DIGITAL ELECTRONICS

Q. Find the simplified expression A’BC’+AC’.

a) B

b) A+C

c) (A+B)C’

d) B’C

TEACHERS ADDA BY TARGET ABHI 7877719287

Q. Evaluate the expression: (X + Z)(X + XZ') + XY + Y.

a) XY+Z'

b) Y+XZ'+Y'Z

c) X'Z+Y

d) X+Y

TEACHERS ADDA BY TARGET ABHI 7877719287

Q. Convert the Z= Σm (m0,m1,m5,m7) equation to POS.

a) Z= π(M2,M3,M5,M6)

b) Z= π(M1,M3,M4,M5)

c) Z= π(M0,M2,M1,M5)

d) Z= π(M2,M3,M4,M6)

TEACHERS ADDA BY TARGET ABHI 7877719287

Q. How many combination input in 2 bit Half adder_____

a) 3 combinational inputs

b) 4 combinational inputs

c) 6 combinational inputs

d) 8 combinational inputs

Q. Define Demultiplexer.

a) Many input and one output

b) one input and many output

c) one input and one output

d) Many input and many output

TEACHERS ADDA BY TARGET ABHI 7877719287

Q. How many select lines would be needed in 16 to 1 multiplexer?

a) 2

b) 4

c) 8

d) 3

TEACHERS ADDA BY TARGET ABHI 7877719287

Q. The De multiplexer is otherwise called as

a) Input selector

b) Output selector

c) Data distributor

d) Output distributor

6. How many AND gates are required for a multiplexer 4:1?

a) 2

b) 6

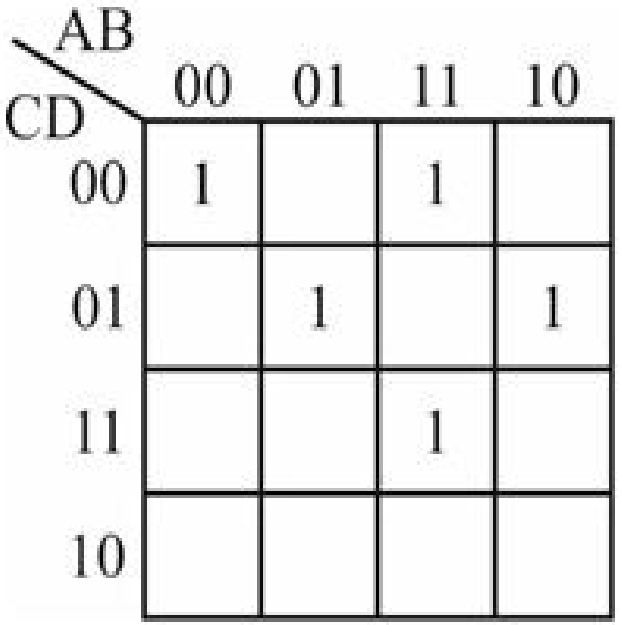
c) 8

d) 4

TEACHERS ADDA BY TARGET ABHI 7877719287

Which of the logic equation is the represented canonical sum of product form for the given K-Map?

1. $\overline{ABCD} + \bar{A}B\bar{C}D + \overline{AB}CD + ABC\bar{D} + A\overline{BC}D$
2. $\overline{ABCD} + \bar{A}B\bar{C}D + AB\overline{CD} + ABCD + A\overline{BC}D$
3. $\overline{ABCD} + AB\overline{CD} + AB\overline{CD} + A\overline{BC}D$
4. $\overline{ABCD} + AB\overline{CD} + \overline{AB}CD + ABCD + A\overline{BC}D$



TEACHERS ADDA BY TARGET ABHI 7877719287

Which of the following gates is also called 'all or nothing gate’?

A. AND
B. OR
C. NOR
D. XOR

TEACHERS ADDA BY TARGET ABHI 7877719287

How many 32 K × 1 RAM chips are needed to provide a memory capacity of 256 K-bytes ?

1. 8
2. 128
3. 32
4. 64

TEACHERS ADDA BY TARGET ABHI 7877719287

Gray code representation of $(27)_{10}$ is _____.

1. 11010
2. 11100
3. 11011
4. 10110

TEACHERS ADDA BY TARGET ABHI 7877719287

What is the relation between N select inputs and M inputs lines in a multiplexer?

1. $2N = M$
2. $2^M = N$
3. $2^N = M$
4. $2M = N$

TEACHERS ADDA BY TARGET ABHI 7877719287

How many rows are there in a truth table having 'n' variables?

1. n^2
2. $\log_2 n$
3. $2n$
4. 2^n

TEACHERS ADDA BY TARGET ABHI 7877719287

Solve the following Boolean algebra expression :

A(A' + C)(A'B + C)(A'BC + C’)

A. AC

B. BC

C. 0

D. AB

TEACHERS ADDA BY TARGET ABHI 7877719287

Combinational & Sequential Logic Circuits

TEACHERS ADDA BY TARGET ABHI 7877719287

INTRODUCTION

- In digital **circuit** theory, **sequential logic** is a type of **logic circuit** whose output depends not only on the present value of its input signals but on the sequence of past inputs, the input history as well.

- This is in contrast to **combinational logic**, whose output is a function of only the present input.

TEACHERS ADDA BY TARGET ABHI 7877719287

- **Difference between combinational and sequential circuit.**

- **Sequential circuits** are those which are dependent on clock cycles and depends on present as well as past inputs to generate any output.
- **Combinational Circuit –**
- In this output depends only upon present input.
- The sequential logic has memory while combinational logic does not.
- They employ a feedback loop to give output back to input. **Sequential** logic **circuits** is a form of binary **circuit**; its design employs one or more inputs and one or more outputs.

TEACHERS ADDA BY TARGET ABHI 7877719287

- A **combinational logic circuit** performs an operation assigned logically by a **Boolean** expression or truth table. Examples of common **combinational logic circuits** include: half adders, full adders, **multiplexers**, **demultiplexers**, encoders and decoders **.**

TEACHERS ADDA BY TARGET ABHI 7877719287

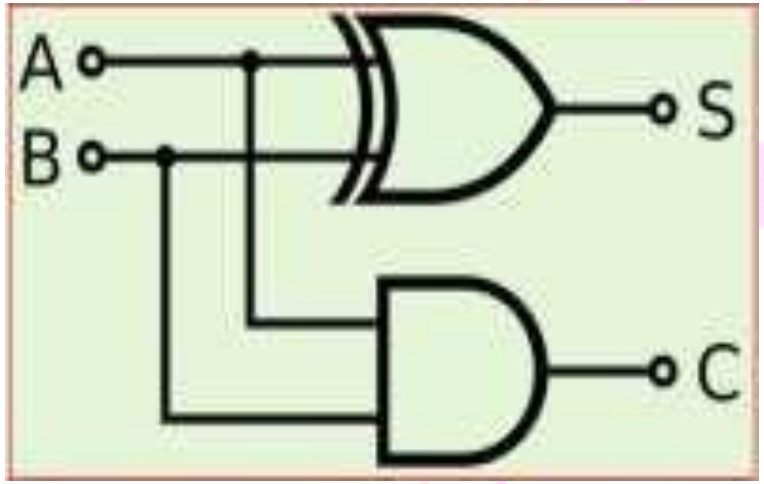
- A **Sequential** logic **circuits** is a form of binary **circuit**; its design employs one or more inputs and one or more outputs, whose states are related to some definite rules that depends on previous states. ... **Examples** of such **circuits** include clocks, flip-flops, bi-stables, counters, memories, and registers.
- There are two **types of sequential circuit**, synchronous and asynchronous. Synchronous **types** use pulsed or level inputs and a clock input to drive the **circuit**(with restrictions on pulse width and **circuit** propagation). Asynchronous **sequential circuits** do not use a clock signal as synchronous **circuits** do.
- **Flip flop** is a **sequential circuit** which generally samples its inputs and changes its outputs only at particular instants of time and not continuously. **Flip flop** is said to be edge sensitive or edge triggered rather than being level triggered like latches.

Half adder

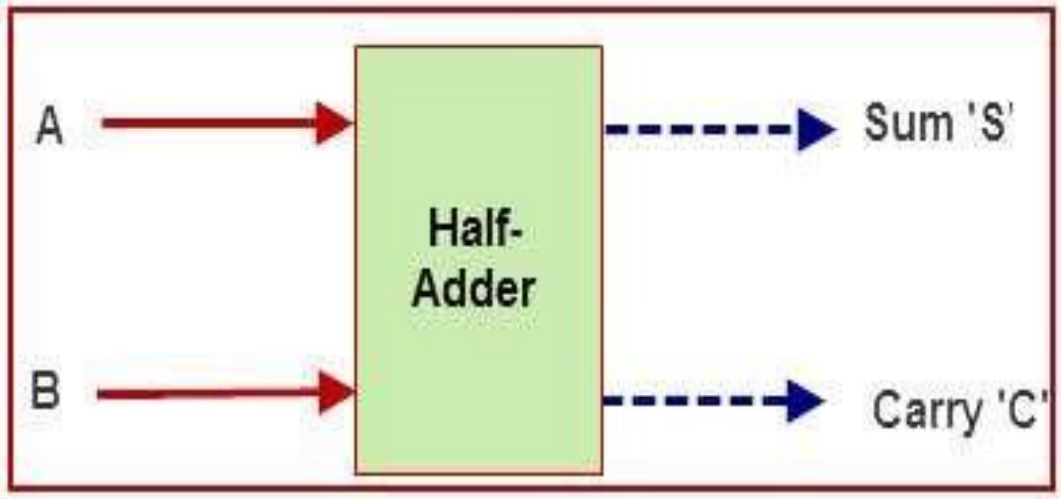
- An **adder** is a digital circuit that performs addition of numbers. Half adder has only two inputs and two outputs. The **half adder** adds **two binary digits** called as augend and addend and produces two outputs as **sum and carry**; XOR is applied to both inputs to produce sum and AND gate is applied to both inputs to produce carry.
- By using half adder, you can design simple addition with the help of logic gates.

TEACHERS ADDA BY TARGET ABHI 7877719287

Half adder



Circuit Implementation



Block Diagram

Truth Table

INPUTS		OUTPUTS	
A	B	SUM	CARRY
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

TEACHERS ADDA BY TARGET ABHI 7877719287

Why it is called as Half-adder ?

- The **half adder** can add only two input bits (A and B) and has nothing to do with the carry if there is any in the input. So if the input to a **half adder** have a carry, then it will neglect it and adds only the A and B bits. That means the binary addition process is not complete and that's why it is **called a half adder**

How Half Adder works ?

- **Half adder** is a simple combinational circuit used to add two single bits. It accepts two inputs and produce two outputs that is a sum output and a carry output. A **half adder** consists of two logic gates 1) XOR and 2) AND gate. And the carry operation performed by AND gate thus carry out put **will** be A+B.

FULL ADDER

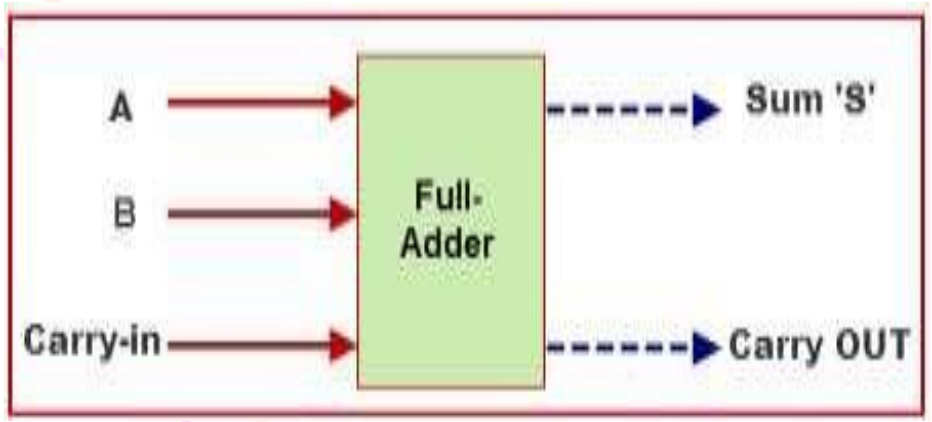
The full adder adds 3 one bit numbers, where two can be referred to as operands and one can be referred to as bit carried in. It produces 2-bit output and these can be referred to as output carry and sum.

