Name of the experiment:

Synchronous Data Transfer.

Objective:

- To design and implemented a digital circuit to transfer data serially.
- At the sender end the parallel data is converted to serial data to transfer the data to a receiver using a signale data line.
- At the receiver end the serial data will be reconstructed to its parallel form.
- Both sender and receiver circuits should be Synchronized using a single clock.

Apparatus:

- 4-bit Binary Counter, IC-74LS161
- 8-input Multiplexer, IC-74LS151
- 8-bit Serial-in-parallel-out Shift Register, IC-74LS164
- Octal D-type latch, IC-74LS373
- Hex Inverter, IC-74LS04
- Digital IC Trainer
- Wires.

Detail design methodology:

- First we took a 4-bit Binary Counter, 74LS161 and a 8-input Multiplexer, 74LS151. Then, we connected the inputs of the counter as the select line of the multiplexer, and took the inputs of the multiplexer from the switch. Besides, we have connected the clock to TTL.

- Secondly, we haven taken a shift register and an inverter. We have invented the Q_3 input of the Counter and connected it to the master register (MR). Then we connected the output of the multiplexer, and the clock of the counter as input of the shift-register, and also attached the master register as the input of the shift Register. As a result, the data will continue to run indefinitely.

- Finally, we took a D-type latch, 74LS373. Then, We have connected the output that came out of the Shift Register as the input of the latch. Besides, we have connected the output of the latch to the LED. That's how our synchronous data transfer working Procedure ended.

Pin Diagram:

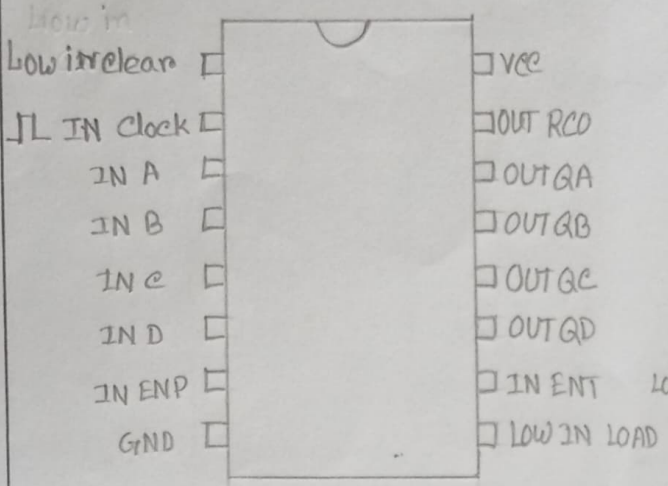


Fig: 4-bit Counter, 74LS161

Fig: 8-input Multiplexer, 74LS151

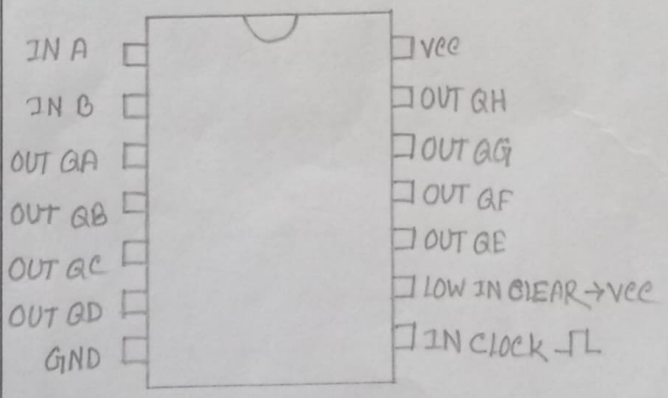


Fig: SIPO → Shift Register, 74LS164

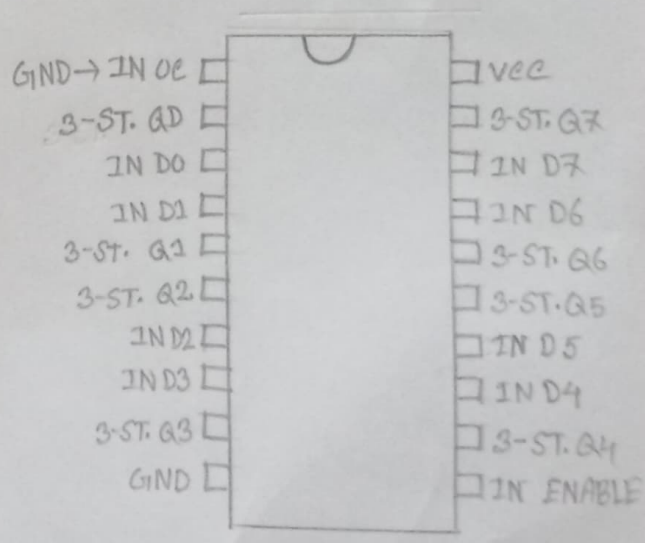
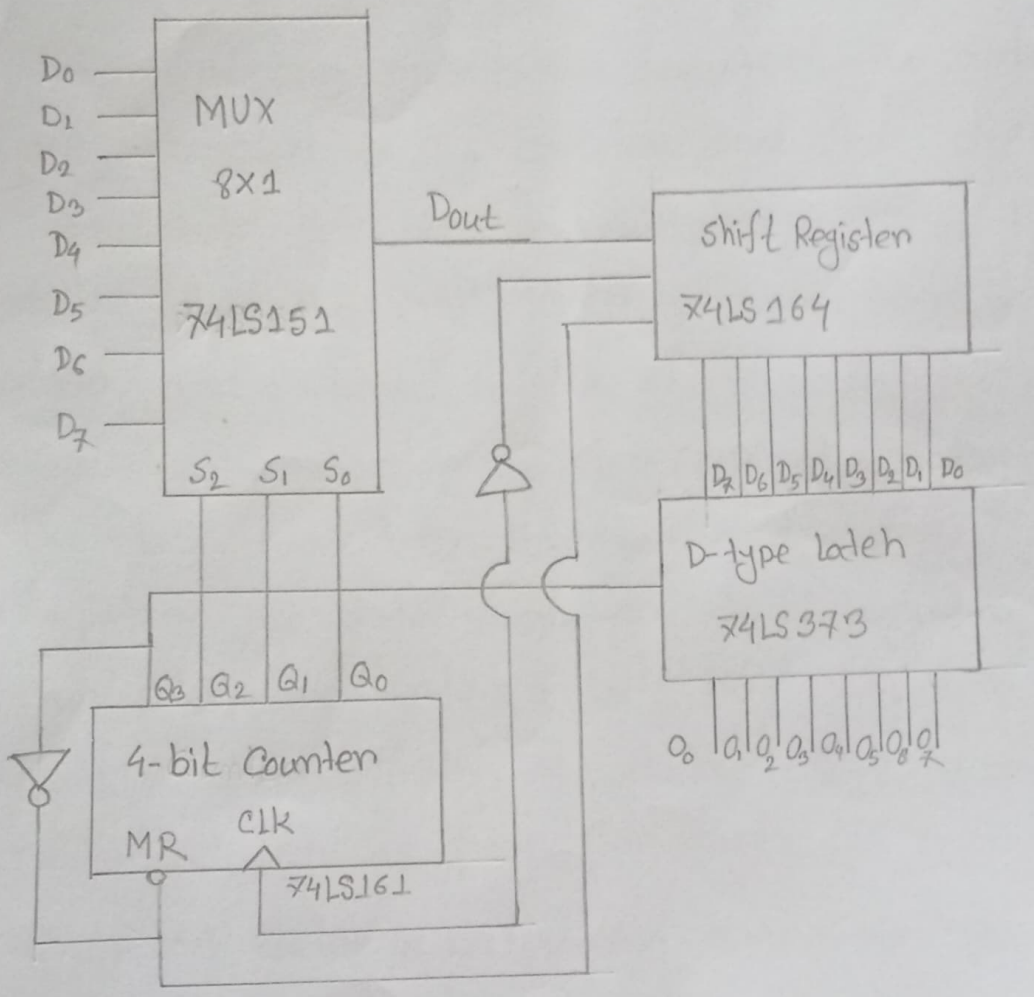


Fig: D-type latch, 74LS373

Circuit Diagram:



Result and Discussion:

In our circuit, whenever we give any data as Parallel input, we get the data correctly as parallel Output.

