Experiment No : 01

Name of the experiment :

To design and implement a digital circuit for synchronous data transfer.

Objective :

I) At the sender end the parallel data is converted to serial data to transfer data using a single data line.

II) At the receiver end, the serial data will be reconstructed to its parallel form.

Theory :

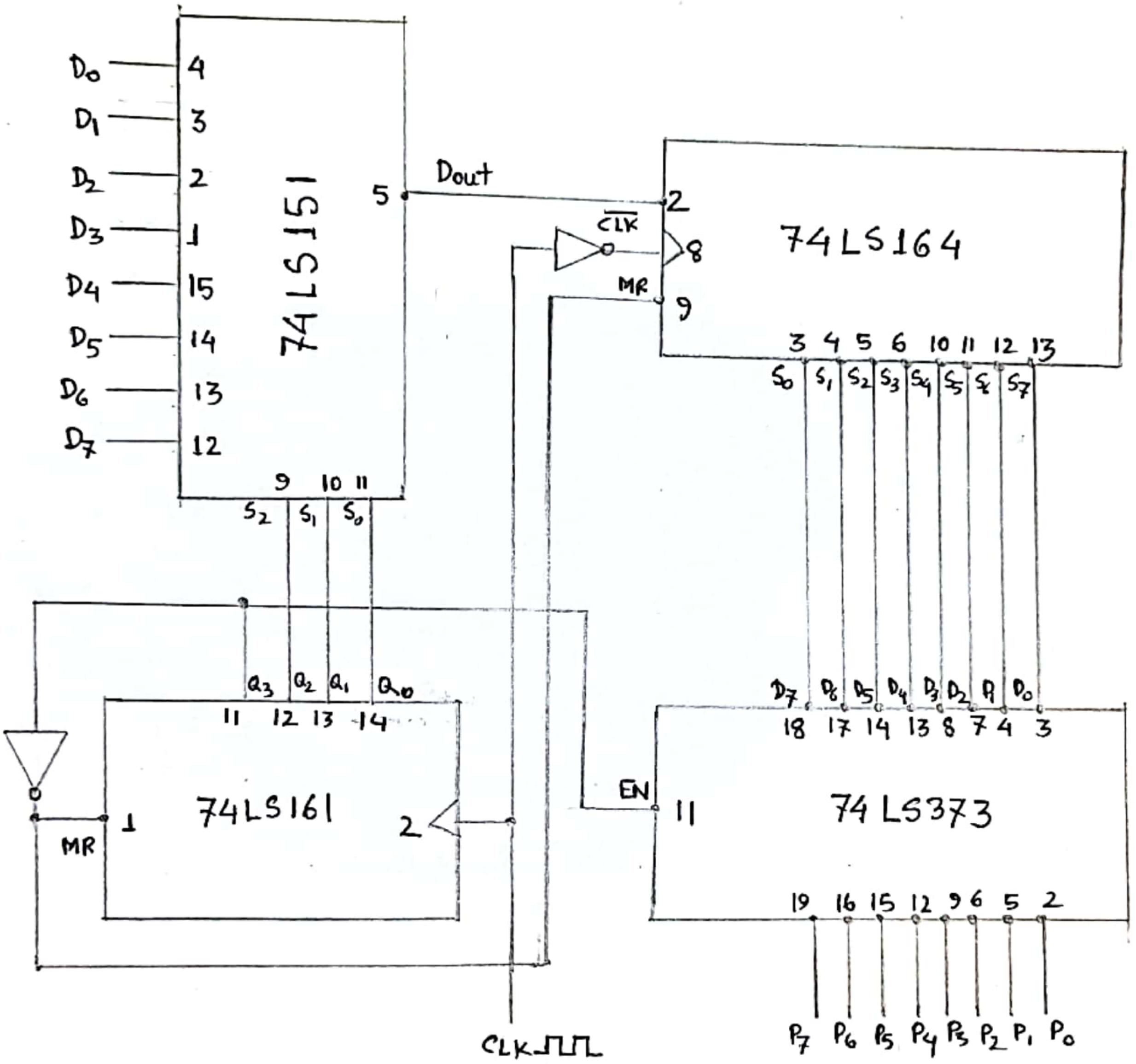
synchronous data transfer process or transmission is a process in which a set of data signals is accompanied by timing signals to ensure that the sender and the receiver are at the same stage and synchronized with one another by a single clock. In synchronous data transfer, the sending and receiving units are enabled with some clock signal. At the sender end the parallel data is converted to serial data to transfer the data to the receiver using a single line. At the receiver end, the serial data will be reconstructed to its parallel form.

