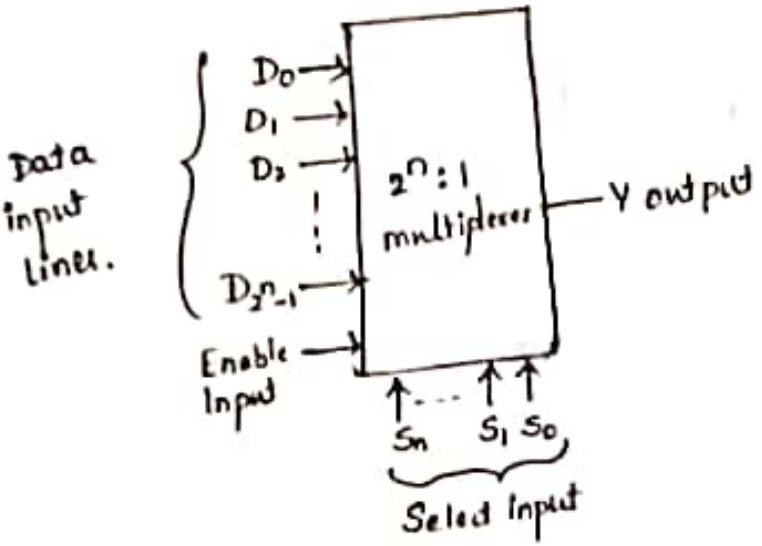
Digital Multiplexers

→ Multiplexers are also called data selector. The basic function of this device is to select one of its 2^n data input lines and place the corresponding information appearing on this line onto a single output line.

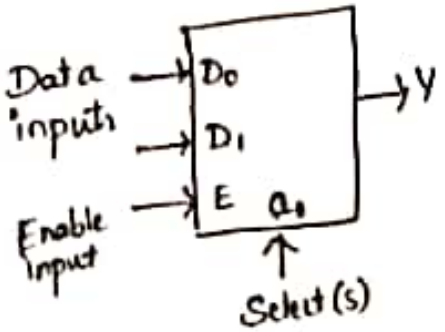
→ Normally there are 2^n input lines and n selection lines whose bit combinations determine which input is selected.

→ Therefore, multiplexer is 'many into one' and it provides the digital equivalent of an analog selector switch.



$2^n:1$ Multiplexer

2:1 multiplexer

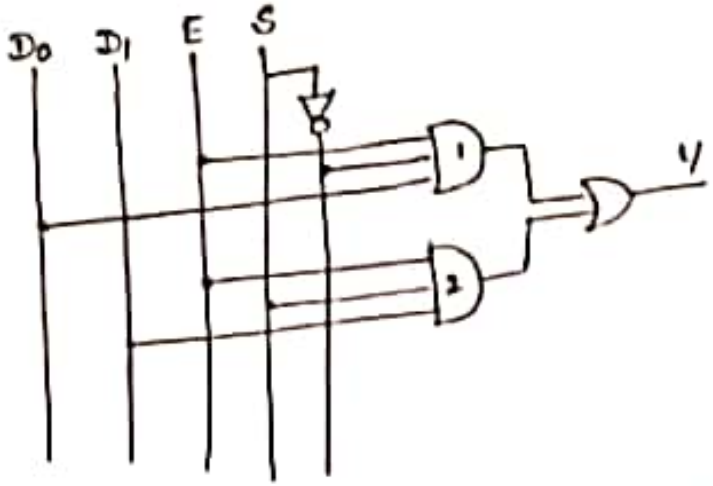


E	a_0	D_1	D_0	Y	
1	0	x	0	0	
1	0	x	1	1	$E\bar{S}D_0$
1	1	0	x	0	
1	1	1	x	1	$ES D_1$
0	x	x	x	0	

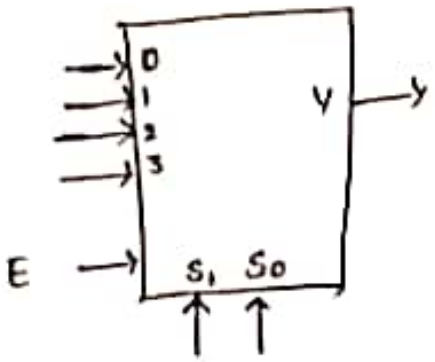
→ when E = 0, irrespective of other inputs Y = 0, when E = 1 the data of the addressed or selected input appears at the input.

∴ From truth table

$Y = E\bar{S}D_0 + ESD_1$

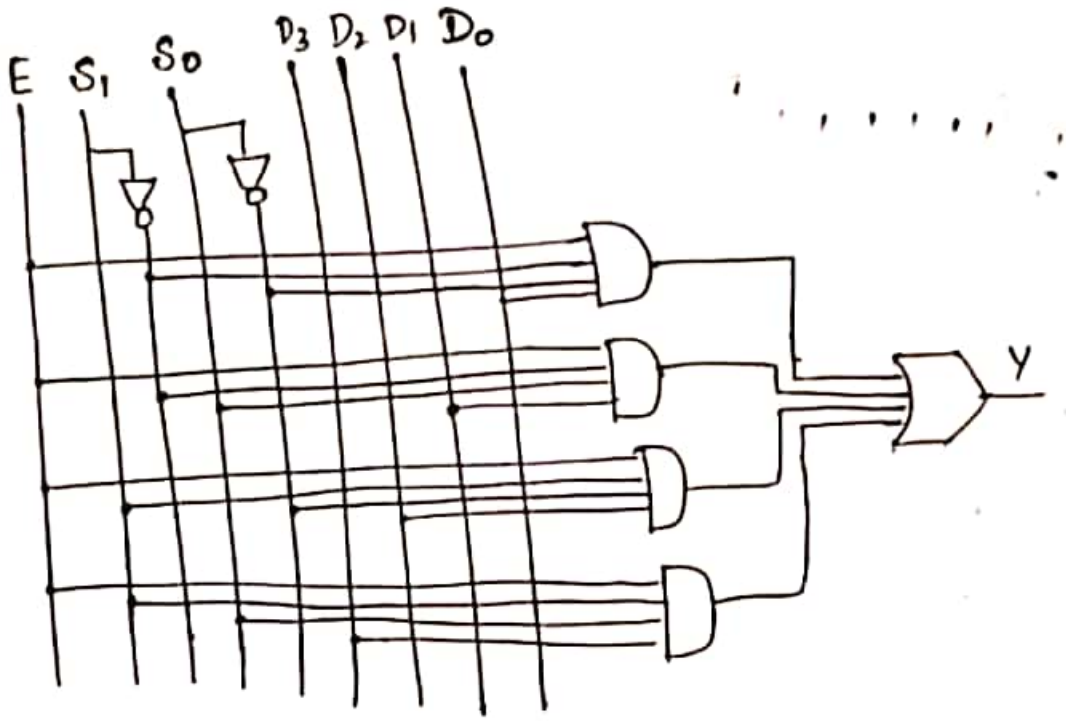


4:1 Mux

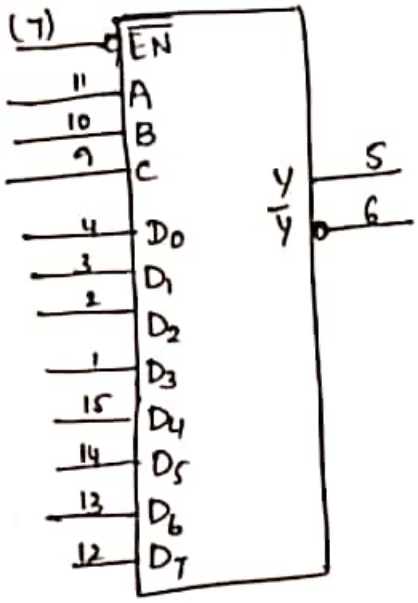


E	S_1	S_0	Y
0	x	x	0
1	0	0	D_0
1	0	1	D_1
1	1	0	D_2
1	1	1	D_3

Enable	Select		Input				output	
E	S_1	S_0	D_3	D_2	D_1	D_0	Y	
0	x	x	x	x	x	x	0	
1	0	0	x	x	x	0	0	
1	0	0	x	x	x	1	1	$E\bar{S}_0\bar{S}_1D_0$
1	0	1	x	x	0	x	0	
1	0	1	x	x	1	x	1	$E\bar{S}_1S_0D_1$
1	1	0	x	0	x	x	0	
1	1	0	x	1	x	x	1	$ES_1\bar{S}_0D_2$
1	1	1	0	x	x	x	0	
1	1	1	1	x	x	x	1	$ES_1S_0D_3$

