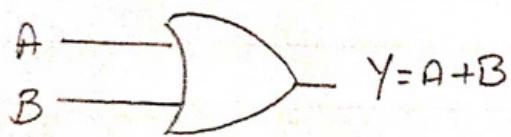


Qpedia.live

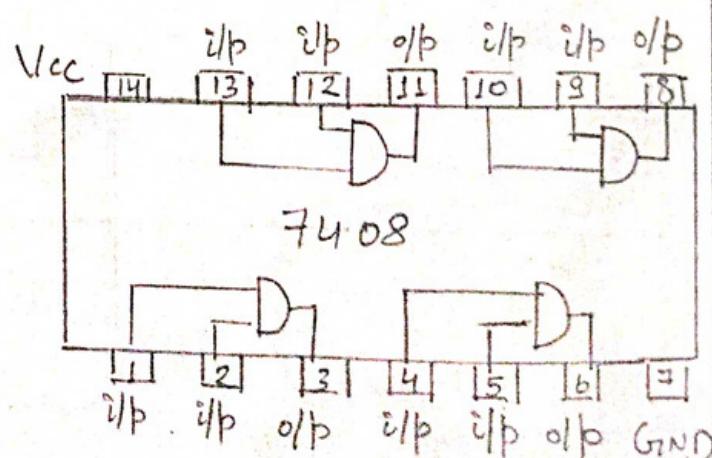
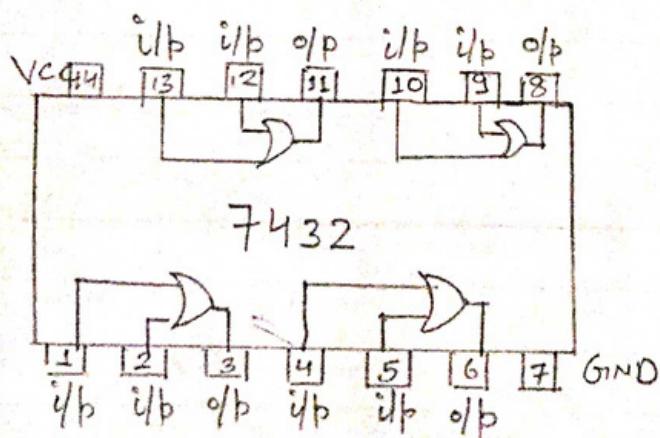
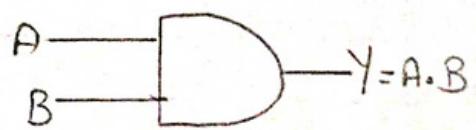
* OR Gate →

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1



* AND Gate →

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1



Experiment No. 1

* Objective →

To study and understand nomenclature pin configuration and verification of truth table of logic gates using transistor logic ICs.

* Apparatus Required →

- 1. Breadboard Trainer kit
- 2. IC - 7432 (OR)
- 3. IC - 7408 (AND)
- 4. IC - 7404 (NOT)
- 5. IC - 7400 (NAND)
- 6. IC - 7402 (NOR)
- 7. IC - 7486 (EX-OR)
- 8. IC - 74266 (EX-NOR)
- 9. Connecting wires

* Theory →

Logic Gate is a circuit with one output and two or more input channels to O/p signals occurs only for certain combination of i/p signals. They are used to perform various computer functions.

* OR Gate (7432) :-

It can have two or more i/p and a single o/p . It is defined as whenever anyone or all the i/p s are high then o/p is high as well otherwise low
 mathematically
$$y = A + B$$
.

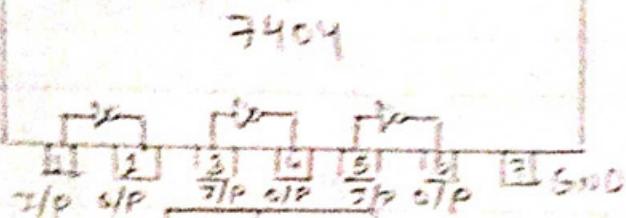
* AND Gate (7408) :-

It can have two or more i/p and a single o/p . It is

* NOT Gate \rightarrow



7404
TIP S/P TIP o/p TIP S/P
Vcc GND GND GND GND

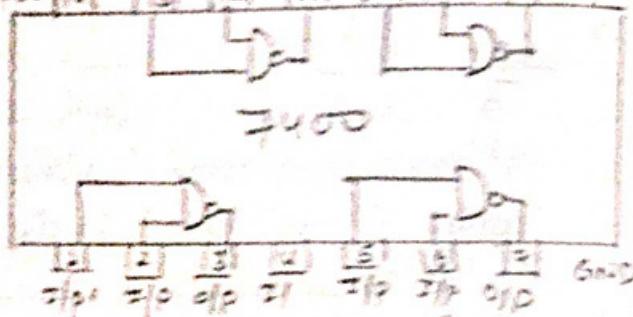


A	Y
0	1
1	0

* NAND Gate \rightarrow

$$A \rightarrow D \rightarrow Y = \bar{A}$$

B \rightarrow D \rightarrow Y = $\bar{A} \cdot \bar{B}$
7400
TIP S/P TIP o/p TIP S/P
Vcc GND GND GND GND



A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

* NOR Gate \rightarrow



7402
TIP S/P TIP o/p TIP S/P
Vcc GND GND GND GND



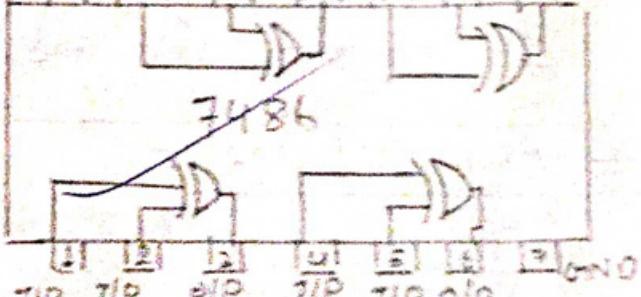
7402

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

* EX-OR Gate \rightarrow

$$A \rightarrow D \rightarrow Y = \bar{A}B + A\bar{B}$$

B \rightarrow D \rightarrow Y = $\bar{A}B + A\bar{B}$
7486
TIP S/P TIP o/p TIP S/P
Vcc GND GND GND GND



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

defined as when all inputs are high output is high otherwise low. mathematically $y = A \cdot B$

* NOT Gate (7404) :-

The simplest form of logic circuit is the inverter or not gate. It can have one input and one output. It is defined as when input is low output is high or vice-versa. mathematically $y = \bar{A}$.

* NAND Gate (7400) :-

The term NAND is a contraction of expression NOT AND. A NAND gate, therefore is AND AND gate followed by a inverter. If only both of input are high then only output mathematically $y = \bar{A} \cdot \bar{B}$.

