

Design of an Arithmetic Logic Unit (ALU)
Embedded Design: Enabling Robotics
EECE2160

Michael BRODSKIY
Brodskiy.M@Northeastern.edu

February 23, 2023

Date Performed: February 16, 2023
Partner: Dylan POWERS
Instructor: Professor SHAZLI

Abstract

The purpose of this lab was to design an arithmetic-logic unit (ALU) that operates with 8-bit unsigned integers. The ALU performs addition, subtraction, multiplication, and division with switches providing the inputs both multiplexer and binary inputs; the answer is then outputted on the 7-segment digital displays in decimal and on LEDs in binary. To accomplish this, the lab was broken into two main parts: the first was to design a logic circuit to display a given 8-bit integer on the 7-segment display, and the second was to combine the circuit from the first part with additional logic to design the ALU logic circuit.

KEYWORDS: ALU, unsigned, arithmetic, 7-segment display, logic unit

1 Equipment

Available equipment included:

- DE1-SoC board
- DE1-SoC Power Cable
- USB-A to USB-B Cable
- Computer
- Quartus Prime Schematic Software

2 Introduction

When designing increasingly complex schematic designs for digital logic circuits, there needs to be preparation ahead of time on paper or of subcomponents of the final schematic. Lab 3 used the 7-segment display schematic created in a previous lab as a component in the ALU design. The purpose of the lab was to design an arithmetic-logic unit (ALU) that operates with 8-bit unsigned integers in the Quartus Prime Software highlighting the necessity for preparation with complex logic circuit designs. The ALU performs addition, subtraction, multiplication and division on 8-bit unsigned integers and displayed results from 0-255 in decimal and binary formats. As for the inputs, the ALU received a constant ‘32’ decimal, an 8-bit number designated by the switches on the DE1-SoC board, and a control input that was always designated by the switches on the board. The constant ‘32’ and the binary 8-bit number were the two numbers undergoing the arithmetic operations, and the control input indicated what operation was to be performed.

3 Discussion & Analysis

3.1 Assignment 1

The purpose of Assignment 1 was to take the existing 7-segment display logic circuit and extend the design to display a decimal value from 0-255 on 3 displays if an 8-bit integer is given. To do this, three 7-segment displays logic blocks were utilized with one for each display. However, before the values could be displayed, the binary input needed to be converted to decimal. To convert from binary to decimal, three LPM_DIVIDE basic function blocks were used to divide the original binary number and resulting quotients by 10. The first LPM_DIVIDE block took the 8-bit input at the numerator and a contant ‘10’ as the denominator. Then, the remainder of this operation was sent to one of the 7-segment displays and the quotient was sent to the second LPM_DIVIDE function block. This was repeated twice with the exception that the quotient of the third LPM_DIVIDE function block was not connected to anything. The final logic circuit is shown in Figure 1.

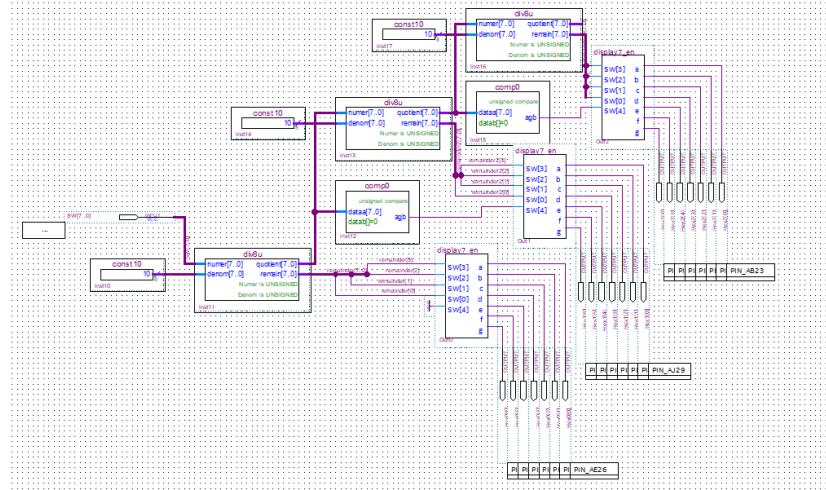


Figure 1: 8-Bit Unsigned Integer on 7-Segment Display Circuit

Unlike previous logic circuit verifications, this circuit was solely verified through uploading to the DE1-SoC board and changing the switches to provide different inputs. The results of the verification are shown in Figures 2-5.

