Design of a 4-bit ALU
Using VHDL
CENG 3511

Mike Moore

August 31, 2014

Date Performed:	November 17 2013
Instructor:	Dr. George Collins

Abstract

The objective of the project is to implement a four-bit arithmetic logic unit for a processor. The basic requirements include the following fundamental operations: addition, increment, decrement, transfer, and the basic logical operations (AND, OR, NOT, XOR). A behavioral approach is taken to implement this ALU. The VHDL defines inputs for two four bit registers, a four bit op-code field, and a single carry in bit. The outputs are the four bit data out, and the single carry out bit. From a block diagram of the ALU design, a VHDL module is developed to satisfy the design requirements. This VHDL module is then put under test and the results are examined.

Contents

1 Introduction

An Arithmetic Logic Unit (ALU) is a fundamental component within the central processing unit (CPU). It is responsible for both the arithmetic and logical operations of the CPU, and as such, it is responsible for a large part of the CPU work load. Modern ALU designs can be quite complex, and often they are not actually a single unit within the CPU. For simplicity, this report focuses on a simplified four bit ALU design.

2 Preliminary Design

The design begins with an evaluation of a list of design requirements. The requirements include a table of operations that the ALU must implement. From this table, a set of validation criteria can be derived. This validation criteria directly leads to a set of tests that can be used to validate that the ALU performs all of the desired functions.

2.1 ALU Requirements

The ALU was designed to meet the following list of requirements:

- The unit has two 4-bit inputs, register A and register B.
- The unit must be able to add, increment, decrement, transfer and also perform basic logic operations AND, OR, NOT, XOR.
- The unit will have 4-bit select line called Operation Select, which would direct the unit as to which operation to perform.
- In addition the unit must also be able to do shift operations (right shift, left shift etc).
- The unit has a Carry-in and also has Carry-out.

We see from the above requirements list that we will need to design an ALU that has a total of thirteen input bits. Eight will be used for the two four bit operand registers, and another four will be used for the op-code. Finally, a single carry in bit is also required. The outputs are a little bit simpler. We will only need four bits for the data out, and a single bit for the carry out.

2.2 ALU Block Diagram

With this understanding of our requirements, the next step is to depict the ALU in a functional block diagram form. This will later allow for us to more easily translate the preliminary design to a VHDL module. The block diagram description of our ALU is provided below. Note how the inputs and outputs directly map to statements in the list of requirements.

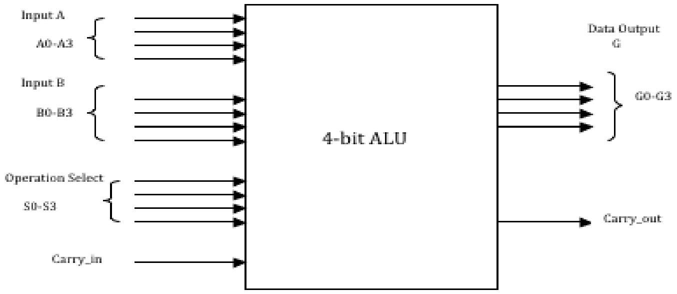
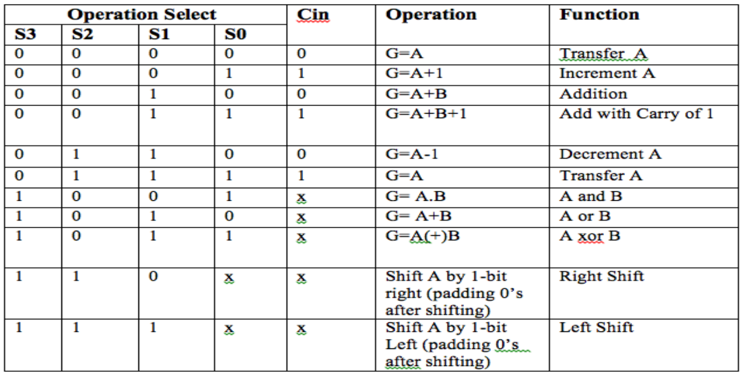


Figure 1: Four Bit ALU Block Diagram

2.3 ALU Operations

The required operations for the ALU are listed below.



Operation Select				Cin	Operation	Function
S3	S2	S1	S0			
0	0	0	0	0	G=A	Transfer A
0	0	0	1	1	G=A+1	Increment A
0	0	1	0	0	G=A+B	Addition
0	0	1	1	1	G=A+B+1	Add with Carry of 1
0	1	1	0	0	G=A-1	Decrement A
0	1	1	1	1	G=A	Transfer A
1	0	0	1	x	G= A.B	A and B
1	0	1	0	x	G= A+B	A or B
1	0	1	1	x	G=A(+)B	A xor B
1	1	0	x	x	Shift A by 1-bit right (padding 0's after shifting)	Right Shift
1	1	1	x	x	Shift A by 1-bit Left (padding 0's after shifting)	Left Shift

Figure 2: Four Bit ALU Required Operations