# 4CS015 – Workshop #5 TO BE SUBMITTED

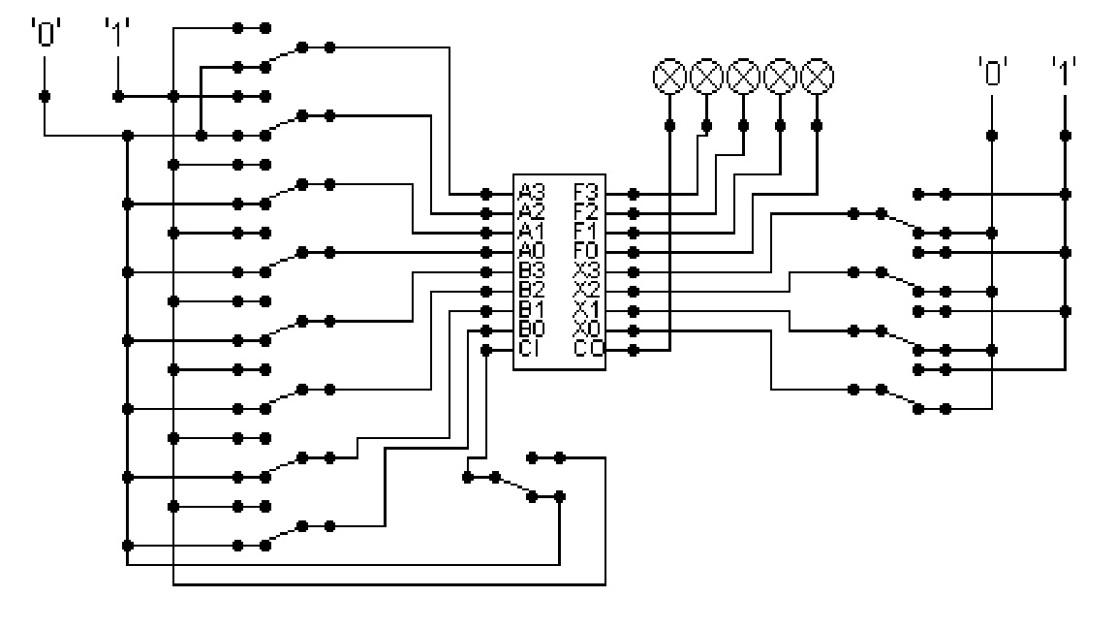
Name: Akriti Kumari Dev

Student ID: 2329584

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)***:

A=11=1011

B=4=0100

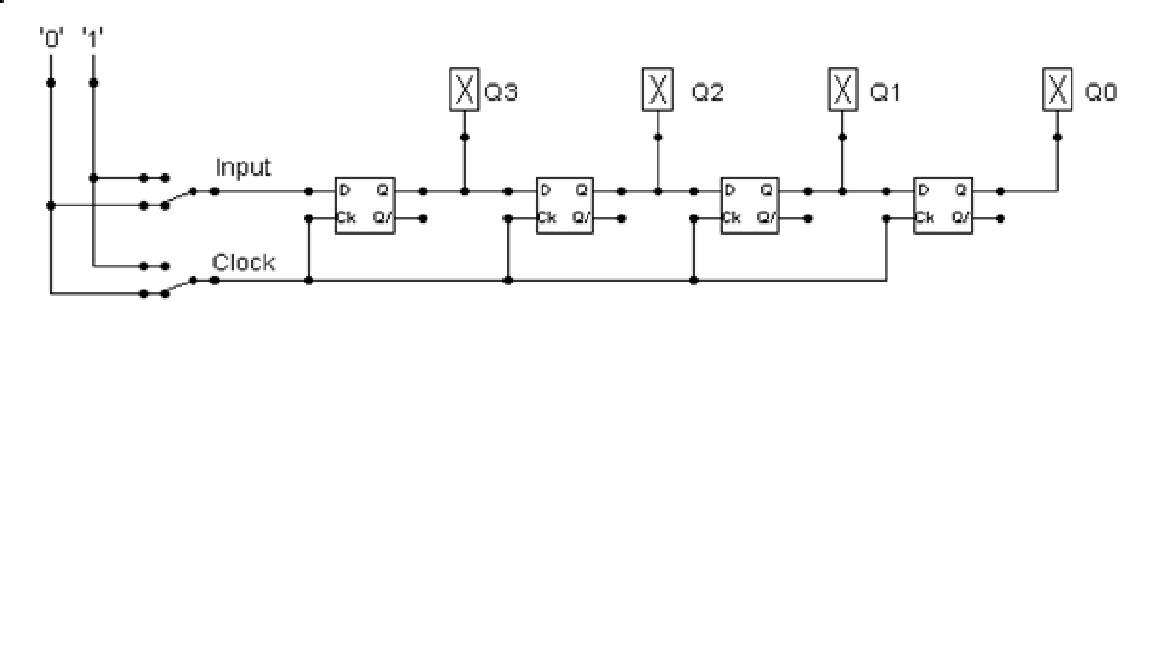
| FUNCTION | OUTPUT |
| --- | --- |
| AND(0000) |  |
| OR(0001) |  |
| XOR(0010) |  |
| NAND(0011) |  |
| NOR(0100) |  |
| NOT A(0101) |  |
| ADD(1010) |  |
| SUBTRACT(1011) |  |

