

MODULE 3 - DATA PROCESSING CIRCUITS

Multiplexers, Demultiplexers, 1-8-16 decoders, BCD-to-decimal decoders, seven-segment decoders, Encoders, Exclusive-OR Gates, Parity Generators & Checkers, Magnitude Comparators.

Clocks: Clock waveform, Edge triggering & level triggering

4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8, 4.9

7.1

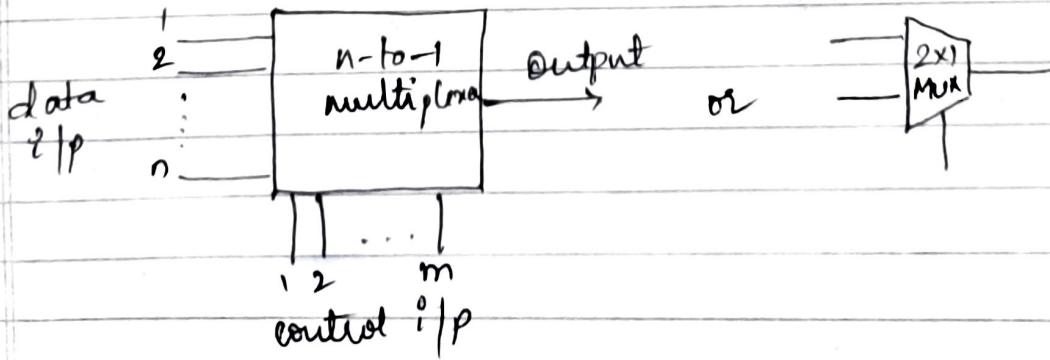
Data processing chs are logic chs that process the binary data. (Ex: text, voice,

A multiplexer is a device that selects two several analog or digital i/p signals & forwards it to a single output line.

A multiplexer has 2^n inputs, with n select lines which are used to select which i/p line to send to the output.

Applications of multiplexers:-

- effective data exchange.
- transmission of data over a single transmission line.
- It is a combinational ch that selects binary information from one of many i/p lines & directs into the o/p line.
 - also called as data selector
 - control i/p's are called as select i/p's or select lines.
 - a multiplexer ch would have n i/p signals, m control signals & 1 output signal.



$n \rightarrow$ no of ips
 $m \rightarrow$ no of select lines.

$$n \leq 2^m$$

$$m = \log_2 n, \text{ if } n=4$$

$$n=4$$

$$m = \log_2 2^2$$

$$\begin{aligned} m &= 2 \log_2 2 \\ &= 2 \log_2 2 = 2 \cdot 1 = \underline{\underline{2}} \end{aligned}$$

Advantages:-

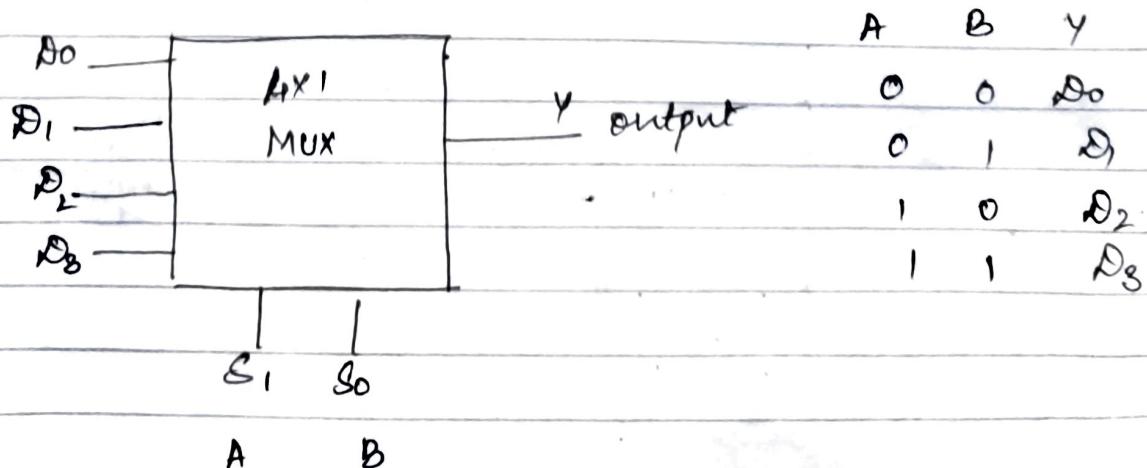
1. Reduces the no of wires, reduces the complexity and cost.
2. Implementation of various cts using MUX

Design of a 4:1 Multiplexer:-

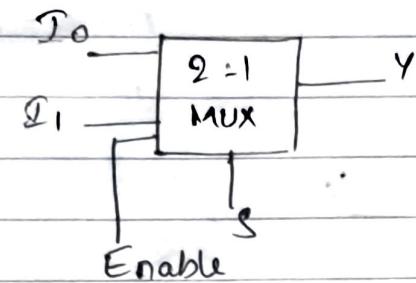
Control ips = 2 = A, B.

Data ips: 4 = D₀, D₁, D₂, D₃.

Output: Y



2:1 MUX



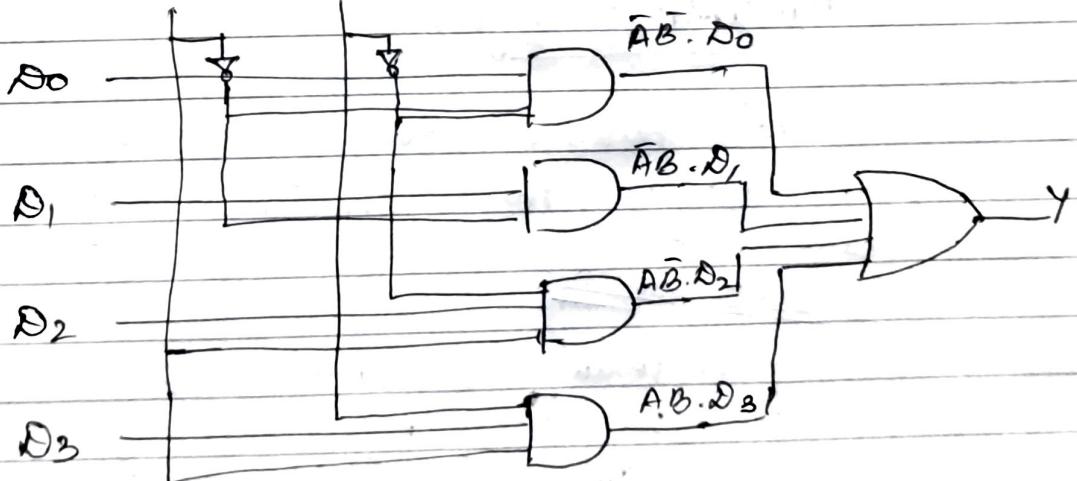
E	S	y
0	x	0
1	0	I_0
1	1	I_1

$$Y = \bar{A}\bar{B} \cdot D_0 + \bar{A} \cdot B \cdot D_1 + A\bar{B} \cdot D_2 + AB \cdot D_3$$

$$A=0, B=0 \quad Y = 1 \cdot 1 \cdot D_0 + 1 \cdot 0 \cdot D_1 + 0 \cdot 1 \cdot D_2 + 0 \cdot 0 \cdot D_3$$

