4CS015 – Workshop #5 TO BE SUBMITTED

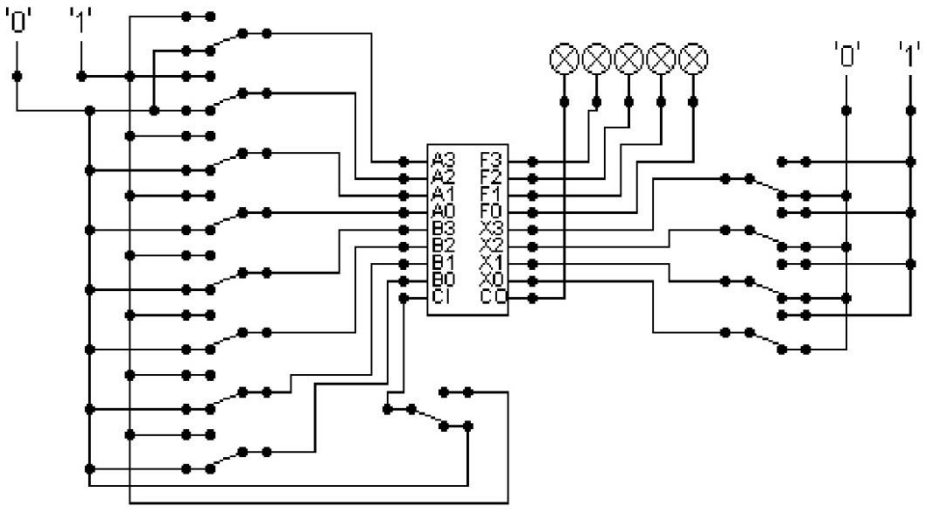
Name: Kamal Dhital

Student ID: 2407046

Workshop tasks:

Arithmetic Logic Unit:

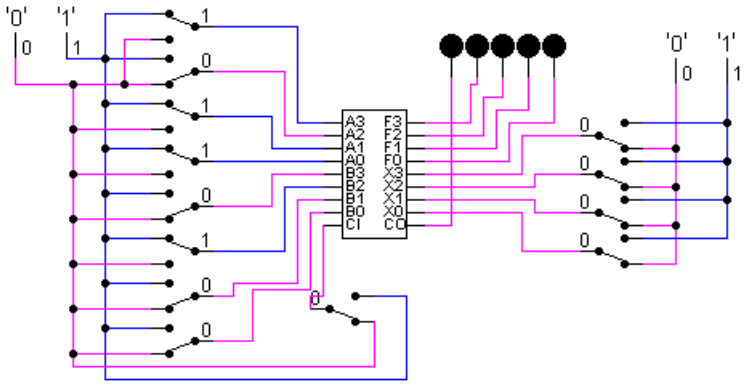
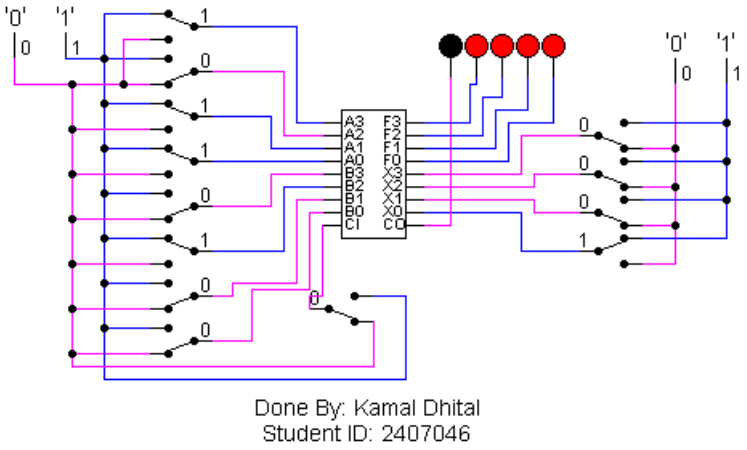
Load the LogSim Arithmetic Logic Unit Circuit **alu.cct** from inside the logsim application (You'll find it in the logsim folder) (***You may need to right-click on the link to download the file instead of opening it in the browser)***. It should look like this:

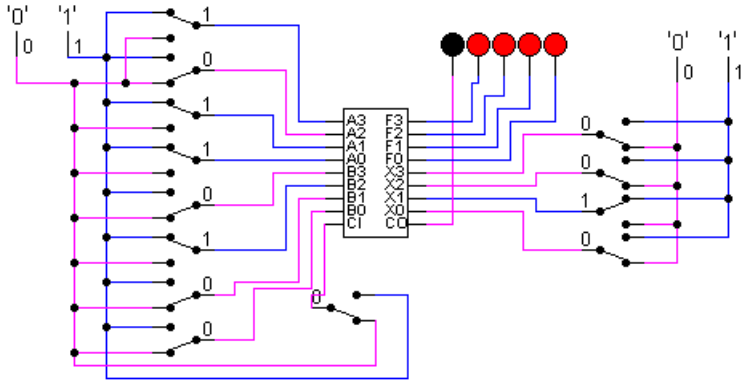
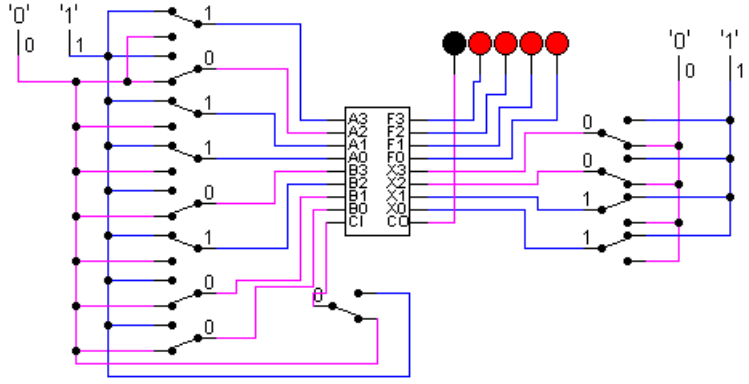
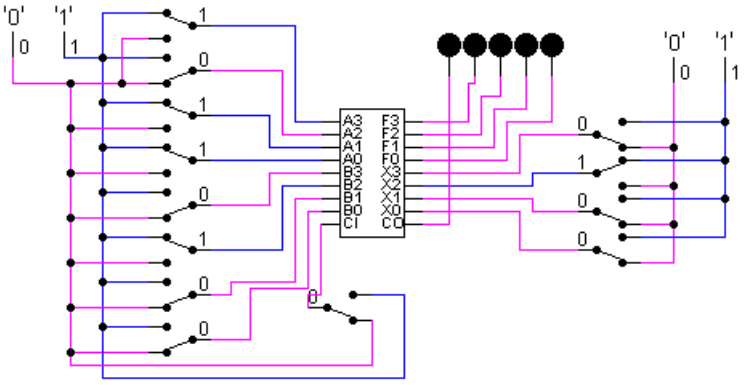


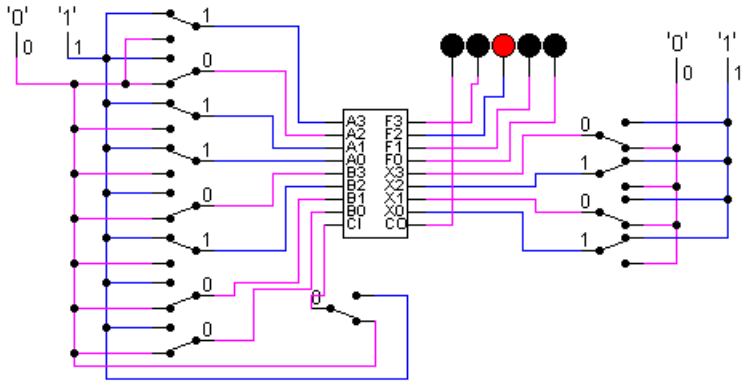
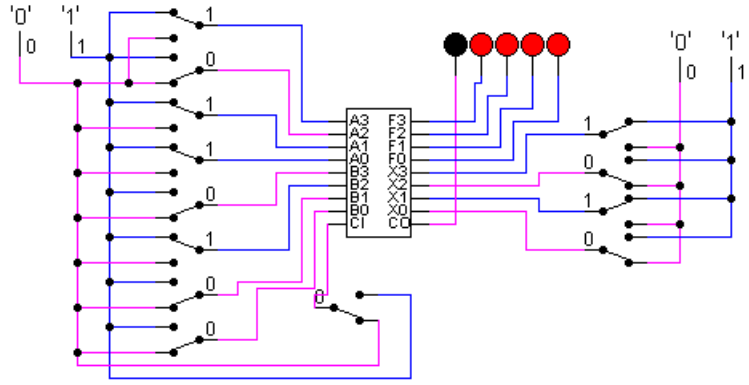
The circuit behaves like a simple arithmetic logic unit. The inputs A0-A3 represent a 4-bit binary number. Inputs B0-B3 represent another binary number. A0 and B0 are the least significant bits respectively. The following table details the functions supported by the chip. All other control lines = 0.