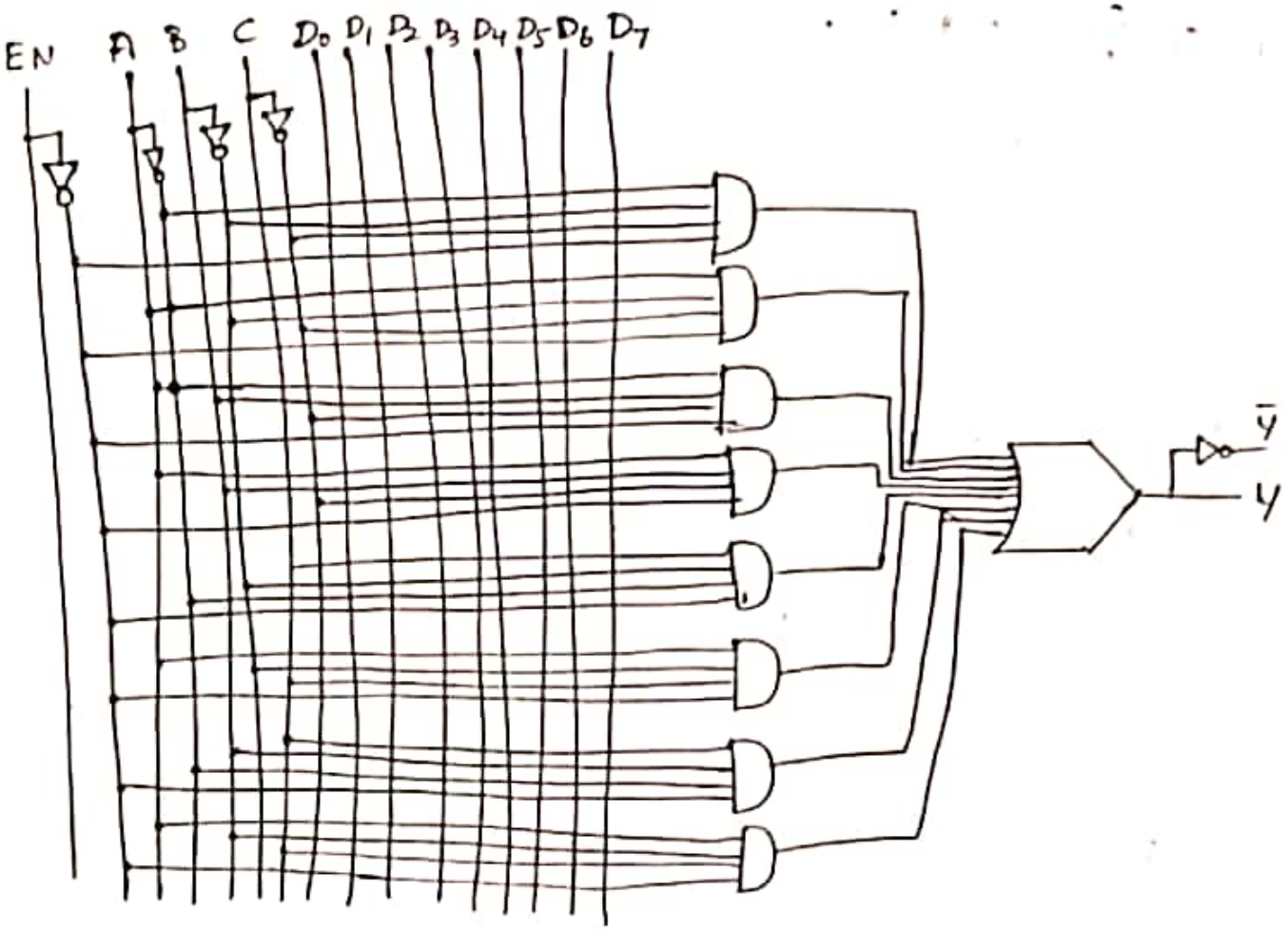


74151 : 8 to 1 Multiplexer (Mux) [74151]

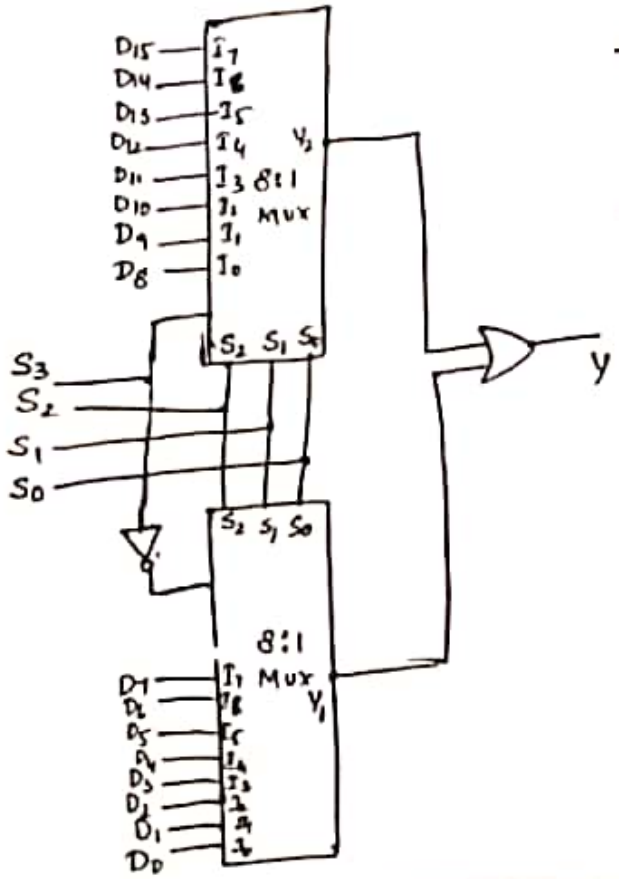


Enable $\overline{EN}$	Select C B A	Y	$\overline{Y}$
1	X X X	0	1
0	0 0 0	D_0	$\overline{D_0}$
0	0 0 1	D_1	$\overline{D_1}$
0	0 1 0	D_2	$\overline{D_2}$
0	0 1 1	D_3	$\overline{D_3}$
0	1 0 0	D_4	$\overline{D_4}$
0	1 0 1	D_5	$\overline{D_5}$
0	1 1 0	D_6	$\overline{D_6}$
0	1 1 1	D_7	$\overline{D_7}$

→ The 74151 is a 8 to 1 mux. It has eight inputs. It provides two output, one is active high the other is active low.



1) Design 16:1 MUX using 8:1 mux

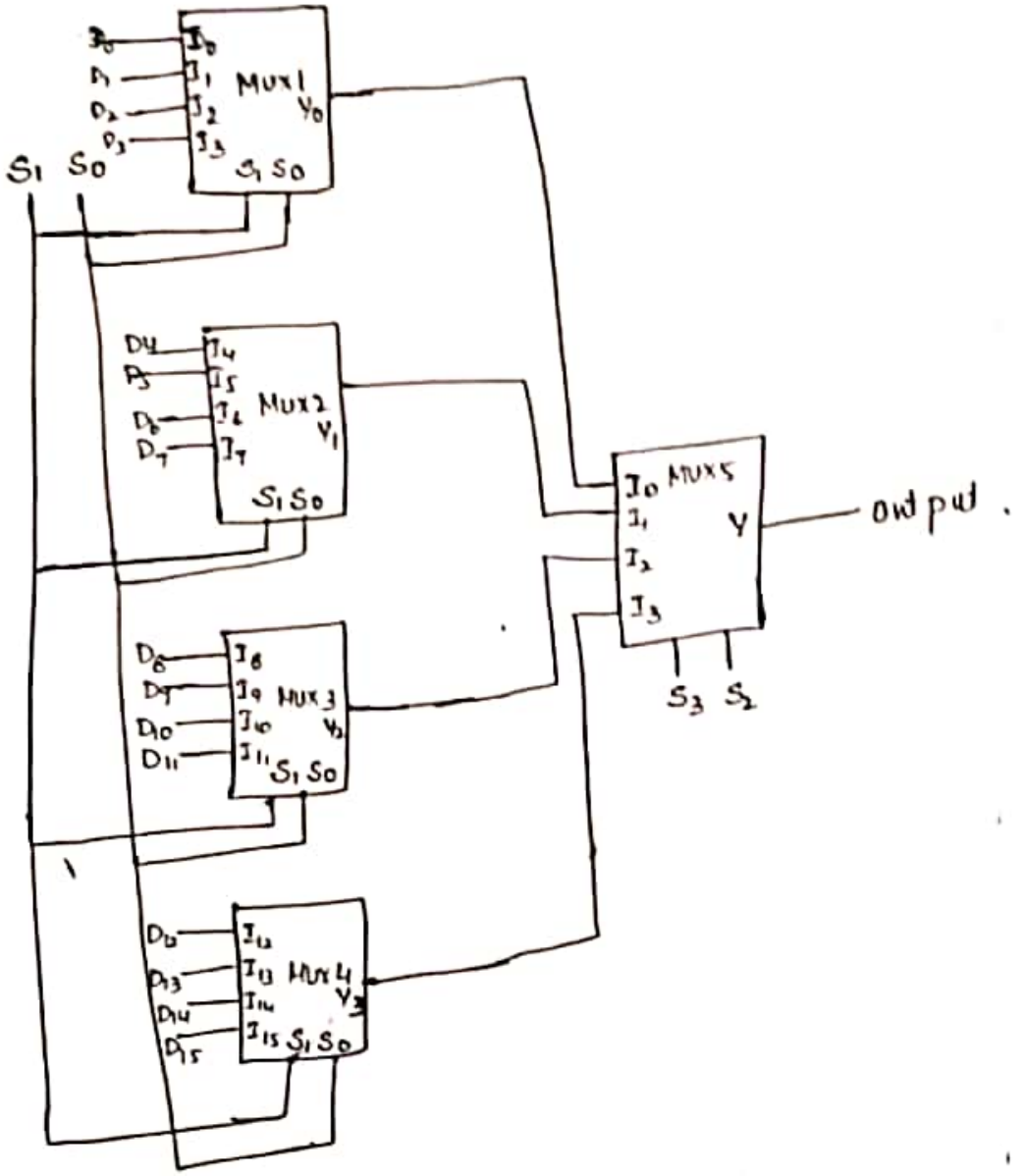


→ Connect the select line (S_2, S_1 & S_0) of two mux in parallel

→ Connect most significant select line (S_3) such that when $S_3=0$ MUX 1 is enabled and when $S_3=1$, MUX 2 is enabled.

