

Final Project Lab Report

1.0 Introduction

1.1 Problem Description

Arithmetic Logic Unit (ALU) is the foundation building block operating on integer binary numbers that constructs our Central Processing Unit (CPU), Graphics Processing Unit (GPU), and Floating-Point Unit (FPU), etc. We are building the ALU to better understand the functioning of our computers and to incorporate our knowledge of bitwise logic and logic gates. Our motivation begins with our obsession towards those tiny and micro construction of our circuits. We are interested in microchips that drives our life everywhere in the world. We hope to understand from the bottom of those computation and signal processing.

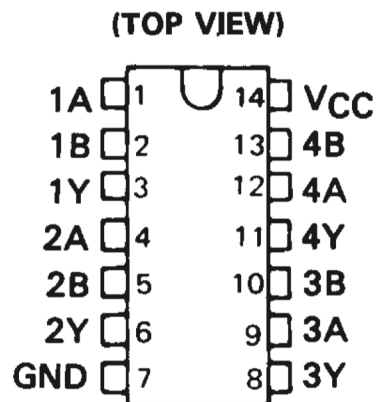
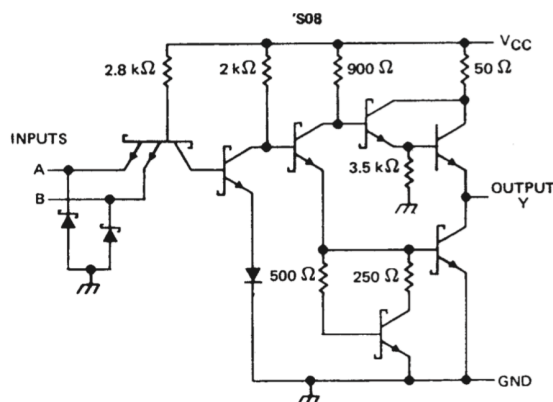
1.2 Design Concept

The ALU is able to do the basic calculation and logic calculation including addition, subtraction, AND, and OR. We implement the circuit using logic gates including AND, OR, NOT and XOR, multiplexers (MUX), and tri-state buffers. Due to limitation of breadboard area and wire length, we implement the ALU in two separate parts. There is one circuit for arithmetic calculation and the other one for logic calculation. The end of each circuit is attach to the same dip switch, so we are able to observe arithmetic calculation result and logic calculation result at the same time. To realize arithmetic calculation, including addition and subtraction, we build a full adder for each bit, and connect the carry-out of the last bit with the carry-in of the next bit. For switching between addition and subtraction, we use XOR gates and tri-state buffer. To realize logic calculation, we use AND and OR gates with a 1:2 MUX, which is able to determine a single output. After implementing logic gates and other components, the ALU is attached to 5 volts power source and output signals through LEDs.

2.0 Analysis of Components

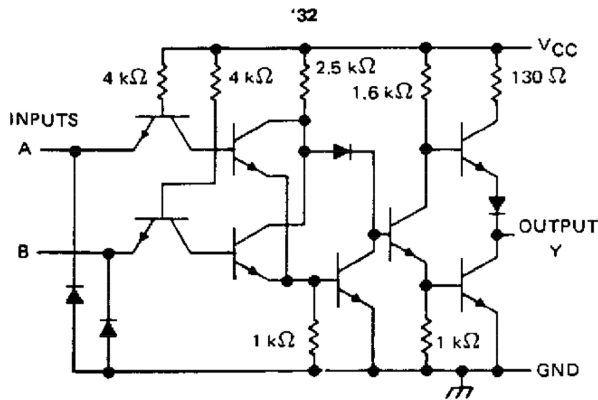
2.1 Characterization of Components

AND gate (SN74LS08N)

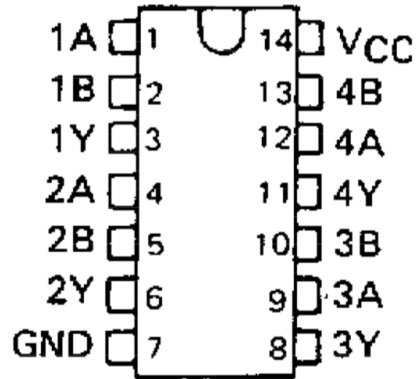


This chip contains four 2-input AND gates. The inputs are labeled with A and B, and outputs are labeled with Y. By connecting V_{cc} to 5 volts and GND to 0 volt, the chip is able to do AND calculation.

