4-Bit Arithmatic Unit Design

Asif Mohaimen, Asif Mohaimen, Asif Mohaimen

Course Teacher
Is not my concern

Digital Logic Design
Dept. of Computer Science & Engineering
North South University
December 2016

Abstract

The task was to create an arithmetic logic unit that will compute four operations in general. They are completed in one circuit using the Logisim simulation tool by Dr. Carl Burch

Contents

List of Figures

List of Tables

1 Introduction

1.1 Arithmetic Logic Unit

An arithmetic logic unit (ALU) is a combinational digital electronic circuit that performs arithmetic and bitwise operations on integer binary numbers.

The inputs to an ALU are the data to be operated on, called operands, and a code indicating the operation to be performed and, optionally, status information from a previous operation; the ALU's output is the result of the performed operation. In many designs, the ALU also exchanges additional information with a status register, which relates to the result of the current or previous operations.

1.2 Performed Operations

1.2.1 Arithmetic Operations

- **Add:** A, B and Carry_{in} are summed and the sum appears at Output and $\text{Carry}_{\text{out}}$.
- **Subtract:** B is subtracted from A and the difference appears at Output and $\text{Carry}_{\text{out}}$. For this function, $\text{Carry}_{\text{out}}$ is effectively a "borrow" indicator.
- **Increment:** A is increased by one and the resulting value appears at output.
- **Decrement:** A is decreased by one and the resulting value appears at output.

1.2.2 Logical Operations

- **AND:** the bitwise AND of A and B appears at output.
- **OR:** the bitwise OR of A and B appears at output.
- **Exclusive-OR:** the bitwise XOR of A and B appears at output.
- **Invert:** all bits of A (or B) are inverted and appear at output.

2 Methodology

2.1 Full Adder Design

For the completion of the project we had to design a full adder at first. For this we have used two XOR Gates, two AND gates and one OR gate. The diagram was like it is shown in figure 2.1.

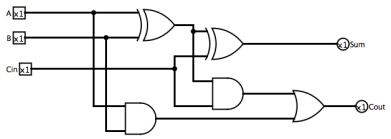

Figure 2.1: Full Adder diagram

The truth table for this full adder is shown in table 2.1. The SUM is created just

Table 2.1: Truth Table for Full Adder

Inputs			**Outputs**	
A	**B**	**Cin**	**Sum**	**Cout**
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

by using the boolean equations shown below:

$$S = A \oplus B \oplus C$$

$$C_{out} = (A \bullet B) + (C_{in} \bullet (A \oplus B))$$

2.2 4 bit adder

After creating the Full adder We have just concatenated four full adders together to produce an output of the summation of a 4-bit number. The logic circuit for this adder is shown in Figure 2.2

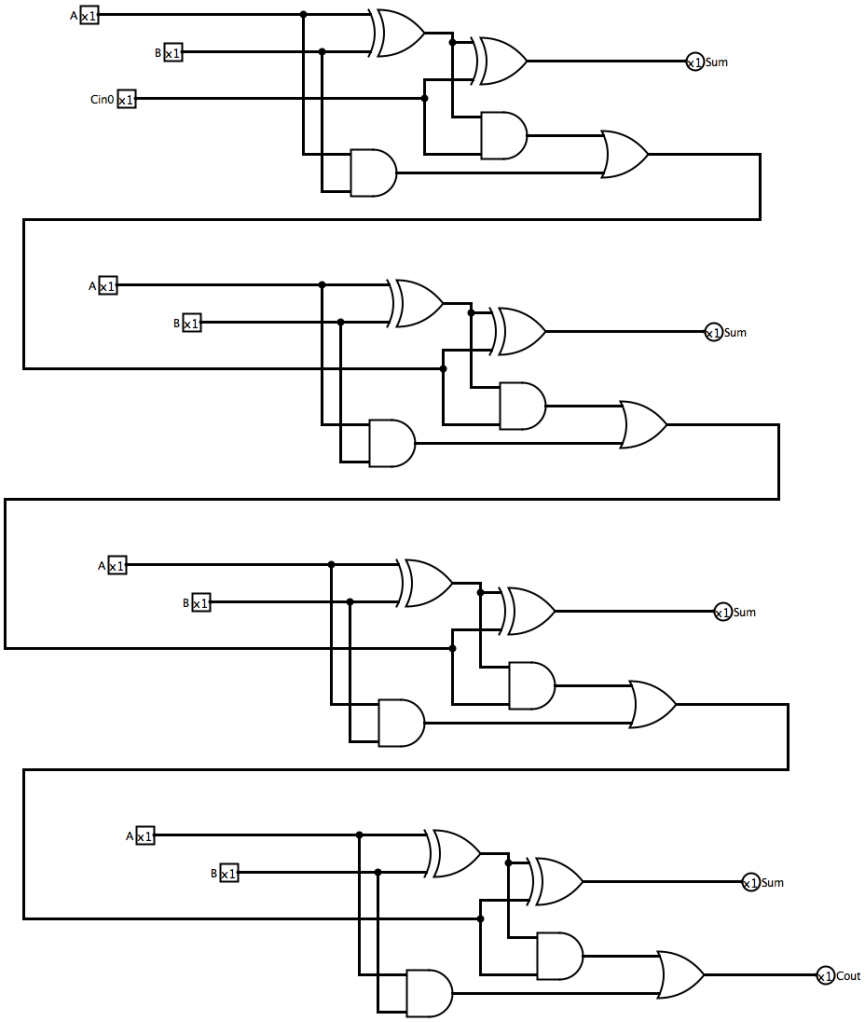

Figure 2.2: 4 bit Full Adder diagram

This module will solely serve the purpose of the addition operation.

