2. Digital Logic and Microprocessor

Er. Pralhad Chapagain

Syllabus

2.1 Digital logic: Number Systems, Logic Levels, Logic Gates, Boolean algebra, Sum-of-Products Method, Product-of-Sums Method, Truth Table to Karnaugh Map. (AExE0201)

2.2 Combinational and arithmetic circuits: Multiplexetures, Demultiplexetures, Decoder, Encoder, Binary Addition, Binary Subtraction, operation on Unsigned and Signed Binary Numbers. (AExE0202)

2.3 Sequential logic circuit: RS Flip-Flops, Gated Flip-Flops, Edge Triggered Flip-Flops, Mater- Slave Flip-Flops. Types of Registers, Applications of Shift Registers, Asynchronous Counters, Synchronous Counters. (AExE0203)

2.4 Microprocessor: Internal Architecture and Features of microprocessor, Assembly Language Programming. (AExE0204)

2.5 Microprocessor system: Memory Device Classification and Hierarchy, Interfacing I/O and Memory Parallel Interface. Introduction to Programmable Peripheral Interface (PPI), Serial Interface, Synchronous and Asynchronous Transmission, Serial Interface Standards. Introduction to Direct Memory Access (DMA) and DMA Controllers. (AExE0205)

2.6 Interrupt operations: Interrupt, Interrupt Service Routine, and Interrupt Processing. (AExE0206)

2.1 Digital logic: Number Systems, Logic Levels, Logic Gates, Boolean algebra, Sum-of-Products Method, Product-of-Sums Method, Truth Table to Karnaugh Map. (AExE0201)

Number System

- Binary
- Octal
- Decimal
- Hexadecimal

- Conversions

POSITIVE AND NEGATIVE LOGIC

POSITIVE LOGIC:

- When we use binary 1 for high voltage and binary 0 for low voltage then it is called positive logic.

NEGATIVE LOGIC:

- When we use binary 0 for high voltage and binary 1 for low voltage then it is called Negative logic.

Truth Table

A	Y=A’
1	0
0	1

Negative Logic NOT gate

Truth Table

A	B	Y=A+B
1	1	1
1	0	0
0	1	0
0	0	0

Negative Logic OR gate

Truth Table

A	B	Y=A.B
1	1	1
1	0	1
0	1	1
0	0	0

Negative Logic AND gate

- Positive logic AND gate is equivalent to Negative logic OR gate and vice versa.
- Positive logic NAND gate is equivalent to Negative logic NOR gate and vice versa.
- Positive logic XOR gate is equivalent to Negative logic XNOR gate and vice versa.

Logic Gates

- Basic Gates (NOT, AND, OR)
- Universal Gates (NAND, NOR)
- Exclusive Gates (XOR , XNOR)

Boolean Algebra

- Boolean Algebra is used to analyze and simplify the digital (logic) circuits. It uses only the binary numbers i.e. 0 and 1. It is also called as Binary Algebra or logical Algebra. Boolean algebra was invented by George Boole in 1854.
- Boolean Laws

Commutative law

Any binary operation which satisfies the following expression is referred to as commutative operation.

(i) $A.B = B.A$ (ii) $A + B = B + A$

Boolean Algebra

Associative law

This law states that the order in which the logic operations are performed is irrelevant as their effect is the same.

(i) $(A.B).C = A.(B.C)$ (ii) $(A + B) + C = A + (B + C)$

Distributive law

Distributive law states the following condition.

$$A.(B + C) = A.B + A.C$$

AND law

These laws use the AND operation. Therefore they are called as **AND** laws.

(i) $A.0 = 0$ (ii) $A.1 = A$

(iii) $A.A = A$ (iv) $A.\overline{A} = 0$

Boolean Algebra

OR law

These laws use the OR operation. Therefore they are called as **OR** laws.

$$(i)\ A + 0 = A \qquad (ii)\ A + 1 = 1$$

$$(iii)\ A + A = A \qquad (iv)\ A + \overline{A} = 1$$

INVERSION law

This law uses the NOT operation. The inversion law states that double inversion of a variable results in the original variable itself.

$$\overline{\overline{A}} = A$$

DE MORGANS THEOREM

Theorem 1

$$\overline{A.B} = \overline{A} + \overline{B}$$

NAND = Bubbled OR

Theorem 2

$$\overline{A + B} = \overline{A}.\overline{B}$$

NOR = Bubbled AND

FOR MCQ: https://www.sanfoundry.com/discrete-mathematics-questions-answers-de-morgan-laws/

BOOLEAN ALGEBRA

Dual Theorem:

- Starting with a Boolean relation, we can derive another Boolean relation, called its dual by the following steps:
  - Changing each OR sign into AND sign
  - Changing each AND sign into OR sign

Ex

S.N.	Given Expression	Dual of given expression
1	A + AB = A	A. (A + B) = A
2	A + A'B = A + B	A. (A' + B) = A.B
3	A + A' = 1	A.A' = 0
4	(A + B)(A + C) = A + BC	A.B + A.C = A. (B + C)

STANDARD FORM AND CANONICAL FORM

CANONICAL FORM:

- Max Term
- Min Term

Max Term:

- Each Max term is obtained from an OR logic of n variables with each variable being unprimed if the corresponding bit is zero (0) and primed if one (1).

A	B	C	Max term	designation
0	0	0	A+B+C	M_0
0	0	1	A+B+C’	M_1
0	1	0	A+B’+C	M_2
0	1	1	A+B’+C’	M_3
1	0	0	A’+B+C	M_4
1	0	1	A’+B+C’	M_5
1	1	0	A’+B’+C	M_6
1	1	1	A’+B’+C’	M_7

STANDARD FORM AND CANONICAL FORM

Min Term:

- Each Min term is obtained from an AND logic of n variables with each vari…nding bit is one (1) and primed if …

A	B	C	Min term	designation
0	0	0	A’B’C’	m_0
0	0	1	A’B’C	m_1
0	1	0	A’BC’	m_2
0	1	1	A’BC	m_3
1	0	0	AB’C’	m_4
1	0	1	AB’C	m_5
1	1	0	ABC’	m_6
1	1	1	ABC	m_7

STANDARD FORM AND CANONICAL FORM

STANDARD FORMS:

- In standard form the terms that form the function may contain one, two or any number of literals/ variables. There are two types of standard forms.
  - Sum of Product (SOP)
  - Product of Sum (POS)

Sum of Product (SOP)

- SOP is a Boolean expression containing terms with AND logic of 1 or more literals.
- E.g. F=XYZ + X'YZ + X'Y'Z

Product of Sum(POS)

- POS is a Boolean expression containing terms with OR logic of 1 or more literals.
- E.g. F=(X + Y + Z) (X' + Y + Z) (X' +Y' +Z)

Boolean Algebra - MCQ

1. Algebra of logic is termed as _____
a) Numerical logic
b) Boolean algebra
c) Arithmetic logic
d) Boolean number

2. Boolean algebra can be used _____
a) For designing of the digital computers
b) In building logic symbols
c) Circuit theory
d) Building algebraic functions

KARNAUGH MAP (K-MAP) (SOP)

- K-MAP is regarded as a diagrammatic or pictorial form of a truth table.
- The map is a diagram made up of squares.
- Each square represents one min/ max term.
- The MAP represents a visual diagram of all possible ways of function, may ne expressed in a standard form.

Basic K-MAP

A \ B		B'	B
		0	1
A'	0	A'B'	A'B
A	1	AB'	AB

Fig: Two variable K-MAP

KARNAUGH MAP (K-MAP) (SOP)

Three variable K-MAP

- There are 8 min terms for 3 binary variables.
- A MAP consists of 8 squares.
- The min terms are arranged not in a binary sequence but in sequence similar to gray code.
- The characteristics of the sequence is that only one bit is changes from one sequence to another.

BC / A		B'C'	B'C	BC	BC'
		00	01	11	10
A'	0	m_0	m_1	m_3	m_2
A	1	m_4	m_5	m_7	m_6

Fig: Three variable K-MAP

KARNAUGH MAP (K-MAP) (SOP)

Four variable K-MAP

- There are 16 min terms for 4 binary variables.
- A MAP consists of 16 squares.

AB \ CD		C'D'	C'D	CD	CD'
		00	01	11	10
A'B'	00	m_0	m_1	m_3	m_2
A'B	01	m_4	m_5	m_7	m_6
AB	11	m_{12}	m_{13}	m_{15}	m_{14}
AB'	10	m_8	m_9	m_{11}	m_{10}

Fig: Four variable K-MAP

Simplification:

- One square box represents one min term giving a term of four literals.
- Two adjacent square box represents a term of three literals
- Four adjacent square box represents a term of two literals.
- Eight square box represents one min term giving a term of one literals.
  - Sixteen adjacent square box represents a function 1
- Zero square box represents a function 0.

KARNAUGH MAP (K-MAP) (SOP)

DON'T CARE CONDITION:

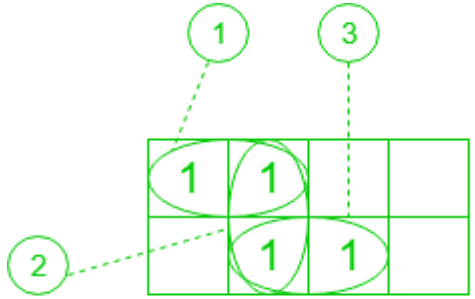
- There are some condition of inputs for which output is not specified and such output does not affect the whole system, which are known as Don't Care condition.
- The Don't care min terms are denoted by 'X' sign.

IMPLICANTS IN K-MAP

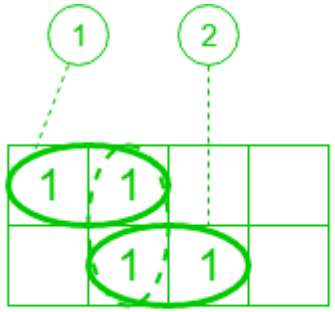
- **Prime Implicants**
  - A group of square or rectangle made up of bunch of adjacent minterms which is allowed by definition of K-Map are called prime implicants(PI) i.e. all possible groups formed in K-Map.
- **Essential Prime Implicants**
  - These are those sub cubes (groups) which cover at least one minterm that can't be covered by any other prime implicant. Essential prime implicants(EPI) are those prime implicants which always appear in final solution.

KARNAUGH MAP (K-MAP) (SOP)

E.G.



No. of Prime Implicants = 3



No. of Essential Prime Implicants = 2

KARNAUGH MAP (K-MAP) (SOP)

K-MAP EXAMPLE:

- Simplify using K-MAP and design a logic circuit.

(A

11,13) and the don't care condition D

CD / AB		C'D'	C'D	CD	CD'
		00	01	11	10
A'B'	00	X	1	1	X
A'B	01	0	X	1	0
AB	11	0	1	0	0
AB'	10	0	0	1	0

A'D

B'CD

BC'D

F = A'D + B'CD + BC'D

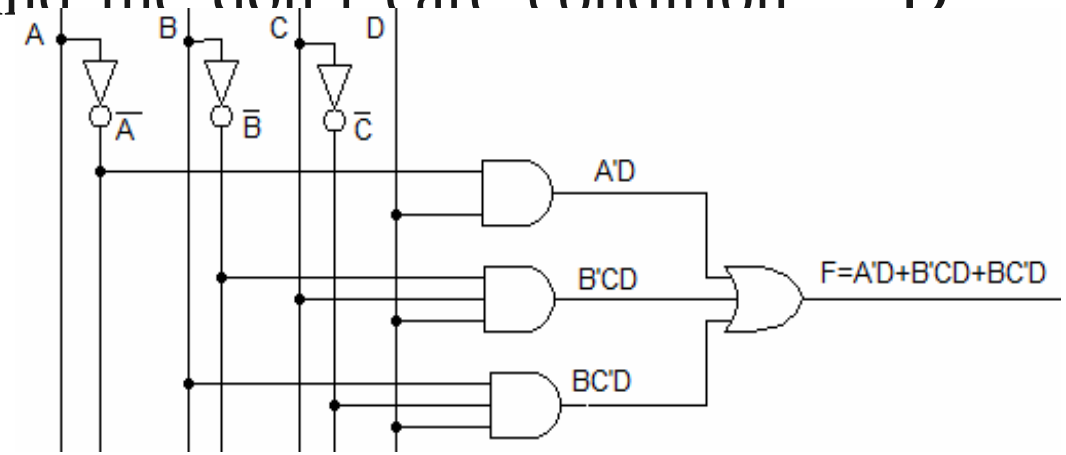


Fig: Combinational Design

Boolean Algebra - MCQ

4. F(X,Y,Z,M) = X`Y`Z`M`. The degree of the function is _____
a) 2
b) 5
c) 4
d) 1

5. A _____ value is represented by a Boolean expression.
a) Positive
b) Recursive
c) Negative
d) Boolean

7. What are the canonical forms of Boolean Expressions?
a) OR and XOR
b) NOR and XNOR
c) MAX and MIN
d) SOM and POM

8. Which of the following is/are the universal logic gates?
a) OR and NOR
b) AND
c) NAND and NOR
d) NOT

9. The logic gate that provides high output for same inputs _____
a) NOT
b) X-NOR
c) AND
d) XOR

10. The _____ of all the variables in direct or complemented from is a maxterm.
a) addition
b) product
c) moduler
d) subtraction

Boolean Algebra – MCQ –SET B

1. Which of the following bits is the negation of the bits “010110”?
a) 111001
b) 001001
c) 101001
d) 111111

2. Which of the following option is suitable, if A is “10110110”, B is”11100000” and C is”10100000”?
a) C=A or B
b) C=~A
c) C=~B
d) C=A and B

3. How many bits string of length 4 are possible such that they contain 2 ones and 2 zeroes?
a) 4
b) 2
c) 5
d) 6

4. If a bit string contains {0, 1} only, having length 5 has no more than 2 ones in it. Then how many such bit strings are possible?
a) 14
b) 12
c) 15
d) 16

5. If A is “001100” and B is “010101” then what is the value of A (Ex-or) B?
a) 000000
b) 111111
c) 001101
d) 011001

6. The Ex-nor of this string “01010101” with “11111111” is?
a) 10101010
b) 00110100
c) 01010101
d) 10101001

7. What is the one’s complement of this string “01010100”?
a) 10101010
b) 00110101
c) 10101011
d) 10101001

8. What is the 2’s complement of this string “01010100”?
a) 10101010
b) 00110100
c) 10101100
d) 10101001

Boolean Algebra – MCQ –SET B

9. If in a bits string of {0,1}, of length 4, such that no two ones are together. Then the total number of such possible strings are?

a) 1

b) 5

8 7

d) 4

10. Let A: "010101", B=?, If { A (Ex-or) B } is a resultant string of all ones then which of the following statement regarding B is correct?

a) B is negation of A

b) B is 101010

c) {A (and) B} is a resultant string having all zeroes

) All of the mentioned

Boolean Algebra – MCQ –SET C

1. The terms in SOP are called _____
a) max terms
b) min terms
c) mid terms
d) sum terms

2. A sum of products expression is a product term (min term) or several min terms ANDed together.
a) True
b) False

3. Which of the following is an incorrect SOP expression?
a) x+x.y
b) (x+y)(x+z)
c) x
d) x+y

4. The corresponding min term when x=0, y=0 and z=1.
a) x.y.z'
b) X'+Y'+Z
c) X+Y+Z'
d) x'.y'.z

5. LSI stands for _____
a) Large Scale Integration
b) Large System Integration
c) Large Symbolic Instruction
d) Large Symbolic Integration

6. Which operation is shown in the following expression: (X+Y').(X+Z).(Z'+Y')
a) NOR
b) ExOR
c) SOP
d) POS

Boolean Algebra – MCQ –SET C

7. The number of min terms for an expression comprising of 3 variables?

a) 8
b) 3
c) 0
d) 1

8. The number of cells in a K-map with n-variables.

a) 2n
b) n^2
a) 2^n
d) n

9. The output of AND gates in the SOP expression is connected using the _____ gate.

a) XOR
b) NOR
c) AND
a) OR

10. The expression A+BC is the reduced form of _____

a) AB+BC
b) (A+B)(A+C)
c) (A+C)B
d) (A+B)C

Boolean Algebra – MCQ –SET D

1. A Karnaugh map (K-map) is an abstract form of _____ diagram organized as a matrix of squares.
a) Venn Diagram
b) Cycle Diagram
c) Block diagram
d) Triangular Diagram

2. There are _____ cells in a 4-variable K-map.
a) 12
b) 16
c) 18
d) 8

3. Each product term of a group, w'.x.y' and w.y, represents the _____ in that group.
a) Input
b) POS
c) Sum-of-Minterms
d) Sum of Maxterms

4. It should be kept in mind that don't care terms should be used along with the terms that are present in _____
a) Minterms
b) Expressions
c) K-Map
d) Latches

5. The prime implicant which has at least one element that is not present in any other implicant is known as _____
a) Essential Prime Implicant
b) Implicant
c) Complement
d) Prime Complement

6. Product-of-Sums expressions can be implemented using _____
a) 2-level OR-AND logic circuits
b) 2-level NOR logic circuits
c) 2-level XOR logic circuits
d) Both 2-level OR-AND and NOR logic circuits

Boolean Algebra – MCQ –SET D

7. Each group of adjacent Minterms (group size in powers of twos) corresponds to a possible product term of the given _____

a) Function
b) Value
c) Set
d) Word

8. Don't care conditions can be used for simplifying Boolean expressions in _____

a) Registers
b) Terms
c) K-maps
d) Latches

2.2 Combinational and arithmetic circuits: Multiplexetures, Demultiplexetures, Decoder, Encoder, Binary Addition, Binary Subtraction, operation on Unsigned and Signed Binary Numbers. (AExE0202)

ADDER - HALF

- Binary adders nay be of two types:
  - Half Adder
  - Full Adder

Sum ar

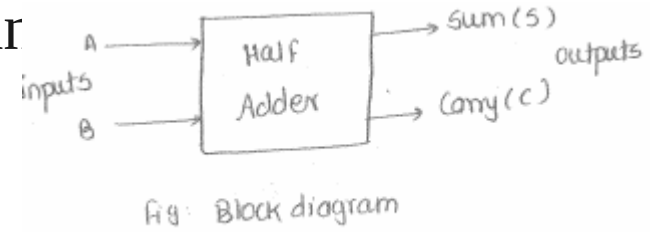


Fig: Block diagram

Half Adder:

- Half adder is a combinational logic circuit with two inputs and two outputs.
- It is the basic building block for addition of two single bit numbers
- This circuit has two outputs namely,
- Tr

Inputs		Outputs	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

ADDER - HALF

K-map for sum

A \ B	0	1
0	0	[1]
1	[1]	0

K-map for carry

A \ B	0	1
0	0	0
1	0	[1]

$Sum = \bar{A}B + A\bar{B} = A \oplus B$

$Carry = AB$

Combinational circuit is:

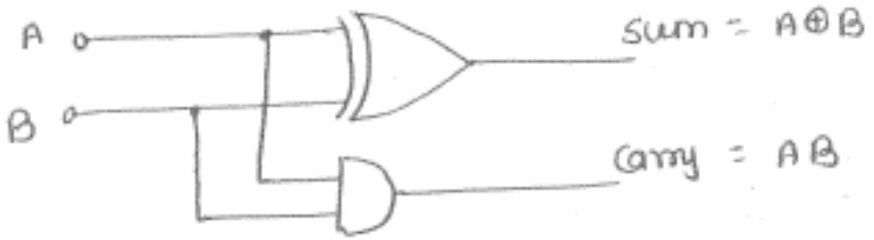


Fig: An Half adder circuit

Half adder using basic gates:

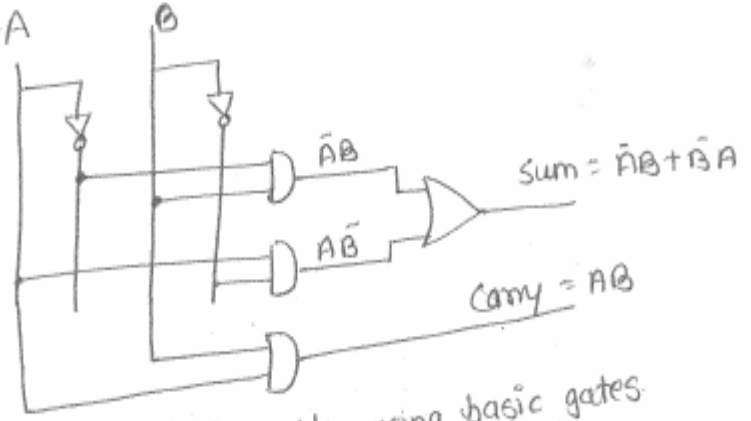


Fig: An Half adder using basic gates

- Limitations:
  - The addition of three bits is not possible to perform by using an half adder circuit.

ADDER - FULL

Full Adder:

- To overcome the drawback of an half adder circuit, a 3-single bit adder circuit called full adder is developed.
- Basically, a full adder is a three input and two output combinational circuit.

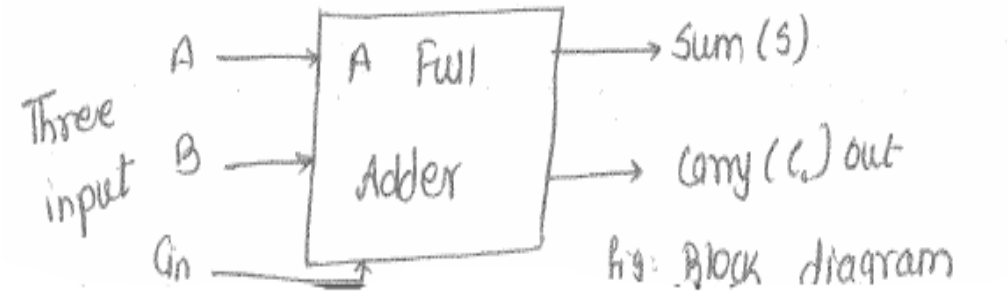


Fig: Block diagram

- T

Inputs			Outputs	
A	B	C_{in}	Sum	Carry (C_o)
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

ding block of the 4 bit/ 8 bit binary/ BCD adder Ics such as 7483.

