

TUTORIAL / PRACTICAL NO.

EXPERIMENT - 1

AIM: Realization of basic gates implementation on bread board.

APPARATUS REQ.: IC-7404, 7804, 7432, LEDs-4, Breadboard connecting wires, and power supply.

THEORY:

NOT GATE:

Y/P	O/P
0	1
1	0

AND GATE:

Y/P	O/P
0	0
0	1
1	0
1	1

OR GATE:

Y/P	O/P
0	0
0	1
1	0
1	1

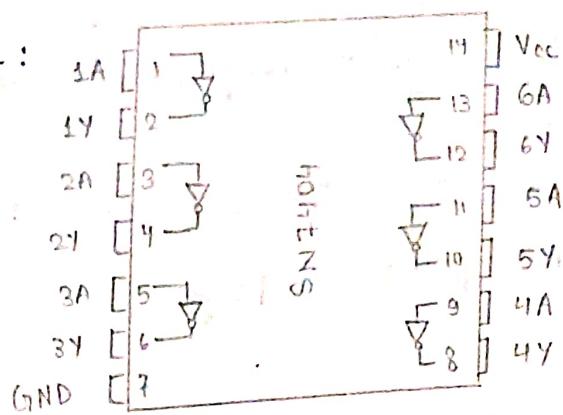
PROCEDURE:

For realization-

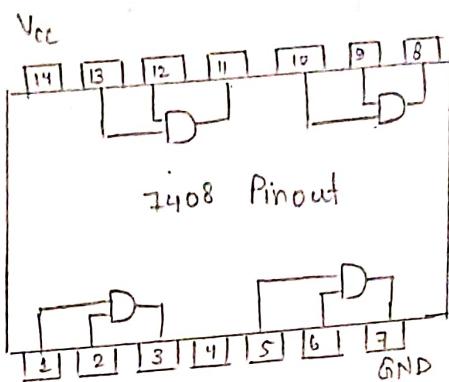
- ICs 7404, 7408, 7432 are mounted on bread board

Logic/pin Diagram

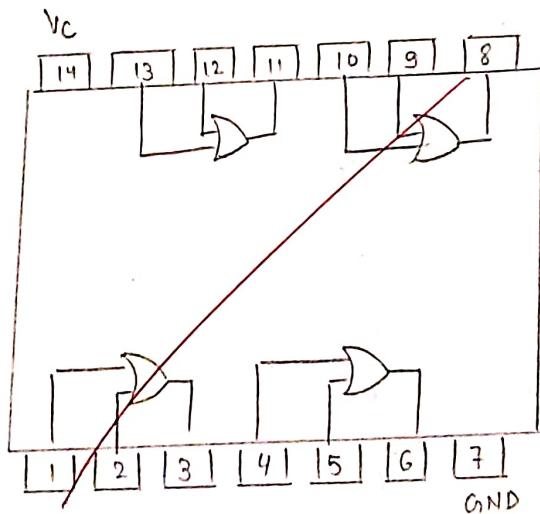
NOT Gate:



AND Gate:



OR Gate



TUTORIAL / PRACTICAL NO.

- Ground and Vcc pins of each IC is connected accordingly.
- Connections are completed as per requirements.
- LEDs are connected at each output terminal.
- Inputs are given as per truth table.

CONCLUSION :

- Output tested as per truth table through LEDs
- Basic gates NOT, AND and OR successfully realized on bread board.

TUTORIAL / PRACTICAL NO.

EXPERIMENT - 2

AIM: Design-

Half adder and full adder using AND, OR, NOT & XOR.

Half sub. and full sub. using AND, OR, NOT & XOR

4x1 Multiplexer realization on bread board.

APPARATUS REQ: IC 7404, 7408, 7432, LEDs-5, Breadboard connecting wires and power supply.

THEORY:

Half Adder: It adds two-bit binary digits A & B and yields two output Sum(s) and Carry(c)

$$S = A \oplus B \quad C = A \cdot B$$

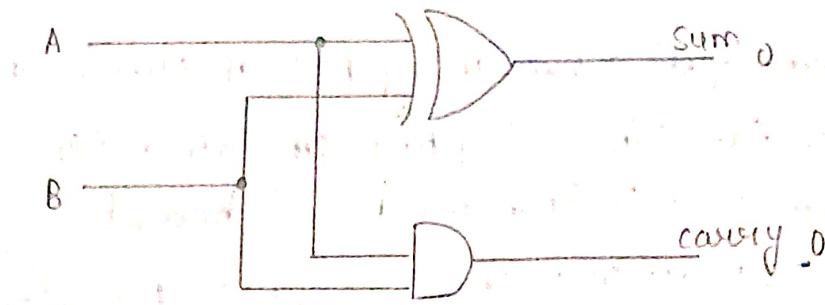
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Full Adder: It adds two-bit binary no. A & B, and also adds carry from previous less significant bit.

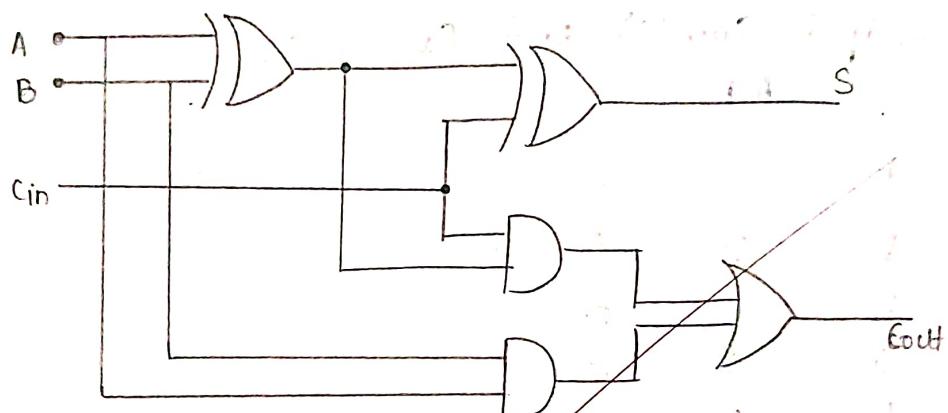
$$S = A \oplus B \oplus \text{Cin} \quad C = AB + BC + CA$$

A	B	Cin	S	C
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

Half Adder



Full Adder



TUTORIAL / PRACTICAL NO.

