# 4CS015 – Workshop #5 TO BE SUBMITTED

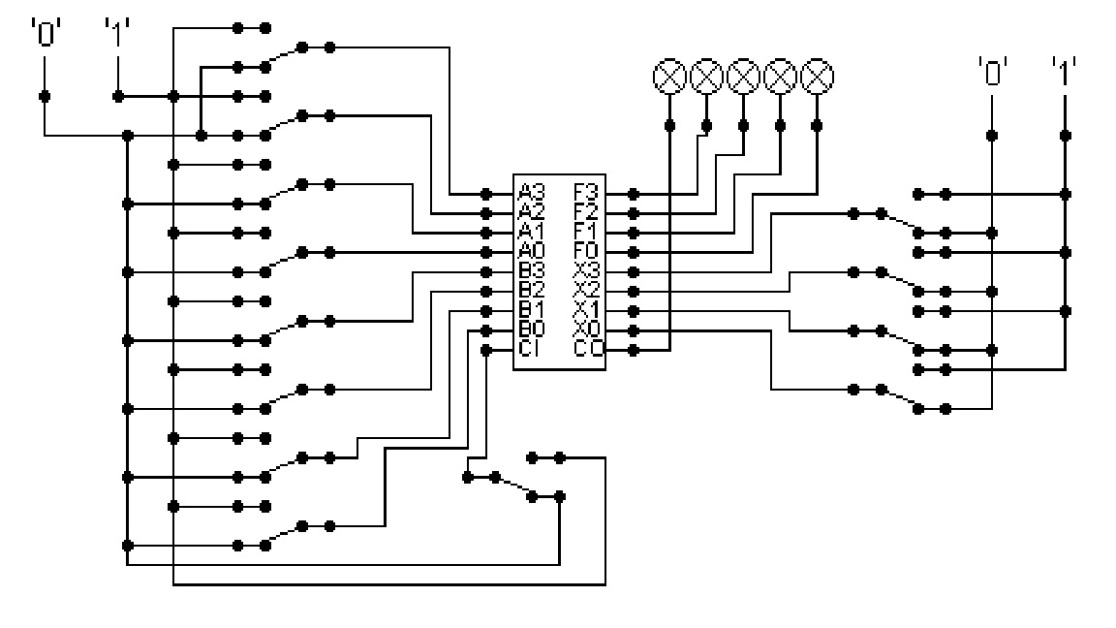
Name: Mansoud Mansouri

Student ID: 1916829

This is a marked workshop. It forms the second part of your portfolio. You will need to complete the workshop and then submit a copy of this document with a title that follows the following format (“DENNETT 1234567 wsp5.docx”), via CANVAS, by the deadline.

