Logical Design

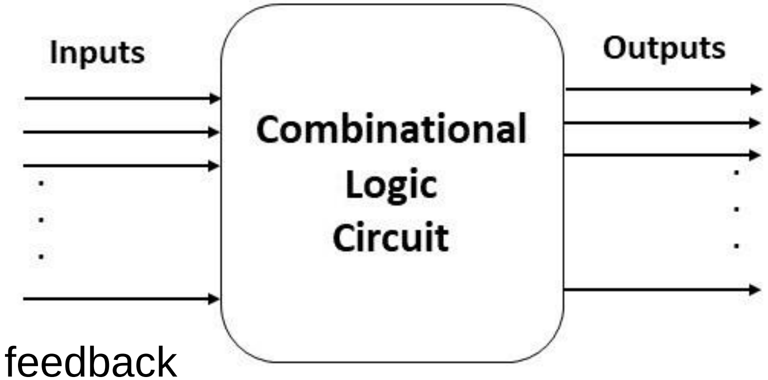
Lecture 12

Dr Zaid Al-Bayati

Digital circuits

Combinantial circuit

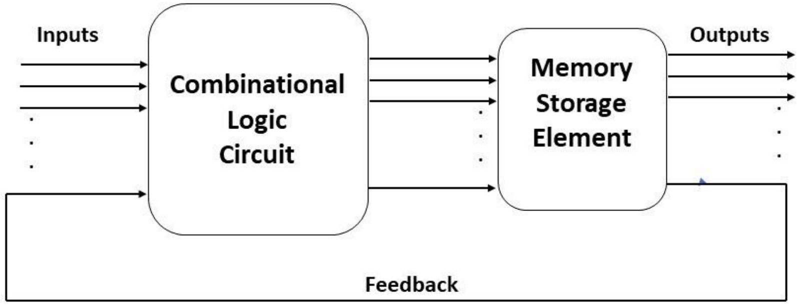
- Output of combinantial circuit based on the combinantial of present input only.



- No feedback
- No memory
- Ex: encoder, decoder, multiplexer and demultiplexer, adder, subtractor.

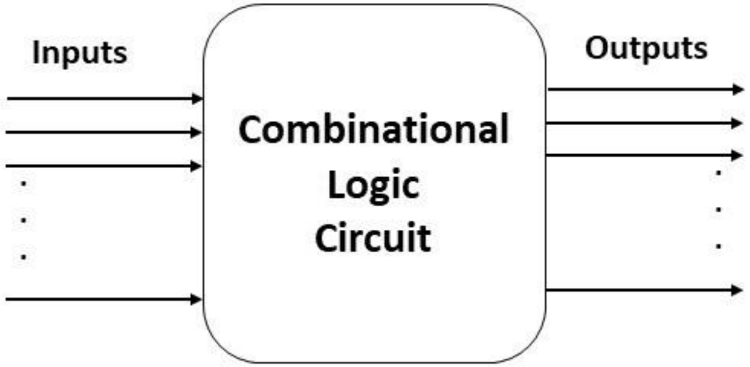
Sequential circuit

- Output of sequential circuit depends on the current input and a previous output



- Feedback is present
- Memory is present
- Ex: flipflop, registers and counters

Combinational Circuits



Connecting logical gates together to produce a specified output with no storage involved.

Combinational Circuits

Combinational circuits are divided to three main parts:

1. Arithmetic and logical function

   Adder, subtractor and comparator.

2. Code convertors

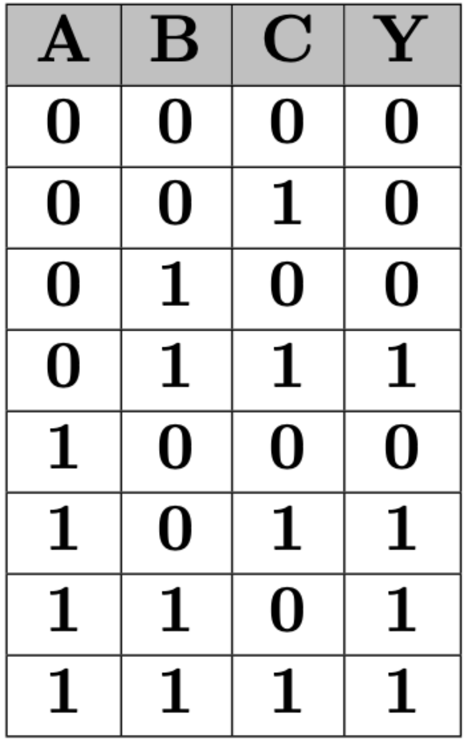
   Binary, cray and BCD

3. Data transmission

   Encoder, Decoder, Multiplexer and Demultiplexer.