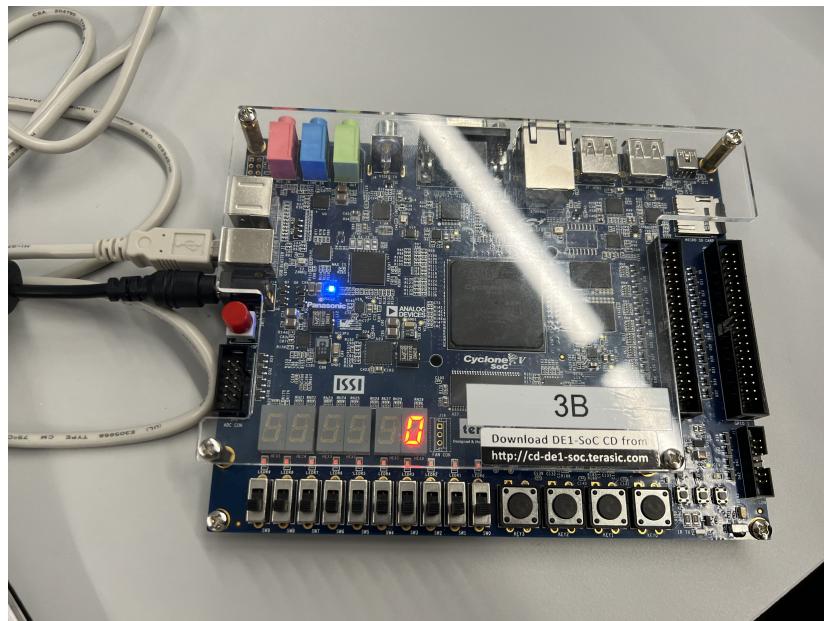


Figure 2: 00000000_2 Input on Switches, Displaying 0

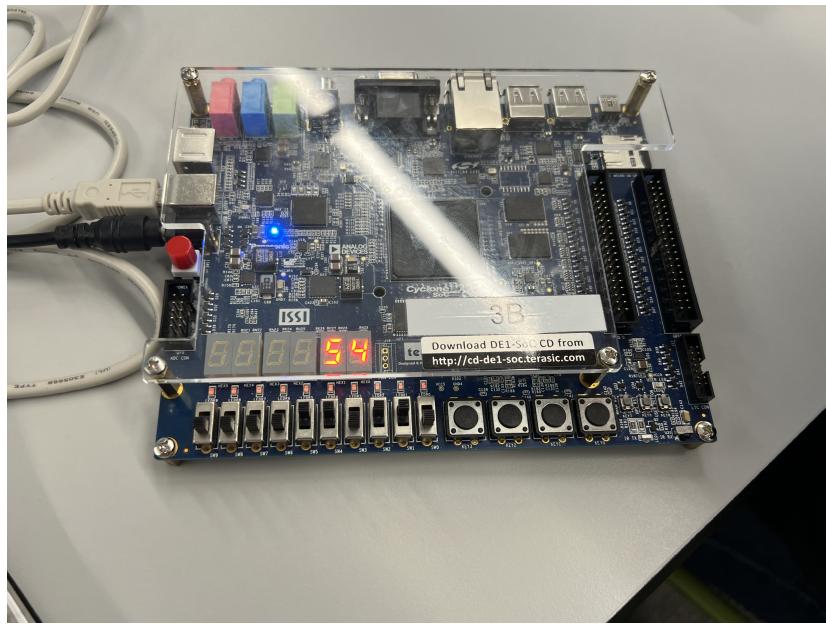


Figure 3: 00110110_2 on the Switches, Displaying 54

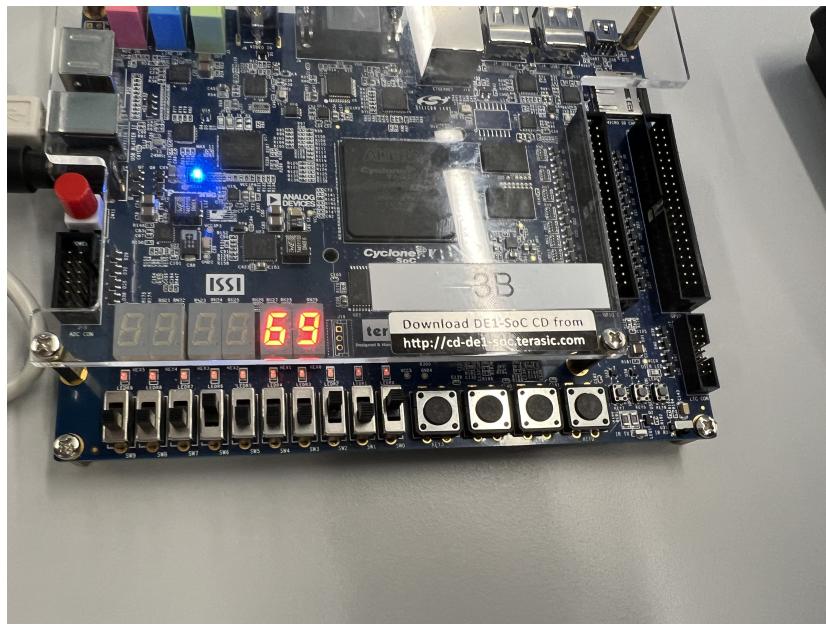


Figure 4: 01000101_2 on the Switches, Displaying 69

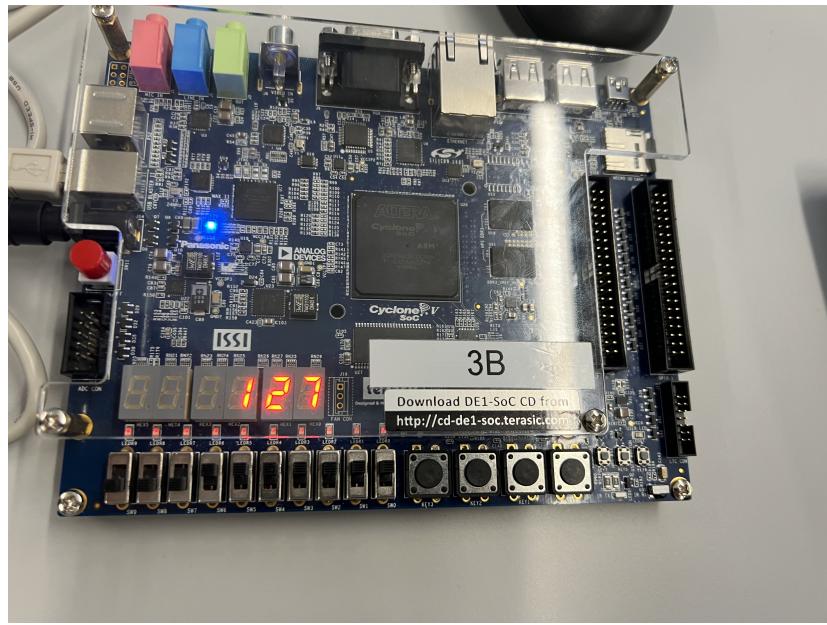


Figure 5: 0111111_2 on Switches Displaying 127

3.2 Assignment 2

The purpose of Assignment 2 was to create the ALU logic circuit with LPM modules addition, subtraction, multiplication and division. Given that there are limited switches on the DE1-SoC board, one 8-bit number was inputted to each LPM module, using SW[7..0], along with a constant decimal integer of '32.' The outputs from these LPM modules were then all fed into a single 8-bit 4 to 1 multiplexer, which served to indicate what operation was to be performed with the control inputs of SW[8] and SW[9]. For the code of the inputs, 00 designated addition, 