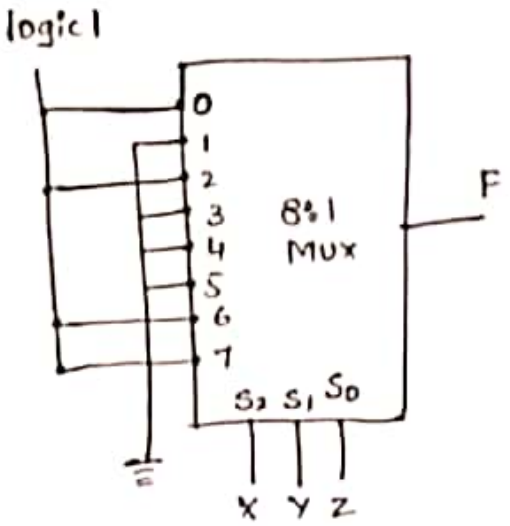
→ logically OR the outputs of two multiplexers to obtain the final output Y.

a) Design 16:1 MUX Using 4:1 MUX



1) Implement the given function using multiplexer

$F(x, y, z) = \Sigma(0, 2, 6, 7)$



→ Connect input corresponds to the minterm to logic 1

→ Connect remaining input to logic 0

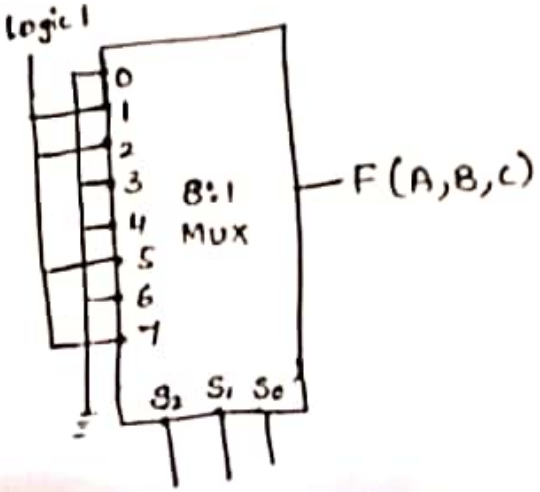
→ Connect input variables to select line of MUX

2) Implement the Boolean function represented by the given truthtable using Multiplexer

A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Soln:- $2^3 = 8:1$ MUX

$\therefore Y = \Sigma m(1, 2, 5, 7)$

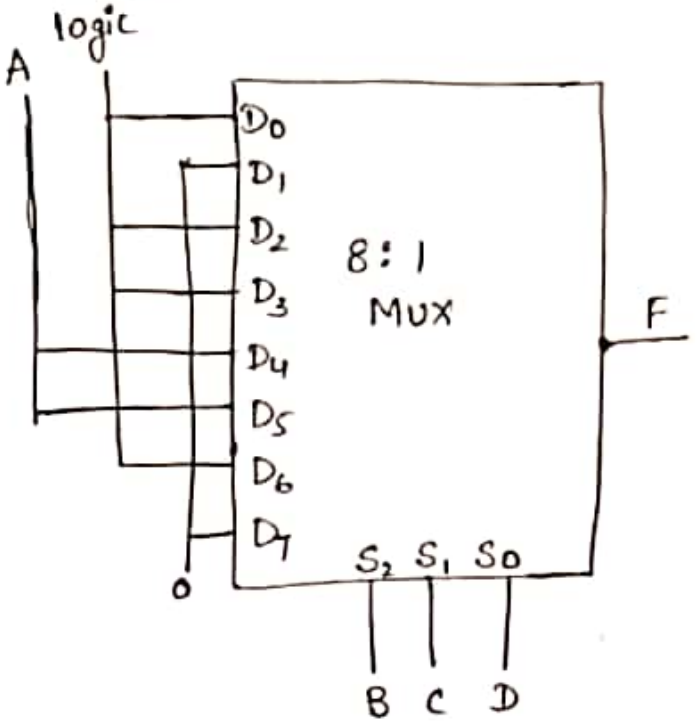


3) Implement the following Boolean fun using 8:1 MUX

Soln:- $F(A,B,C,D) = \Sigma m(0,2,6,10,11,12,13) + d(3,8,14)$

	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
$\bar{A}$	(0)	1	(2)	(3)	4	5	(6)	7
A	(8)	9	(10)	(11)	(12)	(13)	(14)	15
	1	0	1	1	A	A	1	0

Here, don't Care are treated as 1



Select B C D as Select lines

4) Implement $f(a,b,c,d) = \Sigma m(0,1,5,6,7,9,10,15)$ using

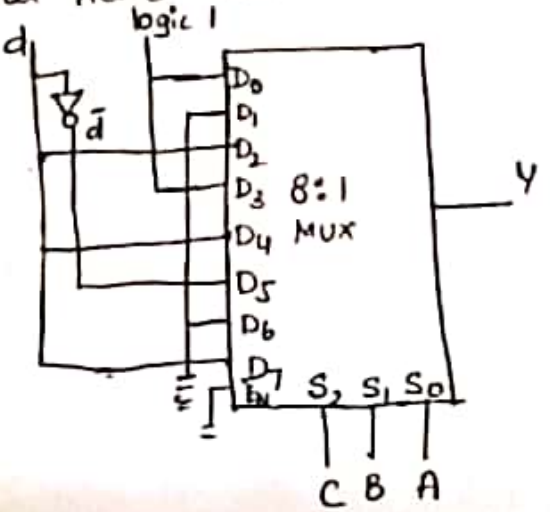
i) 8:1 MUX with a, b, c as select lines

ii) 4:1 MUX with a, b, c as select lines

d a b c d

Soln:-

	$\bar{d}$	d	
D_0	(0)	(1)	1
D_1	2	3	0
D_2	4	(5)	d
D_3	(6)	(7)	1
D_4	8	(9)	$\bar{d}$
D_5	(10)	11	$\bar{d}$
D_6	12	13	0
D_7	14	(15)	d



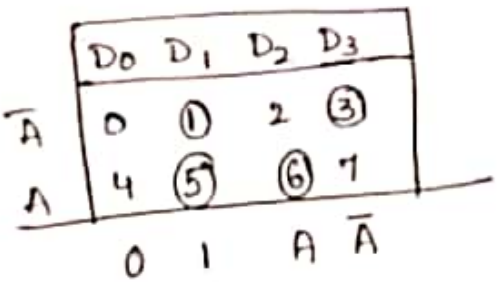
3) Implement the following Boolean function using 4:1 MUX

$F(A,B,C,D) = \Sigma m(1,3,5,6)$

Soln:-

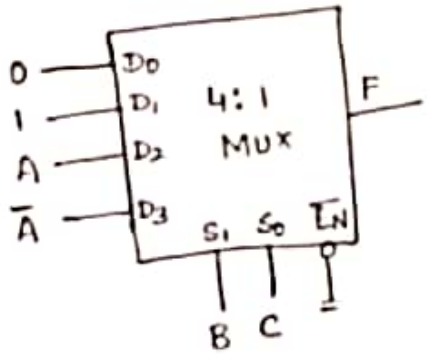
→ Connect least significant Variables as a select inputs of MUX. Here, Connect C to S_0 & B to S_1.

→ Derive inputs for MUX using implementation table

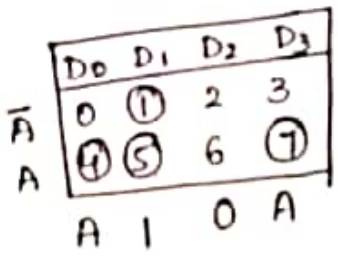


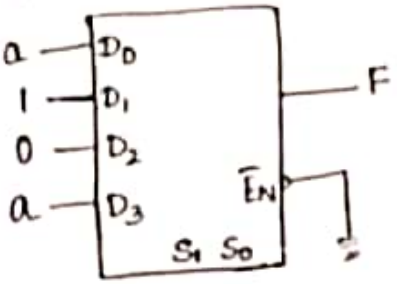
A / BC

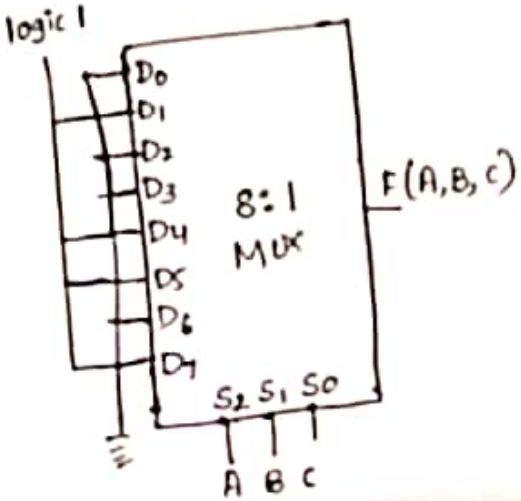
E	C	B	A	Y
E	S_0	S_1		
1	0	0	D_0	D_0
1	0	1	D_1	D_1
1	1	0	D_2	D_2
1	1	1	D_3	D_3



4) $f(A,B,C) = \Sigma(1,4,5,7)$ using 4:1 MUX and 8:1 MUX

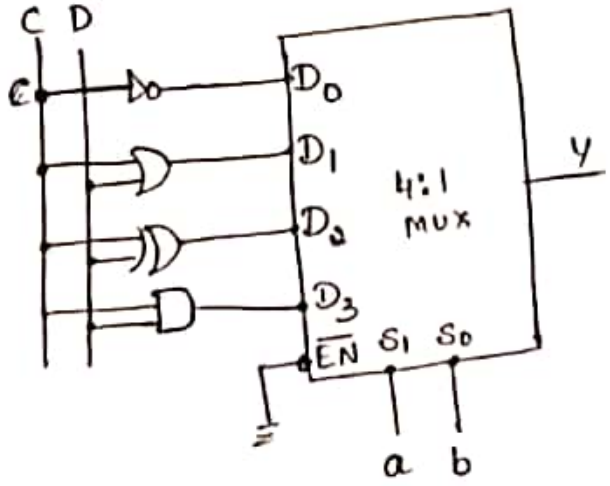






	00 $\bar{c}\bar{d}$	01 $\bar{c}d$	10 $c\bar{d}$	11 cd	
D_0	(0)	(1)	2	3	$\rightarrow \bar{c}\bar{d}+\bar{c}d=\bar{c}$
D_1	4	(5)	(6)	(7)	$\rightarrow \bar{c}d+c\bar{d}+cd=d+c\bar{d}=C+d$
D_2	8	(9)	(10)	11	$\rightarrow \bar{c}d+c\bar{d}=c\oplus d$
D_3	12	13	14	(15)	$\rightarrow cd$

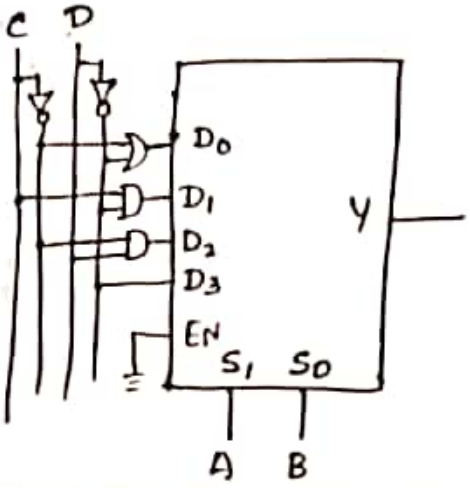
$A+\bar{A}B = A+B$



5) Implement the following Boolean function using a 4:1 MUX with A and B as select lines $Y=f(A,B,C,D)=\Sigma m(0,1,2,4,6,9,12,14)$