Looking at our circuit we see that the data first goes parallel to the counter and from there the same data goes to the multiplexer. Then, the parallel data from the multiplexer is converted into serial data and entered in the shift register. This is because the master resister (MR) of the counter and the clock have been inverted and entered the shift register. The data from the shift register is re-converted and entered into the D-type latch as Parallel data. Finally, the data is extracted from the D-type latch as parallel output.

Since our data is outputting correctly, we can say that our circuit is correct.

18.01.22

Precautions:

- The circuit should be connected carefully on the bread-board accurately.
- The ICs need to be properly positioned and pinned to the pin diagram.
- Power supply should not be provided before the end of Circuit implementation.

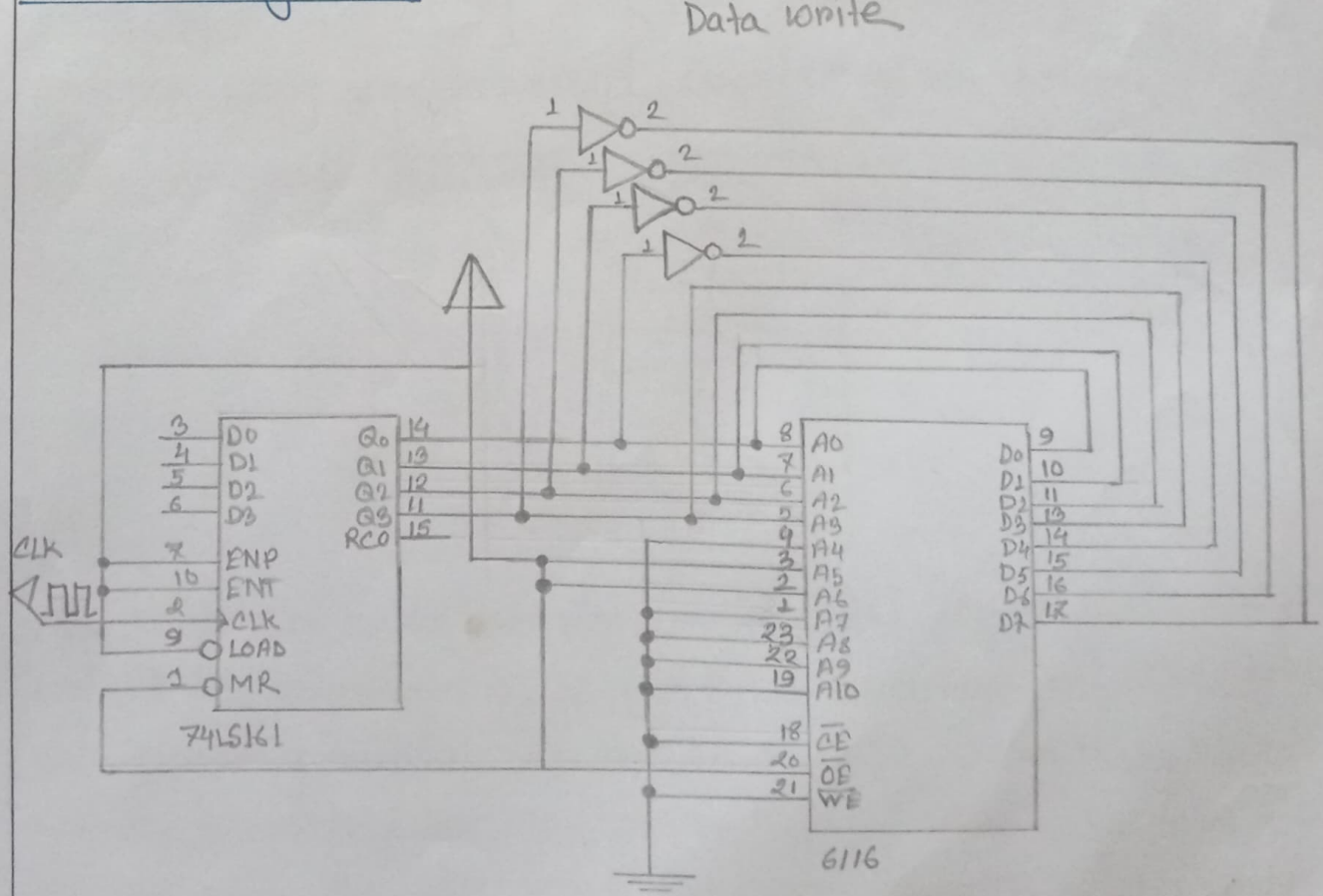
Name of the Experiment :