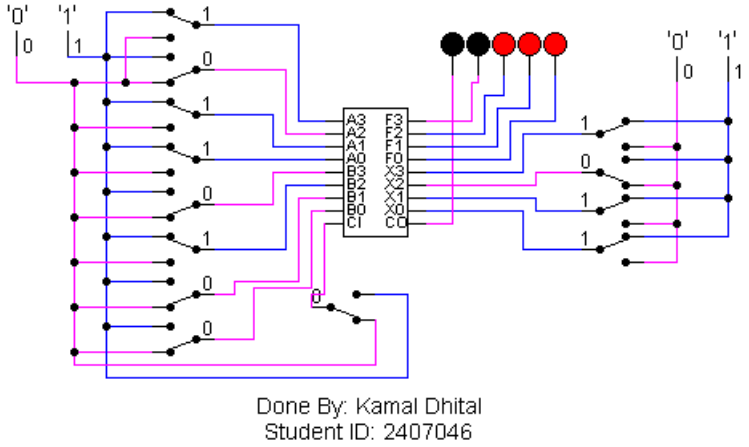
Function	AND	OR	XOR	NAND	NOR	NOT A	ADD	SUBTRACT
X3 – X0	0000	0001	0010	0011	0100	0101	1010	1011

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

FUNCTION	OUTPUT
AND	 1011 (AND) 0100 = 0000
OR	 1011 (OR) 0100 = 1111

XOR	 Done By: Kamal Dhital Student ID: 2407046 1011 (XOR) 0100 = 1111
NAND	 Done By: Kamal Dhital Student ID: 2407046 1011 (NAND) 0100 = 1111
NOR	 Done By: Kamal Dhital Student ID: 2407046

	1011 (NOR) 0100 = 0000
NOT A	 Done By: Kamal Dhital Student ID: 2407046 1011 (NOT A) 0100 = 0100
ADD	 Done By: Kamal Dhital Student ID: 2407046 1011 + 0100 = 1111

SUBTRACT	 1011-0100 = 0111

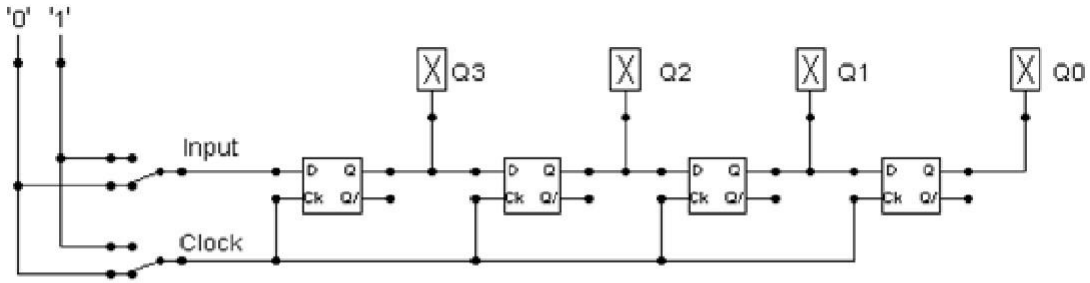
The logical operations are bitwise. Manually prove each operation has returned the correct result by ***(15 marks)***:

Example: 1 0 1 1

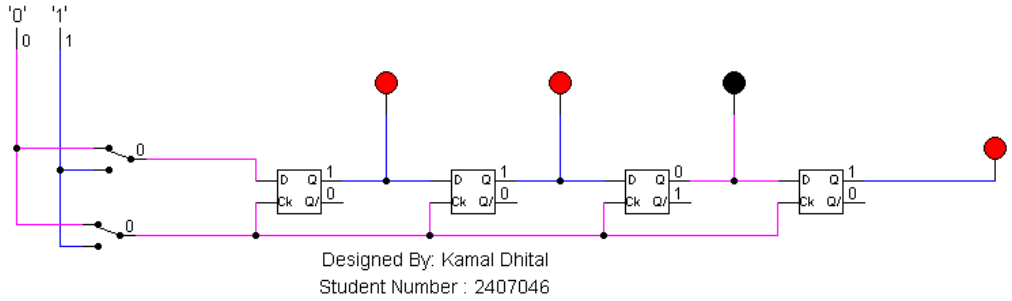
1 0 1 0 AND OPERATION

1 0 1 0 RESULT

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

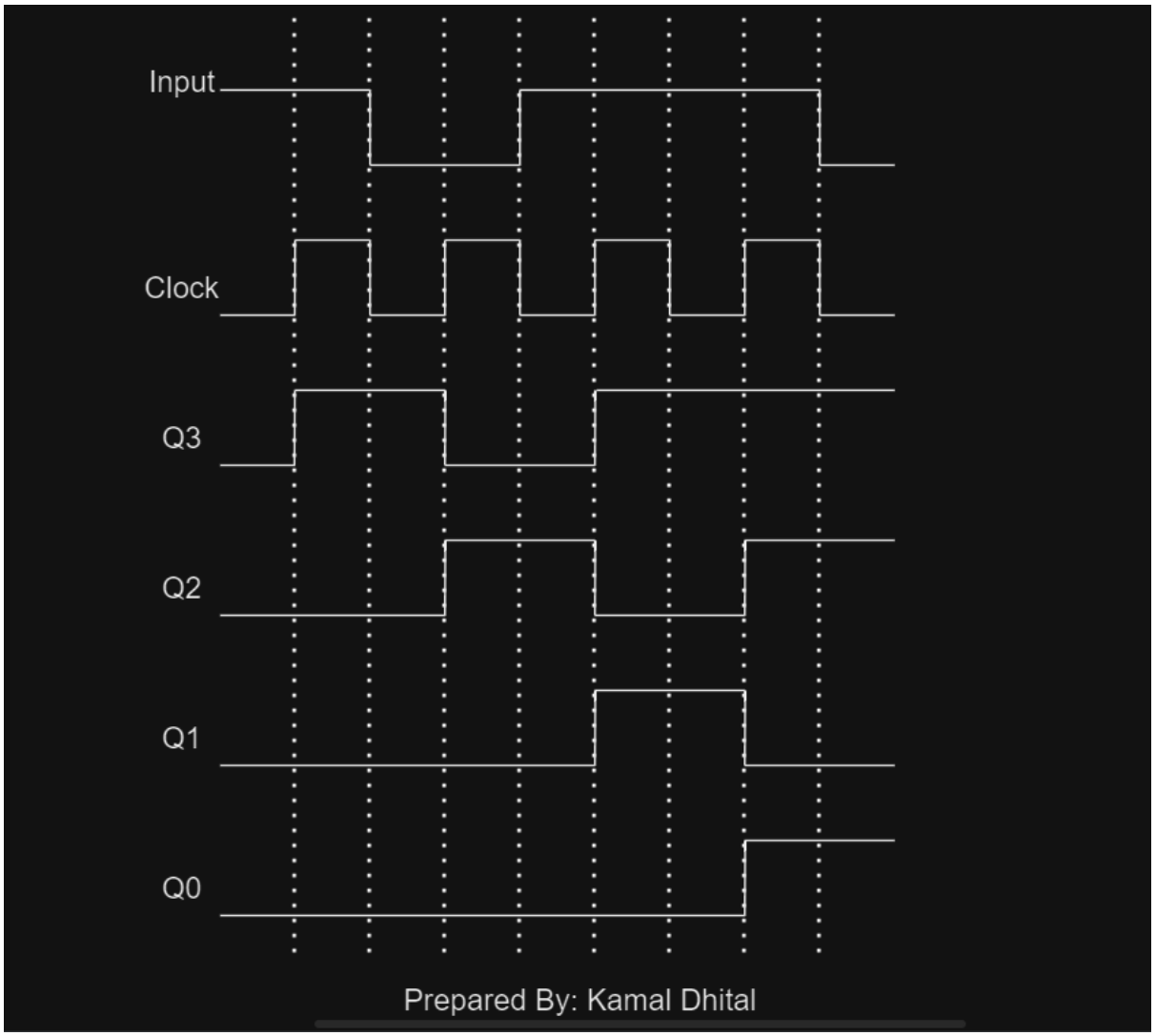


Build the circuit above and complete the following timing diagram by filling in the table spaces with '1' or '0'. ***(15 marks)***



Input	Clock	Q3	Q2	Q1	Q0
1	0	0	0	0	0
1	1	1	0	0	0
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	1	0	0
1	1	1	0	1	0
1	0	1	0	1	0
1	1	1	1	0	1
0	0	1	1	0	1

Input	1	1	0	0	1	1	1	1	0
	Oldest								*Newest*
Clock	0	1	0	1	0	1	0	1	0
Q0									
Q1									
Q2									
Q3									



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

A device that takes input in serial format and converts it into parallel format using D flip-flop to store serial bits and provide the output in parallel form. It takes 4 bit of serial input data and convert it into 4 bits of parallel output and gives to the user.