ADDER - FULL

K-map for sum (S)

A \ BCin	00	01	11	10
0	0	[1]	0	[1]
1	[1]	0	[1]	0

$S = A'B'C_{in} + A'BC_{in}' + AB'C_{in}' + ABC_{in}$

$= A \oplus B \oplus C_{in}$

K-map for carry out (C_o)

A \ BCin	00	01	11	10
0	0	0	[1]	0
1	0	[1]	[1]	[1]

$C_o = AC_{in} + BC_{in} + AB$

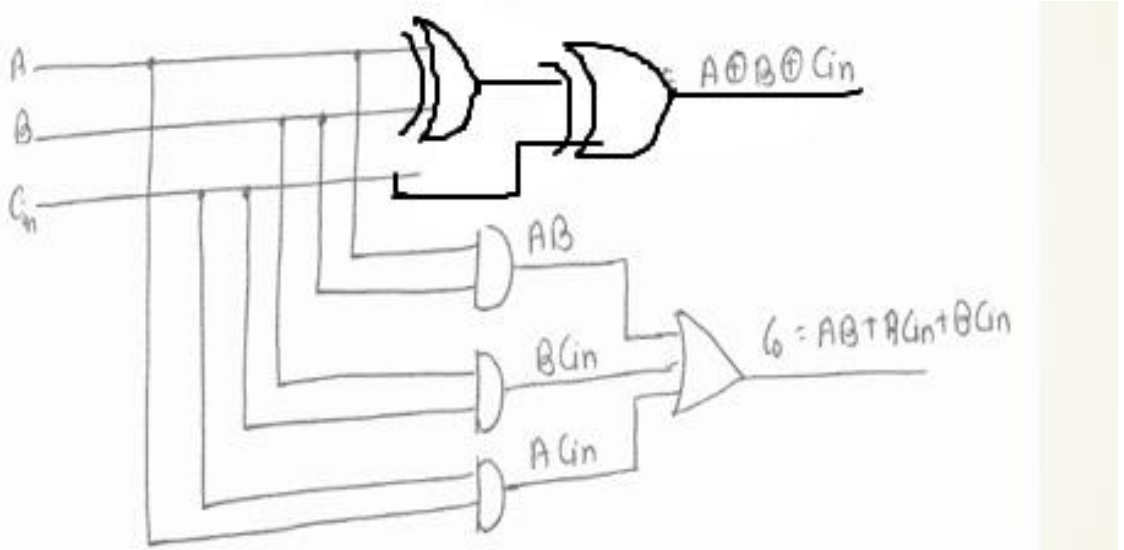


Fig: A full adder circuit

SUBTRACTOR - HALF

- Binary Subtractor nay be of two types:
  - Half Subtractor
  - Full Subtractor

Half Subtractor:

- Half subtractor may be defined as a combinational circuit with two inputs and two outputs (i.e. difference and borrow)

- In subtraction (A-B), A is called m… S…

Inputs		Outputs	
A	B	Difference (A-B)	Borrow (B_o)
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

K-map for Difference (D)

- Truth

A \ B	0	1
0	0	1
1	1	0

$D = \bar{A}B + A\bar{B} = A \oplus B$

SUBTRACTOR - HALF

K-map for borrow o/p

A \ B	0	1
0	0	1
1	0	0

Borrow = $\bar{A}B$

Logic diagram

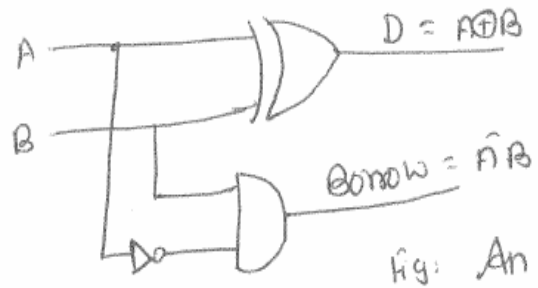


Fig: An half subtractor circuit.

Using Basic gate:

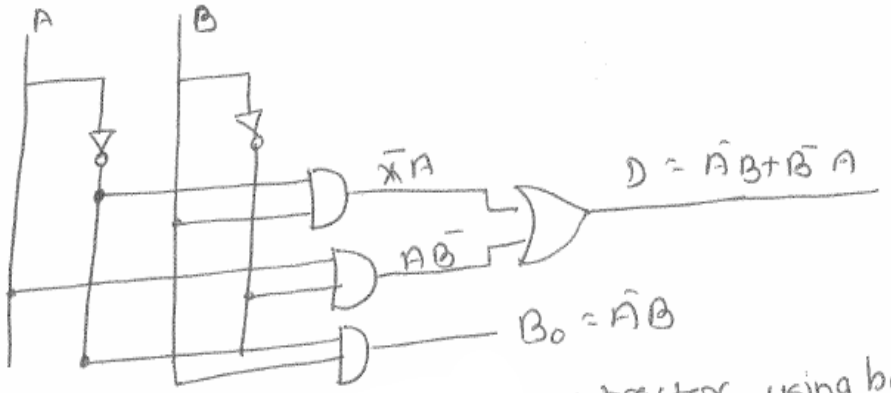


Fig: Half subtractor using basic gates

SUBTRACTOR - FULL

Full Subtractor:

- A Full Subtractor is a combinational circuit with three inputs A, B, B_{in} and two outputs D (Difference) and Borrow ($B_{o.}$)
- Here A is the minuend, B is the subtrahend, B_{in} is the borrow produced by the previous stage, D is the difference output and B_o is the borrow output.
- T

Inputs			Outputs	
A	B	B_{in}	Difference (A-B-B_{in})	Borrow (B_o)
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

SUBTRACTOR - FULL

K-map

for difference output

A \ BBin	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$$D = \bar{A}\bar{B}B_{in} + \bar{A}B\bar{B}_{in} + A\bar{B}\bar{B}_{in} + ABB_{in}$$

$$= B_{in}(\bar{A}\bar{B} + AB) + \bar{B}_{in}(\bar{A}B + A\bar{B})$$

$$= B_{in}(\overline{A \oplus B}) + \bar{B}_{in}(A \oplus B)$$

$$= A \oplus B \oplus B_{in}$$

for Borrow output

A \ BBin	00	01	11	10
0	0	1	1	1
1	0	0	1	0

$$B_0 = \bar{A}B_{in} + \bar{A}B + BB_{in}$$

SUBTRACTOR - FULL

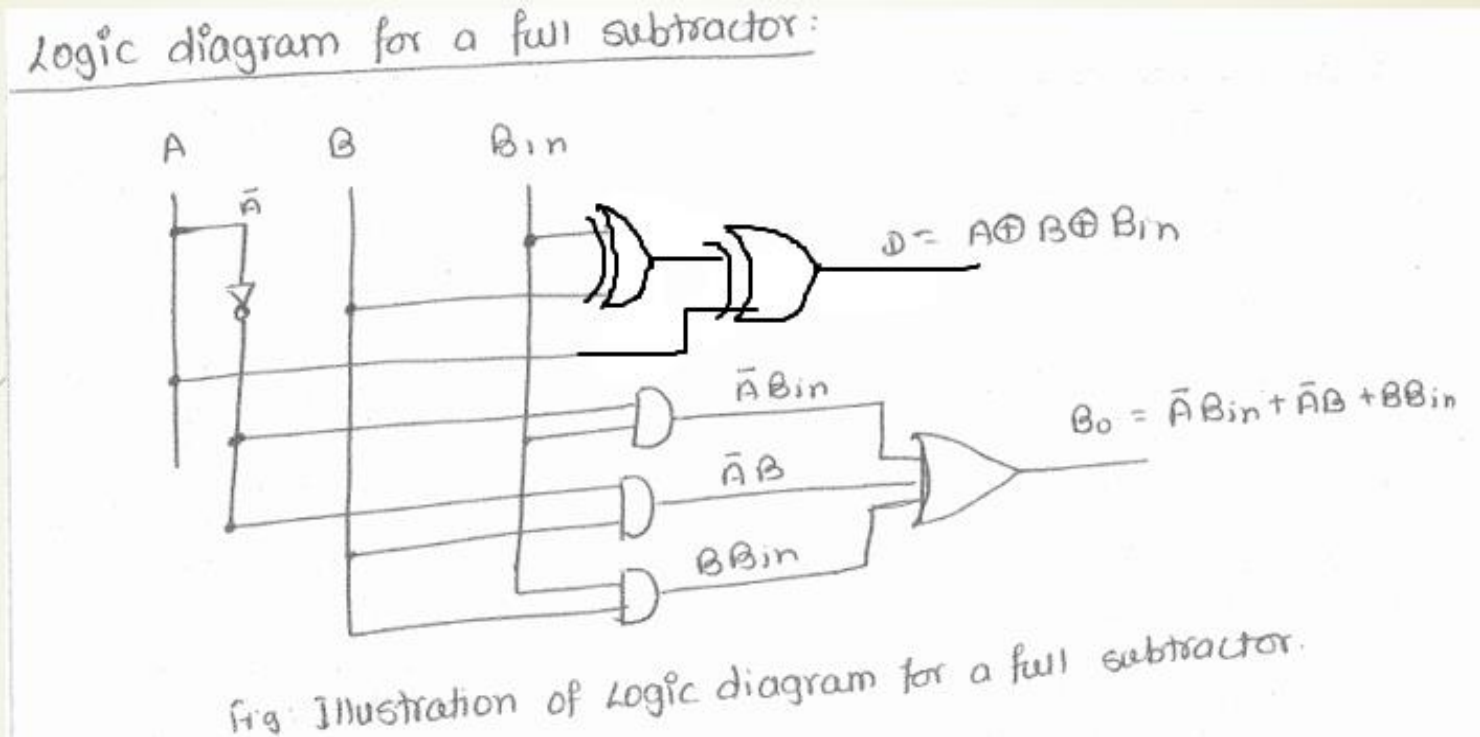
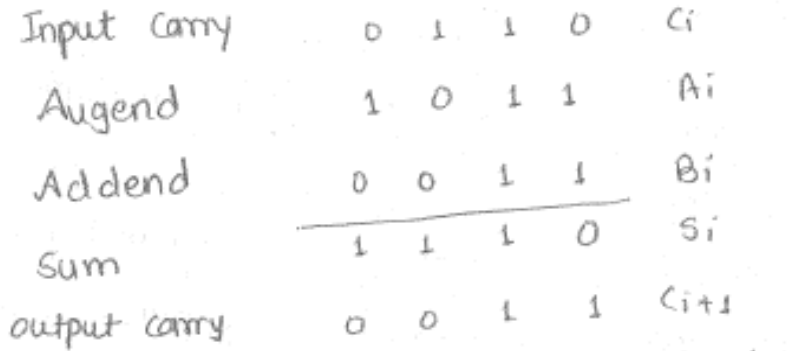


Fig: Illustration of Logic diagram for a full subtractor.

BINARY PARALLEL ADDER

- A full adder is capable of adding only two single digit binary numbers along with a carry input.
- But, in practice, we need to add binary numbers which are much longer than just one bit.
- To add two n- bit binary numbers, we need to use the n-bit parallel adder.
- It makes use of a number of full adders in cascade.
- Th is full adder is connected to the carry input of



A 4-bit Binary Parallel Adder:

BINARY PARALLEL ADDER

- A binary parallel adder is a digital function that produces the arithmetic sum of two binary numbers in parallel.
- It consists of full-adder connected in cascade. with the output carry from c rry of the next full-adder.

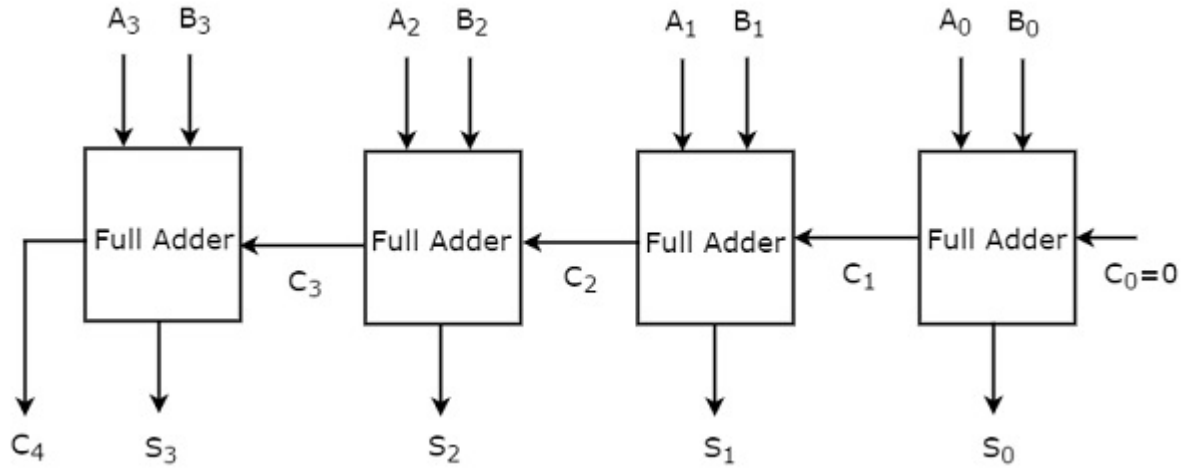
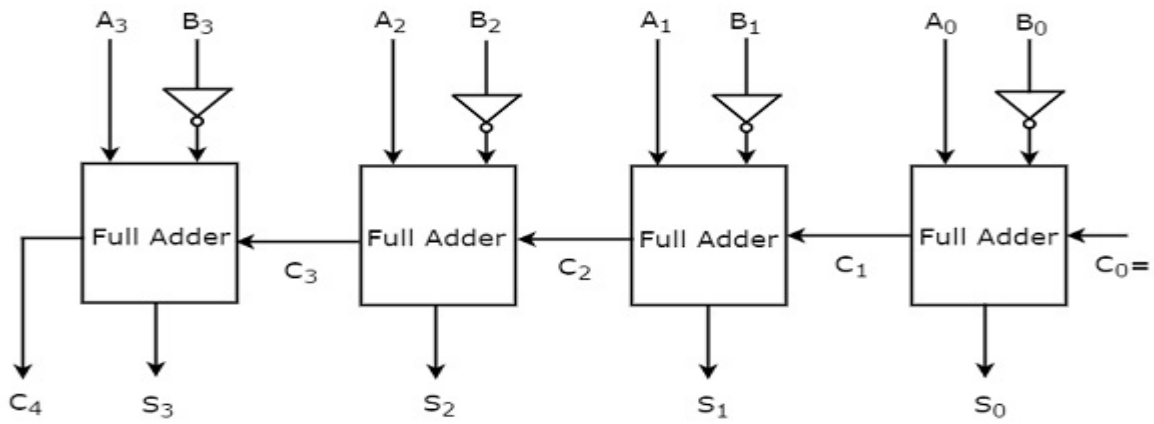


Fig: 4-bit Binary Parallel adder

4-BIT BINARY PARALLEL SUBTRACTOR

- The 4-bit binary subtractor produces the subtraction of two 4-bit numbers.
- Let the 4 bit binary numbers, $A=A_3A_2A_1A_0$ and $B=B_3B_2B_1B_0$.
- Internally, the operation of 4-bit Binary subtractor is similar to that of 4-bit Binary adder.
- If the normal bits of binary number A, complemented bits of binary number B [illegible] olied to 4-bit Binary adder, ther[illegible]
- The block [illegible]vn in the following figure.



Note: A-B=A+B'+1

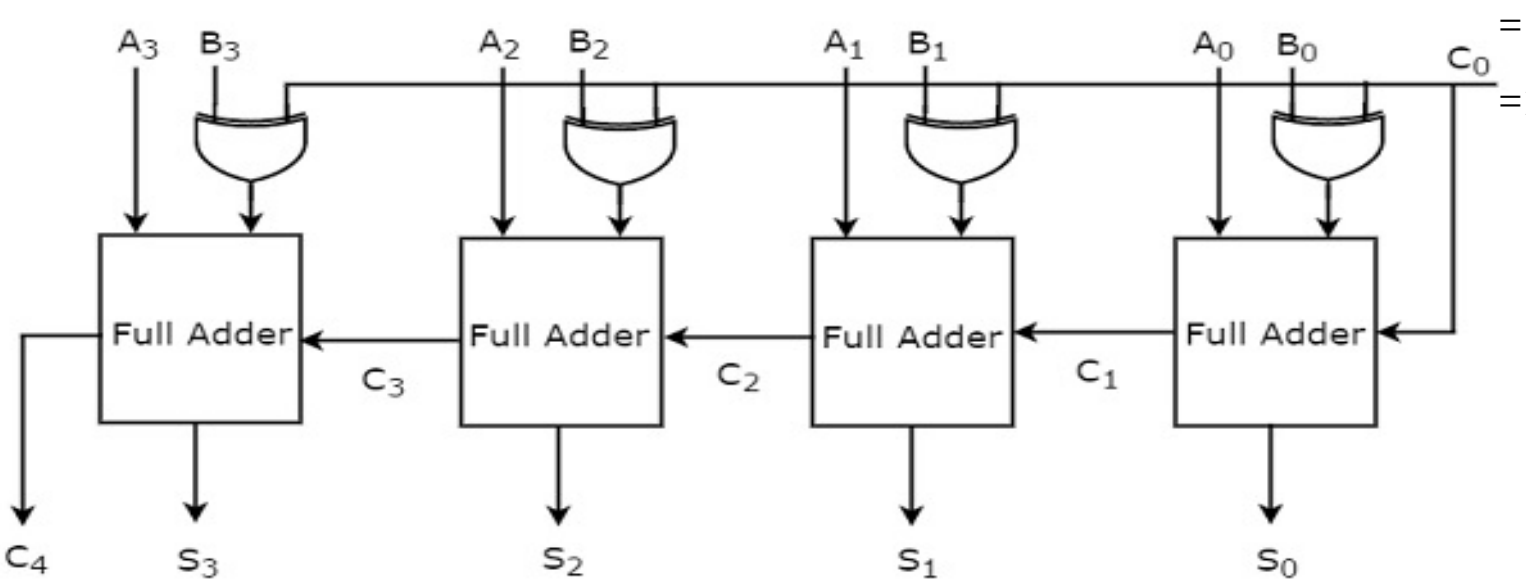
4-BIT BINARY PARALLEL SUBTRACTOR

- This 4-bit binary subtractor produces an output, which is having at most 5 bits.
- If Binary number A is greater than Binary number B, then MSB of the output is zero and the remaining bits hold the magnitude of A-B.
- If Binary number A is less than Binary number B, then MSB of the output is one. So, take the 2's complement of output in order to get the magnitude of A-B

4-BIT BINARY PARALLEL ADDER/SUBTRACTOR

- The circuit, which can be used to perform either addition or subtraction of two binary numbers at any time is known as Binary Adder / subtractor.
- Both, Binary adder and Binary subtractor contain a set of Full adders, which are cascaded.
- The input bits of binary number A are directly applied in both Binary adder and Binary subtractor.
- The input bits of binary number B are directly applied to Full adders in Binary adder, whereas the complemented bits of binary number B are applied to Full adders in Binary subtractor.
- The initial carry, $C_0 = 0$ is applied in 4-bit Binary adder, whereas the initial carry borrow, $C_0 = 1$ is applied in 4-bit Binary subtractor.
- We know that a 2-input Ex-OR gate produces an output, which is same as that of first input when other input is zero. Similarly, it produces an output, which is complement of first input when other input is one.

4-BIT BINARY PARALLEL ADDER/SUBTRACTOR



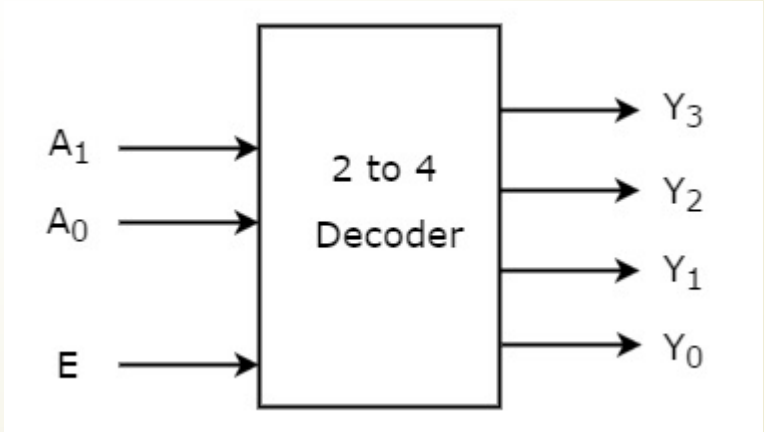
- If initial carry, C_0 is zero, then each full adder gets the normal bits of binary numbers A & B. So, the 4-bit binary adder / subtractor produces an output, which is the addition of two binary numbers A & B.
- If initial borrow, C_0 is one, then each full adder gets the normal bits of binary number A & complemented bits of binary number B. So, the 4-bit binary adder / subtractor produces an output, which is the subtraction of two binary numbers A & B.

DECODERS

- Decoder is a combinational circuit that has ‘n’ input lines and maximum of 2^n output lines.
- One of these outputs will be active High based on the combination of inputs present, when the decoder is enabled.
- That means decoder detects a particular code. The outputs of the decoder are nothing but the min terms of ‘n’ input variables lines, when it is enabled.

2 to 4 Decoder

- Let 2 to 4 Decoder has two inputs A1 & A0 and four outputs Y3, Y2, Y1 & Y0. The block diagram of 2 to 4 decoder is shown in the following figure.



DECODERS

One of these four outputs will be ‘1’ for each combination of inputs when enable, E is ‘1’. The Truth table of 2 to 4 decoder is shown below.

