4CS015 – Workshop #5

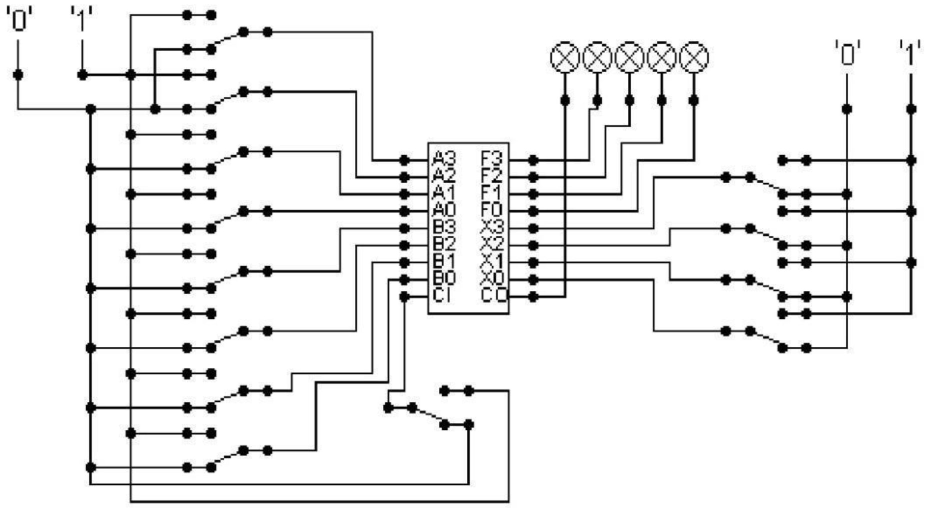
Name: Nayan Raj Khanal

Student ID: 2227486

Workshop tasks:

Arithmetic Logic Unit:

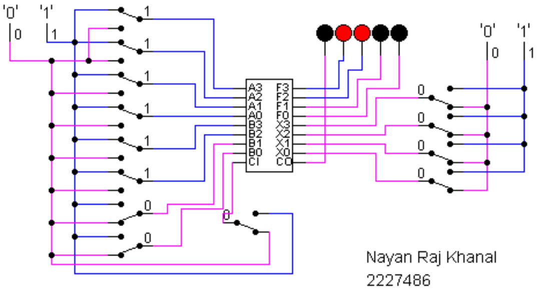
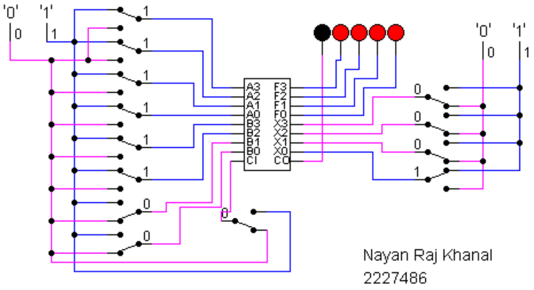
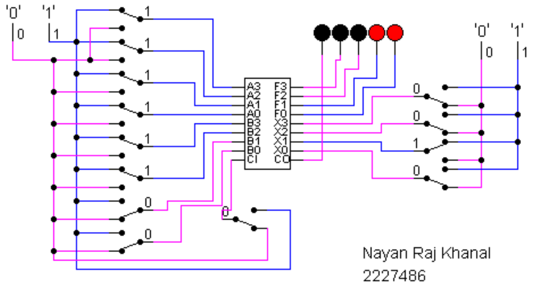
Load the LogSim Arithmetic Logic Unit Circuit **alu.cct** from inside the logsim application (You'll find it in the logsim folder) (***You may need to right-click on the link to download the file instead of opening it in the browser)***. It should look like this:

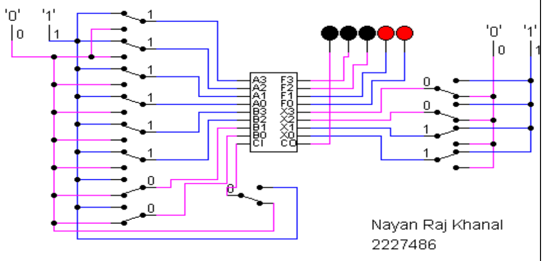
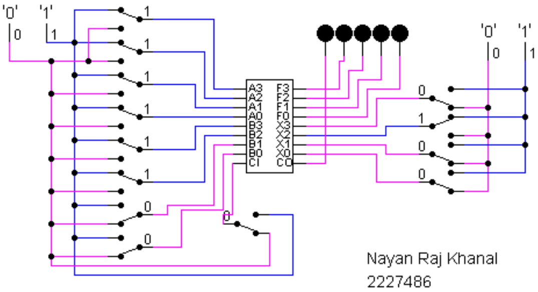
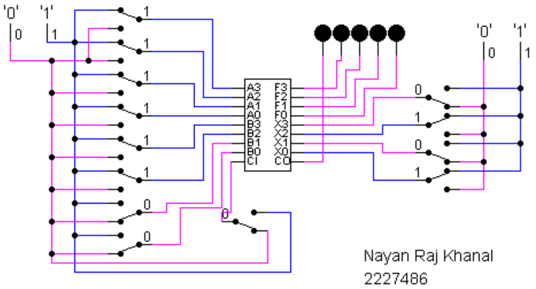
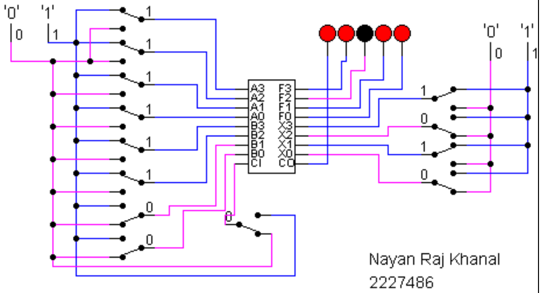


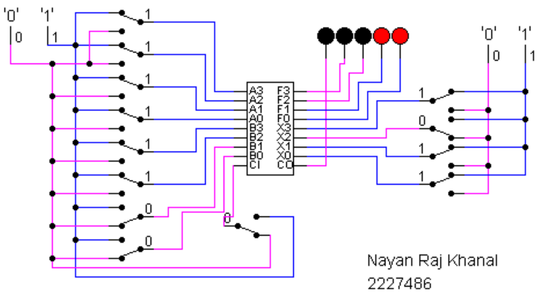
The circuit behaves like a simple arithmetic logic unit. The inputs A0-A3 represent a 4-bit binary number. Inputs B0-B3 represent another binary number. A0 and B0 are the least significant bits respectively. The following table details the functions supported by the chip. All other control lines = 0.

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

Use A= 15 B=12, complete the following table in binary ***(15 marks)***:

FUNCTION	OUTPUT
AND	01100  '0' '1' 0 1 A3 A2 A1 A0 B3 B2 B1 B0 CI F3 F2 F1 F0 X3 X2 X1 X0 CO Nayan Raj Khanal 2227486
OR	`01111  '0' '1' 0 1 A3 A2 A1 A0 B3 B2 B1 B0 CI F3 F2 F1 F0 X3 X2 X1 X0 CO Nayan Raj Khanal 2227486
XOR	00011  '0' '1' 0 1 A3 A2 A1 A0 B3 B2 B1 B0 CI F3 F2 F1 F0 X3 X2 X1 X0 CO Nayan Raj Khanal 2227486
NAND	00011

	 '0' '1' A3 A2 A1 A0 B3 B2 B1 B0 CI F3 F2 F1 F0 X3 X2 X1 X0 CO Nayan Raj Khanal 2227486
NOR	00000  '0' '1' A3 A2 A1 A0 B3 B2 B1 B0 CI F3 F2 F1 F0 X3 X2 X1 X0 CO Nayan Raj Khanal 2227486
NOT A	00000  '0' '1' A3 A2 A1 A0 B3 B2 B1 B0 CI F3 F2 F1 F0 X3 X2 X1 X0 CO Nayan Raj Khanal 2227486
ADD	11011  '0' '1' A3 A2 A1 A0 B3 B2 B1 B0 CI F3 F2 F1 F0 X3 X2 X1 X0 CO Nayan Raj Khanal 2227486

SUBTRACT	00011  '0' '1' 0 1 1 1 1 1 1 1 0 0 0 A3 A2 A1 A0 B3 B2 B1 B0 CI F3 F2 F1 F0 X3 X2 X1 X0 CO '0' '1' 0 1 1 0 1 1 Nayan Raj Khanal 2227486