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

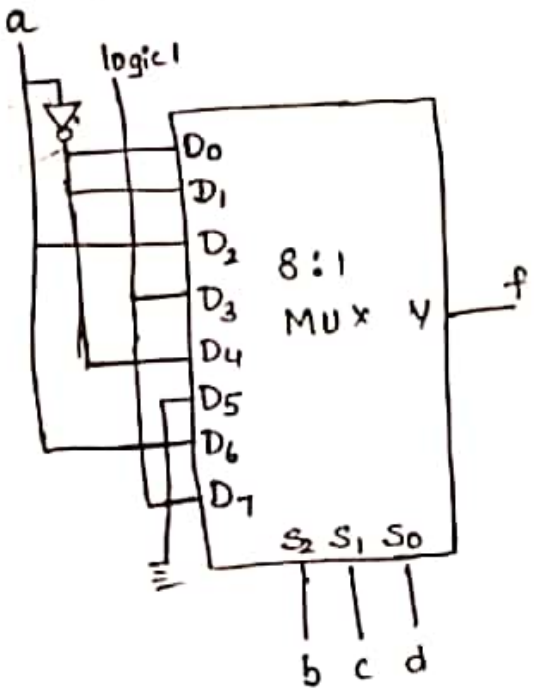
	$\bar{c}\bar{d}$	$\bar{c}d$	$c\bar{d}$	cd	
D_0	(0)	(1)	(2)	3	$\rightarrow \bar{c}\bar{d}+\bar{c}d+c\bar{d}=\bar{c}+c\bar{d}=\bar{c}+\bar{d}$
D_1	(4)	5	(6)	7	$\rightarrow c\bar{d}$
D_2	8	(9)	10	11	$\rightarrow \bar{c}d$
D_3	(12)	13	(14)	15	$\rightarrow \bar{c}\bar{d}+c\bar{d}=\bar{d}$



6) Implement the function using 8:1 MUX

$f(a,b,c,d) = \Sigma m(0,1,3,4,7,10,11,14,15)$

	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
$\bar{a}$	(0)	(1)	2	(3)	(4)	5	6	(7)
a	8	9	(10)	(11)	12	13	(14)	(15)
	$\bar{a}$	$\bar{a}$	a	1	$\bar{a}$	0	a	1

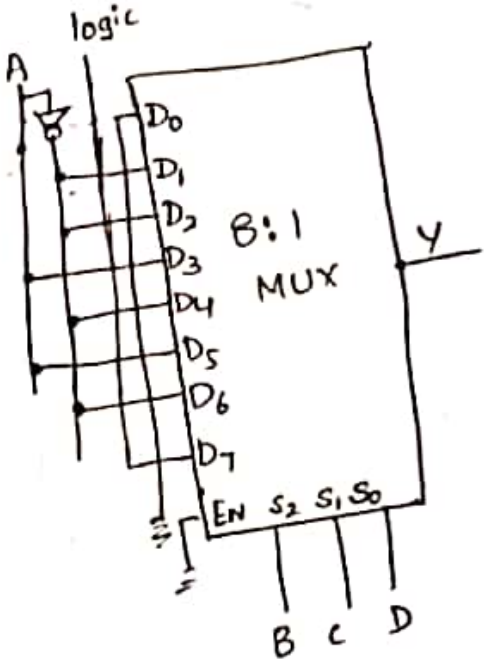


7) Implement the following with 8:1 MUX

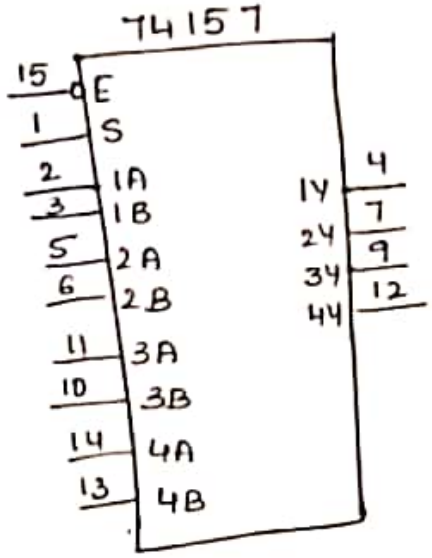
$F(A,B,C,D) = \Pi M(0,3,5,8,9,10,12,14)$

Soln:- Circle Maxterms that are included in the Boolean function

	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
$\bar{A}$	0	(1)	(2)	3	(4)	5	(6)	(7)
A	(8)	9	10	(11)	12	(13)	14	(15)
	0	$\bar{A}$	$\bar{A}$	A	$\bar{A}$	A	$\bar{A}$	1

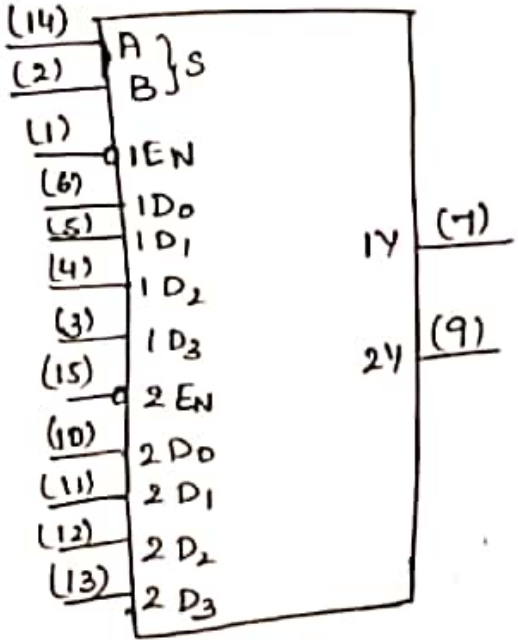


74157 Quad 2-Input MUX



Inputs		outputs			
$\overline{E}$	S	1Y	2Y	3Y	4Y
1	X	0	0	0	0
0	0	1A	2A	3A	4A
1	1	1B	2B	3B	4B

74153 Dual 4 to 1 MUX



Inputs				output	
1EN	2EN	B	A	1Y	2Y
0	0	0	0	$1D_0$	$1D_0$
0	0	0	1	$1D_1$	$2D_1$
0	0	1	0	$1D_2$	$2D_2$
0	0	1	1	$1D_3$	$2D_3$
0	1	0	0	$1D_0$	0
0	1	0	1	$1D_1$	0
0	1	1	0	$1D_2$	0
0	1	1	1	$1D_3$	0
1	0	0	0	0	$2D_0$
1	0	0	1	0	$2D_1$
1	0	1	0	0	$2D_2$
1	0	1	1	0	$2D_3$
1	1	X	X	0	0

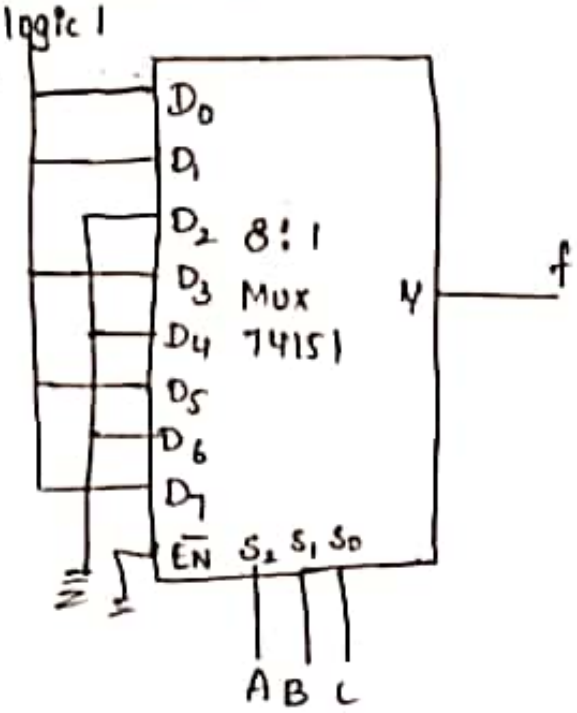
1) Realize the following Boolean function $f(A,B,C) = \Sigma(0,1,3,5,7)$ using i) 8:1 MUX (74151)

ii) 4:1 MUX (74153)

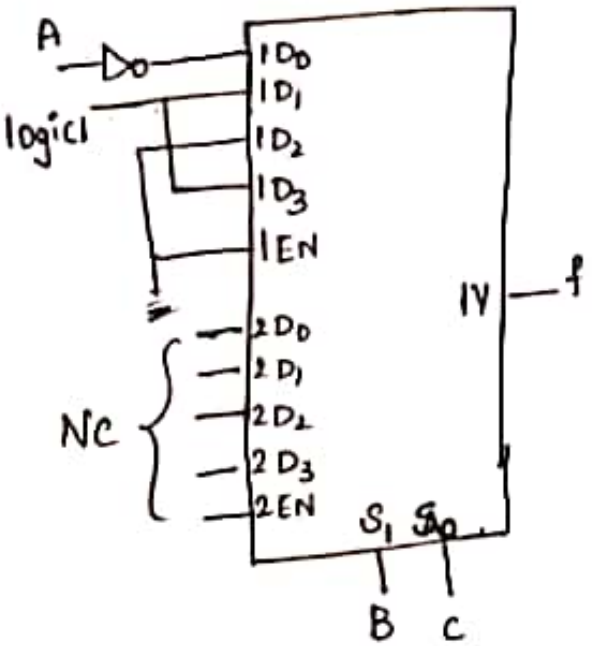
Soln:-

1) $f(A,B,C,D) = \Sigma(0,1,3,5,7)$

8:1 MUX



ii) 4:1 MUX



	D_0	D_1	D_2	D_3
$\bar{A}$	0	1	2	3
A	4	5	6	7
	$\bar{A}$	1	0	1

2) Implement a full Subtractor using a 4:1 MUX

Soln:-

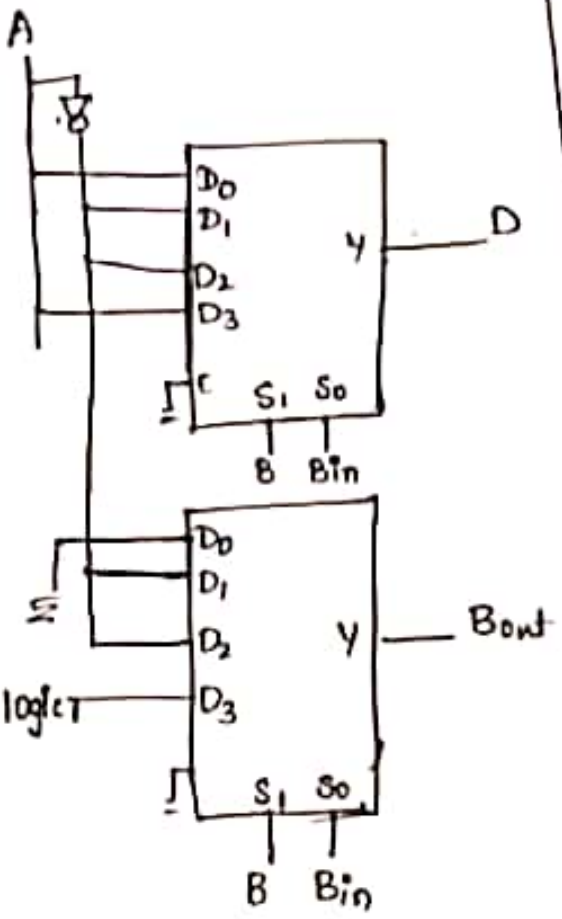
Input			output	
A	B	Bin	D	Bout
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