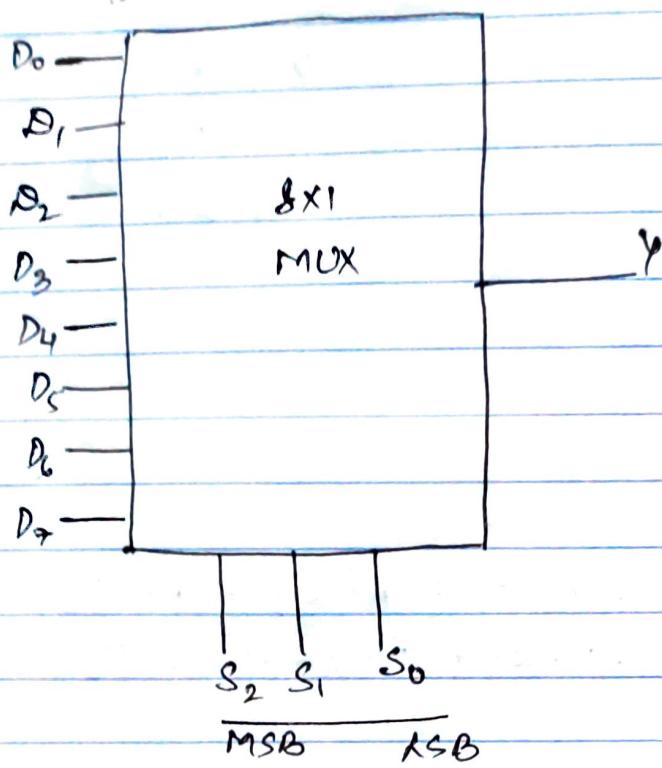
$$\boxed{Y = D_0}$$

Ckt design -



- multiplexer o/p Y is same as D_0 ie if $D_0=0$, $Y=0$, & $D_0=1$, $Y=1$

8:1 MUX



$$n = 8 = 2^3$$

$$m = 3$$

S_2	S_1	S_0	Y
0	0	0	D_0
0	0	1	D_1
0	1	0	D_2
0	1	1	D_3
1	0	0	D_4
1	0	1	D_5
1	1	0	D_6
1	1	1	D_7

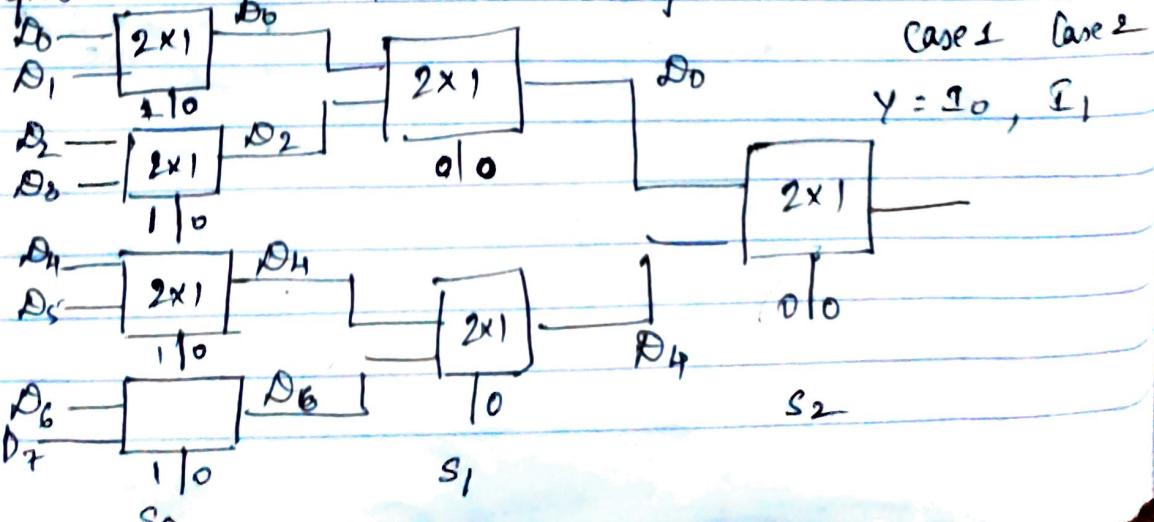
Multiplexer Logic

The bubbles on the signal line signifies the output is the complement of the selected data bit.

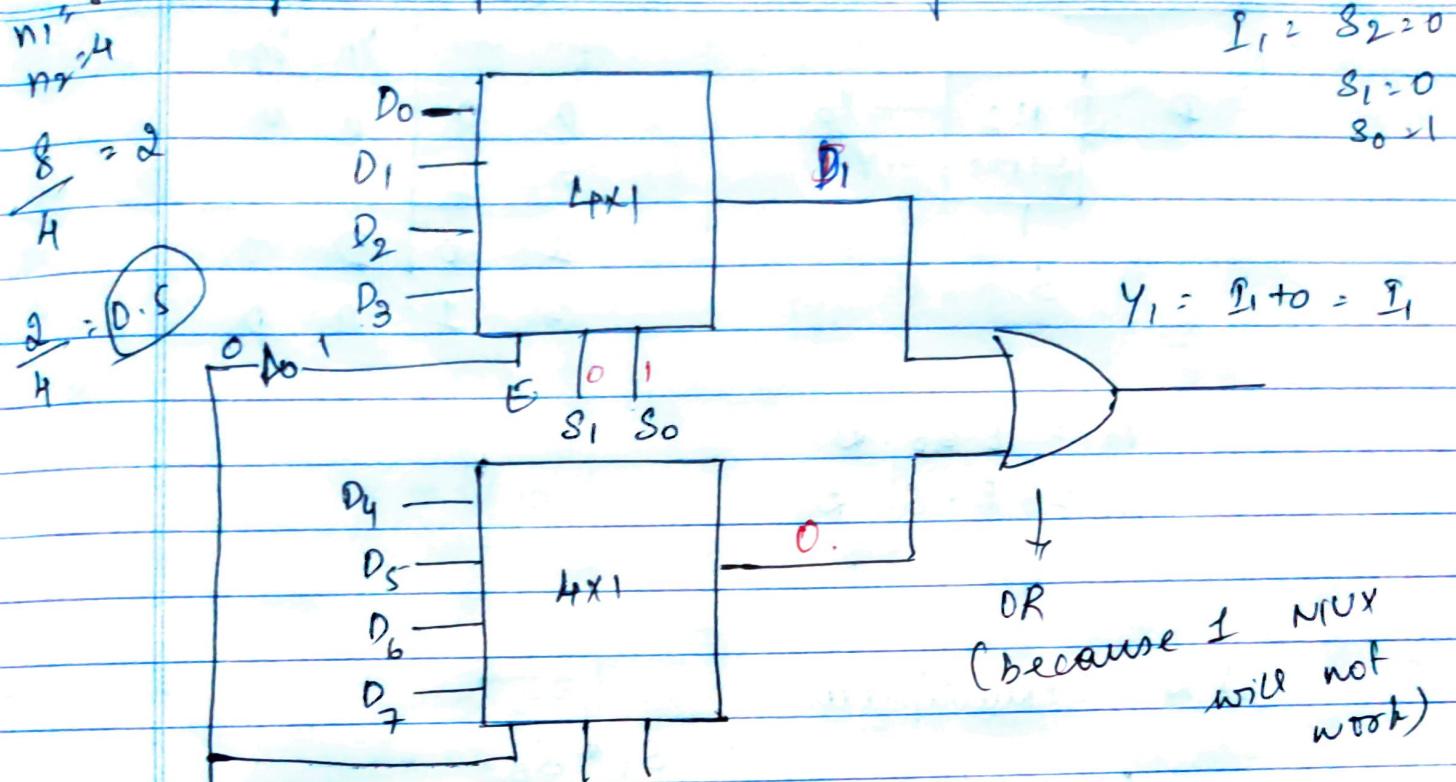
Conversion process

$$\begin{array}{l}
 \frac{8}{2} = 4 \\
 \frac{4}{2} = 2 \\
 \frac{2}{2} = 1
 \end{array}$$