Synchronous transmission mode is used when large amount of data need to transfer very quickly from one location to another location. The data is sent in blocks instead of individual characters. In this transmission mode, the master does not need any acknowledgement signal from the slave when the data is sent by the master to the slave but both the sender and receiver circuits should be synchronized using a single clock.

Apparatus

I) 4 bit Binary Counter (IC 74LS161)
II) 8 input Multiplexer (IC 74LS151)
III) 8 bit Serial-in-Parallel out shift register (IC 74LS164)
IV) D Latch
V) Digital Trainer Board
VI) Jumper wires etc.

Circuit Diagram :



Working Procedure:

I) At first, we take a 8 bit data signal to 8×1 MUX as input data where 3 bit select line was taken from the output of a 4 bit binary counter. The 4th bit was inverted to reset of output signal of the counter was inverted to reset the counter, where as the output of MUX entered into the 8-bit shift register as an input signal.

II) In the shift register, the inverted clock signal of counter was connected to the shift register as input clock. Then by selecting load-1 (high) we got the parallel output (8 bit) from 8 bit serial input.

III) Finally, the output of shift register was connected to D-latch to store data bit. In D-latch, 4th bit of the output signal was connected to the latch as enable. Thus, by connecting D-latch with shift register it can be observed that, the 8 bit output signal was displayed in parallel manner.

Result and Discussion :

In this experiment, parallel input was taken at the sender-end and it converted to serial data to transfer the data to the receiver end using a single data line. At receiver end, the serial data was converted to its parallel form.

From the experimental it can be seen that, the data was transferred successfully through synchronous data transmission and we have got the expected output. So, it can be said that the experiment was accurate.

Precaution

i) The IC ~~should~~ was checked before the implementation.

ii) V_{cc} and GND was connected properly.

iii) The circuit was implemented carefully.

iv) Power supply was connected to the circuit after the implementation of the circuit.

Experiment No: 02

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:

i) 4-bit Binary Counter, 74LS161

ii) static RAM, 6116

iii) Hex Inrverter, 74LS04

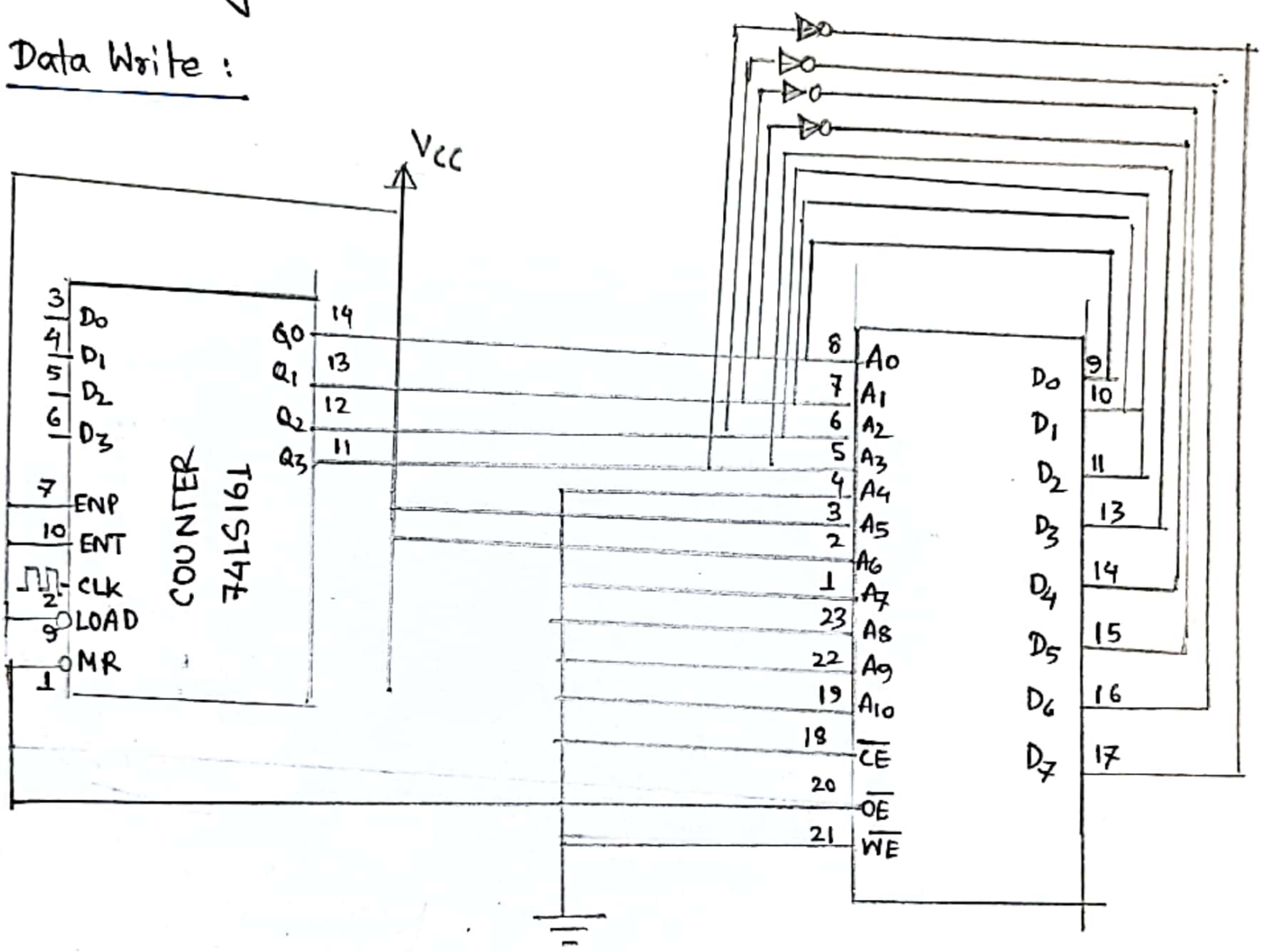
iv) Digital IC Trainer

Objective: The objective of this experiment is to design a circuit where a static RAM store data Following the table below:

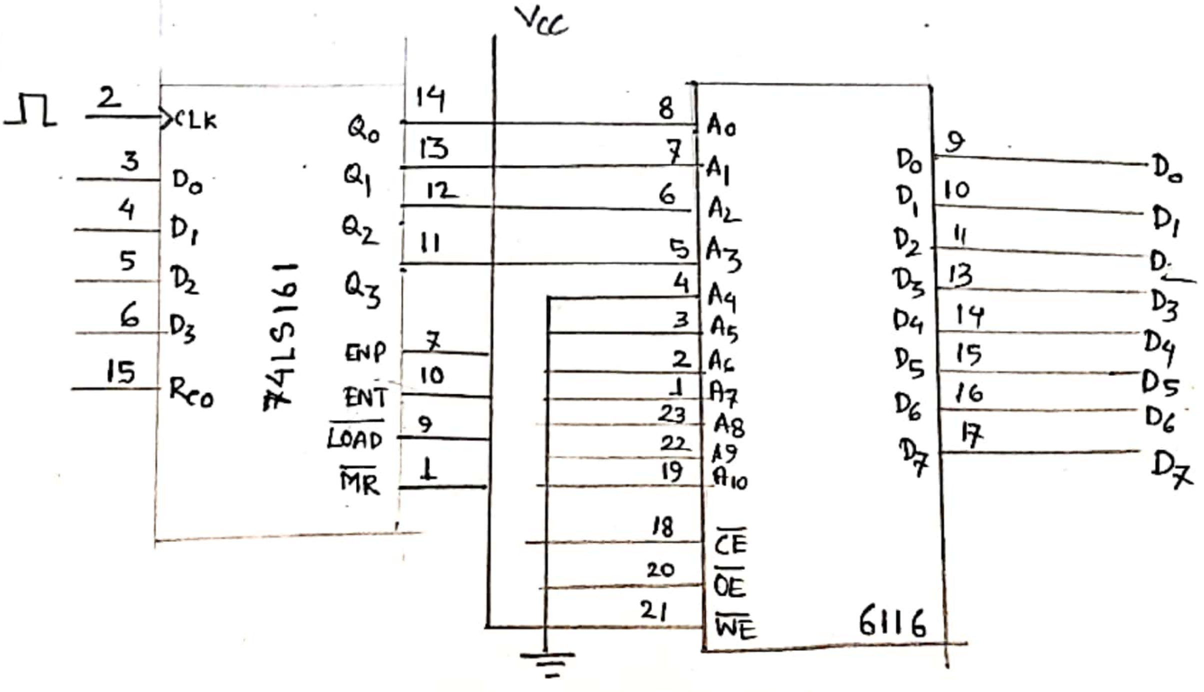
Address	Data
60	F0
61	E1
62	D2
⋮	⋮
6F	0F

