Design of an Arithmetic Logic Unit

Michael Brodskiy
Professor: S. Shazli

February 9, 2023

1 Introduction

- The purpose of this lab is to design an arithmetic-logic unit (ALU) that operates on 8-bit unsigned integers
- The ALU will support four different operations: addition, subtraction, multiplication, and division
- The results will be displayed in binary format on the LEDs, and also in decimal format on the 7-Segment displays on the DE1-SoC board

2 Lab 3 — Preparation of Logic Decisions

- In order to focus on the Schematic designs during this lab session, you need to prepare your designs on paper before implementing them
- This lab reuses some components designed in the previous labs including the 8-bit adder and the 7-Segment display with enable
- Using a circuit schematics editor to create all connections from scratch is a pretty time-consuming task, so we will be using some pre- designed versions of them existing in libraries included with the Quartus Prime software
- Specifically, we will use the Intel FPGA integer Arithmetic IP cores to perform mathematical operations
- These components are available on the IP Catalog library located on right side of the Project page as shown on Figure 1; It can also be accessed from the “Tools Menu” if you can’t find it

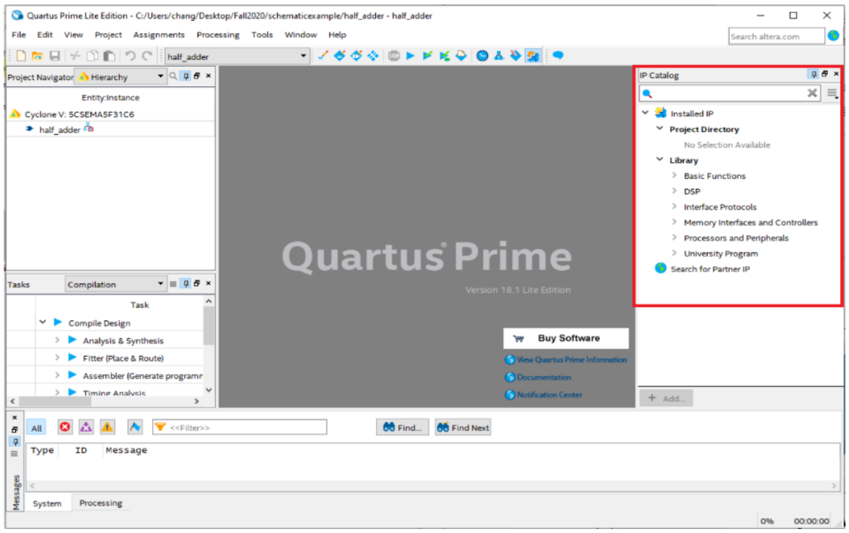

Figure 1: IP Catalog Library

- For this lab, you will need your 7-Segment display with enable files from Lab 2, and the rest of the components will come from the usual components library or the IP Catalog library

3 8-Bit Unsigned Integer on the 7-Segment Display

- The ALU design we will build operates on 8-bit unsigned integers (positive integers only) and we would like to see the results in decimal format
- That is, the result will be an integer between 0 and 255 for 8-bits. For this part, we will extend our 7-Segment display so that, if given an 8-bit integer, it will display the value (0 to 255) of the integer on up to 3 displays

4 Building the ALU Components

- The addition, subtraction, multiplication, and division operations supported by the ALU are implemented in similar LPM modules
- You have already created a division block, div8u component created in the previous section

5 The 8-Bit ALU Design

- The last step in our hardware design consists in building the complete 8-bit ALU, based on all logic blocks created in this and the previous lab, including the 8-bit adder, 8-bit subtractor, 8-bit multiplier, 8-divider, and 8-bit 4-to-1 MUX
- The following logic block shows the interface for the 8-bit ALU:

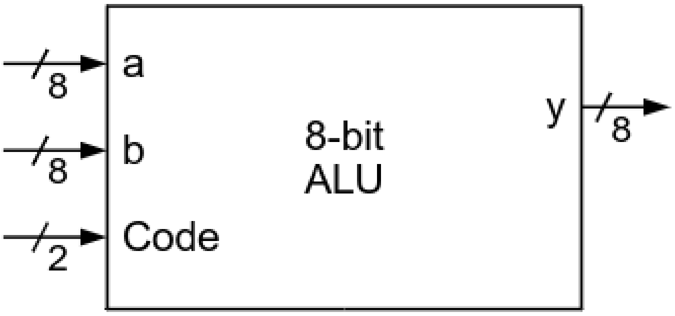

Figure 2: 8-Bit ALU Design

- As you can see, the 8-bit ALU takes two 8-bit input operands a and b, and provides its result in output y
- A 2-bit input code determines the operation performed by the ALU, hence the value of output y, as follows:
  - If code is 00, the ALU performs addition.
  - If code is 01, the ALU performs subtraction.
  - If code is 10, the ALU performs multiplication.
  - If code is 11, the ALU performs division