* NOR Gate (7402) :-

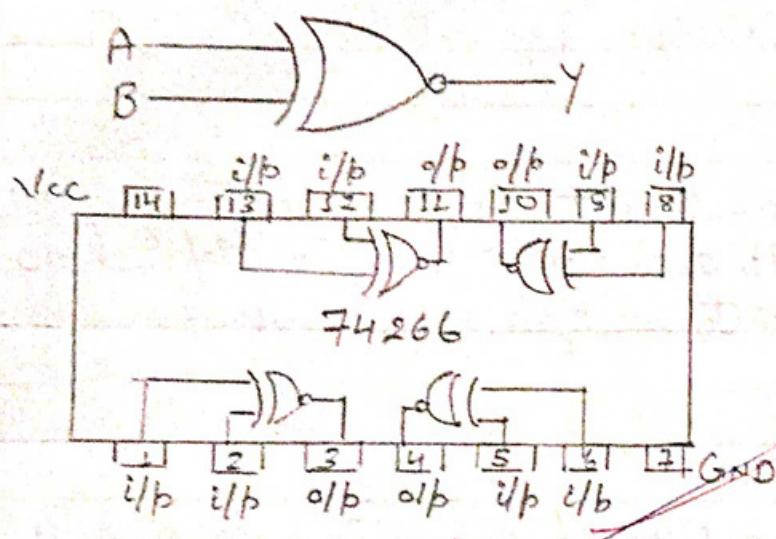
The NOR gate is an improved logic element used for implementing decision making logic function. The term NOR is a contraction of expression NOT OR. It is defined as whenever all inputs are low output is high otherwise it is always low.

mathematically $y = \bar{A} + \bar{B}$.

* Ex-OR Gate (7486) :-

The Ex-OR gate operation is widely used in a digital circuit. It is not a basic operation and can be performed using the basic gates AND, OR and NOT or universal gates NAND or NOR. When both inputs are same mathematically $y = A\bar{B} + \bar{A}B$.

* Ex-NOR Gate →



A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

* Ex-NOR Gate (74266) :-

The Ex-NOR gate is not a basic operation and can be performed using basic gates - AND, OR and NOT as universal gates NAND or NOR. In this gate output is high when both inputs are same otherwise low mathematically.

$$Y = AB + \overline{AB}$$

* Observations :-

We can observe all the truth table of all logic gates.

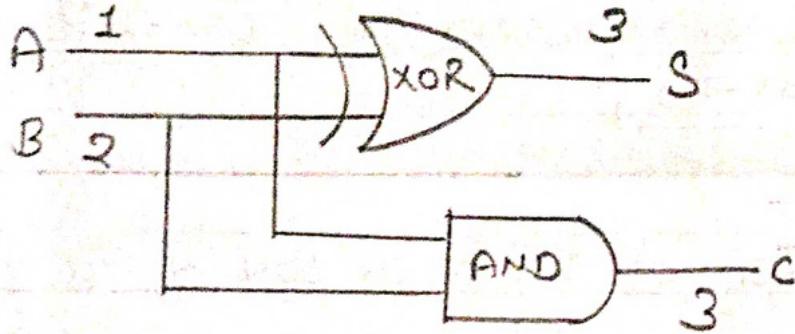
* Result :-

Truth table of various logic gates (OR, NOR, NOT, AND, NAND, Ex-OR, Ex-NOR) are verified.

* Precautions :-

1. All connections should be according to circuit.
2. All connections must be tight.
3. Reading must be taken carefully.
4. Switch off power supply after completing the experiments.

Neelima
09/10/23



* Circuit Diagram Of Half Adder

Input		Output	
A	B	S(Sum)	C(Carry)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

* Truth Table

Experiment No. 2* Objective →

To study how arithmetic operation like addition is performed at bit level using basic hardware circuits.

* Apparatus Required →

- | | |
|-----------------------------|-------------------------|
| 1: Bread board Brain or kit | 4: IC - 7432 (OR Gate) |
| 2: IC - 7486 (X-OR Gate) | 5: IC - 7404 (NOT Gate) |
| 3: IC - 7408 (AND Gate) | 6: Connecting wire |

* Half Adder →

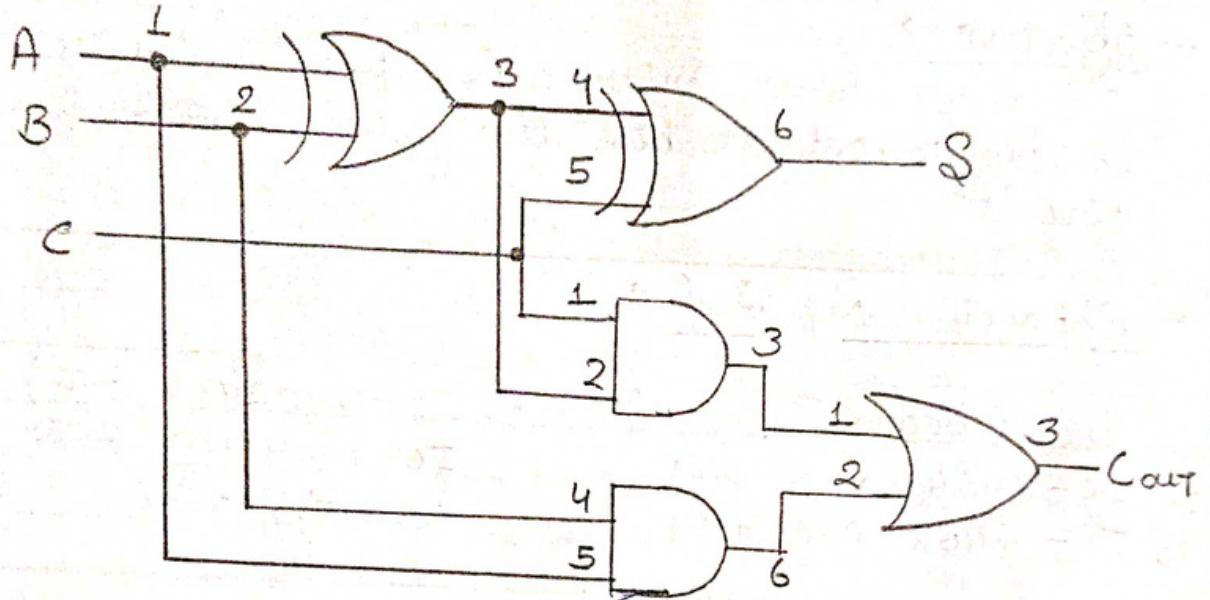
The half adder adds two single binary digits A and B it has two outputs, sum(S) and carry(C). The carry signal represents an overflow into the next digit of a multi-digit addition, incorporates an XOR gate for S and an AND gate for C. with the addition of an OR gate to combine their carry outputs two half adders can be combined to make a full adder.

