



Poornima COLLEGE OF ENGINEERING

DETAILED LECTURE NOTES

Unit-4

PAGE NO.

Combinational Circuit

In combinational circuit the output of any time depends only on the input values at that time.

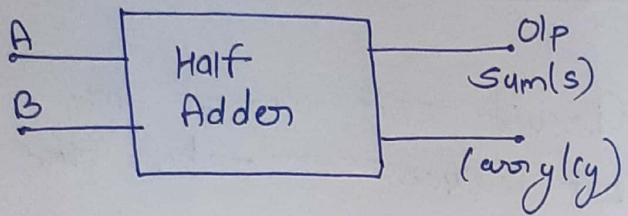
Procedure for design a combinational circuit

- identify the input and output and draw a block diagram.
- Draw a truth table
- write down the switching expression for the output.
- simplify the switching expression by algebraic or K-map method.
- implement the simplified expression using K-map.

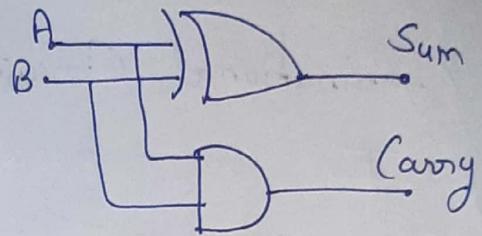
Half Adder → it is the combinational circuit

which performs addition of two binary digit called a half adder. It has two input and two output.

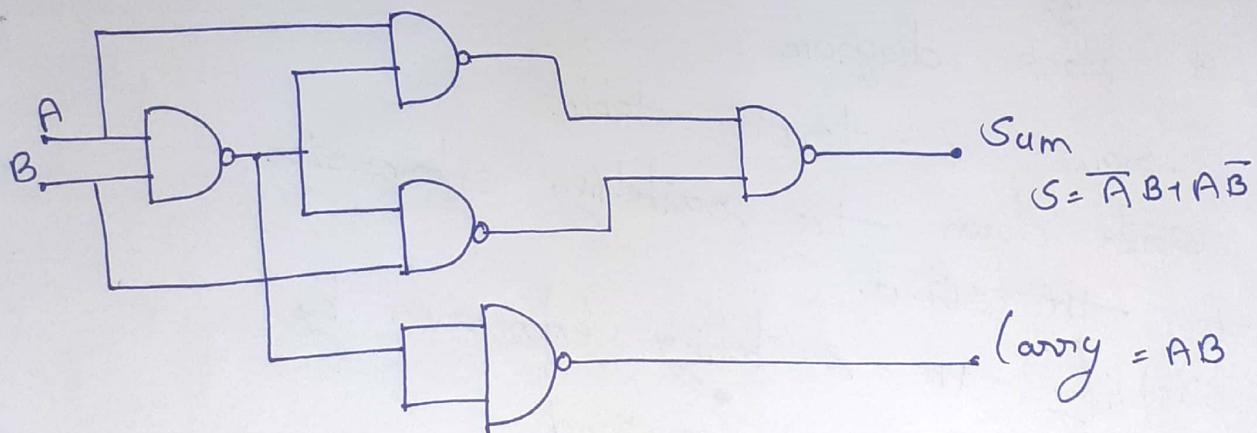
Input		Output	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



Logic Symbol



Logic Diagram.



using NAND Gate

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

$$S = A \oplus B$$

$$C = AB$$



POORNIMA

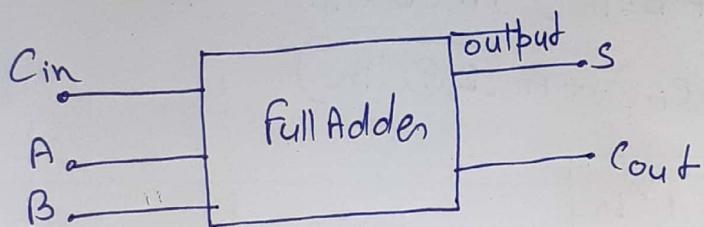
COLLEGE OF ENGINEERING

DETAILED LECTURE NOTES

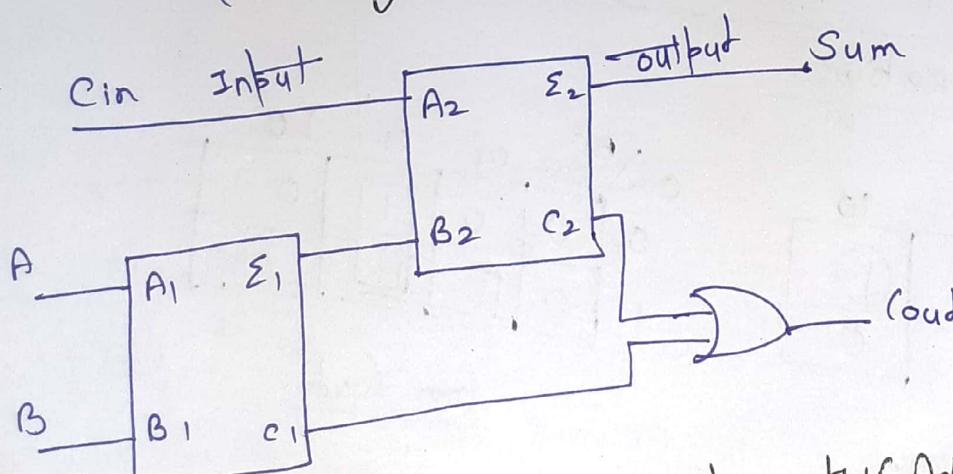
PAGE NO.

Full Adder

A half adder has two input and there is no provision to add a carry coming from the lower order bits when multiple addition is performed. A full adder is a combinational circuit that perform the arithmetic sum of three input bits and produces sum output and carry.



(a) logic Symbol



(b) symbol using two Half Adder

If consist of three input and two output
 Truth table of full adder

Input			Output	
A	B	C _{in}	S	C _{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	1
0	1	1	0	0
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$S = \overline{A}\overline{B}C_{in} + \overline{A}B\overline{C}_{in} + A\overline{B}\overline{C}_{in} + ABC_{in}$$

$$\begin{aligned} S &= \overline{A}(\overline{B}C_{in} + B\overline{C}_{in}) + A(\overline{B}\overline{C}_{in} + BC_{in}) \\ &= \overline{A}(B \oplus C_{in}) + A(\overline{B} \oplus C_{in}) \end{aligned}$$

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

$$C_{out} = AB + BC_{in} + AC_{in}$$

K-map simplification

AB		00	01	11	10
C _{in}		0	1	0	1
0	0	1	0	1	0
1	1	0	1	0	1

for Sum.

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

$$C_{out} = AB + BC_{in} + AC_{in}$$

AB		00	01	11	10
C _{in}		0	0	1	0
0	0	0	1	1	1
1	0	1	1	1	1

for C_{out}



POORNIMA

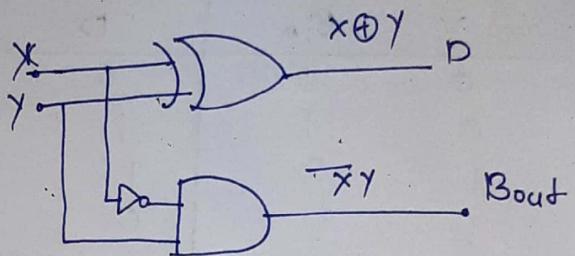
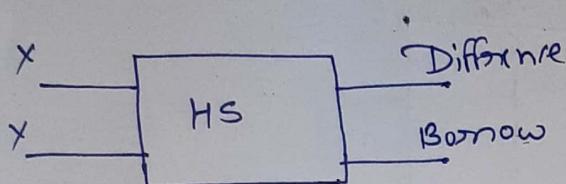
COLLEGE OF ENGINEERING

DETAILED LECTURE NOTES

Half Subtractor

PAGE NO.

It is used to perform subtraction of two bit
It has two input and two output.



(a) Logic Symbol

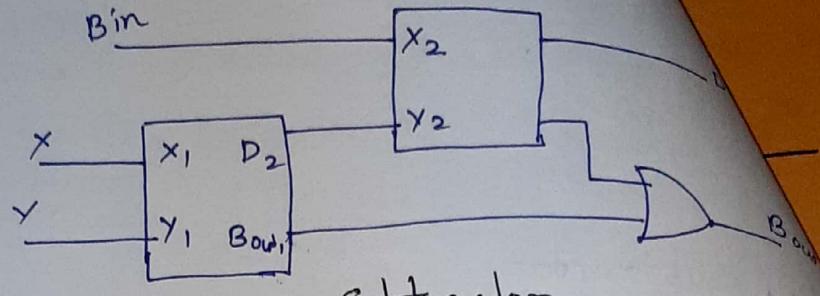
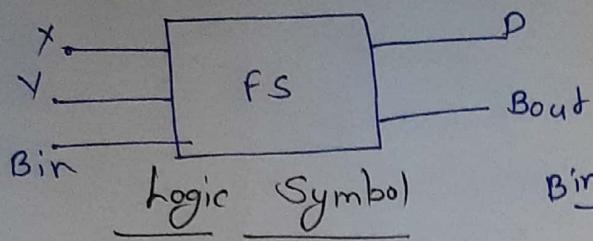
(b) Logic Diagram

Inputs		Output	
X	Y	Difference D	Borrow B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$$D = \overline{X}Y + X\overline{Y} = X \oplus Y$$

$$B = \overline{X}Y$$

Full Subtractor — it performs subtraction involving three bits — it takes borrow from the previous input.