for D

	D_0	D_1	D_2	D_3
$\bar{A}$	0	(1)	(2)	3
A	(4)	5	6	(7)
	A	$\bar{A}$	$\bar{A}$	A

for Bout

	D_0	D_1	D_2	D_3
$\bar{A}$	0	(1)	(2)	(3)
A	4	5	6	(7)
	0	$\bar{A}$	$\bar{A}$	1



Full adder using 4:1 MUX

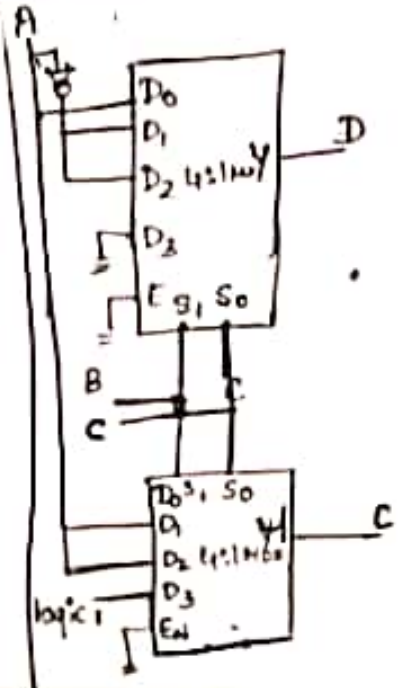
Input			output	
A	B	C	D	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

For D:

	D_0	D_1	D_2	D_3
$\bar{A}$	0	(1)	(2)	3
A	(4)	5	6	7
	A	$\bar{A}$	$\bar{A}$	0

For C:

	D_0	D_1	D_2	D_3
$\bar{A}$	0	1	2	(3)
A	4	(5)	(6)	(7)
	0	A	A	1



2) Implement $y = ad + b\bar{c} + bd$ using 4:1 Mux using ab as select input.

Soln:-

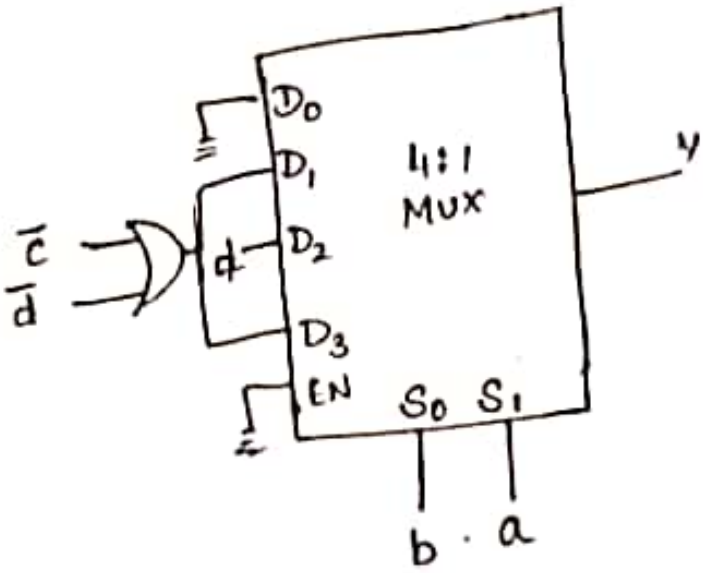
$$y = ad + b\bar{c} + bd$$

$$= abcd + ab\bar{c}d + a\bar{b}cd + a\bar{b}\bar{c}d + ab\bar{c}d + \bar{a}b\bar{c}d +$$
$$ab\bar{c}\bar{d} + \bar{a}b\bar{c}\bar{d} + abcd + \bar{a}bcd + ab\bar{c}d + \bar{a}b\bar{c}d$$

$$= \Sigma(15, 13, 11, 9, 5, 12, 4, 7)$$

$$= \Sigma(4, 5, 7, 9, 11, 12, 13, 15)$$

	$\bar{c}\bar{d}$	$\bar{c}d$	$c\bar{d}$	cd	
	0	1	2	3	
D_0					0
D_1	(4)	(5)	6	(7)	$\bar{c}\bar{d} + \bar{c}d + cd = \bar{c} + d$
D_2	8	(9)	10	(11)	$\bar{c}d + cd = d$
D_3	(12)	(13)	14	(15)	$\bar{c} + d$



Adder :

1) Half adder
2) Full adder.

1) Half adder

2 i/p - A . B

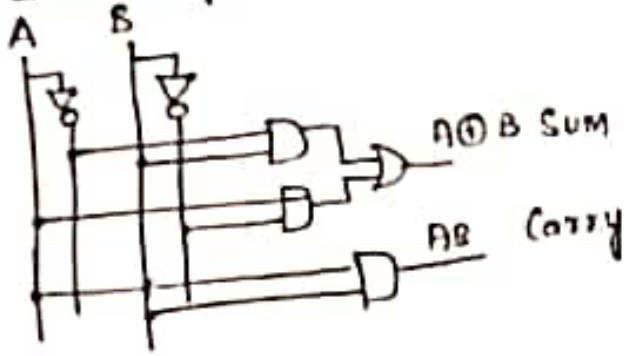
2 o/p → Sum & Carry

I/P		O/P	
A	B	Sum	carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

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

$Carry = AB$

Circuit diagram



2) Full adder

3 i/p → A, B, C

2 o/p → Sum & Carry

I/P			O/P	
A	B	C	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Sum

A \ BC	00	01	11	10
$\bar{A}$	0	1	3	2
A	4	5	7	6

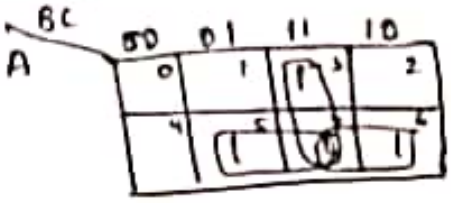
$Sum = A\bar{B}\bar{C} + \bar{A}\bar{B}C + ABC + \bar{A}B\bar{C}$

$= A(\overline{BC} + BC) + \bar{A}(\bar{B}C + B\bar{C})$

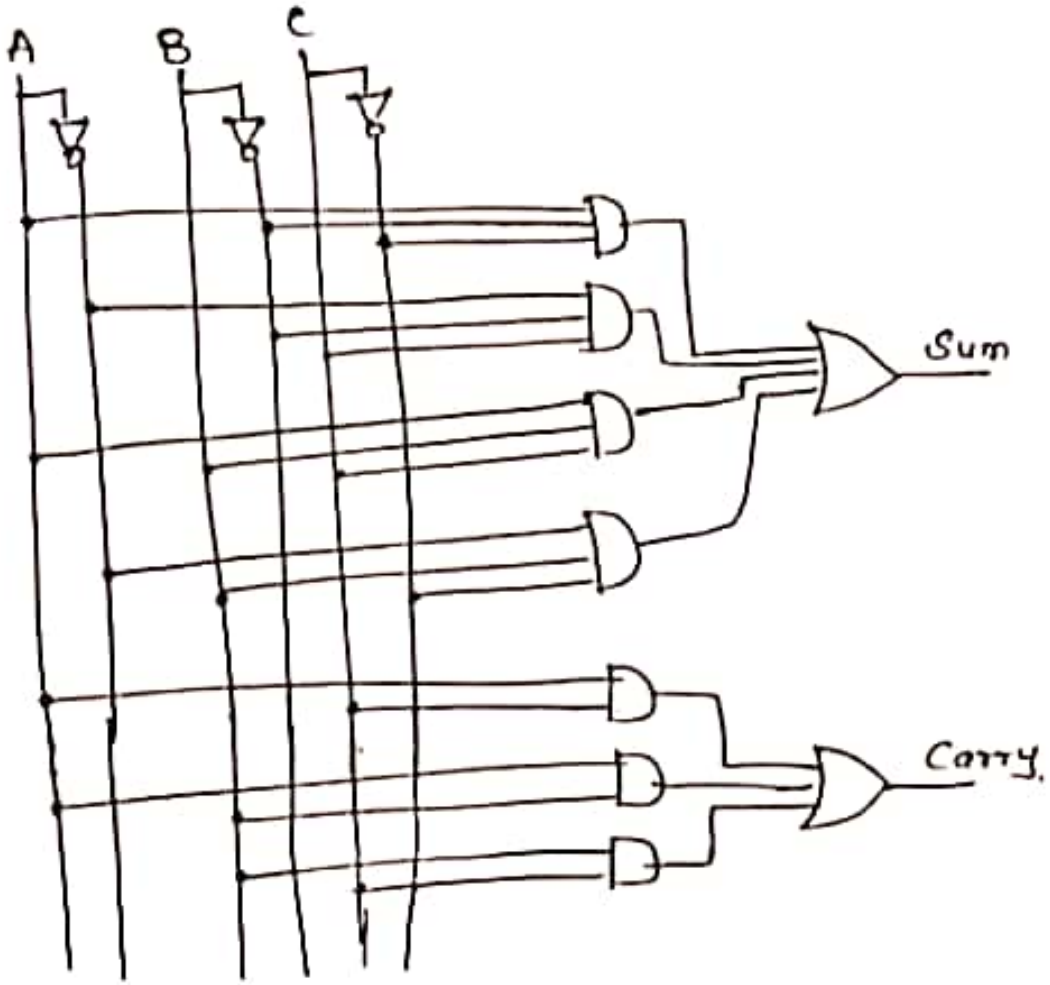
$= A(\overline{B \oplus C}) + \bar{A}(B \oplus C)$