OR gate (SN7432N)

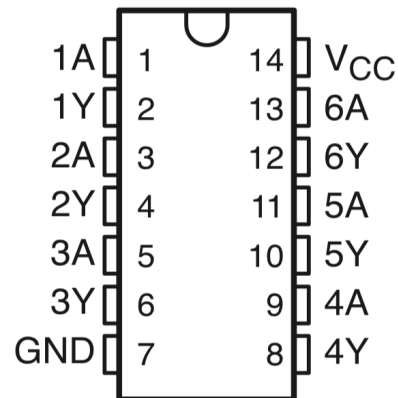
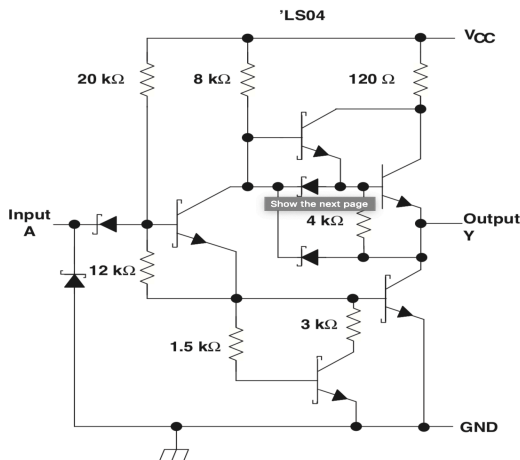


(TOP VIEW)



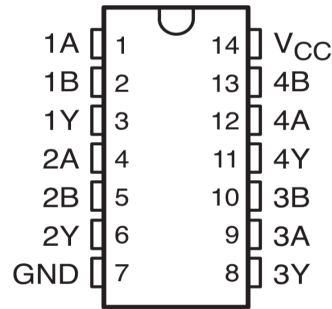
This chip contains four 2-input OR gates. The inputs are labeled with A and B, and outputs are labeled with Y. By connecting V_{cc} to 5 volts and GND to 0 volt, the chip is able to do OR calculation.

NOT gate (SN74LS04N)



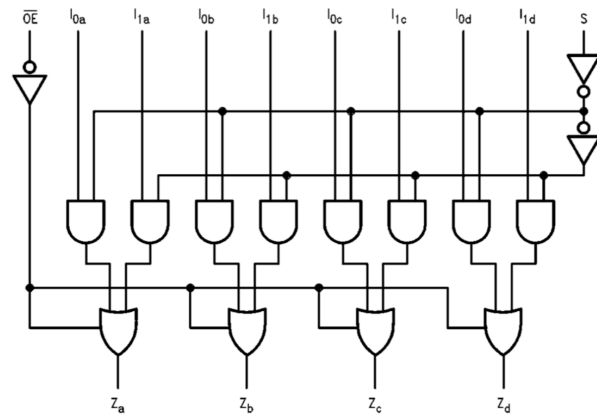
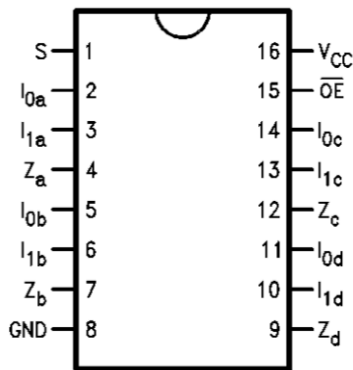
This chip contains six 2-input AND gates. The inputs are labeled with A, and outputs are labeled with Y. By connecting V_{cc} to 5 volts and GND to 0 volt, the chip is able to do NOT calculation.