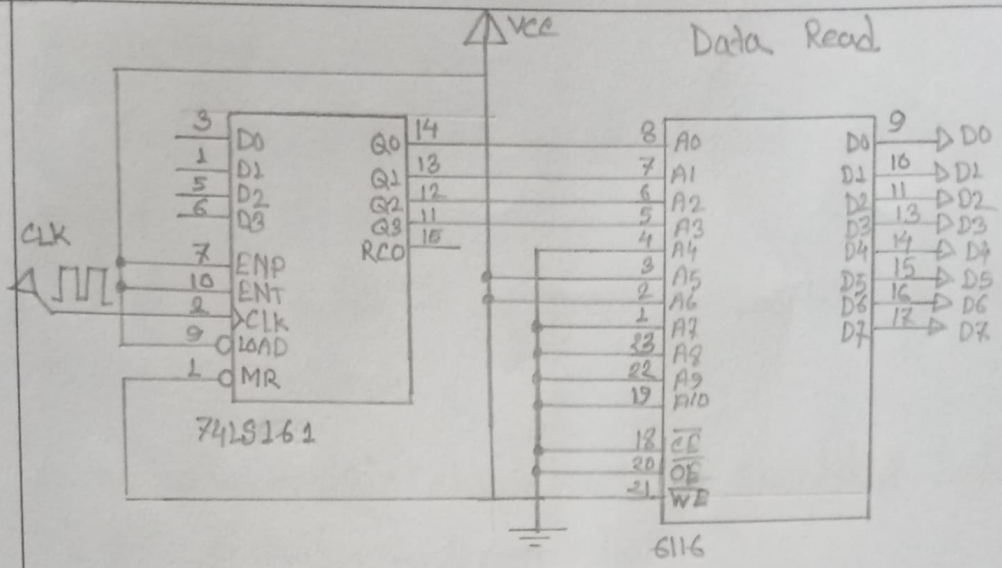
To design and implement a memory subsystem to store data in memory and then display the stored data into LED.

Apparatus:

- 4-bit Binary Counter, IC-74LS161
- Static RAM, 6116
- Hex Inverter, IC-74LS04
- Digital IC Trainer

Circuit Diagram:





Result and Discussion:

From this experiment we can say that the LEDs are showing the same result that we stored in RAM's write mode by the following data into corresponding memory address using synchronized counter given below:

Address	Data
60	F0
61	E1
62	D2
⋮	⋮
6F	0F

At the time of write operation, we scan the 'write' mode circuit with a full counter cycle just to ensure that the data was properly wrote. At the time of checking 'read' mode, we disconnected all the connection of RAM came from the the counter the RAM could not get any input. Finally the output was obtained exactly same as we desired.

Precautions:

- The circuit should be connected carefully on the bread-board.
- The ICs need to be properly positioned and pinned to the diagram.
- Power supply should not be provided before the end of the circuit implementation.
- As RAM is a volatile memory, the power supply should be on during the read/write operation.

