# 4CS015 – Workshop Portfolio Part-2

Name: SOYAM BAJGAIN

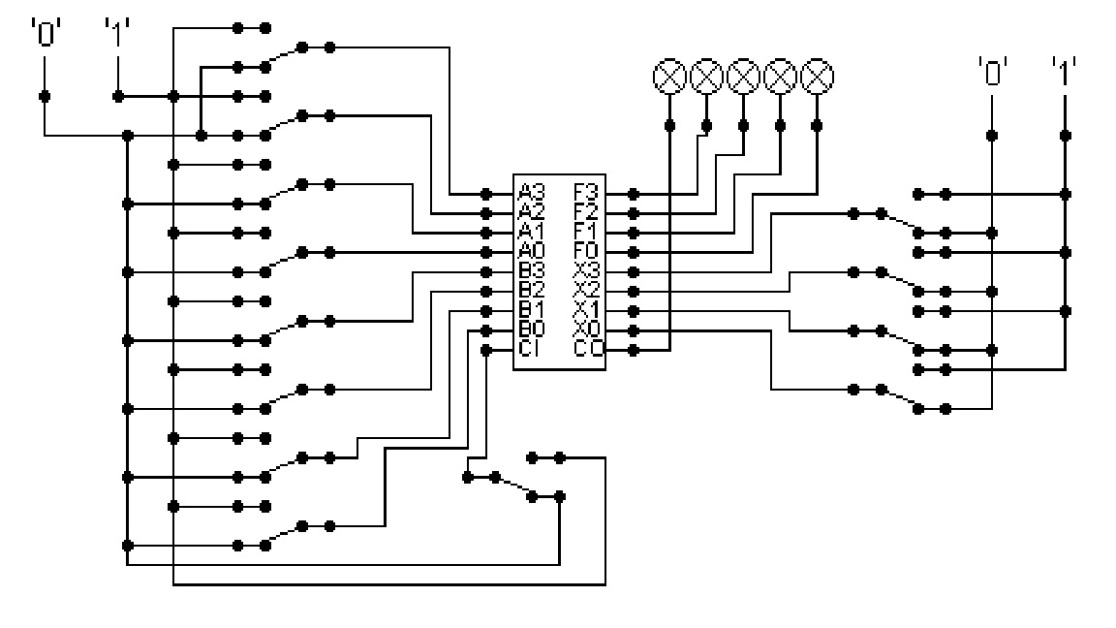
Student ID: 2358512

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

YourName\_UniID\_Wps5.docx

**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)***:

Answer:

A=11 in binary is 1011.

B=4 in binary is 0100.

|  |  |  |
| --- | --- | --- |
| FUNCTION | X3 – X0 | OUTPUT |
| AND | 0000 | 00000 |
| OR | 0001 | 01111 |
| XOR | 0010 | 01111 |
| NAND | 0011 | 01111 |
| NOR | 0100 | 00000 |
| NOT A | 0101 | 00100 |
| ADD | 1010 | 01111 |
| SUBTRACT | 1011 | 00111 |