**Workshop tasks:**

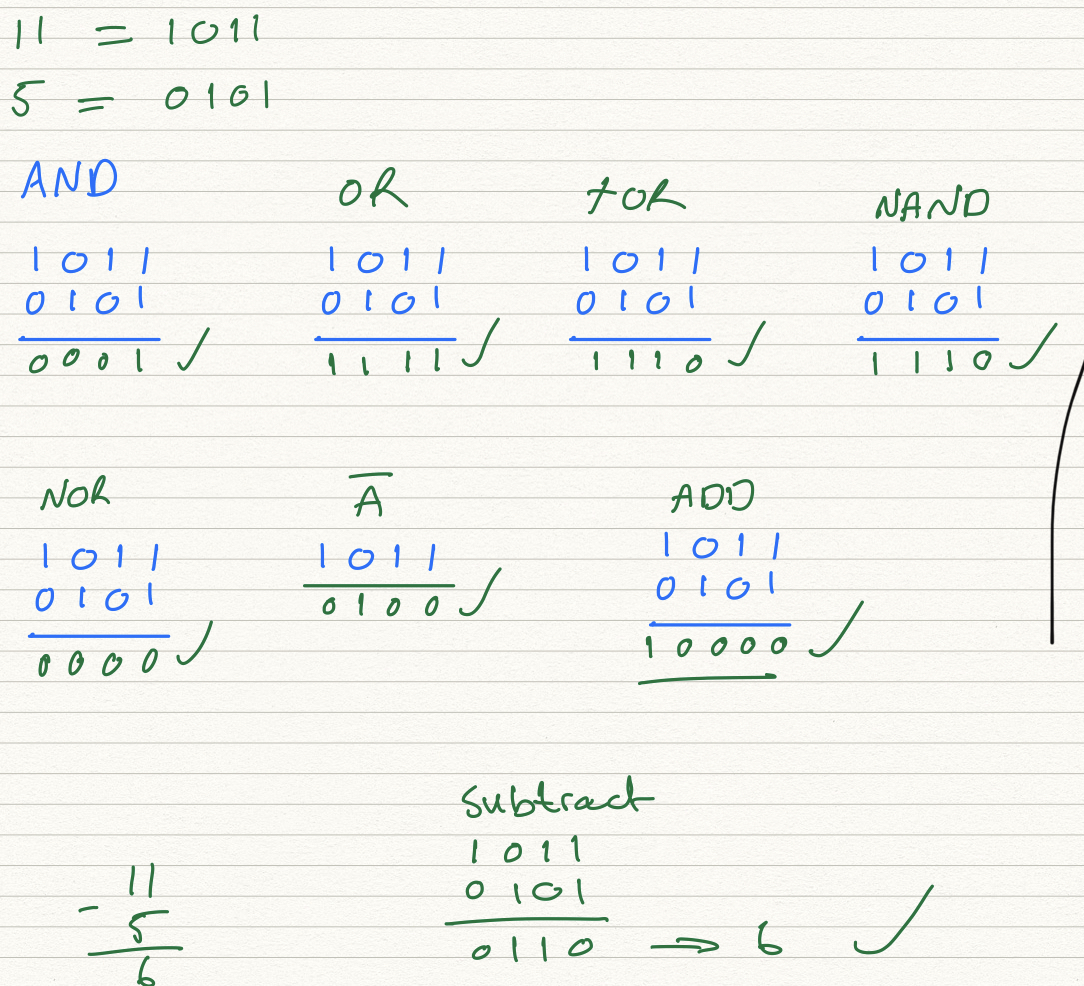
Arithmetic Logic Unit

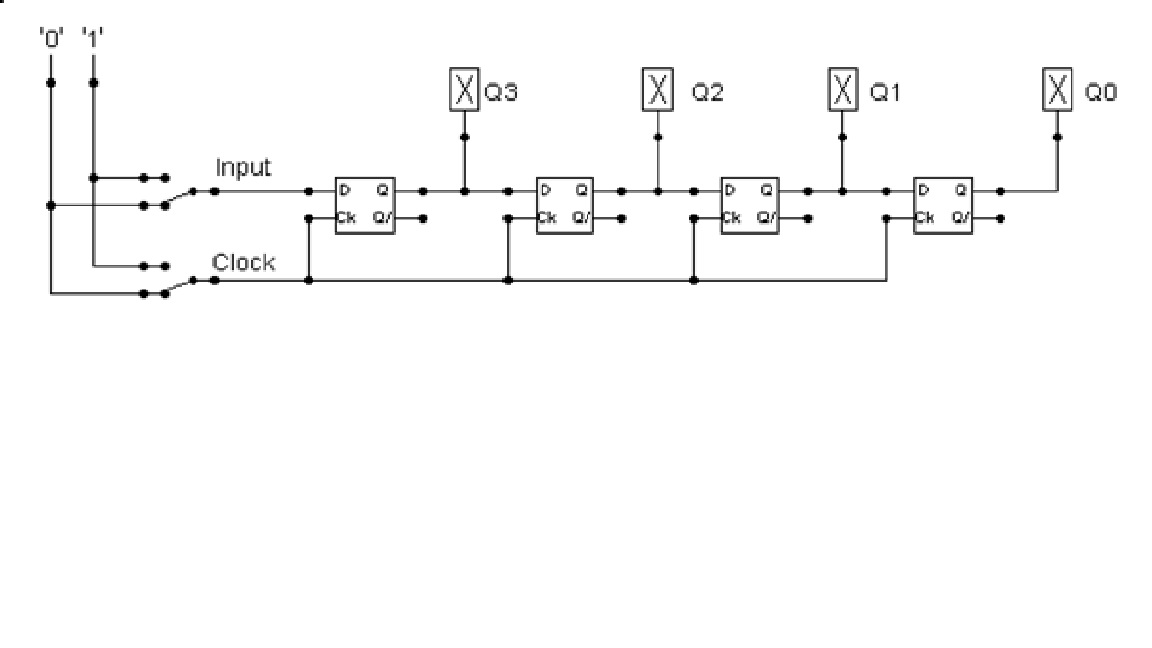
Load the LogSim Arithmetic Logic Unit Circuit **alu.cct** from inside the logsim application (You'll find it in the logsim folder) (***You may need to right-click on the link to download the file instead of opening it in the browser)***. It should look like this:  
  