Implementing 8x1 MUX using 2x1 MUX



Implementing 8:1 mux using 4x1 MUX

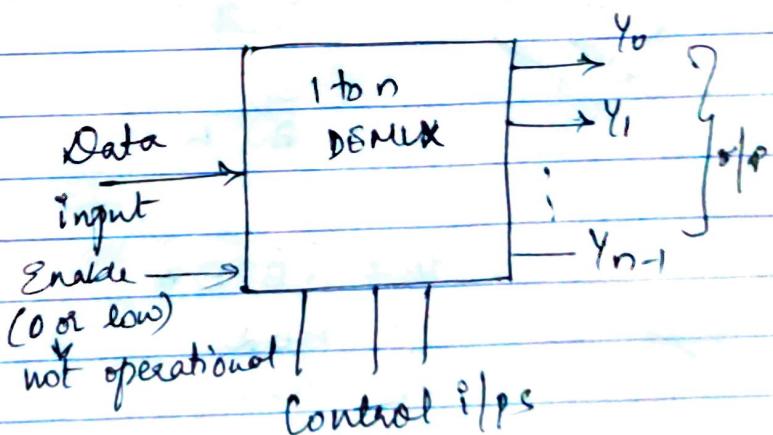


when $S_2 = 0$, the second multiplexor will not work

Demultiplexers (DEMUX)

- one into many chrt
- combinational chrt
- 1 i/p signal
- m control or select lines.
- n o/p signals

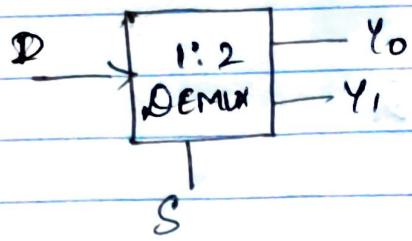
$$n \leq 2^m$$



- also called as a data distributor
- n - o/p lines, m - select lines

$$n = 2^m$$

1:2 DEMUX

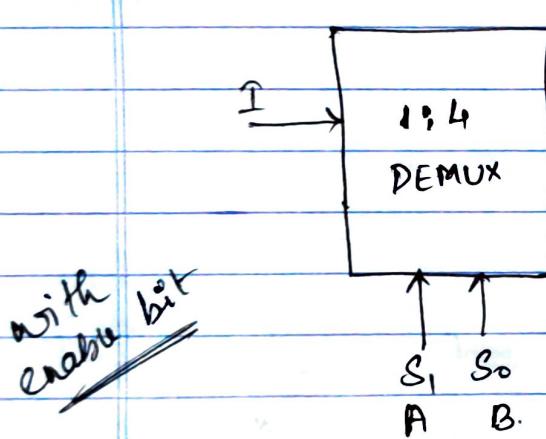


E	S0	Y0	Y1
-	0 0	0 0	
-	0 1	0 0	
-	1 0	D 0	
-	1 1	0 D	

$$Y_0 = E \bar{S}_0 \cdot D$$

$$Y_1 = E \cdot S_0 \cdot D$$

1:4 Demultiplexer.



$S_1 \oplus S_0$

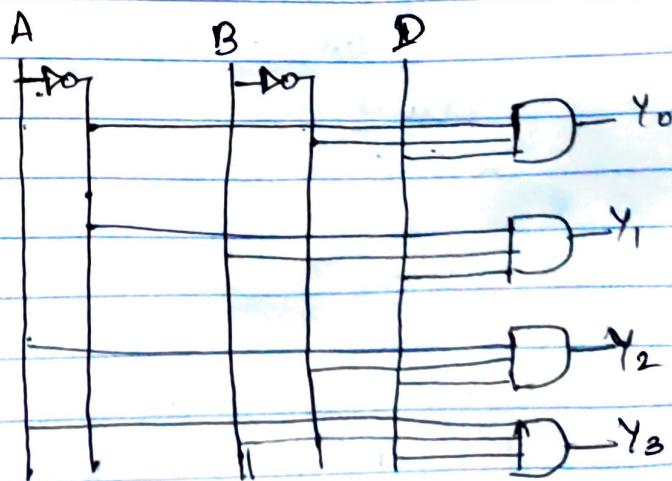
$S_1 \oplus S_0$		A	B	Y0	Y1	Y2	Y3
-	-	0 0	0 0	D	0	0 0	0
-	-	0 1	0 1	0	D	0 0	0
-	-	1 0	1 0	0	0	D	0
-	-	1 1	1 1	0	0	0	D

$$Y_0 = \bar{A} \bar{B} \cdot D$$

$$Y_1 = \bar{A} B \cdot D$$

$$Y_2 = A \bar{B} \cdot D$$

$$Y_3 = A B \cdot D$$

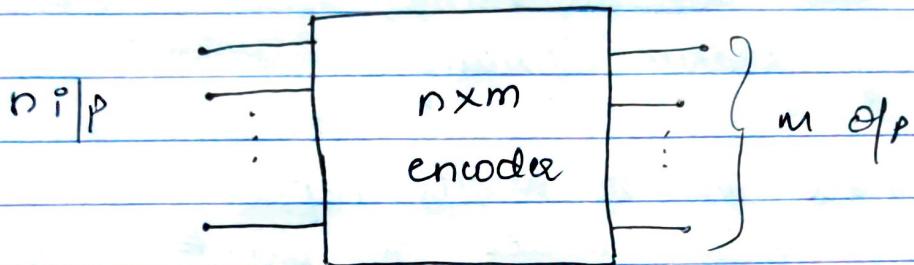


Introduction to decoders:-

(MSI - medium scale integrated ckt)