01 designated subtraction, 10 designated multiplication and 11 designated division. The multiplexer then output an 8-bit binary, positive integer that needed to be displayed on both the 7-segment displays and the LEDs. For the 7-segment displays, the logic circuit from Assignment 1 was used. The circuit was compressed into a function block called `display_8bits` and added after the multiplexer. As for the LEDs, each LED from 0-7 was assigned a bit. If a bit was 1 then the LED would be on, but if the bit was a 0 then the LED would be off. The aforementioned schematic can be seen in Figure 6.

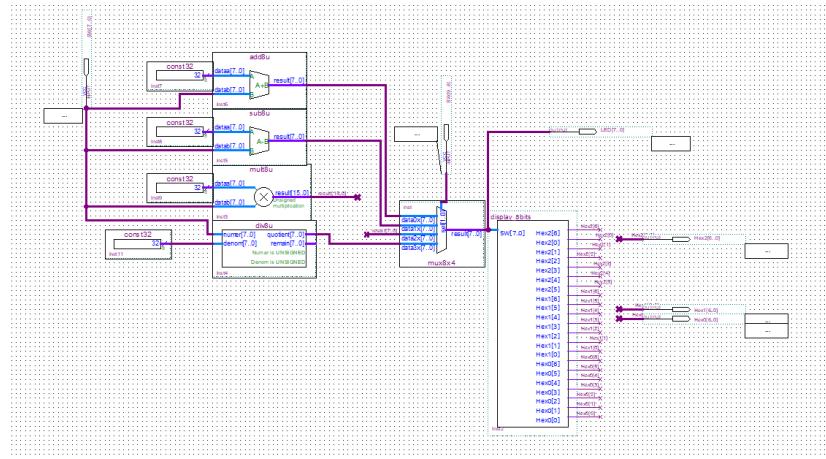


Figure 6: ALU Design Schematic

Similar to Assignment 1, this logic circuit was not simulated in Waveform Simulation. The circuit was verified through uploading it to the board and testing different inputs. All operations were tested at least twice, and photos of the tests are shown in Figures 7-18 below.

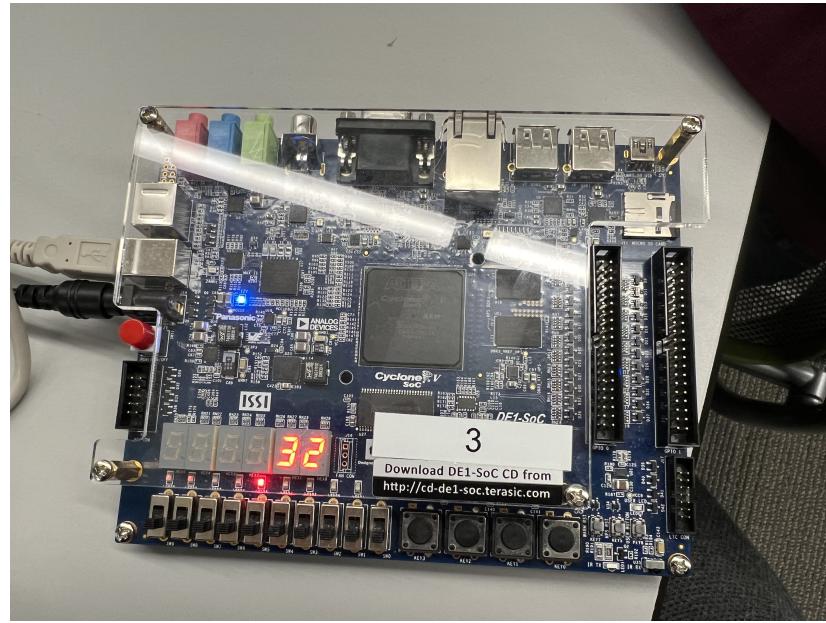


Figure 7: Addition — Constant 32 and Binary Input of 00000000_2 (0)

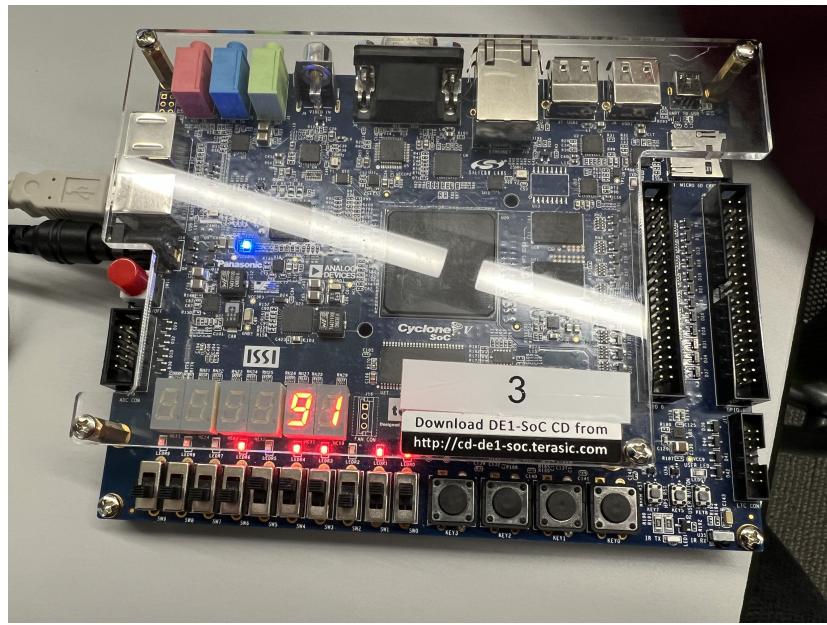


Figure 8: Addition — Constant 32 and Binary Input of 00111011_2 (59)