Enable	Inputs		Outputs			
E	A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	x	x	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

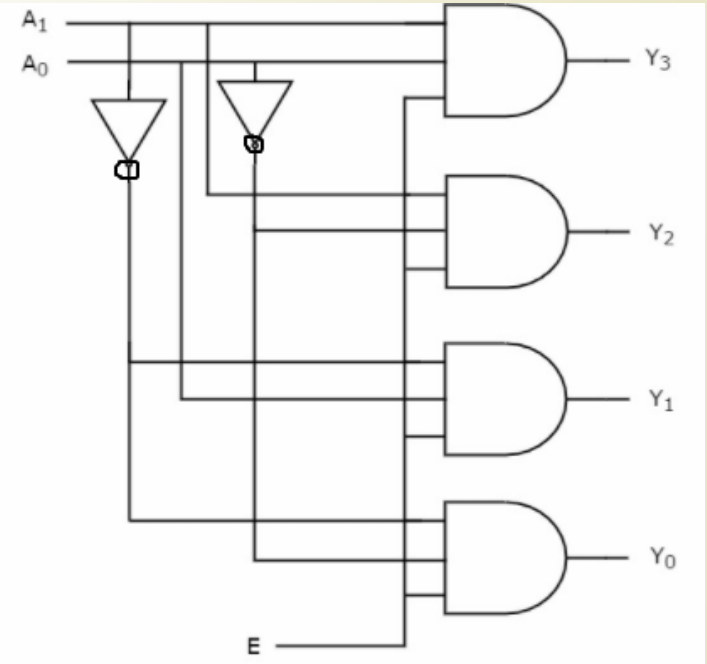
From Truth table, we can write the **Boolean functions** for each output as

$$Y_3 = E.A_1.A_0$$

$$Y_2 = E.A_1.A_0'$$

$$Y_1 = E.A_1'.A_0$$

$$Y_0 = E.A_1'.A_0'$$



DECODERS

3 X 8 DECODER:

Truth Table

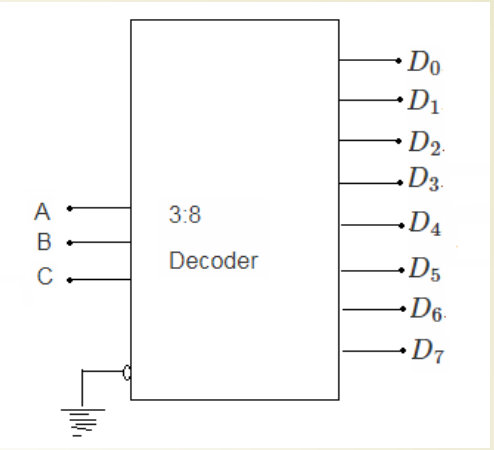
A	B	C	D0	D1	D2	D3	D4	D5	D6	D7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Function

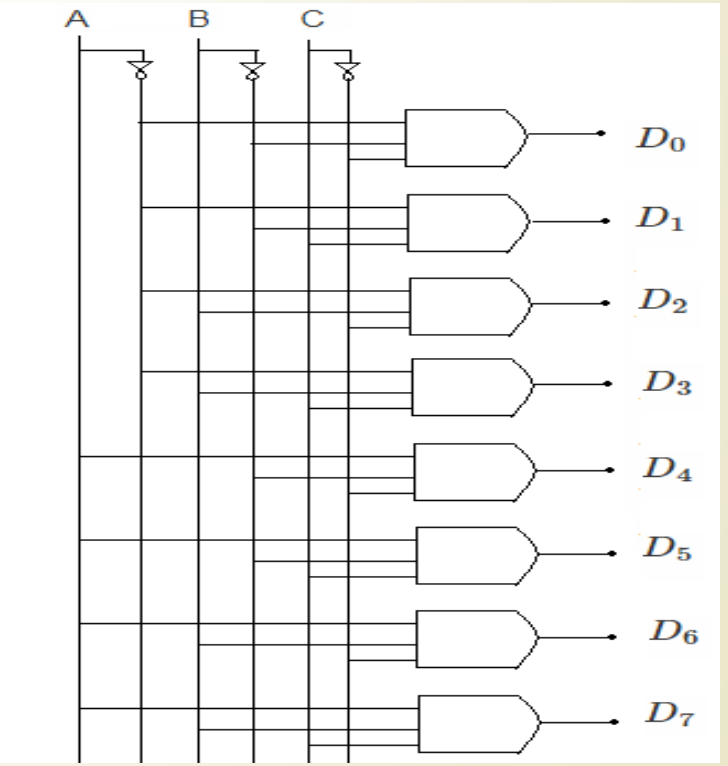
$$D_0 = \bar{A}\bar{B}\bar{C}, \quad D_1 = \bar{A}\bar{B}C, \quad D_2 = \bar{A}B\bar{C},$$

$$D_3 = \bar{A}BC, \quad D_4 = A\bar{B}\bar{C}, \quad D_5 = A\bar{B}C,$$

$$D_6 = AB\bar{C}, \quad D_7 = ABC$$



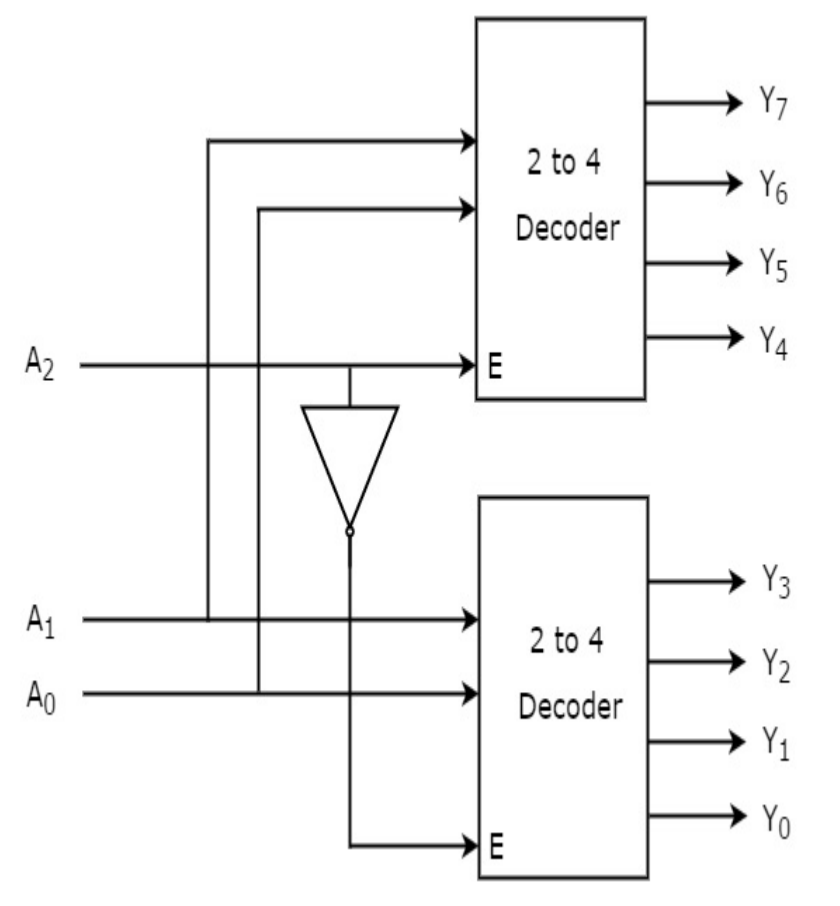
Block Diagram



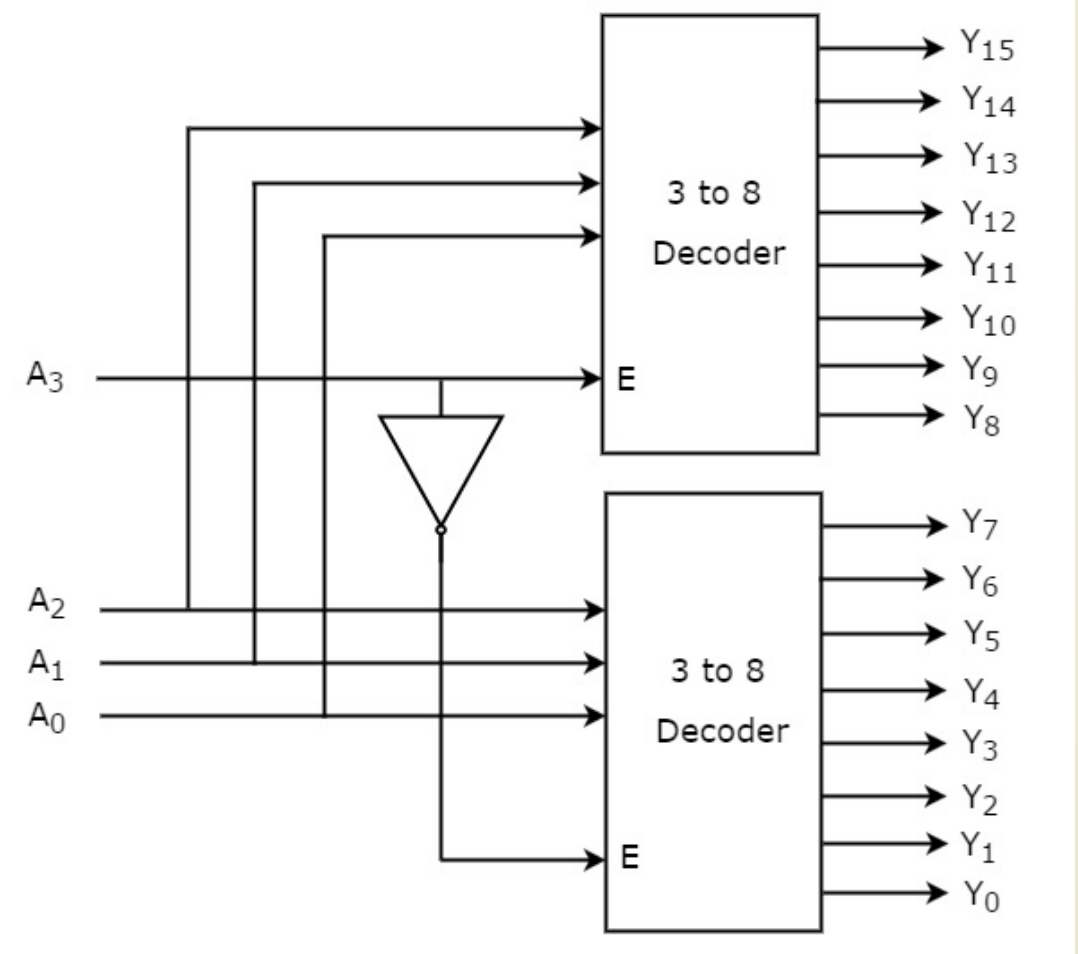
Logic Diagram

IMPLEMENTATION OF HIGHER ORDER DECODERS

3 to 8 decoder using 2 to 4 decoder

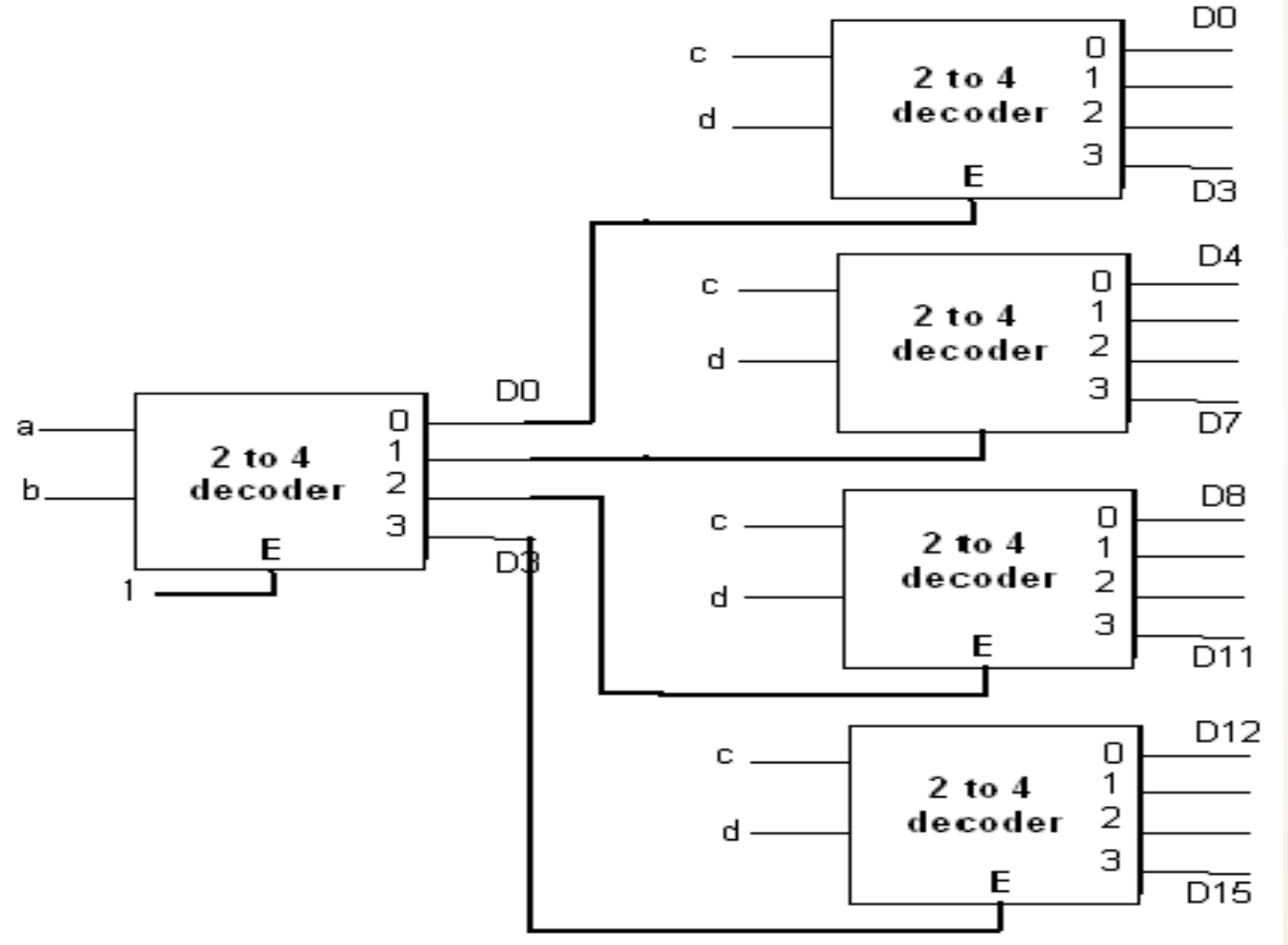


4 to 16 decoder using 3 to 8 decoder



IMPLEMENTATION OF HIGHER ORDER DECODERS

4 to 16 decoder using 2 to 4 decoder

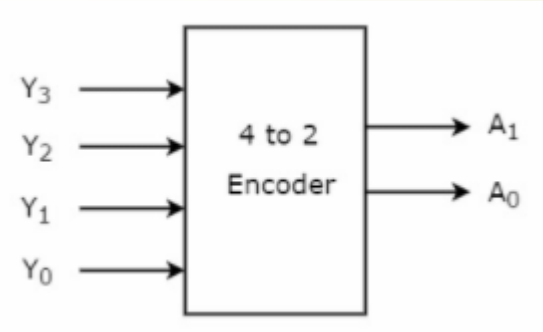


ENCODERS

- An Encoder is a combinational circuit that performs the reverse operation of Decoder.
- It has maximum of 2^n input lines and 'n' output lines.
- It will produce a binary code equivalent to the input, which is active High.
- Therefore, the encoder encodes 2^n input lines with 'n' bits. It is optional to represent the enable signal in encoders.

4 to 2 Encoder

- Let 4 to 2 Encoder has four inputs Y_3, Y_2, Y_1 & Y_0 and two outputs A_1 & A_0. The block diagram of 4 to 2 Encoder is shown in the following figure.



ENCODERS

At any time, only one of these 4 inputs can be '1' in order to get the respective binary code at the output.

The Truth table of 4 to 2 encoder is shown below.

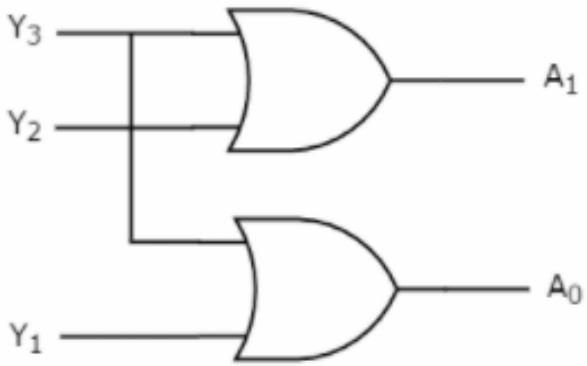
Inputs				Outputs	
Y_3	Y_2	Y_1	Y_0	A_1	A_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

From Truth table, we can write the **Boolean functions** for each output as

$$A_1 = Y_3 + Y_2$$

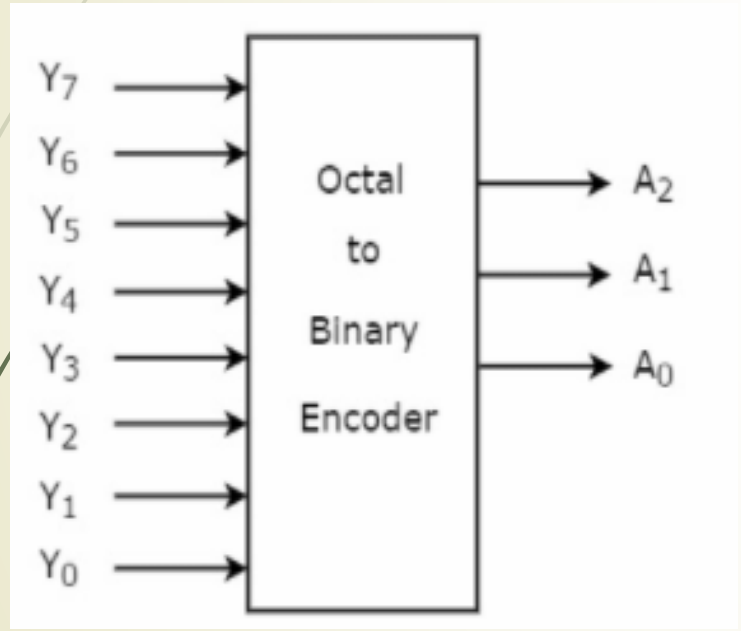
$$A_0 = Y_3 + Y_1$$

Logic Diagram



ENCODERS – OCTAL TO BINARY ENCODER

- Octal to binary Encoder has eight inputs, Y7 to Y0 and three outputs A2, A1 & A0.
- Octal to binary encoder is nothing but 8 to 3 encoder. The block diagram of octal to binary Encoder is shown in the following figure



Truth Table and Function

Inputs								Outputs		
Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0	A_2	A_1	A_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

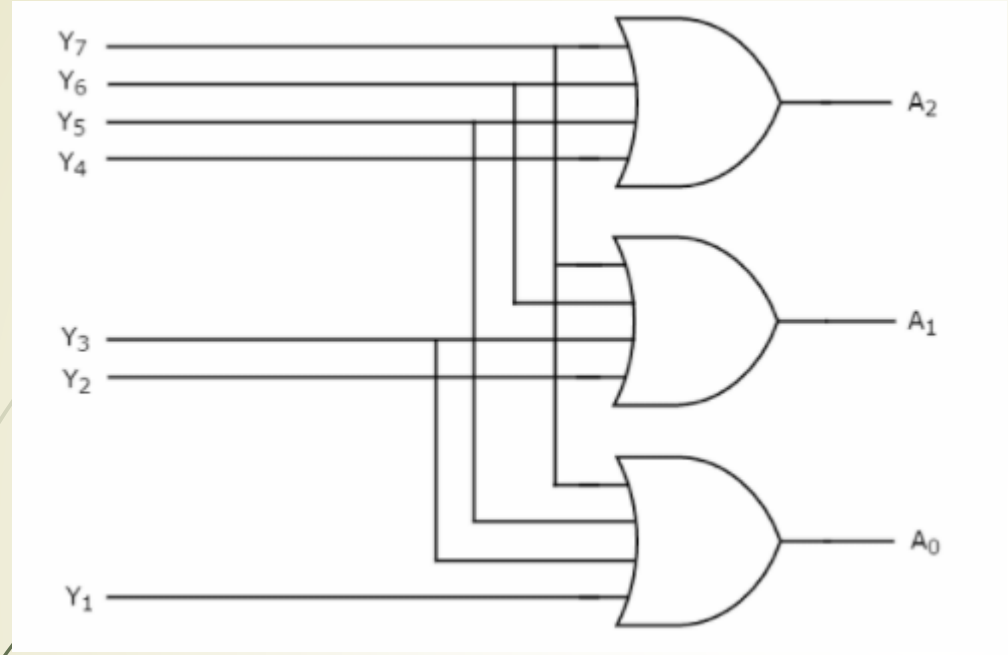
From Truth table, we can write the **Boolean functions** for each output as

$$A_2 = Y_7 + Y_6 + Y_5 + Y_4$$

$$A_1 = Y_7 + Y_6 + Y_3 + Y_2$$

$$A_0 = Y_7 + Y_5 + Y_3 + Y_1$$

ENCODERS – OCTAL TO BINARY ENCODER



Logic Diagram

Drawbacks of Encoder

- There is an ambiguity, when all outputs of encoder are equal to zero
- If more than one input is active High, then the encoder produces an output, which may not be the correct code.

ENCODERS – PRIORITY ENCODER

We considered one more output, V in order to know, whether the code available at outputs is valid or not.