$= A \oplus B \oplus C$

Carry



Carry = AC + AB + BC



Half Subtractor

I/P		output	
A	B	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

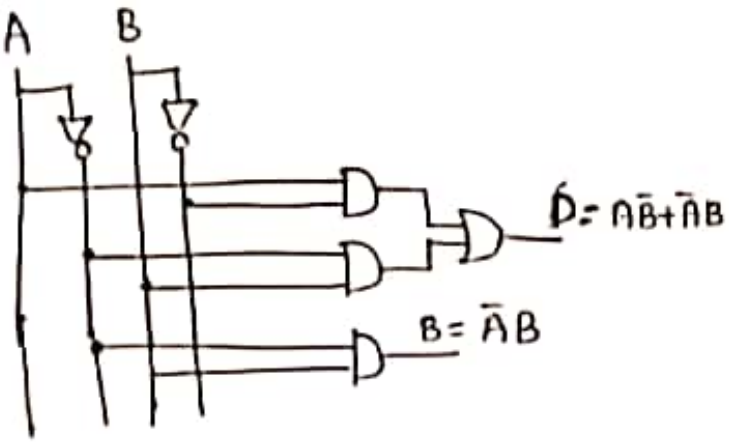
Difference

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

$= A \oplus B$

Borrow

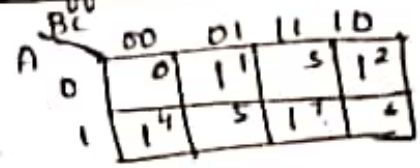
$B = \bar{A}B$



Full Subtractor

Input			output	
A	B	C	Difference	Borrow
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

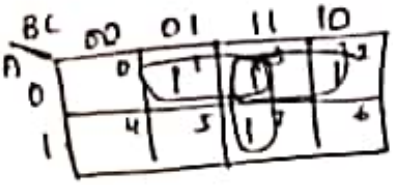
Difference



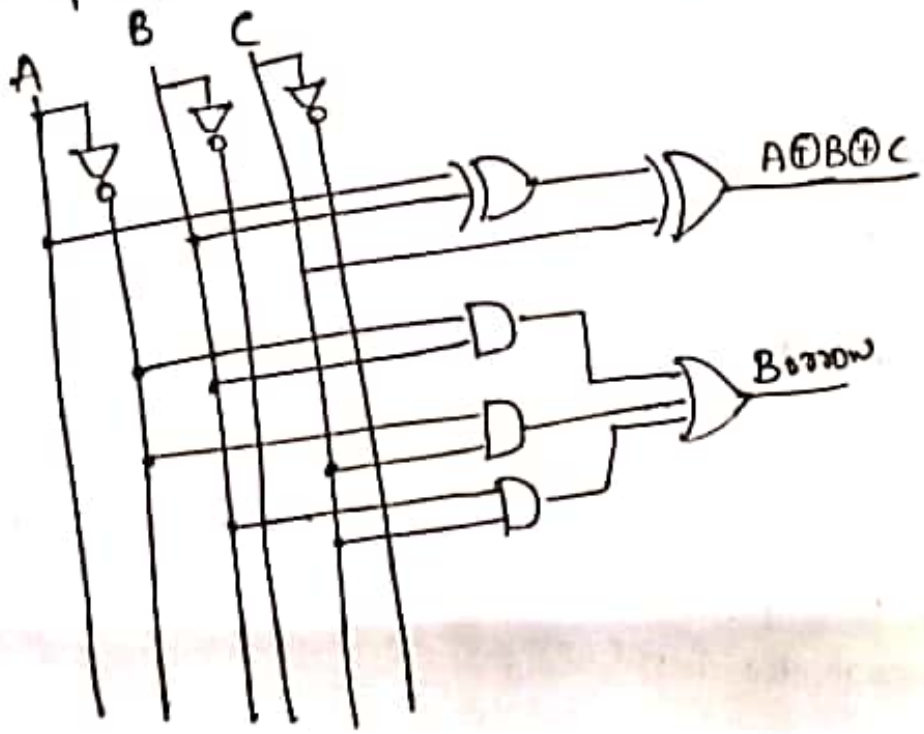
$Y = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + A\bar{B}\bar{C} + ABC$

$B_0D = A \oplus B \oplus C$

Borrow



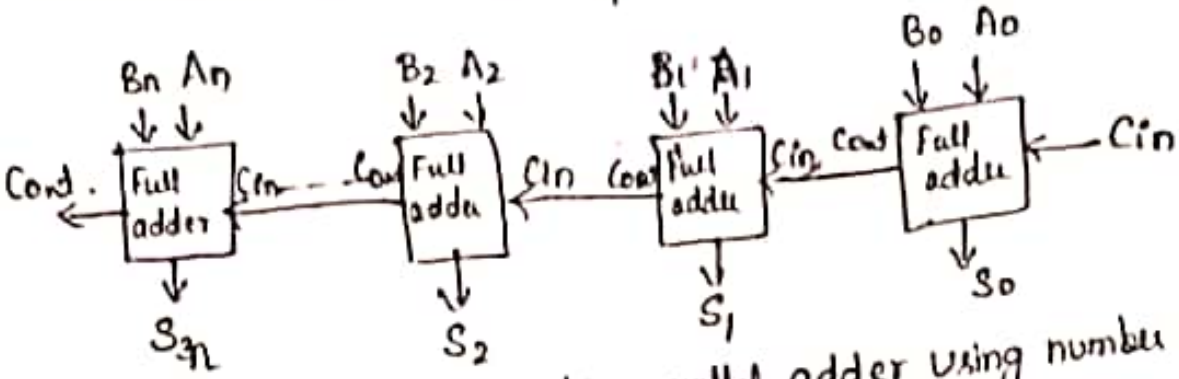
$B = \bar{A}B + \bar{A}C + BC$



Decoding

Cascading Full Adders

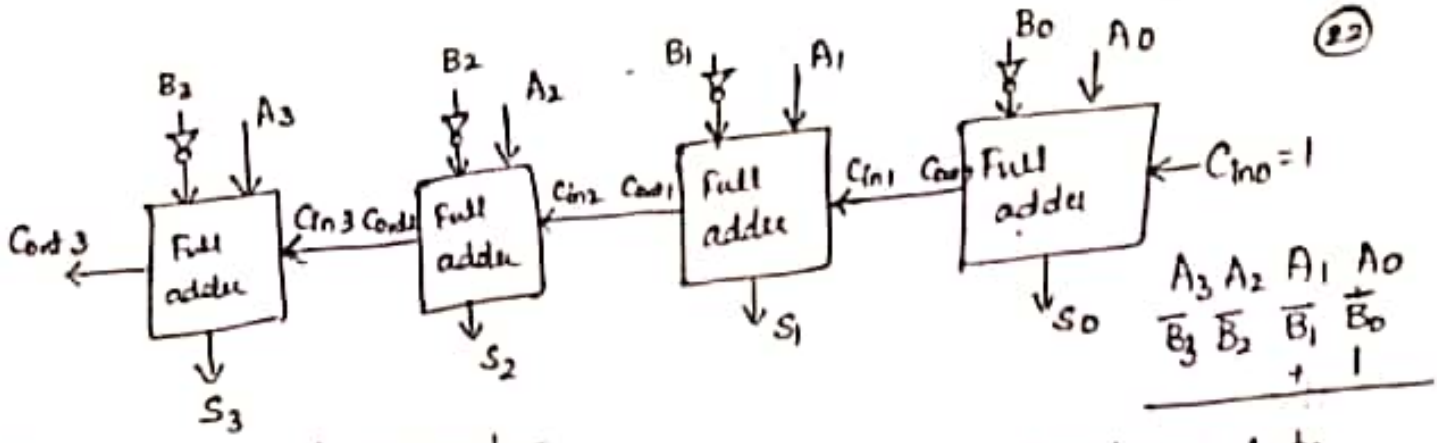
→ A single full adder is capable of adding two one bit numbers and an input carry.
→ In order to add binary numbers with more than one bit, additional full adders must be employed.
→ A n-bit parallel adder can be constructed using number of full adder circuits connected in parallel



→ The block diagram of n-bit parallel adder using number of full adder circuit connected in cascade is the carry output of each adder is connected to carry input of next higher-order adder.
→ It should be noted that either a half adder can be used for the least significant position of the carry input of a full adder is made 0 there is no carry into the Least Significant bit position.