XOR gate (SN74AHCT86N)



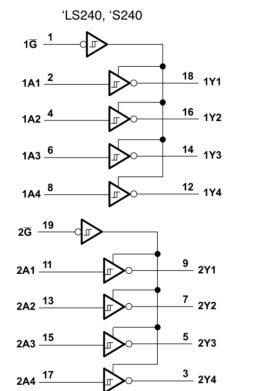
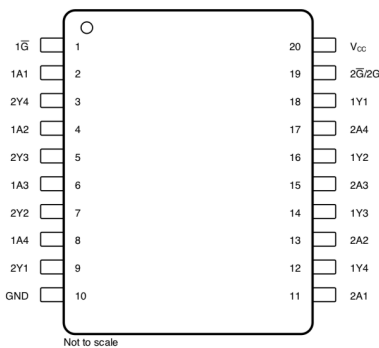
This chip contains four 2-input XOR gates. The inputs are labeled with A and B, and outputs are labeled with Y. By connecting V_{CC} to 5 volts and GND to 0 volt, the chip is able to do XOR calculation.

Multiplexer (74F257PC)



The 74F257A is has four 2-input multiplexer labeled with l_0 and l_1 with 3-state outputs labeled with z. After connecting V_{CC} to 5 volts and GND to 0 volt, the chip is able to select between two inputs by connecting OE.

Tri-state buffer



SN74LS241N can choose between selected combination of inverted and non-inverted output. It works to select between addition and subtraction.

Dip switch

Dip switches are used to provide different input values by selecting between 0 volt and 1 volt.

LED and Resistors

These components are used for illustration of bit-wise calculation.

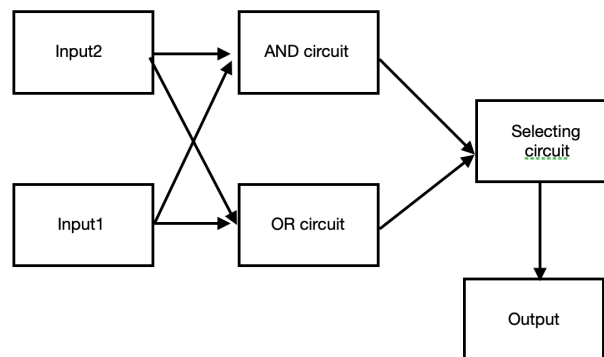
2.2 Design Analysis

After characterization of the components of the Arithmetic Logic Unit, we decide to use the full adder and extra XOR gates and tri-state buffers to construct selection process between addition and subtraction. To switch from addition to subtraction, we need to invert the second number and plus one. Hence, tri-state buffer becomes our optimal choice. Since 1:2 MUX has the ability to select between two inputs, we are able to choose between AND and OR logic calculation by providing different control signal.

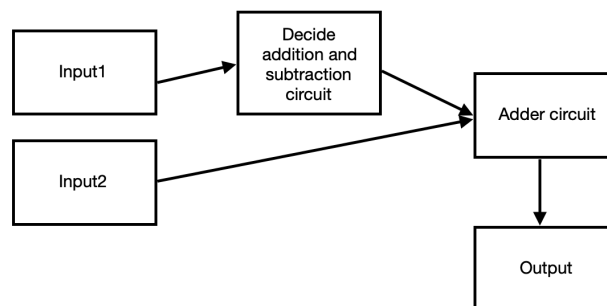
3.0 Design Description

3.1 Block Diagram

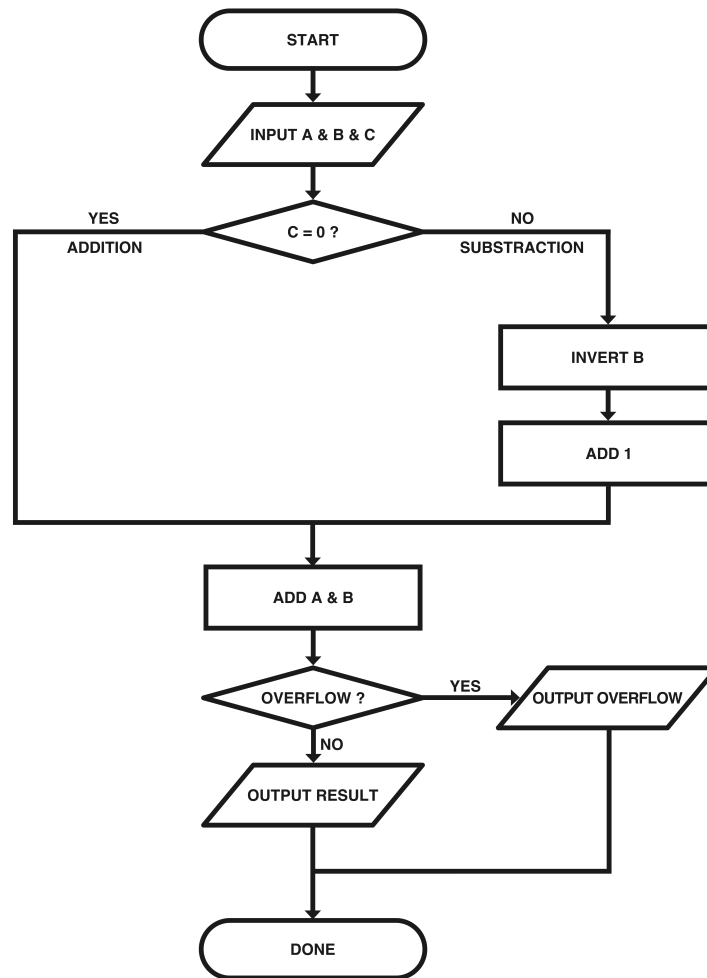
a) Block diagram for logic operation



b) Block diagram for arithmetic operation



c) Flow Chart of Addition/Subtraction

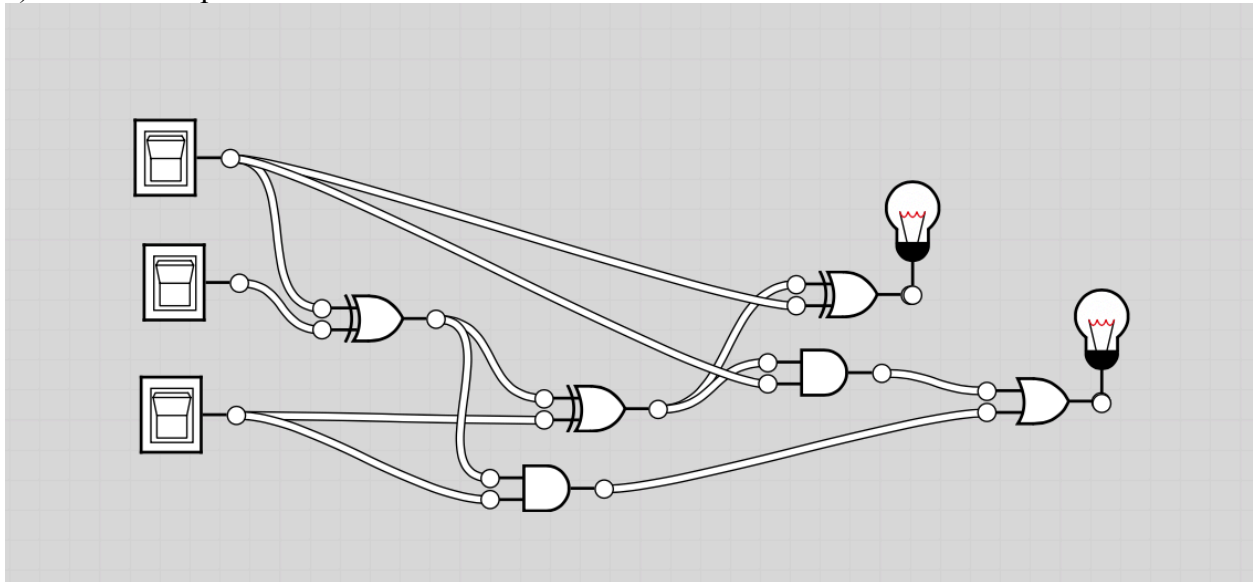


ADDITION & SUBTRACTION FLOW CHART

The above is the flowchart of addition and subtraction operation of ALU. First, we want to take in three inputs, A, B and C. Depending on the value of input C, the ALU will decide to carry out addition or abstraction. If the result's return is false, then the input B will be inverted and add 1. After that, the ALU will add two numbers together to get the final result. If the result's return is 1, the ALU will directly add two numbers together. One the ALU gets the result, it will check whether or not the result overflows. If the result overflows, it will output error, otherwise it will output the result.