Using half Subtraction

Input		Bin	D	Output	
X	Y			Bout	
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	

$$D = \overline{XY} \text{ Bin} + \overline{X} Y \overline{\text{Bin}} + X \overline{Y} \text{ Bin} + XY \text{ Bin}$$

$$D = (\overline{XY} + XY) \text{ Bin} + (\overline{X} Y + X \overline{Y}) \overline{\text{Bin}}$$

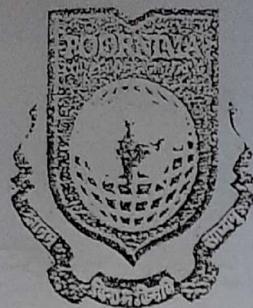
$$D = (\overline{X \oplus Y}) \text{ Bin} + (X \oplus Y) \overline{\text{Bin}}$$

$$\boxed{D = X \oplus Y + \text{Bin}}$$

K-map for Bout

Bin	xy		00	01	11	10
	0	1	0	1	0	0
0	0	1	1	0	1	0
1	1	0	0	1	1	0

$$\boxed{Bout = \overline{XY} + \overline{X} \text{ Bin} + Y \text{ Bin}}$$



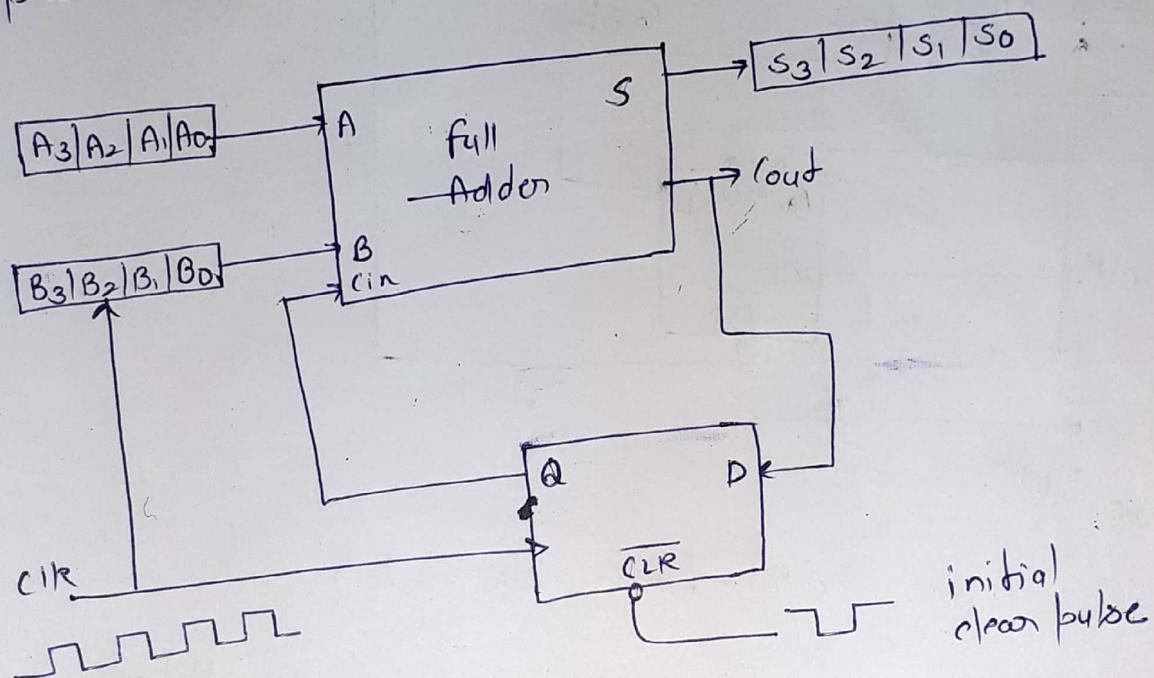
POORNIMA

COLLEGE OF ENGINEERING

DETAILED LECTURE NOTES

Serial Adder

In parallel adder the required circuitry is very large with the increased speed. In serial addition, the addition is carried out by bit by one bit. So it requires simple circuitry than the parallel adder.



4 bit Serial Adder

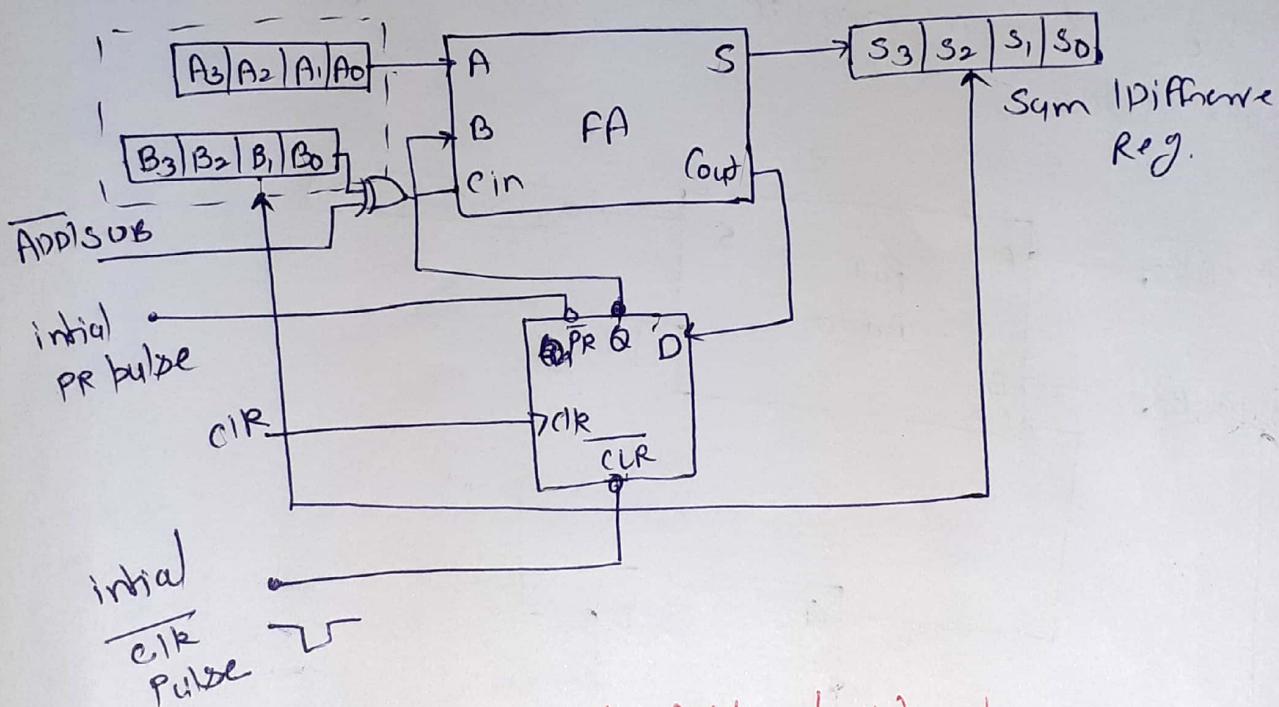
Two shift register is used to store the bit + single full adder is used to add one pair of bit at the particular time. The D flip flop is used to store the carry output of the full adder so it can be added to the

next significant bit.

The sum output of the full adder is fed it to the most significant bit of the sum register. for each clock pulse the content of the shift register shifted one pulse at the right most position.

4 bit Serial Adder Subtractor

It can be designed in the similar manner as per the actual counterpart of the parallel Adder.



4 bit serial Adder Subtractor

When $\overline{\text{ADD/SUB}} = 0$ the uncomplemented B applied to the FA, but when it is 1 the complemented value given to the FA.

The rest operation is same as per the other circuit.



DETAILED LECTURE NOTES

4 bit Carry Look Ahead Adder

Is based on the principle that lower order bits of the augend, added to the high order carry which is generated. This adder reduces the carry delay by reducing the number of gate through which a carry signal must propagate.

Row	A	B	Cin	S	Cout	
0	0	0	0	0	0	no carry generation Cout = 0
1	0	0	1	1	0	
2	0	1	0	1	0	→ carry propagation Cout = Cin
3	0	1	1	0	1	
4	1	0	0	1	0	
5	1	0	1	0	1	→ carry generation Cout = 1
6	1	1	0	0	1	
7	1	1	1	1	1	

In row 0, the carry output is always 0 and independent of carry input, in rows 6 and

7 the Cout is always one, independent of Cin.

These are known as carry generated input.

In rows 2, 3, 4, 5 the carry output is equal to Cin

From the Table

$$G_i = A_i B_i C_{in} + A_i \bar{B}_i \bar{C}_{in}$$
$$= A_i B_i (C_{in} + \bar{C}_{in})$$

$$\boxed{G_i = A_i B_i}$$

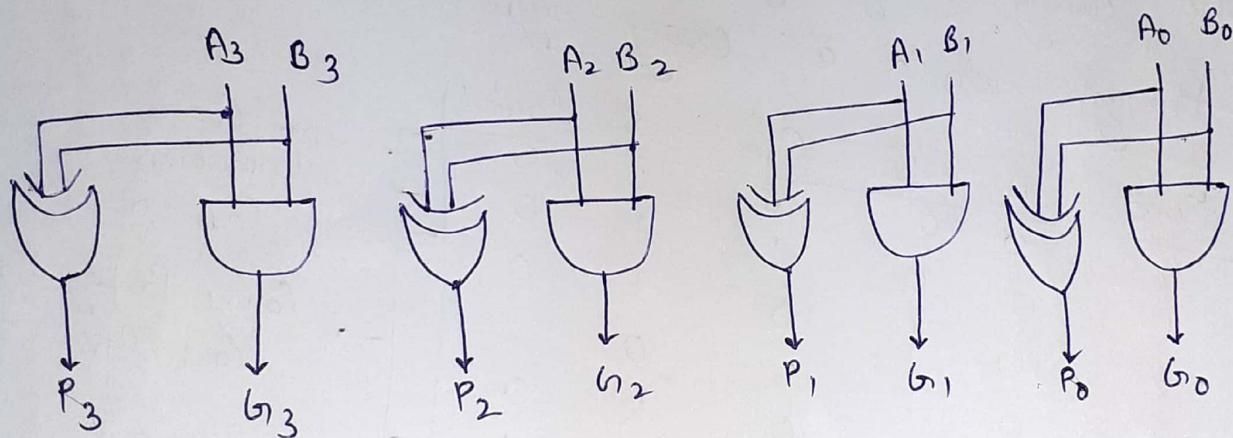
for Carry Propagation

$$P_i = A_i \bar{B}_i + B_i \bar{A}_i$$

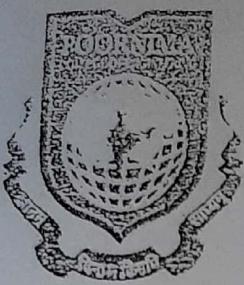
$$\boxed{P_i = A_i \oplus B_i}$$

$$P_0 = G_0 + P_0 C_{in}$$

$$C_i = G_i + P_i C_0$$



4 bit Carry Look ahead Adder.