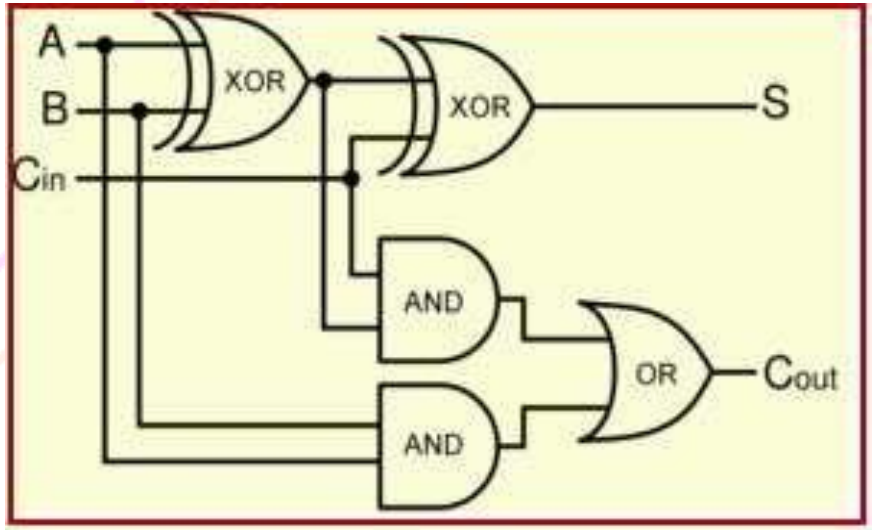
The full-adder has three inputs and two outputs. The first two inputs are A and B and the third input is an input carry as C-IN. When a full-adder logic is designed, you string eight of them together to create a byte-wide adder and cascade the carry bit from one adder to the next.

TEACHERS ADDA BY TARGET ABHI 7877719287

FULL ADDER

INPUTS			OUTPUT	
A	B	C-IN	C-OUT	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1





Circuit Implementation

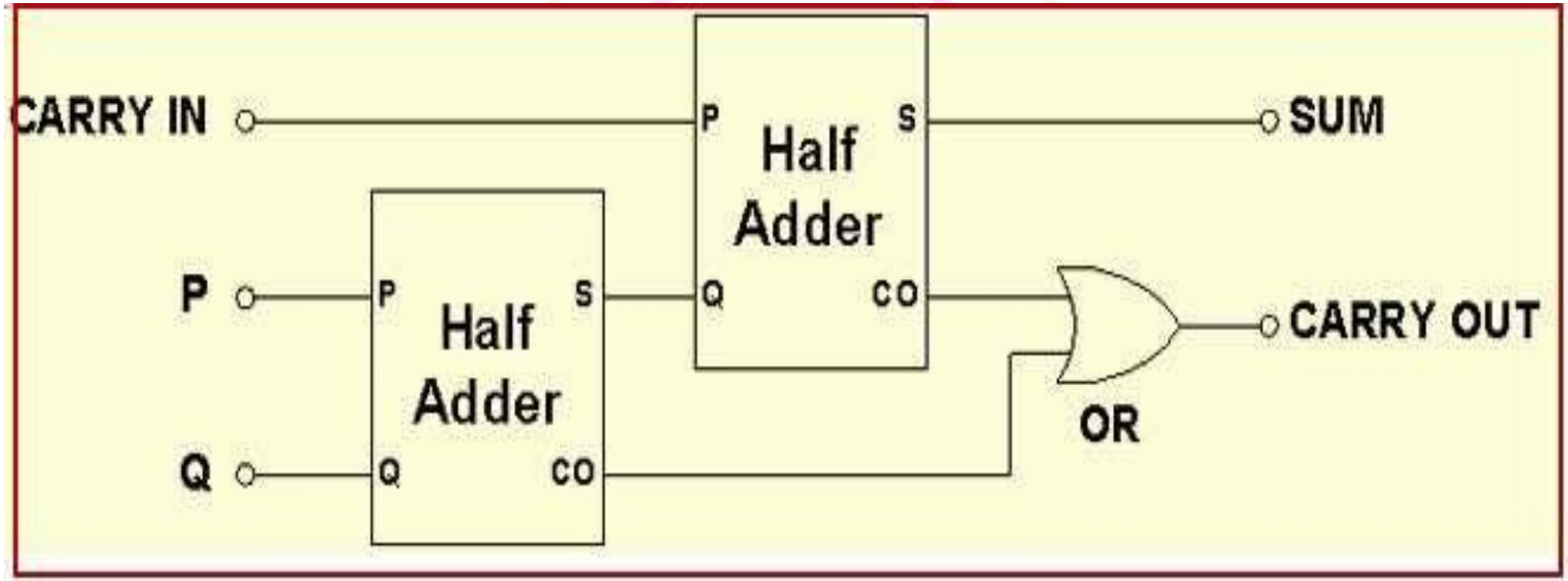
TEACHERS ADDA BY TARGET ABHI 7877719287

•With the truth-table, the full adder logic can be implemented. You can see that the output S is an XOR between the input A and the half-adder, SUM output with B and C-IN inputs. We take C-OUT will only be true if any of the two inputs out of the three are HIGH.

•So, we can implement a full adder circuit with the help of two half adder circuits. At first, half adder will be used to add A and B to produce a partial Sum and a second half adder logic can be used to add C-IN to the Sum produced by the first half adder to get the final S output.

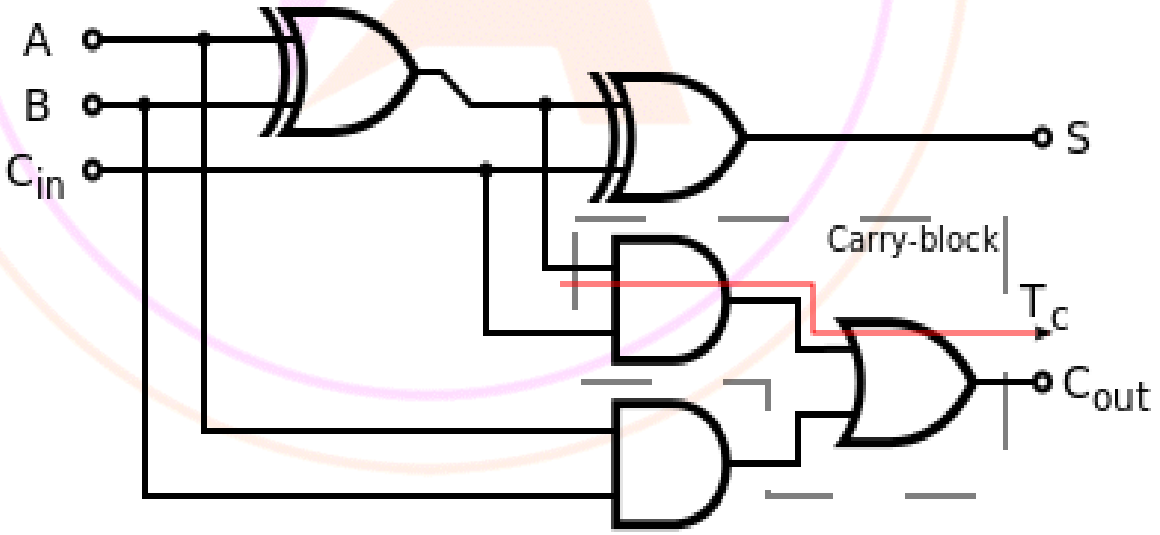
• If any of the half adder logic produces a carry, there will be an output carry. So, C-OUT will be an OR function of the half-adder Carry outputs.

•The implementation of larger logic diagrams is possible with the above full adder logic a simpler symbol is mostly used to represent the operation. Given below is a simpler schematic representation of a full adder.



TEACHERS ADDA BY TARGET ABHI 7877719287

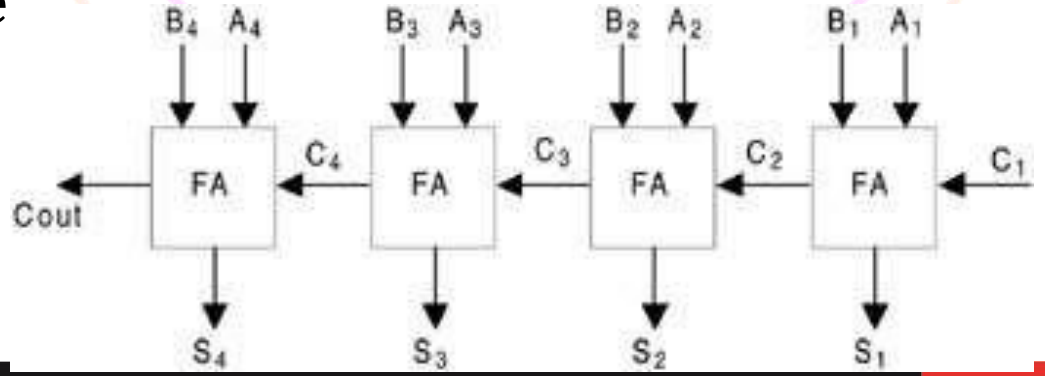
- A **full adder** is a digital **circuit** that performs addition. **Full adders** are implemented with logic gates in hardware.
  A **full adder** adds three one-bit binary numbers, two operands and a carry bit. The **adder** outputs two numbers, a sum and a carry bit.
- The **Boolean expression for a full adder** is as follows.
- For the CARRY-OUT (Cout) bit: CARRY-OUT = A AND B OR Cin(A XOR B) = A.B + Cin(A ⊕ B)



TEACHERS ADDA BY TARGET ABHI 7877719287

4-Bit Parallel Binary adder :

- A **binary parallel adder** is a digital function that produces the arithmetic sum of two **binary** numbers in **parallel.**
- **FOUR-BIT** BINARY **PARALLEL ADDER** is a circuit in which two binary numbers each of n **bits** can be added by means of a full **adder** circuit. Consider the example that two **4-bit** binary numbers $B_4B_3B_2B_1$ and $A_4A_3A_2A_1$ are to be added with a carry input C_1.
- A group of four bits is called a nibble. A basic 4-bit parallel adder is implemented with four full-adder stages as shown in Figure



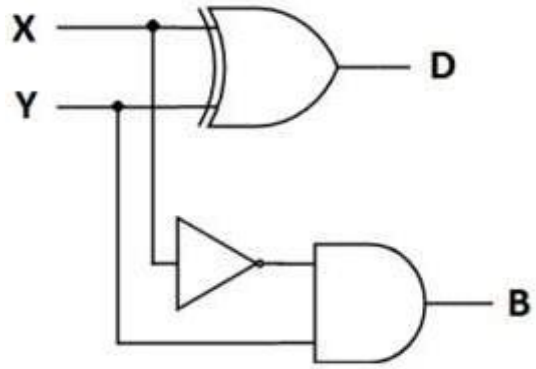
TEACHERS ADDA BY TARGET ABHI 7877719287

Half Substractor :

- A half subtractor is an arithmetic circuit that subtracts two bits and produces their difference. The circuit has two inputs minuend (X) and subtrahend (Y) and two output bits, one is the difference bit (D) and the other is the borrow bit (B).
- As like addition operation of 2 binary digits, which produces SUM and CARRY, the subtraction of 2 binary digits also produces two outputs which are termed as **difference and borrow**. The simplest possible subtraction of 2-bit binary digits consists of four possible operations, they are **0-0, 0-1, 1-0** and **1-1**. The operations **0-0, 1-0** and **1-1** produces a subtraction of 1-bit output whereas, the remaining operation **0-1** produces a 2-bit output. They are referred as **difference** and **borrow** bit respectively. This **borrow** bit is used for subtraction of the next higher pair bit.
- So, we can define half subtractor as a combinational circuit which is capable of performing subtraction of 2-bit binary digits is known as a half subtractor. Here, the binary digit from which the other digit is subtracted is called **minuend** and the binary digit which is to be subtracted is known as the **subtrahend**.