Circuit Diagram:

Data Write :



Data Read:



Experimental Data:

In `Write' mode we wrote FO in the address 60;61,62.... 6F respectively in E1, D2, ...OF. In 'Read' Mode, the data was stored in RAM and thus the output was also the same as the data we wrote before. The output data was displayed in the LEDs which followed the given sequence.

Result and Discussion:

From the experiment, we can observe that the output shown in LED was absolutely same as the binary value that we have stored in RAM in write mode. We wrote data FO, E1, D2.... OF in addresses 60, 61, 62, ...6f respectively. At the time of write operation, we ran the write mode circuit with 2 full counter cycle just to ensure the data was properly wrote. At the time of checking 'read mode', we disconnected all the input connections of RAM that came from the counter so that the RAM could not get any input.

Finally, the output was obtained exactly same as we desired. Therefore, the circuit was error-free and correct.

Precautions:

i) ICs should be connected to VCC and GND carefully and it was strictly followed.

ii) In 'write mode', the circuit was given enough time to properly write data.

III) The input lines were disconnected during the checking of output mode.

IV) As RAM is a volatile memory, the power supply should be on during the read/write operation.

Experiment No: 03

Name of the Experiment: To design and implement a circuit to convert analogue signal to digital data by using an Analogue to Digital Converter and then store the data in a latch and display the converted digital data using LED.

Apparatus:

i) 8 bit μP compatible A/D converter ADC0809

ii) Octal D type Latch (74LS373)