Half Subtractor: It subtracts two-bit and produces output difference (D) and Borrow (B_0)

$$D = A \oplus B \quad B_0 = \bar{A} \cdot B$$

A	B	D	B_0
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Full Subtractor: A full sub. subtracts two-bit nos. A & B by generating output and borrow consisting input & borrow.

$$D = A \oplus B \oplus B_{in} \quad B_{out} = \bar{A}B_{in} + \bar{A}B + B\bar{B}_{in}$$

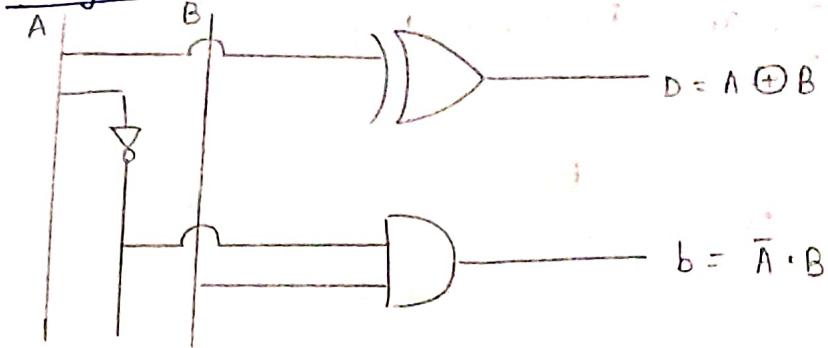
A	B	B_{in}	D	B
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

4x1 Multiplexer: It has 4 data inputs (I_3, I_2, I_1, I_0) and two selection lines (S_1, S_0) that shows which data input is selected.

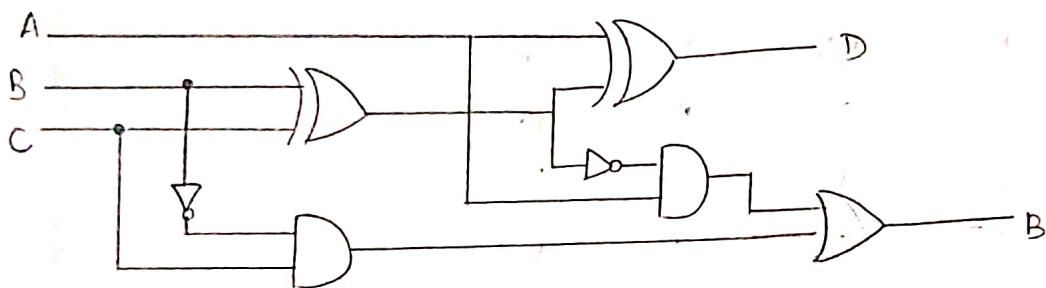
$$\gamma = \bar{S}_1 \bar{S}_0 + \bar{S}_1 S_0 + S_1 \bar{S}_0 + S_1 S_0$$

S_1	S_0	γ
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

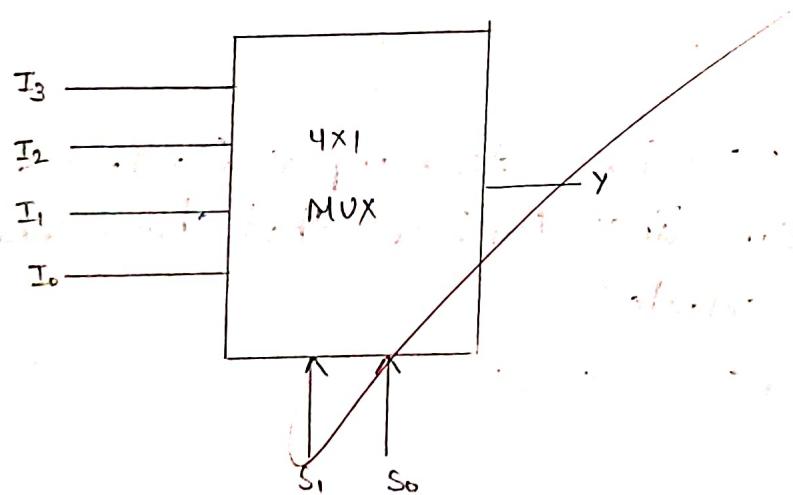
Half Subtractor



Full Subtractor



4x1 MUX



TUTORIAL / PRACTICAL NO.

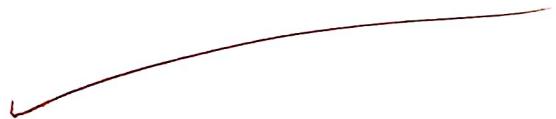
Procedure

For designing half/full adder, subtractor and MUX

- ICs 7404, 7408, 7486, 7432 are mounted on breadboard as per given logic diagram.
- Ground and Vcc pins of each IC is connected acc.
- Connections are completed as per req.
- LEDs are connected at each output terminal.
- Inputs are given as per truth table.

CONCLUSION

- Half adder and full subtractor are implemented on bread board.
- Half and full sub. are successfully implemented on bread board.
- 4x1 MUX are successfully realized on bread board.



TUTORIAL / PRACTICAL NO.

EXPERIMENT - 3

AIM: Realization of basic gates NOT, OR, AND, NOR, XOR, XNOR, Full adder/subtractor using NAND gate
APPARATUS REQ: IC 7400, breadboard, LED, connecting wires and power supply

THEORY:

IC 7400: is a small-scale package that contains dual input NAND gates. Any other logic gates can be made from NAND gate. It is made using transistors. It is universal gate.

NAND Gate:

A	B	O/P
0	0	0
0	1	1
1	0	1
1	1	1

NOT Gate: is made by joining of input of NAND together.