  
  
The circuit behaves like a simple arithmetic logic unit. The inputs A0-A3 represent a 4 bit binary number. Inputs B0-B3 represent another binary number. A0 and B0 are the least significant bits respectively. The following table details the functions supported by the chip. All other control lines = 0.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Function | AND | OR | XOR | NAND | NOR | NOT A | ADD | SUBTRACT |
| X3 – X0 | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 1010 | 1011 |

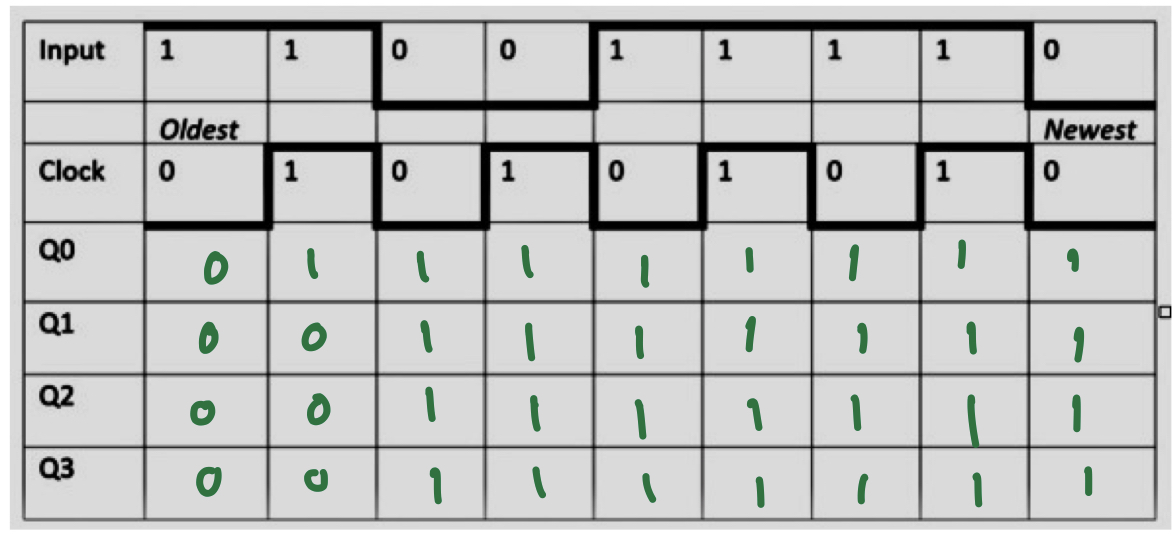
Use A= 11 B=4, complete the following table in binary ***(15 marks)***:

|  |  |
| --- | --- |
| FUNCTION | OUTPUT |
| AND | 0001 |
| OR | 1111 |
| XOR | 1110 |
| NAND | 1110 |
| NOR | 0000 |
| NOT A | 0100 |
| ADD | 0000 carry = 1 |
| SUBTRACT | 0110 |

