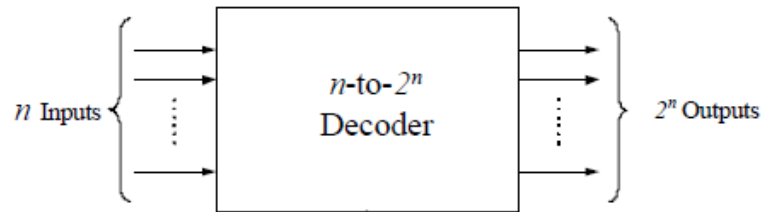


## Decoder

As its name indicates, a decoder is a circuit component that decodes an input code. Given a binary code of  $n$ -bits, a decoder will tell which code is this out of the  $2^n$  possible codes (See Figure ). Thus, a decoder has  $n$ - inputs and  $2^n$  outputs. Each of the  $2^n$  outputs corresponds to one of the possible  $2^n$  input combinations.



In general, output  $i$  equals 1 if and only if the input binary code has a value of  $i$ .

### Example: 2-to-4 decoder

Let us discuss the operation and combinational circuit design of a decoder by taking the specific example of a 2-to-4 decoder. It contains two inputs denoted by  $A$  and  $B$  and four outputs denoted by  $D_0$ ,  $D_1$ ,  $D_2$ , and  $D_3$  as shown in figure.

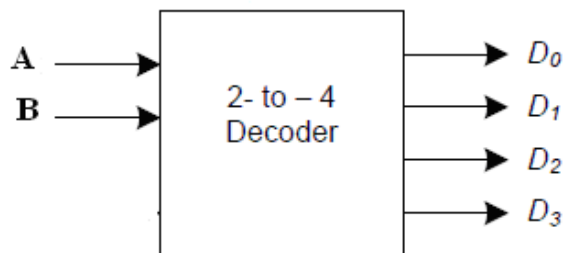


Figure Block Diagram of 2-to-4 Decoder

B	A	$D_0$	$D_1$	$D_2$	$D_3$
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Truth table of 2-to-4 Decoder

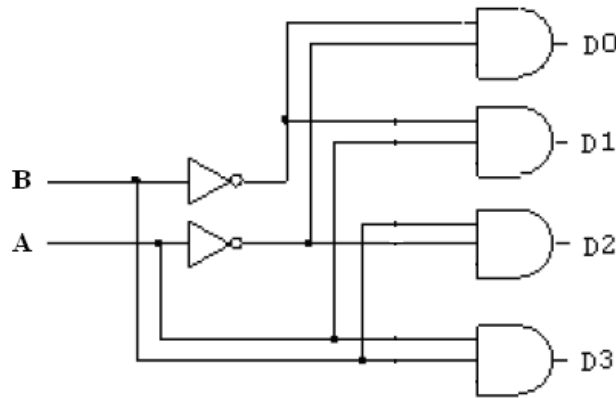
As we see in the truth table , for each input combination, one output line is activated, that is, the output line corresponding to the input combination becomes 1, while other lines remain inactive. For example, an input of 00 at the input will activate line  $D_0$ . and 01 at the input will activate line  $D_1$ , and so on.

$$D_0 = \bar{A}\bar{B}$$

$$D_1 = A\bar{B}$$

$$D_2 = \bar{A}B$$

$$D_3 = AB$$



The logic diagram of 2-to- 4 Decoder

**Example: 3-to-8 decoder**

It contains three inputs denoted by a, b and c, with eight outputs denoted by D0, D1, D2, D3, D4, D5, D6 and D7 as shown in figure

c	b	a	D0	D1	D2	D3	D4	D5	D6	D7
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