Design a combinational circuit

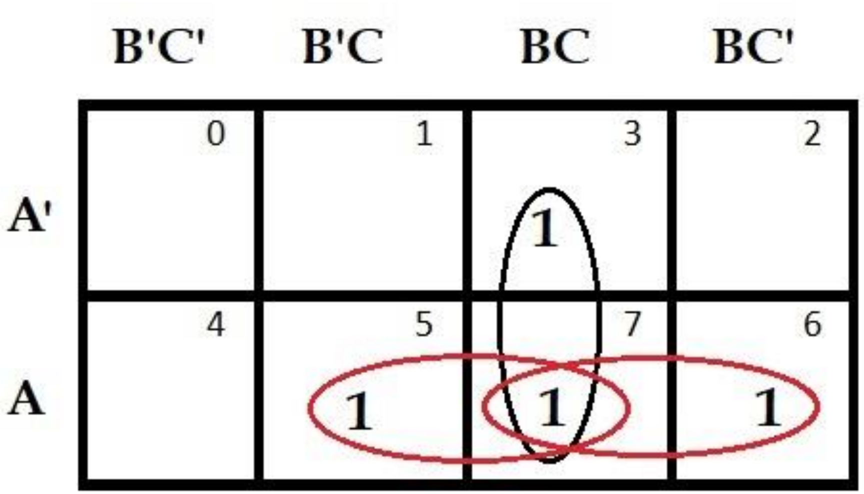
Example: Design a **combinational circuit** with three variables that will produce a logic 1 output when more than one input variable is at logic 1.



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

$$Y = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

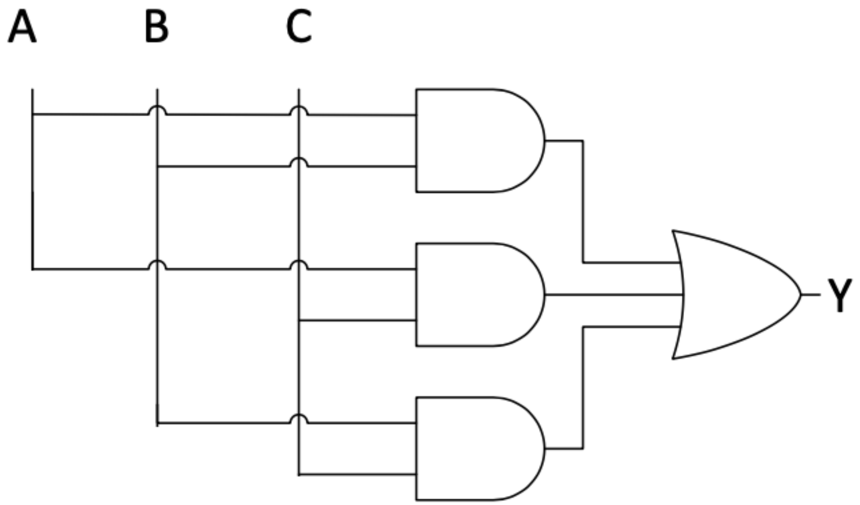
Example - continued



	B'C'	B'C	BC	BC'
A'	0	1	3 1	2
A	4	5 1	7 1	6 1

After simplification as follows:

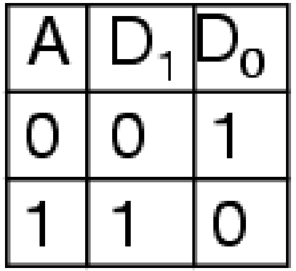
Y =AC +AB +BC



Decoder

A **decoder** is a combinational logic circuit that is used to change the code into a set of signals. It is called a decoder because it does the reverse of encoding. decoder which takes an n-digit binary number and decodes it into 2^n data lines.

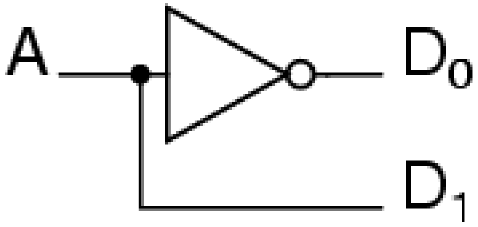
The simplest is the 1-to-2 line decoder. The truth table is:



A	D_1	D_0
0	0	1
1	1	0

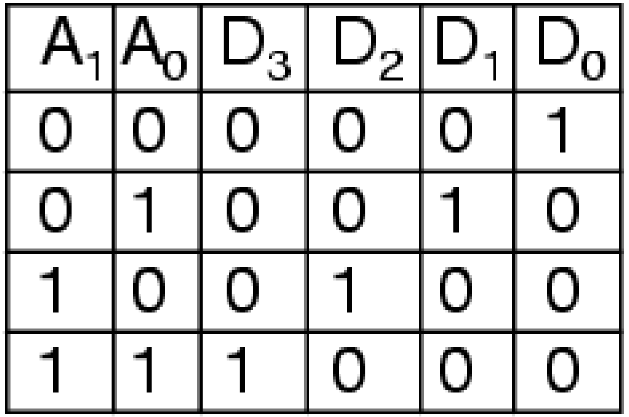
A is the address and D is the data-line. D_0 is NOT A and