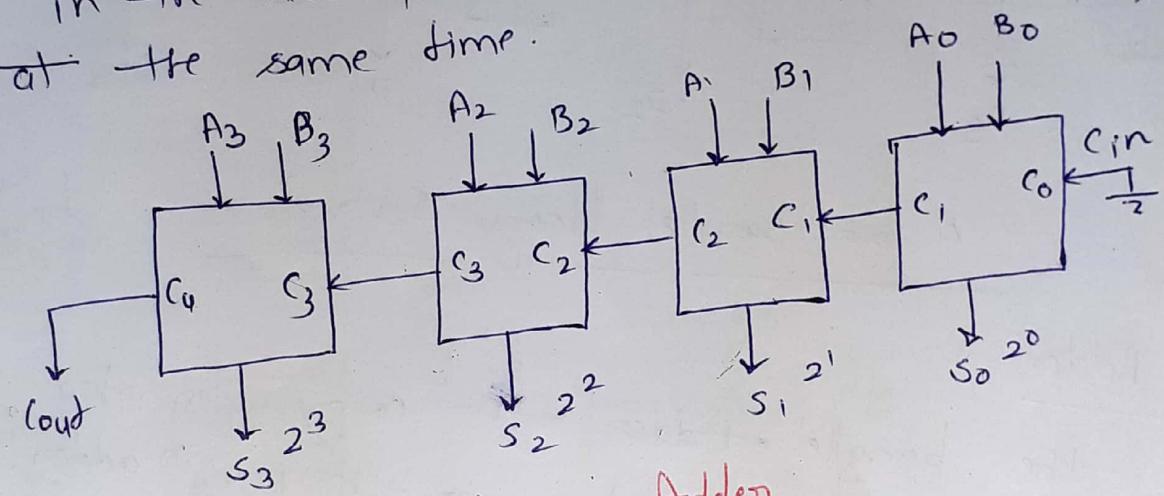
POORNIMA

COLLEGE OF ENGINEERING

DETAILED LECTURE NOTES

Parallel Binary Adder

The submission of the augend and addend are fed into the adder circuit for addition in the new position and taking the place at the same time.



4 bit binary Adder

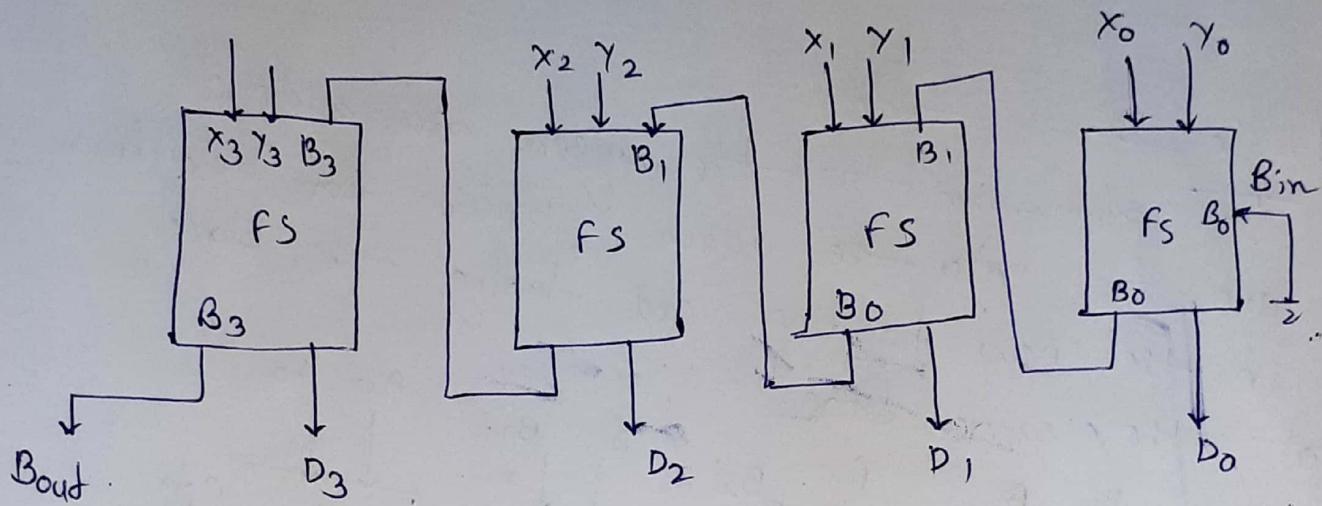
The carry input to the second stage given to the first stage. and become as the third input to that stage.

The parallel binary adder its generate output immediately when the input applied, its speed of operation is limited by the carry propagation delay through all stage. The full adder has to be generated from the previous full adder and it is the time difference between

The parallel input.

4 bit binary Parallel Subtractor

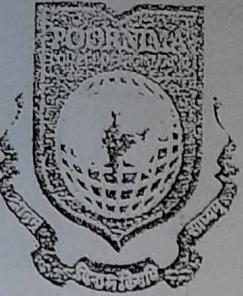
It is implemented by cascading several full adders.
A parallel binary subtraction is also implemented by cascading several full subtractors.



4 bit parallel binary Subtractor

Fast Adder

In the parallel adder the carry generated by i^{th} adder fed as carry input to the $(i+1)^{\text{th}}$ adder. The sum output only be found when carry is propagated through the intermediate adder. One of the method for eliminate the process makes the adding process is faster. It is look carry ahead addition.



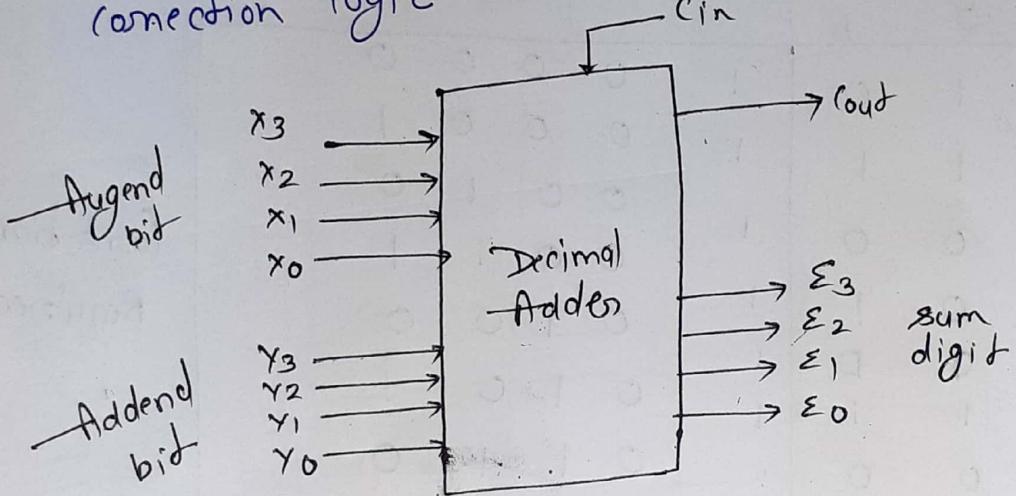
POORNIMA

COLLEGE OF ENGINEERING

DETAILED LECTURE NOTES

BCD Adder

It is the circuit that add two BCD digit in parallel and produce a sum digit which is also in BCD number. It must include the connection logic in its internal construction.



BCD Adder

It can perform the following operation.

- Add two 4 bit BCD numbers.
- If the four bit sum is equal to less than 9, the sum is in proper BCD form, no correction is needed.
- If sum is greater than 9 or if carry is generated from the sum, the sum is not in the BCD form. Then the digit 6 should be added to the sum to produce the BCD result. The carry generated given to next decimal no.

Decimal Digit	Unconnected BCD Sum					Connected BCD sum.				
	C_3	S_3	S_2	S_1	S_0	C_{out}	S_3	S_2	S_1	S_0
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	1	0
2	0	0	1	0	0	0	0	1	0	0
3	0	0	1	1	0	0	1	0	0	0
4	0	1	0	0	0	0	1	0	1	0
5	0	1	0	1	0	0	1	1	0	0
6	0	1	1	0	0	0	1	1	1	1
7	0	1	1	1	0	1	0	0	0	0
8	1	0	0	0	0	1	0	0	0	0
9	1	0	0	1	0	1	0	0	0	0
10	1	0	1	0	0	1	0	0	0	1
11	1	0	1	1	0	1	0	0	1	0
12	1	1	0	0	0	1	0	0	1	1
13	1	1	0	1	0	1	0	1	0	0
14	1	1	0	0	1	1	0	1	0	1
15	1	1	1	0	0	1	0	1	0	1
16	1	0	0	0	0	1	0	1	1	0
17	1	0	0	0	1	1	0	1	1	1
18	1	0	0	1	0	1	1	0	0	0
19	1	0	0	1	1	1	1	0	0	1