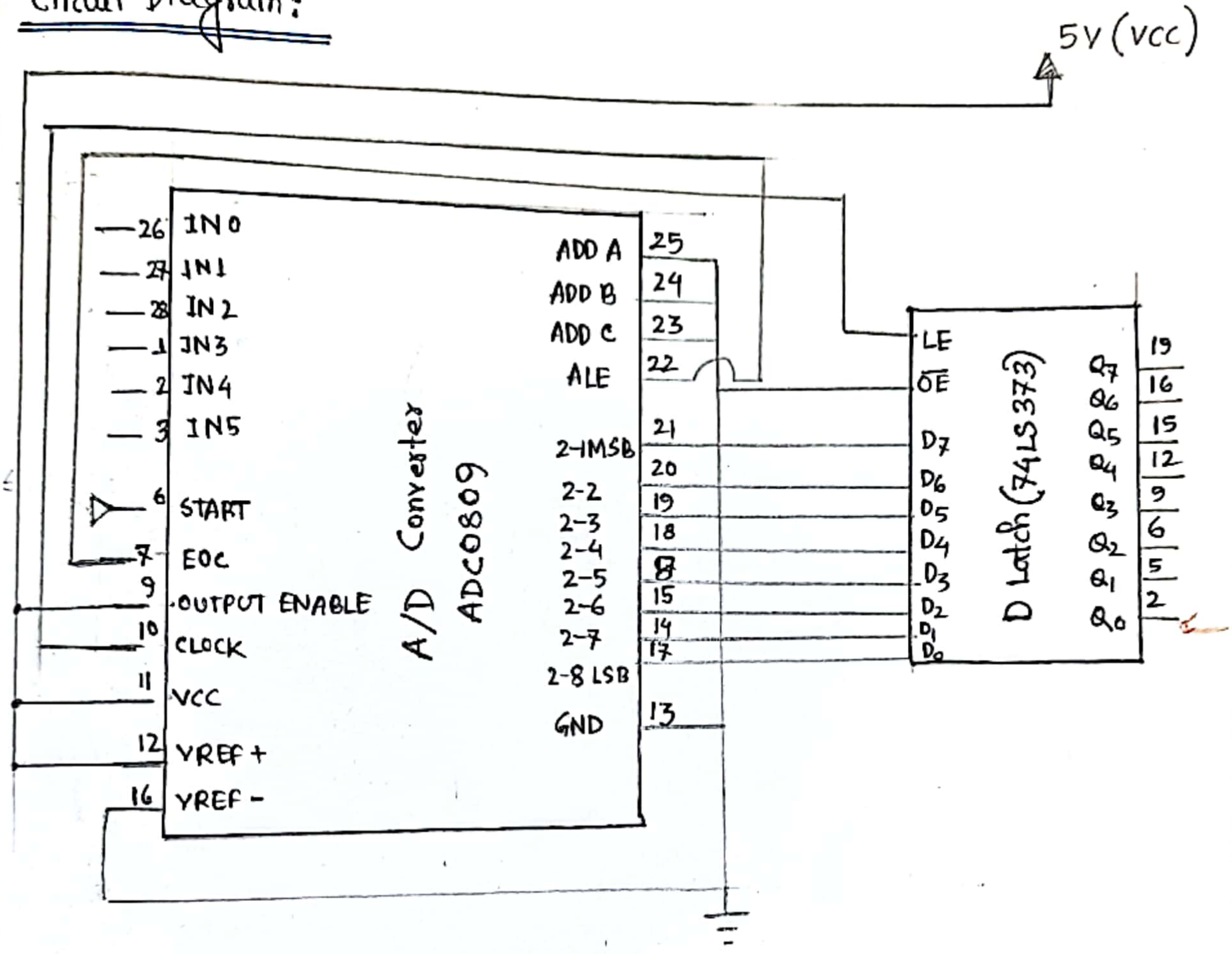
iii) Digital IC Trainer

iv) Variable Resistor

Objective:

The objective of this experiment is to design a circuit with controls to initialize the conversion, decode port if multiple analogue input lines available on ADC IC and required to digitalize multiple analogue input signals and store digital signal in a latch.

Circuit Diagram:



Experimental Data :

When the voltage was lowest, then all 8-bits were turned off. When the voltage was given full with variable resistor (in this case, 5 volt), all the LEDs were turned on. In the interval, if we turned the regulator from left to right or clockwise direction, it showed an increasing sequence of binary 8 bits.

Result and Discussion:

From the experimental data, we have observed that the analogue signal was converted to digital data. The output LEDS were showing larger binary number in 8 LEDs or bits when we were increasing the voltage using variable resistor. The output was 0000 0000 when we put lowest voltage (0 volt) and it was 1111 1111 when we put the highest voltage (5 Volt). The experiment was dividing 0V to 5V into $2^8 = 256$ seperate levels. So, for changing 0·01953V, each time the binary sequence was increased by 1. Though it was not possible to change exactly 0·01953 Volt and to check if the sequence was increased by 1 with variable resistor, the experiment was error-free and successful. In each single pulse, the EOC (End of Conversion) was sending an end of conversion signal which also verfied the circuit was working perfectly. Therefore, the experiment had no error and successful.

Precautions

I) The wires, breadboard and ICs were checked before the implementation of the circuit.

II) V_{cc} and GND should be connected properly.

III) Variable resistor was checked before application and EOC was checked if it gives a signal in each single pulse or not.

IV) Power supply was turned on after the implementation of the circuit.