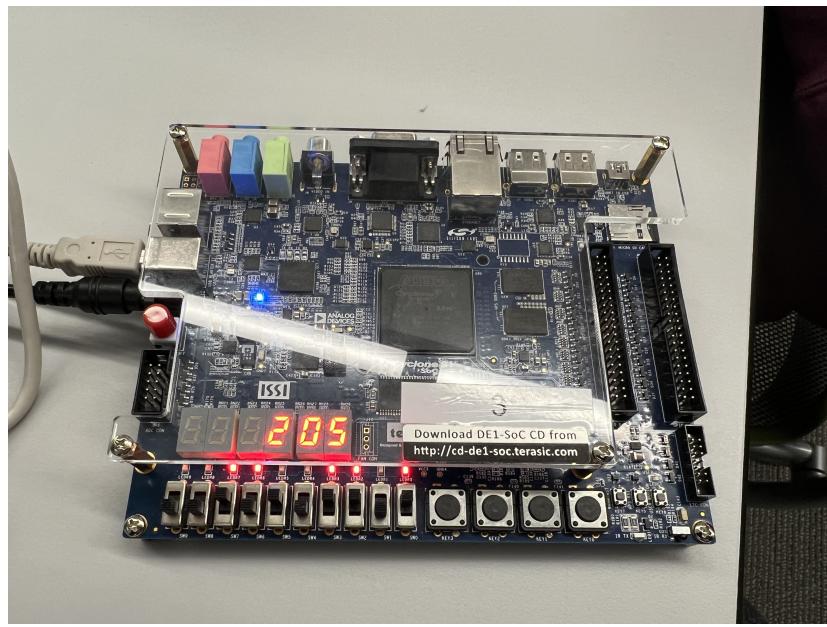


Figure 9: Addition — Constant 32 and Binary Input of 10101101_2 (173)

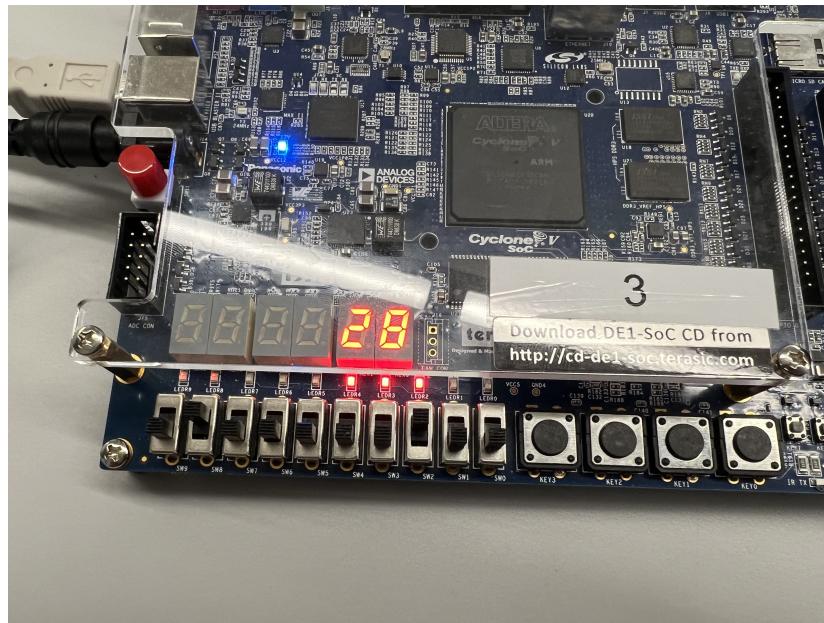


Figure 10: Subtraction — Constant 32 minus Binary Input of 00000100_2 (4)

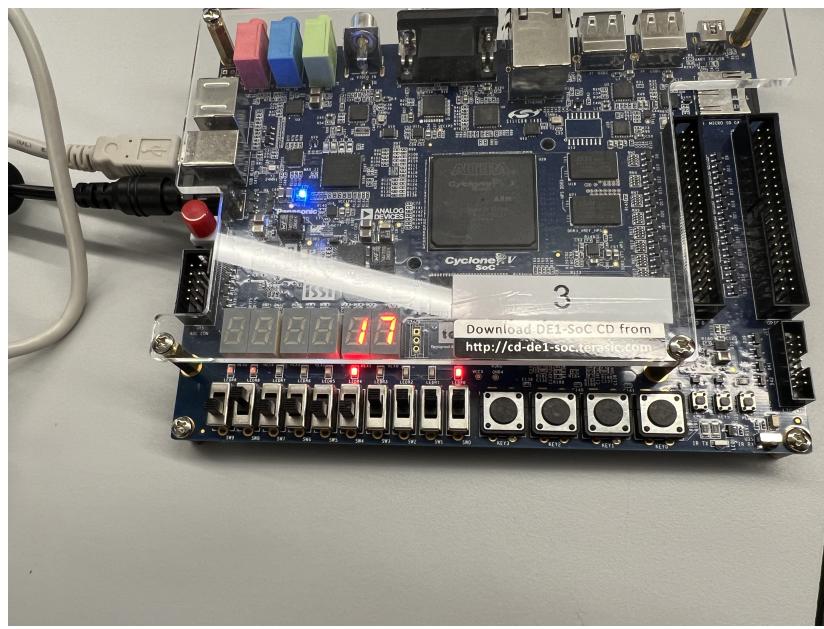


Figure 11: Subtraction — Constant 32 minus Binary Input of 00001111_2 (15)