* Boolean Expression →

Input are A and B

$$\text{Then output } S = \overline{A}B + A\overline{B} = A \oplus B$$

$$C = AB$$



* Circuit Diagram

* Observation Table →

Input			Output	
A	B	Cin	S (Sum)	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

* Full Adder →

Adder circuit is a combinational digit circuit that is used to adding two number. A typical Adder circuit produces a sum bit (denoted by S) and carry bit (denoted by C) as the output.

A full Adder is a combinational circuit that forms the arithmetic sum of input. It consists of three inputs and two outputs.

A full adder is useful to add three bits at a time but a half adder cannot do so.

In full adder is useful to add three bits at a time but a half adder cannot do so. In full adder sum output will be taken from X-OR gate, carry, output will be taken from OR gate.

* Boolean Expression →

Inputs are A and B and C_{in}

Then output

$$S = A \oplus B \oplus C_{in} \text{ AND}$$

~~$C_{out} = AB + AC + BC$~~

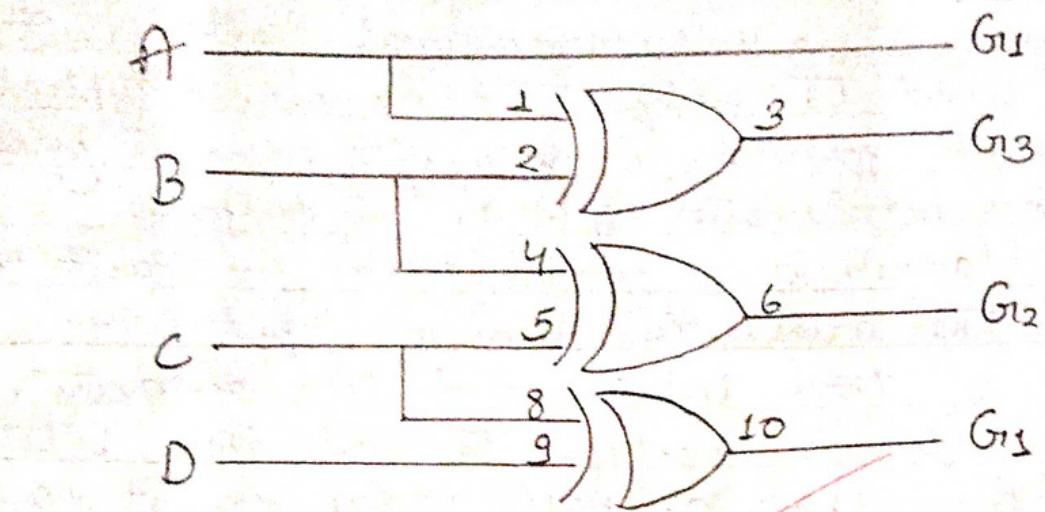
* Result →

All outputs verified the result of Truth table.

* Precautions →

1. All the connection should be tight and proper
2. Handle the IC's carefully
3. Check the connection once again before switching on.
4. Switch off trainers kit after experiment.

* Circuit Diagram of Gray-to-Binary Conversion →



* Observation Table →

S.No.	Input(4 Bit Binary Code)	Output(4 Bit Gray code)
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100

Experiment No. 3

* Objective →

To convert Binary code to Gray code and Gray code to Binary code.

* Apparatus Required →

- 1: Bread board
 - 2: IC - 7486
 - 3: connecting wires etc.

* Theory →

* Binary to Gray Conversion →

For Binary code to Gray code conversion, follow the steps below

Step 1 Record the msB as it is

Step 3 Add (Apply X-OR) the previous bit to the next bit, recording the sum and neglecting the carry.

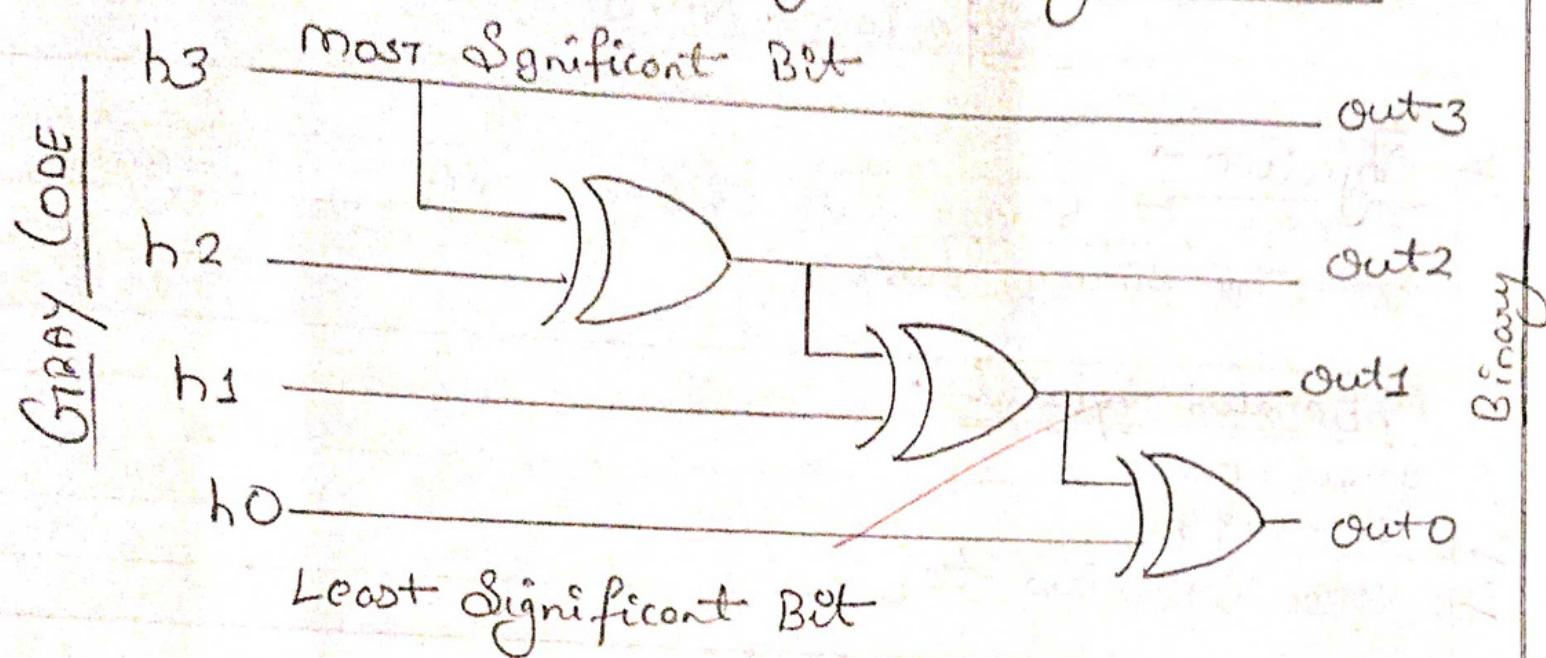
Step 3 Record successive sums until completed.