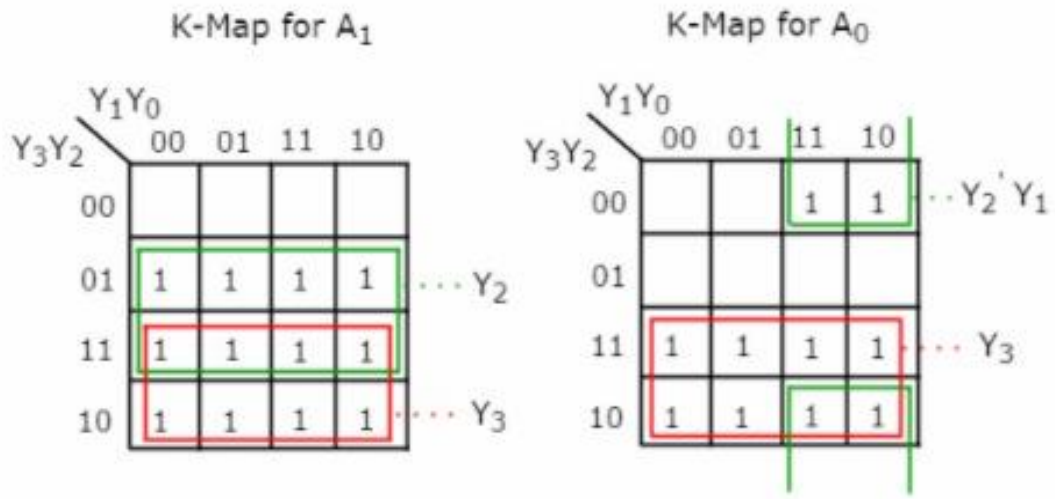
- If at least one input of the encoder is ‘1’, then the code available at outputs is a valid one. In this case, the output, V will be equal to 1.
- If all the inputs of encoder are ‘0’, then the code available at outputs is not a valid one. In this case, the output, V will be equal to 0.

Truth Table

Inputs				Outputs		
Y_3	Y_2	Y_1	Y_0	A_1	A_0	V
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	x	0	1	1
0	1	x	x	1	0	1
1	x	x	x	1	1	1

ENCODERS – PRIORITY ENCODER

K-MAP and Function



The simplified **Boolean functions** are

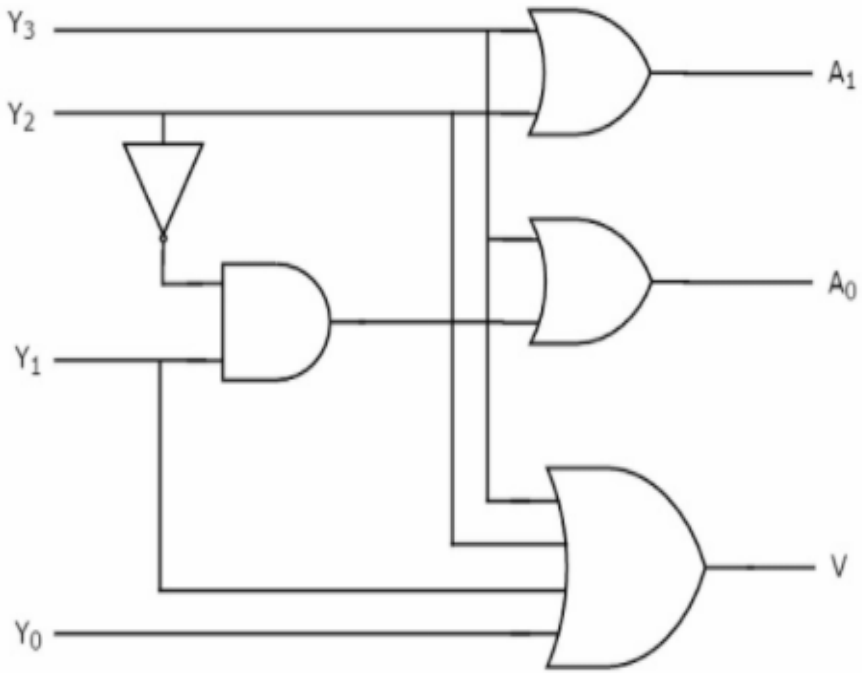
$$A_1 = Y_3 + Y_2$$

$$A_0 = Y_3 + Y_2'Y_1$$

Similarly, we will get the Boolean function of output, V as

$$V = Y_3 + Y_2 + Y_1 + Y_0$$

Logic Diagram

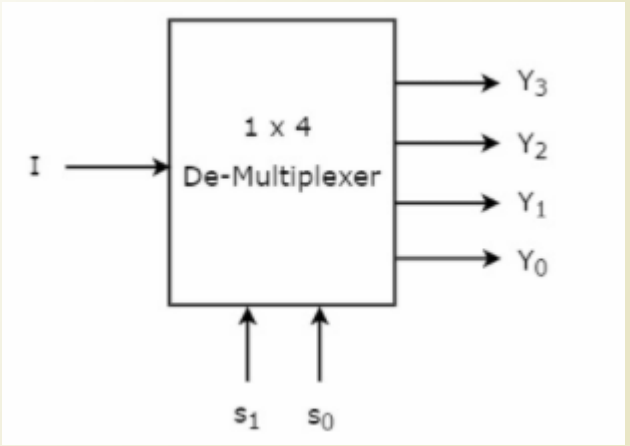


DEMULTIPLEXERS (DEMUX)

- A demultiplexer (or demux) is a device that takes a single input line and routes it to one of several digital output lines.
- A demultiplexer of 2^n outputs has n select lines, which are used to select which output line to send the input.
- A demultiplexer is also called a data distributor.

1x4 De-Multiplexer

- 1x4 De-Multiplexer has one input I, two selection lines, s1 & s0 and four outputs Y3, Y2, Y1 &Y0. The block diagram of 1x4 De-Multiplexer is shown in the following figure.



DEMULTIPLEXERS (DEMUX)

Truth Table

Selection Inputs		Outputs			
S_1	S_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	I
0	1	0	0	I	0
1	0	0	I	0	0
1	1	I	0	0	0

From the above Truth table, we can directly write the **Boolean functions** for each output as

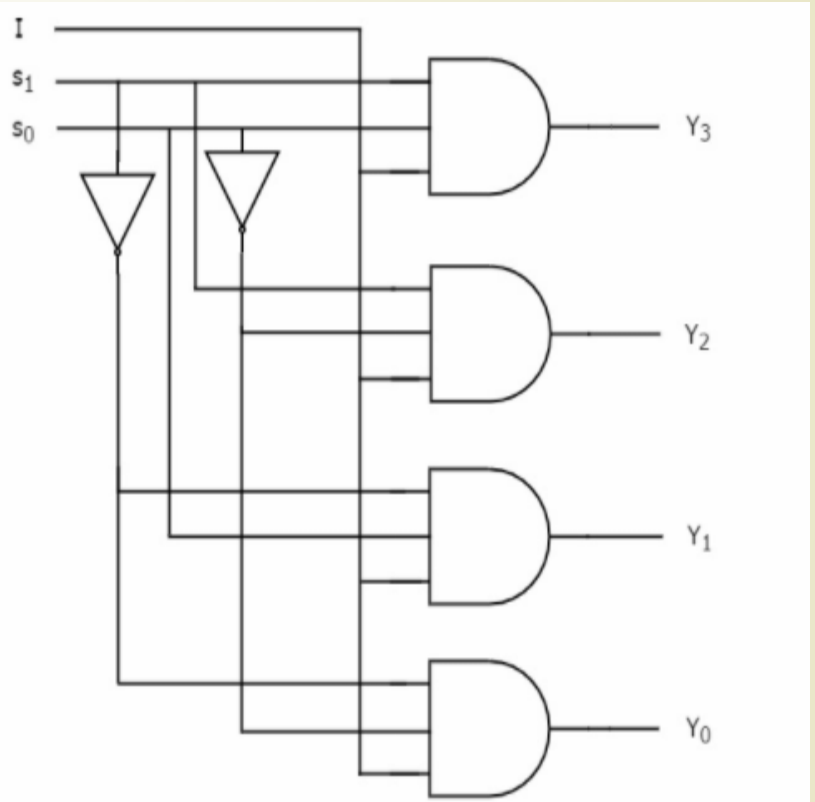
$$Y_3 = s_1 s_0 I$$

$$Y_2 = s_1 s_0' I$$

$$Y_1 = s_1' s_0 I$$

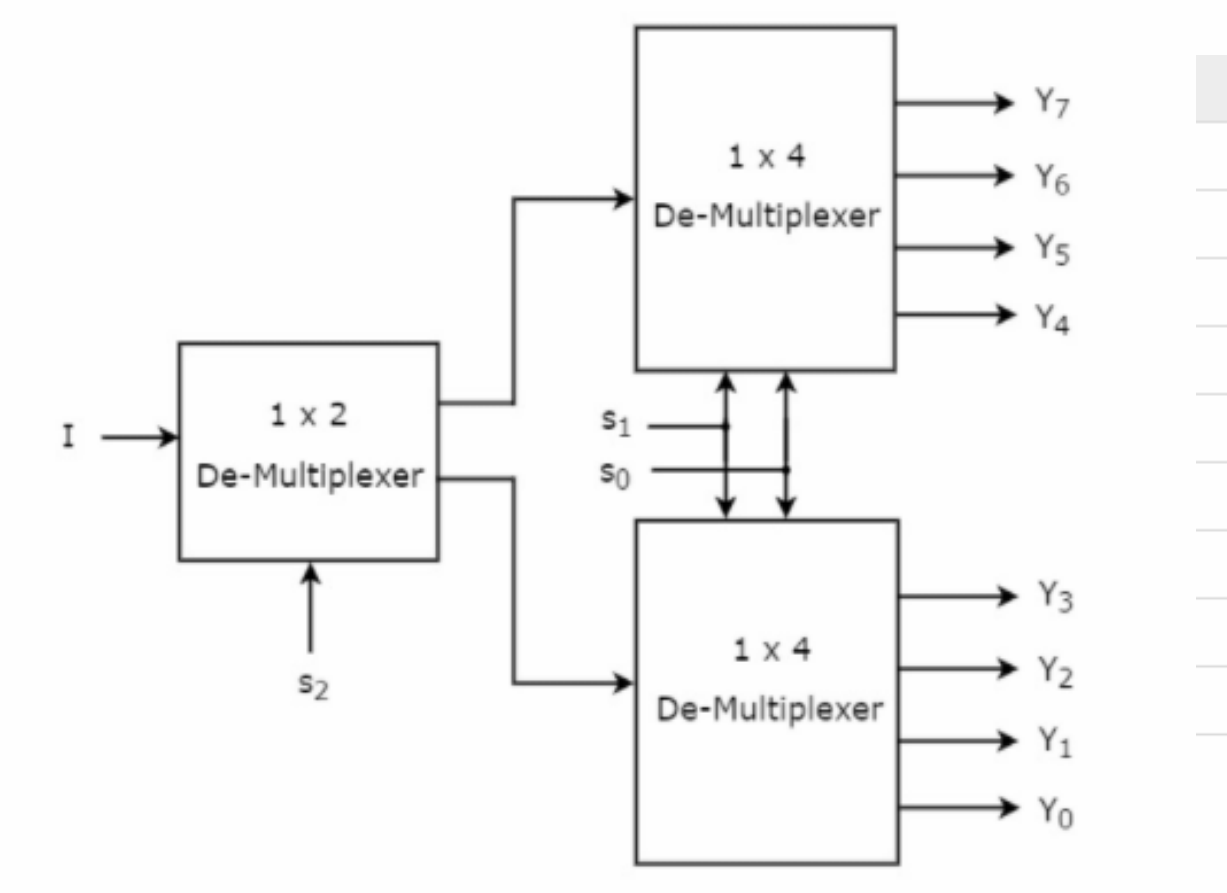
$$Y_0 = s_1' s_0' I$$

Logic Diagram



IMPLEMENTATION OF HIGHER ORDER DE-MULTIPLEXERS

1 X 8 DE-MUX using 1 X 4 and 1 X 2 DE-MUX



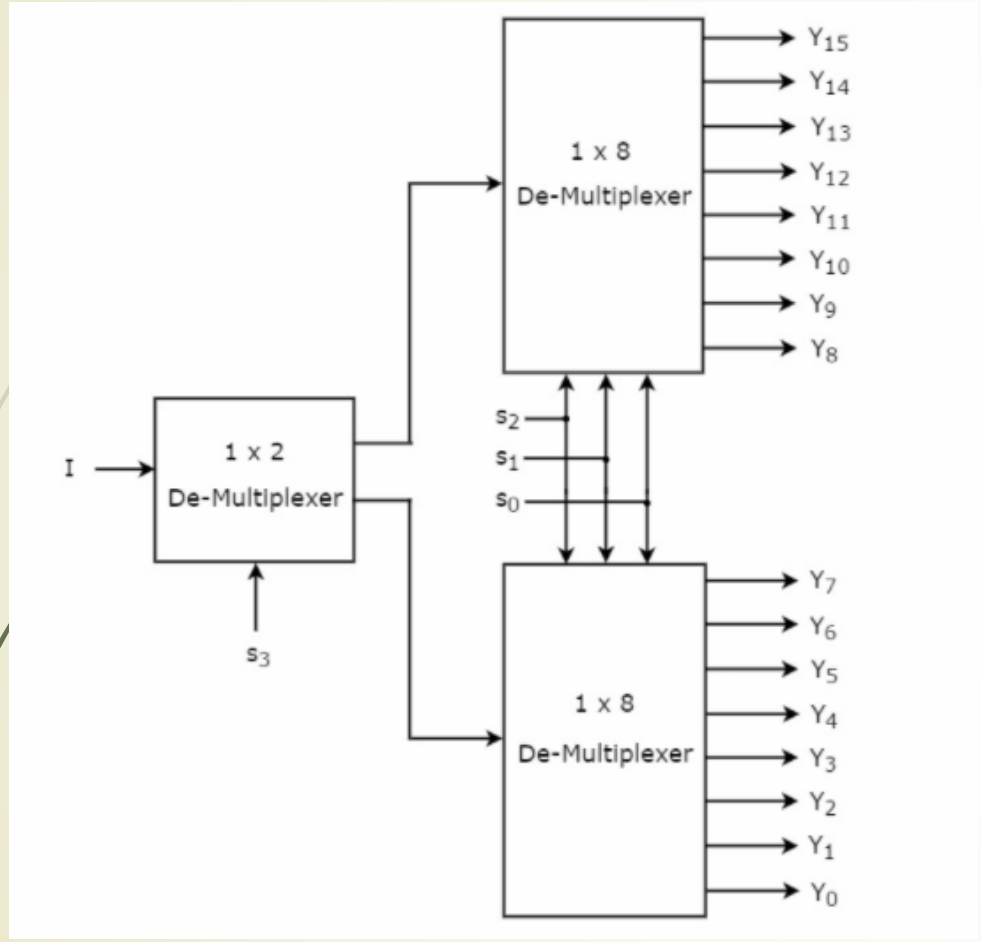
Logic Diagram

Truth Table

Selection Inputs			Outputs							
s_2	s_1	s_0	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	0	I
0	0	1	0	0	0	0	0	0	I	0
0	1	0	0	0	0	0	0	I	0	0
0	1	1	0	0	0	0	I	0	0	0
1	0	0	0	0	0	I	0	0	0	0
1	0	1	0	0	I	0	0	0	0	0
1	1	0	0	I	0	0	0	0	0	0
1	1	1	I	0	0	0	0	0	0	0

IMPLEMENTATION OF HIGHER ORDER DE-MULTIPLEXERS

1 X 16 DE-MUX using 1 x 8 and 1 x 2 DE-MUX



Logic Diagram

1 X 16 DE-MUX using 1 x 4

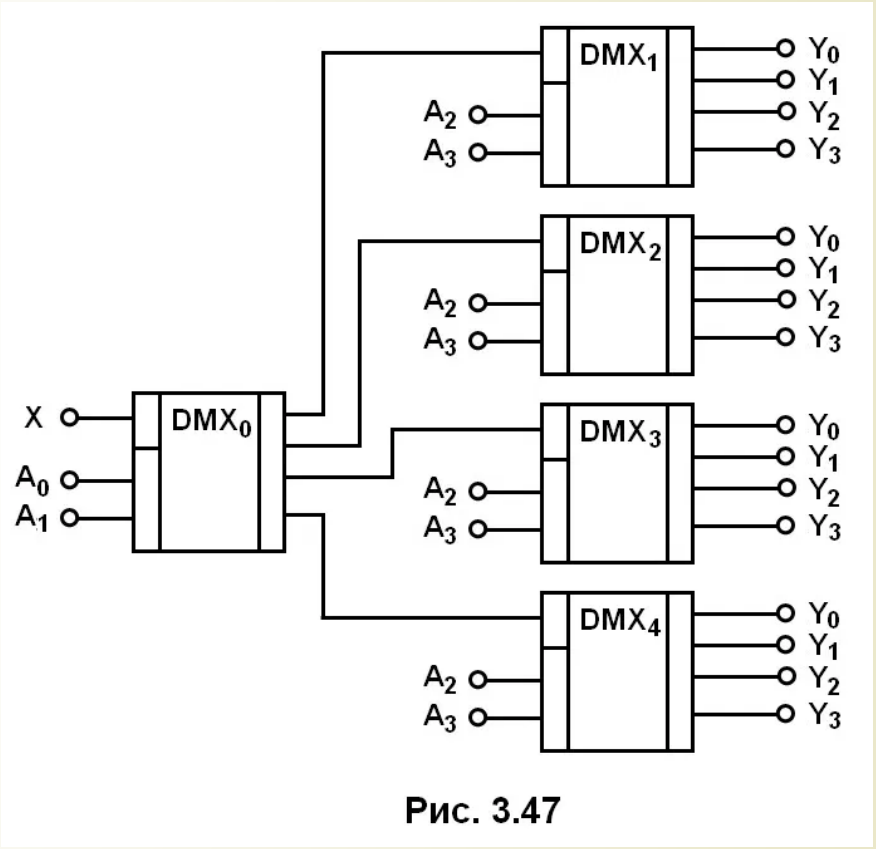


Рис. 3.47

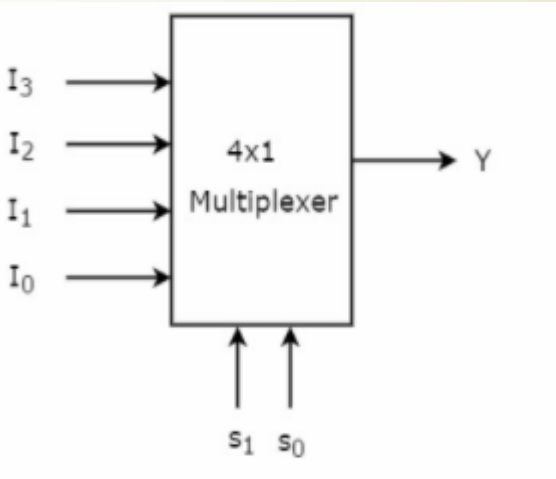
X = INPUT, A_3,A_2,A_1,A_0 are selection line

MULTIPLEXERS (MUX)

- Multiplexer is a combinational circuit that has maximum of 2^n data inputs, ‘n’ selection lines and single output line.
- One of these data inputs will be connected to the output based on the values of selection lines.
- Since there are ‘n’ selection lines, there will be 2^n possible combinations of zeros and ones.
- So, each combination will select only one data input. Multiplexer is also called as Mux.

4x1 Multiplexer

- 4x1 Multiplexer has four data inputs I_3, I_2, I_1 & I_0, two selection lines s_1 & s_0 and one output Y. The block diagram of 4x1 Multiplexer is shown in the following figure.



MULTIPLEXERS (MUX)

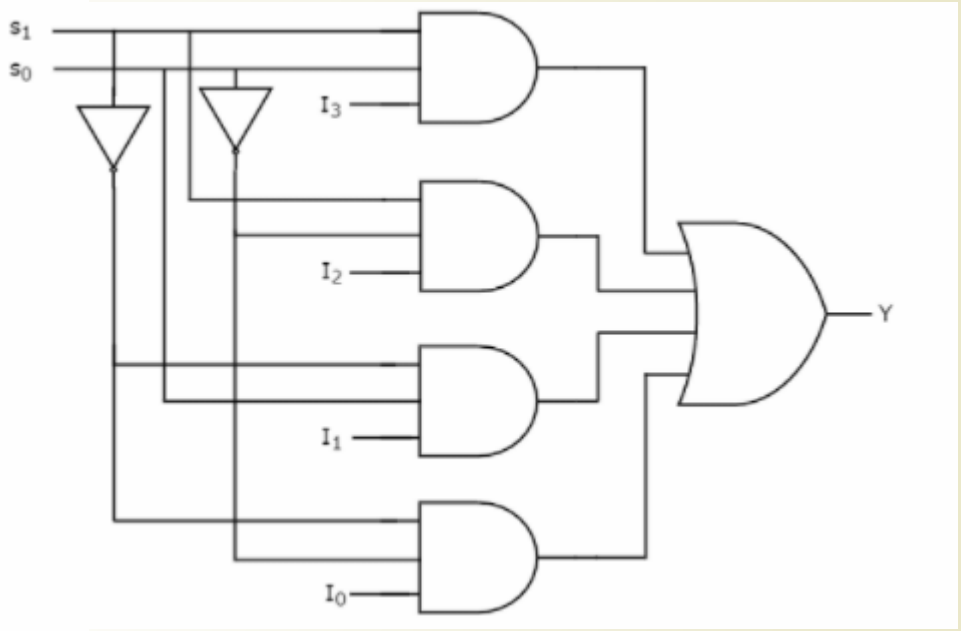
Truth table of 4x1 Multiplexer is shown below.