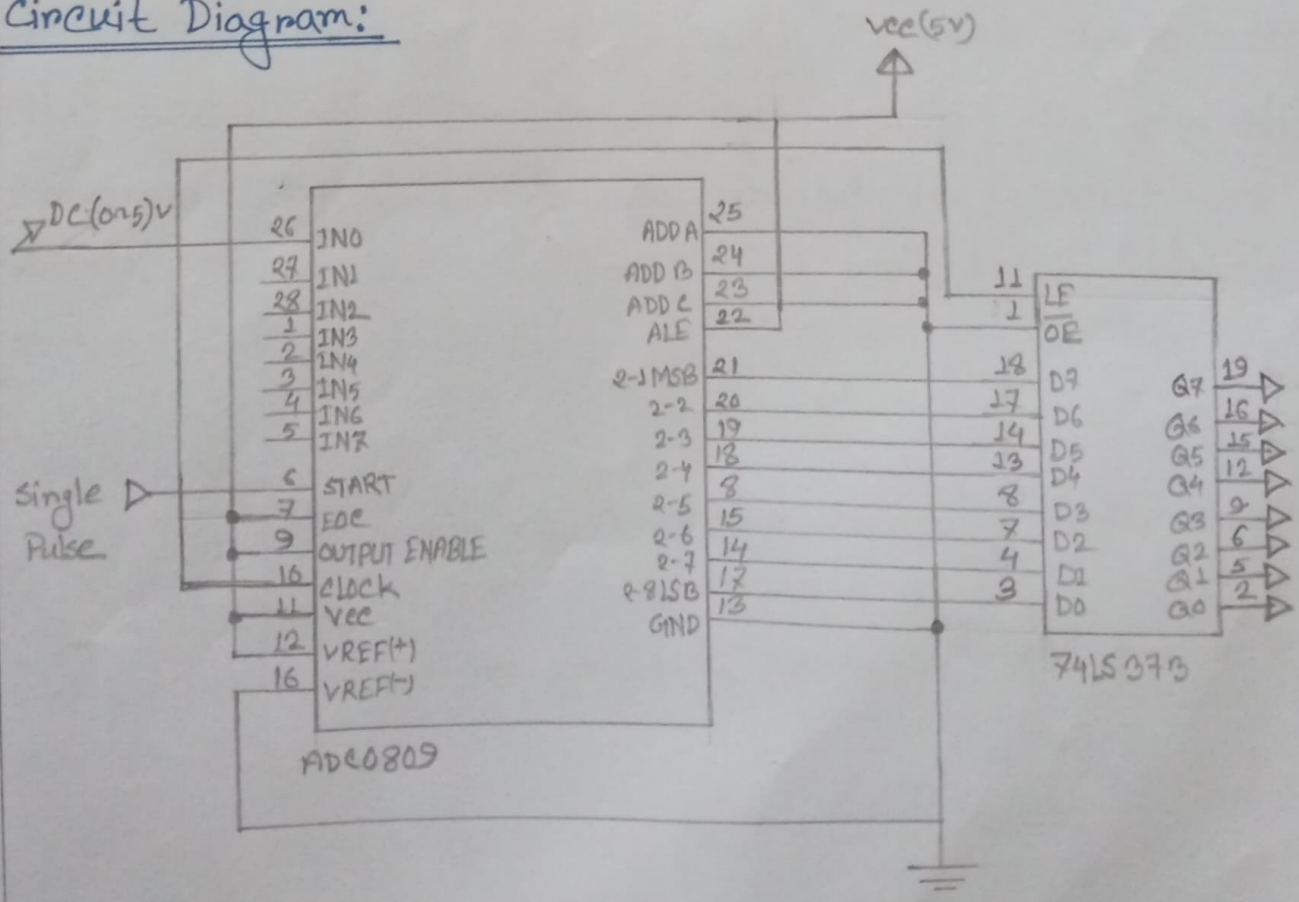
Name of the Experiment:

To design and implement a circuit to convert analog signal (Potential difference) into digital data by using an Analog to Digital Converter then store the data in a such latch and display the converted digital data using LED.

Apparatus:

- 8-bit µp compatible A/D Converter ADC0809
- Octal D-type latch, IC-74LS373
- Digital IC Trainer.

Circuit Diagram:



Result and Discussion:

From the experiment, we have observed that the analog signal was converted to digital data. The output LED's were showing larger binary number when we were changing the voltage using variable register, when we rotated the register variable from left to right we got the lowest value 0000 0000 and from right to left a maximum value 1111 1111. The experiment divided (0~5) v into $2^8 = 256$ seperate levels, and changing 0.01953 v, each time the binary sequeree was increased by 1. Through it was not possible to change exactly 0.01935 volt. In each single pulse, the EOC (End of Conversion) was sending the end of conversion signal which also varified the circuit was working perfectly. Therefore, the experiment had no error and successful.