MSB LSB

Binary	\rightarrow	1	0	0	1
Code		↓			
Gray	\rightarrow	1	1	0	1

Teacher's Sign: .

* Circuit Diagram of Gray-to-Binary Conversion →



* Observation Table →

Q.No	Input (4 bit Gray Code)	Output (4 bit Binary Code)
0	0000	0000
1	0001	0001
2	0011	0010
3	0010	0011
4	0110	0100
5	0111	0101
6	0101	0110
7	0100	0111

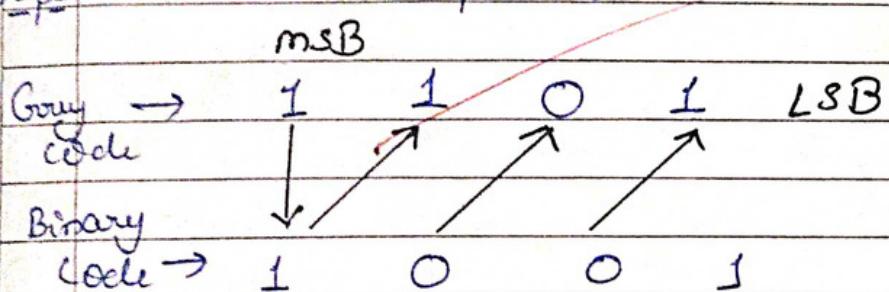
* Gray to Binary Conversion →

For Gray to Binary conversion follow the step below

Step 1 The msB of Gray and Binary are same. So write it directly.

Step 2 Add (Apply X-OR) binary msB to the next bit of Gray code. Record the result and ignore the carries.

Step 3 continue this process until the LSB is reached.



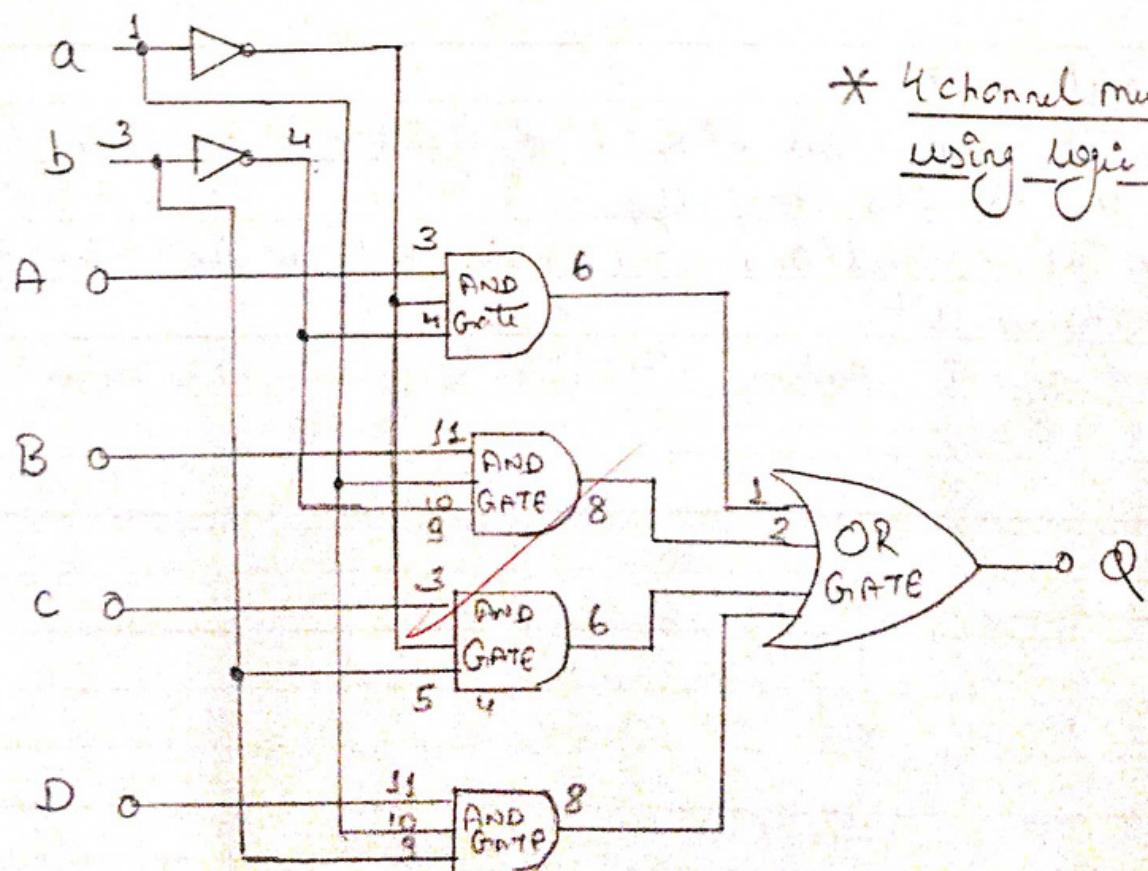
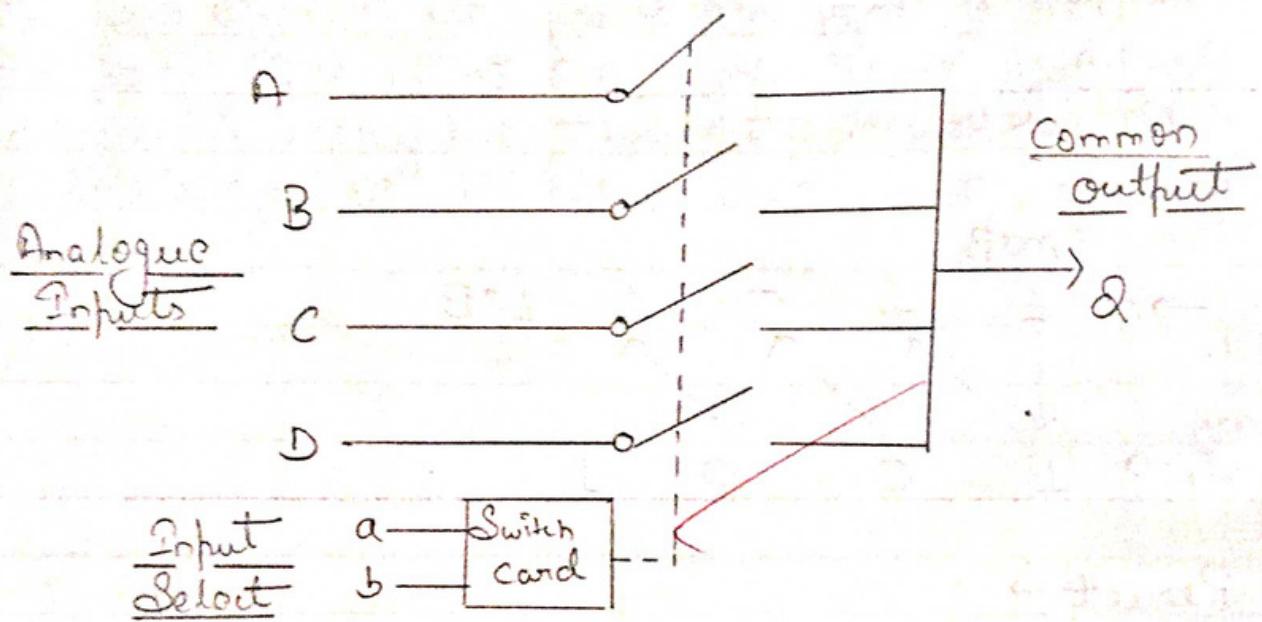
* Result →

All output are verified.