Selection Lines		Output
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

From Truth table, we can directly write the **Boolean function** for output, Y as

$$Y = S_1'S_0'I_0 + S_1'S_0I_1 + S_1S_0'I_2 + S_1S_0I_3$$

Logic Diagram



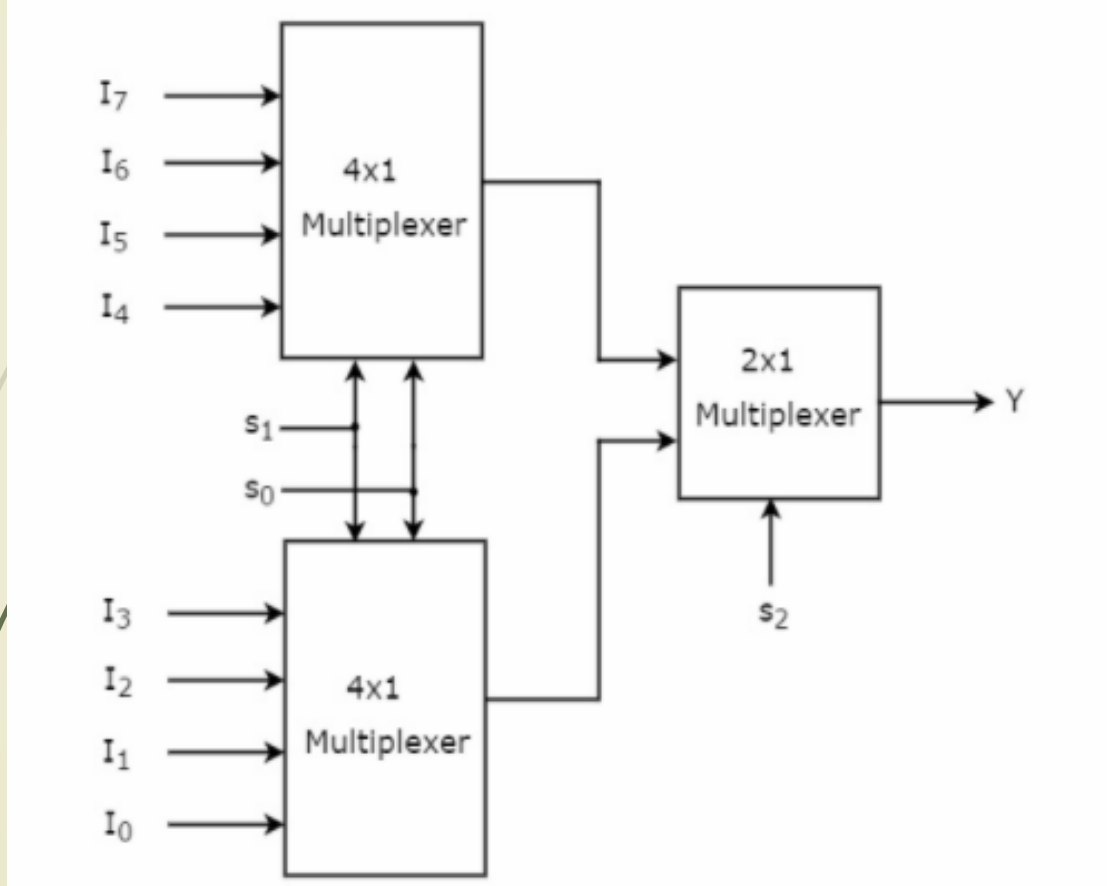
MULTIPLEXERS (MUX)

Application

- Communication system
- Telephone network
- Computer memory
- Transmission from the computer system of a satellite
- Data acquisition system

IMPLEMENTATION OF HIGHER ORDER MULTIPLEXERS

8 X 1 MUX using 4 x 1 and 2 x 1 MUX

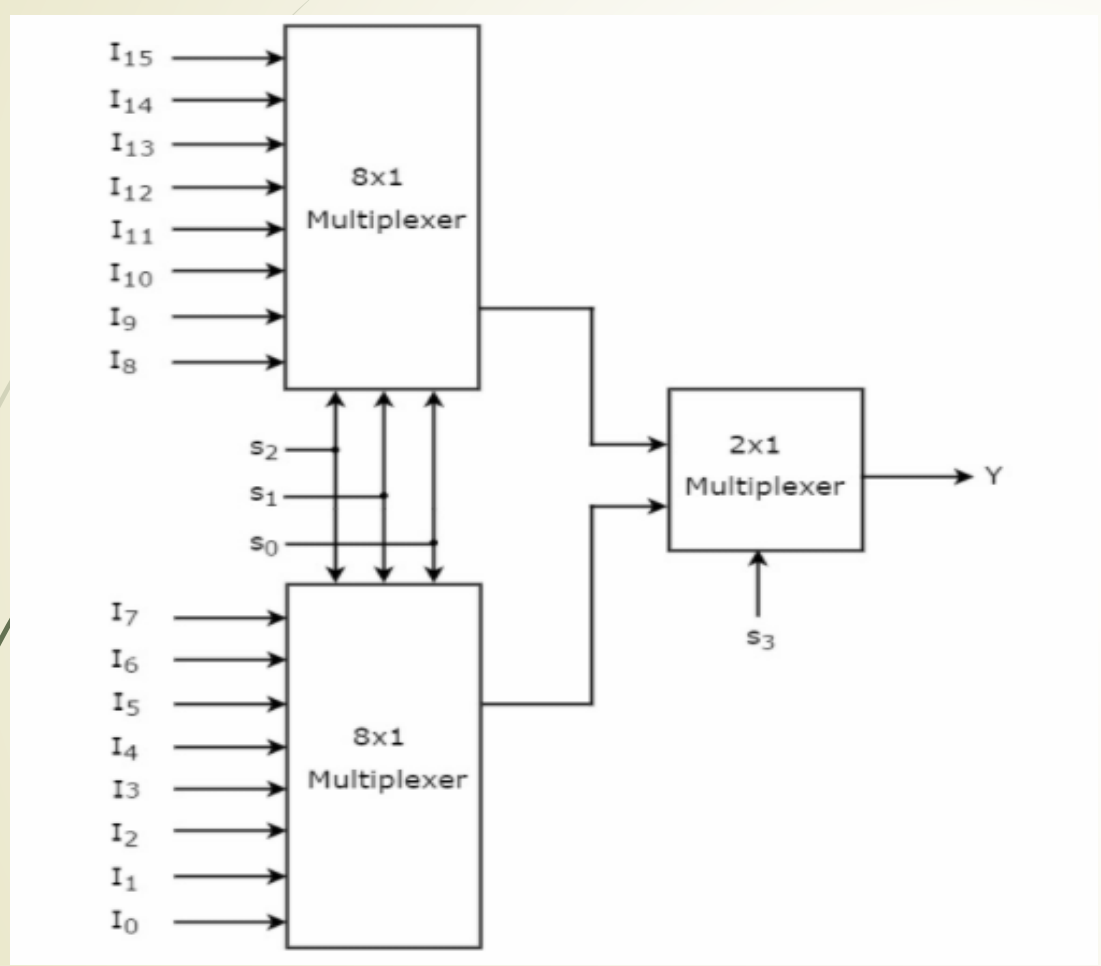


Truth Table

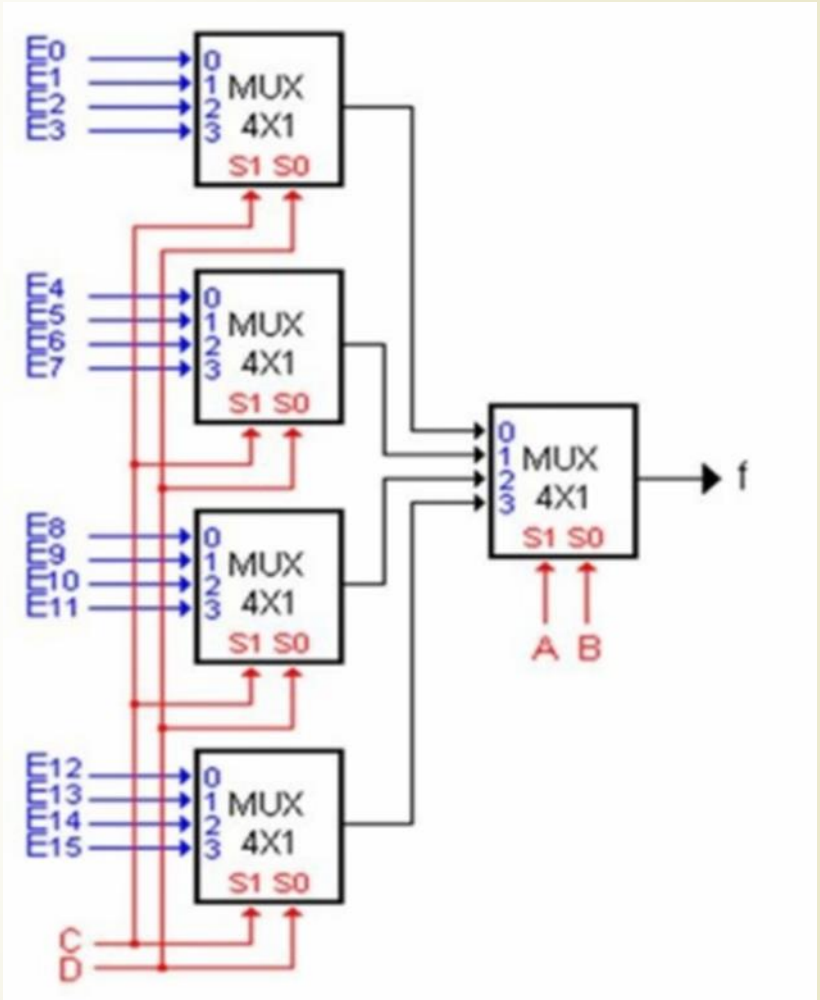
Selection Inputs			Output
S_2	S_1	S_0	Y
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7

IMPLEMENTATION OF HIGHER ORDER MULTIPLEXERS

16 X 1 MUX using 8 x 1 and 2 x 1 MUX



16 X 1 MUX using 4 x 1



BINARY ADDITION

Binary Addition Rules

		Carry Over	Result
1.	0 + 0	0	0
2.	0 + 1	0	1
3.	1 + 0	0	1
4.	1 + 1	1	0
5.	1 + 1 + 1	1	1

BINARY SUBTRACTION

Rules of Binary Subtraction

$\Rightarrow 1 - 0 = 1$

$\Rightarrow 1 - 1 = 0$

$\Rightarrow 0 - 0 = 0$

$\Rightarrow 0 - 1 = 1$

(This can not be done directly, hence we borrow one digit from the digit to the left or the next higher order digit.)

SIGNED AND UNSIGNED BINARY NUMBERS

Unsigned Numbers

As we already know, the unsigned numbers don't have any sign for representing negat numbers. So the unsigned numbers are always positive. By default, the decimal numb representation is positive. We always assume a positive sign in front of each decimal digit.

Signed Numbers

The signed numbers have a sign bit so that it can differentiate positive and negative integer numbers. The signed binary number technique has both the sign bit and the magnitude of the number. For representing the negative decimal number, the corresponding symbol in front of the binary number will be added.

SIGNED AND UNSIGNED BINARY NUMBERS

1. **Sign-Magnitude form**

   In this form, a binary number has a bit for a sign symbol. If this bit is set to 1, the number will be negative else the number will be positive if it is set to 0. Apart from this sign-bit, the n-1 bits represent the magnitude of the number.

2. **1's Complement**

   By inverting each bit of a number, we can obtain the 1's complement of a number. The negative numbers can be represented in the form of 1's complement. In this form, the binary number also has an extra bit for sign representation as a sign-magnitude form.

3. **2's Complement**

   By inverting each bit of a number and adding plus 1 to its least significant bit, we can obtain the 2's complement of a number. The negative numbers can also be represented in the form of 2's complement. In this form, the binary number also has an extra bit for sign representation as a sign-magnitude form.

Combinational and Arithmetic Circuits – SET A

1. Which of the following is correct for multiplexer?
a) Several inputs and single output
b) Single input and several outputs
c) Single input and single output
d) Several inputs and several outputs

2. Multiplexers work with _____
a) Analog signal
b) Digital signal
c) Both analog and digital signal
d) None of the mentioned

3. TDM stands for _____
a) Time direct measurement
b) Time division multiplexing
c) Time direct multiplexing
d) Time division measurement

4. Which of the following is analogous to multiplexer?
a) Data selector
b) Data multiplexer
c) Data filter
d) None of the mentioned

5. Which of the following represent multiple input single output switch?
a) Multiplexer
b) De multiplexer
c) Both multiplexer and demultiplexer
d) None of the mentioned

6. Schematic symbol of multiplexer is _____
a) Isosceles triangle
b) Isosceles trapezoid
c) Equilateral triangle
d) Rectangle

Combinational and Arithmetic Circuits – SET A

7. In digital multiplexer selector line is _____
a) Analog value
b) Digital value
c) Unpredictable
d) None of the mentioned

8. Which of the following is not a multiplexer?
a) 8-to-1 line
b) 16-to-1 line
c) 4-to-1 line
d) 1-to-4 line

Combinational and Arithmetic Circuits – SET B

1. What is a multiplexer?
a) It is a type of decoder which decodes several inputs and gives one output
b) A multiplexer is a device which converts many signals into one
c) It takes one input and results into many output
d) It is a type of encoder which decodes several inputs and gives one output

2. Which combinational circuit is renowned for selecting a single input from multiple inputs & directing the binary information to output line?
a) Data Selector
b) Data distributor
c) Both data selector and data distributor
d) DeMultiplexer

3. The enable input is also known as _____
a) Select input
b) Decoded input
c) Strobe
d) Sink

4. Which is the major functioning responsibility of the multiplexing combinational circuit?
a) Decoding the binary information
b) Generation of all minterms in an output function with OR-gate
c) Generation of selected path between multiple sources and a single destination
d) Encoding of binary information

5. What is the function of an enable input on a multiplexer chip?
a) To apply Vcc
b) To connect ground
c) To active the entire chip
d) To active one half of the chip

6. One multiplexer can take the place of _____
a) Several SSI logic gates
b) Combinational logic circuits
c) Several Ex-NOR gates
d) Several SSI logic gates or combinational logic circuits

Combinational and Arithmetic Circuits – SET B

7. A digital multiplexer is a combinational circuit that selects _____
One digital information from several sources and transmits the selected one
b) Many digital information and convert them into one
c) Many decimal inputs and transmits the selected information
d) Many decimal outputs and accepts the selected information

8. In a multiplexer, the selection of a particular input line is controlled by _____
a) Data controller
Selected lines
c) Logic gates
d) Both data controller and selected lines

9. If the number of n selected input lines is equal to 2^m then it requires ____ select lines.
a) 2
m
c) n
d) 2^n

10. How many select lines would be required for an 8-line-to-1-line multiplexer?
a) 2
b) 4
c) 8
3

12. How many NOT gates are required for the construction of a 4-to-1 multiplexer?
a) 3
b) 4
2
d) 5

Combinational and Arithmetic Circuits – SET C

3. On subtracting (01010)2 from (11110)2 using 1's complement, we get _____
a) 01001
b) 11010
c) 10101
d) 10100

4. On subtracting (010110)2 from (1011001)2 using 2's complement, we get _____
a) 0111001
b) 1100101
c) 0110110
d) 1000011

5. On subtracting (001100)2 from (101001)2 using 2's complement, we get _____
a) 1101100
b) 011101
c) 11010101
d) 11010111

6. On addition of 28 and 18 using 2's complement, we get _____
a) 00101110
b) 0101110
c) 00101111
d) 1001111

7. On addition of +38 and -20 using 2's complement, we get _____
a) 11110001
b) 100001110
c) 010010
d) 110101011

8. On addition of -46 and +28 using 2's complement, we get _____
a) -10010
b) -00101
c) 01011
d) 0100101

Combinational and Arithmetic Circuits – SET D

1. A decoder converts n inputs to _____ outputs
a) n
b) n^2
c) 2^n
d) n^n

2. Which of the following are building blocks of encoders
a) NOT gate
b) OR gate
c) AND gate
d) NAND gate

3. Which of the following can be represented for decoder?
a) Sequential circuit
b) Combinational circuit
c) Logical circuit
d) None of the mentioned

4. BCD to seven segment conversion is a _____
a) Decoding process
b) Encoding process
c) Comparing process
d) None of the mentioned

7. Invalid BCD can be made to valid BCD by adding with _____
a) 0101
b) 0110
c) 0111
d) 1001

8. Decoder is constructed from _____
a) Inverters
b) AND gates
c) Inverters and AND gates
d) None of the mentioned

Combinational and Arithmetic Circuits – SET D

9. Which of the following represents a number of output lines for a decoder with 4 input lines?

a) 15
b) 16
c) 17
d) 18

15). A 4 to 2 encoder requires ___ number of logic gates?

- 2
- 3
- 4
- 5

26). An encoder generates ___ type code on each input?

- Hexa
- Binary
- Octal
- ASCII

7). Which of the following is the output "A,B" for an encoder with 4bits "Y1,Y2,Y3,Y4" as " 0010"?

- 00
- 01
- 10
- 11

17). Which of the following is the output for 8 to 3 type encoder for input D [7] to D [0] "00000001"?

- XXX
- 000
- 001
- 010

83). Which of the following is the output of 3to8 type decoder when input is 1,100?

- A *B bar * C bar
- A * B* C
- A+B bar +C
- A bar +B +C

2.3 Sequential logic circuit: RS Flip-Flops, Gated Flip-Flops, Edge Triggered Flip-Flops, Mater- Slave Flip-Flops. Types of Registers, Applications of Shift Registers, Asynchronous Counters, Synchronous Counters. (AExE0203)

SEQUENTIAL LOGIC

- Sequential circuit produces an output based on current input and previous input variables.
- That means sequential circuits include memory elements which are capable of storing binary information.
- That binary information defines the state of the sequential circuit at that time.
- A latch is capable of storing one bit of information.
- As shown in figure there are two types of input to the combinational logic :
  - External inputs which not controlled by the circuit.
  - Internal inputs which are a function of a previous output states.

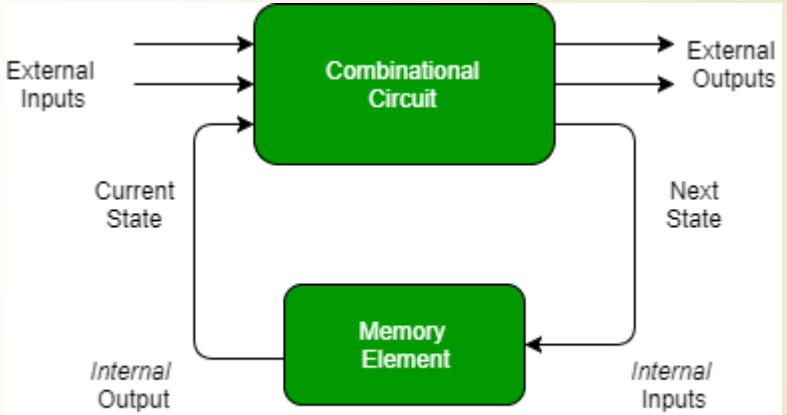


Figure: Sequential Circuit

SEQUENTIAL LOGIC - TYPES

Asynchronous sequential circuit

- These circuit do not use a clock signal but uses the pulses of the inputs.
- These circuits are faster than synchronous sequential circuits because they change their state immediately when there is a change in the input signal.
- We use asynchronous sequential circuits when speed of operation is important and independent of internal clock pulse.
- But these circuits are more difficult to design and their output is uncertain.

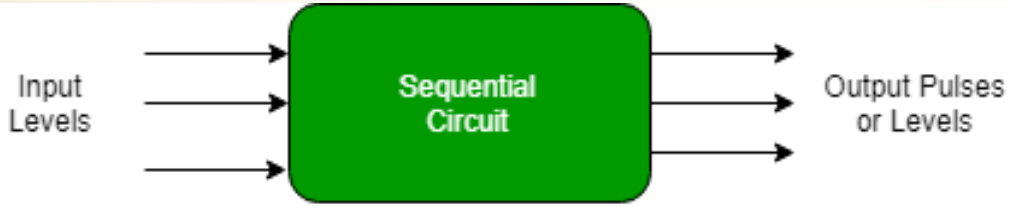


Figure: Asynchronous Sequential Circuit

SEQUENTIAL LOGIC - TYPES

Synchronous sequential circuit

- These circuit uses clock signal and level inputs (or pulsed) with restrictions on pulse width and circuit propagation.
- The output pulse is the same duration as the clock pulse for the clocked sequential circuits.
- Since they wait for the next clock pulse to arrive to perform the next operation, so these circuits are bit slower compared to asynchronous.
- Level output changes state at the start of an input pulse and remains in that until the next input or clock pulse.
- We use synchronous sequential circuit in synchronous counters,

  flip flops, and in the design of

  state management machines.

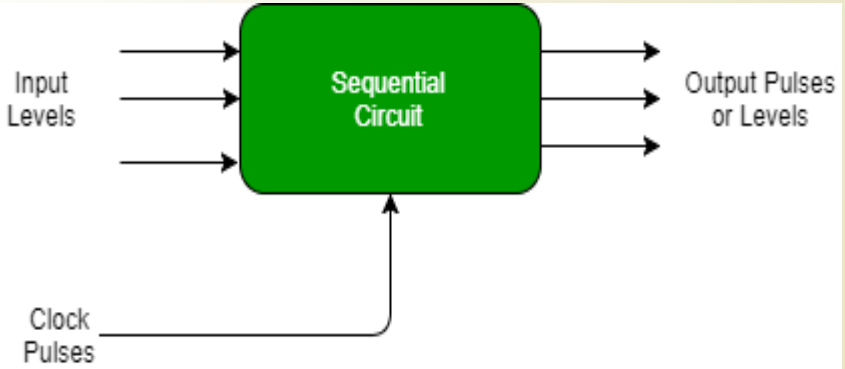
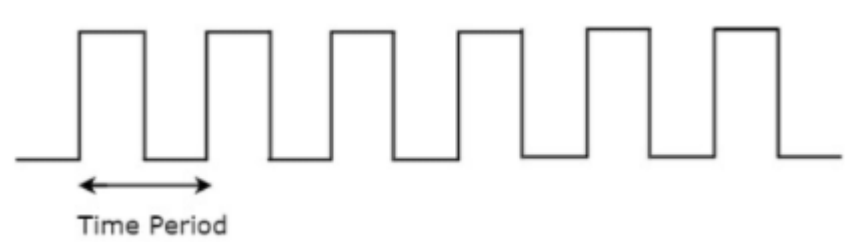


Figure: Synchronous Sequential Circuit

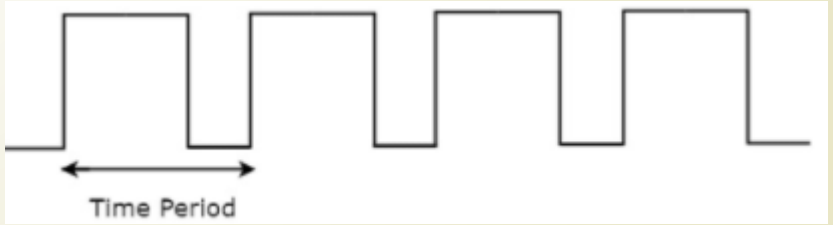
CLOCK SIGNAL AND TRIGGERING

Clock signal:

- Clock signal is a periodic signal and its ON time and OFF time need not be the same.
- We can represent the clock signal as a square wave, when both its ON time and OFF time are same.
- The pattern repeats with some time period. In this case, the time period will be equal to either twice of ON time or twice of OFF time.