The logical operations are bitwise. Manually prove each operation has returned the correct result by  ***(15 marks)***:  
Example:  1 0 1 1  
                 1 0 1 0 AND OPERATION  
                 1 0 1 0 RESUL

Serial to Parallel Decoder ***(30 marks)***:  


Build the circuit above and complete the following timing diagram by filling in the table spaces with ‘1’ or ‘0’. ***(15 marks)***



Describe what the circuit does. ***(15 marks)***

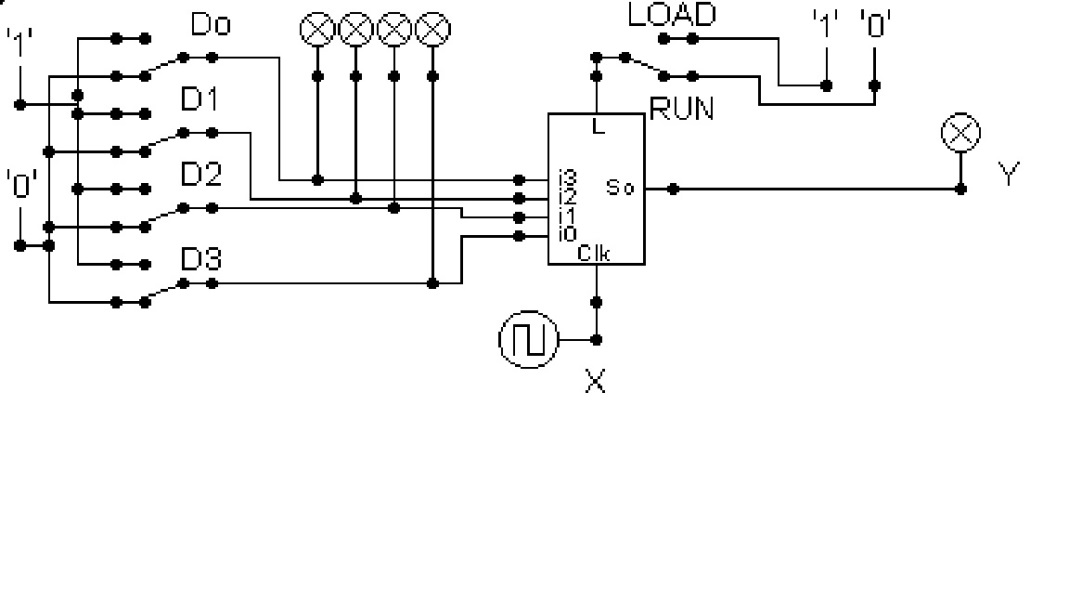
***It’s a series to parallel register, which means it can store one input before transferring the data into multiple outputs. It is constructed from four flip flops or latches, where each latch is able to remember one bit of data because the output one is connected to the second one.***

***In above digram there is four latches connected together. The output one is the input for the next. Each latch is also connect to a clock. The latch’s input only changes with the next clock pulse therefore the output of a flip flop effects the next flip flop only when the clock pulses.***

***So in above diagram it will take four clock pulses for a bit of data to be moved from left to the right.***

***Because it is a series to parallel, the data can only come in/input can only be in series, and you can have multiple parallel outputs.***

Parallel to Serial converter

Open the LogSim circuit **week5.cct** from the Logsim folder. It should look like this:  
  
  
  
Describe what this circuit does. ***(15 marks)***

It’s a shift register with parallel inputs and series output. The inputs can be multiple inputs in parallel and shift to be sent out one output.

Design and add to the above circuit an additional circuit that takes the Clock X and the Output Y and decodes Y into 4 output indicators so that they match D0 – D3. Insert the LogSim GIF output of your design in the space below.

The highest marks will go to those who design the circuit such that it **AUTOMATICALLY** stops (not pauses) when the input to the circuit matches the output to the circuit

*Note: Save your GIF image when your output indicators match the input D0 - D3*. (35 marks)