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

1 1 1 1
1 1 0 0 AND
1 1 0 0 RESULT

1 1 1 1
1 1 0 0 OR
1 1 1 1 RESULT

1 1 1 1
1 1 0 0 XOR
0 0 1 1 RESULT

1 1 1 1
1 1 0 0 NAND
0 0 1 1 RESULT

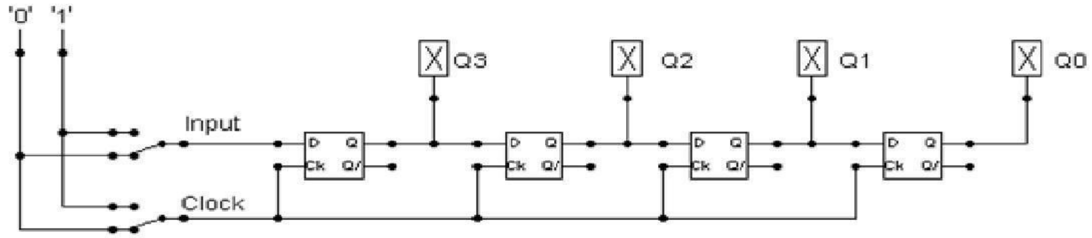
1 1 1 1
1 1 0 0 NOR
0 0 0 0 RESULT

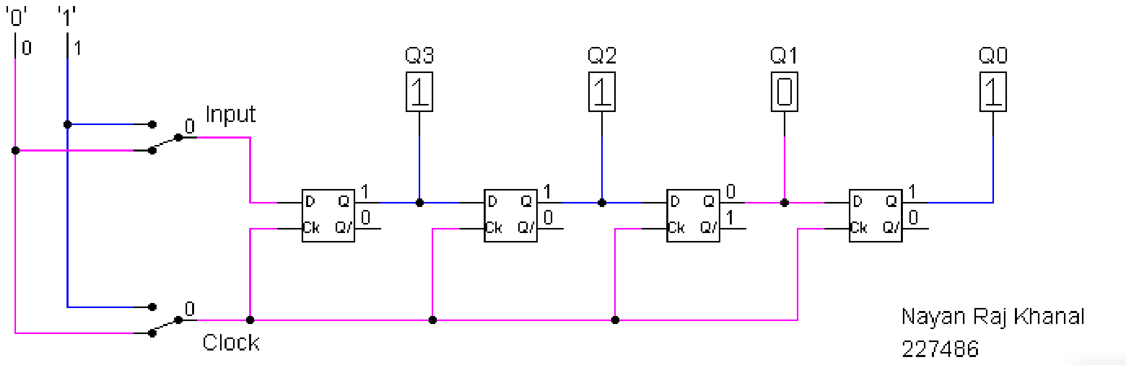
1 1 1 1
1 1 0 0 NOT A
0 0 0 0 RESULT

1 1 1 1
1 1 0 0 ADD
1 1 0 1 1 RESULT

1 1 1 1
1 1 0 0 SUB
0 0 1 1 RESULT

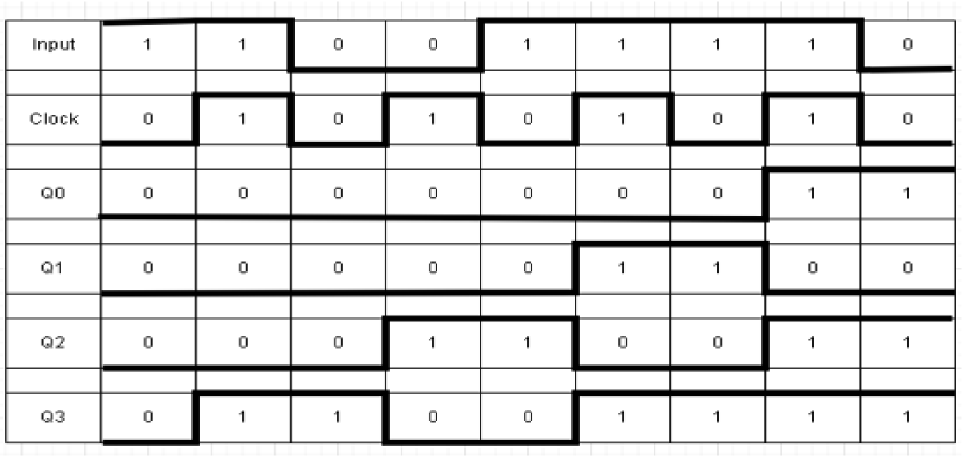
Serial to Parallel Decoder ***(30 marks)***:





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

Input	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									



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

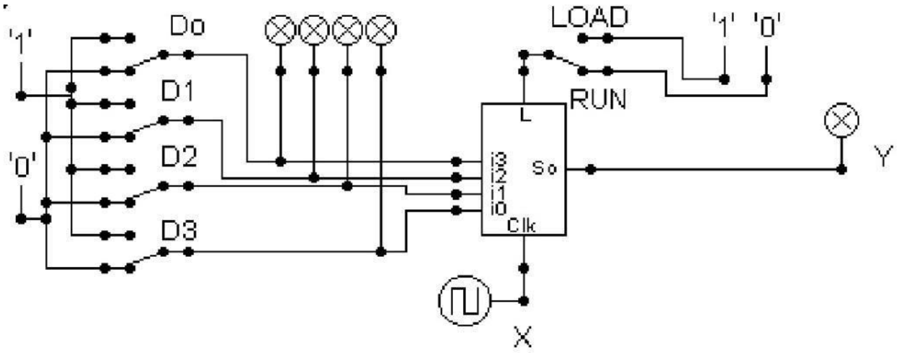
Nayan Raj Khanal 2227486

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

The following diagram shows serial and parallel inputs and outputs respectively. The outputs being Q0, Q1, Q2, and Q3. Also, four D-flips flops are included in the diagram. The input is only passed on each rising edge of the clock, that is, when it goes from '0' to '1'. At first, all of the values will be zero. After that input 1 is passed, resulting the clock to be set to zero. As the clock is at zero, the input value is not passed, making all outputs are set to zero. Likewise, when the input is 1, the clock is 1 as well, and now the input is transferred to Q3. Only when the clock value is 1, the input is taken and the received input is passed to the right causing shifting. This results in the output of the previous value becoming the input to the next.

<u>Parallel to Serial converter</u>

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



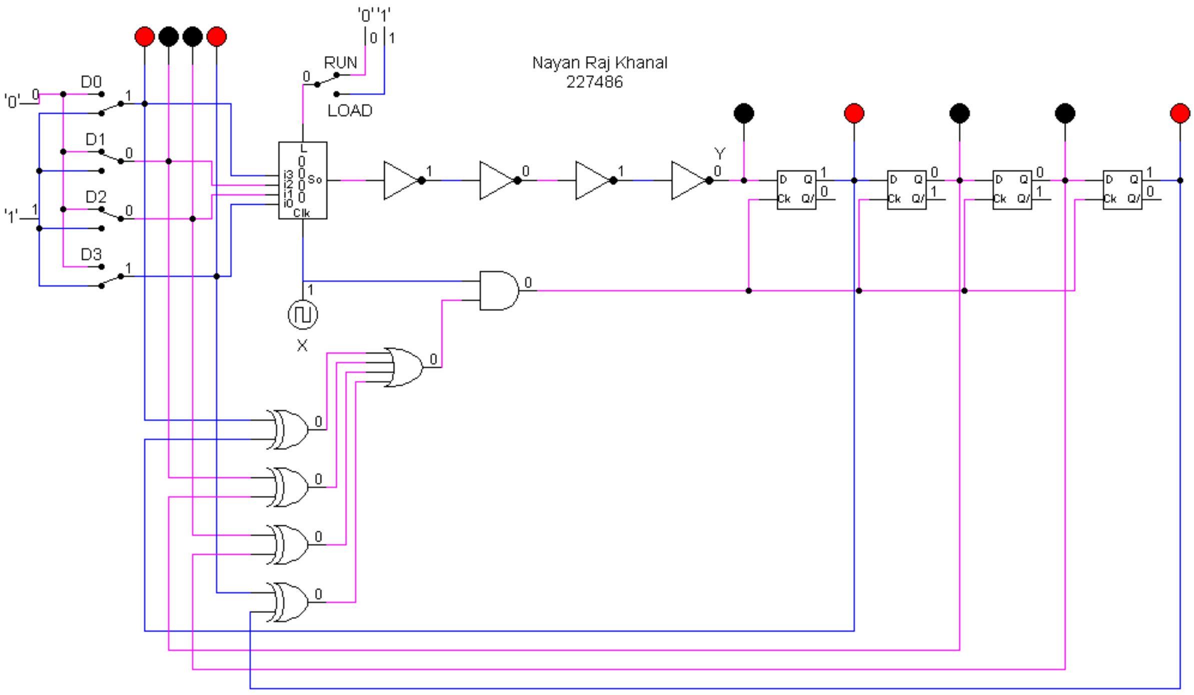
Describe what this circuit does. ***(15 marks)***

The diagram above depicts parallel and serial input and output. D0, D1, D2 and D3 are the available inputs. By placing the inputs in D0-D3 load gets enabled which then as the term suggests loads the provided input into the shift register. It works during the rising edge of the clock that is only when the clock X has the value of 1, the output is serially presented in Y after the input is loaded.

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)


'0' '1'
RUN
LOAD
Nayan Raj Khanal
227486
D0
D1
D2
D3
'0'
'1'
L
i3
i2
i1
i0
So
Clk
X
Y
D
Q
Ck
Q/