- We can represent the clock signal as train of pulses, when ON time and OFF time are not same
- In this case, the time period will be equal to sum of ON time and OFF time.



CLOCK SIGNAL AND TRIGGERING

Types of Triggering

- Level triggering
- Edge triggering

Level triggering

- There are two levels, namely logic High and logic Low in clock signal.
- Following are the two types of level triggering.
  - Positive level triggering
  - Negative level triggering

- If the sequential circuit is operated with the clock signal when it is in Logic High, then that type of triggering is known as **Positive level triggering**. It is highlighted in above figure.

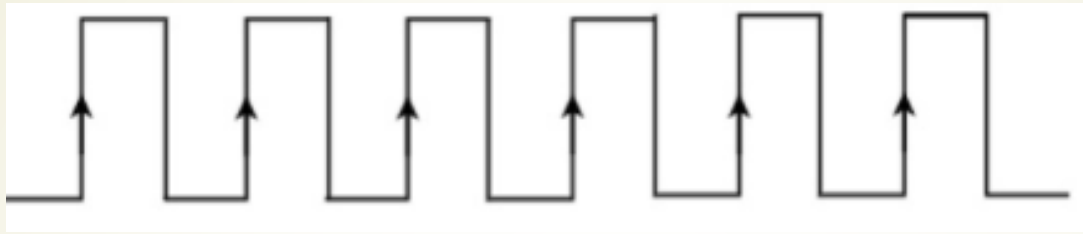
CLOCK SIGNAL AND TRIGGERING

- If the sequential circuit is operated with the clock signal when it is in Logic Low, then that type of triggering is known as **Negative level triggering**. It is highlighted in the following figure.

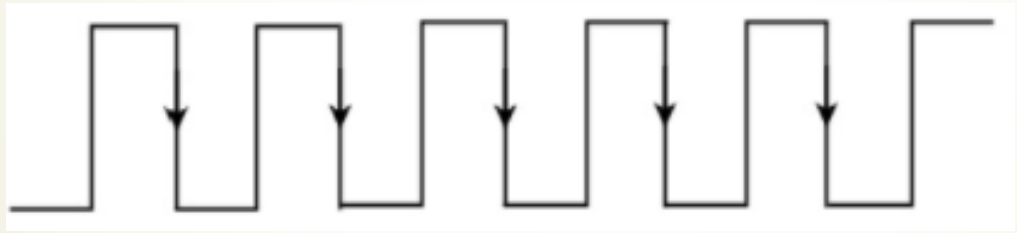
Edge triggering

- There are two types of transitions that occur in clock signal. That means, the clock signal transitions either from Logic Low to Logic High or Logic High to Logic Low..
  - Positive edge triggering
  - Negative edge triggering
- If the sequential circuit is operated with the clock signal that is transitioning from Logic Low to Logic High, then that type of triggering is known as **Positive edge triggering**. It is also called as rising edge triggering. It is shown in the following figure.

CLOCK SIGNAL AND TRIGGERING



- If the sequential circuit is operated with the clock signal that is transitioning from Logic High to Logic Low, then that type of triggering is known as **Negative edge triggering**. It is also called as falling edge triggering. It is shown in the following figure.



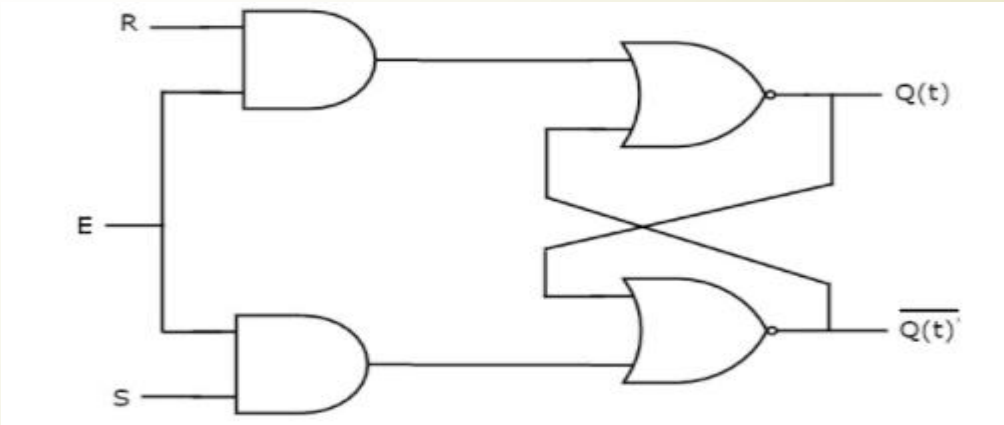
LATCHES

- There are two types of memory elements based on the type of triggering that is suitable to operate it.
  - Latches
  - Flip-flops
- Latches operate with enable signal, which is level sensitive. Whereas, flip-flops are edge sensitive.

SR Latch

- SR Latch is also called as Set Reset Latch.
- This latch affects the outputs as long as the enable, E is maintained at '1'. The circuit diagram of SR Latch is shown in the following figure

S	R	Q $t+1$
0	0	Q t
0	1	0
1	0	1
1	1	-

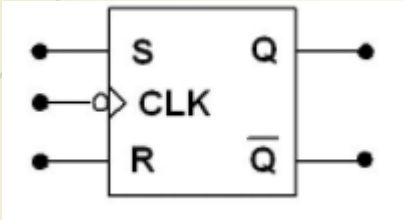


FLIP-FLOP

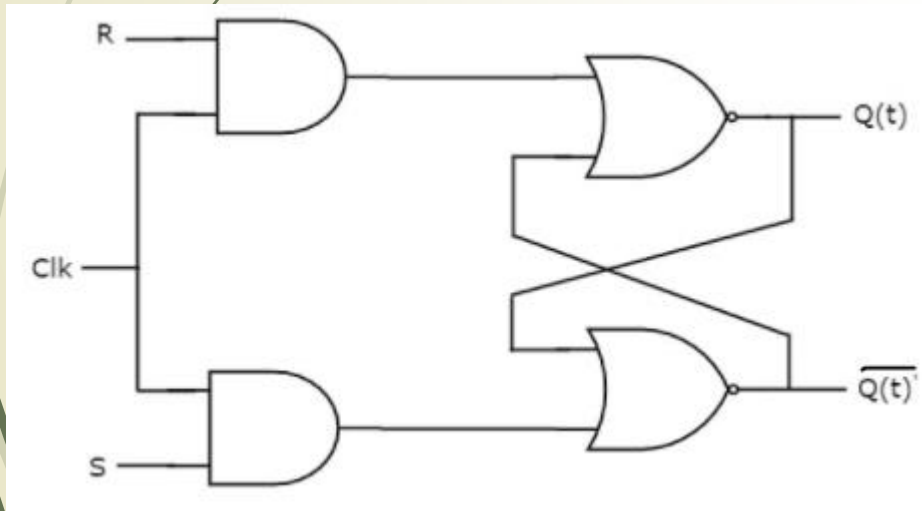
- Flip-flop is a basic digital memory circuit, which stores one bit of information.
- Flip flops are the fundamental blocks of most sequential circuits.
- Flip-flops are used as memory elements in clocked sequential circuit.
- Flip-flop circuit has two outputs, one for the normal value and one for the complement value of the bit stored in it.
- Binary information can be enter a flip-flop in a variety of ways, a fact, which gives rise to different types of flip-flops.
  - SR Flip-Flop
  - D Flip-Flop
  - JK Flip-Flop
  - T Flip-Flop

FLIP-FLOP – SR FLIP-FLOP

SR (Set-Reset) flip-flop operates with only positive clock transitions or negative clock transitions. Whereas, SR latch operates with enable signal. The circuit diagram of SR flip-flop is shown in the following figure.



Graphical Symbol



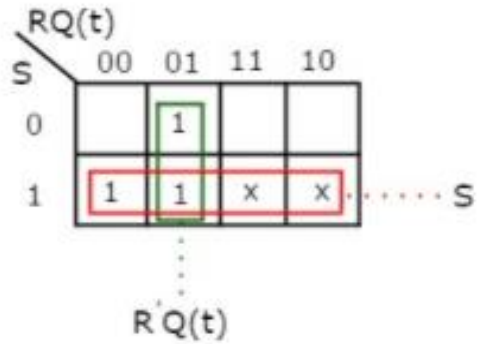
Logic diagram

Clock	S	R	Q(t)	Q(t+1)	Comment
↓	0	0	0	0	Hold
↓	0	0	1	1	Hold
↓	0	1	0	0	Reset
↓	0	1	1	0	Reset
↓	1	0	0	1	Set
↓	1	0	1	1	Set
↓	1	1	0	X	Indeterminant
↓	1	1	1	X	Indeterminant

Characteristic Table

FLIP-FLOP – SR FLIP-FLOP

K-Map for next state, Q $t+1$ is shown in the following figure.

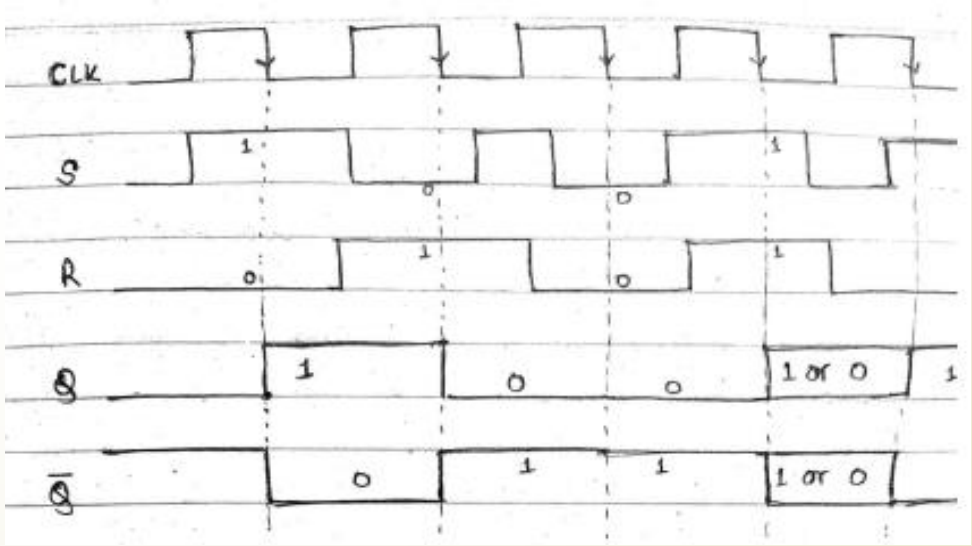


$$Q(t+1) = S + R'Q(t)$$

K-MAP and Boolean function

SR Flip-flop			
Q(t)	Q(t+1)	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

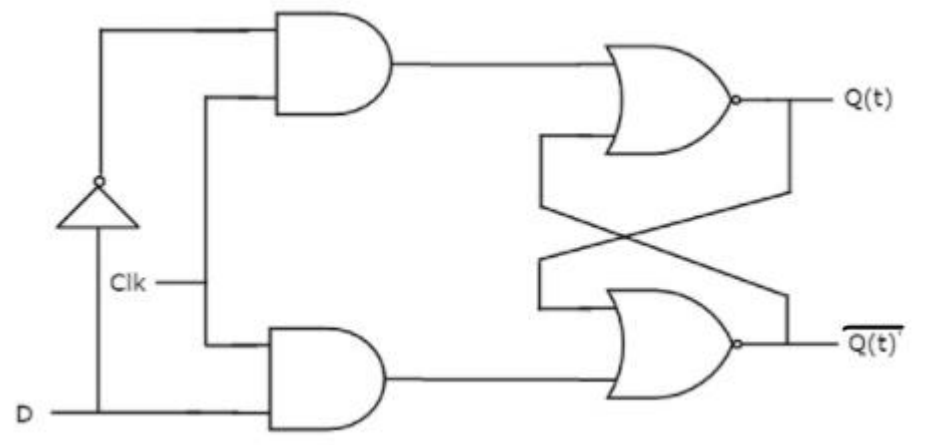
Excitation Table



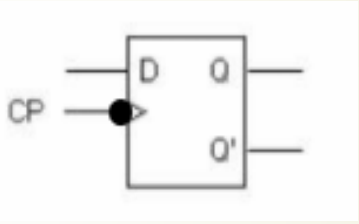
Timing diagram

FLIP-FLOP – D FLIP-FLOP

- D (Data) flip-flop operates with only positive clock transitions or negative clock transitions. Whereas, D latch operates with enable signal.
- That means, the output of D flip-flop is insensitive to the changes in the input, D except for active transition of the clock signal. The circuit diagram of D flip-flop is shown in the following figure.



Logic diagram



Graphical Symbol

Clock	D	Q(t)	Q(t+1)	Comment
↓	0	0	0	Reset
↓	0	1	0	Reset
↓	1	0	1	Set
↓	1	1	1	Set

Characteristic Table

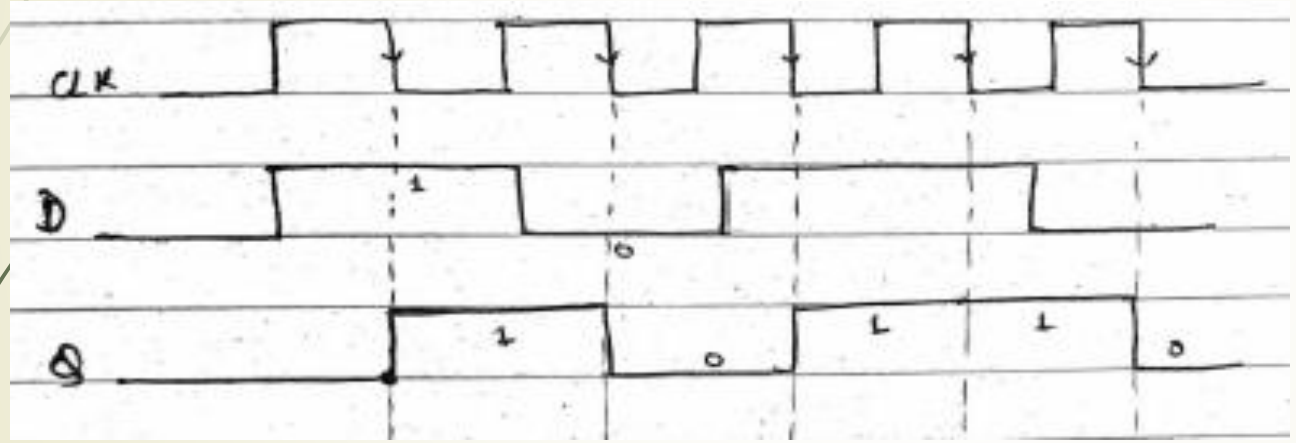
FLIP-FLOP – D FLIP-FLOP

$$Q(t+1) = D$$

Boolean function

D Flip-flop		
Q(t)	Q(t+1)	D
0	0	0
0	1	1
1	0	0
1	1	1

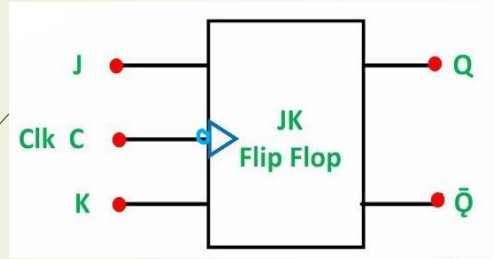
Excitation Table



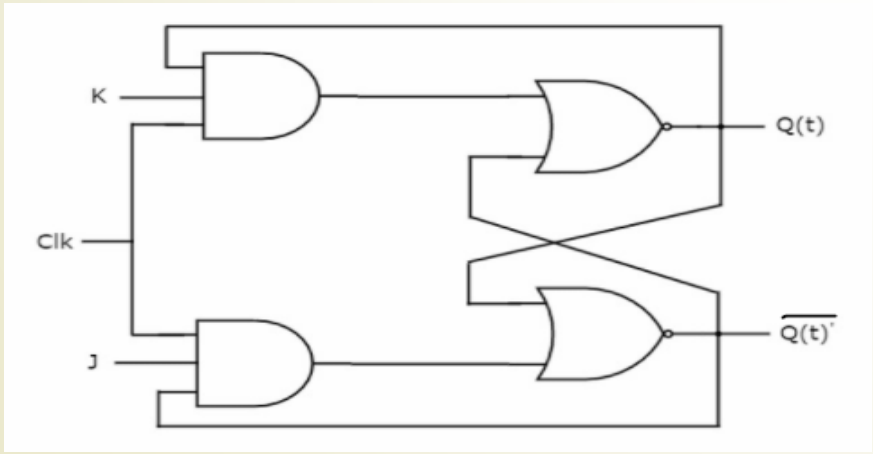
Timing diagram

FLIP-FLOP – JK FLIP-FLOP

- JK flip-flop is the modified version of SR flip-flop.
- It operates with only positive clock transitions or negative clock transitions. The circuit diagram of JK flip-flop is shown in the following figure.
- The indeterminant state of RS flip-flop is defined in JK flip-flop.



Graphical Symbol



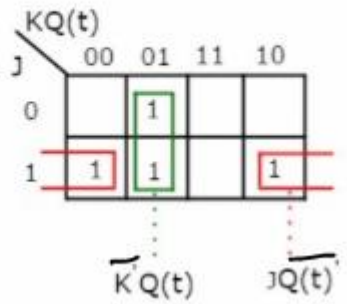
Logic diagram

Clock	J	K	Q(t)	Q(t+1)	Comment
↓	0	0	0	0	Hold
↓	0	0	1	1	Hold
↓	0	1	0	0	Reset
↓	0	1	1	0	Reset
↓	1	0	0	1	Set
↓	1	0	1	1	Set
↓	1	1	0	1	Complement
↓	1	1	1	0	Complement

Characteristic Table

FLIP-FLOP – JK FLIP-FLOP

Three variable K-Map for next state, $Q(t+1)$ is shown in the following figure.

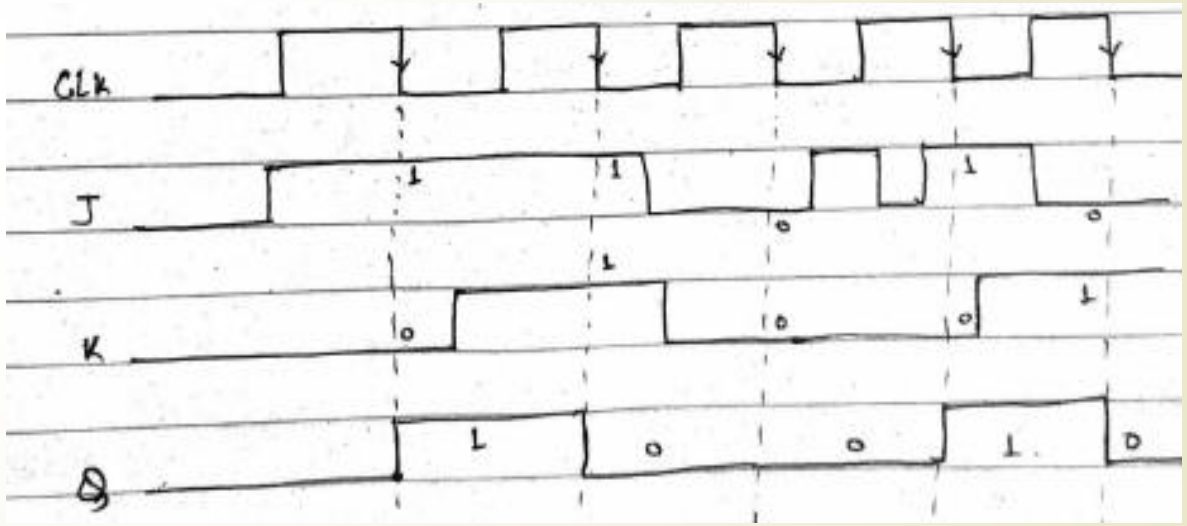


$$Q(t+1) = JQ(t)' + K'Q(t)$$

K-MAP and Boolean function

JK flip-flop			
Q(t)	Q(t+1)	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

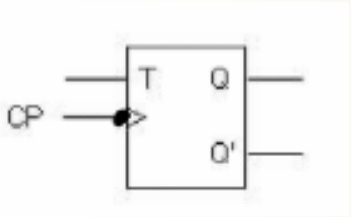
Excitation Table



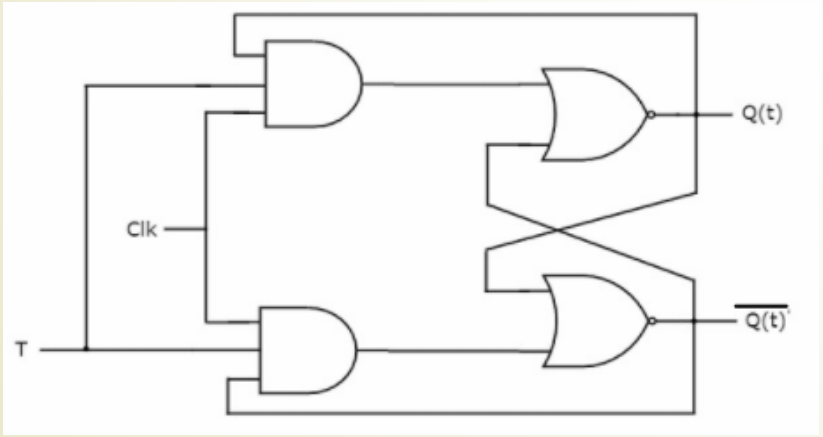
Timing diagram

FLIP-FLOP – T FLIP-FLOP

- The T flipflop is a single input version of the JK flipflop.
- The T flipflop is obtained from JK type if both inputs are tied together.
- The designation T comes from the ability of the flipflop to “Toggle”, or change state. Regardless of the present state of the flipflop, it assumes the complement state when the clock pulse occurs while input T is in logic 1.