2.2.1 4 bit Subtraction

For binary subtraction we have used the most significantly used method for subtraction. Suppose we have to subtract B from A then we have taken the 2's compliment[1] of B and added it with A using the 4 bit Adder we have previously designed before. Our 2's complement module worked like it is shown in figure 2.3. The input number will be given by B pins and output of the 2's compliment of B will be received from the O pins. After that the O pins will be connected to the B inputs of the 4 bit adder to get the final subtraction result. C_{out} pin will show the borrow output in the 4 bit adder.

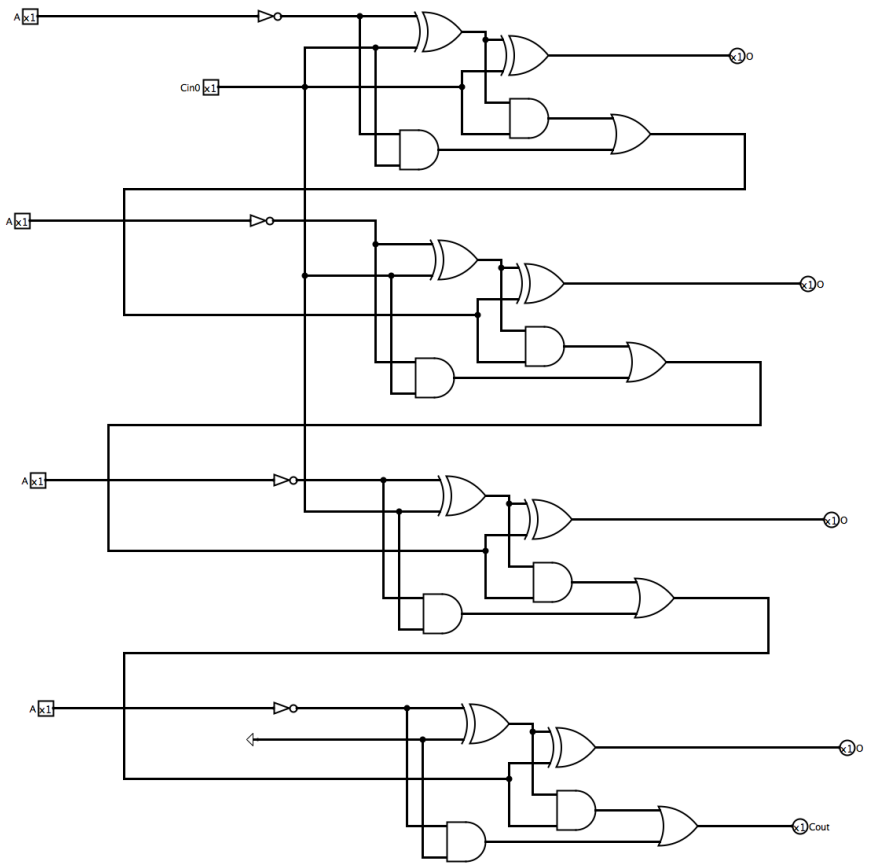

Figure 2.3: 2's compliment of B

[1]Inverting all the bits of a number and increment it by 1

2.3 Increment & Decrement

For the increment & decrement we have just used the two previously built 4 bit adder & subtractor modules. To increment the number by 1 we have just sent the input for B as 0001 to increase or decrease the input number in A by 1.

2.4 Putting it altogether

2.4.1 DEMUX

As we have to put it altogether in the end so we had to create a DEMUX for routing the inputs to separate modules. The smallest DEMUX unit is shown in figure 2.4. This is repeated 8 times to create separate output lines for the 4 modules. The truth table for this small unit goes like it is shown in table 2.2.

Table 2.2: Truth table for DEMUX

Input		Selected Output
A	B	
0	0	Sum
0	1	Subtract
1	0	Increment
1	1	Decrement

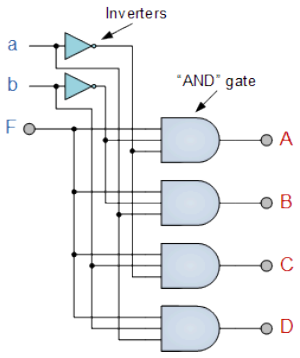

Figure 2.4: DEMUX

2.4.2 MUX

After computing the outputs we also had to display the output in one single output pin. So that we had to use the multiplexer module to perform this task. One smallest unit of the multiplexer is shown in Figure 2.5. There are 5 of them to display the 5 different outputs in 5 separate pins. Which are O_1, O_2, O_3, O_4, and $Carry_{out}/Borrow_{out}$ respectively. The truth table will be the same as it is shown in table 2.2.

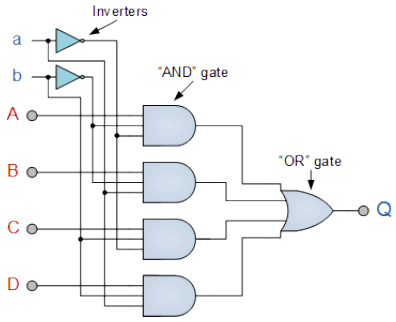

Figure 2.5: MUX

3 Summary

So basically what we have done here was, we have taken two lines for selecting the operation to be performed which is shown in table 2.2. After selecting the operation we have sent our numbers using the input pins and after performing the necessary operations inside the ALU we finally had our output from out 5 output pins.

The End