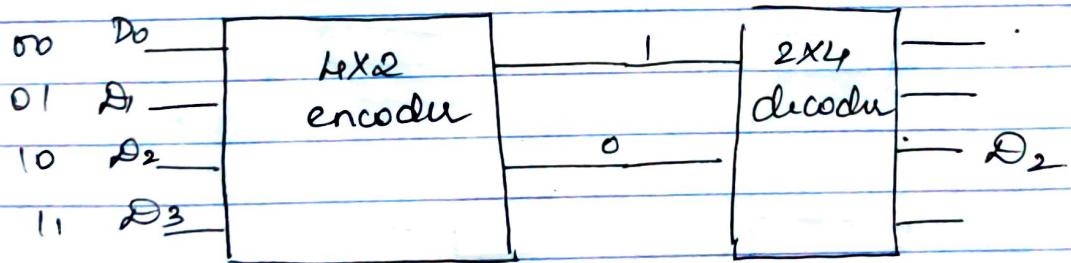
- combinational ckt

- Encoders have 'n' i/p & 'm' o/p

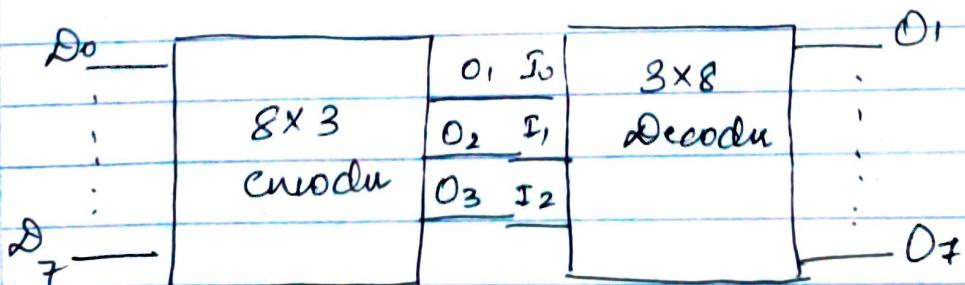


encoding process
conversion of data to binary

- encoders are used to minimize the no of data inputs.



If $n=4, n=2^2$
 $\therefore n=2^m$ \rightarrow encoder.



Decoders:-

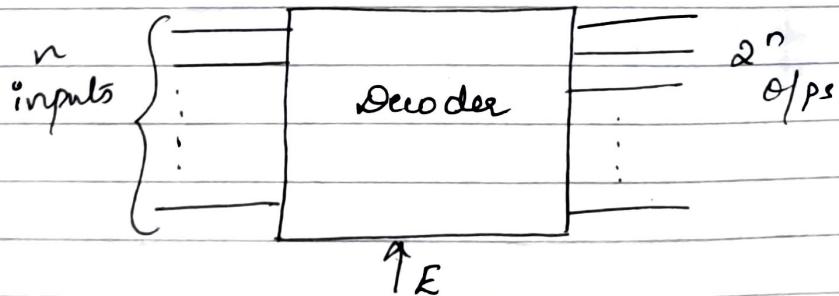
a decoder is a combinational logic circuit that converts binary information from ' n ' coded inputs to a maximum of 2^n unique outputs.

e.g.: data demultiplexing, seven segment displays, memory address decoding.

Decoder is a multi-input, multi-output logic circuit which decodes n inputs into 2^n possible outputs

↓
(Recognize or interpret an electronic signal)

Architecture :-



where $E = \text{Enable}$

Enable = 0 decoder is disabled
Enable = 1 decoder is enabled.

$2:4$ decoder:design:

			2^3	2^2	2^1	2^0	
	A	B	Y_3	Y_2	Y_1	Y_0	
n=0	0	0	0	0	0	1	enable Y_0 .
n=1	0	1	0	0	1	0	
n=2	1	0	0	1	0	0	
n=3	1	1	1	0	0	0	

(At a time only one output will be enabled by the decoder)

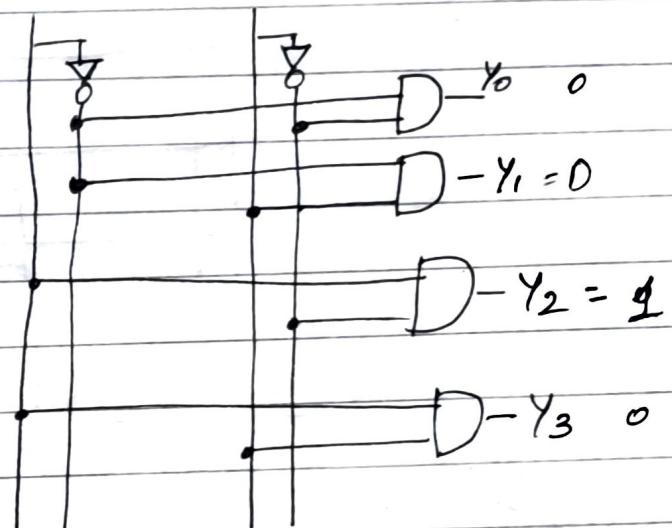
A 1 0
B

$$Y_3 = AB$$

$$Y_2 = A\bar{B}$$

$$Y_1 = \bar{A}B$$

$$Y_0 = \bar{A}\bar{B}$$

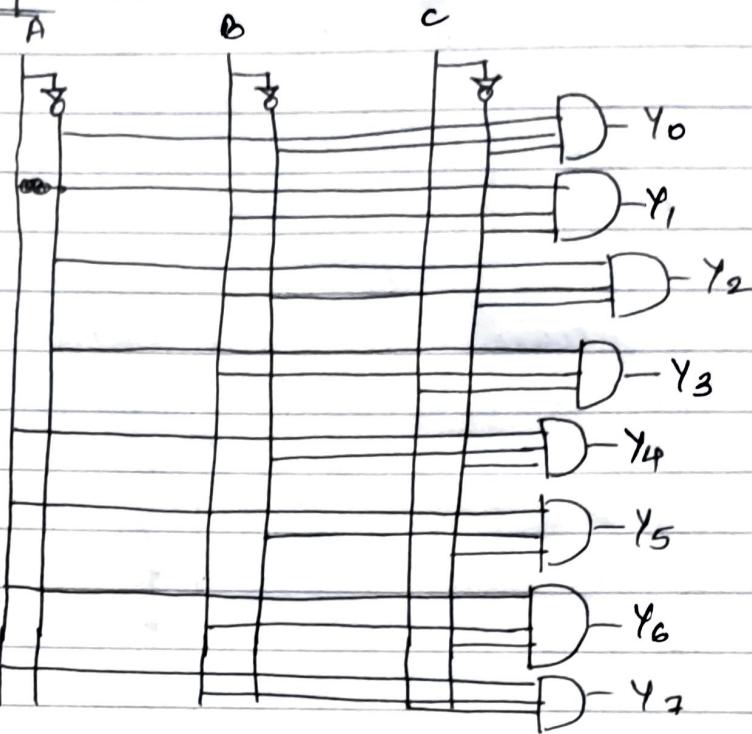


3 to 8 decoder design:-

$$n=8, m=2^n = 8$$

A	B	C	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	1	$Y_0 = \bar{A}\bar{B}\bar{C}$
0	0	1	0	0	0	0	0	0	1	0	$Y_1 = \bar{A}\bar{B}C$
0	1	0	0	0	0	0	0	1	0	0	$Y_2 = \bar{A}BC$
0	1	1	0	0	0	0	0	0	0	0	$Y_3 = \bar{A}B\bar{C}$
1	0	0	0	0	0	0	0	0	0	0	$Y_4 = A\bar{B}\bar{C}$
1	0	1	0	0	0	0	0	0	0	0	$Y_5 = A\bar{B}C$
1	1	0	0	0	0	0	0	0	0	0	$Y_6 = AB\bar{C}$
1	1	1	1	0	0	0	0	0	0	0	$Y_7 = ABC$

logic diagram:



1 of 16 decodes:-

Control line

inputs : a b c d → 4 ips

~~Fourth table~~

a	b	c	d	Y ₁₅	Y ₁	Y ₀
0	0	0	0	0	-	-	-	-	-	1
0	0	0	1	-	-	-	-	-	1	0

1 of 16 \rightarrow only 1 of the 16 o/p lines is high.