- It performs the operation X – Y. It should be noted that the weight of the output borrow bit is -2, while the weight of the output difference bit is +1.
- The truth table of the half subtractor is shown. The Boolean functions for the two outputs can be obtained directly from the truth table as:

  $D = (XY + XY) = X \oplus Y$
- **The half subtractor boolean expressions are :**
- **$D = (X'Y + XY') = X \oplus Y$**
- **$B = X'Y$**

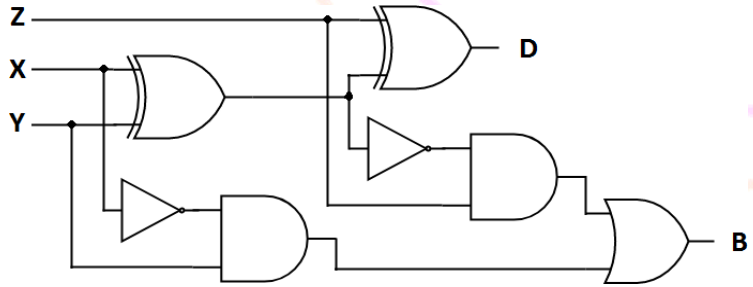


Inputs		Outputs	
X	Y	D	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

TEACHERS ADDA BY TARGET ABHI 7877719287

A full subtractor

- A full subtractor is a combinational circuit that performs a subtraction between two bits, taking into account that a 1 may have been borrowed by a lower significant bit. The circuit has three inputs and two outputs.
- Input variables are minuend (X), subtrahend (Y), and previous borrow (Z); output variables are difference (D) and output borrow (B).
- It performs the operation **X – Y – Z**. It should be noted that the weight of the output borrow bit is -2, while the weight of the output difference bit is +1. The truth table of the full subtractor is shown.
- **The full subtractor boolean expressions are :**
- **(X'Y'Z + X'YZ' + XY'Z' + XYZ) = X ⊕ Y ⊕ Z**
- **(X'Y'Z + X'YZ' + X'YZ + XYZ) = X'(Y ⊕ Z) + YZ**
-



Inputs			Outputs	
X	Y	Z	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

TEACHERS ADDA BY TARGET ABHI 7877719287

- When there is a situation where the minuend and subtrahend number contains more significant bit, then the **borrow** bit which is obtained from the subtraction of 2-bit binary digits is subtracted from the next higher order pair of bits. In such situation, the subtraction involves the operation of 3 bits. Such situation of subtraction can't handle by a simple half subtractor. So, combining two half subtractor we can form another combinational circuit which can perform this type of operation. This circuit is known as the full subtractor.
- So we can define full subtractor as a combinational circuit which takes three inputs and produces two outputs **difference** and **borrow**. Above is the truth table of the full subtractor, we have used three input variables X, Y and Z which refers to the term **minuend, subtrahend** and **borrow** bit respectively. The two outputs **difference** and **borrow** are named as D and B respectively.
- The construction of **full subtractor circuit diagram** involves two half subtractor joined by an OR gate as shown in the above **circuit diagram of the full subtractor**. The two borrow bits generated by two separate half subtractor are fed to the OR gate which produces the final borrow bit. The final difference bit is the combination of the difference output of the first half adder and the next higher order pair of bits.

TEACHERS ADDA BY TARGET ABHI 7877719287

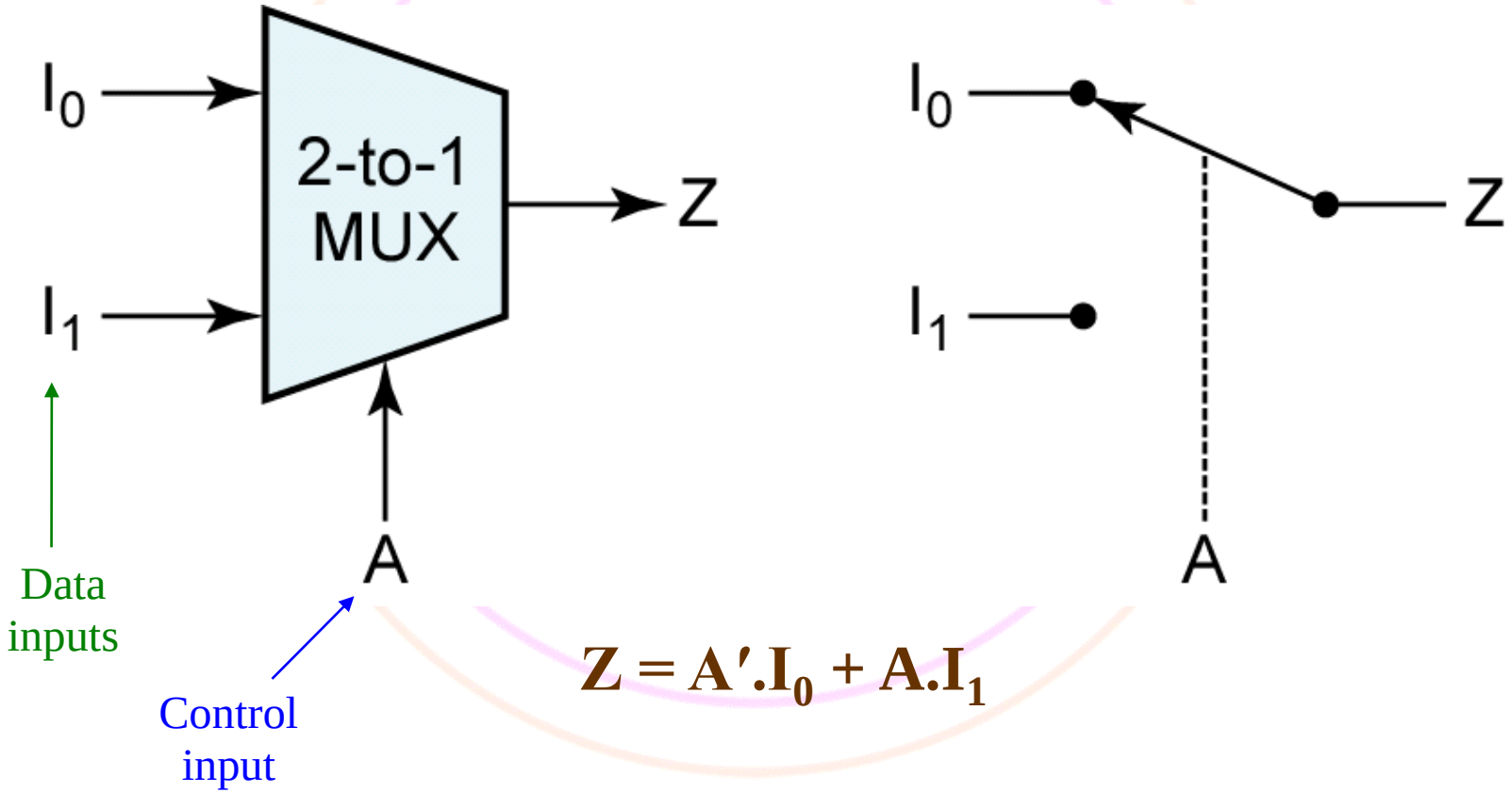
Multiplexers

Multiplexers

- A multiplexer has
  - N control inputs
  - 2^N data inputs
  - 1 output
- A multiplexer routes (or connects) the selected data input to the output.
  - The value of the control inputs determines the data input that is selected.