* Precautions →

- (1) All the connection should be light and proper.
- (2) Handle the Ics carefully.
- (3) Check the connections one again before switching on the trainer kit.
- (4) Switch off the trainer kit after performing the experiment.

Meeting
16/10/23



Experiment No. 4

* Aim →

Implementation and verification of 4×1 multiplexer using logic gates.

* Instruments Required →

Analog and Digital Trainer kit.

* Apparatus Required →

- 1) IC7432, IC7404, IC7411
- 2) connecting leads
- 3) Connecting wires.
- 4) Power Supply

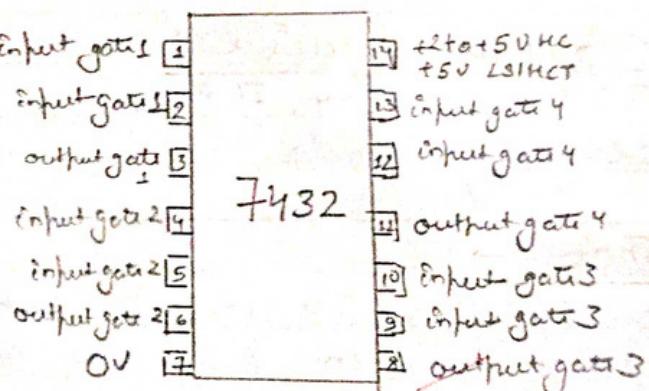
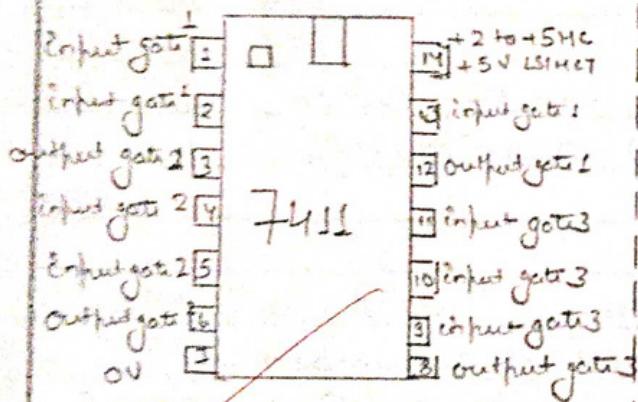
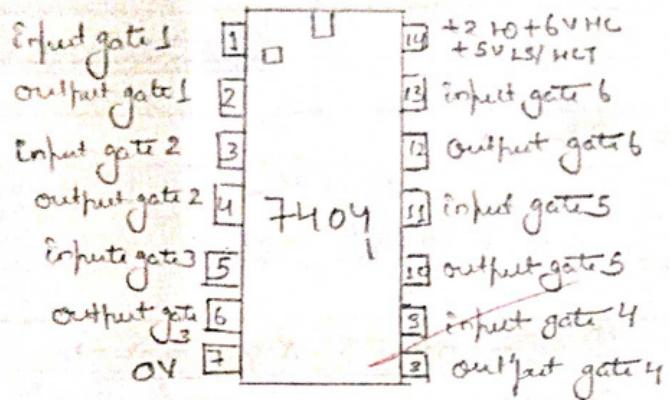
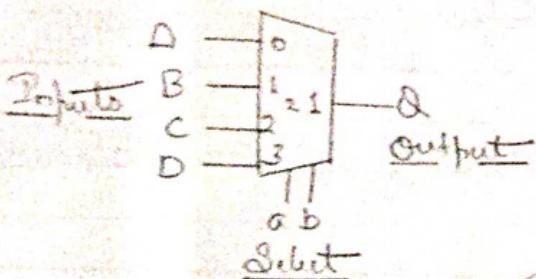
* Theory →

Multiplexers are circuits behaving like a controlled rotary switch, i.e., only one of a number of inputs may be selected as output. In digital electronics, a multiplexer is a combination of logic gates resulting into circuits with two or more inputs (data inputs) and one output.

The selection of the channel to be read into the outputs is controlled by supplying a specific digital word to a different set of inputs (select inputs).

The multiplexer - A data selector, more commonly called a multiplexer, shortened to "mux" or "mpx" are combinational logic switching devices that operate like a very fast acting multiple position rotary switch. They connect or control

* multiplexer symbol →



* Diagrams

Addressing		Output (Input Selected)
A	b	
0	0	A
0	1	B
1	0	C
1	1	D

* Observation - Table

Shiva

multiple input lines called "channels" consisting of either 2, 4, 8 or 16 individuals inputs, one at a time to an output 4x1 mux is shown below.

Multiplexors are not limited to just switching a number of different input lines or channels to one common logic output.

There are also types that can switch their inputs to multiple outputs and have arrangements of 4 to 2, 8 to 3 or even 36 to 4 etc configurations and an example of a simple dual channel 4 inputs multiplexer (4 to 2) is given below -

* Result →

Verify the truth-table of multiplexer.

* Conclusion →

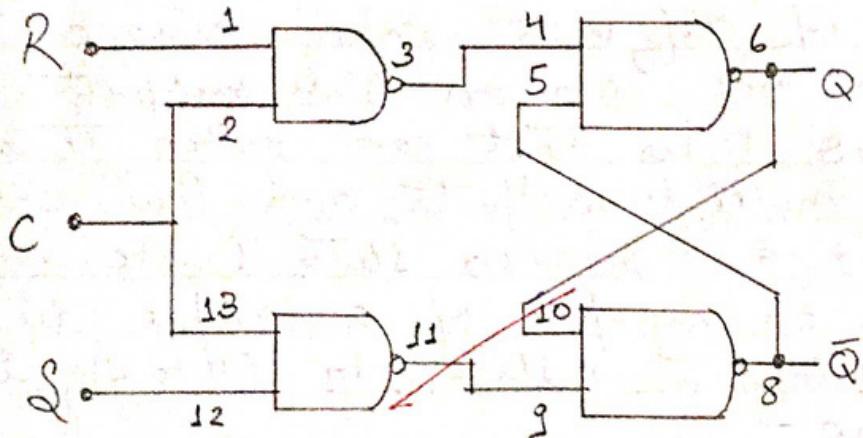
When we are compare the theoretical and practical values then we get both values are same.

