**4CS015 – Workshop #5 TO BE SUBMITTED**

Name:

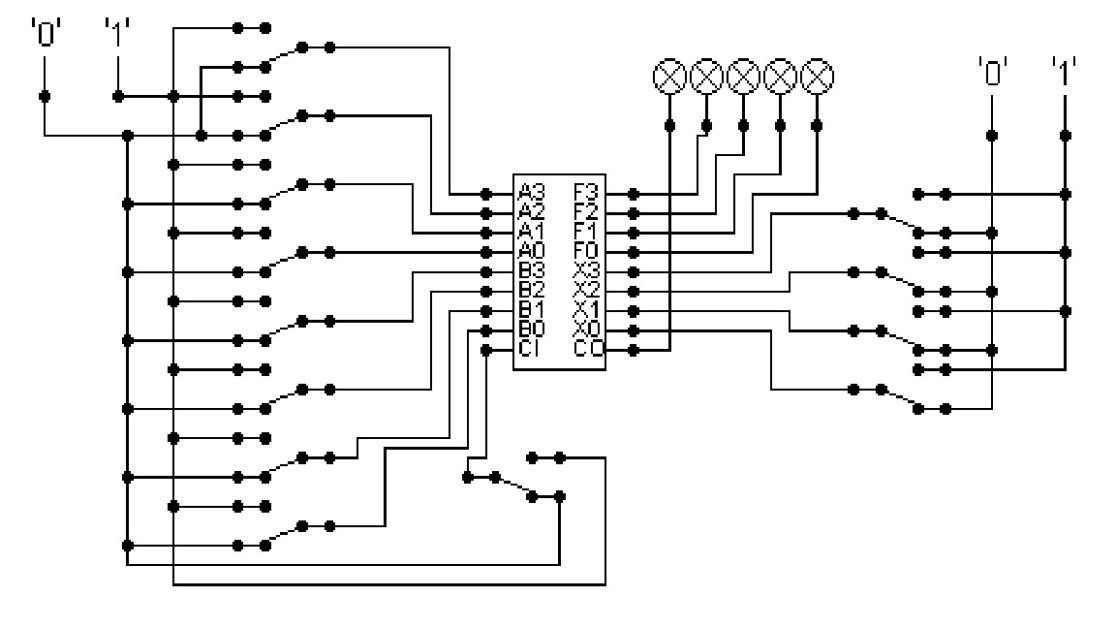
Student ID:

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 |  |
| OR |  |
| XOR |  |

|  |  |
| --- | --- |
| NAND |  |
| NOR |  |
| NOT A |  |
| ADD |  |
| SUBTRACT |  |

The logical operations are bitwise. Manually prove each operation has returned the correct result by ***(15 marks)***:

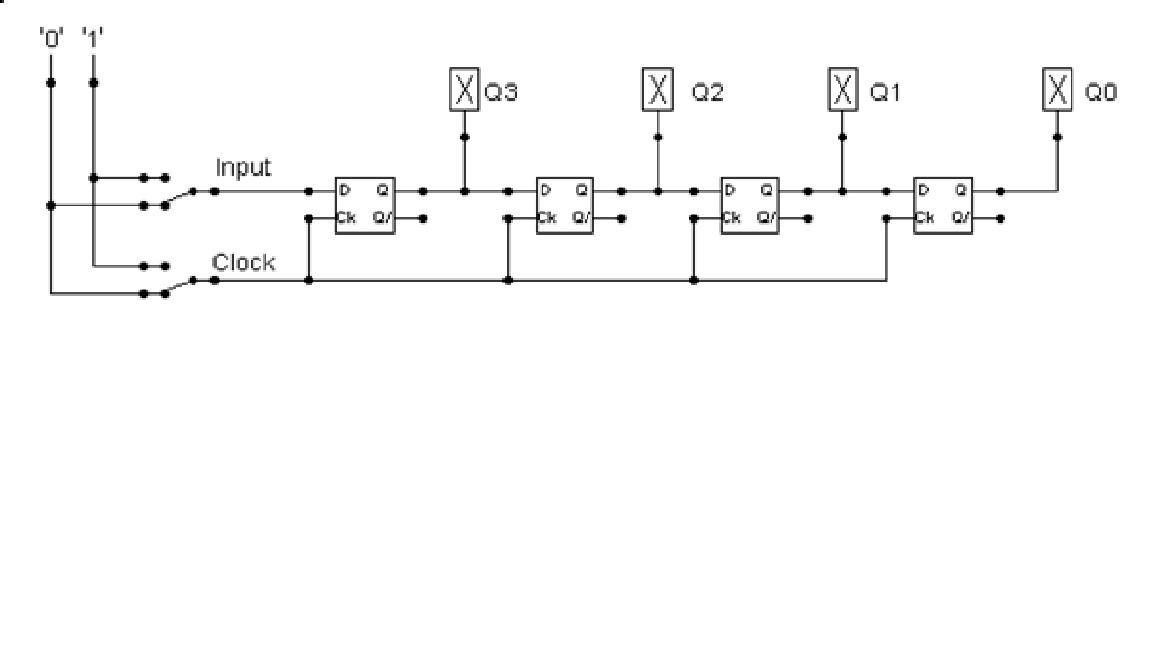
Example: 1 0 1 1

1 0 1 0 AND OPERATION

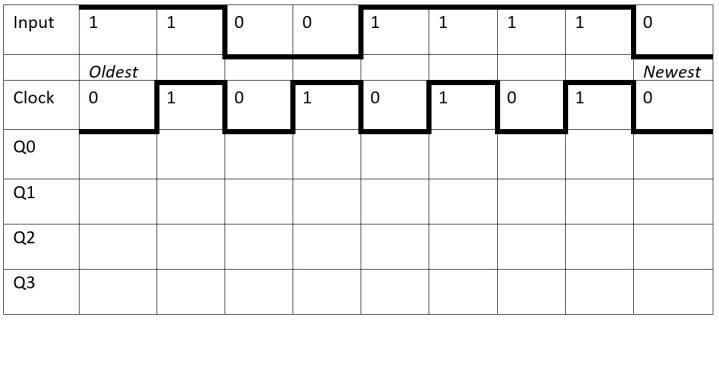
1 0 1 0 RESULT

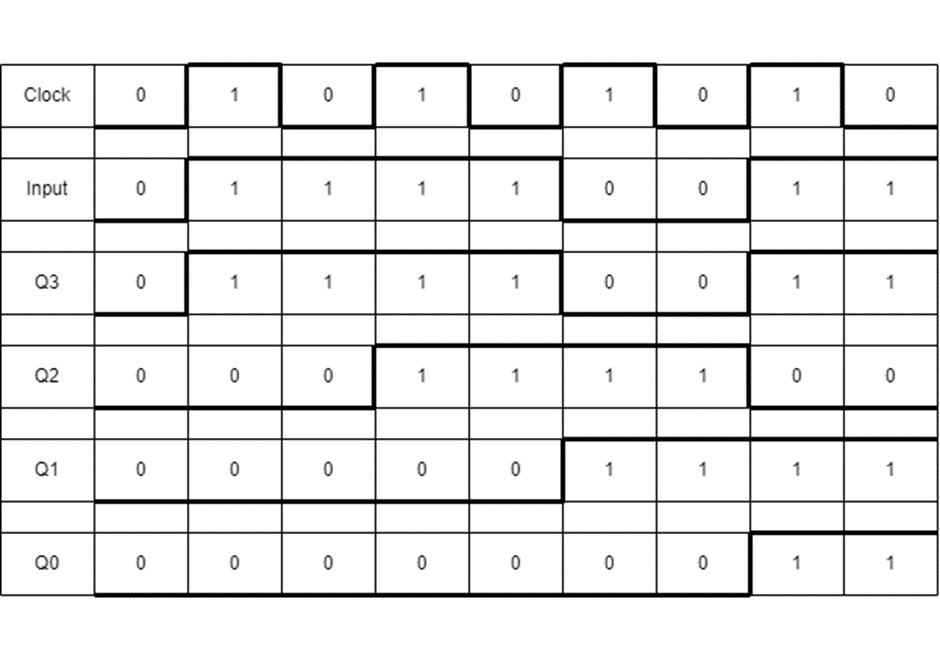
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ADD | OR | | | XOR |
| 1011  AND OPERATION | 1011  0100 OR OPERATION  1111 | | | 1011  0100 XOR OPERATION  1111 |
| NAND | NOR | | NOT A | |
| 1011  0100 NAND OPERATION  111 | 1011  0100 NOR OPERATION  000 | | 1011  0100 NOT A OPERATION  0100 | |
| ADD | | SUBTRACT | | |
| 1011  0100 ADD OPERATION  1111 | | 1011  0100 SUBTRACT OPERATION  0110 | | |