A	O/P
0	1
1	0

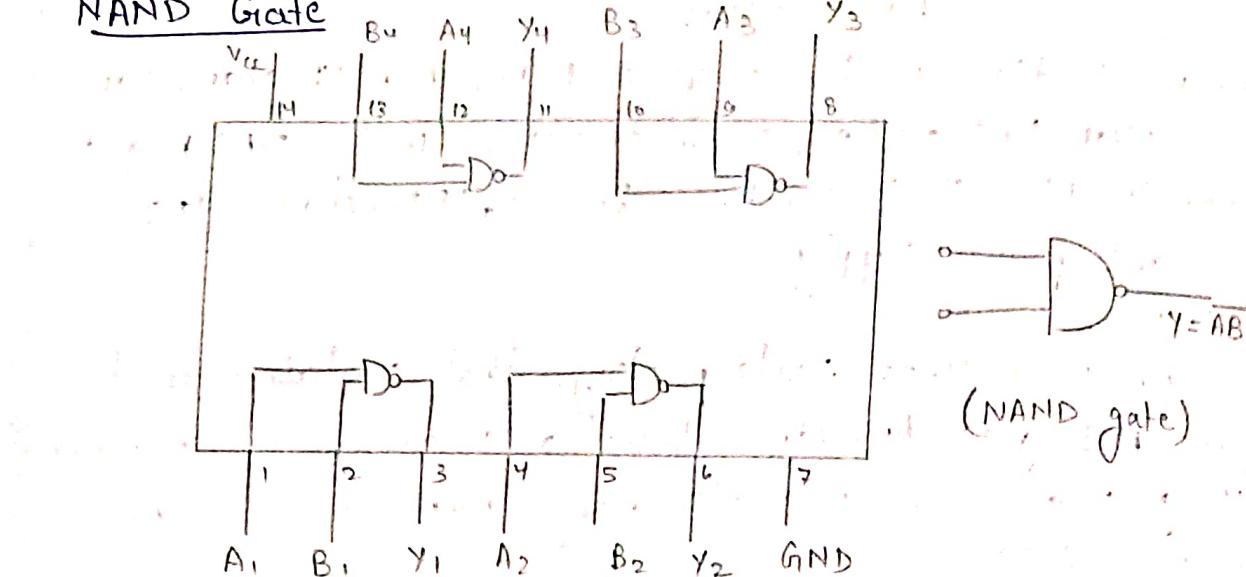
AND Gate: is realized by following NAND gate with NOT

A	B	O/P
1	1	1
1	0	0
0	1	0
0	0	0

OR Gate: is realized by shorting the inputs then joining NAND

A	B	O/P
1	1	1
1	0	1
0	1	1
0	0	0

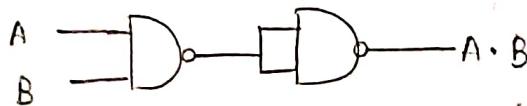
NAND Gate



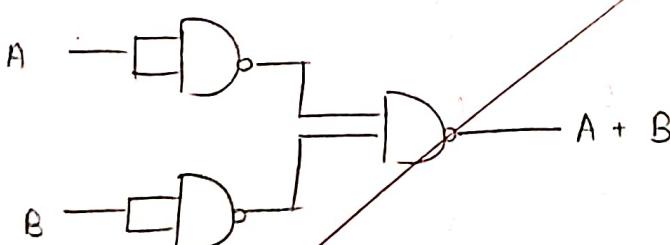
NOT Gate



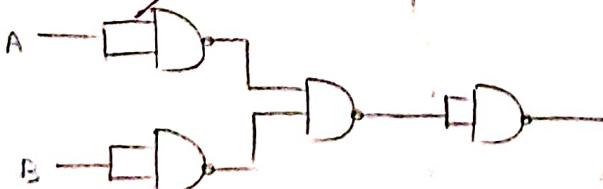
AND gate



OR gate



NOR gate



TUTORIAL / PRACTICAL NO.

NOR Gate: universal gate and compliment of OR gate.

A	B	%P
1	1	0
1	0	0
0	1	0
0	0	1

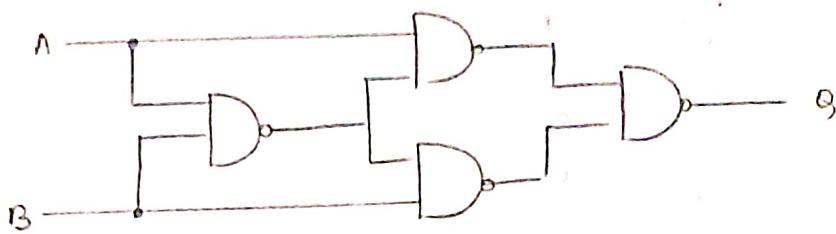
XOR Gate: realized using 4 NAND gate

A	B	%P
1	1	0
1	0	1
0	1	1
0	0	0

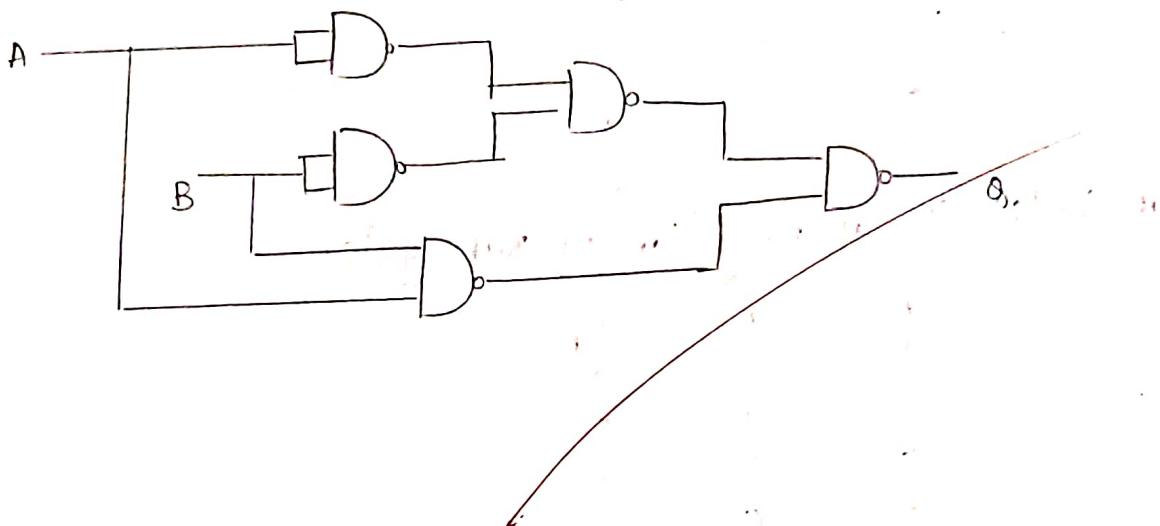
XNOR Gate: It is negation of XOR gate.