- also called as decimal to binary to decimal decoder.

- 4 i/p lines & 16 o/p lines - 4-line to 16-line decoder.

Difference b/w demux and decoder:

a) demux consists of one data i/p, few control i/p's & many outputs.

decoder - converts binary no. to its equivalent decimal number.

b) demux is a chip that receives the information on its single line & then transmits the info on any of the possible 2^n output lines.

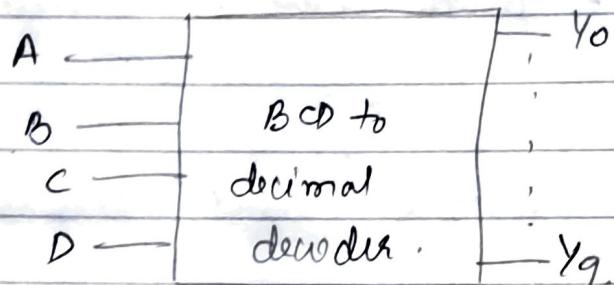
decoder has multiple i/p's & multiple o/p's & is used for the conversion of coded inputs into coded outputs.

c) for demux - the selection of a specific output o/p is controlled by the value of selection lines.

decoder - no selection lines in case of a decoder.

demux: no. of input lines in a demux is 1
 decoder: 2 or more than 2.

BCD - to - decimal decoders:-

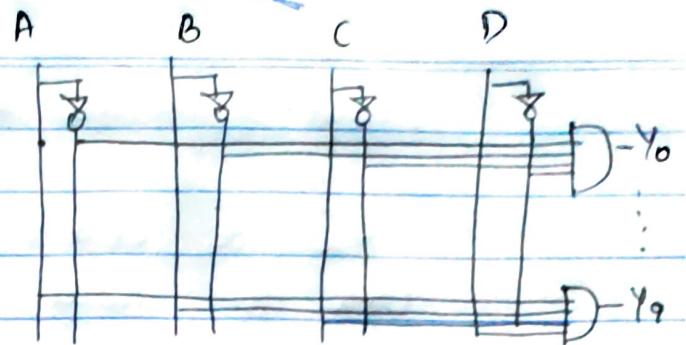


A	B	C	D	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9
0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	1	0	0
0	1	1	1	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	0	0	0	0

		y_2					
		00	01	11	10		
00		y_0	y_1	y_3	y_2	$x'y_2'$	
01		y_4	y_5	y_7	y_6		$x'y_2$
11		X	X	X	X		xyz
10		y_8	y_9	X	X		xyz'
				\downarrow	\downarrow		
				$1yz'$	xyz		

~~1000
0010
1010~~

~~1001
0011
1100~~



$$D_2 = x'y'z'$$

$$D_9 = wz$$

$$D_0 = w'x'y'z'$$

$$D_1 = w'a'y'z$$

$$D_3 = x'yz$$

$$D_4 = xy'z'$$

$$D_5 = xyz$$

$$D_6 = x'yz$$

$$D_7 = xzy$$

$$D_8 = wz'$$

Q.

1-of-10 decoder.

↓

Since only 1 of the 10
ofp lines is high.

BCD - binary coded decimal. - BCD expresses
each digit in a decimal no by its equivalent
nibble.

Ex. 429

✓ 1 —

0100 0010 1001

Ex: 8963

1000 1001 0110 0011

Pin out diagram - 7445

1 - 7 ($y_0 - y_6$), 8 - GND, 9 - 11 ($y_7 - y_9$)

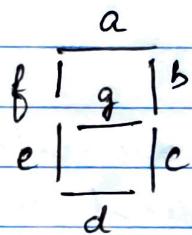
12, 13, 14, 15, 16 - Vcc.

A B C D

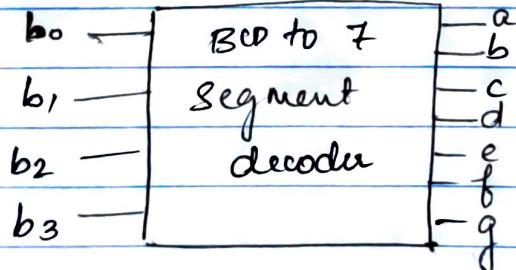
Seven-Segment Decoder:-

In a LED - the free electrons recombine with the holes near the junction - and since the free electrons fall from a higher energy level to a lower one - the energy is emitted in the form of heat and light.

0 1 2 3 4 5 6 7 8 9



1 - 0001 - to be converted so that b a c are illuminated.



the intermediate entity which converts 0001 is called as decoder.

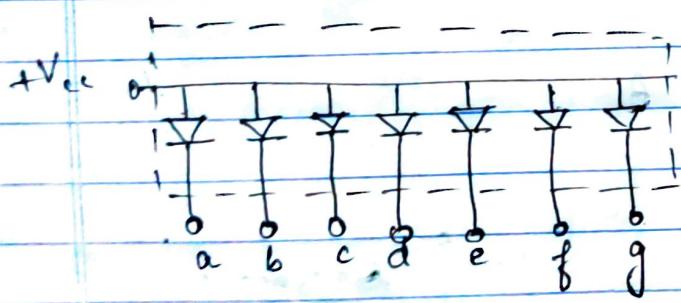
b_3	b_2	b_1	b_0	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0
2	0	0	1	0	1	1	0	1	1	0
3	0	0	1	1	1	1	1	0	0	1
4	0	1	0	0	1	1	0	0	1	1
5	0	1	0	1	0	1	1	0	1	1

	a	b	c	d	e	f	g
0	0	1	0	1	0	1	1
0	1	1	1	1	1	0	0
1	0	0	0	-	1	1	1
1	0	0	1	1	1	1	0
1	0	1	0				
1	0	1	1				
1	1	0	0				
1	1	0	1				
1	1	1	0				
1	1	1	1				

Don't care terms

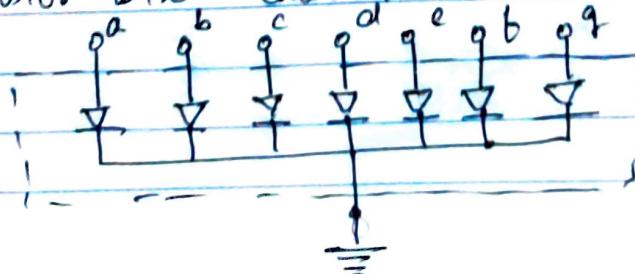
Seven-segment indicators may be:

- a) common-anode type
- b) common-cathode type.



- in this case a current limiting resistor is connected below each LED to ground.
- size of the resistors determines how much current flows through the LED.