Precautions:

- The circuit shoulb be conneeted carefully on the bread-board.
- The Ics need to be properly positioned and pinned to the pin diagram.
- Power supply should not be provided before the end of the circuit implementation.

Name of the Experiment:

To design and implement arithmetic circuits with selections variable S_0 & S_1 and operand A (4-bit), B (4-bit) & C_{in} that generates the following operations:

S_0	S_1	$C_{in} = 0$	$C_{in} = 1$
0	0	$F = A+B$	$F = A+B+1$
0	1	$F = A$	$F = A+1$
1	0	$F = B'$	$F = B'+1$
1	1	$F = A+B'$	$F = A+B'+1$

Apparatus:

- 4-bit Binary Full Adder, IC-74LS283
- Quad 2-input AND gate, IC-74LS08
- Quad 2-input OR gate, IC-74LS32
- Hex inverter, IC-74LS04
- Quad 2-input Exclusive-OR gate, IC-74LS86
- Digital IC Trainer

Truth Table:

Input				Output	
S_0	S_1	A_i	B_i	X_i	Y_i
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	0	1	0	0
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	0	1
1	0	0	1	0	0
1	0	1	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	0	1	0	0
1	1	1	0	1	1
1	1	1	1	1	0

k-Map:

X_i :

A_iB_i	$S_0'S_1'$	$S_0'S_1$	S_0S_1	S_0S_1'
$A_i'B_i'$	0	0	0	0
$A_i'B_i$	0	0	0	0
A_iB_i	1	1	1	0
A_iB_i'	1	1	1	0

$$X_i = A_iS_0' + A_iS_1$$

$$= (S_0' + S_1)A_i$$

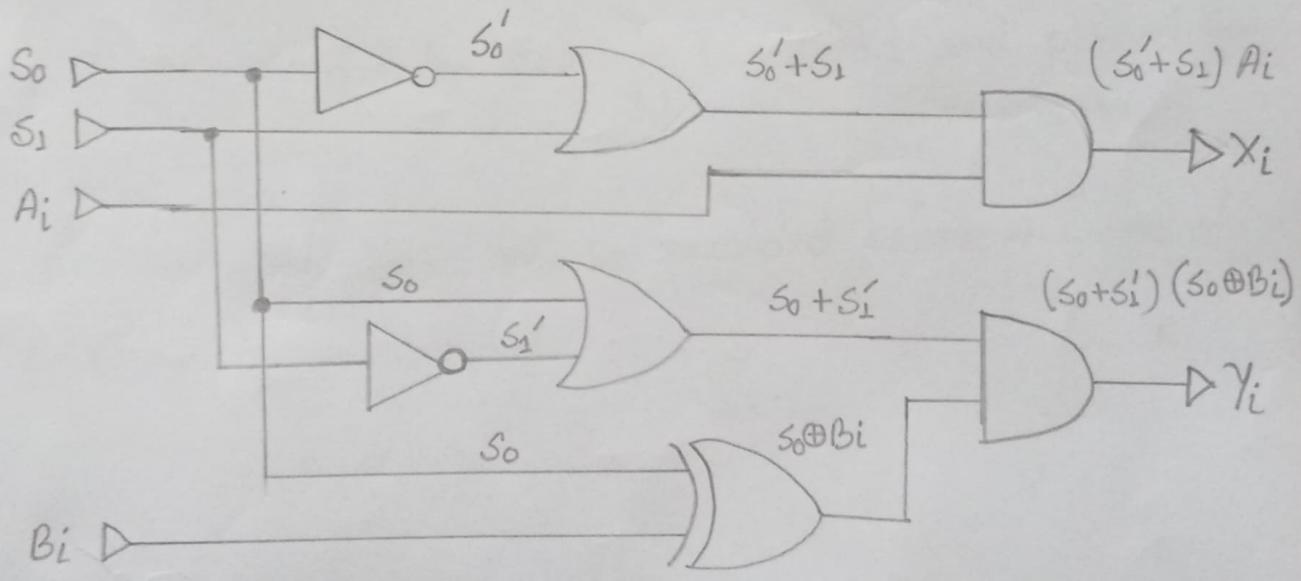
Y_i :