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

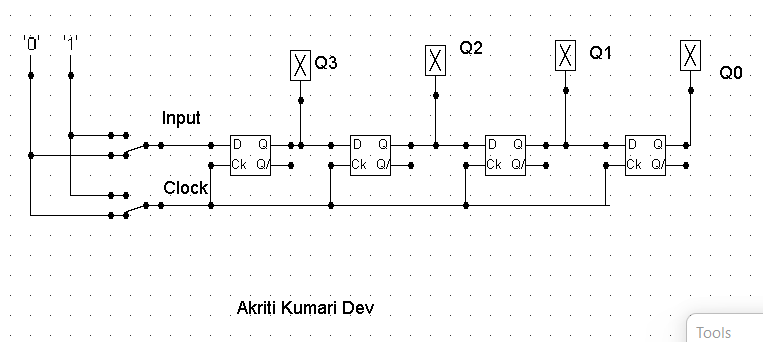
| AND | OR | XOR |
| --- | --- | --- |
| 1011  0100 AND operation  0000 | 1011  0100 OR operation  1111 | 1011  0100 XOR operation  1111 |

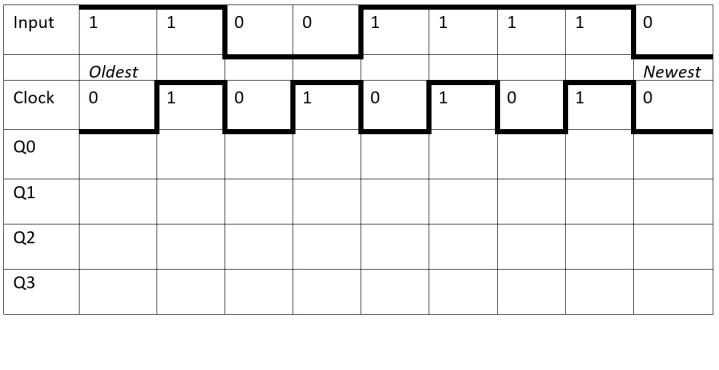
| NAND | NOR | NOT A |
| --- | --- | --- |
| 1011  0100 NAND operation  1111 | 1011  0100 NOR operation  0000 | 1011  0100 NOT A operation  0100 |

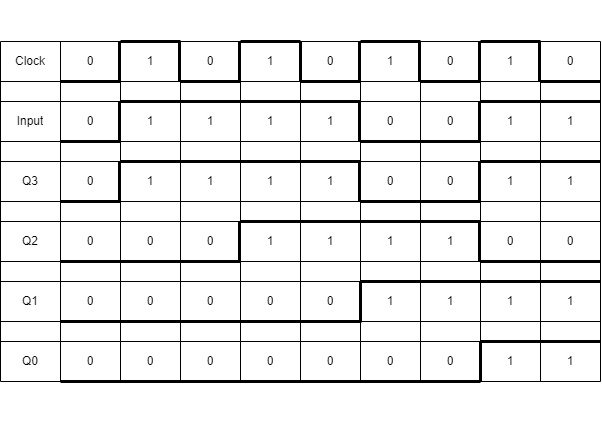
| ADD | SUBTRACT |
| --- | --- |
| 1011  0100 ADD operation  1111 | 1011  0100 SUBTRACT operation  0110 |