no connection
Required

connection
Required

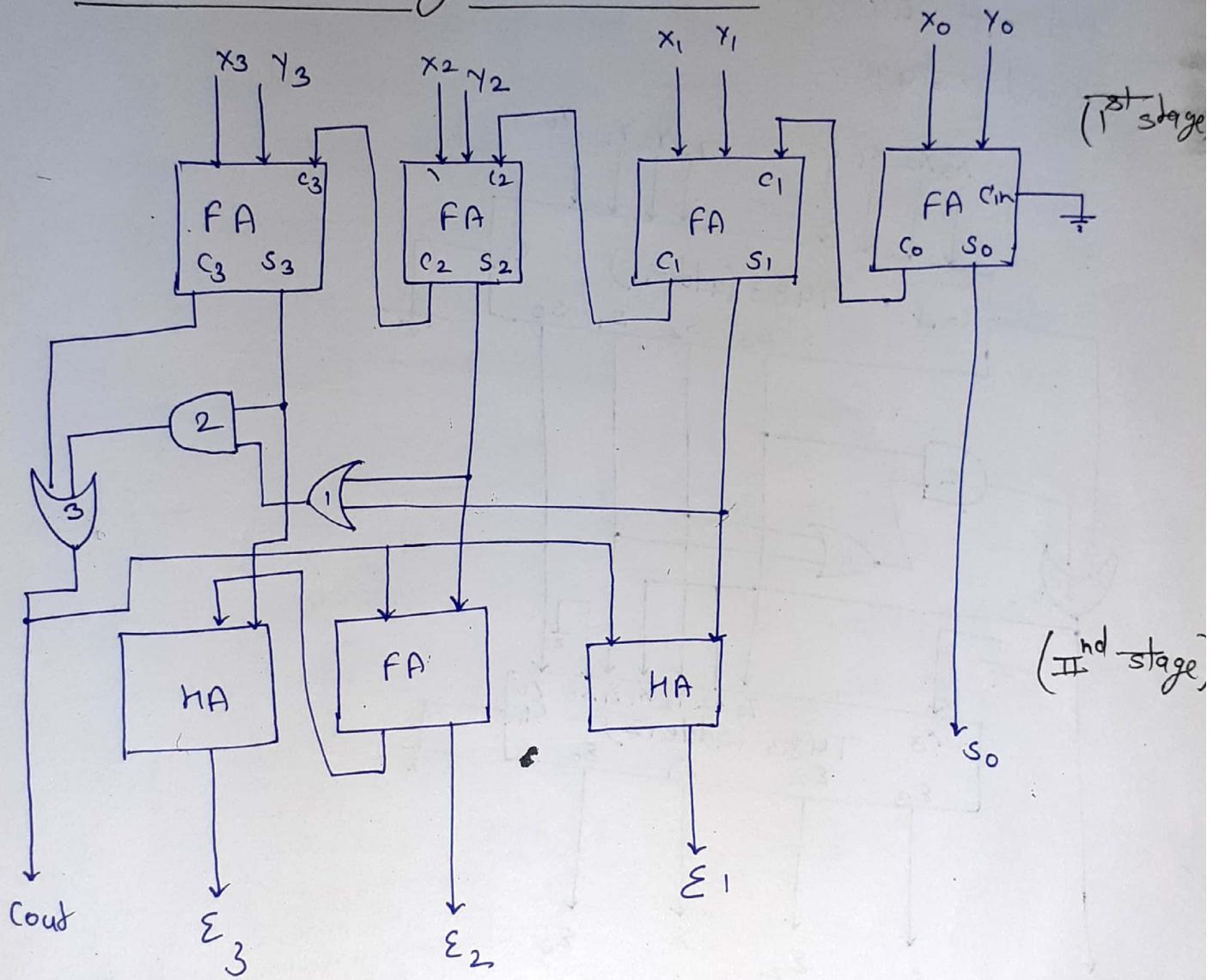


POORNIMA

COLLEGE OF ENGINEERING

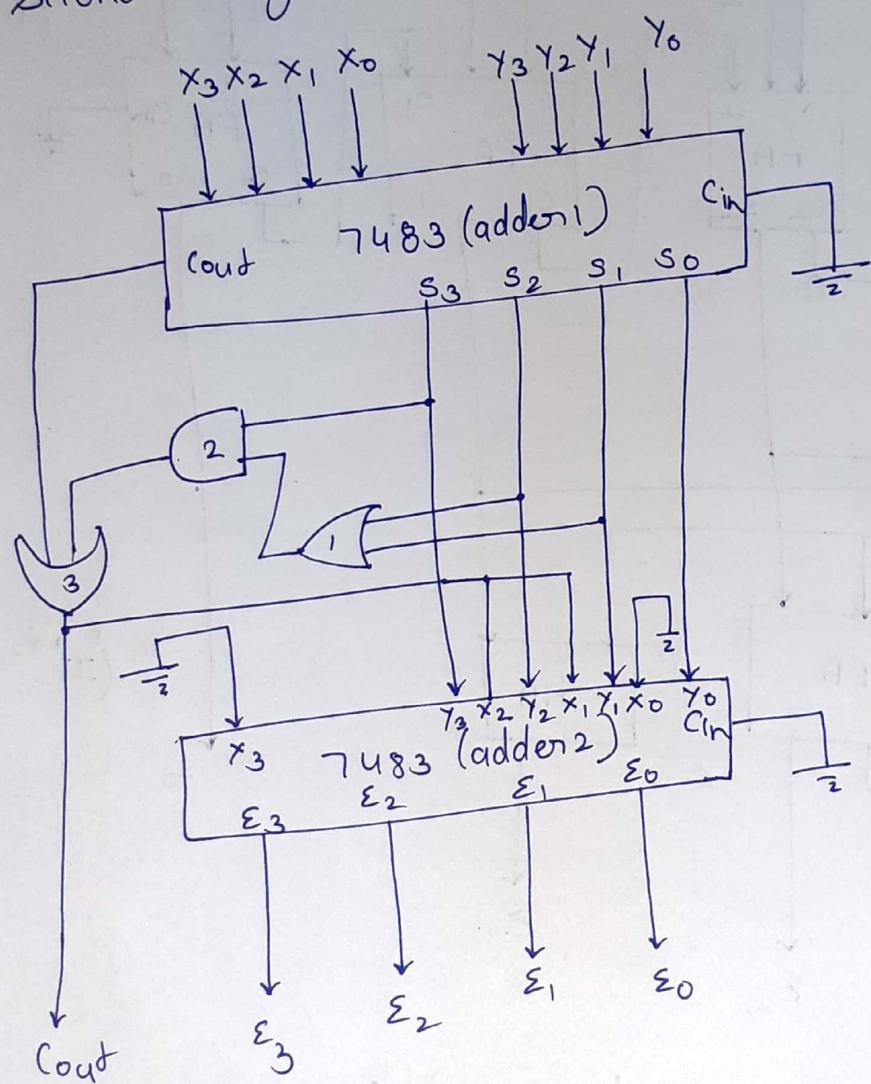
DETAILED LECTURE NOTES

BCD Adder using Full Adder

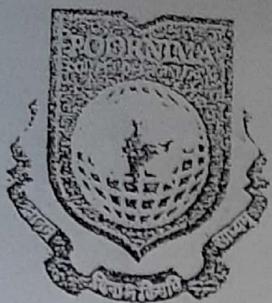


BCD Adder using Full Adder

The first stage of full adder add two 4 bit BCD numbers and its sum ($s_3 s_2 s_1 s_0$) and carry are checked to ascertain whether the result exceeds by 9 or not. If the op of the OR gate is 1, then connection is required and it accomplish by adding 0110 in the second stage.



BCD Adder using 7483



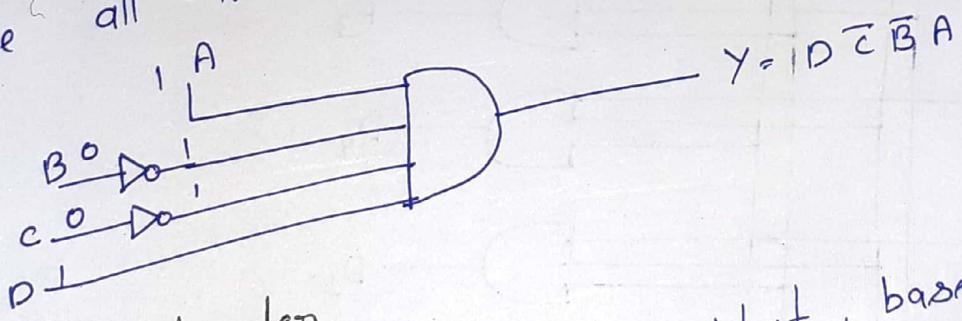
DETAILED LECTURE NOTES

Decoders

A decoder is similar to demultiplexers but without any data input. It is used in Data demultiplexing, digital to analog converter and memory addressing. It converts n bit binary input code into 2^n output lines, such that each output line can be activated only one of the possible combination.

1. Basic Binary Decoder

An AND gate can be used as the basic decoding element such as the output is high if the input binary number is 1001, than to make all the inputs to the AND gate high.