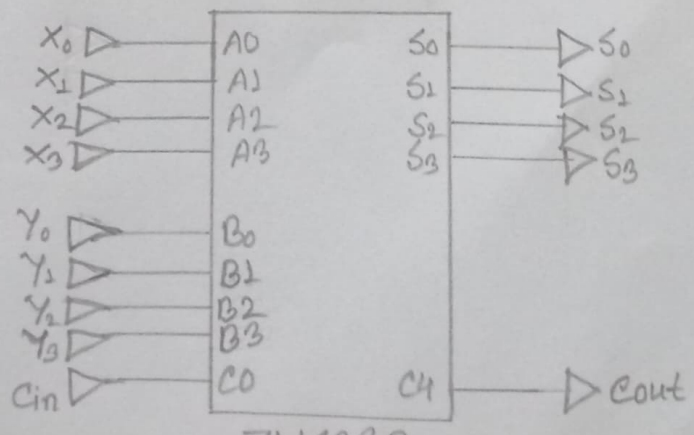
	$S_0'S_1'$	$S_0'S_1$	S_0S_1	S_0S_1'
$A_i'B_i'$	0	0	0	1
$A_i'B_i$	1	1	0	0
A_iB_i	1	1	0	0
A_iB_i'	0	0	0	1

~~$Y_i = B_iS_0' + B_i'S_0$~~

$$Y_i = (S_0 + S_1')(B_i \oplus S_0)$$

Circuit Diagram:





Result and Discussion:

The output of the circuit matches the truth table accurately. Thus the truth table has been verified. But for the weak connection between the wires, sometimes 1/2 of the input did not match. After holding them a little, everything was fine.

Precaution:

- The circuit should be connected carefully on the bread-board.
- The ICs need to be properly positioned and pinned to the pin diagram.
- Power supply should not be provided before the end of the circuit implementation.

Name of the Experiment:

To design and implement arithmetic circuits with selection variable S_0 & S_1 and operand A(4-bits), B(4-bits) & C_{in} that generates the following operations:

S_0	S_1	$C_{in}=0$	$C_{in}=1$
0	0	F=A	F=A+1
0	1	F=A-B-1	F=A-B
1	0	F=B-A-1	F=B-A
1	1	F=A+B	F=A+B+1

Apparatus:

- 4-bit Binary Full Adder, IC-74LS283
- Quad 2-input AND gate, IC-74LS08
- Quad 2-input OR-gate, IC-74LS32
- Hex inverter, IC-74LS04
- Quad 2-input Exclusive-OR gate, IC-74LS86
- Digital IC Trainer.

Truth Table:

Input				Output	
S_0	S_1	A_i	B_i	X_i	Y_i
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	1	0
0	0	1	1	1	0
0	1	0	0	0	1
0	1	0	1	0	0
0	1	1	0	1	1
0	1	1	1	1	0
1	0	0	0	1	0
1	0	0	1	1	1
1	0	1	0	0	0
1	0	1	1	0	1
1	1	0	0	0	0
1	1	0	1	0	1
1	1	1	0	1	0
1	1	1	1	1	1

k-map:

X_i:

	$S_0'S_1'$	$S_0'S_1$	S_0S_1	S_0S_1'
$A_i'B_i'$	0	0	0	1
$A_i'B_i$	0	0	0	1
A_iB_i	1	1	1	0
A_iB_i'	1	1	1	0