* Precautions →

- 1: All connection should be right and tight.
- 2: Before connection, power supply is not ON.
- 3: When practical performed then Power Supply is making OFF.

P-S Flip-flop

* Circuit Diagram →



* Observation Table →

Clock	$Q(t)$	$Q(t+1)$	S	R
0	X	X	X	X
1	0	0	0	X
1	0	1	1	0
1	1	0	0	1
1	1	1	X	0

Experiment No. 5.

* Aim →

Verification of excitation tables of clocked flip-flops RS, JK, T and D flip-flops using NAND and NOR gates.

* Objective →

The importance of this experiment is to study how combinational circuits can be converted into sequential circuit and thus how we can create memory elements. This experiment will also enable understanding difference between flip-flops.

* Apparatus required →

1. Digital Trainer Kit
2. ICs (7400, 7404, 7410)
3. connecting wire

* Theory →

Flip flop is a bistable electronic circuit that has two stable states. That is, its output is either 5V (logic 1) or 0V (logic 0). Flip flop can be referred as memory device since its output will remain unchanged until its output is not changed. It is used to store one binary digit.

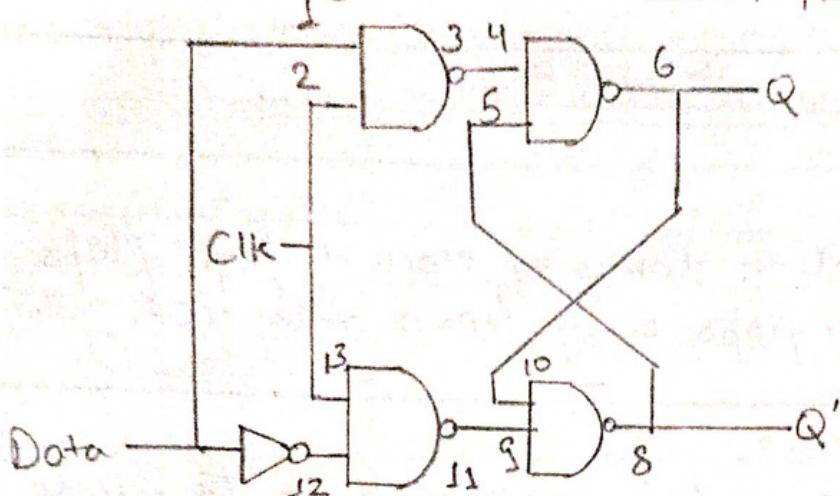
* R-S flip flop →

R-S flip-flop is one that has two inputs R & S and two outputs Q & Q'. An R-S flip flop can be constructed.

Teacher's Sign:

* Circuit Diagram →

(D-Flip flop)

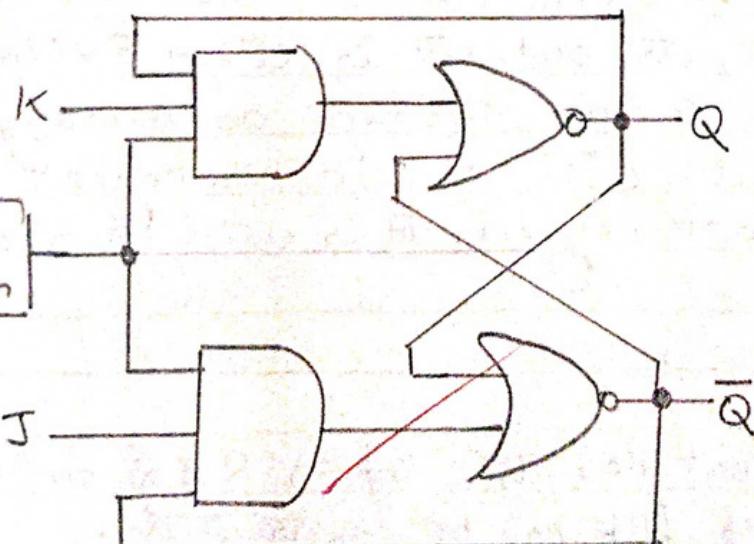


* Observation Table →

clock	$Q(t)$	$Q(t+1)$	D
0	X	X	X
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

* Circuit Diagram →

(J-K Flip flop)



* D-Flip-flop →

To avoid the forbidden case that occurs in R-S flip flop, when $R=S=1$, D flip flop is implemented. In the D flip flop, there is only one input D as shown in figure. we can transmit the value of D at the output of the flip flop when CLK is high.

* J-K Flip Flop →

Setting $R=S=1$ with a R-S Flip Flop Q and \bar{Q} will set to the ~~some logic level~~. This is an illegal condition. The J-K Flip Flop accounts for the illegal inputs. It is used to build counter. The values of J and K determine what a J-K Flip Flop does on the next clock edge. When both are low, the flip flop retains its last state. When J is low and K is high, the flip flop resets. When J is high and K is low the flip flop sets. When both are high the flip flop toggles. In this last mode, the J-K flip flop can be used as a frequency divider.

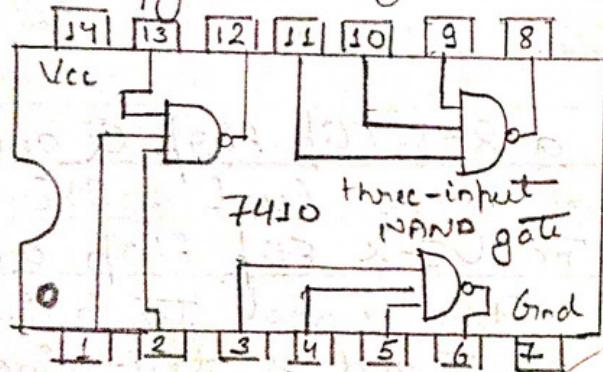
* I-Flip flop →

The T Flip Flop is known as Toggle flip flop. The T flip flop is a modification of the J-K flip flop by connecting both inputs J and K together.

→ Observation Table →

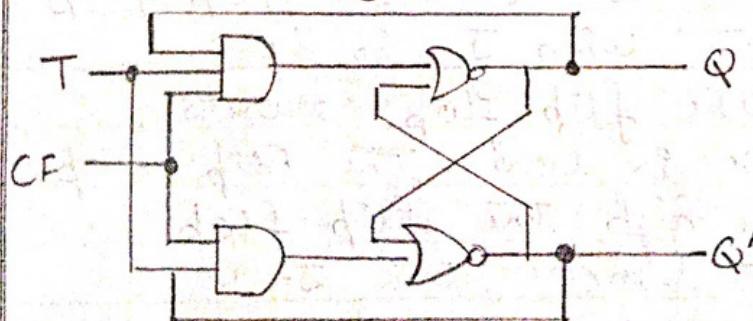
Clock	$Q(t)$	$Q(t+1)$	J	K
0	X	X	X	X
1	0	0	0	X
1	0	1	1	X
1	1	0	X	0
1	1	1	X	1

* Pin configuration of 7410 →



→ Circuit Diagram →

(T-flip-flop)



* Observation Table →

Clock	$Q(t)$	$Q(t+1)$	T
0	X	X	X
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Figure shows the logic diagram of T flip flop
logic symbol and truth table of T flip flop is also shown.