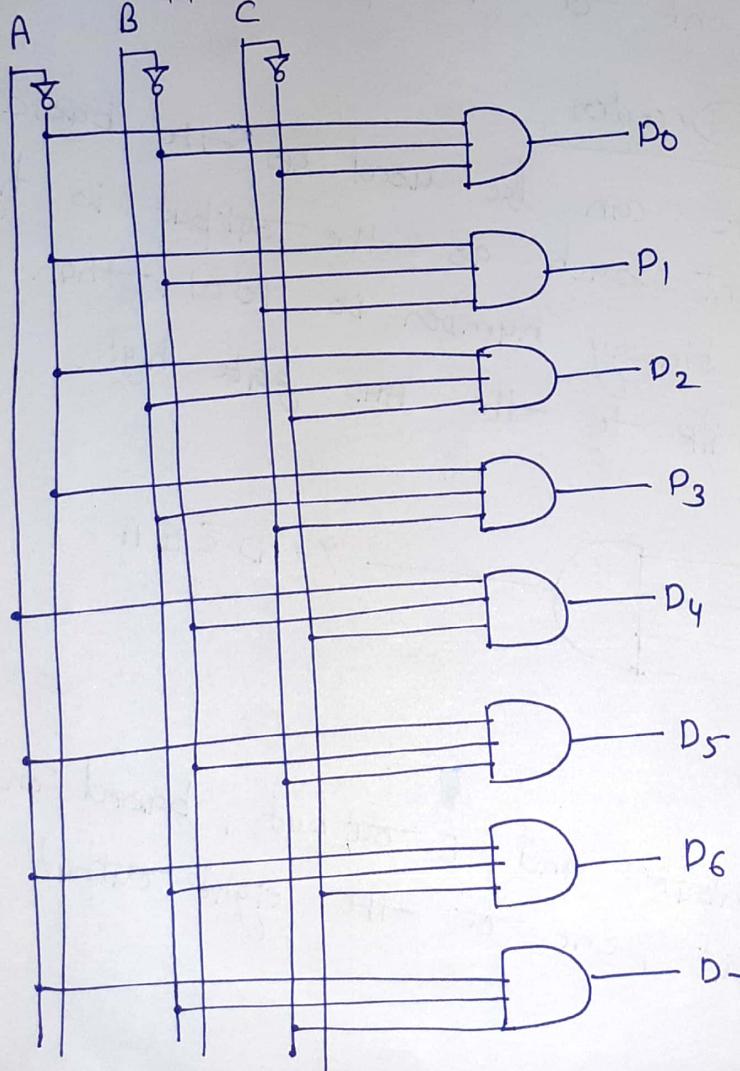
2. 3 to 8 decoder

It has 3 input and 8 output, based on the applied input, one of the eight output is selected.

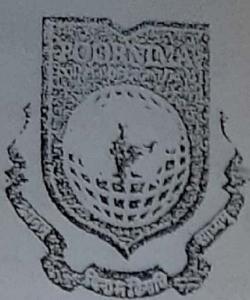
Truth table

Input			Outputs							
A	B	C	D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D 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

One of the eight output is selected as
per the applied input.



Logic Diagram of 3 to 8 decoder.



POORNIMA

COLLEGE OF ENGINEERING

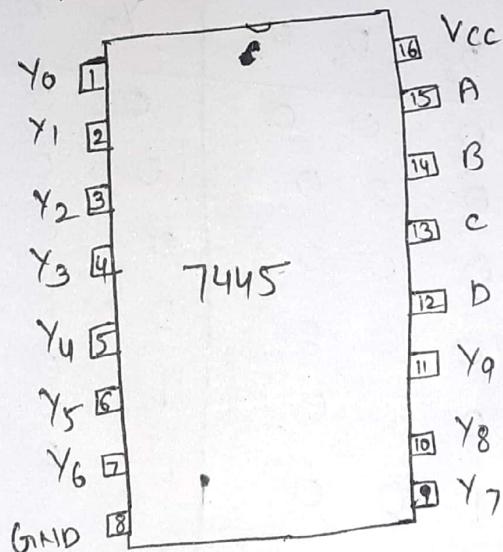
DETAILED LECTURE NOTES

BCD to decimal Decoder

A decoder takes a 4 bit BCD as the input code and produce 10 output corresponding to decimal digit is called a BCD to decimal decoder.

7445 BCD to decimal Decoder

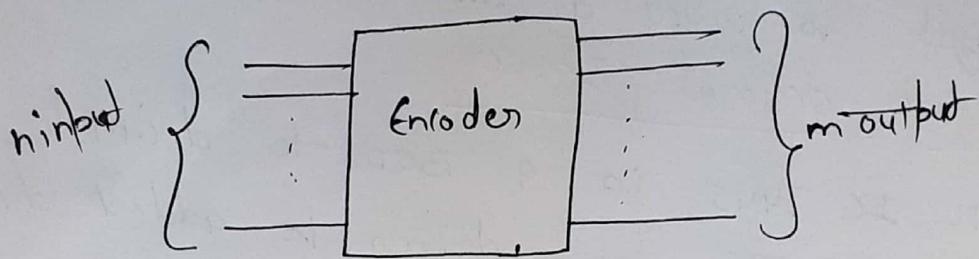
The TTL IC 7445 is a BCD to decimal decoder driver. The term driver is added due to it has open collector output that can operate at higher current. The output of 7445 can sink upto 80mA in the low state and can be raised upto 30V in the high state.



BCD to decimal Decoder

Encoders

An encoder is a digital circuit that perform the inverse operation of a decoder. Hence the opposite of decoding process is called encoding. A encoder is the combinational logic circuit that convert active input signal into a coded output signal. It has n input lines, only one of which is active at any time and m output lines.

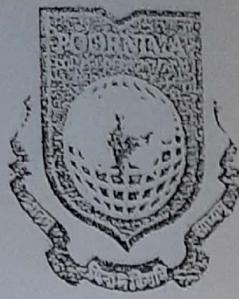


Octal to Binary Encoder

Octal to Binary Encoder accept eight input and produce a 3bit output according to the activated input.

Truth Table of Octal to Binary Encoder

D ₀	Input							Output		
	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇	Y ₂	Y ₁	Y 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	1	1	1



POORNIMA

COLLEGE OF ENGINEERING

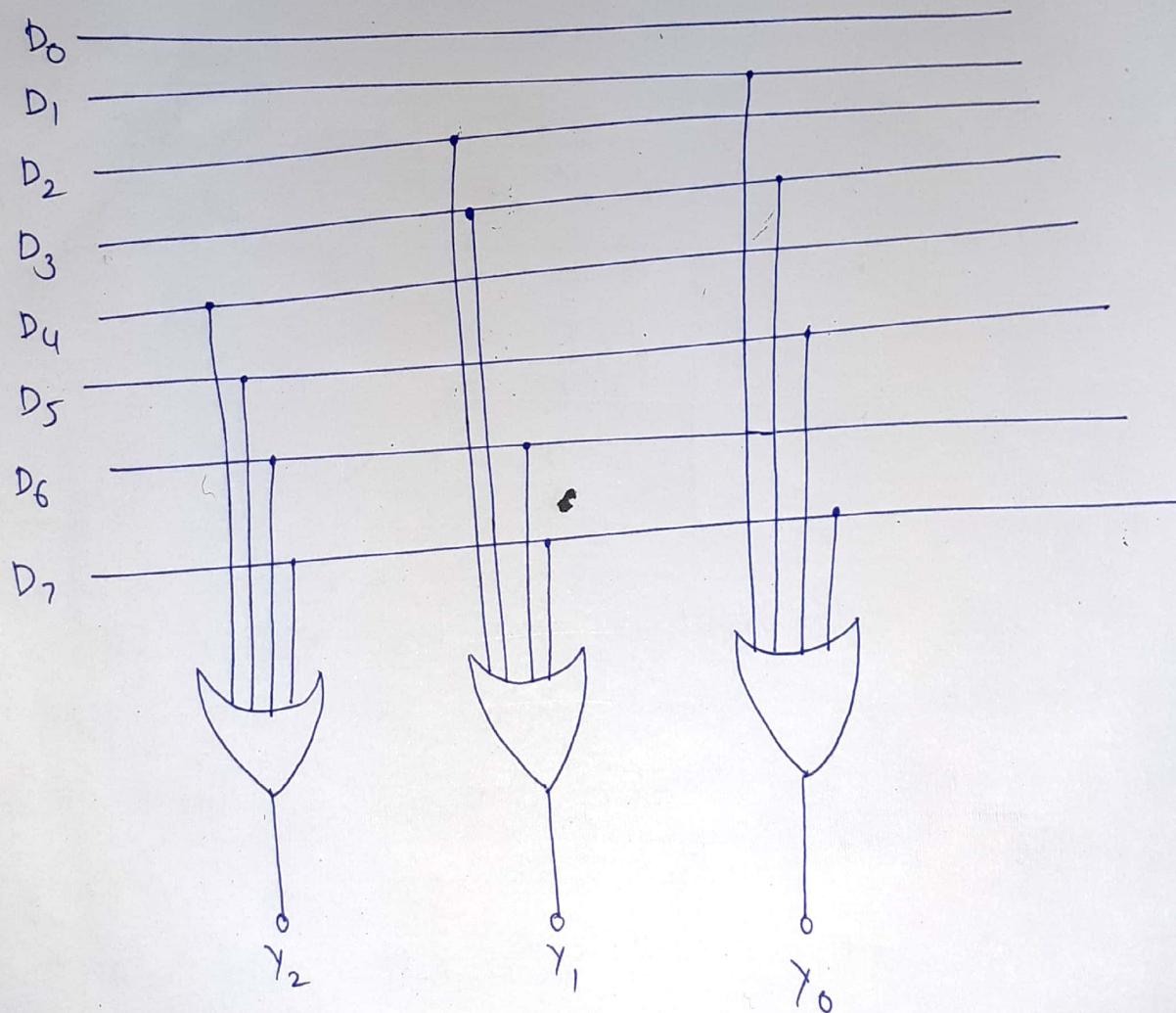
DETAILED LECTURE NOTES

$$Y_0 = D_1 + D_3 + D_5 + D_7$$

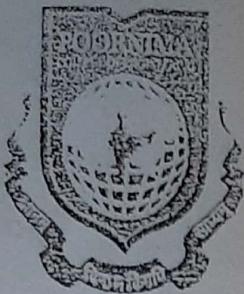
$$Y_1 = D_2 + D_3 + D_6 + D_7$$

$$Y_2 = D_4 + D_5 + D_6 + D_7$$

Using the above expression the octal to binary encoder can be implemented as



Octal to Binary Encoder



POORNIMA

COLLEGE OF ENGINEERING

DETAILED LECTURE NOTES

2. Priority Encoder:

It includes the priority function. The operation of the priority encoder is such that if two or more input are equal to 1 at the same time the input having highest priority will take precedence.