At the beginning, we need to set the input to 0 and change the clock until all the value are 0 and clock itself too. Once the input, clock and the indicator all are in off condition we are ready to start the process. Once it is done, we are going to change the value and get the output.

Let's see what happens we have the input 1101 to change it into the serial to parallel output.

At first, When input = 1 and clock = 0:

In this condition, we are getting the value of Q3 = 0, Q2 = 0, Q1 = 0 and Q0 = 0. It is because if clock is 0 nothing changes in the output.

When input = 1 and clock = 1:

In this condition, the value in the Q3 changes to 1 and the previous value of the Q3 shifted to Q2 and the previous value of Q2 changes into Q1, and the value of Q1 changes into Q0. The value in this state is Q3 = 1, Q2 = 0, Q1 = 0 and Q0 = 0.

When input = 0 and clock = 0:

In this condition, the value is not getting changes as the clock is not in input condition. So, the value of Q3 = 1, Q2 = 0, Q1 = 0, and Q0 = 0.

When input = 0 and clock = 1:

In this condition clock is running so we can definitely see the changes in the output. We get the value of Q3 = 0, Q2 = 1, Q1 = 0 and Q0 = 0.

When input = 1 and clock = 0:

In this condition, the value is not getting changes as the clock is not in input condition. So, the value of Q3 = 0, Q2 = 1, Q1 = 0, and Q0 = 0.

When input = 1 and clock = 1:

Moving forward, clock has changes its values to 1 and the values in this state are also changes. We get the value of Q3 = 1, Q2 = 0, Q1 = 1, Q0 = 0.

When input = 1 and clock = 0:

In this condition, the value is not getting changes as the clock is not in input condition. So, the value of Q3 = 1, Q2 = 0, Q1 = 1, and Q0 = 0.

When input = 1 and clock = 1:

After running the diagram with the input 1 and clock 1 we get the changes in the values as clock runs and the changed values are Q3 = 1, Q2 = 1, Q1 = 0 and Q0 = 0.

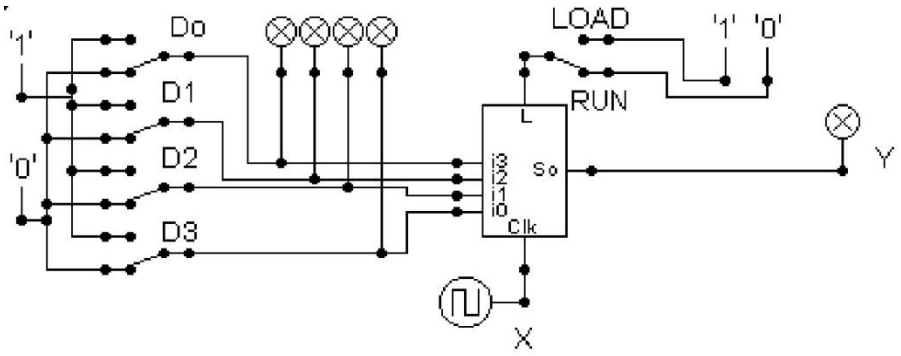
When input = 0 and clock = 0:

In this condition, the value is not getting changes as the clock is not in input condition. So, the value of Q3 = 1, Q2 = 1, Q1 = 0, and Q0 = 1.

We get the required output in parallel format. So, we stop the device and use these values where it is needed. This is how serial to parallel converter works to give the result in parallel format.