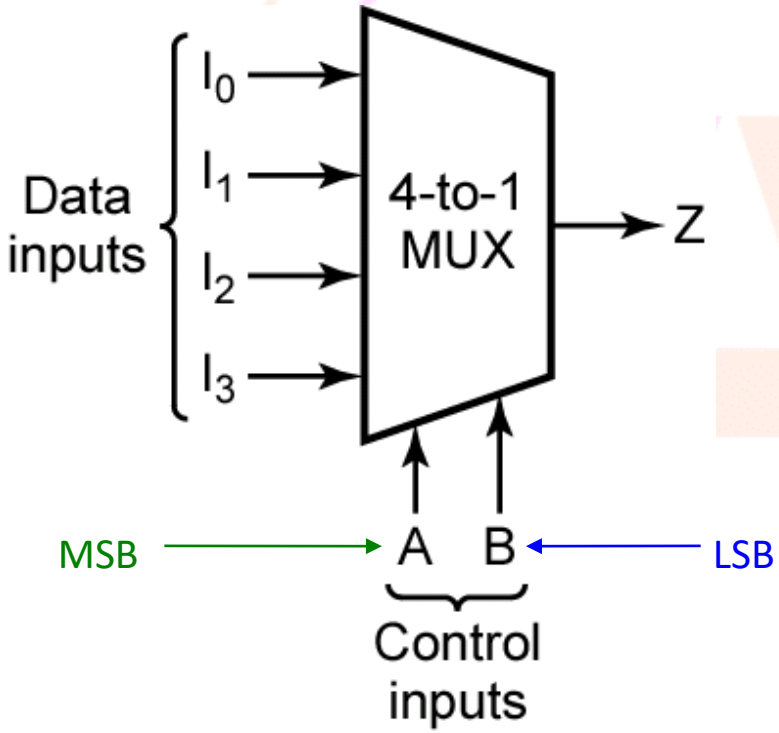
TEACHERS ADDA BY TARGET ABHI 7877719287

Multiplexers



$$Z = A'.I_0 + A.I_1$$

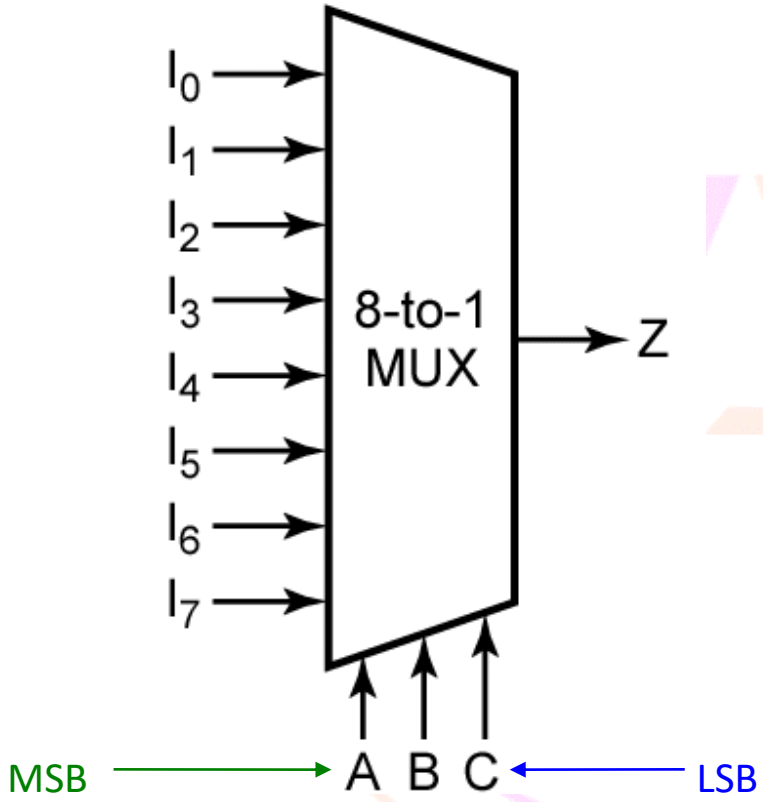
Multiplexers



A	B	F
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

$$Z = A'.B'.I_0 + A'.B.I_1 + A.B'.I_2 + A.B.I_3$$

Multiplexers

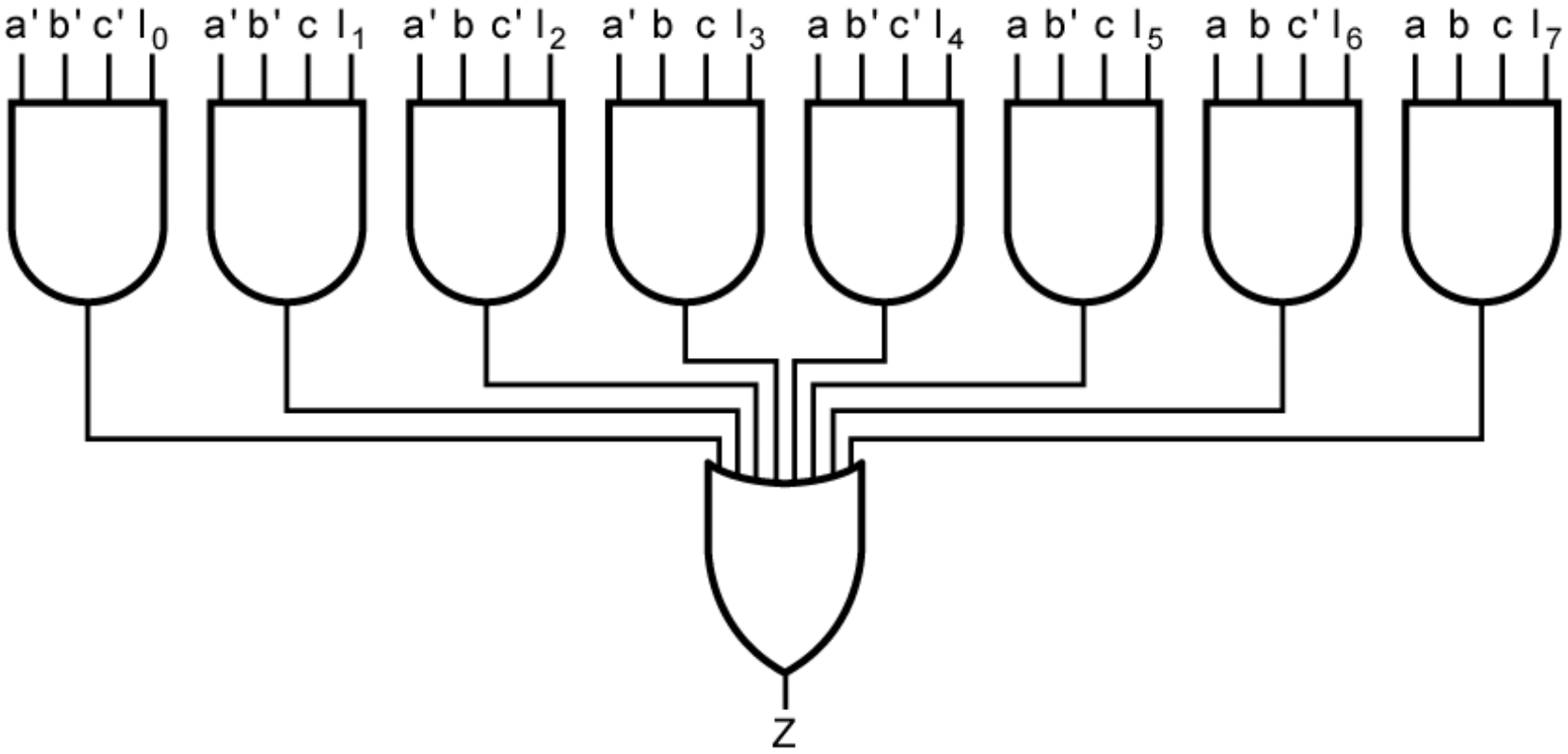


A	B	C	F
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7

$$Z = A'.B'.C'.I_0 + A'.B'.C.I_1 + A'.B.C'.I_2 + A'.B.C.I_3 + A.B'.C'.I_0 + A.B'.C.I_1 + A'.B.C'.I_2 + A.B.C.I_3$$

TEACHERS ADDA BY TARGET ABHI 7877719287

Multiplexers



TEACHERS ADDA BY TARGET ABHI 7877719287

Multiplexers

Design an 8-to-1 multiplexer using 4-to-1 and 2-to-1 multiplexers only.

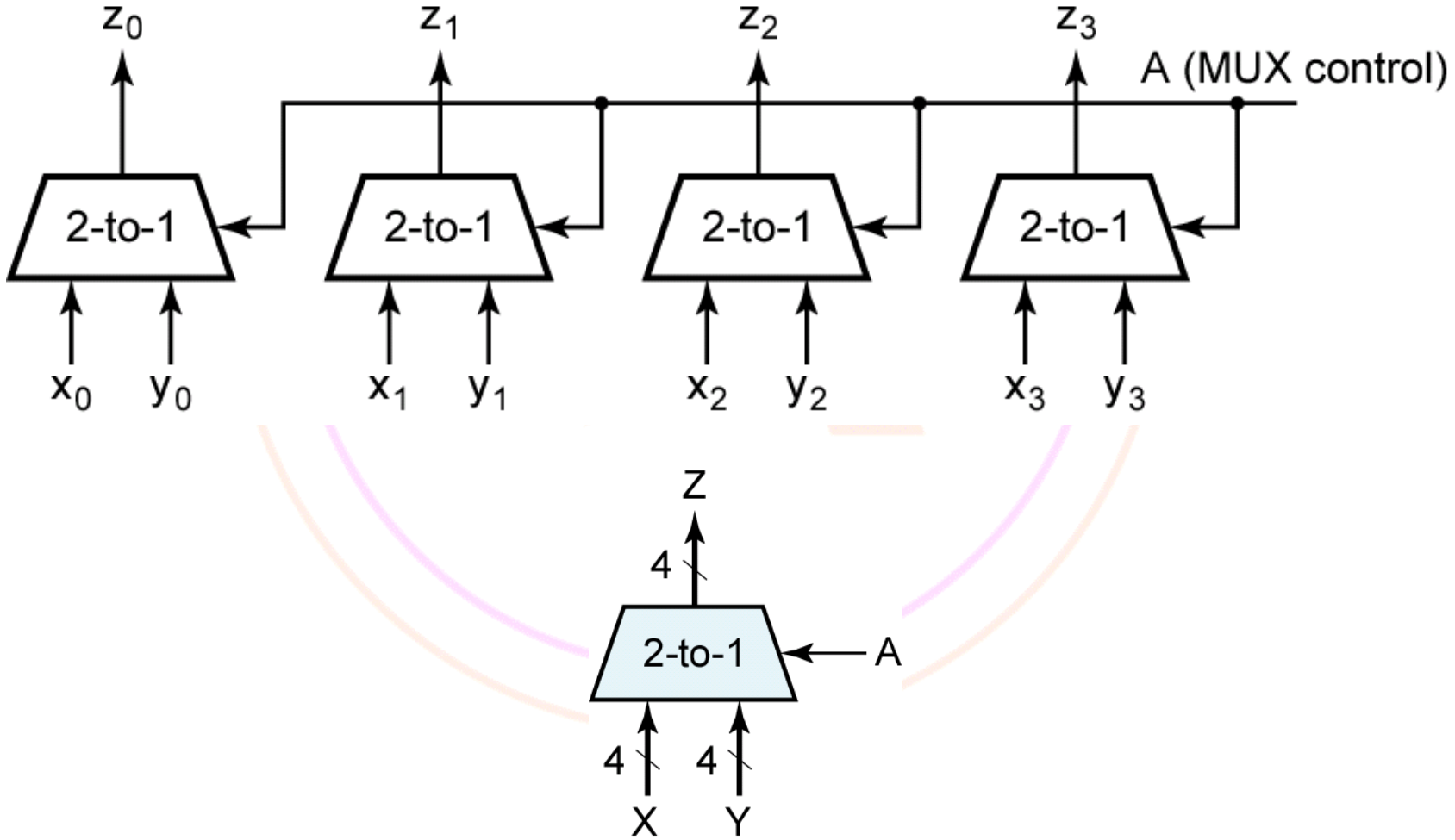
TEACHERS ADDA BY TARGET ABHI 7877719287

Multiplexers

Design a 16-to-1 multiplexer using 4-to-1 multiplexers only.

TEACHERS ADDA BY TARGET ABHI 7877719287

Multiplexer (Bus)



TEACHERS ADDA BY TARGET ABHI 7877719287

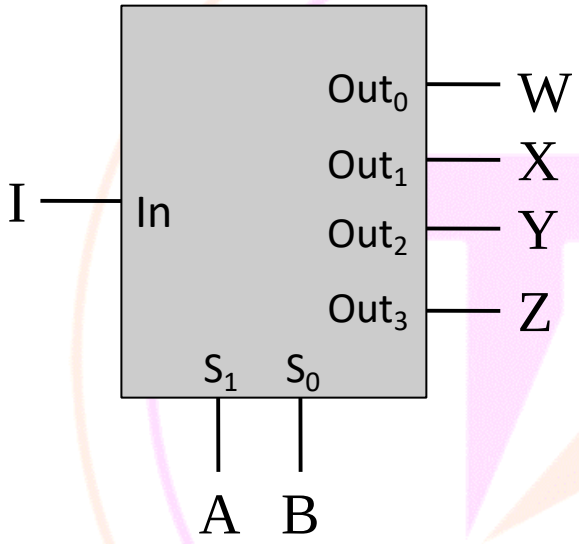
Demultiplexers

Demultiplexers

- A demultiplexer has
  - N control inputs
  - 1 data input
  - 2^N outputs
- A demultiplexer routes (or connects) the data input to the selected output.
  - The value of the control inputs determines the output that is selected.
- A demultiplexer performs the opposite function of a multiplexer.