Serial to Parallel Decoder ***(30 marks)***:  
https://canvas.wlv.ac.uk/courses/8457/assignments/15581?module_item_id=45068

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

******



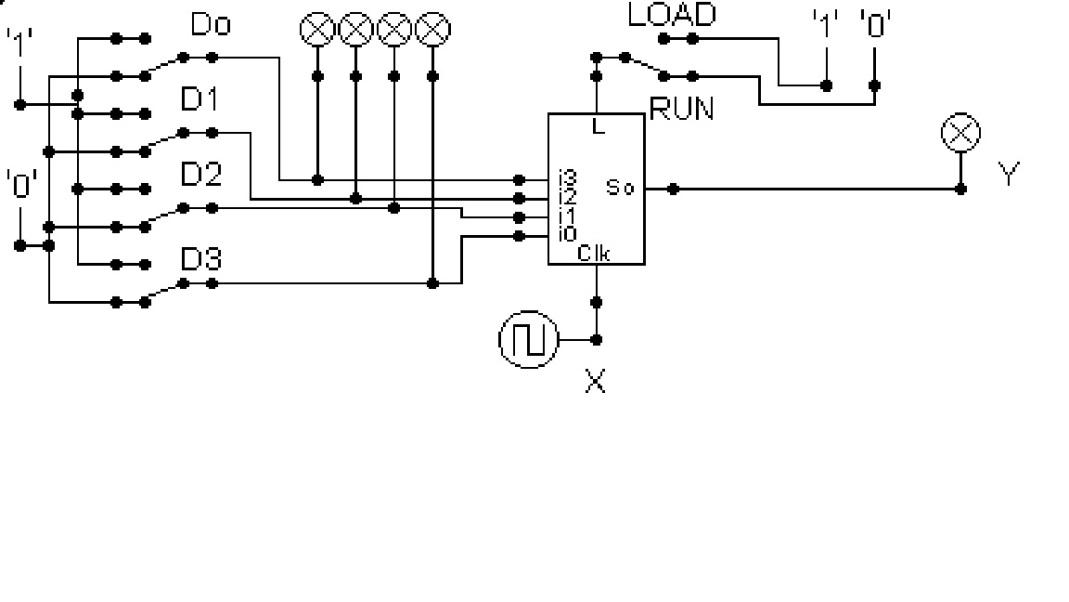


Describe what the circuit does. ***(15 marks)***

A SIPO (serial-in, parallel-out) shift register is a shift register that takes data serially and outputs it in parallel. Each bit of data is initially stored in a separate memory cell, and then until all of the data has been transferred, each bit is transferred one at a time. The data is then available in parallel form on the shift register's output pins. Digital storage, data acquisition, logic circuits, and communication systems are among the uses for SIPO shift registers. In this instance, data are outputted in parallel after being serially input.Throughout a clock cycle, the circuit receives an input value and generates the desired outputs, Q0, Q1, Q2, and Q3. The newest input value in the described circuit is zero when both the clock value and the clock value, which alternates between 0 and 1 after one instance, are 0. The question asks for the execution of two pairs of ones as inputs, followed by their re-input over the course of a few cycles. Additionally, two ones and two zeros must be written one cycle apart. A 0 and 1 are consequently executed, respectively.

The clock and the most recent input both have a value of 0, which likewise applies to all four output values. When the clock value and input value are both 1, all subsequent values except for Q3 (whose value comes out to be 1) convert to zero. When the conditions change, the outputs are modified by 1 bit at a time. When the clock is set to 0 and input is set to 1, Q3 and Q2 similarly switch on but Q1 and Q0 remain off. Furthermore, when the clock value and input value are 1 and 0, respectively, Q2 and Q1 are enabled, and the rest remain dormant. The clock value is zero, while the input value is one. Finally, all outputs except for Q2 are activated.