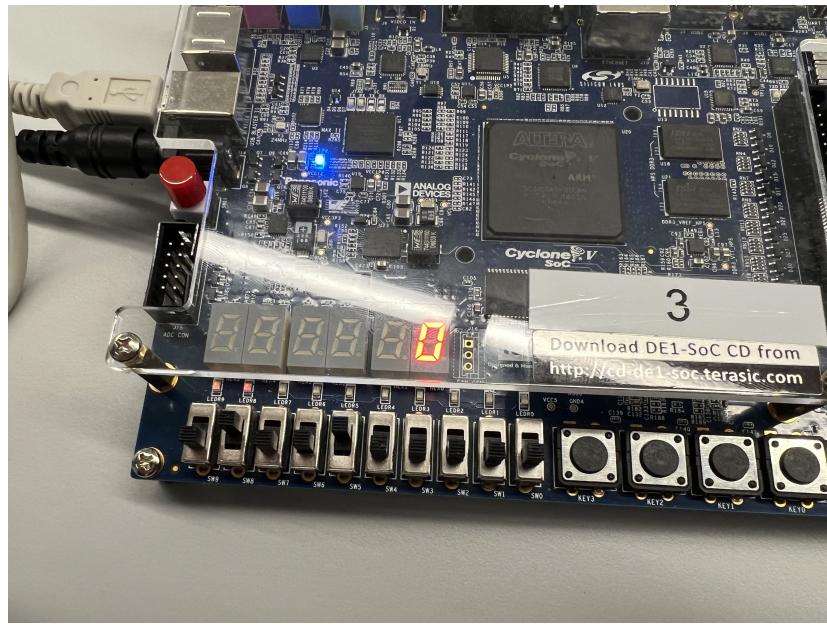


Figure 12: Subtraction — Constant 32 minus Binary Input of 00100000_2 (32)

