐   Simulate the design in Modelsim and obtain a signoff.

4   STRUCTURAL SINGLE BIT ADDER IMPLEMENTATION

Code up a structural/hierarchical implementation of the single bit adder hardware circuit from the prelab. Make sure to:

- ☐   Include an appropriate header and comments.
- ☐   Remove all tabs from your code.
- ☐   Use version control.
- ☐   Use scripting.
- ☐   Save this design in a folder under lab1 labeled something like adderSingleBitStructural.
- ☐   Simulate the design in Modelsim and obtain a signoff.

5   VERSION CONTROL

Paste in a screen shot of your version control commits by right clicking on your lab1 folder and then selecting 'show log'

6 DELIVERABLES

To receive full credit for this lab one must hand in the below items no later than 168 hrs [7 days] after the start of one's lab session. Signoffs can be obtained after the due date as long as the time stamp of the code is from before the deadline.

- ☐ Hard copy of this document.
- ☐ Hard copy of behavioral implementation file [no tabs and print from notepad++ with 'show symbols' on].
  - o adder_single_bit_behavioral.vhd
- ☐ Hard copy of structural implementation files [no tabs and print from notepad++ with 'show symbols' on].
  - o adder_single_bit_structural.vhd
  - o alu_or.vhd
  - o alu_xor.vhd
  - o alu_and.vhd

7 SIGNOFFS   NAME: _____

Category	Initials	Date	Points
Prelab			/10
Behavioral Demonstration			/25
Structural Demonstration			/25
Version Control			/10
Header/Comments/Tabs			/10
Deliverables			/20
Final Grade			/100