Parallel Subtractor

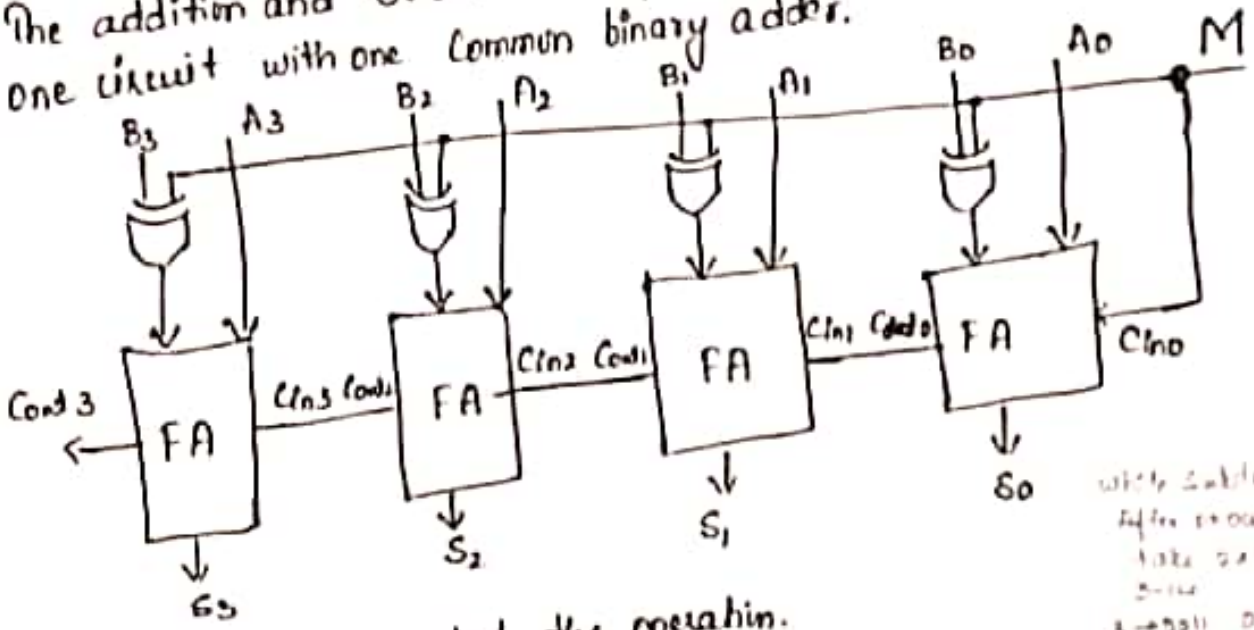
→ The Subtraction of binary number can be most conveniently by means of complements.
→ we know that A-B can be done by taking the 2's complement of B and adding it to A.
→ The 2's complement can be obtained by taking the 1's complement and adding one to the least significant pair of bits.
→ The 1's complement can be implemented with inverters and a one can be added to the sum through the input carry.



$A_3\ A_2\ A_1\ A_0$
$\bar{B}_3\ \bar{B}_2\ \bar{B}_1\ \bar{B}_0$
$+\qquad\qquad\quad 1$

Parallel Adder | Subtractor

→ The addition and Subtraction operation can be Combined into one circuit with one common binary adder.



→ The mode input M Controls the operation.
when M=0, the circuit is an adder
M=1, the circuit is Subtractor.

→ Ex-OR gate receives input M and one of the input of B.

→ when M=0, we have $B \oplus 0 = B$.
→ input A (Carry) is 0 → A+B.

→ when M=1 we have $B \oplus 1 = \bar{B}$ and $C_{in0} = 1$.
The B inputs are all Complemented and 1 is added through the input Carry. (A-B)

Ex OR	
0 0	0
0 1	1
1 0	1
1 1	0

→ The parallel adder is ripple Carry adder in which the Carry output of each full adder stage is Connected to the Carry input of the next higher order stage.

→ ∴ The Sum and Carry output of any Stage Cannot be produced until the input Carry occurs, this leads to a time delay in the addition process. This delay is known as Carry propogation delay.

→ One method of speeding up this process by eliminating inter stage Carry delay is called look ahead - Carry addition.

look - Ahead Carry Adder

→ The parallel adder and Subtractor is ripple Carry type in which the Carry o/p of each full adder type is connected to the i/p of next stage.

→ The Sum and Carry o/p's of any stage cannot be produced until the i/p Carry occurs. This leads to a time delay in the addition process. This delay is known as Carry propogation delay.

→ If each adder is considered to have a propogation delay of 30ns, then S_3 will not reach its Correct Value until 90ns after LSB Carry is generated.

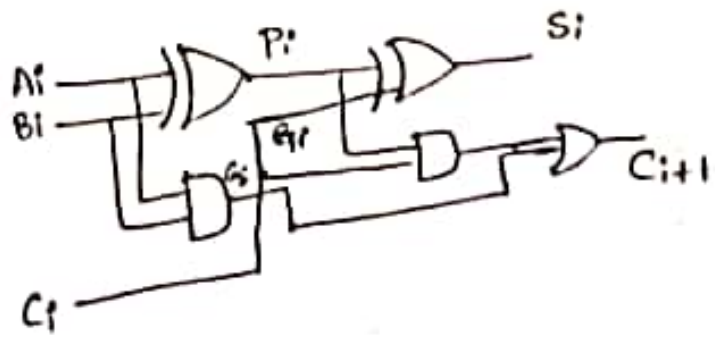
→ Therefore the total time to perform addition is 90+30 = 120ns.

→ one method of speeding up this process by eliminating inter stage Carry delay is Called look ahead- Carry addition.

Consider the circuit of full adder

$P_i = A \oplus B$

$G_i = A_i B_i$



Sum & Carry

$S_i = P_i \oplus C_i$

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

G_i → Carry generate and it produces on Carry when both A_i & B_i are one, regardless of i/p Carry.

P_i → Carry propogate ∵ it is term associated with the propogation of the Carry from C_i to C_{i+1}

$S \rightarrow A \oplus B \oplus C$

$C \rightarrow AB + BC + AC$

$\rightarrow AB + B(A \oplus B) + A(A \oplus B)'$

$\rightarrow AB + AB + C(A \oplus B) + AB'$

$AB + C(A \oplus B)$

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

$C_2 = G_1 + P_1 C_1$

$C_3 = G_2 + P_2 C_2 = G_2 + P_2 (G_1 + P_1 C_1)$

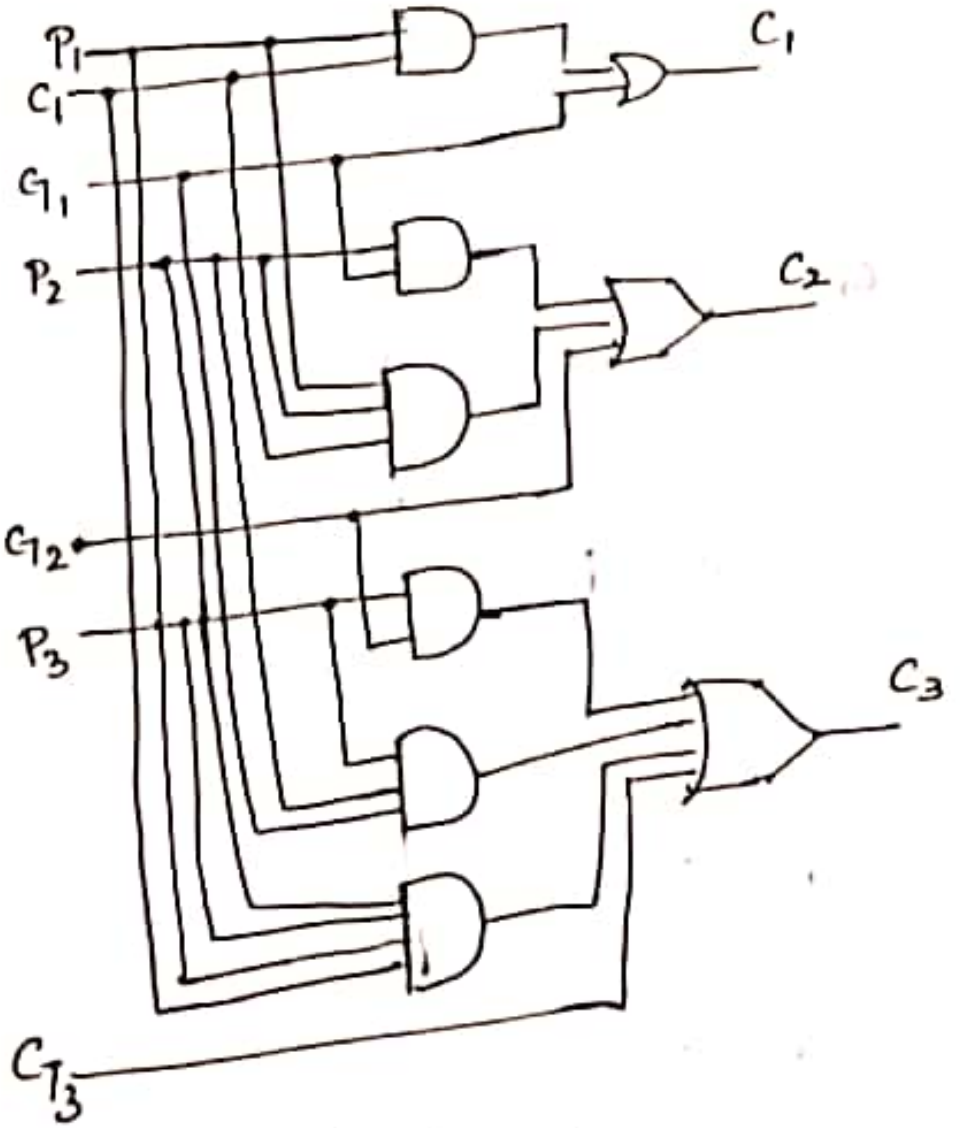
$C_3 = G_2 + P_2 G_1 + P_1 P_2 C_1$

$C_4 = G_3 + P_3 C_3$

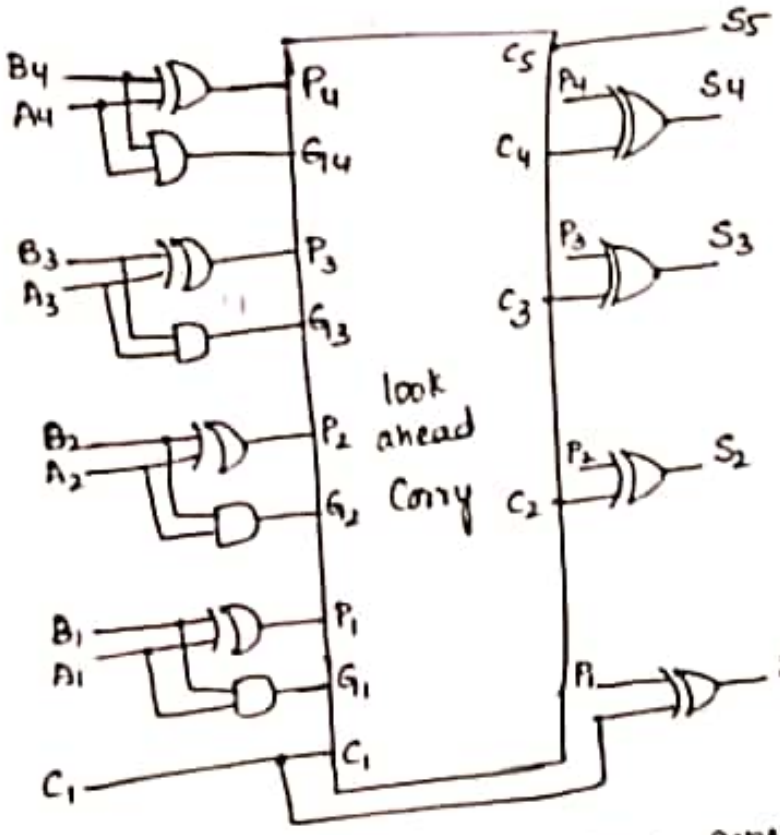
$= G_3 + P_3 (G_2 + P_2 G_1 + P_2 P_1 C_1)$