TEACHERS ADDA BY TARGET ABHI 7877719287

Demultiplexers



W = A'.B'.I

X = A.B'.I

Y = A'.B.I

Z = A.B.I

A	B	W	X	Y	Z
0	0	I	0	0	0
0	1	0	I	0	0
1	0	0	0	I	0
1	1	0	0	0	I

TEACHERS ADDA BY TARGET ABHI 7877719287

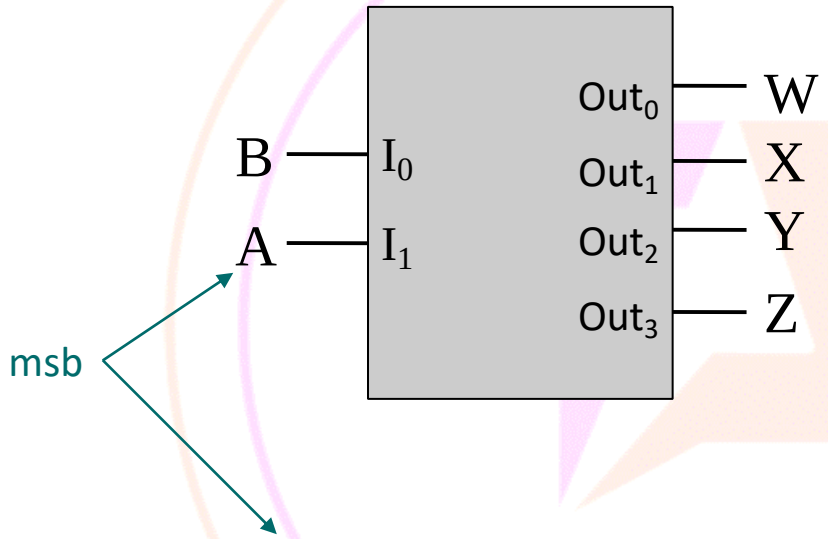
Decoders

Decoders

- A decoder has
  - N inputs
  - 2^N outputs
- A decoder selects one of 2^N outputs by decoding the binary value on the N inputs.
- Analogue to digital : encoder and vise versa

TEACHERS ADDA BY TARGET ABHI 7877719287

Decoders



W = A'.B'

X = A.B'

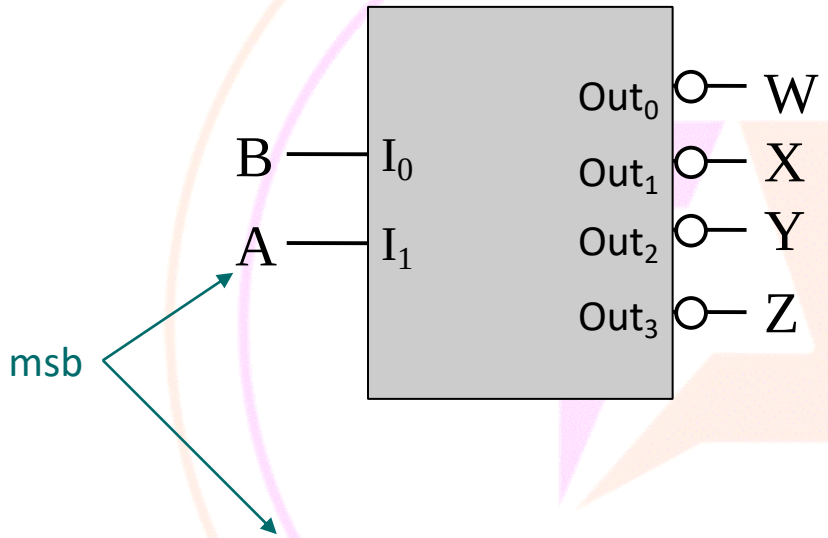
Y = A'.B

Z = A.B

A	B	W	X	Y	Z
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

TEACHERS ADDA BY TARGET ABHI 7877719287

Decoders



W = (A'.B')'

X = (A.B')'

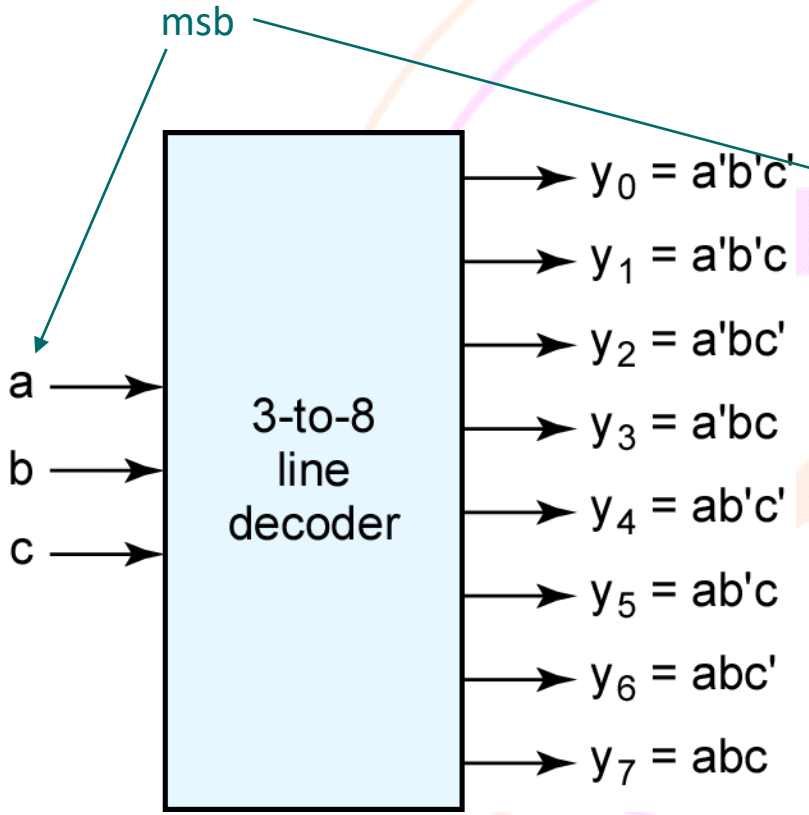
Y = (A'.B)'

Z = (A.B)'

A	B	W	X	Y	Z
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

TEACHERS ADDA BY TARGET ABHI 7877719287

Decoders



a	b	c	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7
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

TEACHERS ADDA BY TARGET ABHI 7877719287

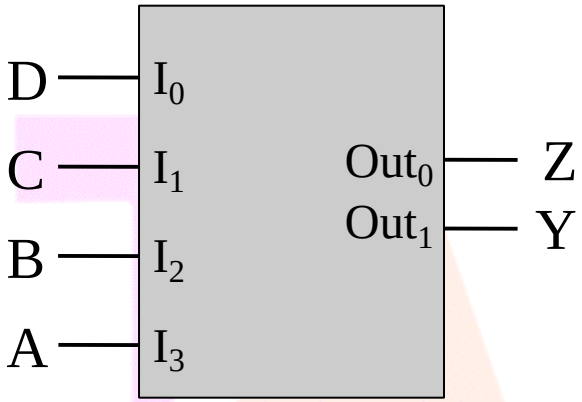
Encoders

Encoders

- An encoder has
  - 2^N inputs
  - N outputs
- An encoder outputs the binary value of the selected (or active) input.
- An encoder performs the inverse operation of a decoder.
- Issues
  - What if more than one input is active?
  - What if no inputs are active?

TEACHERS ADDA BY TARGET ABHI 7877719287

Encoders



A	B	C	D	Y	Z
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

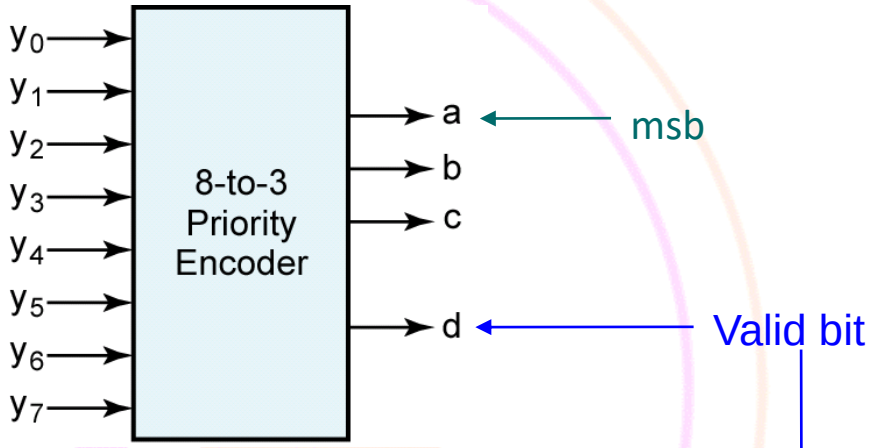
TEACHERS ADDA BY TARGET ABHI 7877719287

Priority Encoders

- If more than one input is active, the higher-order input has priority over the lower-order input.
  - The higher value is encoded on the output
- A valid indicator, d, is included to indicate whether or not the output is valid.
  - Output is invalid when no inputs are active
    - d = 0
  - Output is valid when at least one input is active
    - d = 1

TEACHERS ADDA BY TARGET ABHI 7877719287

Priority Encoders



y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7	a	b	c	d
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1
X	1	0	0	0	0	0	0	0	0	1	1
X	X	1	0	0	0	0	0	0	1	0	1
X	X	X	1	0	0	0	0	0	1	1	1
X	X	X	X	1	0	0	0	1	0	0	1
X	X	X	X	X	1	0	0	1	0	1	1
X	X	X	X	X	X	1	0	1	1	0	1
X	X	X	X	X	X	X	1	1	1	1	1

TEACHERS ADDA BY TARGET ABHI 7877719287

Sequential Logic Circuits

FLIP-FLOPS

- Both **Latches and flip flops** are circuit elements wherein the output not only depends on the current inputs, but also depends on the previous input and outputs. The main **difference between** the **latch and flip flop** is that a **flip flop** has a clock signal, whereas a **latch** does not.
- A **flip-flop** or **latch** is a circuit that has two stable states and can be used to store state information. A **flip-flop** is a bistable multivibrator. The circuit can be made to change state by signals applied to one or more control inputs and will have one or two outputs.
- Latches and flip-flops are the basic elements for storing information. One latch or **flip-flop can store one bit of information**. The main difference between latches and flip-flops is that for latches, their outputs are constantly affected by their inputs as long as the enable signal is asserted. In other words, when they are enabled, their content changes immediately when their inputs change. Flip- flops, on the other hand, have their content change only either at the rising or falling edge of the enable signal. This enable signal is usually the controlling clock signal. After the rising or falling edge of the clock, the flip-flop content remains constant even if the input changes.

TEACHERS ADDA BY TARGET ABHI 7877719287

- There are basically four main types of latches and flip-flops:
- **SR, D, JK, and T**. The major differences in these flip-flop types are the number of **inputs they have and how they change state**. For each type, there are also different variations that enhance their operations.
- Each type can have different variations such as active high or low inputs, whether they change state at the rising or falling edge of the clock signal, and whether they have asynchronous inputs or not. The flip-flops can be described fully and uniquely by its logic symbol, characteristic table, characteristic equation, state diagram, or excitation table, and are summarized in Figure below.