When $T=0$ both AND gates are disabled
and hence there is no change in the previous
output. When $T=1 (J=K=1)$ output logic. Toggles
means that the output is 0 when the previous
state is 1 otherwise output is 1 when
the previous state is 0. So the output is a
complement of the previous output.

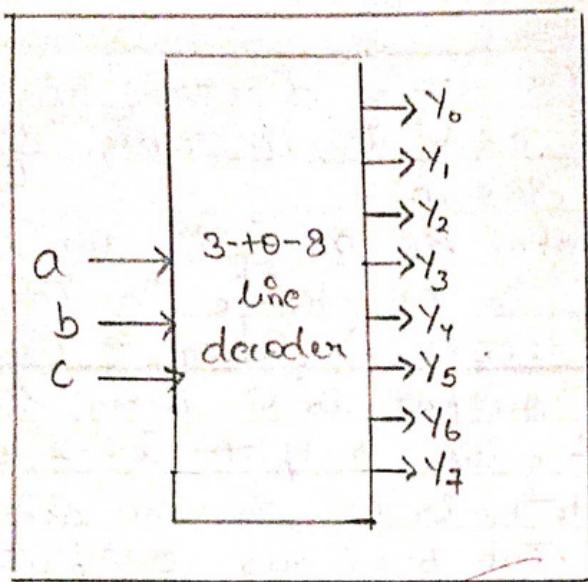
* Result →

The observation tables of various Flip Flop
verified.

* Precautions →

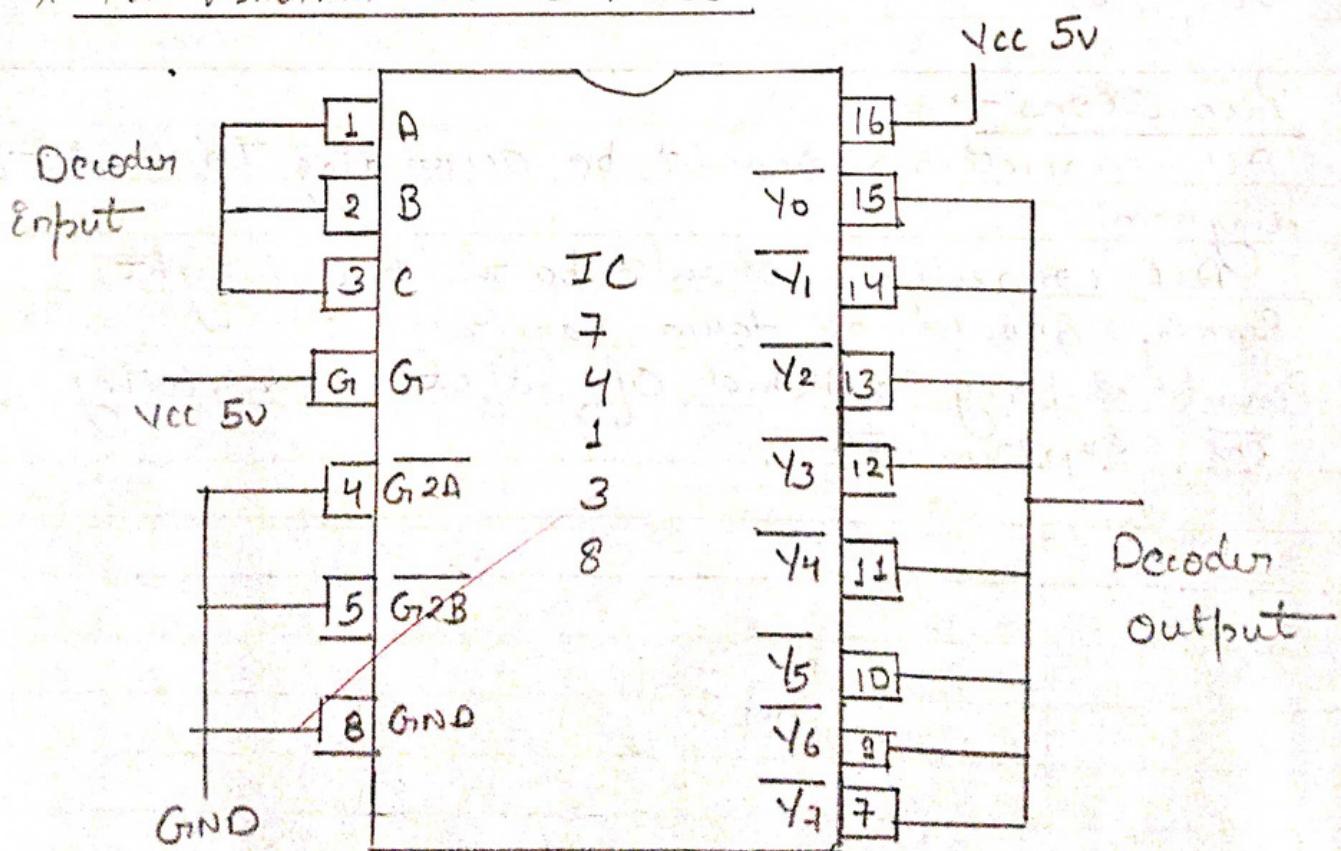
1. All connections should be according to circuit diagram.
2. All connections should be right and tight.
3. Reading should be taken carefully.
4. Power supply switched off after completing the experiment.

~~Neelam~~
06/11/23



* 3 to 8 line Decoder

* PIN Diagram of IC-74138



* Logic Diagram of 4:1 mux

Shiva

Experiment No. 6

* Aim →

To design and implement 4×1 and 8×1 multiplexer and 3 to 8 line Decoder using logic gates.

* Problem Statement →

The problem is to use the IC 74148 having the decoder circuit by analyzing the pin configuration, inputs and outputs and then verify the output behaviour of Decoders ICs.