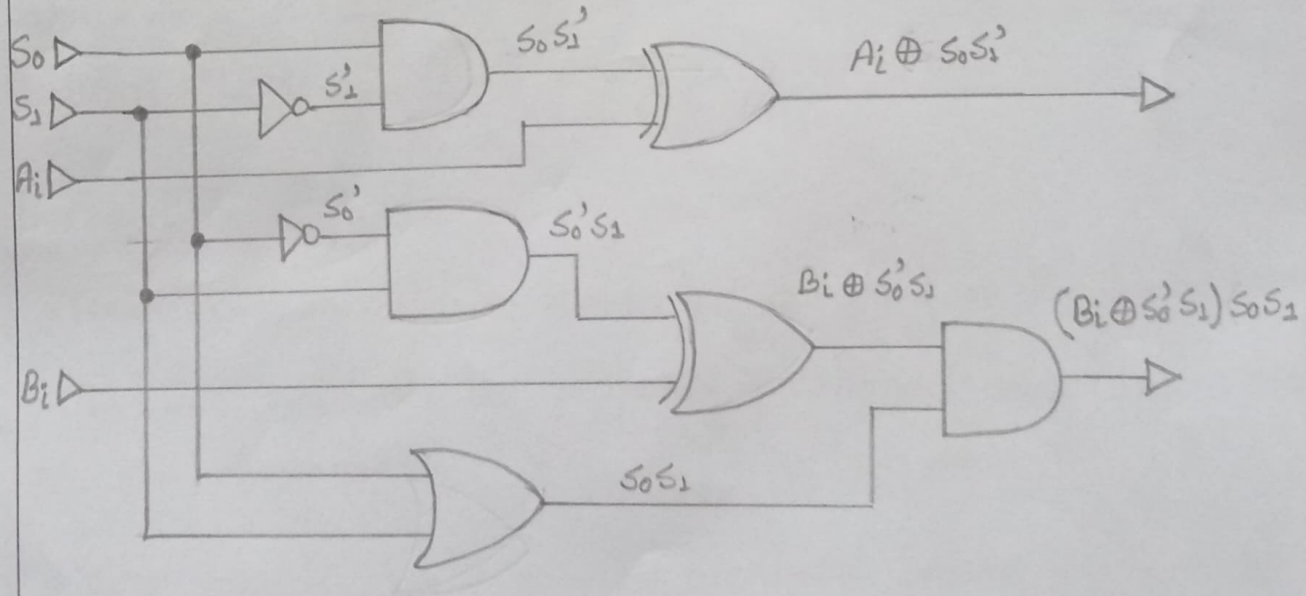
$$X_i = S_0S_1' \oplus A_i$$

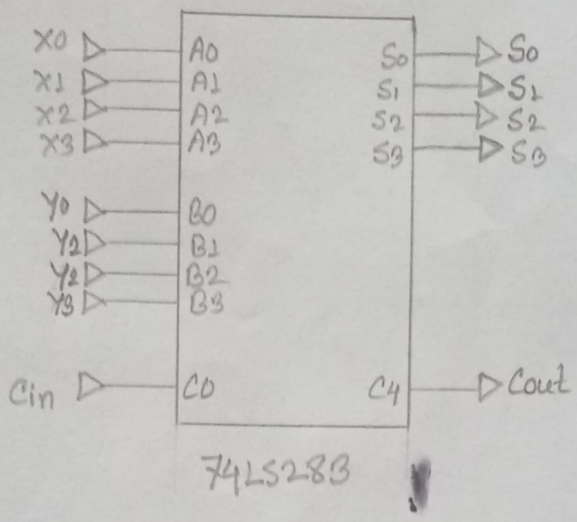
Y_i :

	$S_0'S_1'$	$S_0'S_1$	S_0S_1	S_0S_1'
$A_i'B_i'$	0	1	0	0
$A_i'B_i$	0	0	1	1
A_iB_i	0	0	1	1
A_iB_i'	0	1	0	0

$$Y_i = (S_0 + S_1)(S_0'S_1 \oplus B_i)$$

Circuit Diagram:





Result and Discussion:

The output of the circuit matches the truth table accurately. Thus the truth table has been verified. But for the weak connection between the wires, sometimes. 1/2 of the input did not match. After holding them a little, everything was fine.

Precaution:

- The circuit should be connected carefully on the bread-board.
- The IC's need to be properly positioned and pinned to the pin diagram.
- Power supply should not be provided before the end of the circuit implementation.