TEACHERS ADDA BY TARGET ABHI 7877719287

Flip-Flops : SR, D, JK, and T Flip-Flops

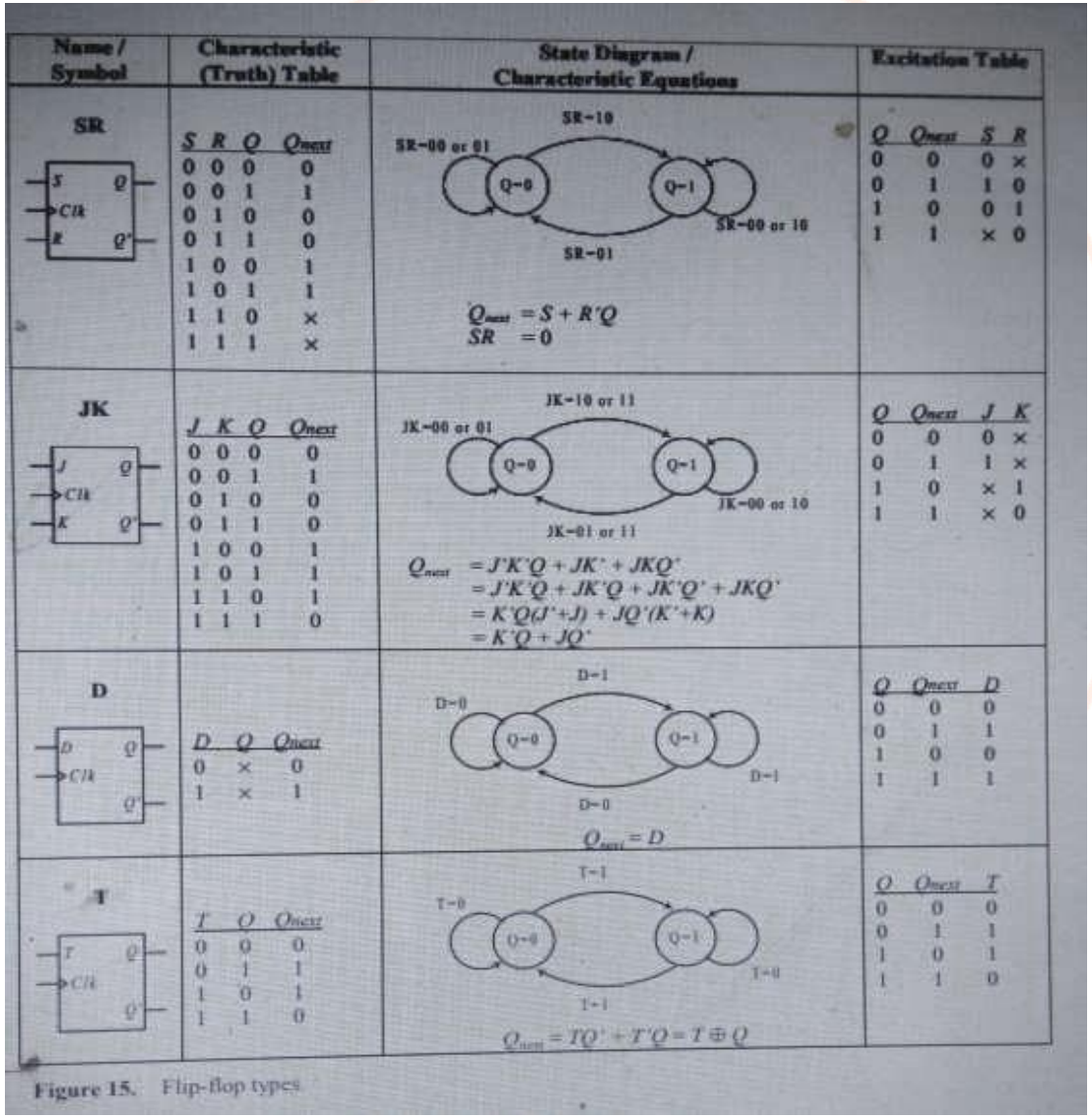


Figure 15. Flip-flop types

TEACHERS ADDA BY TARGET ABHI 7877719287

S-R Latch using NAND & NOR Gate :

- The bistable element is able to remember or store one bit of information. However, because it does not have any inputs, we cannot change the information bit that is stored in it. In order to change the information bit, we need to add inputs to the circuit. The simplest way to add inputs is to replace the two inverters with two NAND gates as shown in Figure. This circuit is called a SR latch.
- In addition to the two outputs Q and Q', there are two inputs S' and R' for set and reset respectively. Following the convention, the prime in S and R denotes that these inputs are active low. The SR latch can be in one of two states: **a set state when Q = 1, or a reset state when Q = 0**.

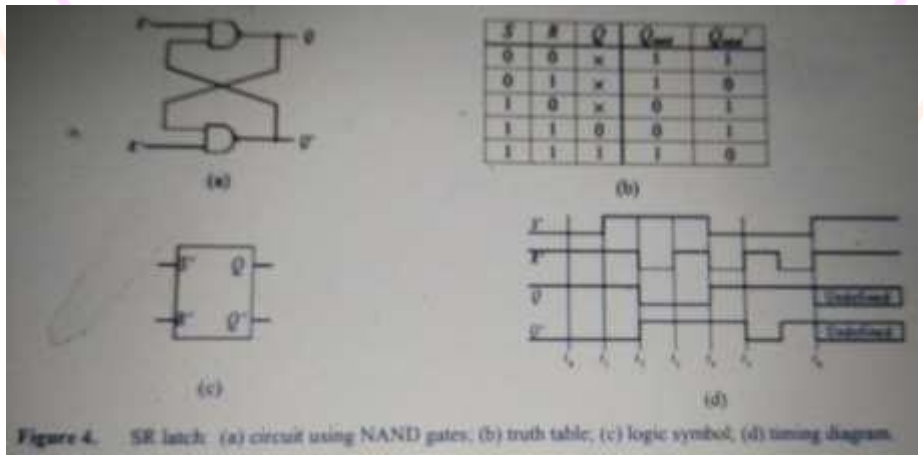


Figure 4. SR latch: (a) circuit using NAND gates; (b) truth table; (c) logic symbol; (d) timing diagram.

- Figure 4. SR latch: (a) circuit using NAND gates; (b) truth table; (c) logic symbol; (d) timing diagram.

TEACHERS ADDA BY TARGET ABHI 7877719287

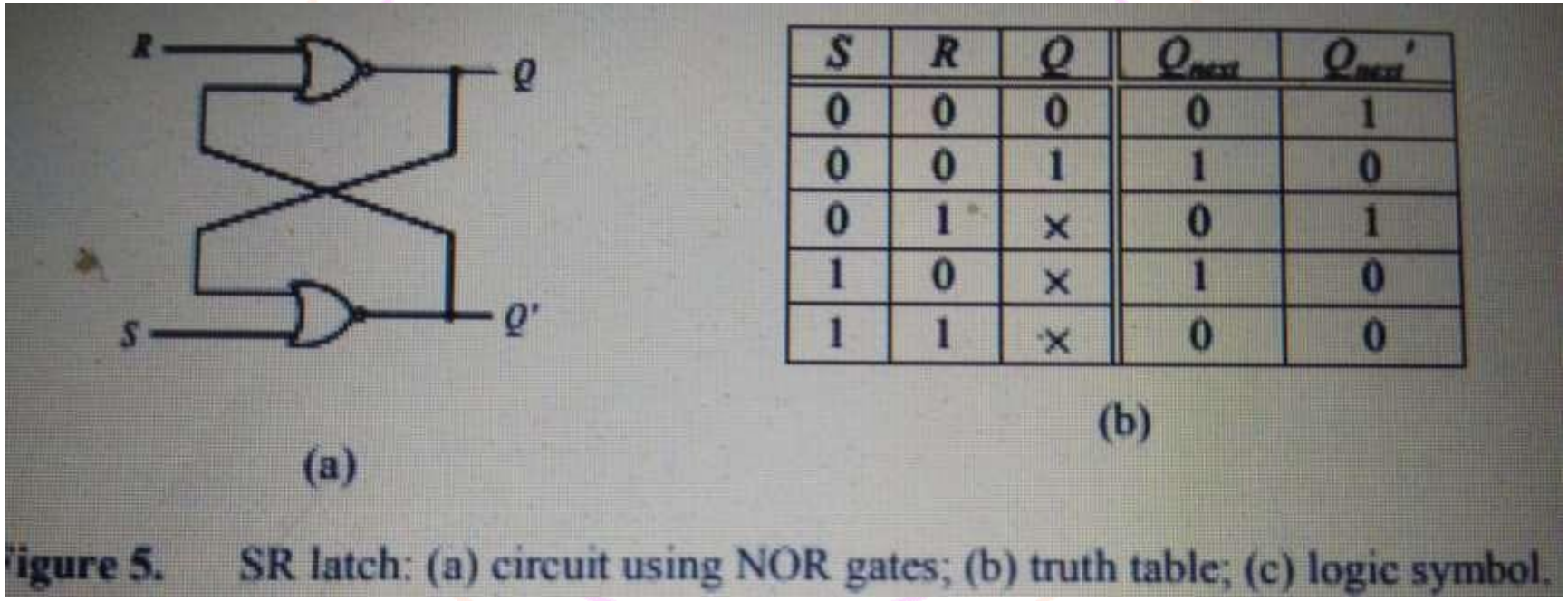
Continued…

- Like the NOR Gate S-R flip flop, this one also has four states. They are
- **S=0, R=1—Q=0, Q'=1**
- This state is also called the **SET** state.
- **S=1, R=0—Q=1, Q'=0**
- This state is known as the **RESET** state.
- In both the states you can see that the outputs are just compliments of each other and that the value of
- Q follows the compliment value of S.
- **S=0, R=0—Q=Q0, & Q' =Q0' No change**
- If both the values of S and R are switched to 0, then the circuit remembers the value of S and R in their previous state.
- **S=1, R=1—Q & Q'= Remember**
- If both the values of S and R are switched to 1 it is an invalid state because the values of both Q and Q' are 1. They are supposed to be compliments of each other. Normally, this state must be avoided.

TEACHERS ADDA BY TARGET ABHI 7877719287

Continued

- Another reason why we do not want both inputs to be asserted i.e. R=S=1 is that when they are both asserted, Q is equal to Q', but we usually want Q to be the inverse of Q'.



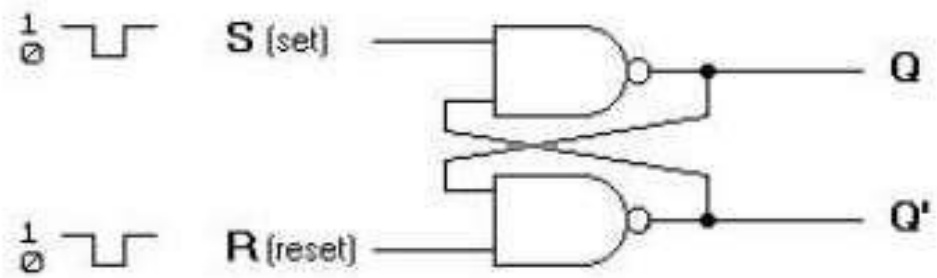
S	R	Q	Q_{next}	Q_{next}'
0	0	0	0	1
0	0	1	1	0
0	1	×	0	1
1	0	×	1	0
1	1	×	0	0

Figure 5. SR latch: (a) circuit using NOR gates; (b) truth table; (c) logic symbol.

- Figure 5. SR latch: (a) circuit using NOR gates; (b) truth table; (c) logic symbol.