Common cathode type: uses a. current-limiting resistor below each LED and +Vcc



Mr Chaudhary
A.T.K.C

K-map for a .

		b ₃	b ₂	b ₁	b ₀	
		00	01	11	10	
00	1	0	1	1	1	→ II
01	0	1	1	1	1	
11	X	X	X	X	X	
10	1	1	X	X	X	→ I
		↓ III	↓ IV			

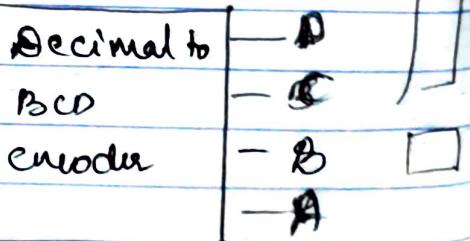
$$\begin{aligned}
 a &= b_3 + b_1 + \overline{b_2} \overline{b_0} + b_2 b_0 \\
 &= b_3 + b_1 + b_2 \oplus b_0
 \end{aligned}$$

	00	01	11	10
00	1 ₆	0 ₁	1 ₃	1 ₂
01	0 ₄	1 ₅	1 ₇	1 ₆
11	X ₁₂	X ₁₃	X ₁₄	X ₁₅
10	1 ₈	1 ₉	X ₁₁	X ₁₀
	↓ III	↓ IV	↓ II	↓ I
	0-15			

Decimal to BCD Encoder:-

$$d = 10 \rightarrow [0 \text{ to } 9]$$

Input	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1



1 - 2

4

8

16

D is high:

$$D = 8 + 9$$

C is high for:

$$C = 5 + 6 + 7 + 4$$

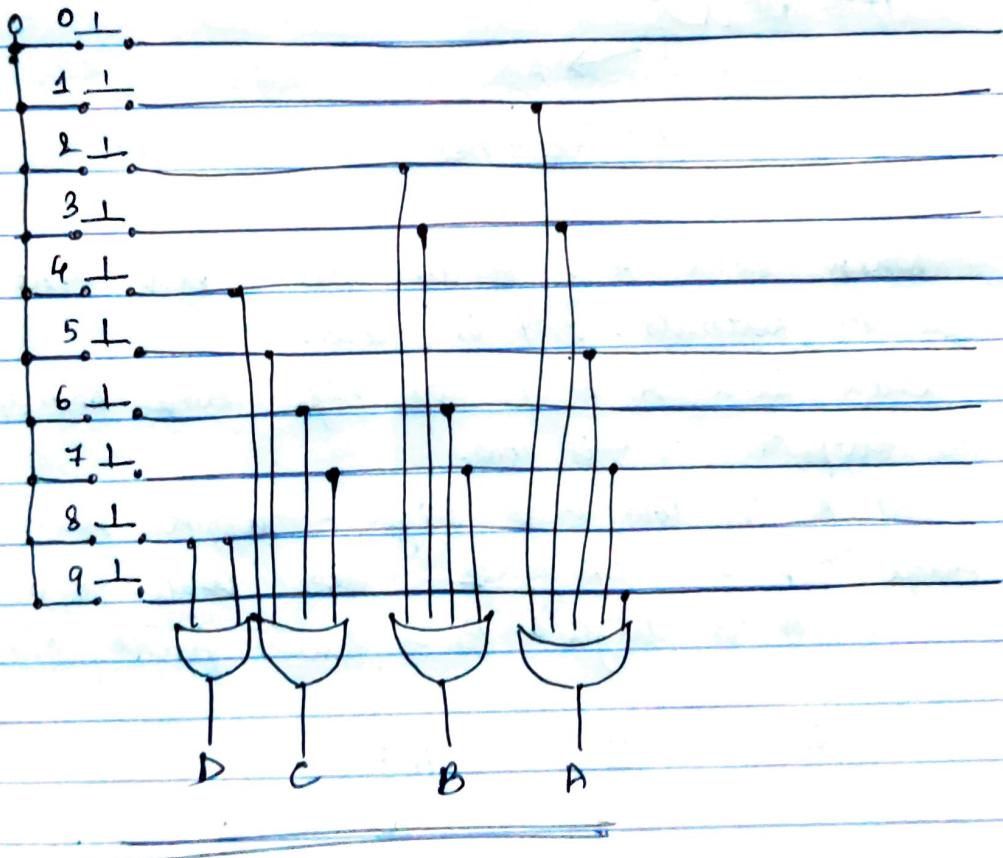
B is high for:

$$B = 2 + 3 + 6 + 7$$

A is high for:

$$A = 1 + 3 + 5 + 7 + 9$$

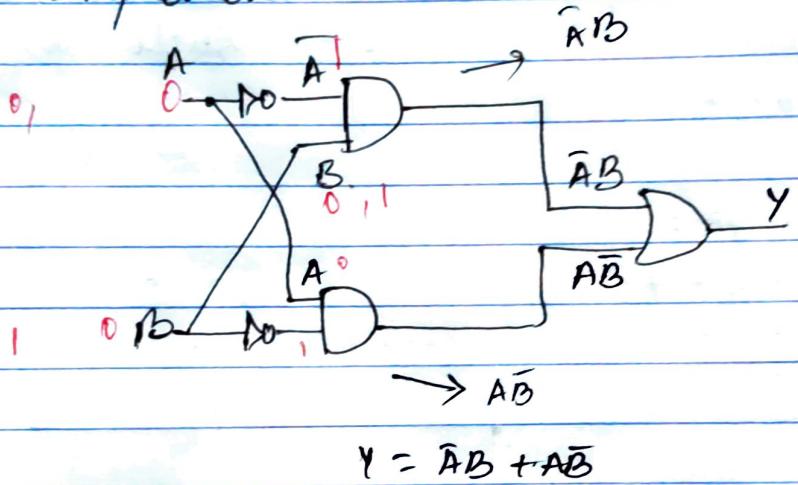
An encoder which accepts an active level on one of its i/p's representing a decimal digit & converts it into a coded o/p such as BCD



Exclusive-OR Gates

- has high o/p only when an odd no of i/p's is high.

Implementation:



When both A & B are low - both the AND gates have low outputs. $\therefore Y$ is low.

When both A & B are high - both AND gates have low outputs. $\therefore Y$ is low.

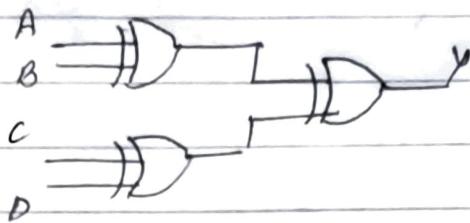
If A is low, B is high - upper AND gate has high o/p - \therefore OR gate has high o/p.

If A is high & B is low - final o/p is high.

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0



A i/p Exclusive OR gate:



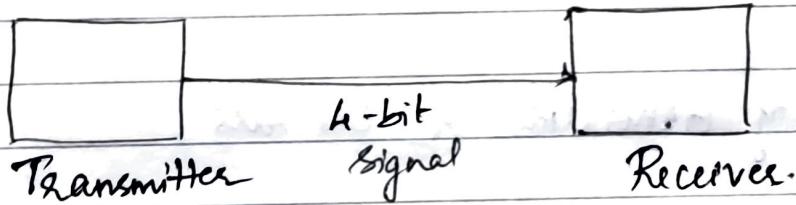
Corresponding truth table.