A	B	%P
1	1	1
1	0	0
0	1	0
0	0	1

XOR gate



XNOR gate



TUTORIAL / PRACTICAL NO.

EXPERIMENT - 4

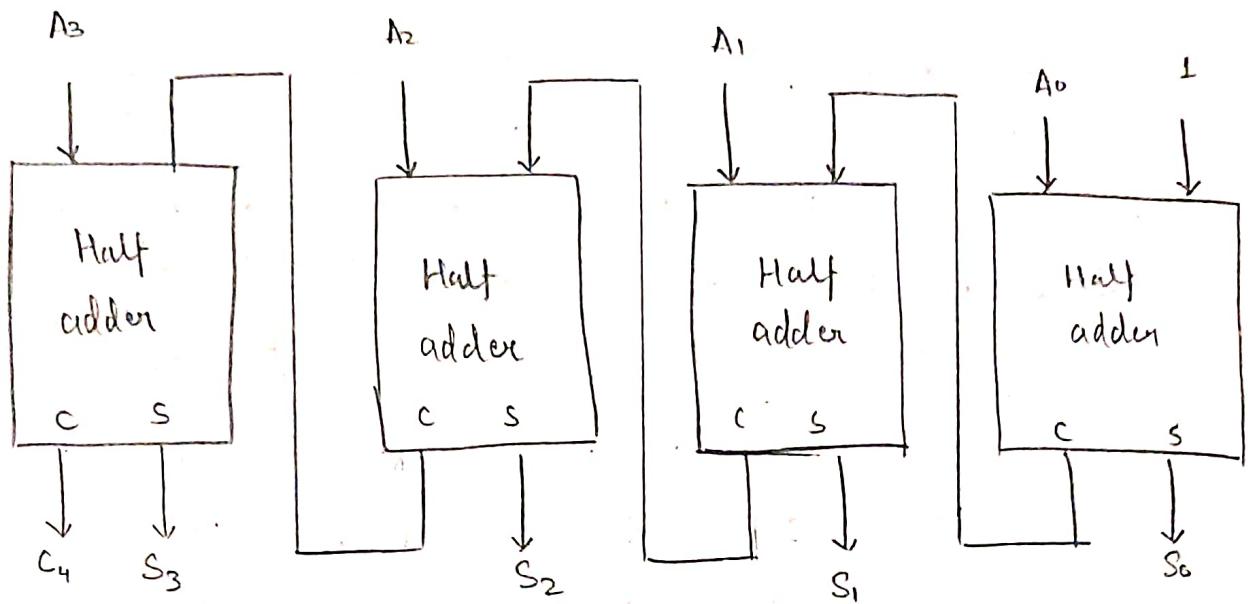
AIM: Design of 4-bit Incrementor using multiplexers
NAND gates

APPARATUS REQ.: IC 7400, IC 7404, IC 7408, IC 7432,
breadboard, LED, connecting wires and power supply.

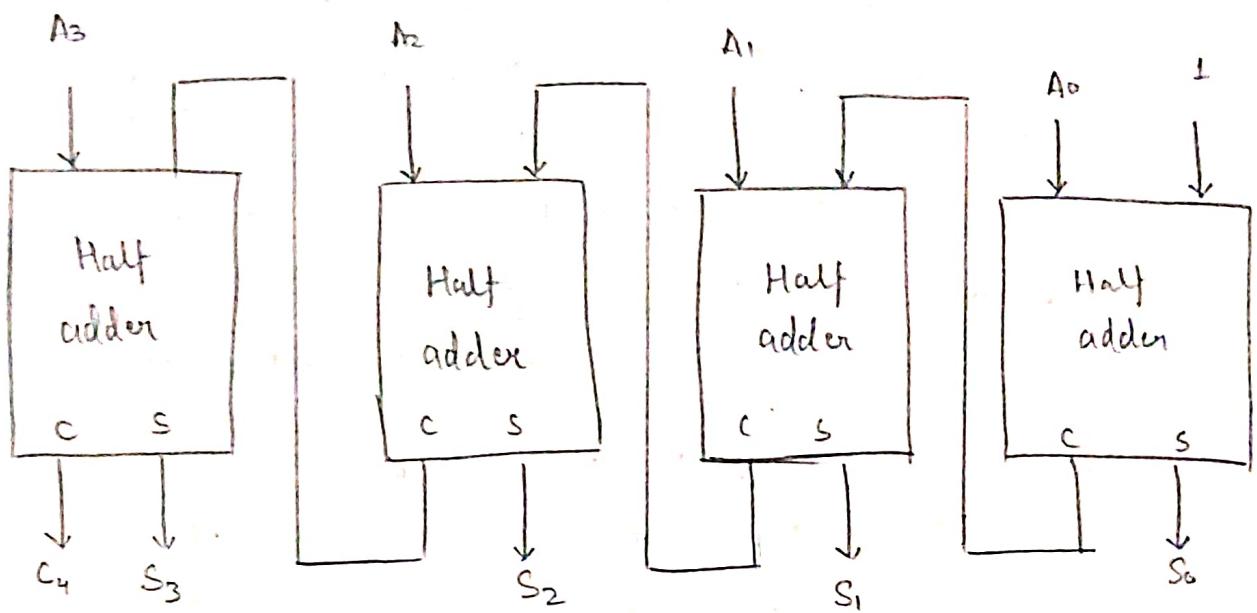
THEORY:

4-bit Incrementor : It adds the 1 to the current value and stored into a register. A 4-bit binary incrementor requires 4 cascaded half adder circuit.