D_1 is A. The circuit looks like



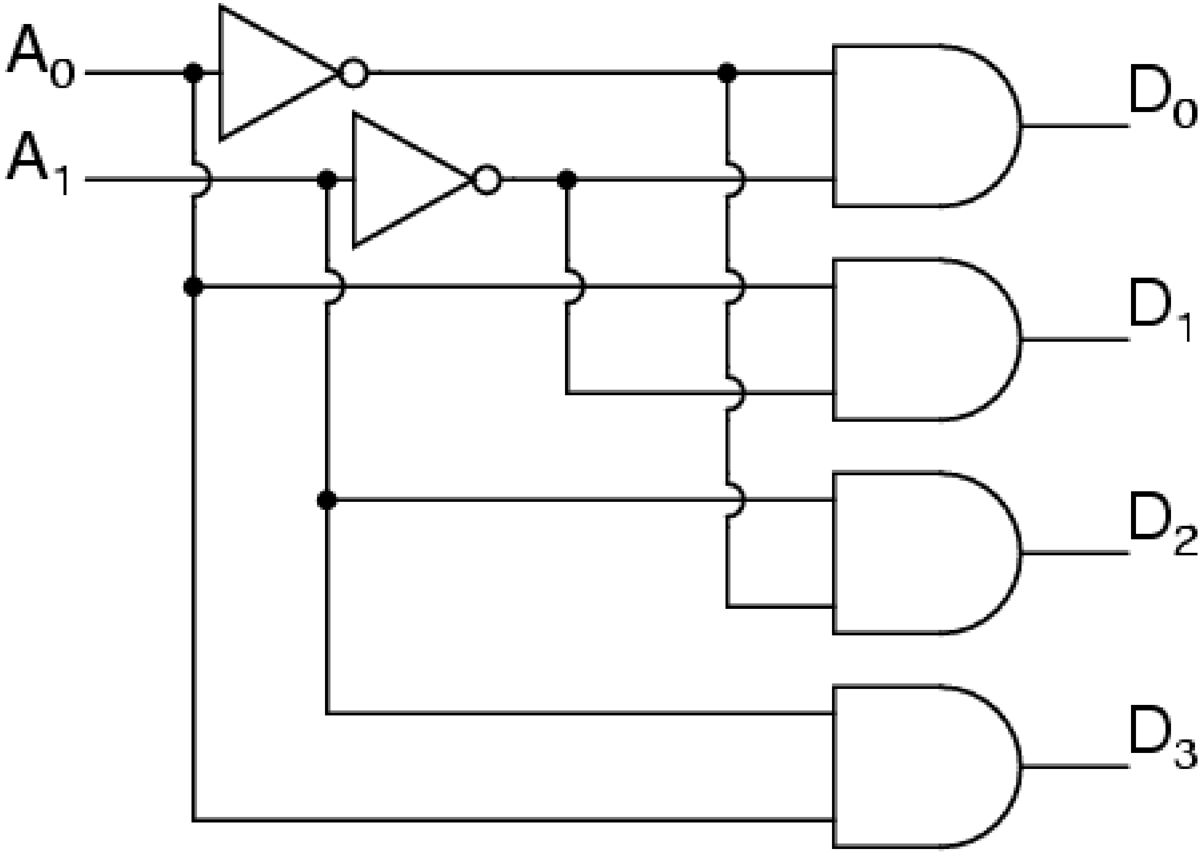
2-to-4 Line Decoder

The truth table is



A_1	A_0	D_3	D_2	D_1	D_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

The 2-to-4 decoder logic circuit is



Exercise: Build a 3-to-8 decoders using 2-to-4 decoders.

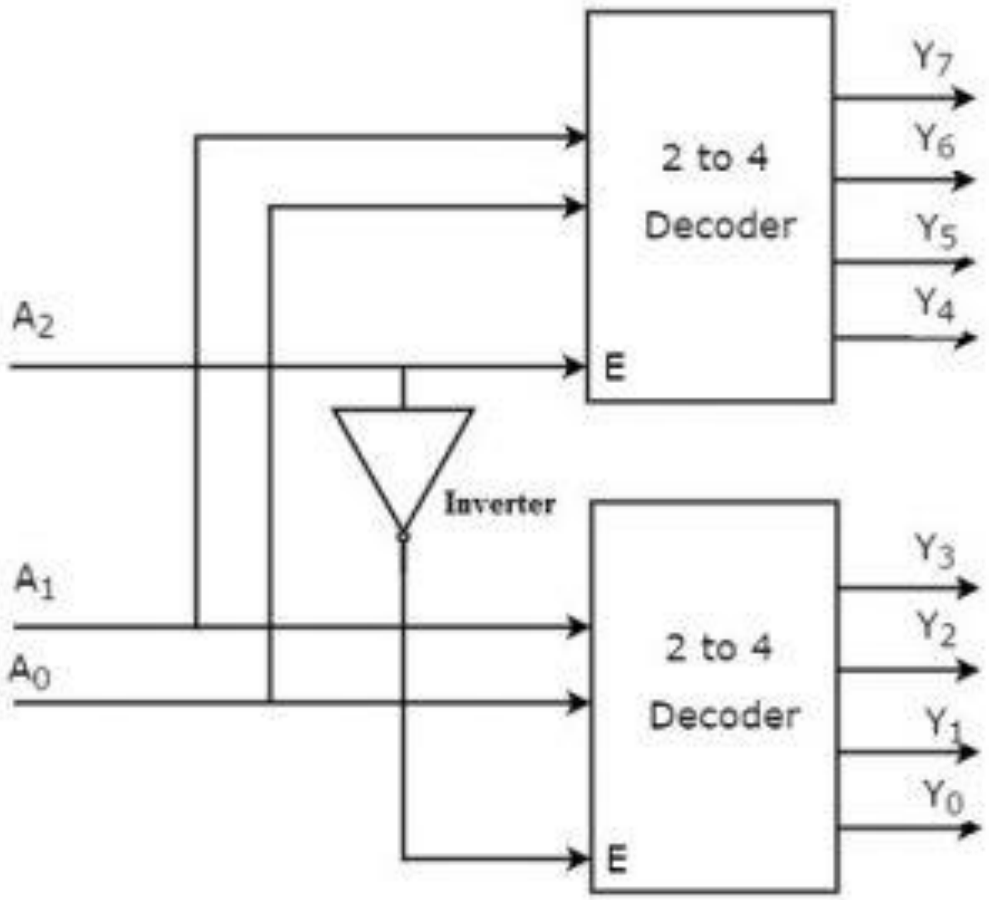
The implementation of this 3 line to 8 line decoder can be done using two 2 lines to 4 line decoders. We have discussed above that 2 to 4 line decoder includes two inputs and four outputs.