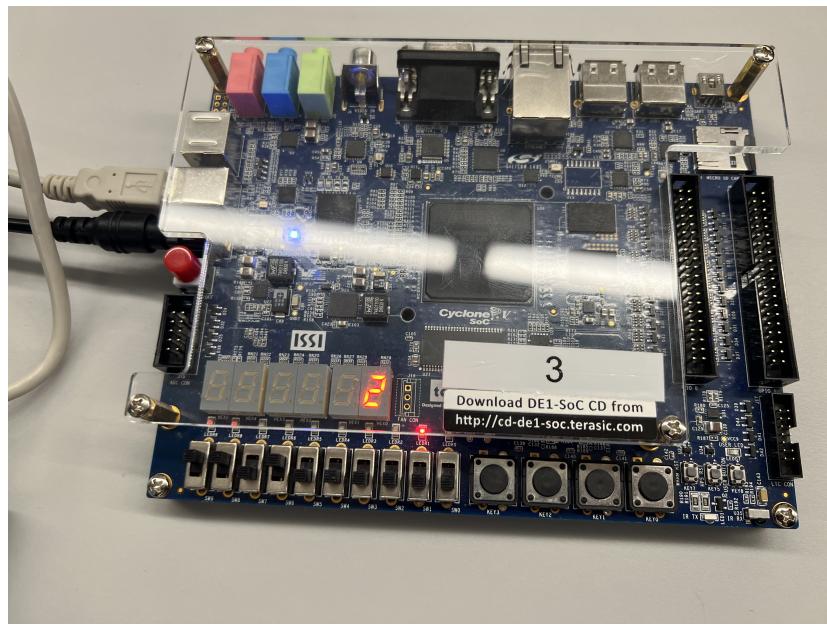


Figure 13: Division — Binary Input of 01000000_2 (64) Divided by the Constant 32

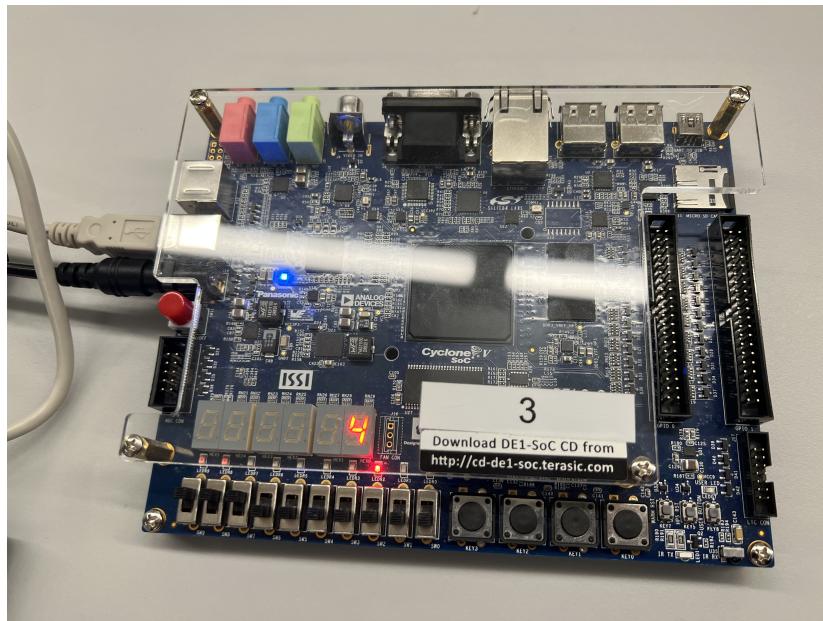


Figure 14: Division — Binary Input of 10000000_2 (128) Divided by the Constant 32