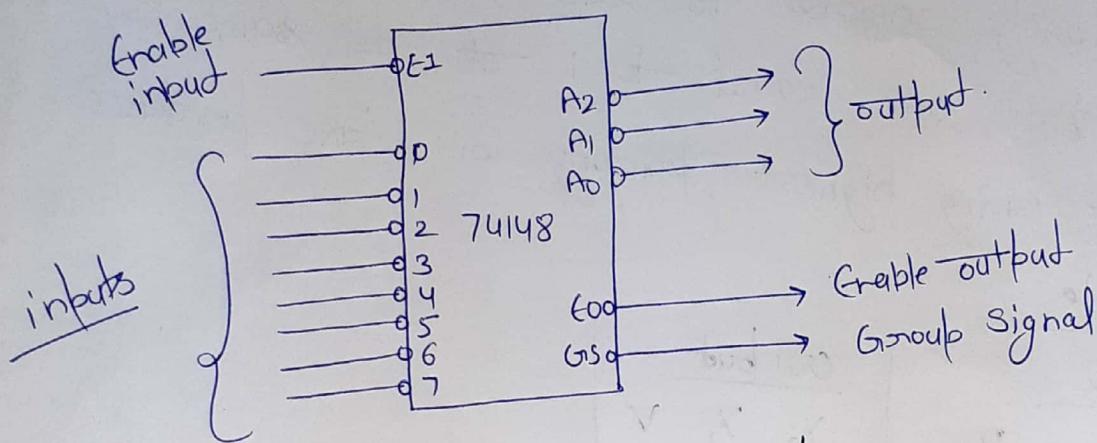
Truth table

Inputs				Output		
D_0	D_1	D_2	D_3	y_2	y_1	V
0	0	0	0	x	x	0
1	0	0	0	0	0	1
x	1	0	0	0	1	1
x	x	1	0	1	0	1
x	x	x	1	1	1	1

The x one don't care condition the design the fact that the binary value they represent must be equal to 0 or 1. Input D_3 has the highest priority, so regardless of the value of other input. When this input is 1 than $y_2 y_1$ must be 1 and so on. A valid input indicator V is set + 1 only when one or more input are equal to 1.

8 to 3 Priority Encoder

It encode 8 data line to 3 bit binary. It is also a priority encoder because it give highest priority to the highest order input. Both the data input and output is active low. In addition an enable input (EI) and enable output (EO) are provided to cascade the 74148 IC. It provide the octal expression without need of any output.



8 to 3 Priority Encoder

EI	Input								Output				
	0	1	2	3	4	5	6	7	A ₂	A ₁	A ₀	GS	EO
1	x	x	x	x	x	x	x	x	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	0	1
0	x	x	x	x	x	x	x	0	0	0	0	0	1
0	x	x	x	x	x	x	0	1	0	0	1	0	1
0	x	x	x	x	x	0	1	1	0	1	0	0	1
0	x	x	x	x	0	1	1	1	0	1	1	0	1
0	x	x	x	0	1	1	1	1	1	0	1	0	1
0	x	x	0	1	1	1	1	1	1	1	0	0	1
0	x	0	1	1	1	1	1	1	1	1	0	0	1
0	0	1	1	1	1	1	1	1	1	1	1	0	1



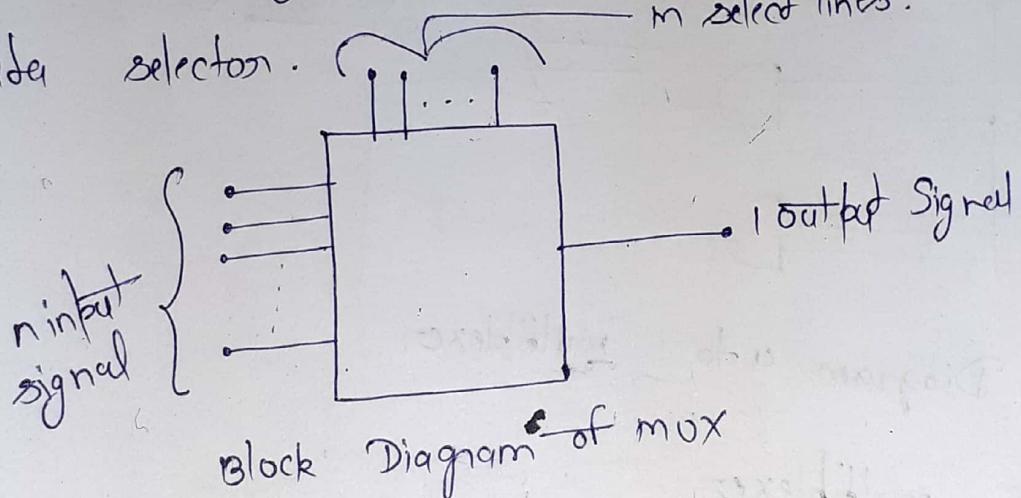
POORNIMA

COLLEGE OF ENGINEERING

DETAILED LECTURE NOTES

Multiplexer

Data Selector: It represents many to one. It is the process of transmitting the large number of information over a single line. A digital multiplexer is a combinational circuit that select one digital information from several source and transmit the selected input on the single output line. A multiplexer is also called data selector.



4 input multiplexer

It has four data input and single output lines and two select lines for selecting any one input to the output line.

$$Y = D_0 \bar{S}_1 \bar{S}_0$$

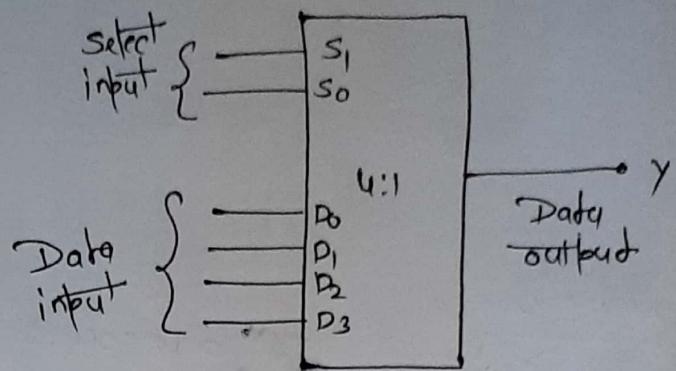
$$Y = D_1 \bar{S}_1 S_0$$

$$Y = D_2 S_1 \bar{S}_0$$

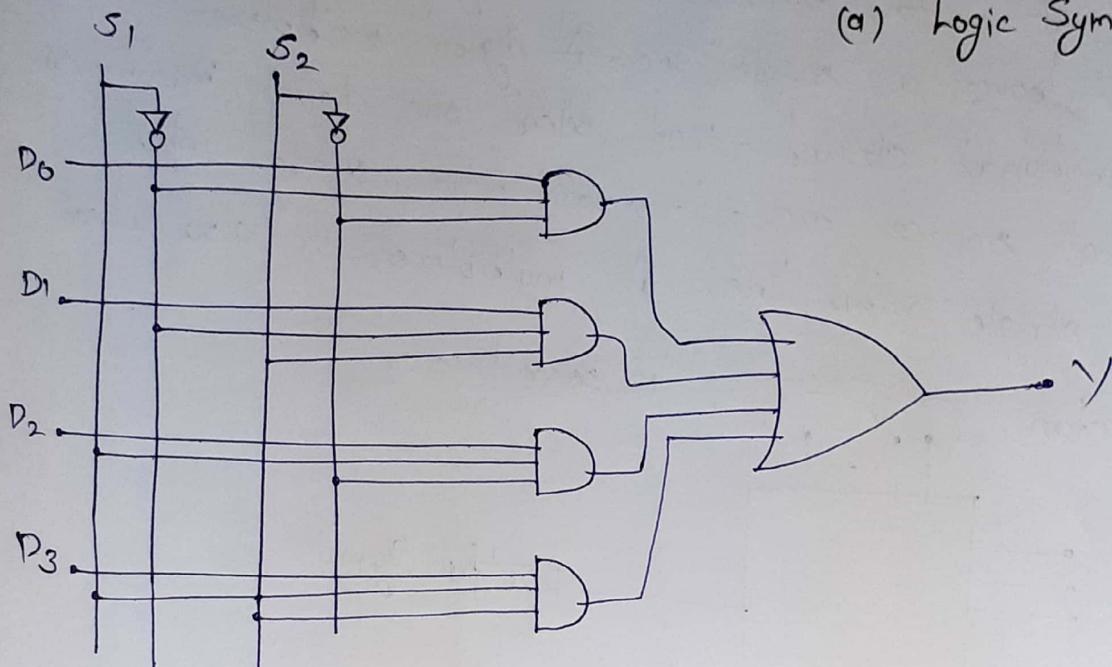
$$Y = D_3 S_1 S_0$$

$$Y = D_0 \bar{S}_1 \bar{S}_0 + D_1 \bar{S}_1 S_0 + D_2 S_1 \bar{S}_0 + D_3 S_1 S_0$$

Data Select Input		Output
S_1	S_0	y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3



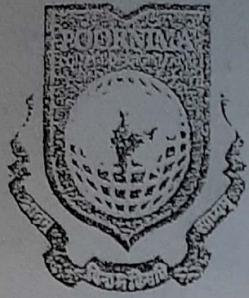
(a) Logic Symbol



Logic Diagram 4 to 1 multiplexer

Application of multiplexer

- Data Routing : it is used to route data from one source to several source
- logic function Generator → it can generate or implement logic function is SOP form directly from the truth table
- Control Sequencer It can be used as control sequencer
- Parallel to Serial converter → it process data in parallel form and take very less time.