So, in 3 lines to 8 line decoder, it includes three inputs like A2, A1 & A0 and 8 outputs from Y7 – Y0. In addition to input pins, the decoder has a enable pin. This enables the pin when negated, to make the circuit inactive.

Exercise: Build a 3-to-8 decoders using 2-to-4 decoders.

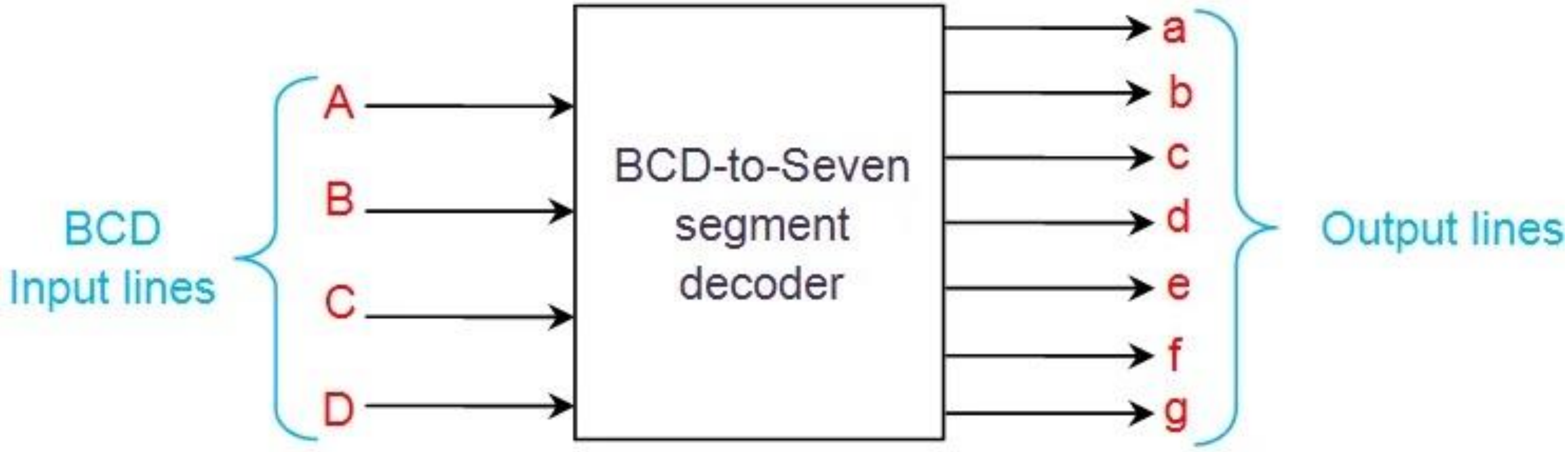
The implementation of this 3 line to 8 line decoder can be done using two 2 lines to 4 line decoders. We have discussed above that 2 to 4 line decoder includes two inputs and four outputs.

So, in 3 lines to 8 line decoder, it includes three inputs like A2, A1 & A0 and 8 outputs from Y7 – Y0. In addition to input pins, the decoder has a enable pin. This enables the pin when negated, to make the circuit inactive.



BCD - to - Seven Segment Decoder (BCD-to-Decimal)

BCD to seven segment decoder is a circuit used to convert the input BCD into a form suitable for the display. It has four input lines (A, B, C and D) and 7 output lines (a, b, c, d, e, f and g) as shown in the Figure bellow:



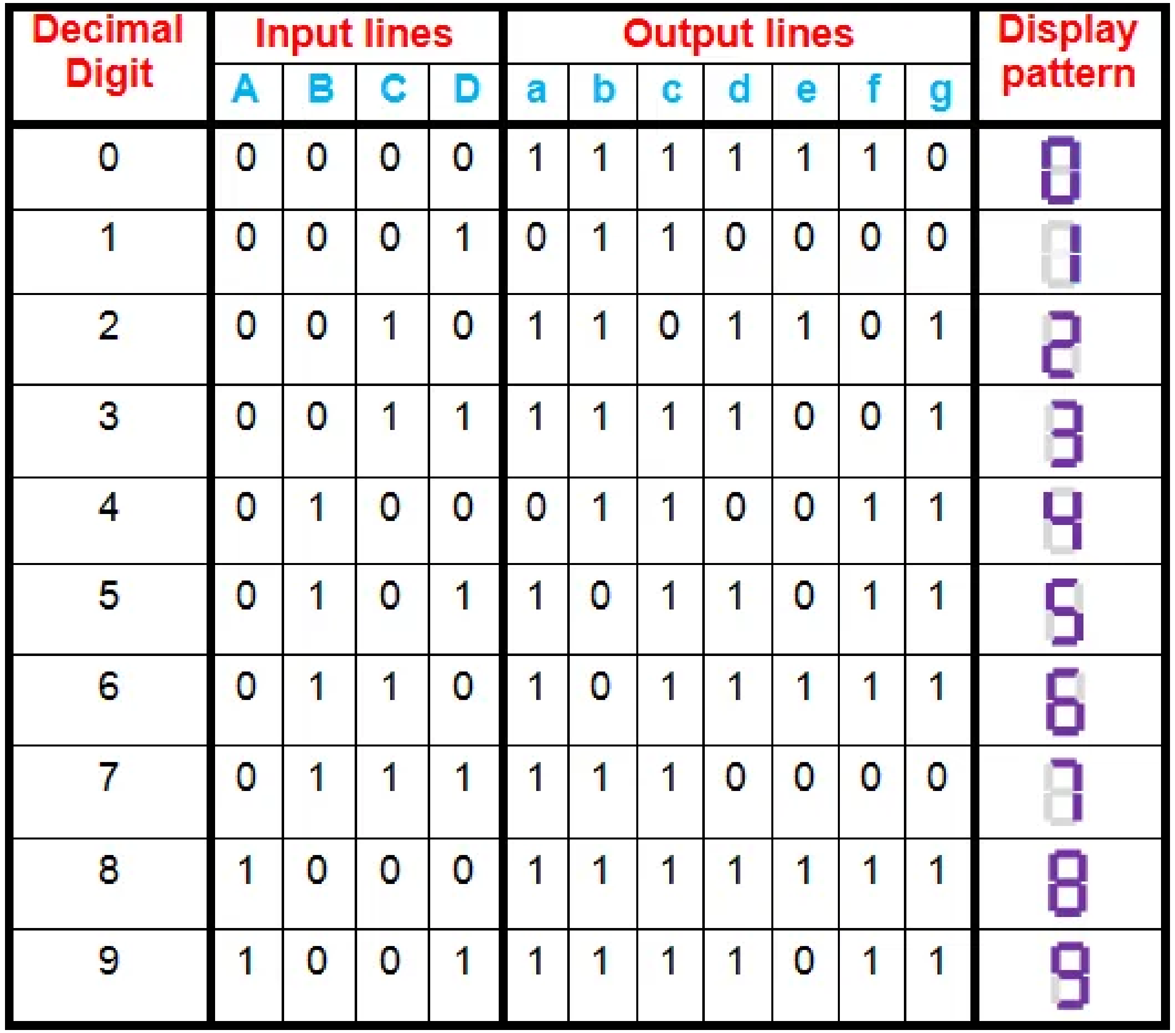
BCD-to-Seven segment decoder

BCD - to - Seven Segment Decoder

This table indicates the segments which are to be driven high to obtain certain decimal digit at the output of the seven-segment display.

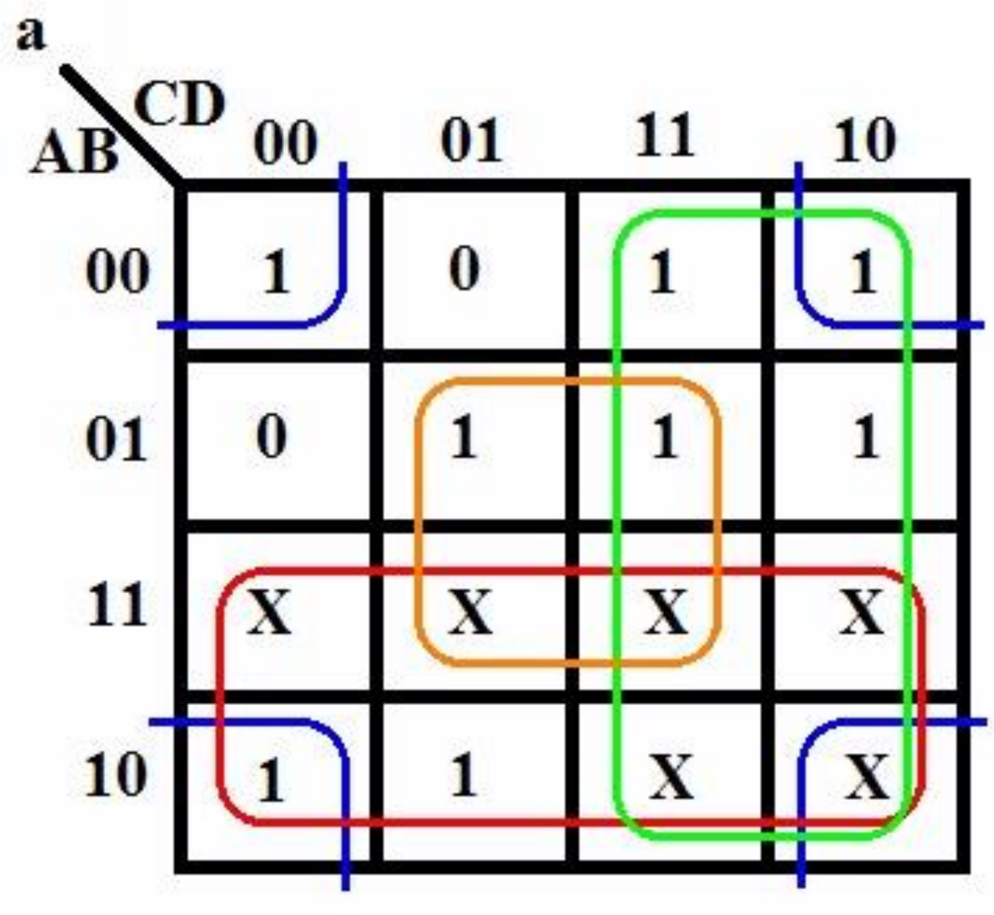
Explanation:

For combination where all the inputs (A, B, C and D) are zero; the output lines are a = 1, b = 1, c = 1, d = 1, e = 1, f = 1 and g = 0. As a result, 7 segment display shows ‘zero’ as output.



Decimal Digit	Input lines				Output lines							Display pattern
	A	B	C	D	a	b	c	d	e	f	g	
0	0	0	0	0	1	1	1	1	1	1	0	0
1	0	0	0	1	0	1	1	0	0	0	0	1
2	0	0	1	0	1	1	0	1	1	0	1	2
3	0	0	1	1	1	1	1	1	0	0	1	3
4	0	1	0	0	0	1	1	0	0	1	1	4
5	0	1	0	1	1	0	1	1	0	1	1	5
6	0	1	1	0	1	0	1	1	1	1	1	6
7	0	1	1	1	1	1	1	0	0	0	0	7
8	1	0	0	0	1	1	1	1	1	1	1	8
9	1	0	0	1	1	1	1	1	0	1	1	9

K- map for a

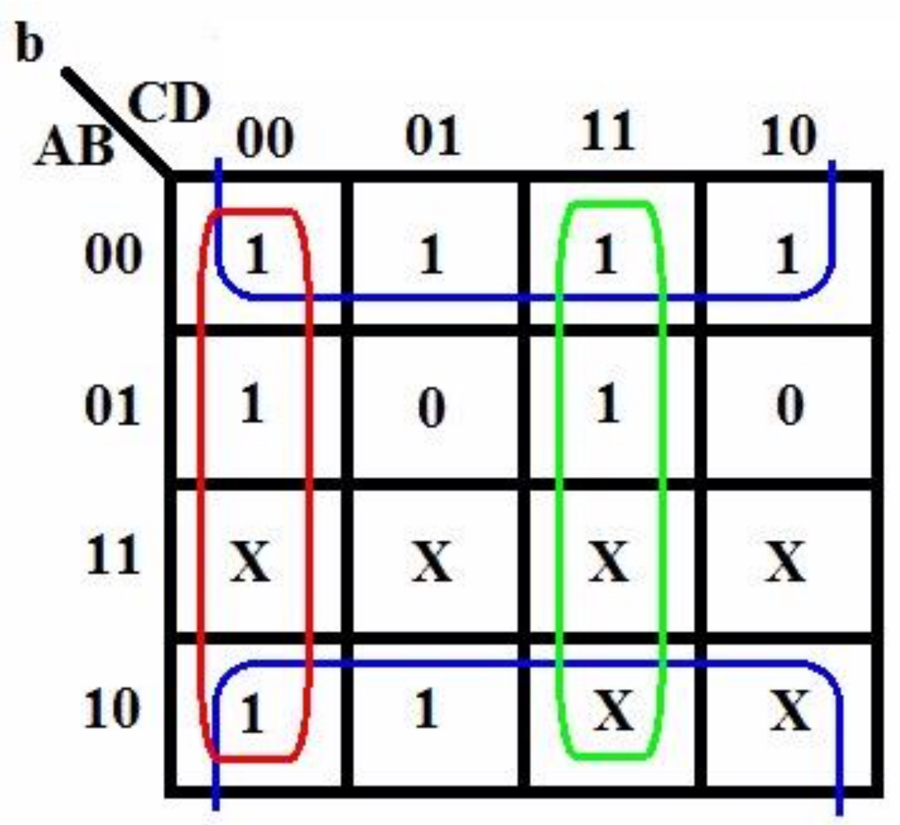


a AB \ CD	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	X	X	X	X
10	1	1	X	X

