Dhirubhai Ambani Institute of Information and Communication Technology

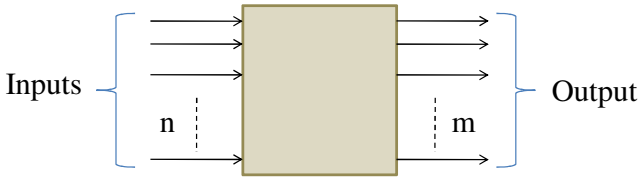
EL114

Digital Logic Design

Dr. Yash Agrawal
Assistant Professor,
DA-IICT, Gandhinagar

Introduction

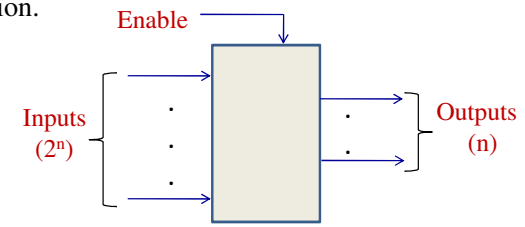
Combinational logic Circuit



Encoder

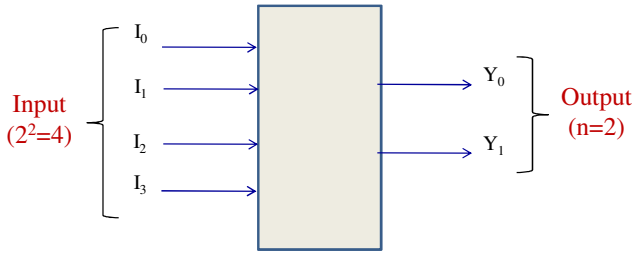
Encoder:

- '2^n' inputs to 'n' outputs.
- It has no select lines.
- Usually single input is assumed to be high. If more than one input is high, then Priority Encoder is used.
- Used in many applications such as memory system, code conversion, implementation of function.



Encoder Example

4:2 Encoder:



4:2 Encoder

First Step:
Truth Table Formation

Input of Encoder				Output of Encoder	
I_3	I_2	I_1	I_0	Y_1	Y_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

4:2 Encoder

Second Step:
Determining Boolean Expression

Input of Encoder				Output of Encoder	
I_3	I_2	I_1	I_0	Y_1	Y_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

$$Y_0 = I_1 + I_3$$

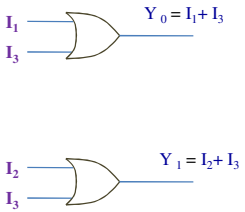
$$Y_1 = I_2 + I_3$$

4:2 Encoder

Third Step:
Realization of Boolean Expression by Logic Gates

$$Y_0 = I_1 + I_3$$

$$Y_1 = I_2 + I_3$$



4:2 Encoder

What if more than one input is High…?

Input of Encoder				Output of Encoder	
I_3	I_2	I_1	I_0	Y_1	Y_0
0	0	0	1		
0	1	0	0		
0	1	1	0		
1	1	1	1		

4:2 **Priority** Encoder

The ambiguity for cases when inputs are High for more than at one place can be solved by assigning priority (P).

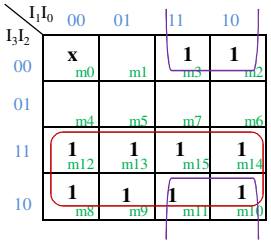
First Step:
Truth Table Formation

Input of Encoder				Output of Encoder		
I_3	I_2	I_1	I_0	P	Y_1	Y_0
0	0	0	0	0	x	x
0	0	0	1	1	0	0
0	0	1	x	1	0	1
0	1	x	x	1	1	0
1	x	x	x	1	1	1

4:2 **Priority** Encoder

Second Step:
Determining Boolean Expression
(For this K-map can be used)

K-map and Boolean Expression determination for Y_0



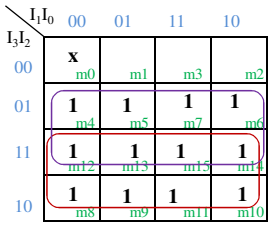
$Y_0 = I_2'I_1 + I_3$

Input of Encoder				Output of Encoder		
I_3	I_2	I_1	I_0	P	Y_1	Y_0
0	0	0	0	0	x	x
0	0	0	1	1	0	0
0	0	1	x	1	0	1
0	1	x	x	1	1	0
1	x	x	x	1	1	1

4:2 **Priority** Encoder

Second Step:
Determining Boolean Expression
(For this K-map can be used)

K-map and Boolean Expression determination for Y_1



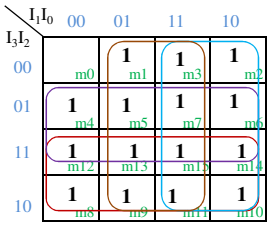
$Y_1 = I_2 + I_3$

Input of Encoder				Output of Encoder		
I_3	I_2	I_1	I_0	P	Y_1	Y_0
0	0	0	0	0	x	x
0	0	0	1	1	0	0
0	0	1	x	1	0	1
0	1	x	x	1	1	0
1	x	x	x	1	1	1