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. AND

1 0 1 1

0 1 0 0 AND OPERATION

0 0 0 0 0 RESULT

1. OR

1 0 1 1

0 1 0 0 OR OPERATION

0 1 1 1 1 RESULT

1. XOR

1 0 1 1

0 1 0 0 XOR OPERATION

0 1 1 1 1 RESULT

1. NAND

1 0 1 1

0 1 0 0 NAND OPERATION

0 1 1 1 1 RESULT

1. NOR

1 0 1 1

0 1 0 0 NOR OPERATION

0 0 0 0 0 RESULT

1. NOT A

1 0 1 1

0 1 0 0 NOT A OPERATION

0 0 1 0 0 RESULT

1. ADD

1 0 1 1

0 1 0 0 ADD OPERATION

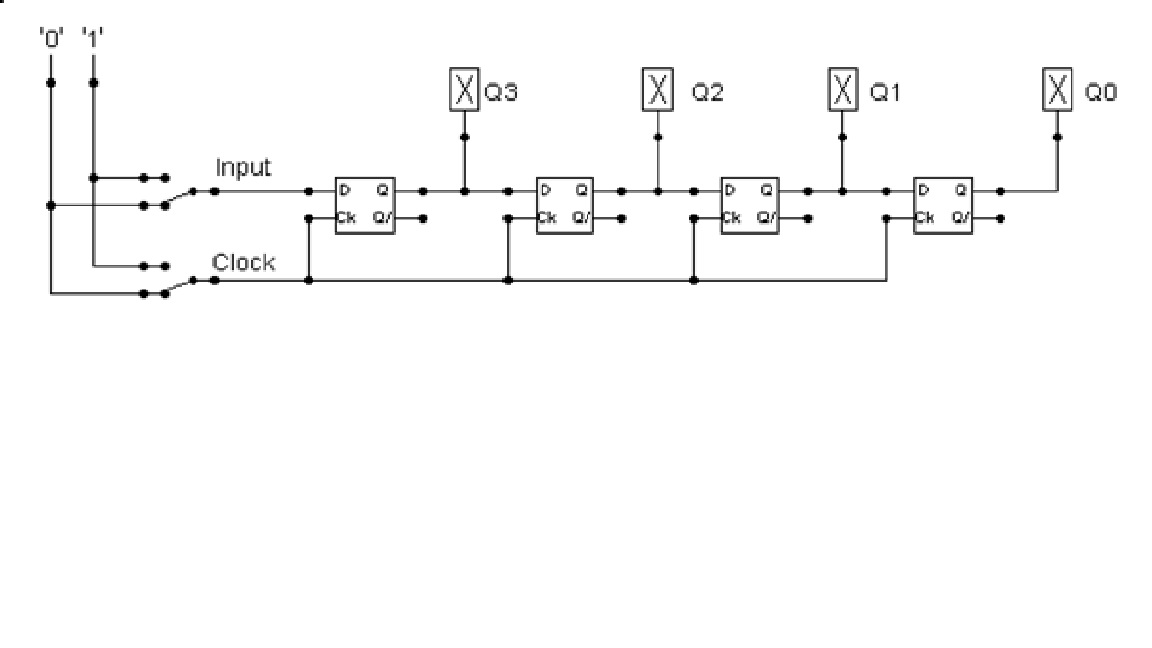
0 1 1 1 1 RESULT

1. SUBTRACT

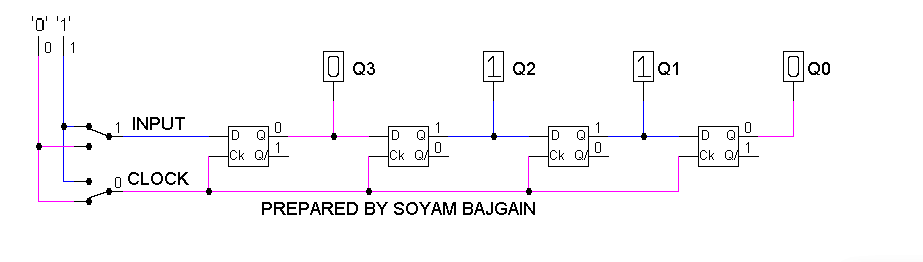
1 0 1 1

0 1 0 0 SUBTRACT OPERATION

0 0 1 1 1 RESULT

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


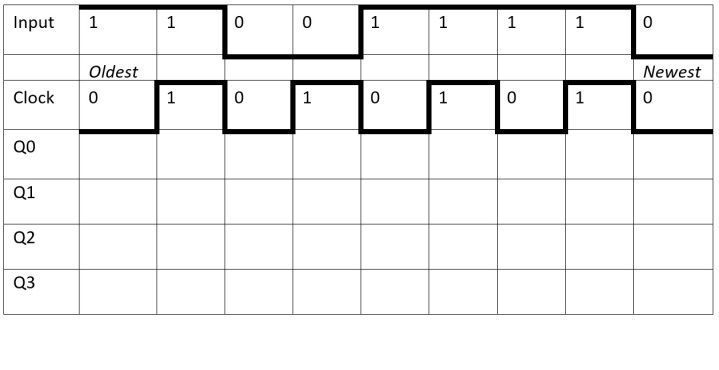
Build the circuit above and complete the following timing diagram by filling in the table spaces with ‘1’ or ‘0’. ***(15 marks)***



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CLOCK | INPUT | Q3 | Q2 | Q1 | Q0 |
| 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 |