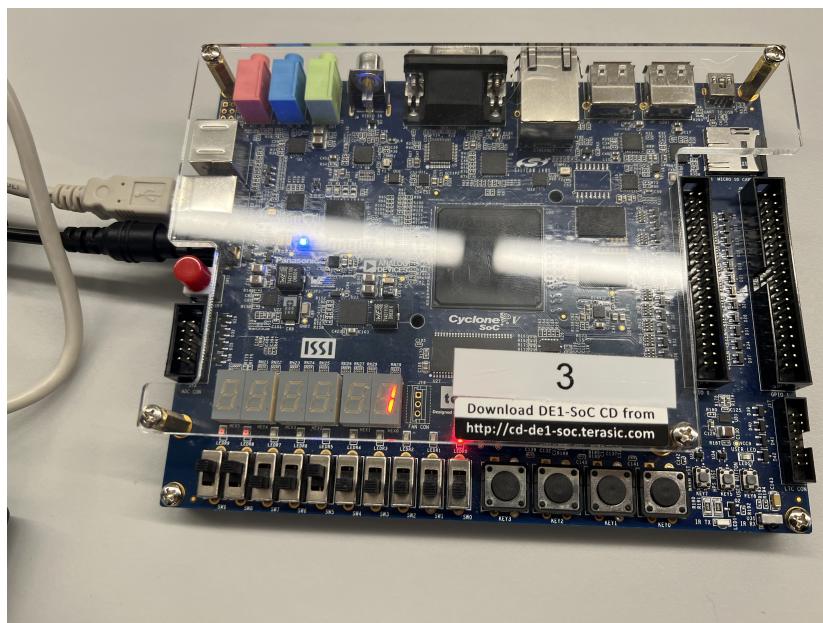


Figure 15: Division — Binary Input of 00100000_2 (32) Divided by the Constant 32

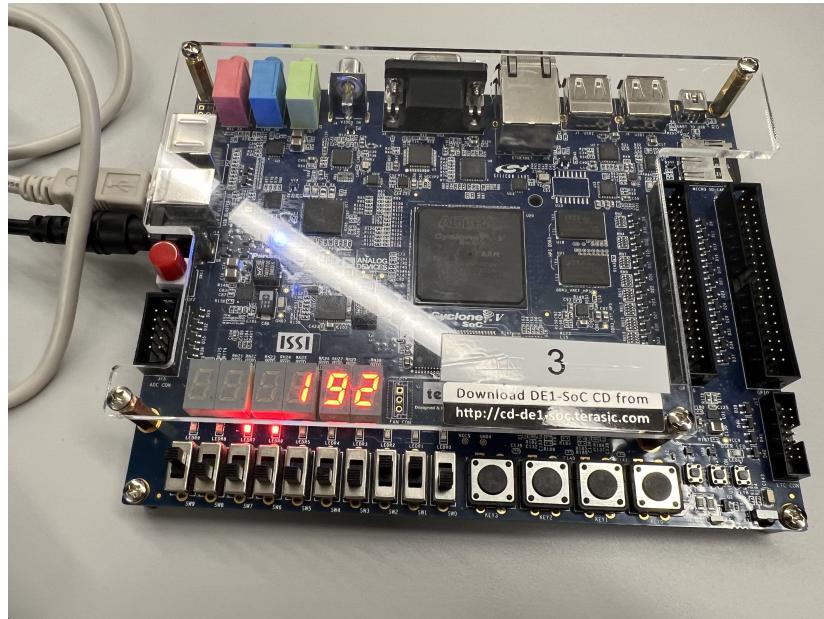


Figure 16: Multiplication — Binary Input of 00000110_2 (6) and the Constant 32

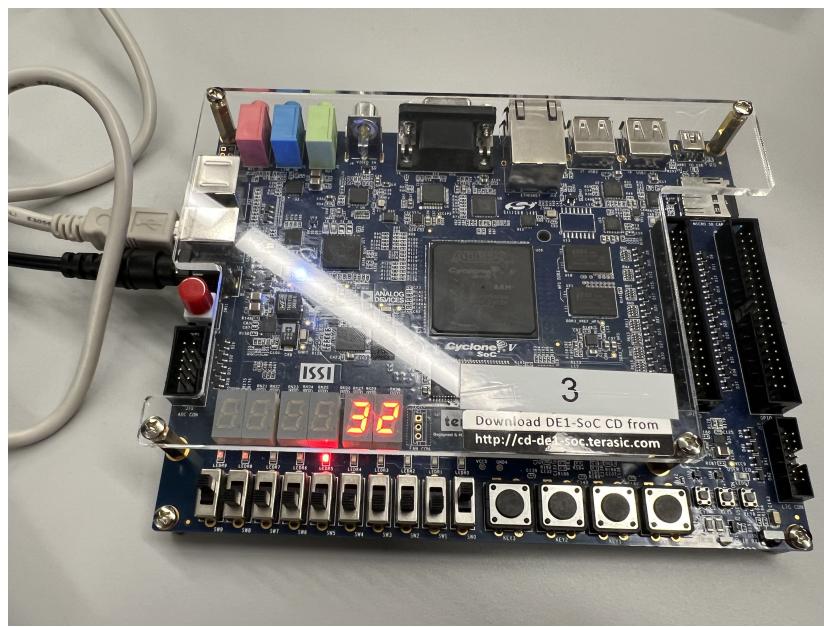


Figure 17: Multiplication — Binary Input of 00000001_2 (1) and the Constant 32

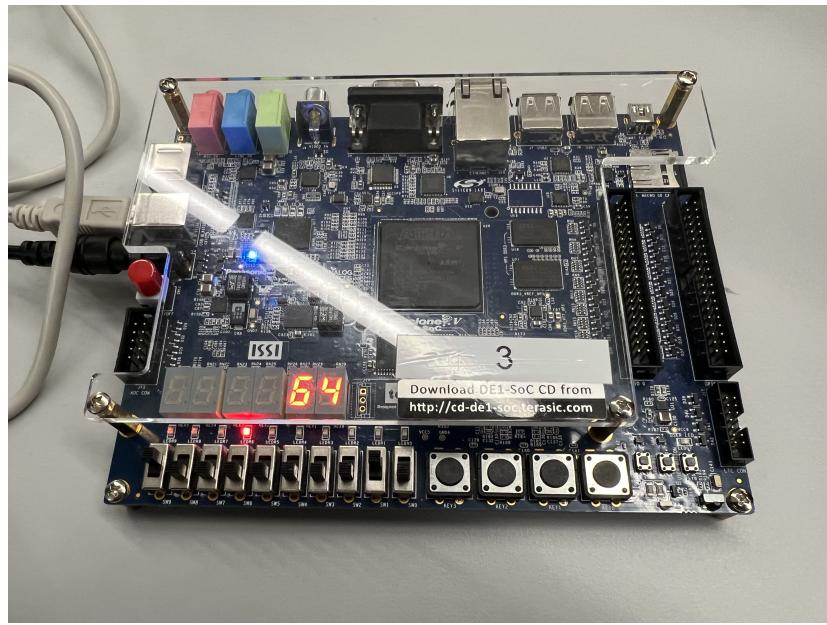


Figure 18: Multiplication — Binary Input of 00000010_2 (2) and the Constant 32

4 Conclusion

The lab resulted in the construction of two logical blocks: one that simply displayed an inputted binary number in decimal on the seven-segment displays, and another that performed arithmetic by using a control switch for a multiplexer, and then outputting to a seven-segment display, as well as binary LEDs. In this manner, an arithmetic logic unit (ALU) was constructed. Most importantly, the concepts of multiplexers was used for the first time, and allowed for an adequate introduction to multiplexers.