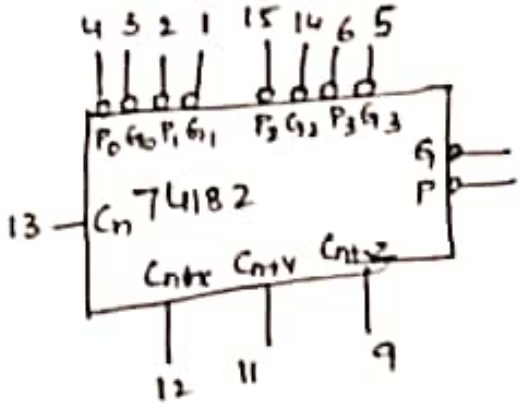
$= G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 C_1$

It can be seen that C_4 do not have to wait for C_3 & C_2 to propogate, in fact C_4 is propogated at the same time as C_2 and C_3.



logic diagram





+ IC 74182 is a look ahead Carry generator.

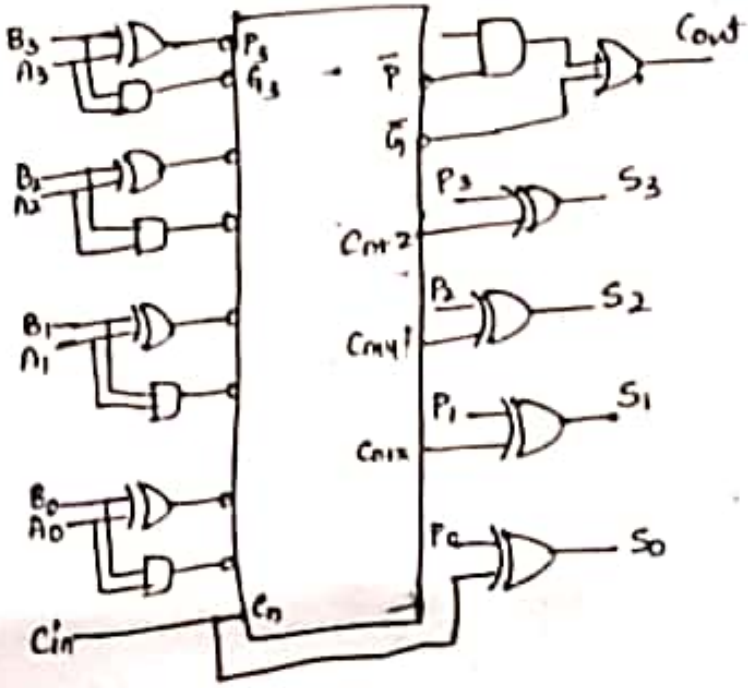
$C_{n+x} = G_0 + P_0 C_n$

$C_{n+y} = G_1 + P_1 G_0 + P_1 P_0 G_n$

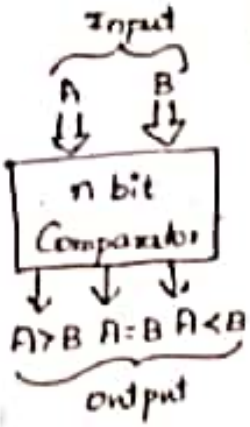
$C_{n+z} = G_2 + P_2 G_1 + P_2 P_1 C_n$

$\overline{G} = \overline{G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0}$

$\overline{P} = \overline{P_3 P_2 P_1 P_0}$



Binary Comparators



→ A Comparator is a special Combinational circuit designed primarily to compare the relative magnitude of two binary numbers.

→ It receives two n-bit numbers A and B as inputs and the output are A>B, A=B and A<B.

→ Depending upon the relative magnitudes of the two numbers, one of the output will be high.

Design 2-bit Comparator using gates

Input				output		
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	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

A > B

A_1A_0 \ B_1B_0	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	1	1	0	0
10	1	1	0	0

A = B

A_1A_0 \ B_1B_0	00	01	11	10
00	1	0	0	0
01	0	1	0	0
11	0	0	1	0
10	0	0	0	1

A_1A_0 \ B_1B_0	00	01	11	10
00	0	1	0	0
01	0	0	0	1
11	0	0	0	0
10	0	0	0	0

$$A > B = A_0\bar{B}_1\bar{B}_0 + A_1\bar{B}_1 + A_1A_0\bar{B}_0$$

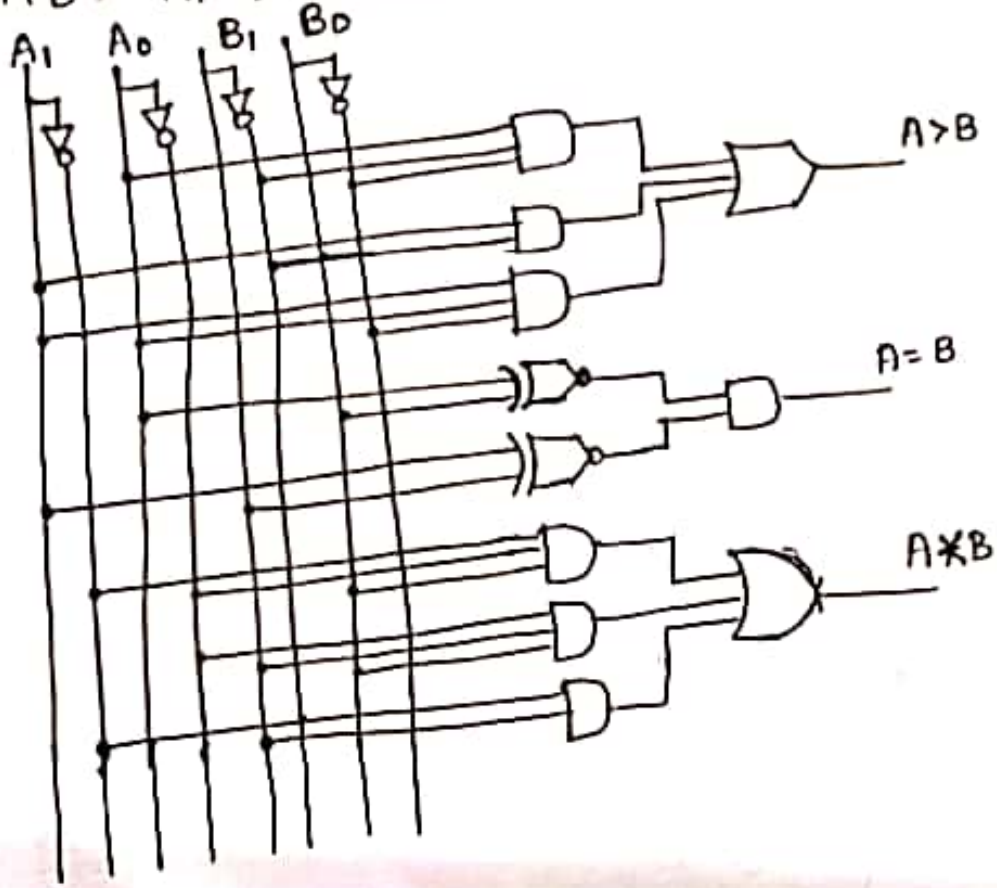
$$A = B = \bar{A}_1\bar{A}_0\bar{B}_1\bar{B}_0 + \bar{A}_1A_0\bar{B}_1B_0 + A_1A_0B_1B_0 + A_1\bar{A}_0B_1B_0$$

$$= \bar{A}_1\bar{B}_1(\bar{A}_0\bar{B}_0 + A_0B_0) + A_1B_1(A_0B_0 + \bar{A}_0\bar{B}_0)$$

$$= (\bar{A}_0\bar{B}_0 + A_0B_0)(\bar{A}_1\bar{B}_1 + A_1B_1)$$

$$= (A_0 \odot B_0)(A_1 \odot B_1)$$

$$A < B = \bar{A}_1\bar{A}_0B_0 + \bar{A}_0B_1B_0 + \bar{A}_1B_1$$



1) Design a 1-bit Comparator using basic gates

Soln:-

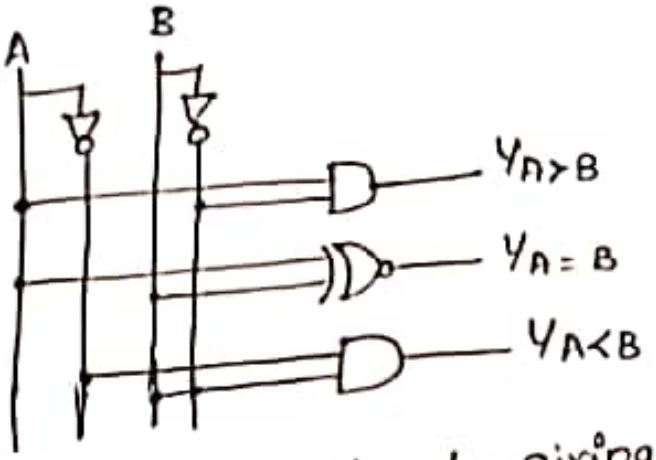
Input		output		
A	B	$Y_{A>B}$	$Y_{A=B}$	$Y_{A<B}$
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

i) $Y_{A>B} = A\bar{B}$

ii) $Y_{A=B} = \bar{A}\bar{B} + AB$

$= A \odot B$

iii) $Y_{A<B} = \bar{A}B$



2) Design a 1bit Comparator using 2:4 decoder giving three outputs G, E and L.

X	Y	G $X>Y$	E $X=Y$	L $X<Y$
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0