$$D_0 = \bar{c}\bar{b}\bar{a}$$

$$D_1 = \bar{c}\bar{b}a$$

$$D_2 = \bar{c}b\bar{a}$$

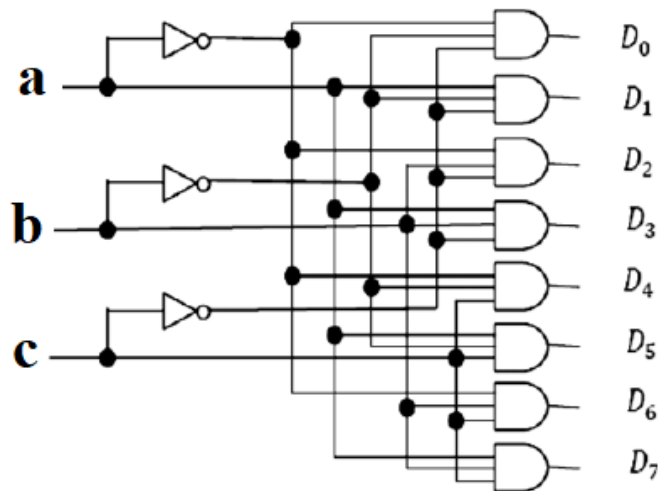
$$D_3 = \bar{c}ba$$

$$D_4 = c\bar{b}\bar{a}$$

$$D_5 = c\bar{b}a$$

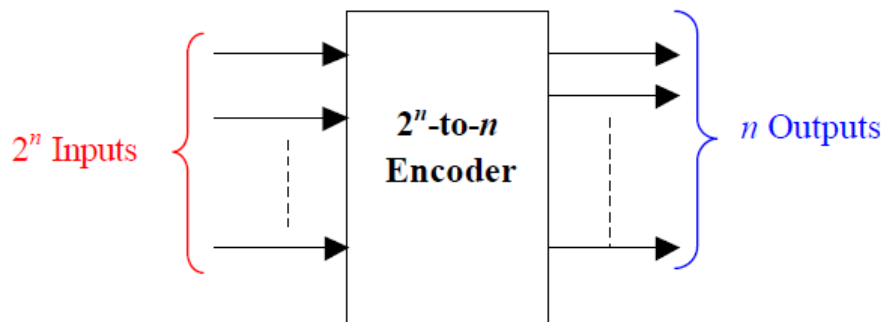
$$D_6 = cb\bar{a}$$

$$D_7 = cba$$



## Encoder

The encoder is a combinational circuit that performs the reverse operation of the decoder. The encoder has a maximum of  $2^n$  inputs and  $n$  outputs. Only one input can be logic 1 at any given time (active input). All other inputs must be 0's, and the Output lines generate the binary code corresponding to the active input. The block diagram of  $2^n$ -to- $n$  encoder as shown below .



### Example: 4-to-2 Encoder

The inputs are 4 and the outputs are 2  
The block diagram and the truth table of a 4-to-2 encoder are shown below .

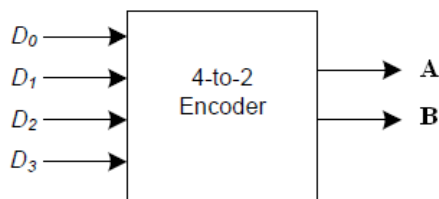


Figure Block Diagram of 4-to-2 Encoder

$D_0$	$D_1$	$D_2$	$D_3$	<b>B</b>	<b>A</b>
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

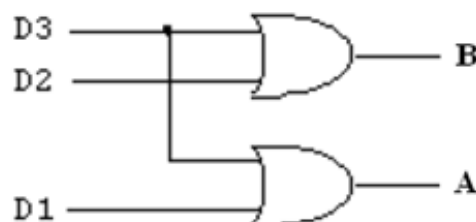
truth table For the 4-to-2 encoder

So that the logic expression of outputs are:

$$\mathbf{A = D1 + D3}$$

$$\mathbf{B = D2 + D3}$$

And the logic diagram of 4-to-2 encoder as shown below:



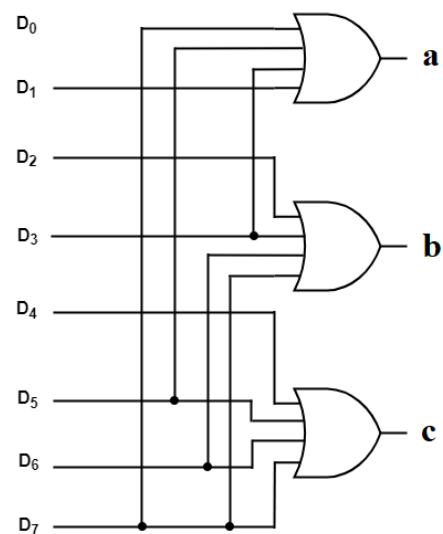
Logic Diagram of the 4-to-2 Encoder

**Example: 8-to-3 Encoder**

The inputs are 8 and the outputs are 3. The truth table and the logic diagram of a 8-to-3 encoder are shown below.

INPUTS								OUTPUTS		
D0	D1	D2	D3	D4	D5	D6	D7	c	b	a
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

$$\begin{aligned}
 a &= D_1 + D_3 + D_5 + D_7 \\
 b &= D_2 + D_3 + D_6 + D_7 \\
 c &= D_4 + D_5 + D_6 + D_7
 \end{aligned}$$



**Binary to Gray / Gray to Binary Conversion:**

The gray code is widely used in many digital systems specially in shaft encoders and analog to digital conversion, but it is difficult to use the gray-code in arithmetic operations, since there are only one bit change between two consecutive gray code number, and it is unweighted code, and the XOR gate is the most suitable gate for this purpose as shown in Figure below:

