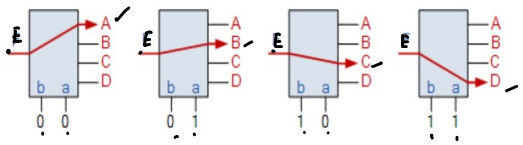
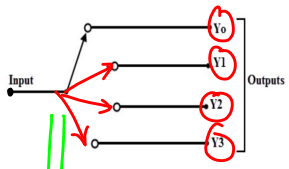
Lec-35

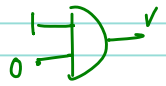
Demultiplexer (Continuation)

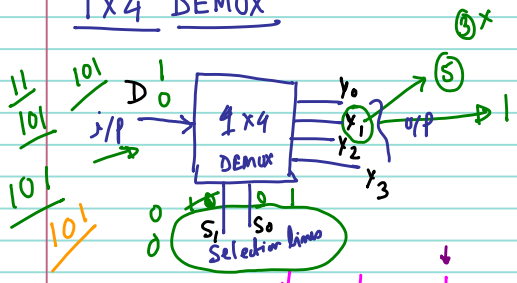




Selection lines

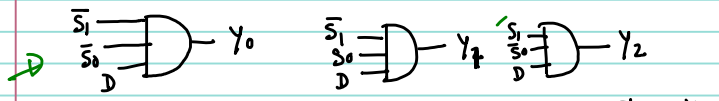
1 X 4 DEMUX

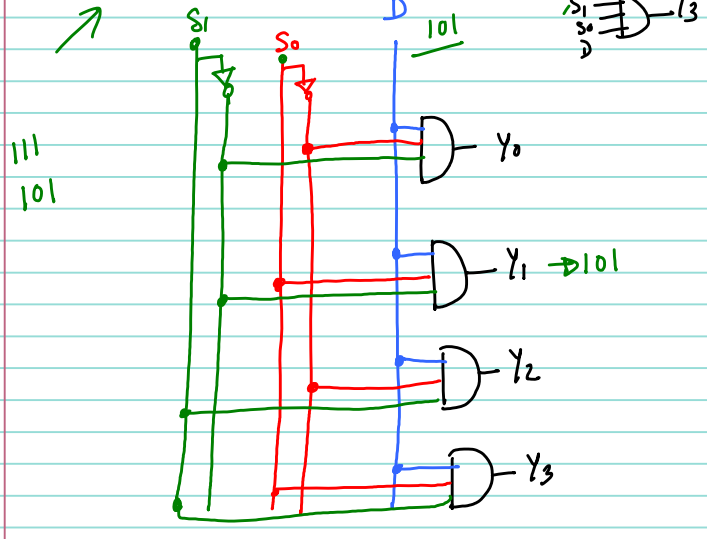




S_1	S_0	D	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0
0	0	1	0	0	0	1
0	1	0	0	0	0	0
0	1	1	0	0	1	0
1	0	0	0	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	0	0
1	1	1	1	0	0	0

$Y_0 = \bar{S_1}\bar{S_0}D$, $Y_1 = \bar{S_1}S_0D$, $Y_2 = S_1\bar{S_0}D$, $Y_3 = S_1S_0D$

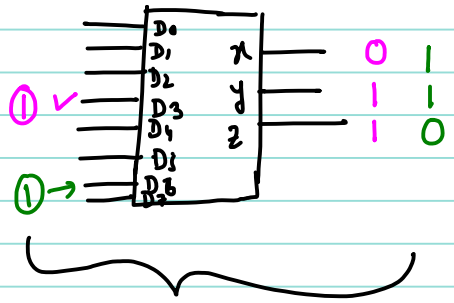




Encoder:

MSB ↓ LSB ↓

x y z



Octal to binary encoder → make the circuit in gate level.

8 / 256

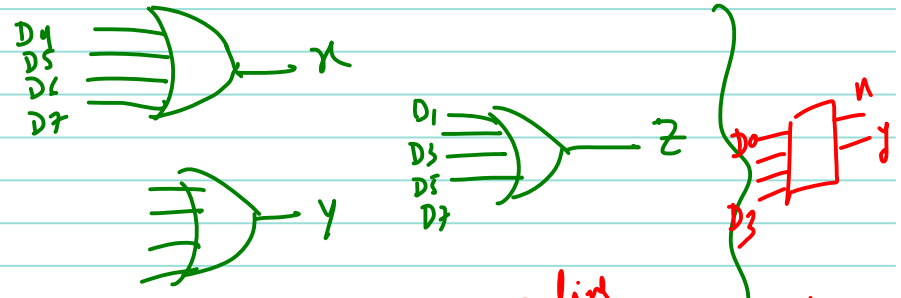
D0	D1	D2	D3	D4	D5	D6	D7	x	y	z
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

$x = D_4 + D_5 + D_6 + D_7$

$y = D_2 + D_3 + D_6 + D_7$

$z = D_1 + D_3 + D_5 + D_7$

'OR' operation between i/ps



Priority Encoder:

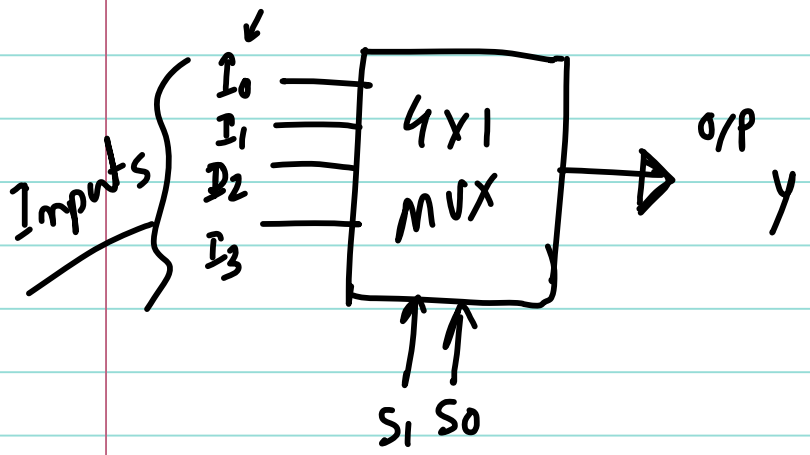
8 line 3 line

4 line 2 line - priority Encoder

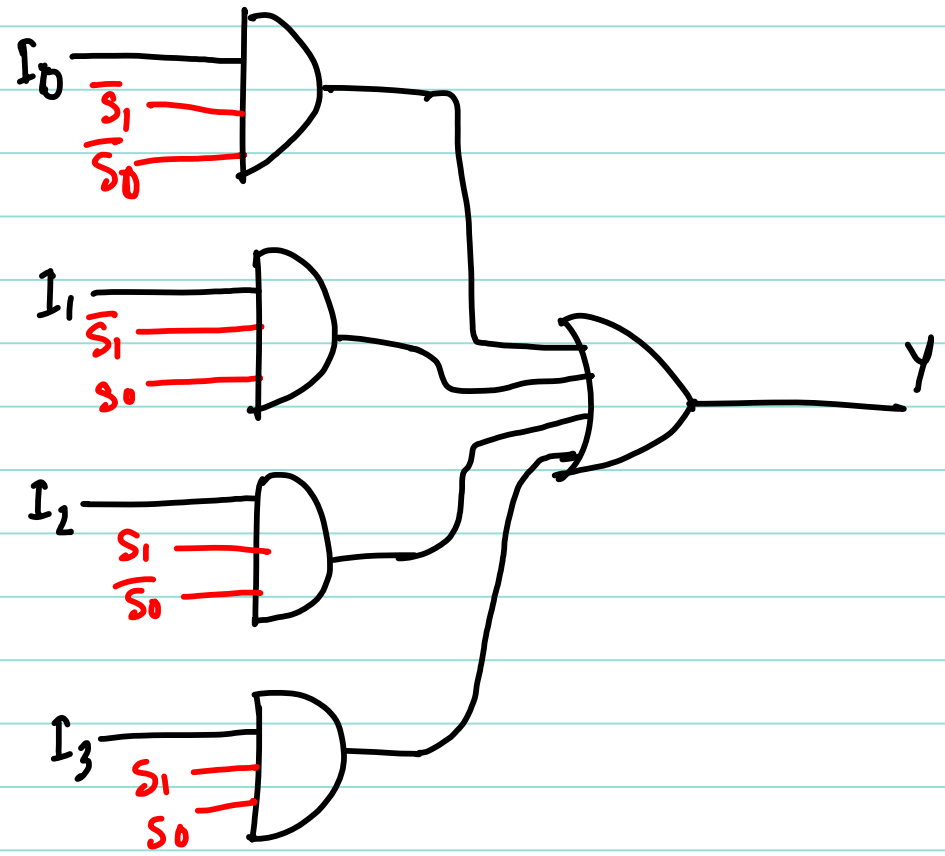
D0	D1	D2	D3	D4	D5	D6	D7	x	y	z
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	1	0	0	0	1	0
0	0	0	1	0	0	1	0	1	0	1
0	0	0	0	1	0	0	1	1	1	0

Multiplexer: (MUX)

4 x 1 MUX



S_1	S_0	y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

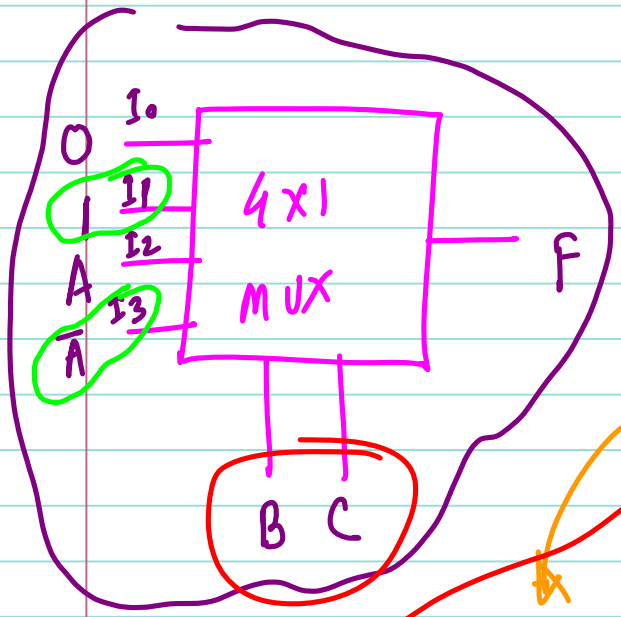


minterms

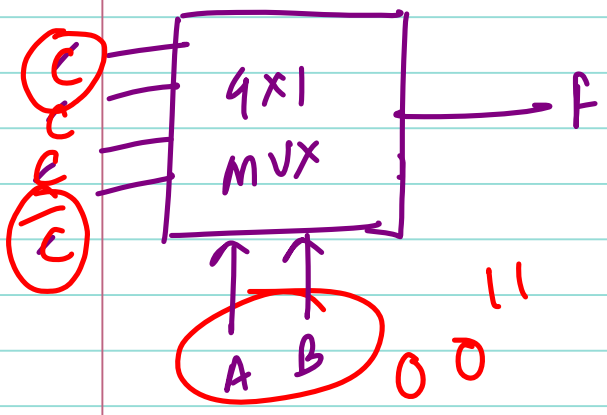
$F(A, B, C) = \sum (1, 3, 5, 6)$ implement.

using a 4x1 MUX

Minterms	A	B	C	F
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0



	I_0	I_1	I_2	I_3
→ $\bar{A}$	0	1	2	3
→ A	4	5	6	7
	0	1	A	$\bar{A}$



	I_0	I_1	I_2	I_3
$\bar{C}$	0	2	4	6
C	1	3	5	7
	C	C	C	$\bar{C}$