TEACHERS ADDA BY TARGET ABHI 7877719287

S-R FLIP FLOP USING NAND GATE



(a) Logic diagram

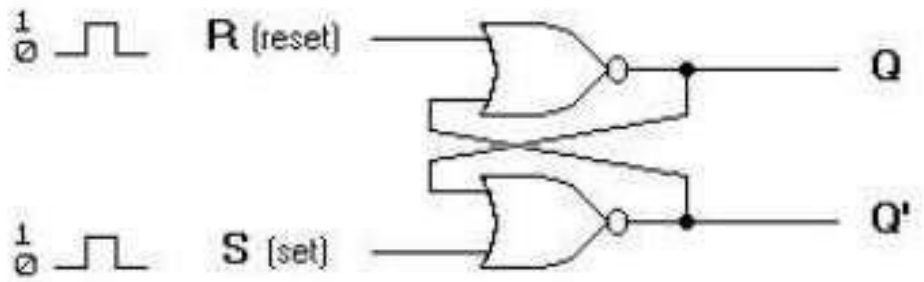
S	R	Q	Q'	
1	0	0	1	
1	1	0	1	(after S=1, R=0)
0	1	1	0	
1	1	1	0	(after S=0, R=1)
0	0	1	1	

(b) Truth table

Basic flip-flop circuit with NAND gates

TEACHERS ADDA BY TARGET ABHI 7877719287

S-R FLIP FLOP USING NOR GATE



(a) Logic diagram

S	R	Q	Q'	
1	0	1	0	
0	0	1	0	(after S=1, R=0)
0	1	0	1	
0	0	0	1	(after S=0, R=1)
1	1	0	0	

(b) Truth table

Basic flip-flop circuit with NOR gates

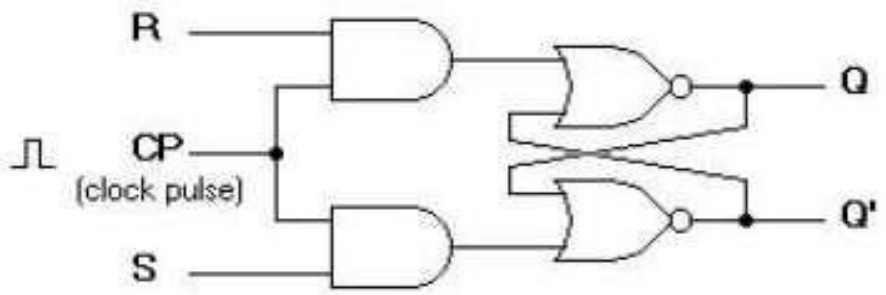
TEACHERS ADDA BY TARGET ABHI 7877719287

Clocked S-R Flip-Flop :

- The problems with S-R flip flops using NOR and NAND gate is the invalid state. This problem can be overcome by using a bistable SR flip-flop that can change outputs when certain invalid states are met, regardless of the condition of either the Set or the Reset inputs. For this, a clocked S-R flip flop is designed by adding two AND gates to a basic NOR Gate flip flop. The circuit diagram and truth table is shown below.
- A clock pulse [CP] is given to the inputs of the AND Gate. When the value of the **clock pulse is '0'**, the **outputs** of both the AND Gates **remain '0'**. As soon as a pulse is given the value of CP turns '1'. This makes the values at S and R to pass through the NOR Gate flip flop. But when the values of both S and R values turn '1', the HIGH value of CP causes both of them to turn to '0' for a short moment. As soon as the pulse is removed, the flip flop state becomes intermediate. Thus either of the two states may be caused, and it depends on whether the set or reset input of the flip-flop remains a '1'longer than the transition to '0' at the end of the pulse. **Thus the invalid states can be eliminated**.

TEACHERS ADDA BY TARGET ABHI 7877719287

Clocked S-R Flip-flop



(a) Logic diagram

Q S R	Q(t+1)
0 0 0	0
0 0 1	0
0 1 0	1
0 1 1	indeterminate
1 0 0	1
1 0 1	0
1 1 0	1
1 1 1	indeterminate

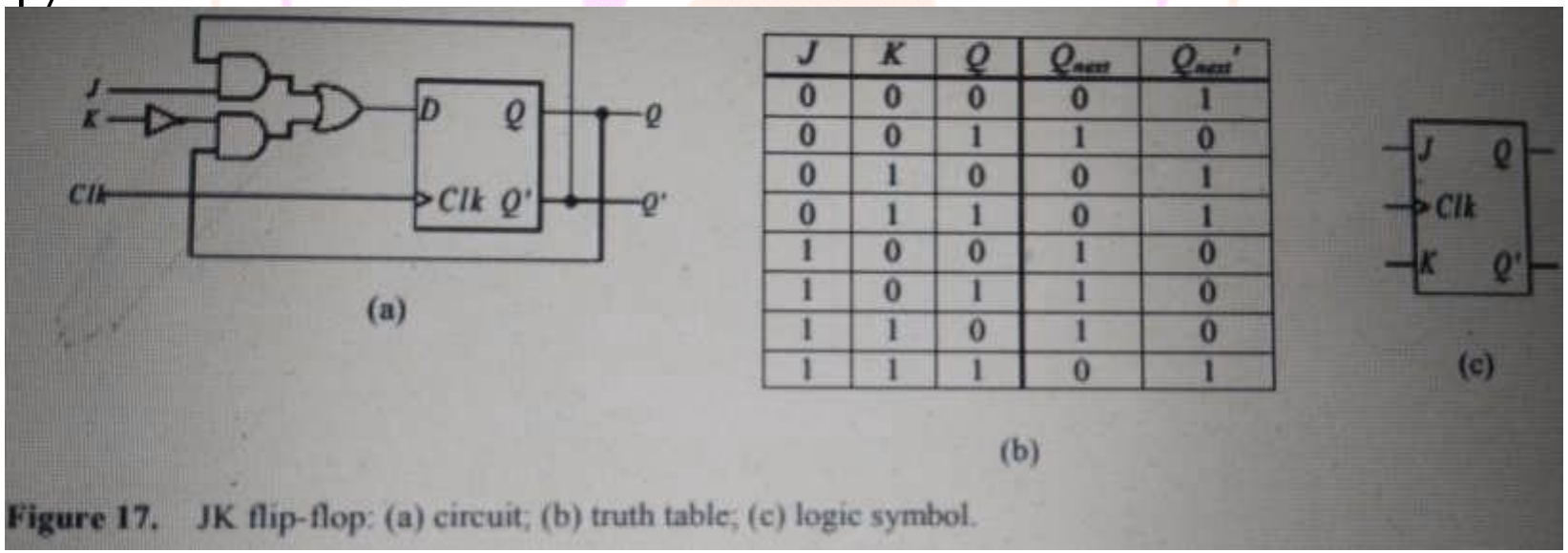
(b) Truth table

Clocked SR flip-flop

TEACHERS ADDA BY TARGET ABHI 7877719287

J-K FLIP-FLOP :

- JK flip-flops are very similar to SR flip-flops. The J input is just like the S input in that when asserted, it sets the flip-flop. Similarly, the K input is like the R input where it clears the flip-flop when asserted. **The only difference is when both inputs are asserted. For the SR flip-flop, the next state is undefined, whereas, for the JK flip-flop, the next state is the inverse of the current state. In other words, the JK flip-flop toggles its state when both inputs are asserted.** The circuit, truth table and the logic symbol for the JK flip-flop is shown in Figure 17.



J	K	Q	Q_{next}	Q_{next}'
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	0	1

(b)

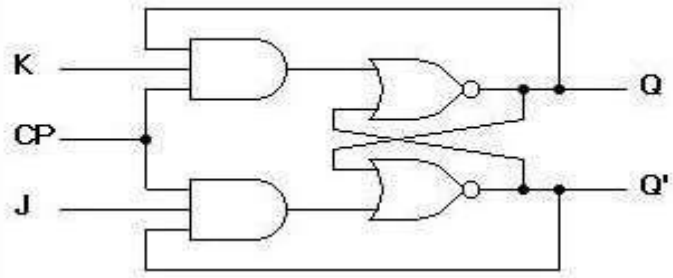
Figure 17. JK flip-flop: (a) circuit; (b) truth table; (c) logic symbol.

Figure 17. JK flip-flop: (a) circuit; (b) truth table; (c) logic symbol.

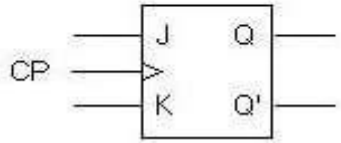
TEACHERS ADDA BY TARGET ABHI 7877719287

J-K Flip-Flop :

- A J-K flip flop can also be defined as a modification of the S-R flip flop. The only difference is that the intermediate state is more refined and precise than that of a S-R flip flop.
- The behavior of inputs J and K is same as the S and R inputs of the S-R flip flop. The letter J stands for SET and the letter K stands for CLEAR.



(a) Logic diagram



(b) Graphical symbol

Q	J	K	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

(c) Transition table

Clocked JK flip-flop

TEACHERS ADDA BY TARGET ABHI 7877719287

J-K Flip-Flop :

- When both the inputs J and K have a HIGH state, the flip-flop switch to the complement state. So, for a value of Q = 1, it switches to Q=0 and for a value of Q = 0, it switches to Q=1.
- The circuit includes two 3-input AND gates. The output Q of the flip flop is returned back as a feedback to the input of the AND along with other inputs like K and clock pulse [CP]. So, if the value of CP is 1, the flip flop gets a CLEAR signal and with the condition that the value of Q was earlier 1. Similarly output Q' of the flip flop is given as a feedback to the input of the AND along with other inputs like J and clock pulse [CP]. So the output becomes SET when the value of CP is 1 only if the value of Q' was earlier 1.
- **The output may be repeated in transitions once they have been complimented for J=K=1 because of the feedback connection in the JK flip-flop. This can be avoided by setting a time duration lesser than the propagation delay through the flip-flop. The restriction on the pulse width can be eliminated with a master- slave or edge-triggered construction.**

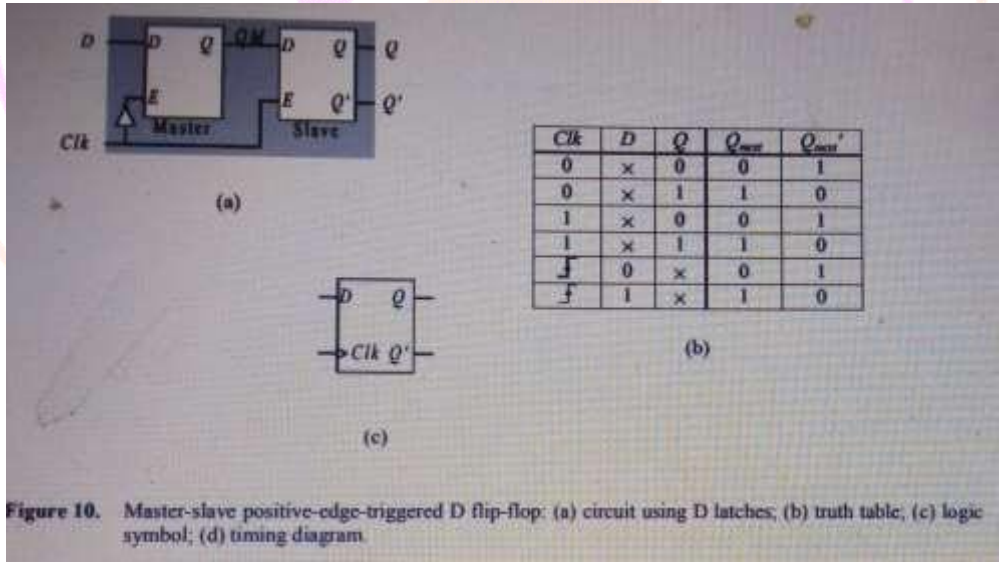
D TYPE FLIP-FLOP :

- Latches are often called level-sensitive because their output follows their inputs as long as they are enabled. They are transparent during this entire time when the enable signal is asserted. There are situations when it is more useful to **have the output change only at the rising or falling edge of the enable signal.** This enable signal is usually the controlling clock signal. Thus, we can have all changes synchronized to the rising or falling edge of the clock. An edge-triggered flip-flop achieves this by combining in series a pair of latches. Figure shows a positive edge-triggered D flip-flop where two D latches are connected in series and a clock signal Clk is connected to the E input of the latches, one directly, and one through an inverter. The first latch is called the master latch. The master latch is enabled when Clk = 0 and follows the primary input D. When Clk is a 1, the master latch is disabled but the second latch, called the slave latch, is enabled so that the output from the master latch is transferred to the slave latch. The slave latch is enabled all the while that Clk = 1, but its content changes only at the beginning of the cycle, that is, only at the rising edge of the signal because once Clk is 1, the master latch is disabled and so the

TEACHERS ADDA BY TARGET ABHI 7877719287

Continued…

- input to the slave latch will not change. The circuit of Figure 10(a) is called a positive edge-triggered flip-flop because the output Q on the slave latch changes only at the rising edge of the clock. If the slave latch is enabled when the clock is low, then it is referred to as a negative edge-triggered flip-flop. The circuit of Figure 10(a) is also referred to as a masterslave D flip-flop because of the two latches used in the circuit. Figure 10(b) and (c) show the truth table and the logic symbol respectively. Figure 10(d) shows the timing diagram for the D flip-flop.