Graphical Symbol



Logic diagram

Clock	T	Q(t)	Q(t+1)	Comment
↓	0	0	0	Hold
↓	0	1	1	Hold
↓	1	0	1	Complement
↓	1	1	0	Complement

Characteristic Table

FLIP-FLOP – T FLIP-FLOP

From the above characteristic table, we can directly write the **next state equation** as

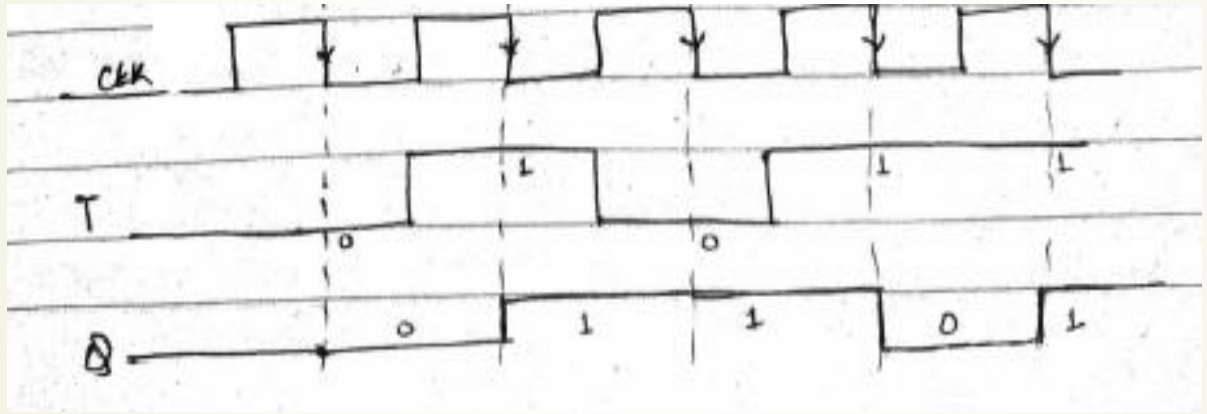
$$Q(t+1) = T'Q(t) + TQ(t)'$$

$$\Rightarrow Q(t+1) = T \oplus Q(t)$$

Boolean function

T flip-flop		
Q(t)	Q(t+1)	T
0	0	0
0	1	1
1	0	1
1	1	0

Excitation Table



Timing diagram

MASTER – SLAVE JK FLIPFLOP

Race Around Condition In JK Flip-flop

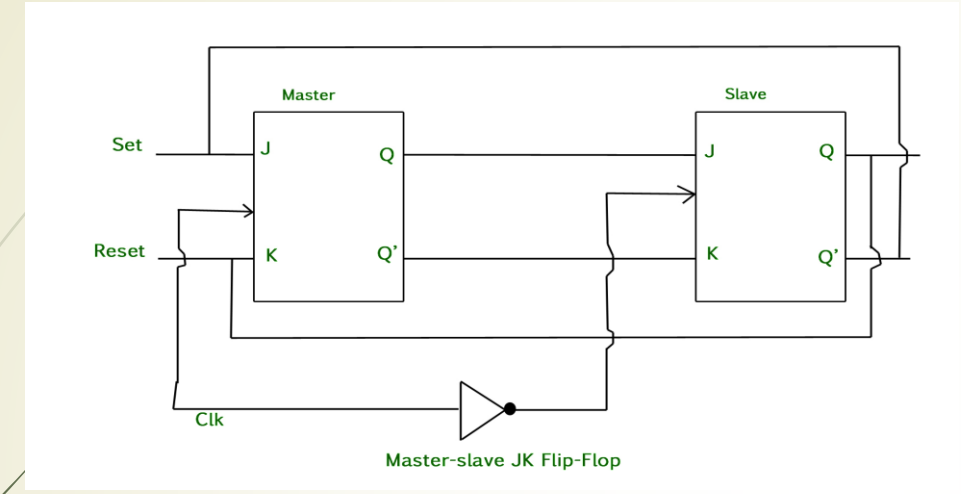
- For J-K flip-flop, if J=K=1, and if clk=1 for a long period of time, then Q output will toggle as long as CLK is high, which makes the output of the flip-flop unstable or uncertain.
- This problem is called race around condition in J-K flip-flop.
- This problem (Race Around Condition) can be avoided by ensuring that the clock input is at logic “1” only for a very short time.
- To eliminate race around condition:
  - Use edge- triggered flip-flop
  - Use **Master Slave JK flip flop**.

MASTER – SLAVE JK FLIPFLOP

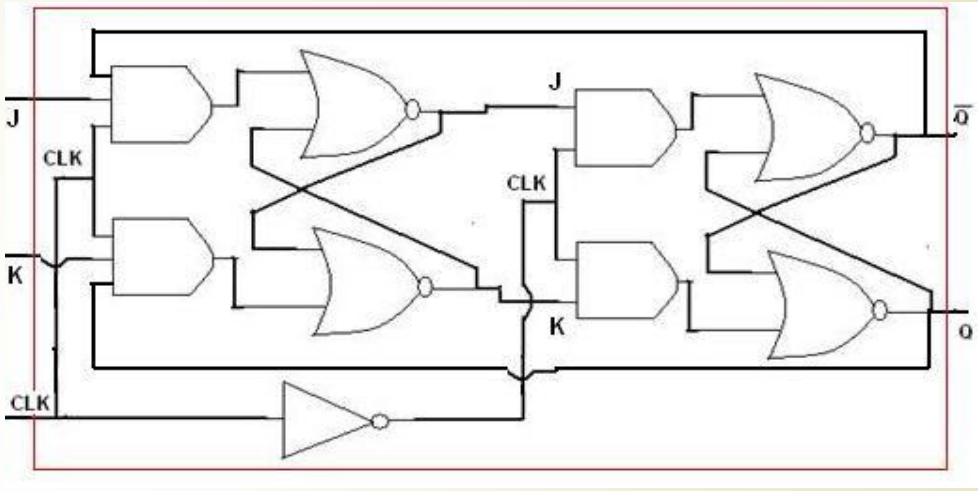
Master Slave JK flip flop

- The Master-Slave Flip-Flop is basically a combination of two JK flip-flops connected together in a series configuration.
- Out of these, one acts as the “master” and the other as a “slave”.
- The output from the master flip flop is connected to the two inputs of the slave flip flop whose output is fed back to inputs of the master flip flop.
- In addition to these two flip-flops, the circuit also includes an inverter.
- The inverter is connected to clock pulse in such a way that the inverted clock pulse is given to the slave flip-flop.
- In other words if CP=0 for a master flip-flop, then CP=1 for a slave flip-flop and if CP=1 for master flip flop then it becomes 0 for slave flip flop.

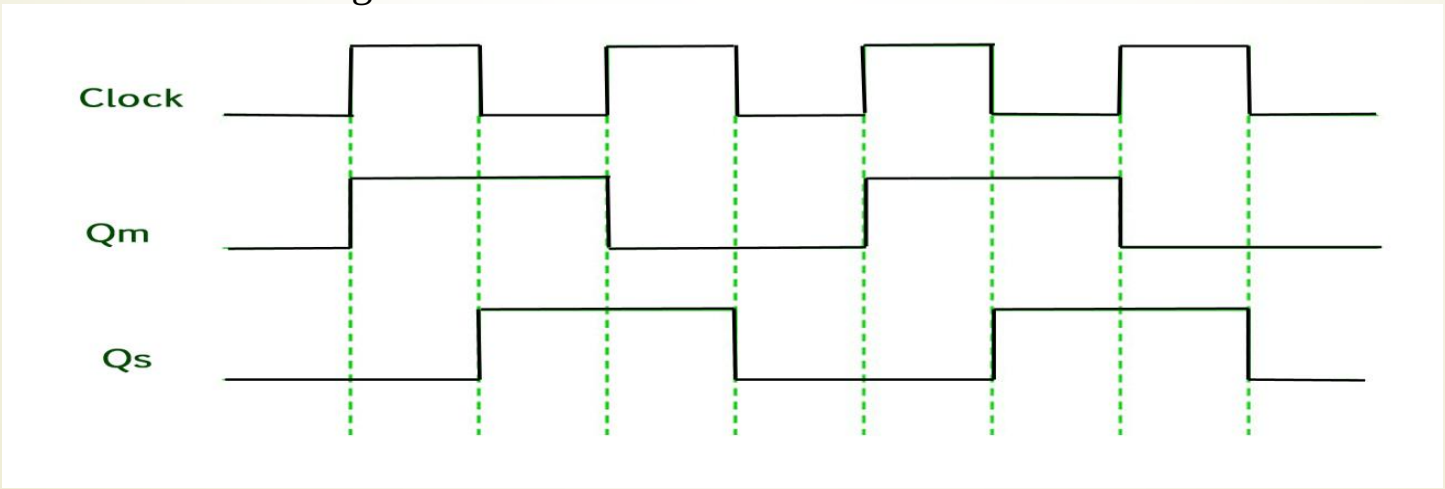
MASTER – SLAVE JK FLIPFLOP



Block diagram



Logic diagram



Timing diagram

COUNTERS

- Counter is a sequential circuit.
- A digital circuit which is used for a counting pulses is known counter.
- Counter is the widest application of flip-flops. It is a group of flip-flops with a clock signal applied.
- Counters are of two types.
- **Asynchronous or ripple counters.**
  - In asynchronous counter we don't use universal clock, only first flip flop is driven by main clock and the clock input of rest of the following flip flop is driven by output of previous flip flops
- **Synchronous counters.**
  - Unlike the asynchronous counter, synchronous counter has one global clock which drives each flip flop so output changes in parallel.
  - The one advantage of synchronous counter over asynchronous counter is, it can operate on higher frequency than asynchronous counter as it does not have cumulative delay because of same clock is given to each flip flop

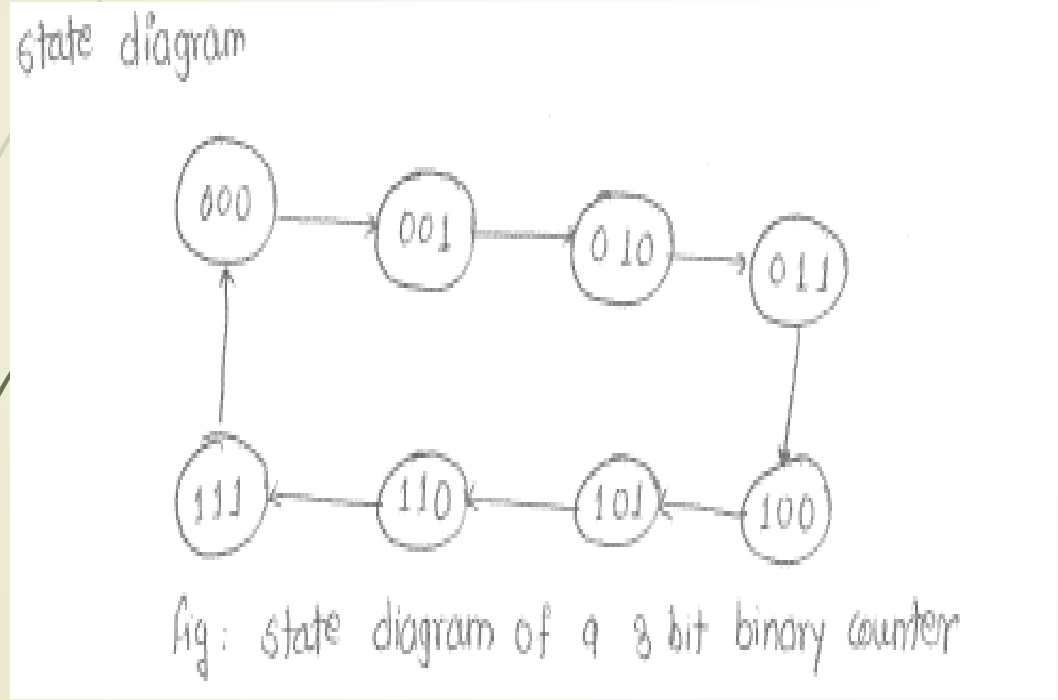
SYNCHRONOUS VS ASYNCHRONOUS COUNTERS

Sr. No.	Parameter	Asynchronous counter	Synchronous counter
1.	Circuit complexity	Logic circuit is simple.	With increase in number of states, the logic circuit becomes complicated.
2.	Connection pattern	Output of the preceding FF, is connected to clock of the next FF.	There is no connection between output of preceding FF and CLK of next one.
3.	Clock input	All the FFs are not clocked simultaneously.	All FFs receive clock signal simultaneously.
4.	Propagation delay	P.D. = n * (td) where n is number of FF and td is p.d. per FF.	P.D. = n * (td)FF + (td)gate. It is much shorter than that of asynchronous counter.
5.	Maximum frequency of operation	Low because of the long propagation delay.	High due to shorter propagation delay.

ASYNCHRONOUS COUNTERS

BINARY UP COUNTER

- An ‘N’ bit Asynchronous binary up counter consists of ‘N’ T flip-flops. It counts from 0 to $2^N - 1$.
- **EXAMPLE: 3-BIT BINARY UP COUNTER**



State diagram

No of negative edge of Clock	Q_2 MSB	Q_1	Q_0 LSB
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Count Sequence

REGISTERS

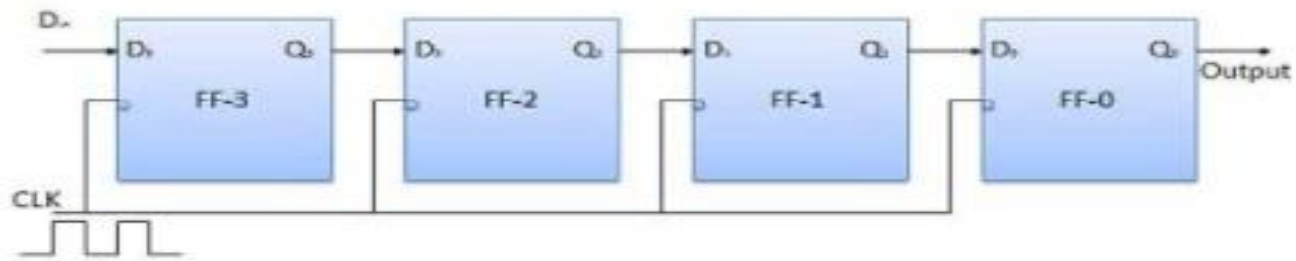
- Flip-flop is a 1 bit memory cell which can be used for storing the digital data.
- To increase the storage capacity in terms of number of bits, we have to use a group of flip-flop.
- Such a group of flip-flop is known as a Register.
- The n-bit register will consist of n number of flip-flop and it is capable of storing an n-bit word.
- The binary data in a register can be moved within the register from one flip-flop to another.
- The registers that allow such data transfers are called as shift registers. There are four mode of operations of a shift register.
  - Serial Input Serial Output
  - Serial Input Parallel Output
  - Parallel Input Serial Output
  - Parallel Input Parallel Output

REGISTERS

101 SERIAL INPUT SERIAL OUTPUT

- Let all the flip-flop be initially in the reset condition i.e. Q3 = Q2 = Q1 = Q0 = 0. If an entry of a four bit binary number 1 1 1 1 is made into the register, this number should be applied to Din bit with the LSB bit applied first. The D input of FF-3 i.e. D3 is connected to serial data input Din. Output of FF-3 i.e. Q3 is connected to the input of the next flip-flop i.e. D2 and so on.
- Also known as, shift right register.

Block Diagram



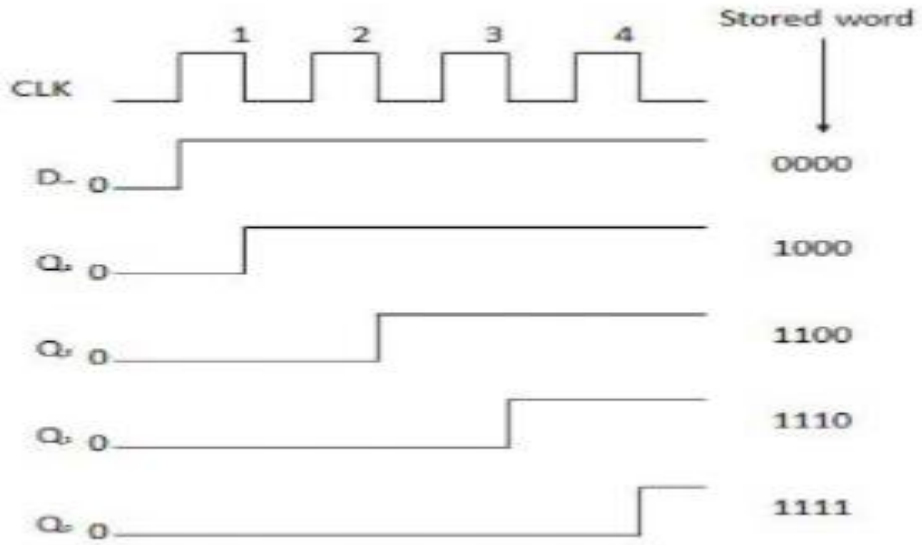
REGISTERS

Truth Table

	CLK	$D_0 = Q_3$	$Q_3 = D_2$	$Q_2 = D_1$	$Q_1 = D_0$	Q_0
Initially			0	0	0	0
(i)	↓	1	1	0	0	0
(ii)	↓	1	1	1	0	0
(iii)	↓	1	1	1	1	0
(iv)	↓	1	1	1	1	1

→ Direction of data travel

Waveforms

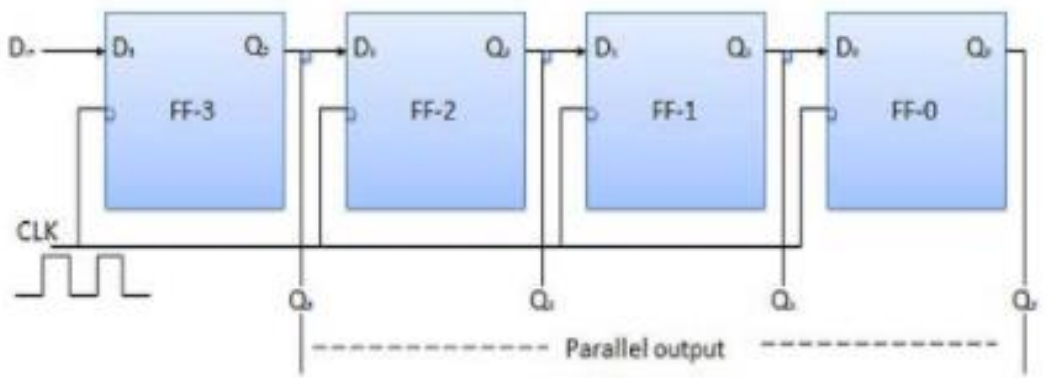


REGISTERS

SERIAL INPUT PARALLEL OUTPUT

- In such types of operations, the data is entered serially and taken out in parallel fashion.
- Data is loaded bit by bit. The outputs are disabled as long as the data is loading.
- As soon as the data loading gets completed, all the flip-flops contain their required data, the outputs are enabled so that all the loaded data is made available over all the output lines at the same time.
- 4 clock cycles are required to load a four bit word. Hence the speed of operation of SIPO mode is same as that of SISO mode.

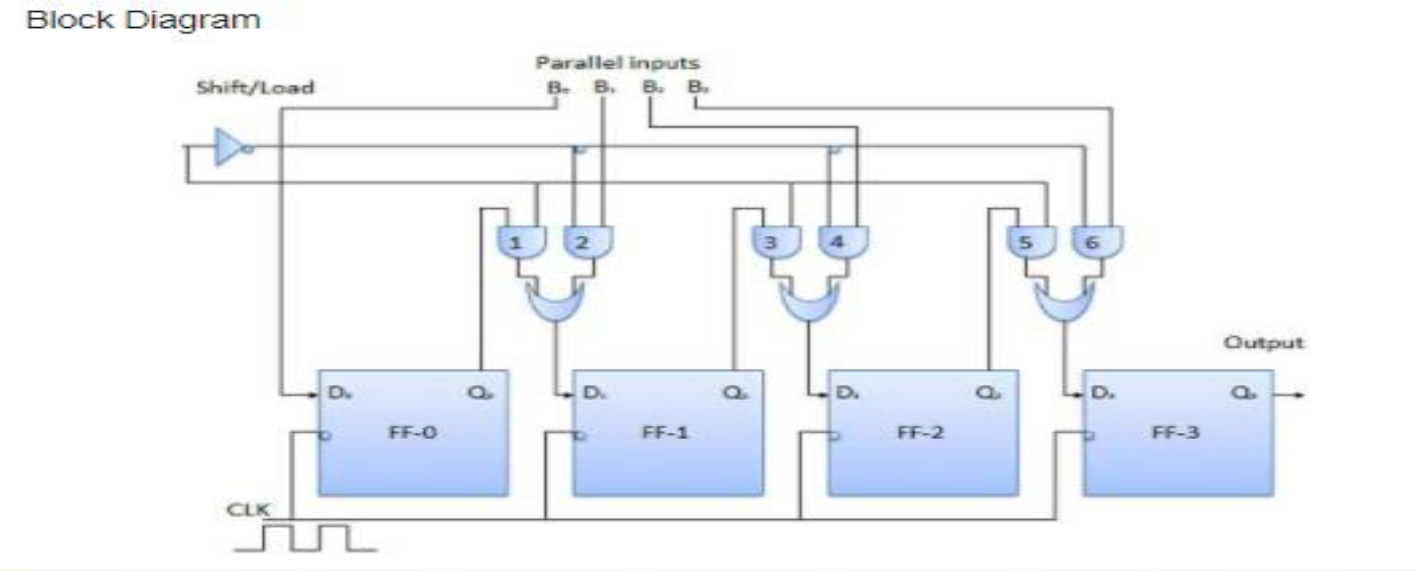
Block Diagram



REGISTERS

PARALLEL INPUT SERIAL OUTPUT (PISO)

- Data bits are entered in parallel fashion.
- The circuit shown below is a four bit parallel input serial output register.
- Output of previous Flip Flop is connected to the input of the next one via a combinational circuit.
- The binary input word B0, B1, B2, B3 is applied though the same combinational circuit.
- There are two modes in which this circuit can work namely - shift mode or load mode.