$$a = A + C + BD + \bar{B}\bar{D}$$

K- map for b

$b = \bar{B} + \bar{C}\bar{D} + C\,D$

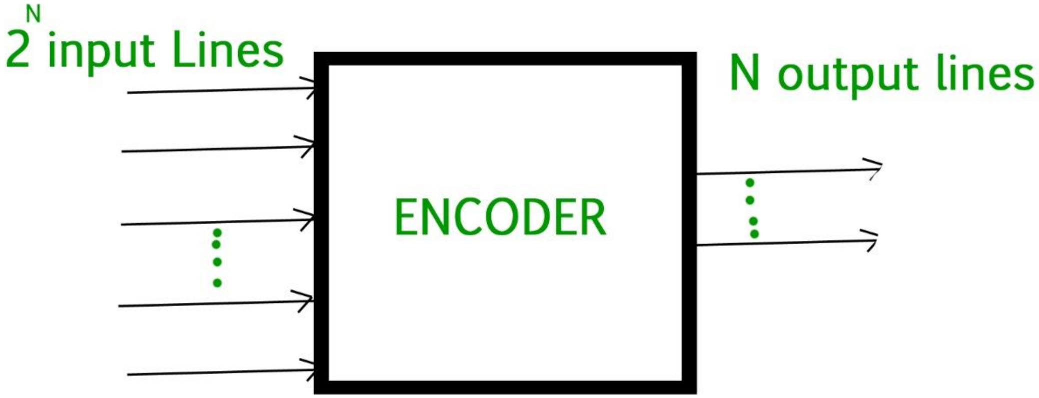


AB CD	00	01	11	10
00	1	1	1	1
01	1	0	1	0
11	X	X	X	X
10	1	1	X	X

Home work: k-map for c, d, e, f, g

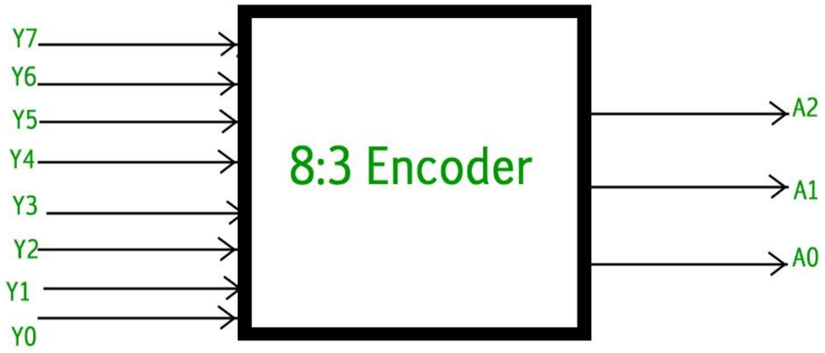
Encoder

An Encoder is a **combinational circuit** that performs the reverse operation of Decoder. It has maximum of **2^n input lines** and **‘n’ output lines**.



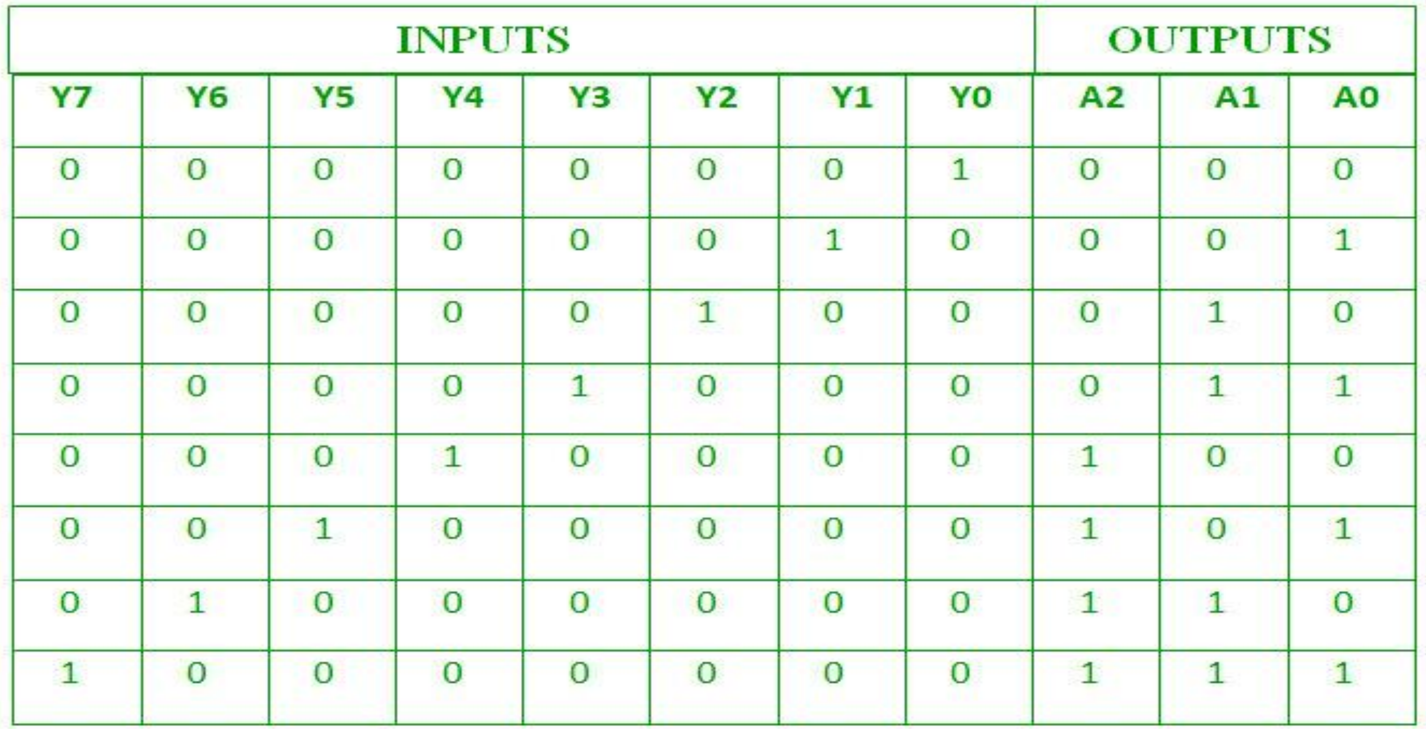
8 : 3 Encoder (Octal to Binary)