Name of the Experiment :

To design and implement arithmatic circuits with selection variable S_0 and 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	$F = A + B$	$F = A + B + 1$
0	1	$F = A$	$F = A + 1$
1	0	$F = B'$	$F = B' + 1$
1	1	$F = A + B'$	$F = A + B' + 1$

Apparatus Required:

1. 4 bit Binary Full Adder (74LS283)
2. Quad 2-input AND Gate (74LS08)
3. Quad 2-input OR Gate (74LS32)
4. Hex Inverter (74LS04)
5. Quad 2-Input Exclusive-OR Gate (74LS86)
6. 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 Simplification :

For Xi,

S_0S_1 \ A_iB_i	00	01	11	10
00	0	0	1	1
01	0	0	1	1
11	0	0	1	1
10	0	0	0	0

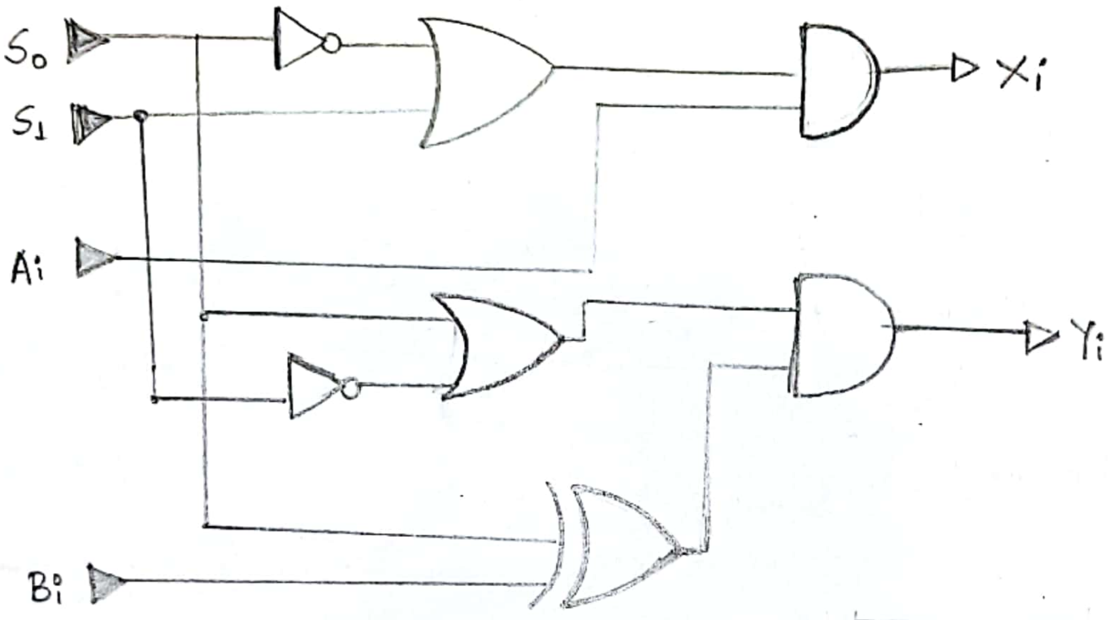
$X_i = \bar{S_0} A_i + S_1 A_i = (\bar{S_0} + S_1) A_i$

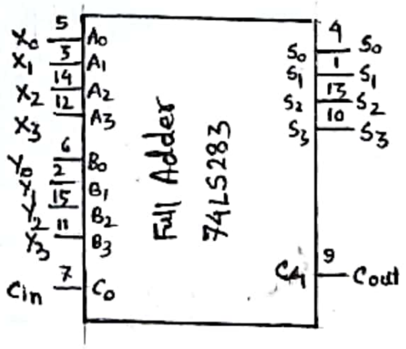
For Yi,

S_0S_1 \ A_iB_i	00	01	11	10
00	0	1	1	0
01	0	0	0	0
11	1	0	0	1
10	1	0	0	1

$$
\begin{aligned}
Y_i &= \bar{S_0}\bar{S_1} B_i + S_0\bar{B_i} \\
&= \bar{S_0}\bar{S_1}B + S_0\bar{B}(\bar{S_1}+1) \\
&= \bar{S_0}\bar{S_1}B + S_0\bar{S_1}\bar{B} + S_0\bar{B} \\
&= \bar{S_0}\bar{S_1}B + S_0\bar{S_1}\bar{B} + S_0S_0\bar{B} + S_0\bar{S_0}B \\
&= \bar{S_0}B(S_0+\bar{S_1}) + S_0\bar{B}\ (S_0+\bar{S_1}) \\
&= (S_0+\bar{S_1})\ (\bar{S_0}B+S_0\bar{B}) \\
&= (S_0+\bar{S_1})\ (S_0 \oplus B_i)
\end{aligned}
$$