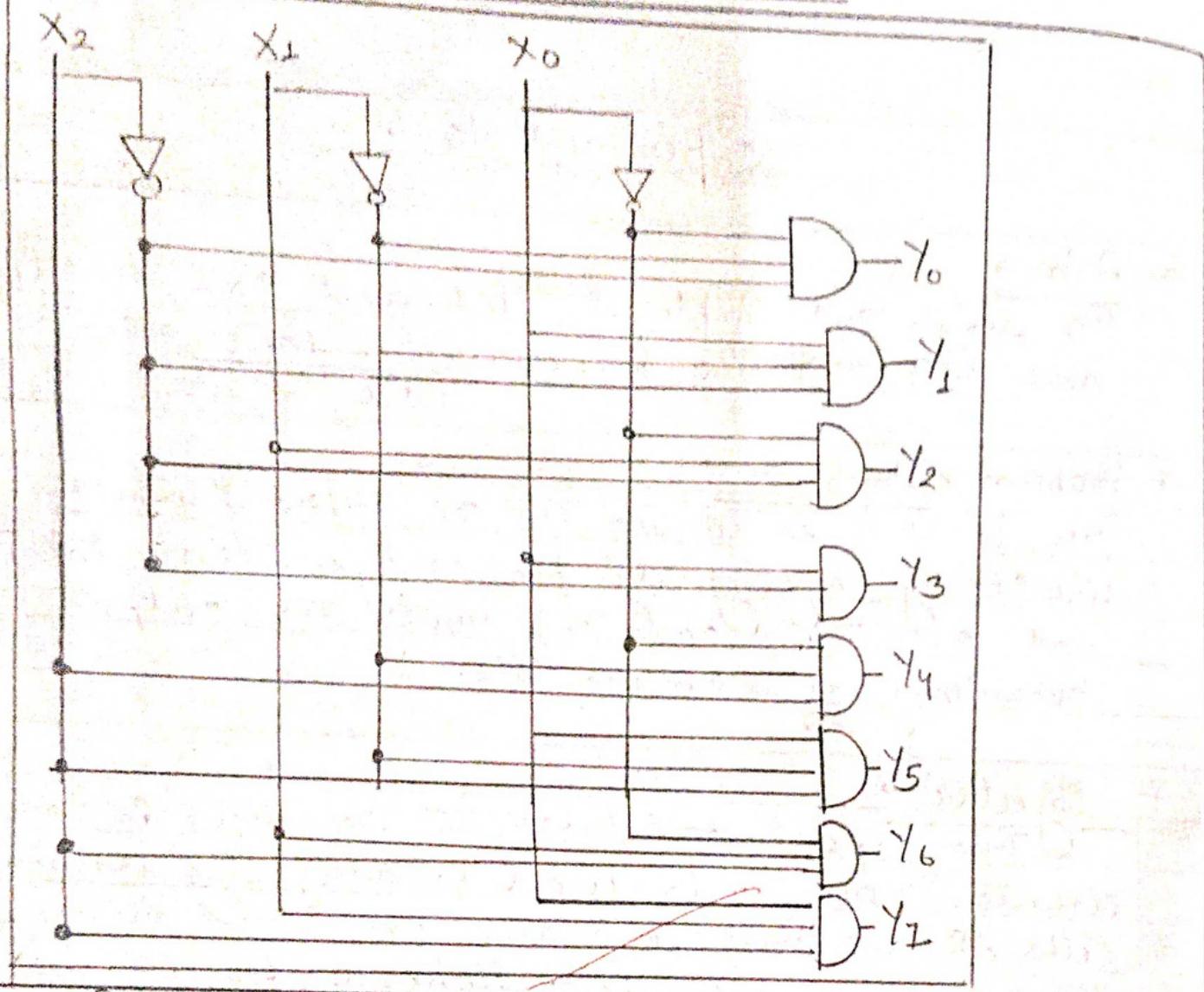
* Objective →

This experiment illustrates Decoders in hardware circuits. Decoder is used in addressing the memory cells when we have to address individual memory location made up of flip-flops. As decoder circuits are used to convert n input lines into 2^n output lines here input can be considered as an address lines and outputs can be used to identify the actual data location in memory.

* Apparatus Required →

1. Breadboard Trainer Kit
2. IC - 7411 (3 Input AND gate)
3. IC - 7404 (NOT Gate)
4. IC - 74138 (Decoder)
5. connecting wires.

* Logic Diagram Of 3-to-8 Decoder →



* For Decoder →

Input lines			Enable lines			out	Put lines							
C	B	A	G ₁	G _{2A}	G _{2B}	Y ₀	Y ₁	Y ₂	Y ₃	Y ₄	Y ₅	Y ₆	Y ₇	
0	0	0	1	0	0	0	1	1	1	1	1	1	1	1
0	0	1	1	0	0	1	0	1	1	1	1	1	1	1
0	1	0	1	0	0	1	1	0	1	1	1	1	1	1
0	1	1	1	0	0	1	1	1	1	0	1	1	1	1
1	0	0	0	0	0	1	1	1	1	1	0	1	1	1
1	0	1	1	1	0	1	1	1	1	1	1	0	1	1
1	1	0	1	0	1	1	1	1	1	1	1	1	0	1
1	1	1	1	0	0	1	1	1	1	1	1	1	1	0

* Theory →

* 3-8 Decoder →

A decoder is a combinational circuit that converts binary information from n line to a maximum of 2^n unique output lines. The name decoder is also used in conjunction with some code converters such as a BCD to Seven segment decoder. The block diagram of a decoder is shown in figure.

* Result →

Mux and Decoder Tables are verified.

* Precautions →

1. All connections should be according to circuit diagram.
2. All connection should be right and tight.
3. Reading should be taken carefully.
4. Power supply off after completing experiment.

~~Neelam
20/11/23~~

Experiment No. 7.

* Aim →

A 8-bit arithmetic logic unit.

* Problem Statement →

Arithmetic logic unit of computer perform basic arithmetic and logical operation of computer.

Here we, will study how to design an ALU circuit first evaluation the arithmetic circuit turn into one following an 8-bit ALU circuit.

* Objective →

The objective is to study the design of ALU and to build the circuitry for using the experiment will enable us to understand how the different units of ALU perform and how the operations are performed depending upon the selection bit used in multiplexer.

* Components Required →

1. Binary adder
2. multiplexer
3. AND gate
4. X-OR gate
5. NOT gate
6. connecting wires.

* Theory →

* Arithmetic circuit →

The basic component of an arithmetic circuit is the parallel adder by controlling data inputs to the adder it is possible to obtain different types of arithmetic operations.

The output of the binary adder is calculated from the following arithmetic sum-

$$D = A + Y + C$$

* Function table of arithmetic circuit →

S_1	S_0	Cin	Y	Output	microoperations.
0	0	0	B	$D = A+B$	Add
0	0	1	B	$D = A+B+1$	Add with carry
0	1	0	B'	$D = A+B'$	Subtract with Borrow
0	1	1	B'	$D = A+B'+1$	Subtract
1	0	0	0	$D = A$	Transfer A
1	0	1	0	$D = A+1$	Increment A
1	1	0	1	$D = A-1$	Decrement A
1	1	1	1	$D = A$	Transfer A

* Function table of logic unit →

$S_1 S_0$	Output	M -operation
0 0	$F = A \wedge B$	AND
0 1	$F = A \vee B$	OR
1 0	$F = A \oplus B$	XOR
1 1	$F = A'$	complement

* Result →

The ALU circuit has been designed and verified according to the function table of arithmetic logic unit.