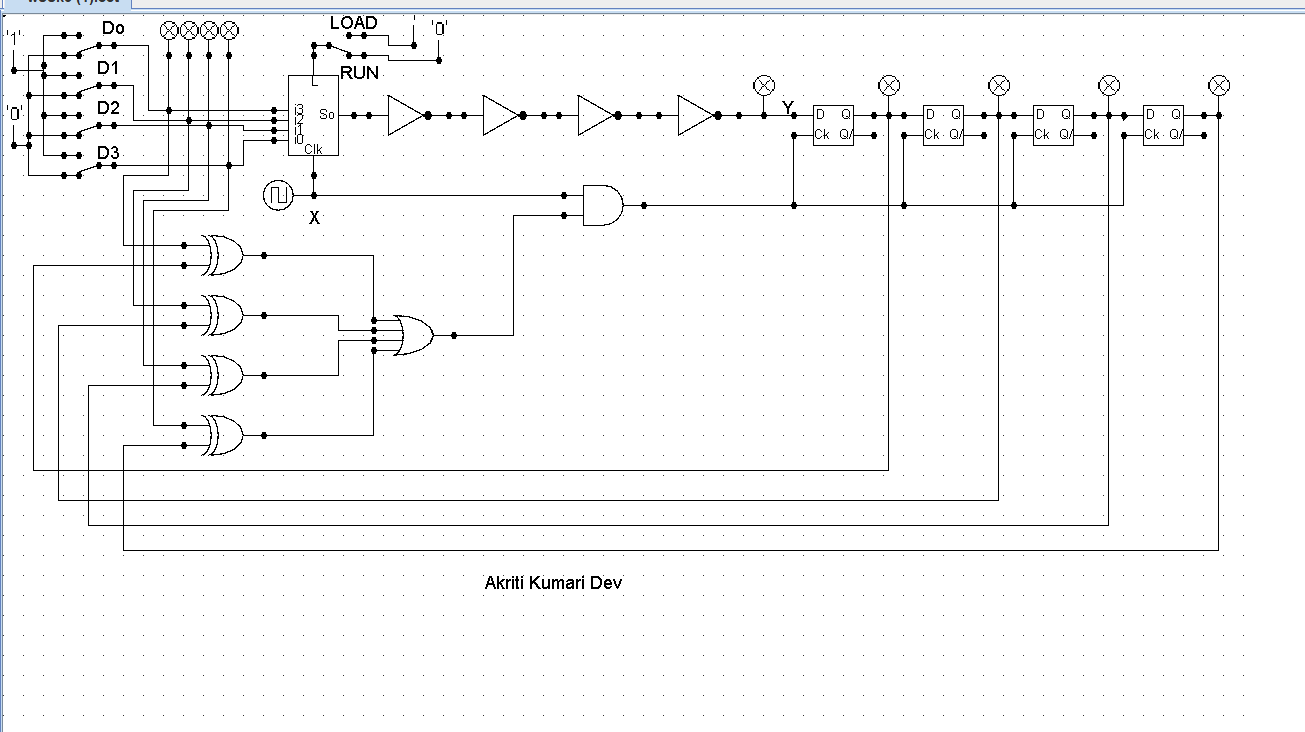
Parallel to Serial converter

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

A form of digital electronic circuit used to store data is called a PISO (Parallel-In, Serial-Out) shift register. Binary numbers are used to store the data, which can be moved from one circuit step to the next. As illustrated in the image, a PISO shift register is made up of a chain of flip-flops, each of which has a single input and single output.One piece of data is stored in each flip-flop, and it is serially shifted from one stage to the next. The first flip-flop receives the first bit of data, which is subsequently moved to the second flip-flop, and so forth. The information is released from the final flip-flop in the chain and made usable. It is utilised in digital logic circuits, microprocessors, and memory systems.

The logic circuit mentioned above represents a PISO register. Four D flip-flops are connected to one another to form the circuit. Despite being right next to each flip flop, the clock input is connected to each one separately at the input entry through a multiplexer. The output of the multiplexer is coupled to the next flip flop, while the output of the previous flip flop and parallel inputs are connected to the input of the multiplexer. Flip-flops are connected in series with one another since they all get the same clock signal. The given circuit converts the input output Y into four output indicators that are equal to the inputs D0 to D3 after them.

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)