Circuit Diagram :





Result and Discussion :

From the experiment, we can observe that the outputs are as expected and given below:

S_0	S_1	A	B	Cin = 0	Cout = 1
0	0	0100	0011	0111	1000
0	1	0111	0011	0111	1000
1	0	1010	1111	0000	0001
1	1	0101	0001	1011	1100
0	0	1011	0001	1100	1101

The arithmatic logic unit can perform four different predefined operations successfully according to the given table. The select lines were used to change the mode of operation. Each time, X_i and Y_i generates the adder which was the A (4 bit) and B (4 bit) for the adder according to the select line. Adder modified A and B according to the select line. Adder and Y_i and showed on LED. simply add the given ~~X_i~~ and Y_i and showed on LED.

Precaution :

I) The ICs were checked before implementation

II) The circuit should be implemented carefully.

III) Vcc and GND were connected properly.

IV) Power supply was turned off until the circuit implementation was finished.

Experiment No : 05

Name of the Experiment : To design and implement an arithmatic control circuit.

Objective :

We have to design and implement an arithmatic control circuit with selection variable S0 and S1 and operand A (4bits), B (4 bits) and 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

i) 4 bit binary full Adder (74LS283)

ii) 2 input AND gate, (74LS08)

iii) 2 input OR gate (74LS32)

iv) Hex Inverter (74LS04)

v) 2 input XOR gate (74LS86)

vi) 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

Kmap Simplification :