Chart, box and whisker chart

Description automatically generated

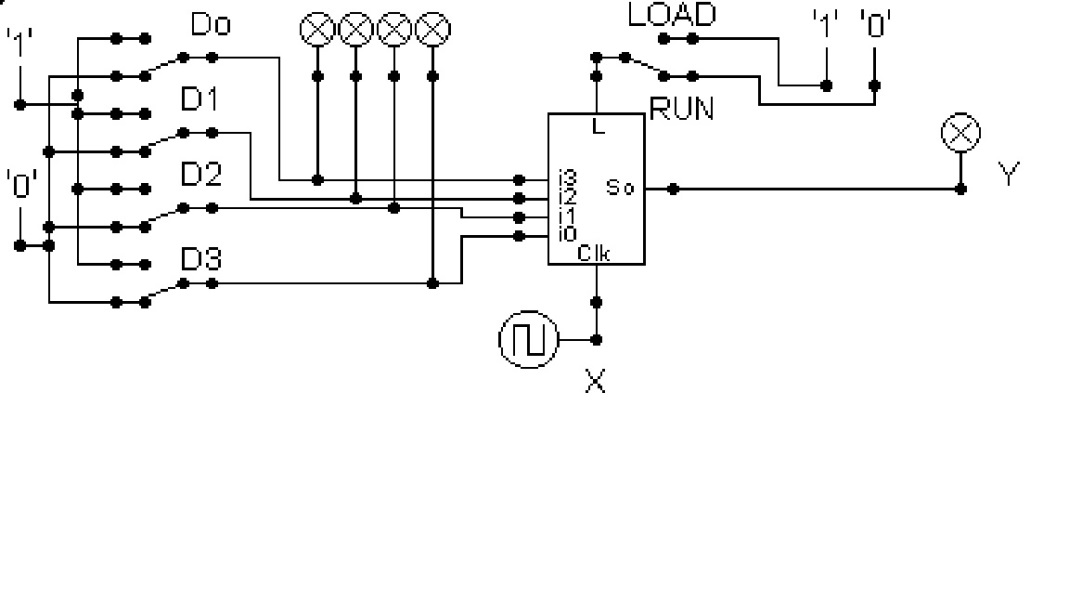


Describe what the circuit does. ***(15 marks)***  
Answer:   
In order to convert serial inputs to parallel outputs, a set of D flip-flops is required. The required amount of serial data (output) is exactly equal to the number of flip-flops. For example, in this case, a four-bit serial inputs (which is later converted into parallel outputs) requires four flip-flops to transmit. A diagram of a four-bit converter is shown above.

The circuit consists of one switch for serial input and one clock. The clock is connected parallelly to the four D Flip-Flops. The aim of connecting one clock to all four D flip-flop is to ensure that they all run at same speed. The first D flip-flop (Q3) has a serial input connected from the switch. The first D flip-flop (Q3) output is connected to the input of second D flip-flop (Q2). The second D flip-flop (Q2) output is connected to the input of third D flip-flop (Q1). The third D flip-flop (Q1) output is connected to the input of fourth D flip-flop (Q0). The first output (Q3), second output (Q2), third output (Q1), fourth output (Q0) is obtained simultaneously as they are parallel to each other.

Initially, the clock is 0 and the input is 1. At this condition, the outputs are 0,1,1,0 in Q0, Q1, Q2 and Q3 respectively. Altogether there are nine inputs in the circuit. And all the outputs with respect to the inputs and clock are given in the table and time diagram above.

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)***   
Answer:

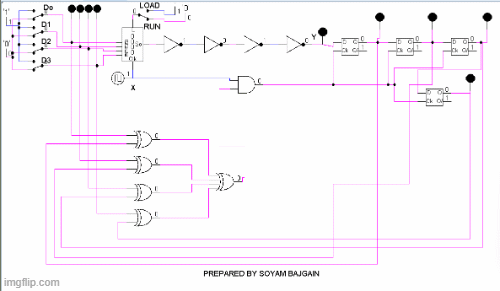
The shift register in which data is loaded onto the register in parallel format and the output is obtained in serial format is known as Parallel in Serial Out shift register, commonly known as PISO shift register.

In the circuit given above, the data is loaded into the register in a parallel format in which all the data bits enter their inputs at the exact same time, to the parallel inputs in D0, D1, D2, and in D3 of the register. A multiplexer is connected to the input of each flip-flop. The previous output and parallel data input are connected to the multiplexer's input, and the multiplexer's output is connected to the next flip-flop.  The data is then obtained sequentially in the normal shift-right mode from the register at Y (output) representing the data present at D0, D1, D2, and D3 respectively. What this circuit actually does is, This circuit takes the input individually in D0, D1, D2, D3 and gives the output from Y.

The working mechanism of Parallel In Serial Out shift register is based on the normal shift register, which means that this register is able to store and shift data entered to it. In this Parallel In Serial Out shift register, as the data are entered individually in parallel way. and it temporarily stores the data and displays it individually in a sequential way (one by one). The clock signal controls the activation of the output stages, which determines at what rate the data is being transferred.

Thus, the parallel to serial converter converts the parallel data into a serial data stream that can be stored in serial format as well as transmitted over a single communication line.

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)

Diagram, schematic

Description automatically generated