The 8 to 3 Encoder or octal to Binary encoder consists of **8 inputs** : Y7 to Y0 and **3 outputs** : A2, A1 & A0. Each input line corresponds to each octal digit and three outputs generate corresponding binary code.



8 : 3 Encoder truth table

The truth table for 8 to 3 encoder is as follows

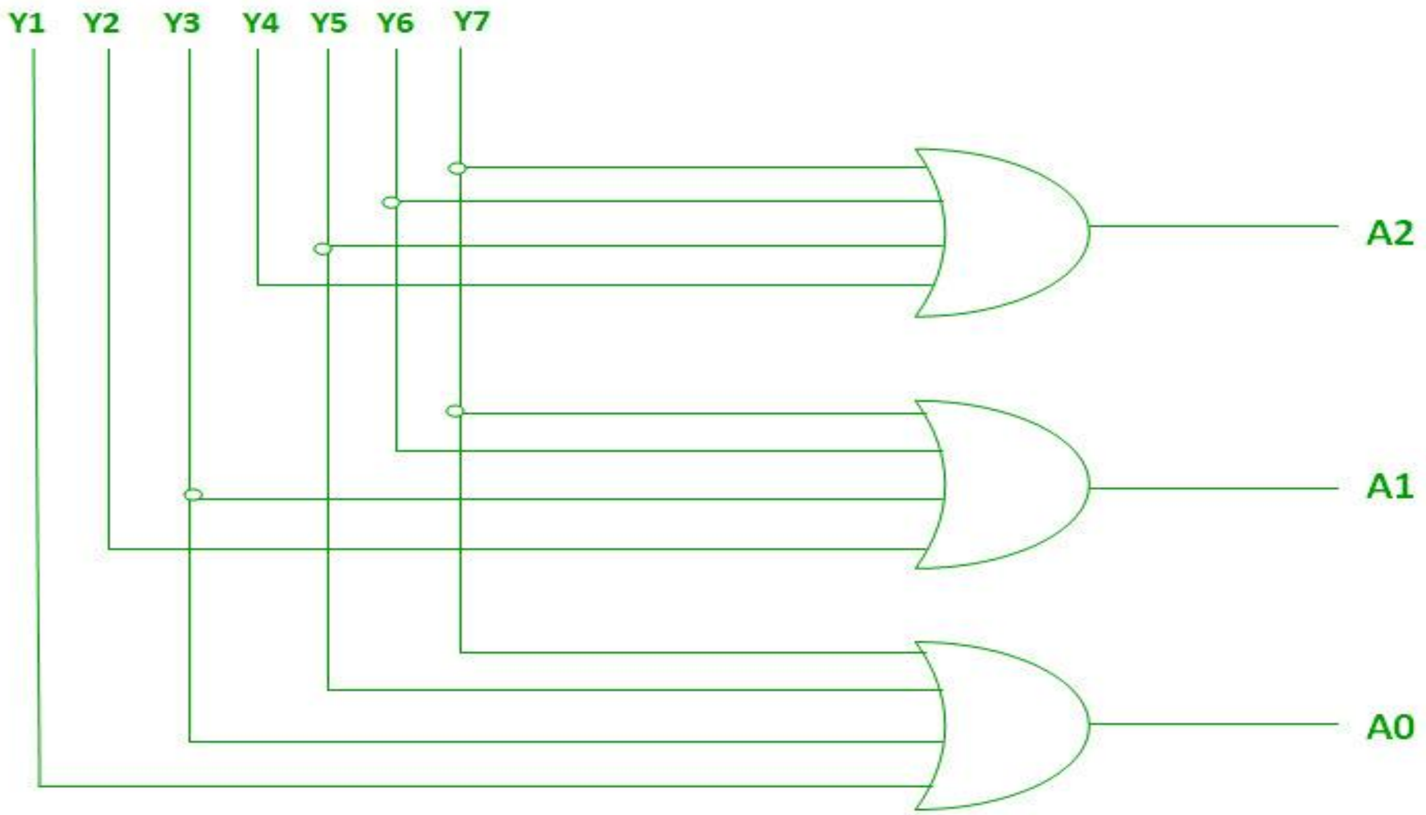


INPUTS								OUTPUTS		
Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	A2	A1	A0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

A2 = Y7 + Y6 + Y5 + Y4
A1 = Y7 + Y6 + Y3 + Y2
A0 = Y7 + Y5 + Y3 + Y1

8 : 3 Encoder implementation

The above Boolean functions A2, A1 and A0 can be implemented using four input OR gates :


Y1
Y2
Y3
Y4
Y5
Y6
Y7
A2
A1
A0