REGISTERS

Load mode

- When the shift/load bar line is low (0), the AND gate 2, 4 and 6 become active they will pass B1, B2, B3 bits to the corresponding flip-flops. On the low going edge of clock, the binary input B0, B1, B2, B3 will get loaded into the corresponding flip-flops. Thus parallel loading takes place.

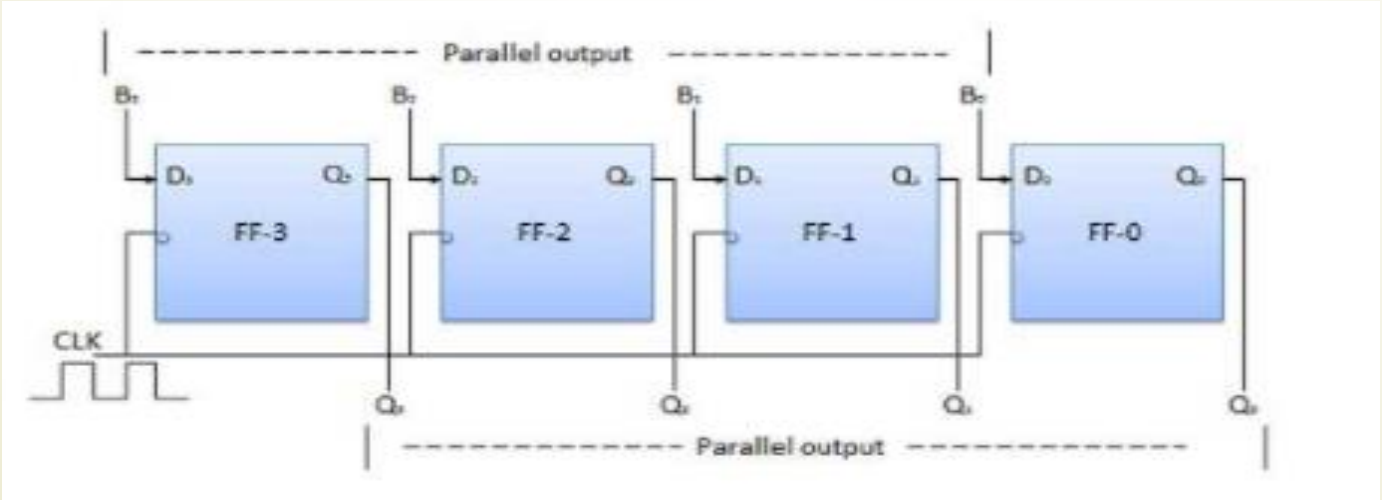
Shift mode

- When the shift/load bar line is high (1), the AND gate 2, 4 and 6 become inactive. Hence the parallel loading of the data becomes impossible. But the AND gate 1,3 and 5 become active. Therefore the shifting of data from left to right bit by bit on application of clock pulses. Thus the parallel in serial out operation takes place.

REGISTERS

PARALLEL INPUT PARALLEL OUTPUT (PIPO)

- In this mode, the 4 bit binary input B0, B1, B2, B3 is applied to the data inputs D0, D1, D2, D3 respectively of the four flip-flops.
- As soon as a negative clock edge is applied, the input binary bits will be loaded into the flip-flops simultaneously. The loaded bits will appear simultaneously to the output side. Only clock pulse is essential to load all the bits.



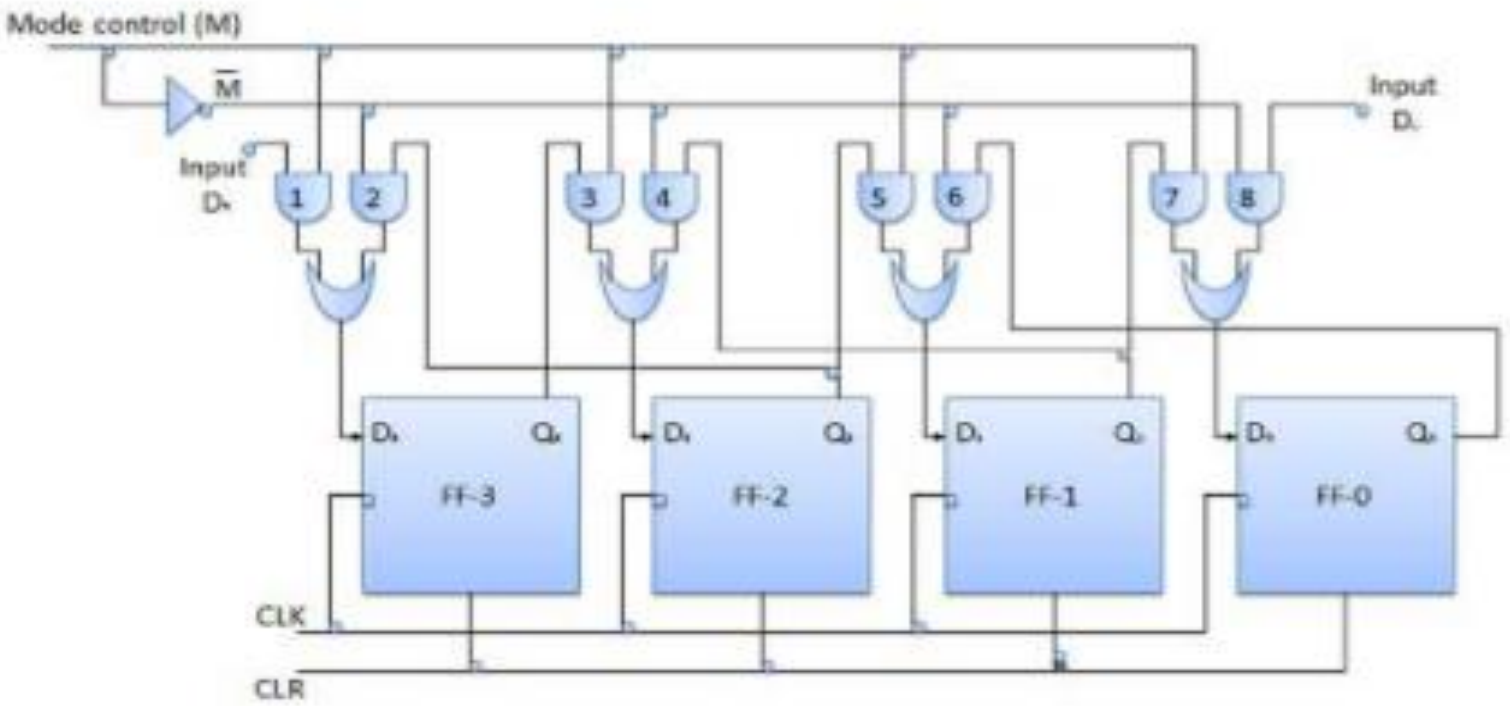
SHIFT REGISTERS

Bidirectional Shift Register

- If a binary number is shifted left by one position then it is equivalent to multiplying the original number by 2. Similarly if a binary number is shifted right by one position then it is equivalent to dividing the original number by 2.
- Hence if we want to use the shift register to multiply and divide the given binary number, then we should be able to move the data in either left or right direction.
- Such a register is called bi-directional register. A four bit bi-directional shift register is shown in fig.
- There are two serial inputs namely the serial right shift data input DR, and the serial left shift data input DL along with a mode select input (M).

SHIFT REGISTERS

Block Diagram



SHIFT REGISTERS

With M = 1 – Shift right operation

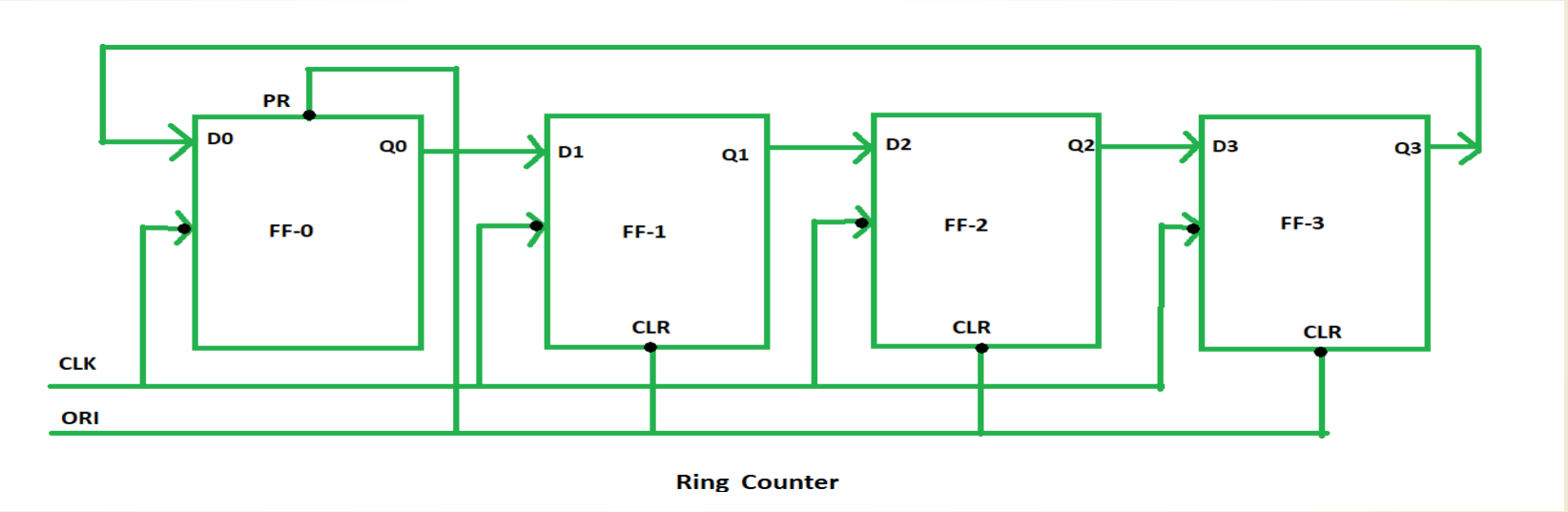
- If M = 1, then the AND gates 1, 3, 5 and 7 are enabled whereas the remaining AND gates 2, 4, 6 and 8 will be disabled.
- The data at DR is shifted to right bit by bit from FF-3 to FF-0 on the application of clock pulses. Thus with M = 1 we get the serial right shift operation

With M = 0 – Shift left operation

- When the mode control M is connected to 0 then the AND gates 2, 4, 6 and 8 are enabled while 1, 3, 5 and 7 are disabled.
- The data at DL is shifted left bit by bit from FF-0 to FF-3 on the application of clock pulses. Thus with M = 0 we get the serial right shift operation.

RING COUNTERS

- Ring counter is a typical application of Shift resister.
- Ring counter is almost same as the shift counter.
- The only change is that the output of the last flip-flop is connected to the input of the first flip-flop in case of ring counter but in case of shift resister it is taken as output. Except this all the other things are same.

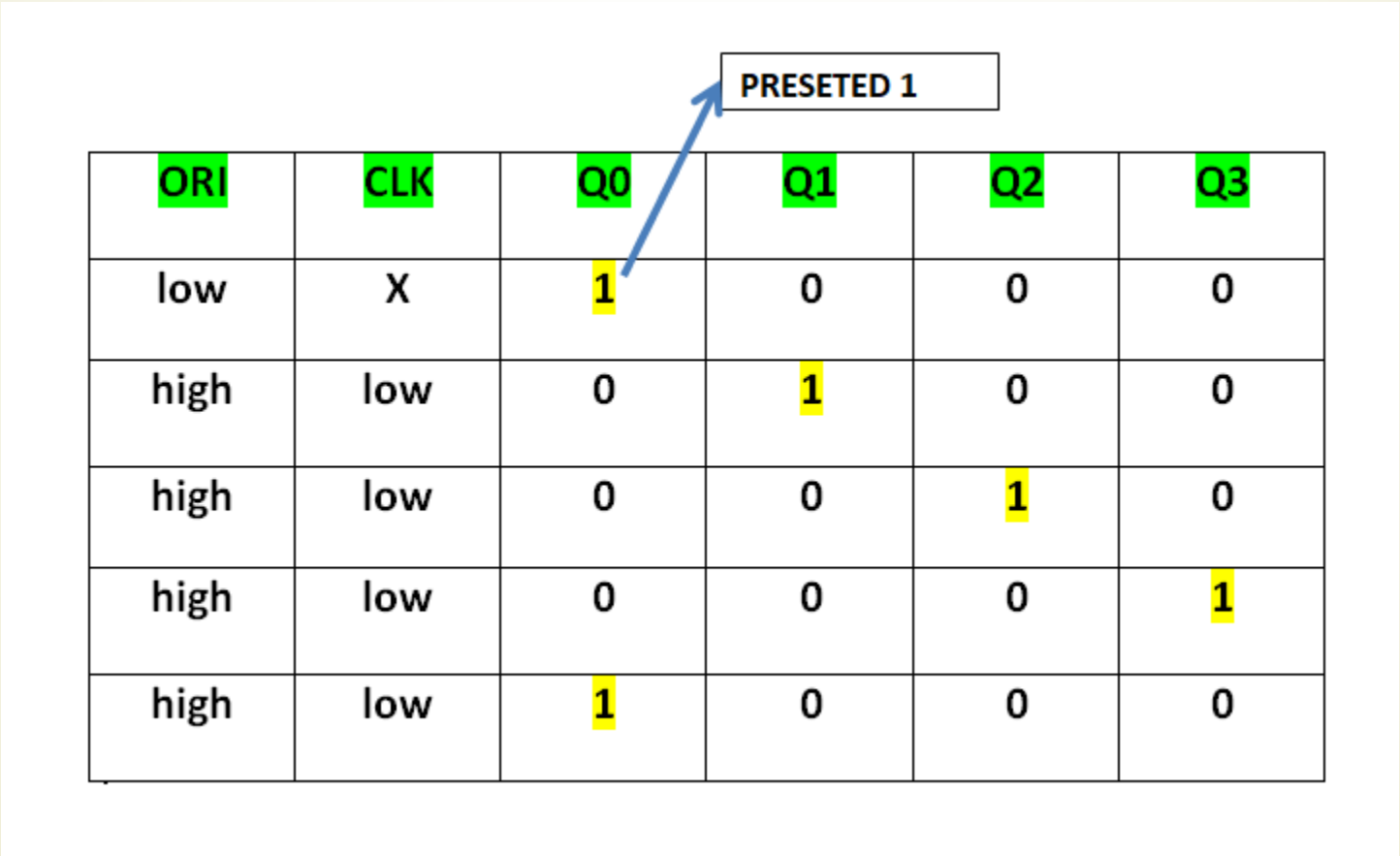


Ring Counter

- In this diagram, we can see that the clock pulse (CLK) is applied to all the flip-flop simultaneously. Therefore, it is a Synchronous Counter.

RING COUNTERS

- Also, here we use Overriding input (ORI) to each flip-flop. Preset (PR) and Clear (CLR) are used as ORI.
- When PR is 0, then the output is 1. And when CLR is 0, then the output is 0. Both PR and CLR are active low signal that is always works in value 0.



ORI	CLK	Q0	Q1	Q2	Q3
low	X	1	0	0	0
high	low	0	1	0	0
high	low	0	0	1	0
high	low	0	0	0	1
high	low	1	0	0	0

RING COUNTERS

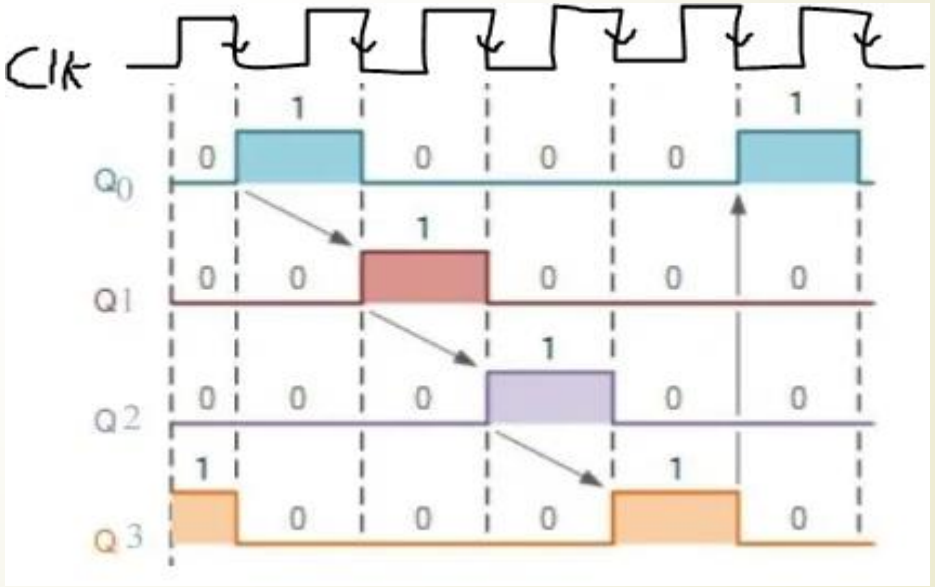
- This Preseted 1 is generated by making ORI low and that time Clock (CLK) becomes don't care.
- After that ORI made to high and apply low clock pulse signal as the Clock (CLK) is negative edge triggered.
- After that, at each clock pulse the preseted 1 is shifted to the next flip-flop and thus form Ring.
- From the above table, we can say that there are 4 states in 4-bit Ring Counter.
- 4 states are:

  1 0 0 0

  0 1 0 0

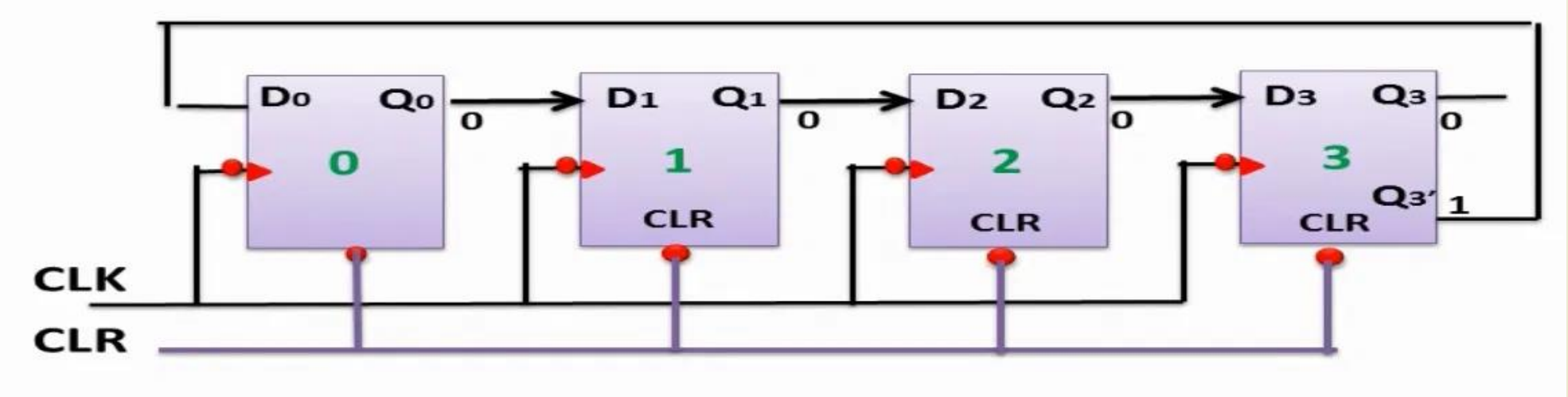
  0 0 1 0

  0 0 0 1



JOHNSON COUNTERS

- In the ring counter we given the output of the last flip flop into the input of the first flip but in the Johnson counter the last flip flop complemented output is given to the input of the first flip flop.
- In Johnson counter the number of states is equal to twice the number of flip flops.
- So if we use 4 flip flops we will have 8 states so the number of the states are double.
- We applied clock simultaneously to all flip flops.
- The clear input is applied to all the flip flops.



JOHNSON COUNTERS

- The output of the first flip flop which is Q0 is given at the input of the second flip flop D1 and the output of the second flip flop which is Q2 is given to input of the third flip flop which is D2 and the complemented output (Q3') will be given to the input of the first flip D0.
- The difference between the ring counter and Johnson counter is that it does not require pre-set.

Clear/ Pre-set	Clock	Q0	Q1	Q2	Q3
0	No clock	0	0	0	0
1	↓	1	0	0	0
1	↓	1	1	0	0
1	↓	1	1	1	0
1	↓	1	1	1	1
1	↓	0	1	1	1
1	↓	0	0	1	1
1	↓	0	0	0	1
1	↓	0	0	0	0

JOHNSON COUNTERS