A ₃	A ₂	A ₁	A ₀	S ₃	S ₂	S ₁	S ₀	C.
0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	01	0	0
0	0	1	0	0	0	1	1	0
0	0	1	1	1	0	0	0	0
0	1	0	0	0	0	1	0	0
0	1	0	1	0	1	0	1	0
0	1	1	0	0	1	1	0	0
0	1	1	1	1	1	0	1	0
1	0	0	0	0	1	0	0	0
1	0	0	1	0	1	0	1	0
1	0	1	0	1	1	0	0	0
1	0	1	1	1	1	0	1	0
1	1	0	0	0	1	1	0	1
1	1	0	1	1	1	0	1	0
1	1	1	0	1	1	1	0	1
1	1	1	1	0	1	1	0	1
1	1	1	1	1	0	0	0	1



(4-bit Binary Incrementor)



(4-bit Binary Incrementor)

C

TUTORIAL/ PRACTICAL NO.

PROCEDURE:

For designing incrementor using NAND gates/MUX following steps are used -

- ICs 7400 / IC 7404, IC 7408, IC 7432 are mounted on breadboard as per logic diagram. of Half adder given in exp 2 and 3.
- Ground and Vcc pins of each IC are connected accordingly.
- Connections are completed as per the req.
- LEDs are connected at each output terminal
- Inputs are given as per the truth table.

CONCLUSION :

- Incrementor circuit successfully implemented using MUX designed using basic gates on bread board
- Incrementor circuit successfully implemented using NAND gates on bread board.

TUTORIAL / PRACTICAL NO.

EXPERIMENT - 5

AIM: To realize 4 bit logical and arithmetic circuit separately and then integrating them into a single arithmetic and logical unit (ALU).

APPARATUS REQ: Breadboard, connecting wires, light emitting diode, battery eliminator and integrated circuit : IC7400, IC7486, IC7408, IC7432, IC7404, IC7483, IC74153

THEORY:

1. Logical Circuits : It takes n-bits of two nos. and perform bitwise logical AND for, NOT, XOR operations and produces the output.