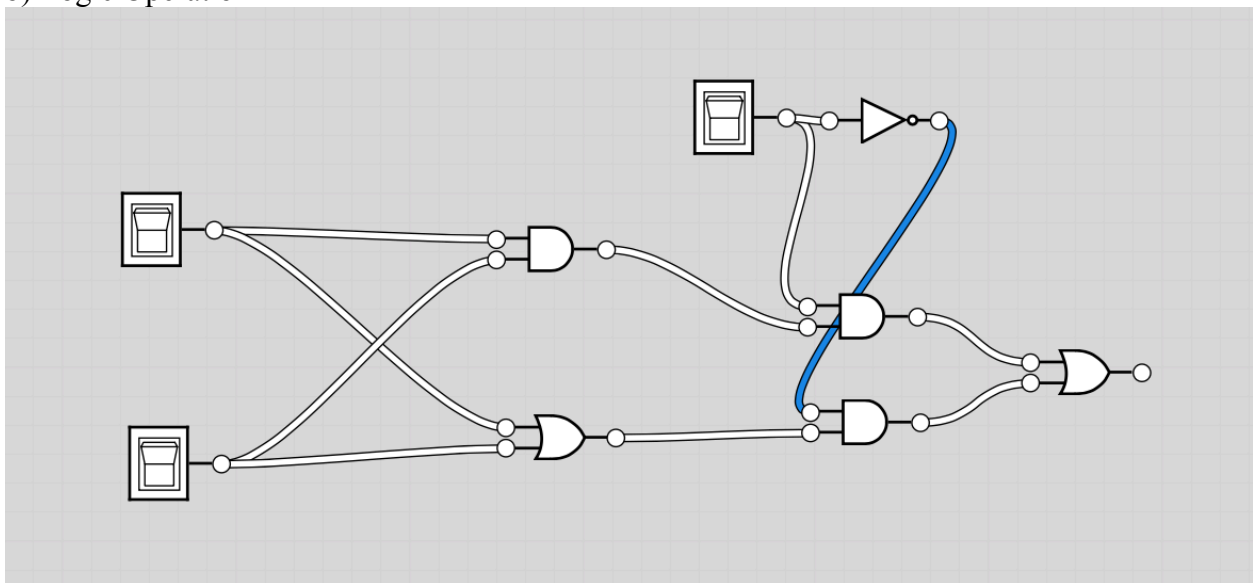
3.2 Circuit Schematics (Bitwise)

a) Arithmetic operation



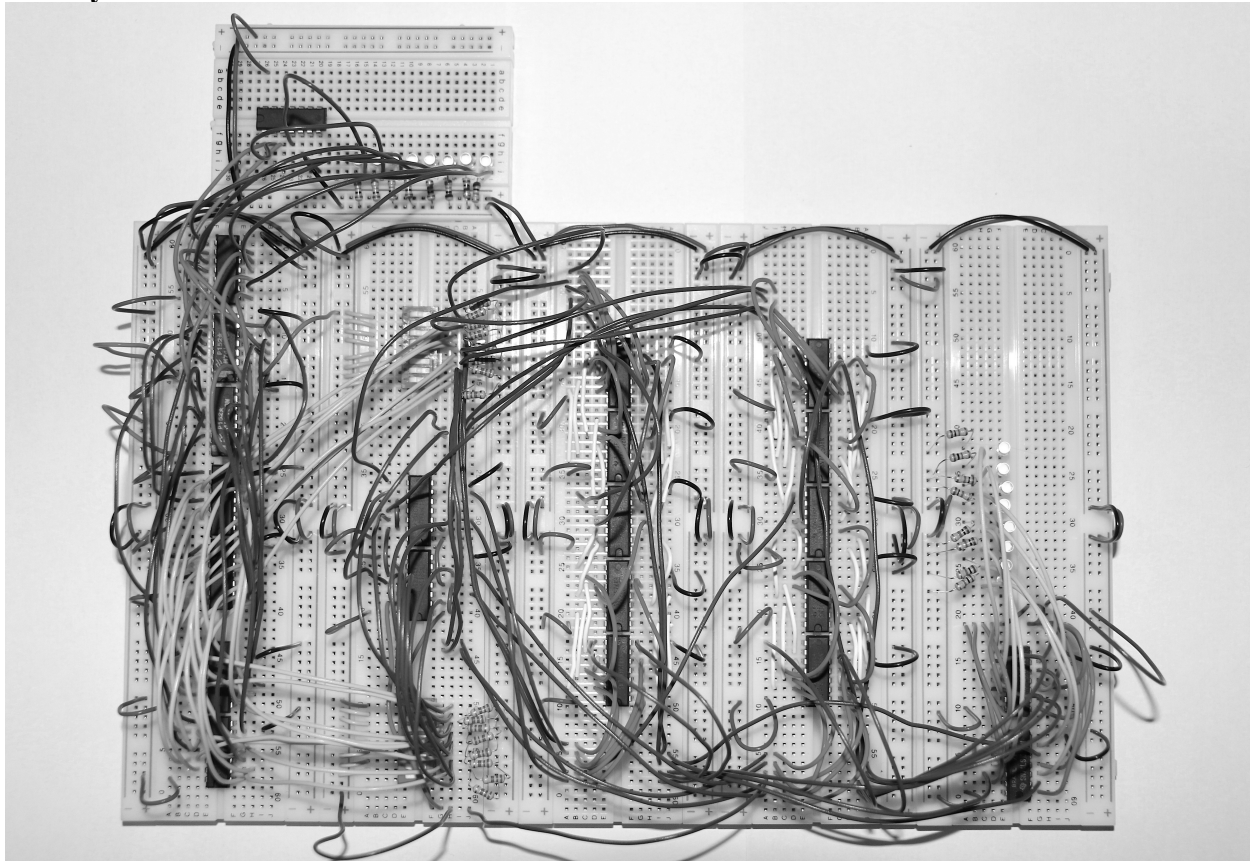
This includes 2 inputs, 1 carry in(control signal), 1 bit full adder