4:2 **Priority** Encoder

Second Step:
Determining Boolean Expression
(For this K-map can be used)

K-map and Boolean Expression determination for P



$P = I_0 + I_1 + I_2 + I_3$

Input of Encoder				Output of Encoder		
I_3	I_2	I_1	I_0	P	Y_1	Y_0
0	0	0	0	0	x	x
0	0	0	1	1	0	0
0	0	1	x	1	0	1
0	1	x	x	1	1	0
1	x	x	x	1	1	1

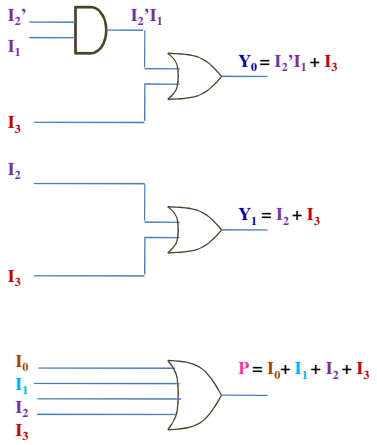
4:2 **Priority** Encoder

Third Step:
Realization of Boolean Expression by Logic Gates

$Y_0 = I_2'I_1 + I_3$

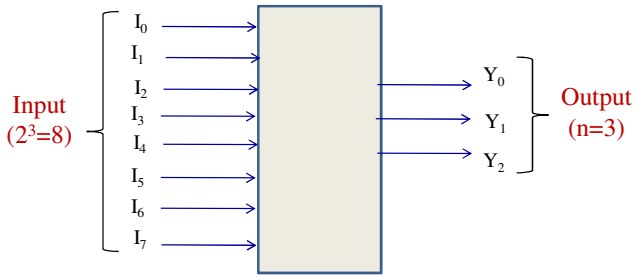
$Y_1 = I_2 + I_3$

$P = I_0 + I_1 + I_2 + I_3$



Encoder Example

8:3 Encoder:



8:3 Encoder

First Step:
Truth Table Formation

Input of Encoder								Output of Encoder		
I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

8:3 Encoder

Second Step:
Determining Boolean Expression

Input of Encoder								Output of Encoder		
I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	(1)
0	0	0	0	0	1	0	0	0	(1)	0
0	0	0	0	1	0	0	0	0	(1)	(1)
0	0	0	1	0	0	0	0	(1)	0	0
0	0	1	0	0	0	0	0	(1)	0	(1)
0	1	0	0	0	0	0	0	(1)	(1)	0
1	0	0	0	0	0	0	0	(1)	(1)	(1)

$Y_0 = I_1 + I_3 + I_5 + I_7$

$Y_1 = I_2 + I_3 + I_6 + I_7$

$Y_2 = I_4 + I_5 + I_6 + I_7$

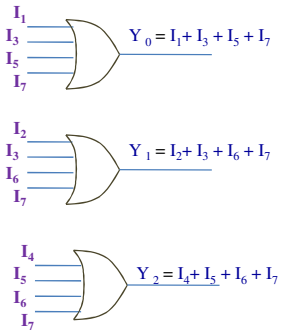
8:3 Encoder

Third Step:
Realization of Boolean Expression by Logic Gates

$Y_0 = I_1 + I_3 + I_5 + I_7$

$Y_1 = I_2 + I_3 + I_6 + I_7$

$Y_2 = I_4 + I_5 + I_6 + I_7$



Numerical

1. Implement Octal to Binary Encoder using 8:3 Encoder. It should able to convert from $(0)_8$ to $(7)_8$.

   For example: $(4)_8 = (100)_2$

Numerical

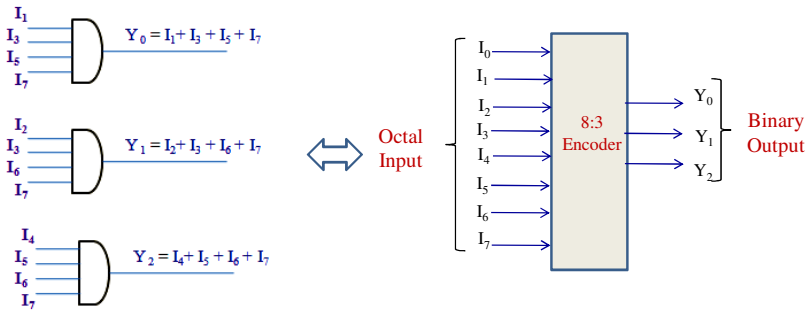
Implementation of Octal to Binary Encoder using 8:3 Encoder.

Octal Number | Input | Output

Oct	I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	1	0	0	0	1
2	0	0	0	0	0	1	0	0	0	1	0
3	0	0	0	0	1	0	0	0	0	1	1
4	0	0	0	1	0	0	0	0	1	0	0
5	0	0	1	0	0	0	0	0	1	0	1
6	0	1	0	0	0	0	0	0	1	1	0
7	1	0	0	0	0	0	0	0	1	1	1

Numerical

Implementation of Octal to Binary Encoder using 8:3 Encoder.



Assignment-5

1. Realize 8:3 Priority Encoder.