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)***





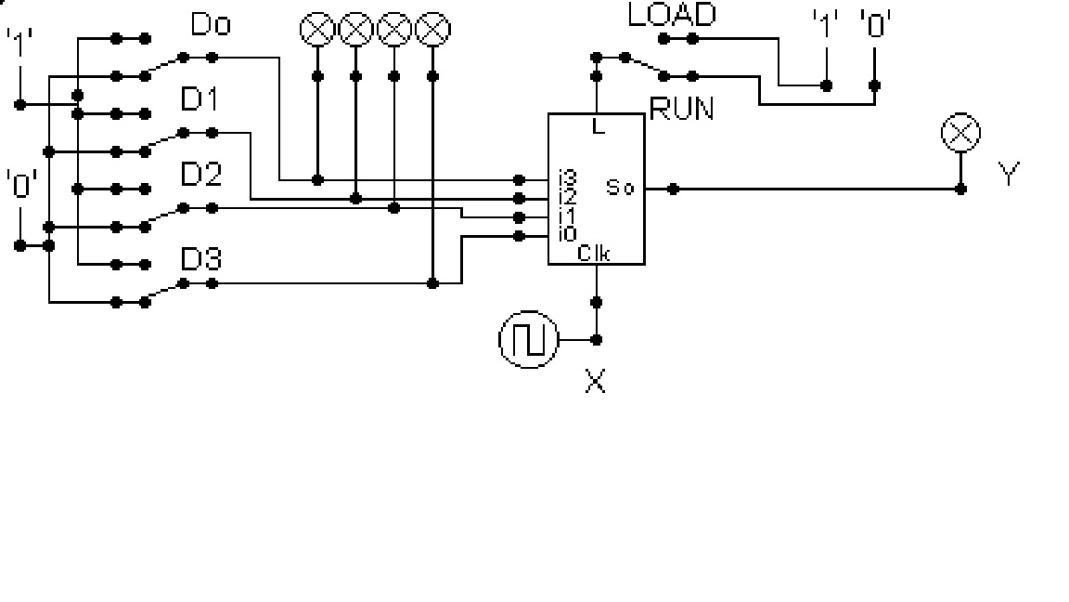
Describe what the circuit does. ***(15 marks)***

(c = Clock, I = Inputlet), Q3, Q2, Q1, Q0 are all zero when I is 1 and c is 0. Q3 is 1 and all other outputs are 0 when o and c are both 1. Q3 is 1 and all other outputs are 0 when I and c are both 0. Q2 is 1 and all other outputs are low when I = 0 and c = 1. (0). Q2 is 1 and all other outputs are 0 when I = 1

and c = 0. Q1 and Q3 are high (1) and Q2 and Q0 are low when I = 1 and c = 1. (0). Q1 and Q3 are both 1 and low when I = 1 and c = 0. Q0, Q2, and Q3 when I = 1 and c = 1

Parallel to Serial converter

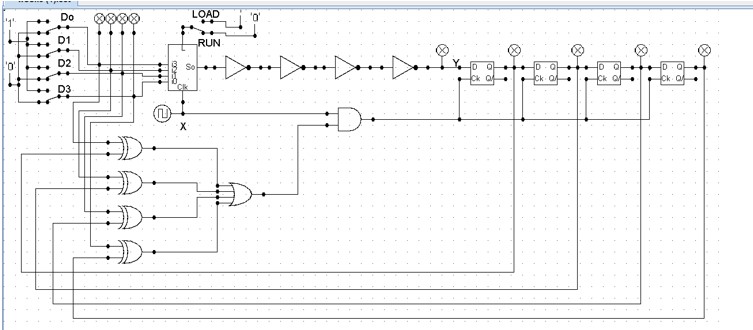
Open the LogSim circuit **week5.cct** from the Logsim folder. It should look like this:



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

Our four inputs are D0, D1, D2, and D3. When we provide high input from all the input sources, a shift register operates serially in parallel. And the output Y is high when the switch is in the LOAD position. However, if we switch to RUN, the register will serially reduce the input value until it eventually reaches zero and outputs Y, which is low (0).

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)