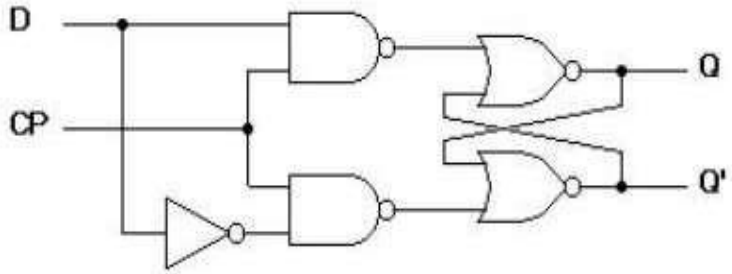
Clk	D	Q	Q_{next}	Q_{next}'
0	×	0	0	1
0	×	1	1	0
1	×	0	0	1
1	×	1	1	0
↓	0	×	0	1
↓	1	×	1	0

Figure 10. Master-slave positive-edge-triggered D flip-flop: (a) circuit using D latches; (b) truth table; (c) logic symbol; (d) timing diagram

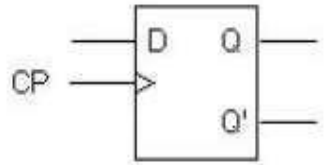
TEACHERS ADDA BY TARGET ABHI 7877719287

D type Flip-Flop :

- The circuit diagram and truth table is given in figure
- D flip flop is actually a slight modification of the above explained **clocked SR flip-flop**. From the figure you can see that the D input is connected to the S input and the complement of the D input is connected to the R input. The D input is passed on to the flip flop when the value of CP is '1'.
- When CP is HIGH, the flip flop moves to the SET state. If it is '0', the flip flop switches to the CLEAR state.



(a) Logic diagram with NAND gates



(b) Graphical symbol

Q D	Q(t+1)
0 0	0
0 1	1
1 0	0
1 1	1

(c) Transition table

Clocked D flip-flop

TEACHERS ADDA BY TARGET ABHI 7877719287

T type Flip-Flop

- The T flip-flop has one input in addition to the clock. T stands for **toggle** for the obvious reason. When T is asserted (T = 1), the flip-flop state toggles back and forth, and when T is de-asserted, the flip-flop keeps its current state. The T flip-flop can be constructed using a D flip-flop with the two outputs Q and Q' feedback to the D input through a multiplexer that is controlled by the T input as shown in Figure 18.

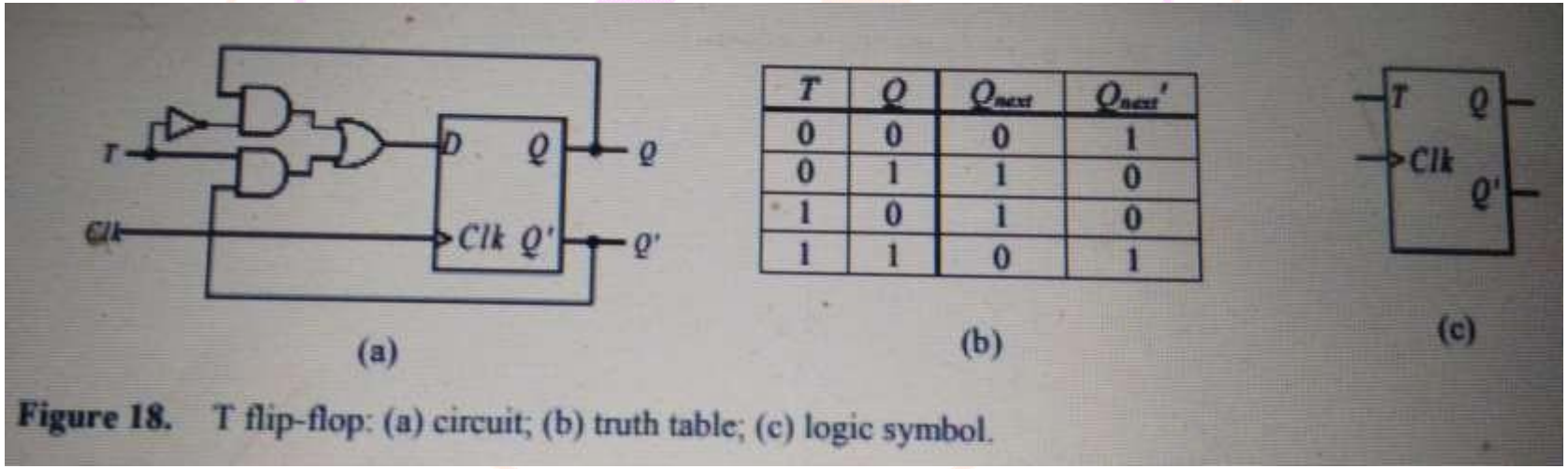


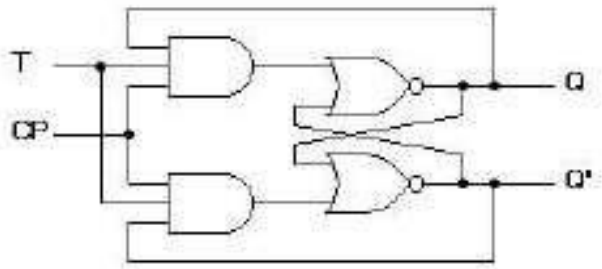
Figure 18. T flip-flop: (a) circuit; (b) truth table; (c) logic symbol.

- Figure 18. T flip-flop: (a) circuit; (b) truth table; (c) logic symbol.

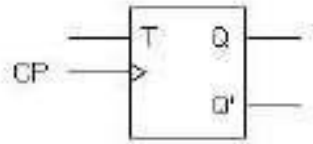
TEACHERS ADDA BY TARGET ABHI 7877719287

T TYPE FLIP-FLOP :

- This is a much simpler version of the J-K flip flop. Both the J and K inputs are connected together and thus are also called a single input J-K flip flop. When clock pulse is given to the flip flop, the output begins to toggle. Here also the restriction on the pulse width can be eliminated with a master-slave or edge-triggered construction. Take a look at the circuit and truth table is shown in figure.



(a) Logic diagram



(b) Graphical symbol

Q T	Q(t+1)
0 0	0
0 1	1
1 0	1
1 1	0

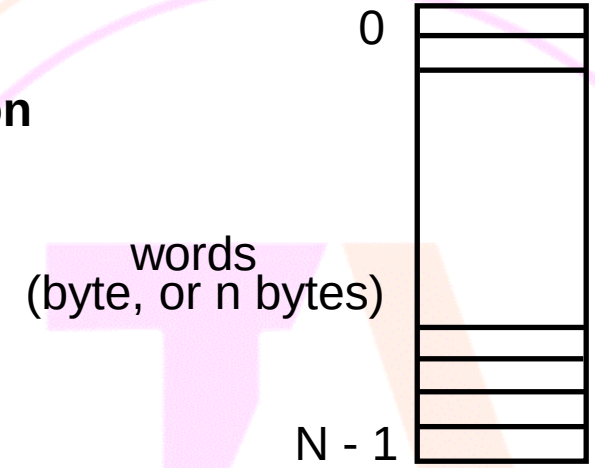
(c) Transition table

Clocked T flip-flop

TEACHERS ADDA BY TARGET ABHI 7877719287

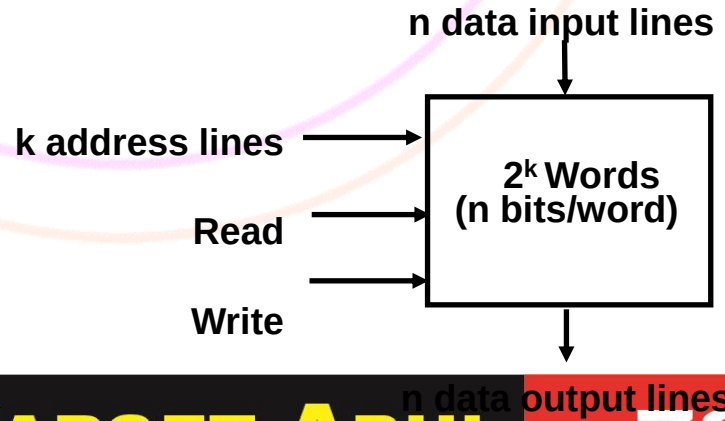
MEMORY COMPONENTS

Logical Organization



Random Access Memory

- **Each word has a unique address**
- **Access to a word requires the same time independent of the location of the word**
- **Organization**

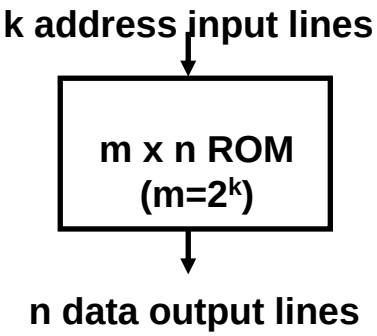


TEACHERS ADDA BY TARGET ABHI 7877719287

READ ONLY MEMORY(ROM)

Characteristics

- **Perform read operation only, write operation is not possible**
- **Information stored in a ROM is made permanent during production, and cannot be changed**
- **Organization**



Information on the data output line depends only on the information on the address input lines.

--> Combinational Logic Circuit

$X_0 = A'B' + B'C$
$X_1 = A'B'C + A'BC'$
$X_2 = BC + AB'C'$
$X_3 = A'BC' + AB'$
$X_4 = AB$

Canonical minterms

$X_0 = A'B'C' + A'B'C + AB'C$
$X_1 = A'B'C + A'BC'$
$X_2 = A'BC + AB'C' + ABC$
$X_3 = A'BC' + AB'C' + AB'C$
$X_4 = ABC' + ABC$

address — ***Output***

ABC	X_0	X_1	X_2	X_3	X_4
000	1	0	0	0	0
001	1	1	0	0	0
010	0	1	0	1	0
011	0	0	1	0	0
100	0	0	1	1	0
101	1	0	0	1	0
110	0	0	0	0	1
111	0	0	1	0	1

TEACHERS ADDA BY TARGET ABHI 7877719287

TYPES OF ROM

ROM

- Store information (function) during production
- Mask is used in the production process
- Unalterable
- Low cost for large quantity production --> used in the final products

PROM (Programmable ROM)

- Store info electrically using PROM programmer at the user's site
- Unalterable
- Higher cost than ROM -> used in the system development phase
  -> Can be used in small quantity system

EPROM (Erasable PROM)

- Store info electrically using PROM programmer at the user's site
- Stored info is erasable (alterable) using UV light (electrically in some devices) and rewriteable
- Higher cost than PROM but reusable --> used in the system development phase. Not used in the system production due to eras ability

INTEGRATED CIRCUITS

Classification by the Circuit Density

SSI - several (less than 10) independent gates
MSI - 10 to 200 gates; Perform elementary digital functions;
Decoder, adder, register, parity checker, etc
LSI - 200 to few thousand gates; Digital subsystem
Processor, memory, etc
VLSI - Thousands of gates; Digital system
Microprocessor, memory module

Classification by Technology

TTL - Transistor-Transistor Logic
Bipolar transistors
NAND
ECL - Emitter-coupled Logic
Bipolar transistor
NOR
MOS - Metal-Oxide Semiconductor
Unipolar transistor
High density
CMOS - Complementary MOS
Low power consumption

TEACHERS ADDA BY TARGET ABHI 7877719287