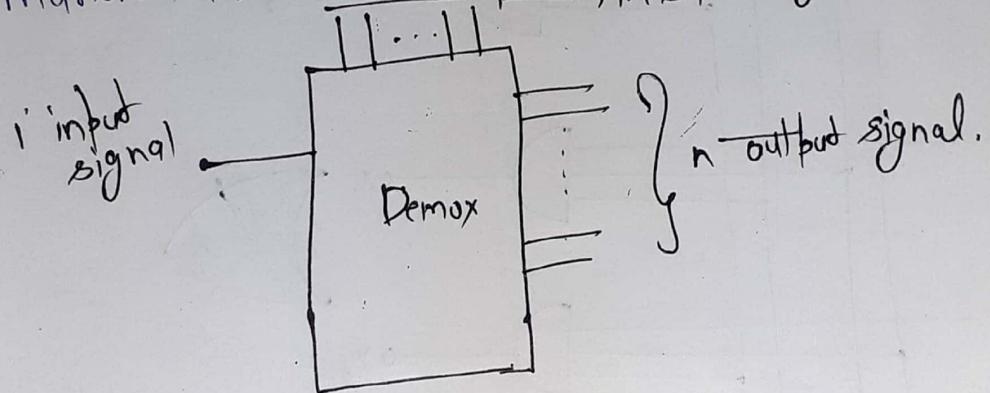
POORNIMA

COLLEGE OF ENGINEERING

DETAILED LECTURE NOTES

Demultiplexer (Data Distribution)

- It means one into many, it is the process of taking information from one input and transmit the same over several output.
- It receive information in the single input and transmit the same information to several output.
- A demultiplexer is a logic circuit that receive information and send to output in select signal.



1:4 Demultiplexer It has a single input and have four output and two select lines.

Data Input	Select Input S ₁ S ₀	Output	X ₃	X ₂	X ₁	X ₀
D	0 0	0	0	0	D	0
D	0 1	0	0	D	0	0
D	1 0	0	D	0	0	0
D	1 1	D	0	0	0	0

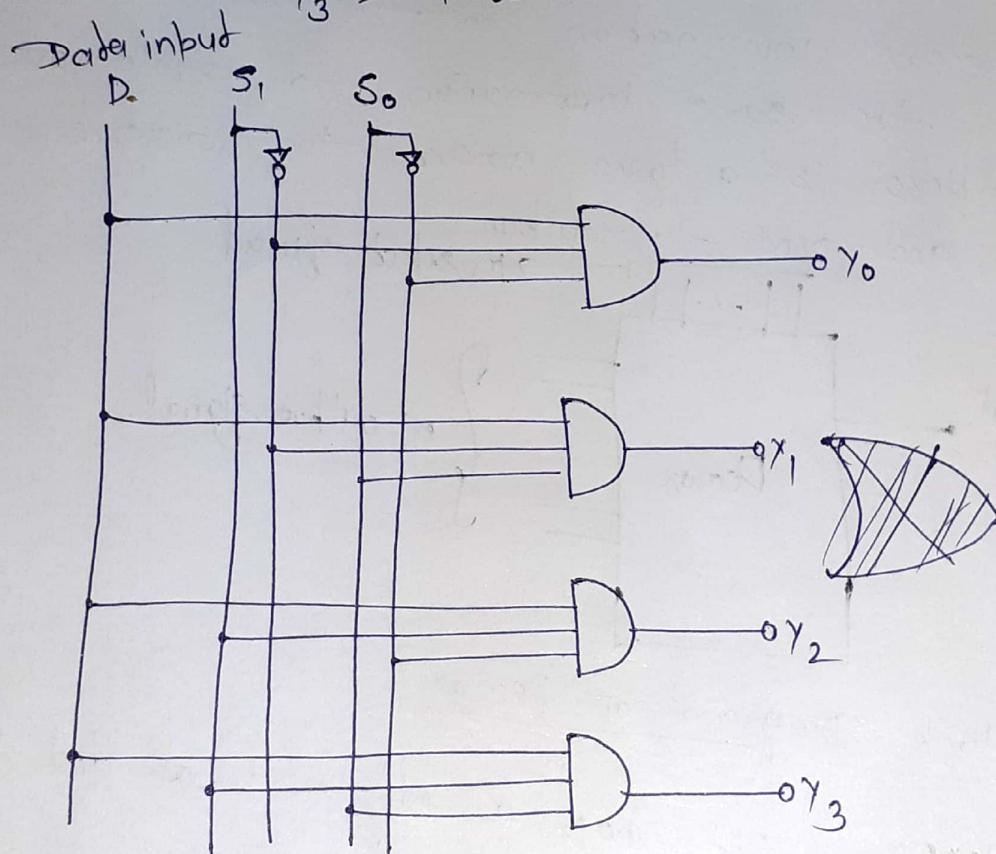
from the truth table it is cleared that the data input is connected to output y_0 when $s_1=0$ and $s_0=0$ and the data input is connected to output y_1 , when $s_1=0$ and $s_0=1$. The data input is connected to output y_2 and y_3 , when $s_1=1$ and $s_0=0$ and when $s_1=1$ and $s_0=1$ respectively.

$$y_0 = \overline{s_1} \overline{s_0} D$$

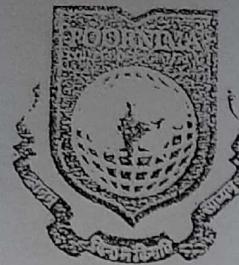
$$y_1 = \overline{s_1} s_0 D$$

$$y_2 = s_1 \overline{s_0} D$$

$$y_3 = s_1 s_0 D$$



logic Diagram of 1-to-4 demultiplexer



POORNIMA

COLLEGE OF ENGINEERING

DETAILED LECTURE NOTES

I to 8 Demultiplexer

It has 3 input and eight output (y_0 to y_7) and three select input (s_2, s_1, s_0). It distribute one input line to eight output line based on the select input.

Data Input D	Select Input $s_2 \ s_1 \ s_0$			Output							
	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	0	D
0	0	0	1	0	0	0	0	0	0	D	0
0	0	1	0	0	0	0	0	0	D	0	0
0	1	0	0	0	0	0	0	0	D	0	0
0	1	0	1	0	0	0	0	D	0	0	0
0	1	1	0	0	0	0	D	0	0	0	0
1	0	0	0	0	0	0	D	0	0	0	0
1	0	0	1	0	0	D	0	0	0	0	0
1	1	0	0	0	D	0	0	0	0	0	0
1	1	1	1	D	0	0	0	0	0	0	0

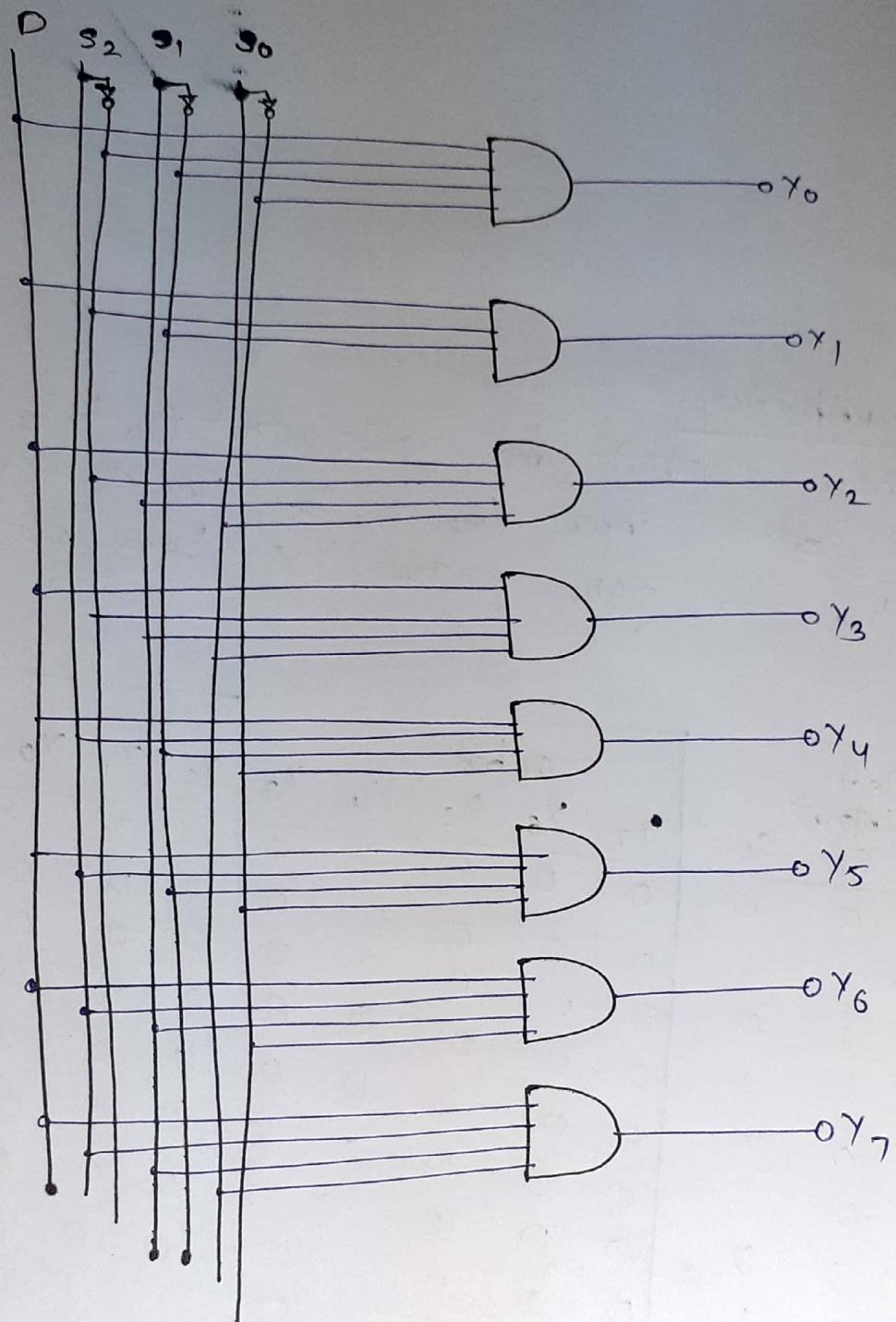
Truth table of 8:1 multiplexer

$$y_0 = \overline{s_2} \overline{s_1} \overline{s_0}$$

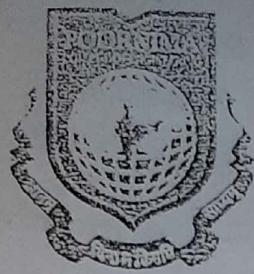
$$y_4 = s_2 \overline{s_1} \overline{s_0}$$

$$y_1 = \overline{s_2} \overline{s_1} s_0 \quad y_2 = \overline{s_2} s_1 \overline{s_0} \quad y_3 = \overline{s_2} s_1 s_0 D$$

$$y_5 = s_2 \overline{s_1} s_0 \quad y_6 = s_2 s_1 \overline{s_0} \quad y_7 = s_2 s_1 s_0 D$$



Logic Diagram of 8:8 demultiplexer.

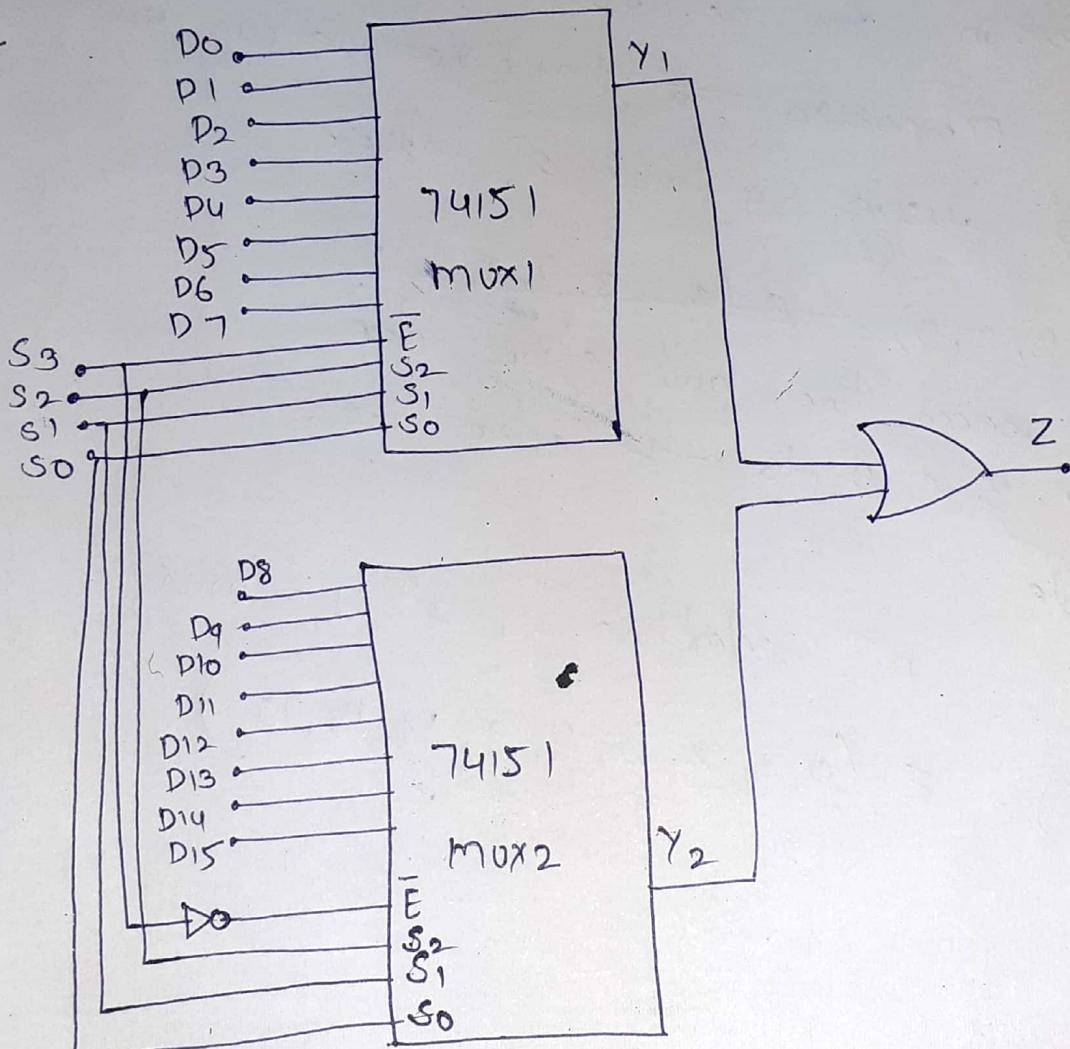


DETAILED LECTURE NOTES

Implementation of higher Order multiplexers

- Q. implement a 16-to-1 multiplexer using two 8-to-1 multiplexers IC (74151)

Ans



16-to-1 multiplexer using two 74151

Implementation of Boolean Expression using multiplexers

Any boolean expression can be implemented using a mux. If a boolean expression has $(n+1)$ variable than n of these variable can be connected to the select line of the multiplexer. The remaining single variable along with constant 1 and 0 is used as the input of the mux.

If A is the single variable than the input of the mux A, \bar{A} , 1 and 0. By this method any expression can be implemented. In general a boolean expression of $(n+1)$ variable can be implemented using a mux 2^n input.

Example $f(A, B, C, D) = \Sigma(0, 1, 3, 4, 8, 9, 15)$

it has 3 select line and eight input
Apply variable B, C, D to the select line.
for n variable $n-1 = \text{select line} ; 2^{n-1}:1 = \text{mux}$

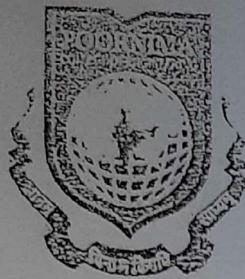
$n = \text{Variable}$

$$f(A, B, C, D) = \Sigma(0, 1, 2, 3, 4, 9, 13, 14, 15)$$

$$n = 4$$

$$\text{selection line} = 3$$

$$2^3:1 = 8:1$$



POORNIMA

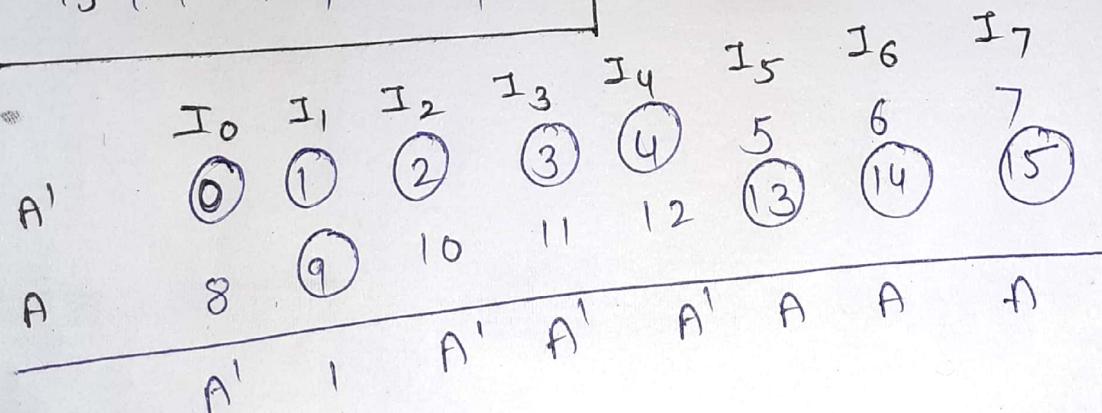
COLLEGE OF ENGINEERING

DETAILED LECTURE NOTES

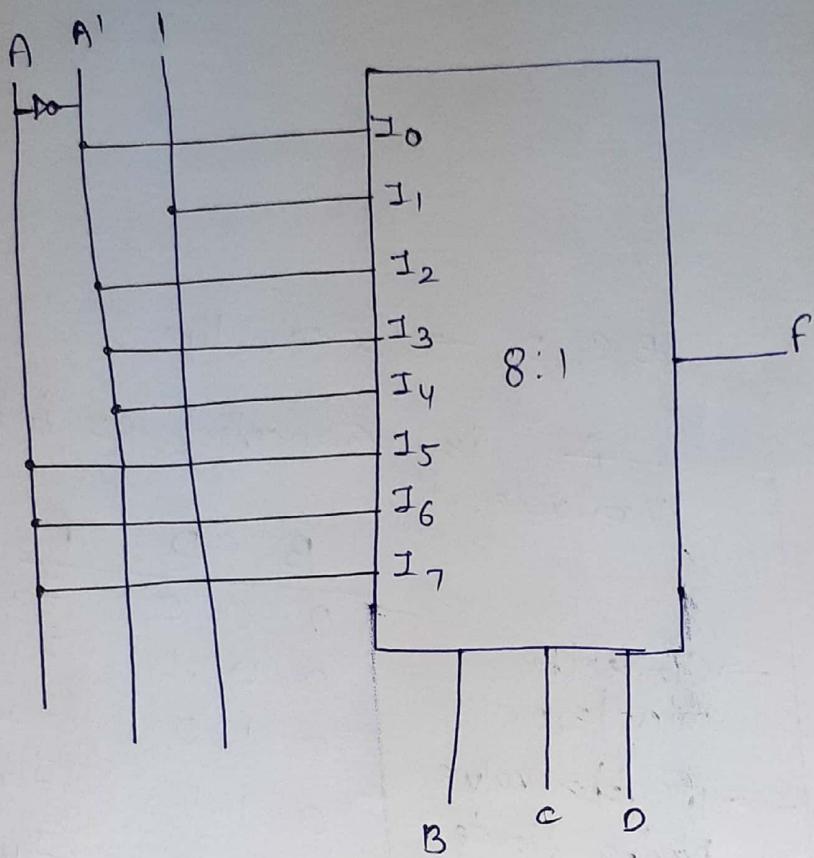
	A	B	C	D	F
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	1

$B, C, D \rightarrow$ select line.
 $\neg A$ is connected as input variable.
 → Cinchette minterm
 → if upper value having circle than A
 → if lower value having circle than A
 → if both than
 → if none of them, 0

B	C	D	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



So as per the equation



8:1 multiplexer for the given boolean
equation