For X_i,

S_0S_1 \ A_iB_i	00	01	11	10
00	0	0	1	1
01	0	0	1	1
11	0	0	1	1
10	1	1	0	0

$$X_i = \overline{S_0} A_i + S_1 A_i + S_0 \overline{S_1}\, \overline{A_i}$$

$$= S_0 \overline{S_1}\, \overline{A_i} + (\overline{S_0} + S_1) A_i$$

$$= S_0 \overline{S_1}\, \overline{A_i} + \overline{S_0 \overline{S_1}}\, A_i$$

$$= (S_0 \overline{S_1}) \oplus A_i$$

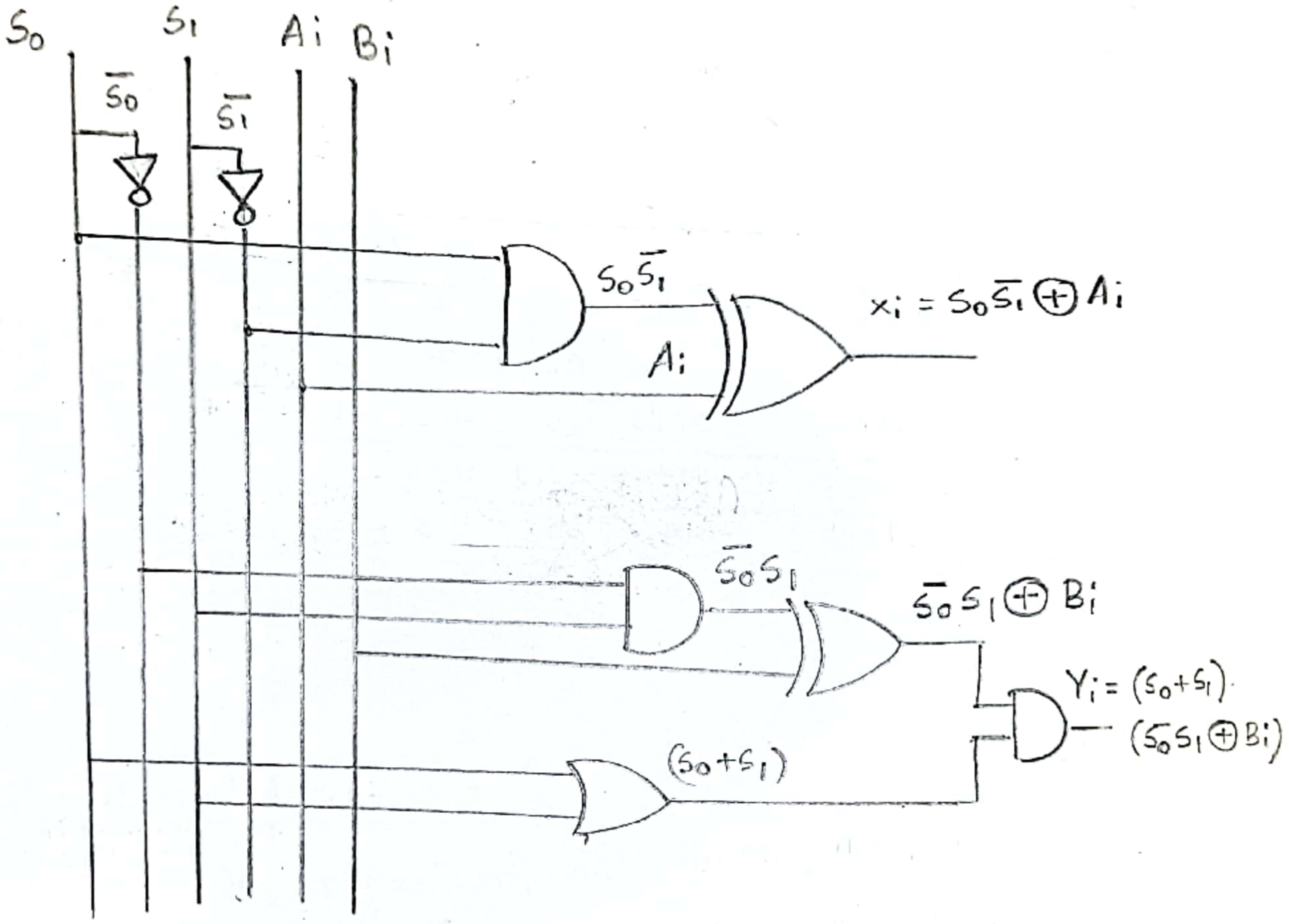
For Y_i,

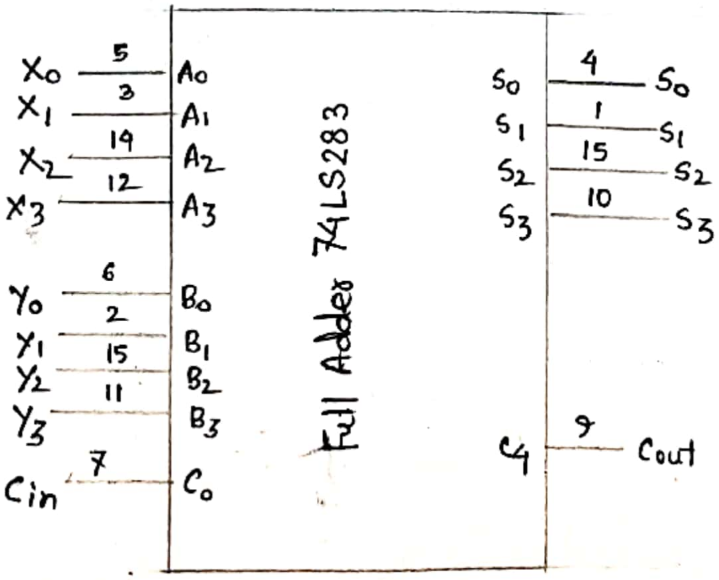
S_0S_1 \ A_iB_i	00	01	11	10
00	0	0	0	0
01	1	0	0	1
11	0	1	1	0
10	0	1	1	0

$$Y_i = \overline{S_0} S_1 \overline{B_i} + \overline{S_0} B_i$$

$$= (S_0 + S_1)\ (\overline{S_0} S_1 \oplus B_i)$$

Circuit Diagram:





Experimental data :

From the experiment, we have got the same output as truth table :

S_0	S_1	A	B	$C_{in} = 0$	$C_{in} = 1$
0	0	0101	1100	0101	0110
0	1	0000	1101	0100	0101
1	0	0011	0101	1001	1010
1	1	0100	1000	1100	1101

Result and Discussion:

From the experiment data table, we can observe that the output came as expected. The ALU can successfully performed 4 different operations using 2 select lines. Thus, the experiment has no error.

Precaution :

1) The IC's were checked before implementation
11) The circuit should be implemented accurately.
111) Power supply was turned on after full circuit implementation.