b) Logic Operation



This includes two inputs, 1 control signal, 1 AND gate, 1 OR gate and a 2 to 1 mux.

3.3 Physical/Mechanical Construction



(Physical Construction - 1)

4.0 Conclusion

4.1 Lessons Learned

It took us a long time as we expected to construct this arithmetic logic unit. We initially determined to build 16 bit ALU with arithmetic calculation including addition, subtraction, multiplication and division. However, bit shifting involving in 2's complement calculation was much harder than we originally imagined. Also, the cost of the breadboards was also considered to be the cause that limit the areas of our construction. Therefore, we finally chose to construct 8-bit ALU with less calculation ability instead of our original proposal.

During our construction process, we unexpectedly encountered two main problems that took us a long time for debugging. First, we did not know that the long breadboards were separated by half. It resulted that some of our LEDs did not output any signal. We originally assumed that our circuits might have some flaws or some wires might be connected to wrong pins. After checking for several times, we could not find any problem. Then, we tried to use voltmeter to detect the output voltage of each IC chip. We soon found that the V_{cc} and GND of the long breadboard were separated by half. The next problem we encountered was a broken gate in XOR chip that prevent us to get a right value. As we were debugging the circuit after fixing the breadboard problem, we then found that one of our LED was not outputting with the right value. We again used voltmeter to detect from the end of the circuit, lining up with the LED, back to the IC chip. We immediately found out the problem. We also faced some other problems like wrong wiring

when we build the circuit for addition/subtraction, and missing state signal for multiplexer. By checking the circuit with the datasheet of those components, we corrected our mistakes.

4.2 Self-assessment

Overall, our project did a great job. It can successfully carry out arithmetic functions that include addition, subtraction and logic operations that include AND and OR. There is also aspects that our project is lack of. First, when we are planning the project, we also thought about doing multiplication and division. However, our project cannot carry out this two operations because of these two operations are too more difficult to implement in just one semester and includes some knowledge that we haven't know yet. Therefore, our project cannot do multiplication and division. For the logic operation parts, our project that besides AND and OR including XOR .etc.

Reference

- [1]R. Mitchell, "How to Build Your Own Discrete 4-Bit ALU", *Allaboutcircuits.com*, 2016. [Online]. Available: <https://www.allaboutcircuits.com/projects/how-to-build-your-own-discrete-4-bit-alu/>. [Accessed: 27- Jan- 2019].
- [2]"Lecture 8: Binary Multiplication & Division."
- [3]"Lecture5 Multiplication and Division."
- [4]B. Easter, *ALU Design*. youtube, 2016.