TRUTH TABLE

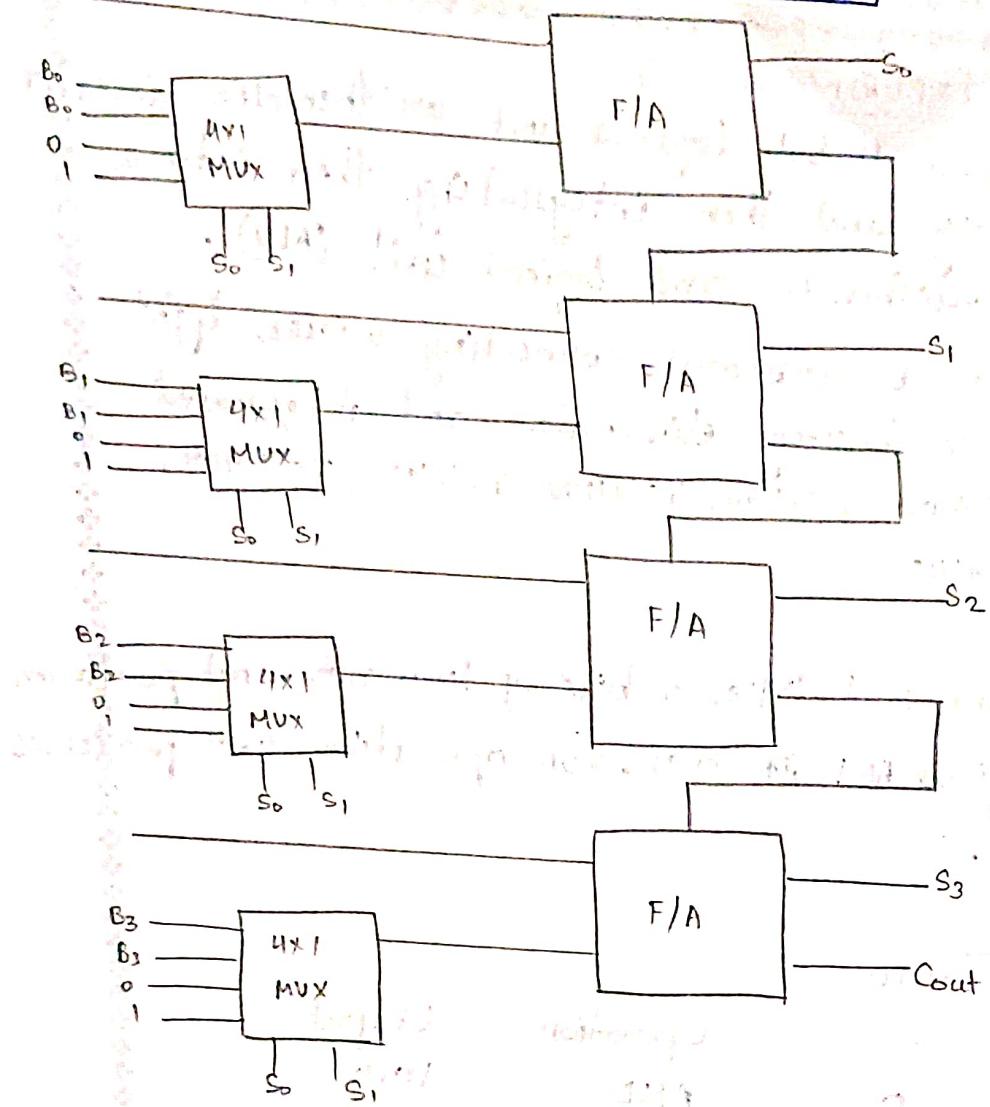
A-1111 B-1010

S ₁	S ₀	Operation	Output
0	0	AND	1010
0	1	OR	1111
1	0	XOR	0101
1	1	NOT	0000

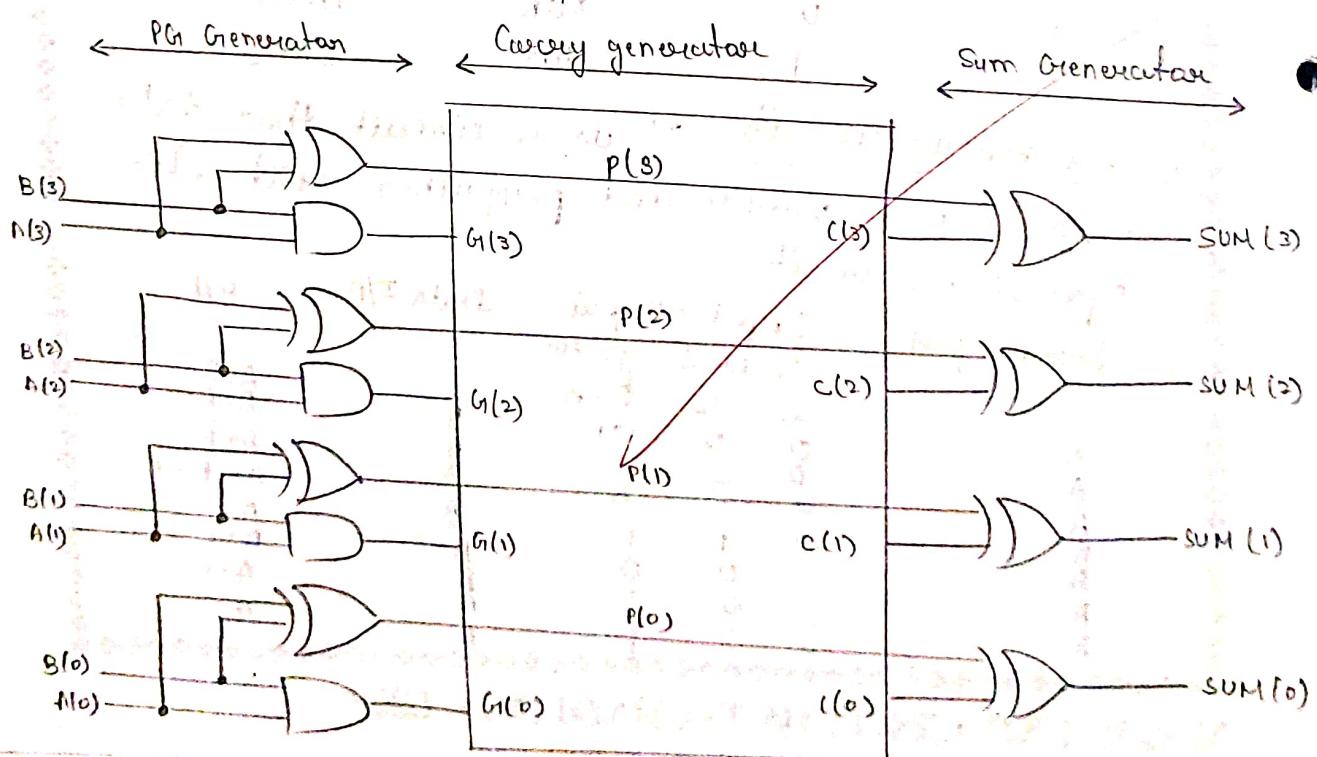
2. Arithmetic circuits : It is a circuit that takes two n-bits words and performs arithmetic operations on it.

Data input	Input / Output Central system			Data I/P	O/P
	S ₁	S ₀	S _{in}		
A	0	0	0	B	A+B
A	0	0	1	B	A+B+1
A	0	1	0	B	A+B
A	0	1	1	B	A+B+1
A	1	0	0	B	A
A	1	0	1	B	A+1
A	1	0	0	B	A-1
A	1	0	0	B	A

Arithmetic Unit



Logic Unit



TUTORIAL / PRACTICAL NO.

3. Arithmetic Logical Unit: combinational circuit that can perform various arithmetic and logical operations over two n-bit words.

NOTE: Integration of ALU can be implemented by either of following -

- i) Using MUX (ie output will be selected based on arithmetic/logic control variable)
- ii) Modification of Boolean exp. of full adders.

PROCEDURE :

1. ICs for resp. logical and arithmetic circuit were mounted on diff. set. of breadboard.
2. The ALU was verified.

PRELATIONS :

1. All connections were right and tight
2. ICs were biased properly
3. All the apparatus were handled with care.
4. All LEDs were tested beforehand.

TUTORIAL / PRACTICAL NO.

EXPERIMENT - G

AIM : To realise Carry Look Ahead Adder using AND/OR/XOR gates.

APPARATUS REQ. : Breadboard, connecting wires, light emitting diode, integrated circuits - IC 7486, IC 7408 IC 7432.

THEORY :

A CLA or fast adder improves speed by reducing amount of time required to determine carry bits. Two variations are defined on the basis of truth table carry generated and carry propagate.

$$P_i = A_i \oplus B_i$$

$$G_i = A_i B_i$$

The sum output and carry output can be expressed in terms of G_i and P_i as

$$G_{i+1} = G_i + P_i C_i$$

$$C_2 = G_1 + P_1 C_1 = G_1 + P_1 G_{10} + P_1 P_0 G_{1n}$$

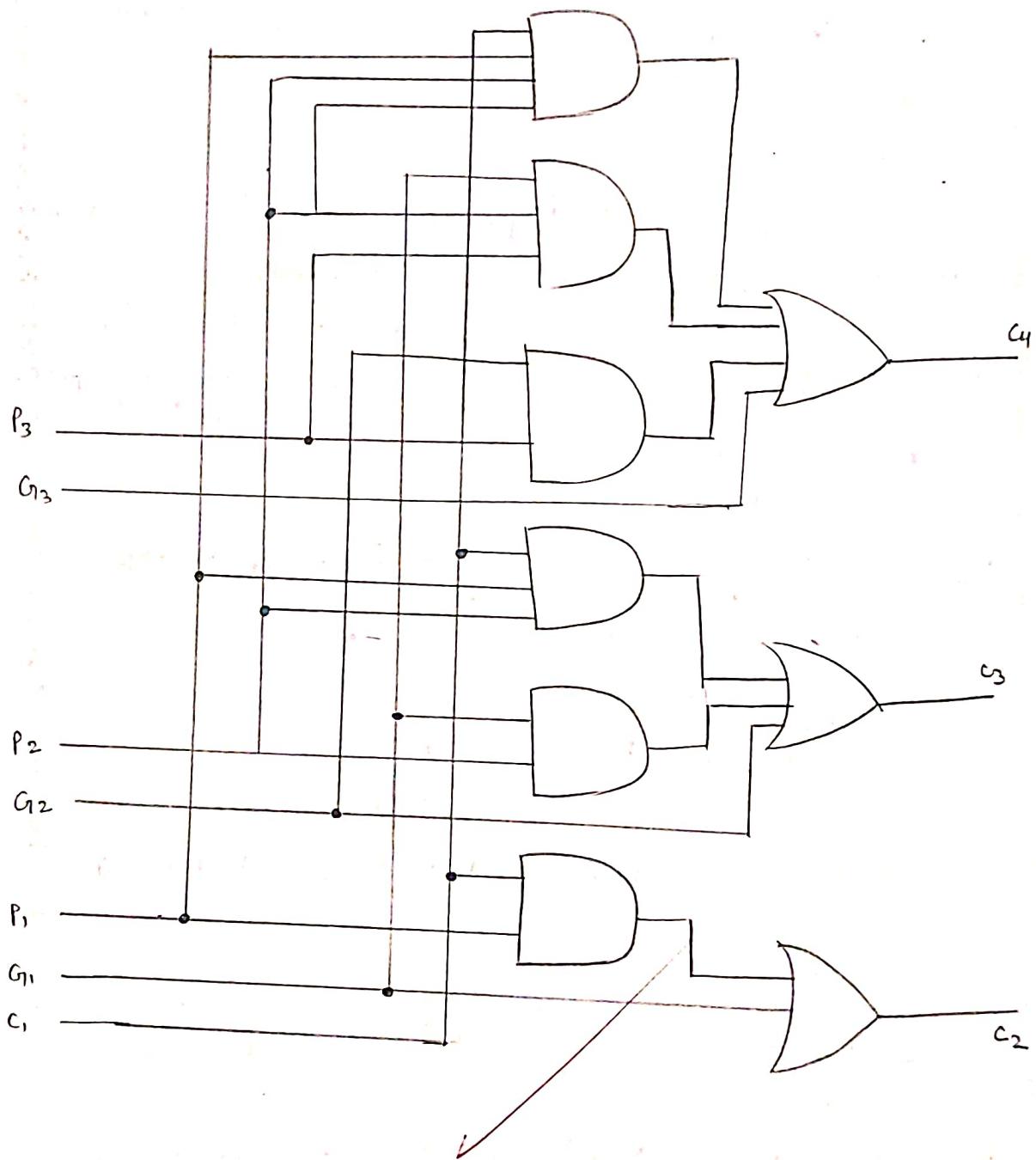
$$C_3 = G_2 + P_2 G_1 + P_2 P_1 G_{10} + P_2 P_1 P_0 G_{1n}$$

$$C_4 = G_3 + P_3 G_2 + P_3 P_2 G_{11} + P_3 P_2 P_1 G_{10} + P_3 P_2 P_1 P_0 G_{1n}$$

PROCEDURE :

1. ICs are mounted on breadboard as per logic diagram.
2. Setup of power supply and ground connections on bread board.
3. ICs were biased properly
4. All LEDs were tested beforehand.

Carry Look ahead Adder



TUTORIAL / PRACTICAL NO.

5) All apparatus were handled with care

CONCLUSION:

The CLA was realised by using 2 input AND/ OR/XOR gates and cross verified for different combinations of input.

APPLICATION:

The circuit is used in place of ripple carry adder for faster computations example, ALU, PU and timers

TUTORIAL / PRACTICAL NO.

EXPERIMENT - 7

AIM : To realize

- 4-bit carry save adder using full adder, AND/XOR/OR
- 4-bit shift register using JK Flip Flop