Parity Generators & Checkers:

(error detection & error correction)

Parity -

used in the detection of a single bit error



- Extra bit - sent with the data bit stream
↓

total no of 1's in the transmitted signal - such that any changes in the data bit can be detected.

Types of parity:

Even parity - Total no of 1's is even ie 0

Odd parity

* Consider the data stream to be 0100
So with parity bit - 01001 → Even parity.

1 1 00 0 → even parity

even no of 1's in data, add 1

Odd parity:-

01 00 0 → parity bit

11 00 1 → parity bit

Consider a scenario:

Data stream to be sent: 0100

Because of noise: 0101 (gets changed)

Based on parity bit considered: 01011 → odd
↓
PB No of 1's

Parity Generators & checkers

- used in the detection of errors in digital transmission sdm.

Odd parity Generator:-

Parity checkers:

- EX-OR gates - used for checking the parity -
since they produce the o/p as 1 - with odd no
of 1's.

where even parity i/p to EXOR gate produces
low o/p, odd parity i/p produces high o/p

Parity Generators:

- addition of an extra bit to the original
binary no - to produce a new binary no with even
or odd parity.

Consider an example:

$x_7 x_6 x_5 x_4 \quad x_3 x_2 x_1 x_0$

0 1 0 0 0 0 0 1

→ even parity.



XOR gate produces the o/p as

$x_8 = 0$

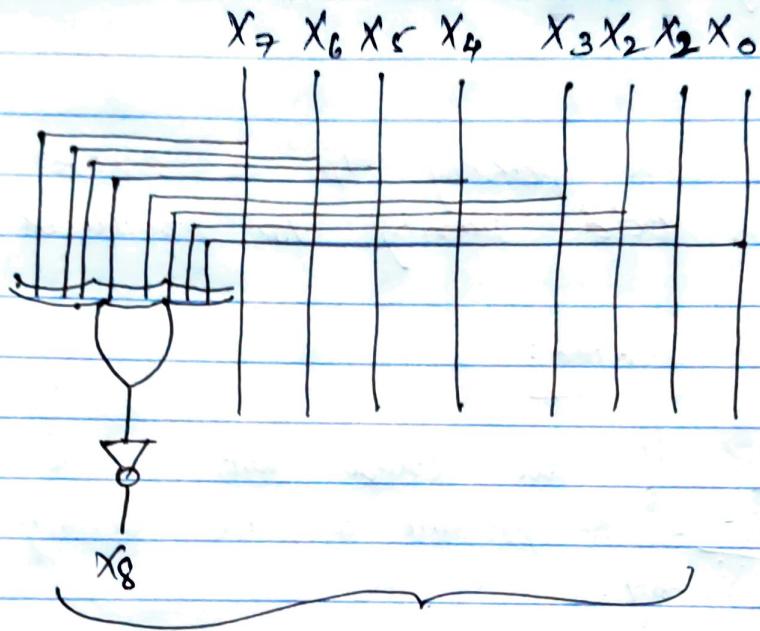
use of an inverter: $x_8 = 1$

where the final o/p with parity bit is:

1 0 1 0 0 0 0 0 1 → odd parity.

if the 8-bit input is changed to 0 1 1 0 0 0 0 1 → odd parity

XOR gate produces the op as 1, but the inverter produces a $\underline{0}$, such that final op is
 $0 \ 0110 \ 0001$



9-bit no with odd parity.

Application of parity checker & generator:

Cause of errors in data transmission:
 noise, disturbance in the medium,

Consider an example:

$$\begin{array}{r} \text{Data to be transmitted : } 0100 \ 0011 \\ + \text{ Odd parity } = 0 \ 0100 \ 0011 \end{array}$$

In case of two errors, odd parity checker at the receiver side will produce a high op - representing odd parity in the data received.

in case of a 1-bit error - the parity-checker will give a low o/p representing even parity - with received data being invalid.

74180 IC

- input data bits vary from X_7 to X_0 .

$X_7 X_6 X_5 X_4 X_3 X_2 X_1 X_0$, 14 - V_{CC} , 7 - GND
 , 2 8 9 10 11 12 13

3 - Even i/p, 4 - odd i/p - used to control the operation of the chip.

Σ even & Σ odd

5 6

Truth Table:-

		Inputs		Outputs	
		Even	Odd	Even	Odd
parity P_i input data	Even	H	L	H	L
	odd	H	L	L	H
	Even	L	H	L	H
	odd	L	H	H	L

control inputs

Magnitude Comparator:-

- used to compare the magnitude of two n-bit binary numbers.

$x = Y$, $x > y$ and $x < y$

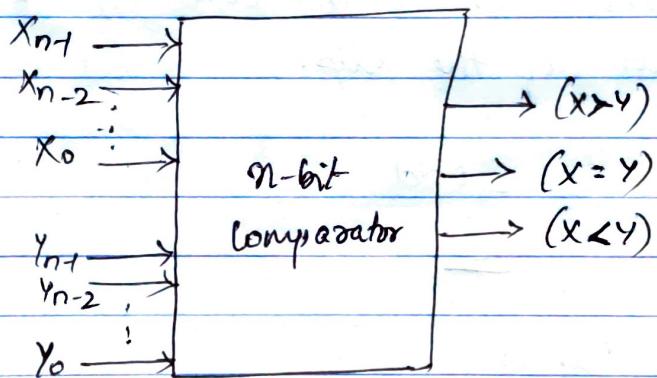
Logical Equations :-

$$(x \succ y) \Rightarrow G; xy'$$

$$(x \prec y) \rightarrow L = x'y$$

$$(x = y) \rightarrow E = x'y' + xy = (xy' + x'y)' = (G+L)$$

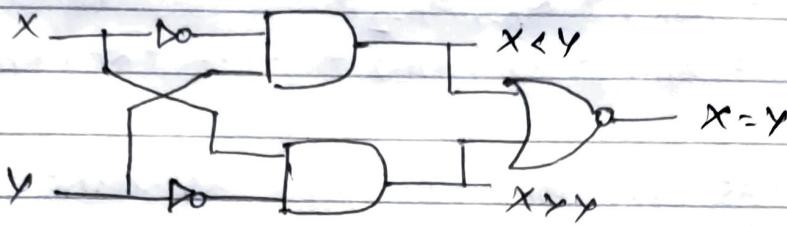
Block diagram of a magnitude comparator:



Truth Table:

Input	Output			
X	Y	$x > y$	$x \prec y$	$x = y$
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

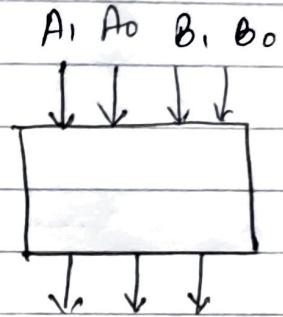
1-bit comparator:



How to design of α -bit comparators:

$$X: X_1, X_0$$

$$Y: Y_1, Y_0$$



Bit-wise greater than term (G):

$$G_1 = X_1 Y_1'$$

$$G_0 = X_0 Y_0'$$

Bit-wise less than term (L):

$$L_1 = X_1' Y_1,$$

$$L_0 = X_0' Y_0$$

Bit-wise equality term (E):

$$E_1 = (G_1 + L_1)', \quad E_0 = (G_0 + L_0)'$$

A_1	A_0	B_1	B_0	$A < B$	$A = B$	$A > B$
0	0	0	0	0	1	0
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	1	0	0
0	1	1	1	1	0	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	1	0
1	0	1	1	0	0	0
1	1	0	0	0	0	1
1	1	0	1	0	0	0
1	1	1	0	0	0	1
1	1	1	1	0	1	0

$$(X = Y) \equiv E_1, E_0$$

$$(X > Y) = G_1 + E_1, G_0$$

$$(X < Y) = L_1 + E_1, L_0$$

$$\left. \begin{array}{l} (X_1, X_0) \\ (Y_1, Y_0) \\ G_1 = 1 \\ E_1 = 1 \\ G_0 = 1 \end{array} \right\}$$

$$X_1 > Y_1$$

$$G_1 = 1$$

$$X_1 = Y_1$$

$$E_1 = 1, \quad X_0 > Y_0$$

\therefore For any α -n-bit numbers: then $G_0 = 1$

$$X: X_{n-1} X_{n-2} \dots X_0 \text{ and } Y: Y_{n-1} Y_{n-2} \dots Y_0$$

$$(X = Y) = E_{n-1} E_{n-2} \dots E_0$$

$$(X \neq Y) = G_{n-1} + E_{n-1} G_{n-2} + \dots + E_{n-1} E_{n-2} \dots E_1 G_0$$

$$(X < Y) = L_{n-1} + E_{n-1} L_{n-2} + \dots + E_{n-1} E_{n-2} \dots E_1 L_0$$

such that E_i , G_i and L_i represent $x_i = y_i$,
 $x_i > y_i$ and $x_i < y_i$ for i th bit respectively.

Clock waveforms:

Static digital logic levels - voltage levels do not change with time.

The digital computer systems operate on the basis of dynamic digital logic levels

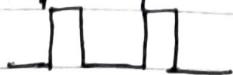
Ideal clock waveform:

→ ← clock cycle time.



Clock waveform: a series of positive or negative pulses which may be symmetrical or asymmetrical in nature.

→ ← clock cycle time.

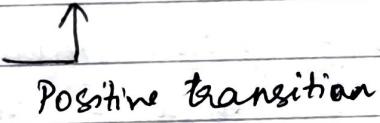


Clock cycle time - basic timing interval during which logic operations must be performed - which is equal to one period of the clock waveform.

Synchronous Operation:

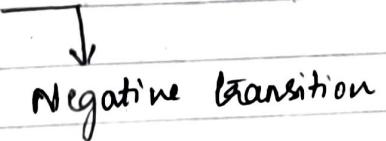
The clock transitions from low to high or from high to low state. - likewise all the circuits in a digital system changes state ~~are~~ in sync with the system clock.

A down-to-high transition - is called as positive transition - signified by the rising edge of the clock waveform.



Digital ckt which changes state during positive transition is called as positive-edge-triggered.

A high-to-low transition - is called as negative transition - signified by the falling edge of the clock waveform.



Digital ckt which changes state during negative transition is called as negative-edge-triggered.

In order to instantly change the state of the system - RESET button is used - leading to an asynchronous operation.

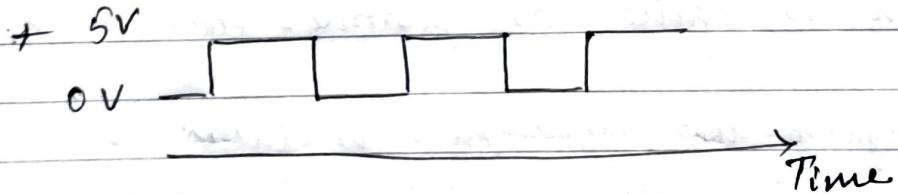
Calculate the clock cycle time for a system that uses a 500-kHz clock or 8-MHz.

$$\text{Cycle time} = \frac{1}{500 \times 10^3} = 2 \mu\text{s}$$

$$\text{Cycle time} = \frac{1}{8 \times 10^6} = 125 \text{ ns.}$$

Characteristics of clock waveform:-

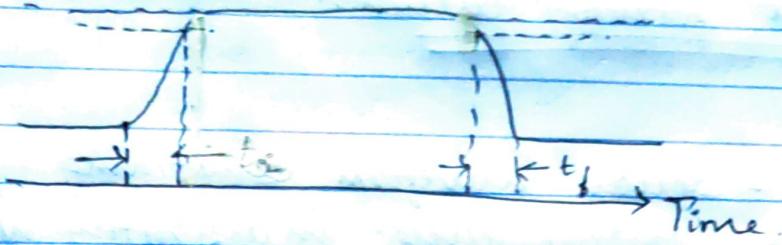
- a) - an ideal clock - has stable clock levels - ie when the clock is high - the level has a steady value of +5V & when the clock is low, value is 0V



Ideal waveform.

- b) Time:

time required for transition from one level to another (ie low to high or high to low) is zero in an ideal clock waveform.

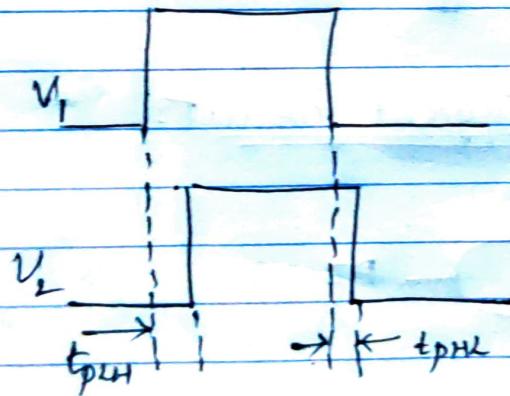


Rise time - t_r - time required for the waveform to make a transition from low to high - t_r

Fall time - t_f - time required for the waveform to make a transition from high to low - t_f

c) Ideal clock is defined by stable frequency.

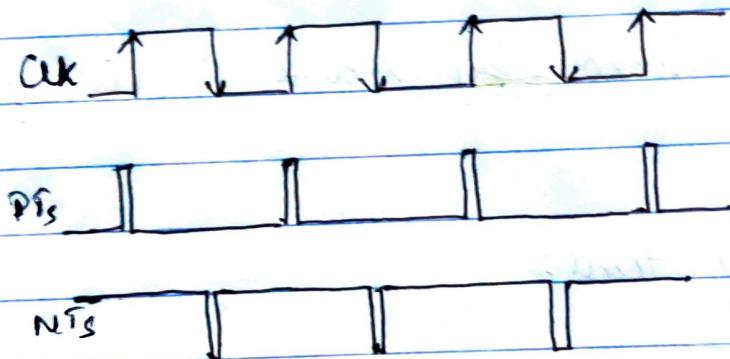
Propagation Delay time:



Pulse forming circuit :-

clock - square waveform switching from high to low voltage levels indefinitely at fixed intervals
- used to synchronize the function of each circuit element.

Pulse - A sudden change in the signal level



Positive-Edge triggered circuit:-

