

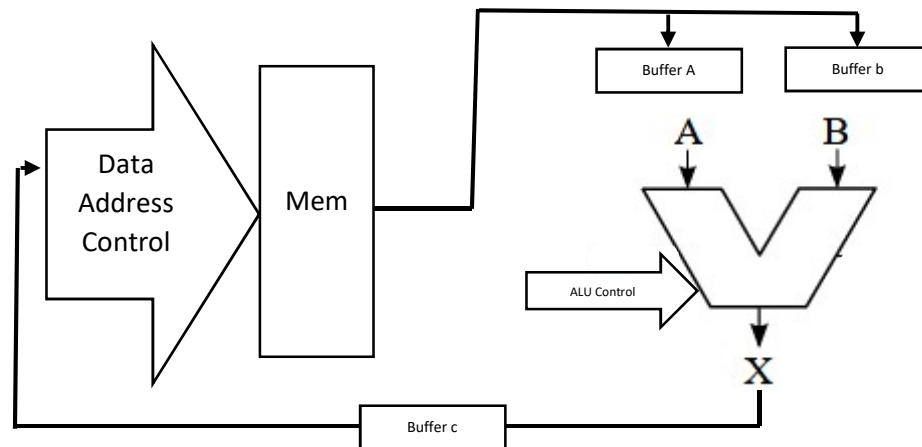
## Cosc 2P12 Assignment 2

(Due date for assignment is Monday Mar 4<sup>th</sup> 4:00 p.m. est)

In this assignment you will be creating a small 4 bit computer. The majority of this computer will have already been completed during lab. If you have not completed the labs, then I suggest you do so.

Create a new Logic Circuit project. You can import components from your lab project(s) using the import function, located under File. At the very least you will need the ALU and Memory. Other components as required.

Consider the following block diagram:



This is the basic block diagram for most simple computers. Typically one would have data stored in memory. As an instruction is decoded, the operands from memory are moved to the appropriate input buffers for the ALU. Once there, ALU function control is specified allowing the ALU to do its work. The result of the ALU computation is temporarily stored in buffer c, prior to being written back to memory.

Your assignment will be to implement this basic system using logic circuit and the components you have built so far. You are free to add any other components you feel you need. All control signals will be manual inputs. That is ALU control will be a sensor. Buffer A, B, and C will have manual write enable buttons. Memory control will have a sensor to Address the memory (see lab 4), sensor for Data input; and controls to write, and enable the outputs.

The goal is for you to be able to place 2 operands into memory, and move them through the computer manually to produce a result. The result is stored back into memory.

Here is the tricky part, to write the ALU result back into memory, you will need the data-in to switch between the sensor input and the buffer c output. Think tri-state!

Don't fret, you will have buttons and toggles all over the place in order to control the CPU. Normally this would be done with control logic governed by the opcode decode and timed with a clock pulse, but that is another course. For now, you will be the control logic.

To test your implementation consider the following.

- Store 6 in memory word 0
- Store 4 in memory word 1
- Move memory word 0 to buffer b
- Move memory word 1 to buffer a
- ALU function Subtract
- Move ALU output to buffer c
- Move buffer c to memory word 2

You can place probes on the buses to see the data as it is moved through the CPU. In fact, you should do this so the marker can see a running implementation.

Does this sound like an awful torturous assignment? No, the fact is with the lab material, you should be able to complete this in under 2 hrs, for the average student. Don't start it the night before unless you are confident you are average or above.

### **Submission**

This submission will be submitted electronically as a logic circuit project. For completeness, you can print the logic circuit – circuits and include them in a pdf as well. Better too much than not enough. **Clearly** label the parts of the project so the TA knows what to look for. Use text boxes within each cct to identify the cct. See labs.

Within your project, be sure to add foot notes in appropriate areas so that the TA can run your mini computer. Make the marker happy!!!

For the electronic submission, use Brightspace, an assignment 2 submission will be available.

- Printout of your logic circuit diagrams, with proper identification on each circuit printout. See lab 1. These will be pdf documents.
- Logic Circuit project with clear labelling of parts, runnable. The TA will run your project to assign marks.
- Package above into a zip file and submit.

The End