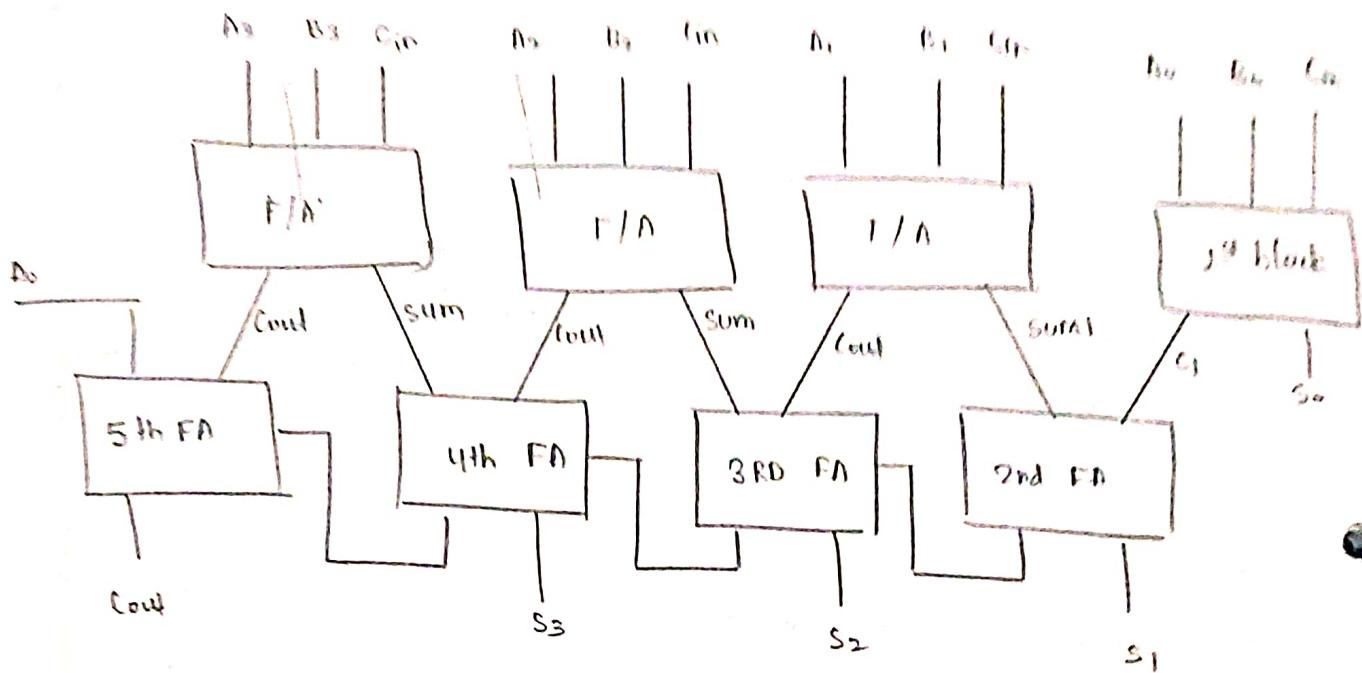
APPARATUS REQ: Breadboard, connecting wires light emitting diodes, battery eliminator, IC 7486, IC 7408, IC 7432, IC 7483 14 bit full adder, IC 7476

THEORY:

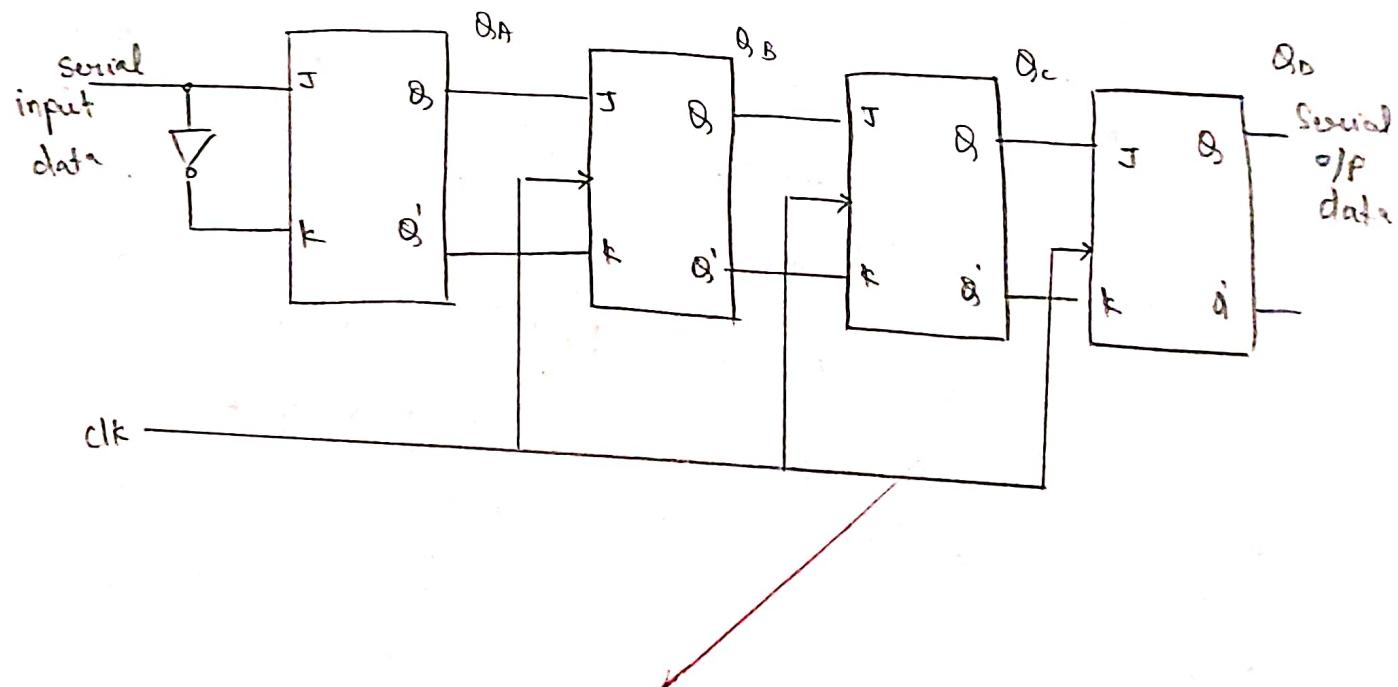
1) Carry Save Adder: digital adder used in computer micro architecture to efficiently compute sum of 3 or more n-bit binary nos. It outputs 2nos. of same dim as inputs, one which is a sequence of partial sum bit and the another which is a seq. of carry bit. These

2) Shift Register: A shift Register is using a cascade flip-flop where the output of flip flop is connected to input of next in chain resulting in circuit that shift by one position the bit array stored in it "shifting in" the data present at its input and "shifting out" the least bit in array at each transition of clock input.

4-bit carry save addition



4-bit shift register



TUTORIAL / PRACTICAL NO.

PROCEDURE

- 1) All ICs were mounted on breadBoard & biased properly.
- 2) LED's were mounted at each end.
- 3) Finally ckt was tested by feeding it with diff. comb. of input.

PRECAUTIONS:

1. All connections were tight
2. ICs were properly biased.
3. All LEDs were tested beforehand.

CONCLUSION

1. 4-bit carry save address using full adder/ AND /XOR/OR gates was realized.
2. 4-bit shift operator was using JK flip-flop was realized.

Prin
27/02/25