Parallel to Serial converter

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



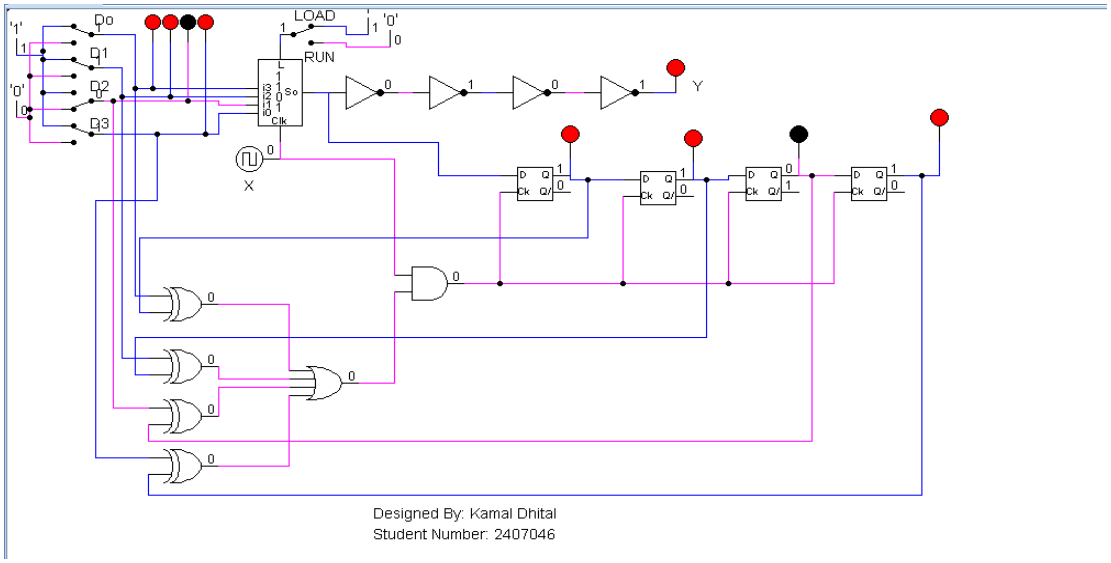
Describe what this circuit does. ***(15 marks)***

In the above diagram, we used the component like switches, clock, shift register, led indicator and the wire to connect one component to another component. At first, we have 4 switches and 0 and 1 is connected with each of the switches. These switches are working as the input for the user. In my case, I have set the input as 1101 and these switches are connected to led and lights it up where input is 1 and lights is off where input is 0. So, we can see the D2 is not hasn't light up as it holds the input 0. Further, these inputs connect with the shift register and some other component like clock and load switch are also connect with the shift register. The output from the shift register is goes by transferring the 4 NOT gate giving the same input as the previous output without using NOT gate and light up the indicator in this case.

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. And explain the working mechanism of our circuit. (35 marks)



In the above diagram, we used the component like switches, clock, D flip-flop, OR gate, XOR gate, AND gate, NOT gate, shift register, led indicator and the wire to connect one component to another component. At first, we have 4 switches and 0 and 1 is connected with each of the switches. These switches are working as the input for the user. In my case, I have set the input as 1101 and these switches are connected to led and lights it up where input is 1 and lights is off where input is 0. So, we can see the D2 is not hasn't light up as it holds the input 0. Further, these inputs connect with the shift register and some other component like clock and load switch are also connect with the shift register. The output from the shift register is goes by transferring the 4 NOT gate giving the same input as the previous output without using NOT gate and light up the indicator in this case.

Furthermore, we have 4 XOR gate getting the one input from the input switches and 1 input from the D flip-flop output. These 4 XOR gate works as the input for one OR gate and the output of OR gate works together with the previous clock to gives the 2 inputs to AND gate. The output of AND gate is works as the clock for each D flop-flop. D flip-flop takes the clock from the AND gate and input from the output of previous shift register and works as the serial in parallel out format. These 4-bit D flip-flop takes the serial input and gives the parallel output. As my input is 1101, I get the result 1101 in the D flip-flop which is the output of this circuit. Let's see how it works.

We are taking the input from the D0, D1, D2 and D3 and the led of D0, D1 and D3 light up as they hold the input 1 and D2 is not lighting up as it holds the input 0. Further these switches are connected to the shift register with the clock and load. This circuit only works when we send the value of 1 in the load. The shift register works itself gives the output of 1 passing though the 4 NOT gate giving the same output 1 at the end and the indicator light up as it holds the value 1. The output of shift register works as the input of the D flip-flop and the clock input for D flip-flop comes from the AND gate which takes the input from the OR gate and the OR gate takes 4 inputs from XOR gate and this XOR gates takes one input from the input field and one input from the output of D flip-flop. The D flip-flop taking the input in every step from the clock but the output of first D flip-flop works as the input for XOR gate and second D flip-flip and so on. In this way, this circuit works and stop when output is as same as input. In my case, I have the input of 1101 and the output also 1101 and the circuit stop working as both input and output are same.