**Computer science department, Langara college**

**Digital Systems Design (with FPGAs)**

**Spring 2019**

**Lab’s soft deadline on March 11th**

# Lab 5: ALU and Datapath

by S. Arash Sheikholeslam

In this lab, you will build your own simple ALU. In part 2 of the lab you will build a onetime pad cryptography device which employs your ALU.

**Preparation:**

The following must be done before you come into the lab:

1. Read this handout before you start the lab.
2. Be ready to answer questions at the end of the lab.
3. Each part of the lab is 5 points.

Part 1 [5 points]

Write an 8-bit ALU code which gets the two inputs (A and B) and a selector signal and two outputs consisting of an 8-bit output and a one bit Carry. Your ALU should perform the following signed operations:

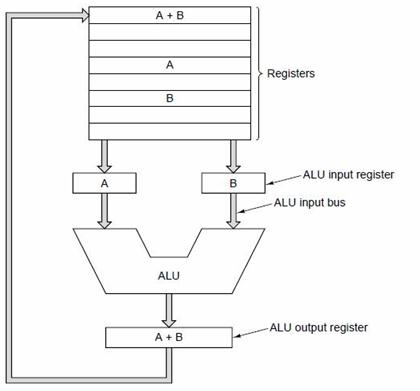
1. A+B and carry
2. A-B and carry
3. A\*B
4. A/B
5. Shift A one bit to the left without roleover, and the most significant bit should go to carry.
6. A and B
7. A or B
8. A xor B

Carry indicates overflow.

To demonstrate the performance of your ALU you need to hookup your input to a state machine.

Since you only have 10 switches on the board, you will use sw(9 downto 7) for your selector and ‘0’ & sw(6 downto 0) as your input. Use a 3 state, state machine to get the inputs A and B and show them on the LED. Show me the block diagram of this state machine before coding it in to your LAB5A.

The following image shows how an ALU appears within a simple processor:



Part 2 [5 points]

In this part we will do something interesting with our ALU. Our ALU is able to perform the xor operation. This enables the CPU that will employ this ALU to have native cryptography capability. We will use this simple function to build an 8-bit **onetime pad** encryption device. Read more on one time pad here (and we will discuss it as well)

<https://en.wikipedia.org/wiki/One-time_pad>

You will be given the code for a ROM that will contain the keys for 4 rounds of 8-bit one time pad and you will build the circuit that employs your ALU’s xor function to perform the one time pad on an 8-bit input sw(7 downto 0) and show the output on LEDR. For each new input you read a new key from the ROM and encrypt the new message until you run out of keys at which point your device is going to start again from the first key.

Note that you decrypt the encrypted message by adding the key to it one more time. How should we change our circuit and inputs such that given an encrypted message, the circuit knows which key to apply to decrypt it?

**Performance in the lab: (10 marks)**

In this lab, you will download the circuit you designed in the Preparation on the FPGA board. The input and output pins of this FPGA are tied to the various lights and switches.

1. Read the manual carefully

1. BE SURE that you set the pin assignments before compiling your design (not doing so could damage the board!). If you have any questions about how to do this, please talk to the TA.

You must demonstrate that your circuit works to the TA or to me by the end of your lab section.

*Marking:*

Your mark for the performance part of the lab will be:

0/10: If you don’t even show up, or if you show up and don’t do anything

4/10: If you make an attempt, but really don’t get anywhere near it working

7/10: If you almost get it working, or if you get it working but can’t answer TA’s questions.

10/10: If you successfully demonstrate your design to the TA, and can answer TA’s questions.