EGC221: Digital Logic Lab

Experiment #7

Arithmetic Logic Unit (ALU)

Schematic Implementation

Student's Name:	Reg. no.:
Student's Name:	Reg. no.:
Semester: Spring 2017	Date: 04 April 2017

Assessment:

Assessment Point	**Weight**	**Grade**
Methodology and correctness of results		
Discussion of results		
Participation		
Assessment Points' Grade:		

Comments:

Experiment #7:

Arithmetic Logic Unit

Objectives:

The objective of this lab is:

1. To design a 4-bit ALU
2. To implement the ALU on an (Altera) FPGA Development Board
3. To verify the operation of the ALU through simulation
4. To experimentally check the operation of the ALU

This lab is different from the other assignments in the sense that it is a design project which gives you more freedom to come up with your own solutions. The following write-up serves as a guideline to help you design the lab. However, if you find more efficient or more elegant ways to implement parts of the ALU, go ahead. Just make sure you justify your design and explain it clearly in the lab report write-up.

Pre-lab assignment:

a. Problem Statement:

An Arithmetic and Logic Unit (ALU) is a combinational circuit that performs arithmetic and logic micro-operations on a pair of n-bit operands (e.g., A[3:0] and B[3:0]). The operations performed by an ALU are controlled by a set of function-select inputs. In this lab you will design a 4-bit ALU with 3 function-select inputs: Selects: S0, S1, and Mode: M. The functions performed by the ALU are specified in Table 1.

Table 1: Functions of ALU

Logic				
M	S1	S0	FUNCTION	OPERATION (bit wise)
0	0	0	A · B	AND
0	0	1	A + B	OR
0	1	0	A $\oplus$ B	XOR
0	1	1	A’	NOT
Arithmetic				
M	S1	S0	FUNCTION	OPERATION
1	0	0	A + B	Addition
1	0	1	A - B	Subtract
1	1	0	A + 1	Increment
1	1	1	A - 1	Decrement

Besides the functions, you are required to provide status bits (flags) that indicate whether or not certain conditions have taken place following an arithmetic or logic operation. For arithmetic operations you need flags for carry (C), negative (N), zero (Z), and overflow (V) conditions. For logic operations you need the carry (C), negative (N), and zero (Z) flags.

A block diagram is given in Figure 1.

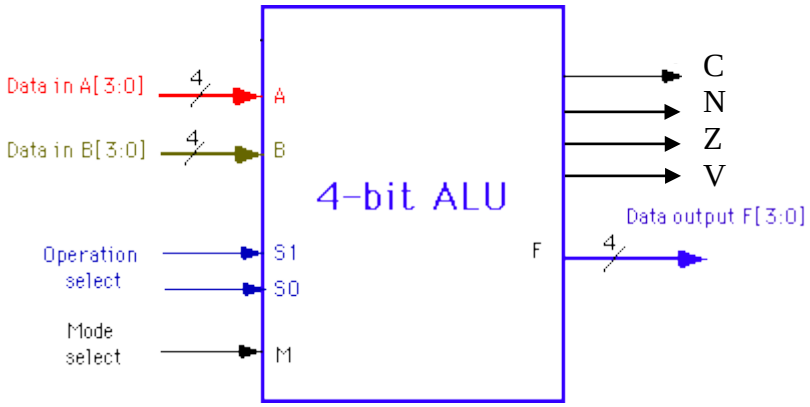

Figure 1: Block diagram of the 4-bit ALU.

When doing arithmetic, we need to decide how to represent negative numbers. As is commonly done in digital systems, negative numbers are represented in twos complement. This has a number of advantages over the sign and magnitude representation such as easy addition or subtraction of mixed positive and negative numbers. Also, the number zero has a unique representation in twos complement. The twos complement of an n-bit number N is defined as:

$2^n - N = (2^n - 1 - N) + 1$

The last representation gives us an easy way to find twos complement: take the bit wise complement of the number and add 1 to it. As an example, to represent the number -5, we take twos complement of 5 (=0101) as follows,

```
  5   0 1 0 1   -->     1 0 1 0  (bit wise complement)
                      +       1
                      ---------
                        1 0 1 1  (twos complement)
```

Numbers represented in twos complement lie within the range $-(2^{n-1})$ to $+(2^{n-1} - 1)$. For a 4-bit number this means that the number is in the range -8 to +7. There is a potential problem we still need to be aware of when working with two's complement, i.e. over- and underflow as is illustrated in the example below,

```
          0 1 0 0    (=carry Ci)
+5          0 1 0 1
+4    +     0 1 0 0
+9      0   1 0 0 1     = -7!
```

also,

```
          1 0 0 0    (=carry Ci)
-7          1 0 0 1
-2      +   1 1 1 0
-9       ± 0 1 1 1     = +7!
```

Both calculations give the wrong results (-7 instead of +9 or +7 instead of -9) which is caused by the fact that the result +9 or -9 is out of the allowable range for a 4-bit twos complement number. Whenever the result is larger than +7 or smaller than -8 there is an overflow or underflow and the result of the addition or subtraction is wrong. Overflow and underflow can be easily detected when the carry out of the most significant stage (i.e. C_4) is different from the carry out of the previous stage (i.e., C_3).

b. Possible design strategies

When designing the ALU we will follow the principle "Divide and Conquer" in order to use a modular design that consists of smaller, more manageable modules. Instead of designing the 4-bit ALU as one circuit we can first design a one-bit ALU, also called a ***bit-slice***. These bit-slices can then be put together to make a 4-bit ALU.

There are different ways to design a bit-slice of the ALU. One method consists of writing the truth table for the one bit ALU. This table has 6 inputs (M, S1, S0, C_0, A_i and B_i) and two outputs F_i and C_{i+1}. This can be done but may be tedious when it has to be done by hand.

An alternative way is to split the ALU into two modules, one “Logic” and one “Arithmetic” module. A possible block diagram of the ALU is shown in Figure 2. It consists of three modules: 2:1 MUX, a Logic unit and an Arithmetic unit.

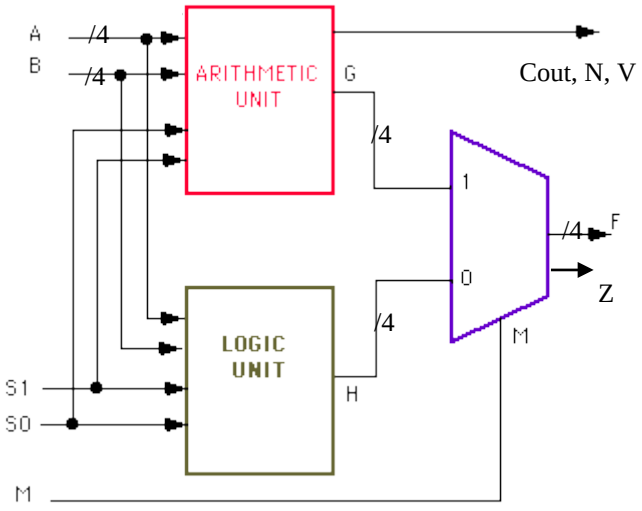

Figure 2: Block diagram of the ALU

This design methodology creates separate 4-bit units for each function. This method would utilize the 4-bit adder/subtractor that you have created in a previous lab, leaving you with some modifications (for increment and decrement) as well as some new designs (for your logic functions.) The 8 functions would then be selected with a 4-bit wide, 8:1 MUX, therefore directing the desired function to the (4-bit) output.

Note: This lab is worth 20 points

Hand-in

You have to hand in a lab report that contains the following:

1. Course Title, Lab number and title, your name(s) and date (You can use the cover sheet provided with this handout.)

2. Theory of topic (ALU)

3. Section on the Pre-lab explaining the detailed design of each macro and its simulation waveform.

4. Conclusion and overall discussion.

The lab report is an important part of the laboratory. Write it carefully, be clear and well organized. It is the only way to convey that you did a great job in the lab. It is necessary that you use computer tools (MS Word, Visio, Altera, etc.) to document the entire lab report.