0.1 Design of Circuits with Limited Gate Fan-In

Fan-in: ***Fan-in is a term that defines the maximum number of digital inputs that a single logic gate can accept.***
Most of the logic gates have one, two, or three inputs, although some have more than three inputs.

Realize the given functions using only two input NAND gates and inverters.

$$\begin{aligned} f_1 &= b'c' + ab' + a'b \\ f_2 &= b'c' + bc + a'b \\ f_3 &= a'b'c + ab + bc' \end{aligned}$$

Solution: For the given function it requires a three-input OR gate.

$$\begin{aligned} f_1 &= b'(a + c') + a'b \\ f_2 &= b(a' + c) + b'c' \\ f_3 &= a'b'c + b(a + c') \\ &= a'(b'c) + b(a + c') \\ &= a'(b + c')' + b(a + c') \end{aligned}$$

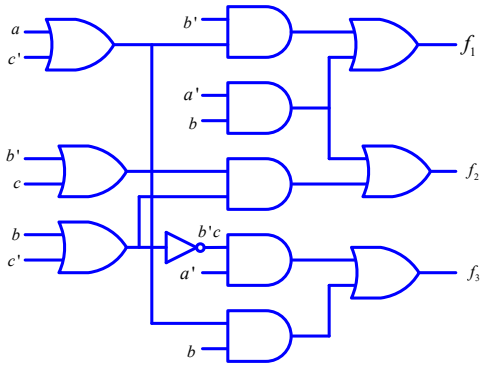


Figure 1

$$\begin{aligned} f_1 &= b'(a + c') + a'b \\ f_2 &= b(a' + c) + b'c' \\ f_3 &= a'b'c + b(a + c') \\ &= a'(b'c) + b(a + c') \\ &= a'(b + c')' + b(a + c') \end{aligned}$$

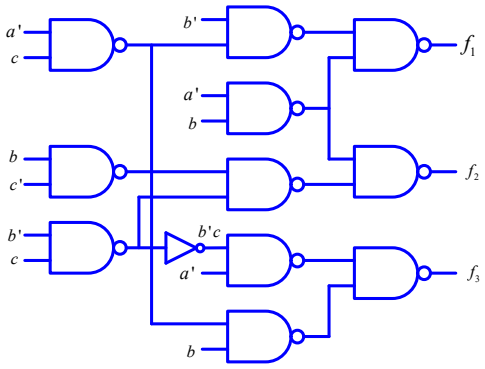


Figure 2

Realize the given functions using only three input NOR gates

$$f(a, b, c, d) = \sum m(0, 3, 4, 5, 8, 9, 10, 14, 15)$$

Solution: For the given function it requires a three-input OR gate.

$$
\begin{aligned}
f' &= a'b'c'd + ab'cd + abc' + a'bc + a'cd' \\
f' &= b'd(a'c' + ac) + a'c'(b + d') + abc' \\
f &= [b + d'(a + c)(a' + c')][a + c + b'd][a' + b'c]
\end{aligned}
$$

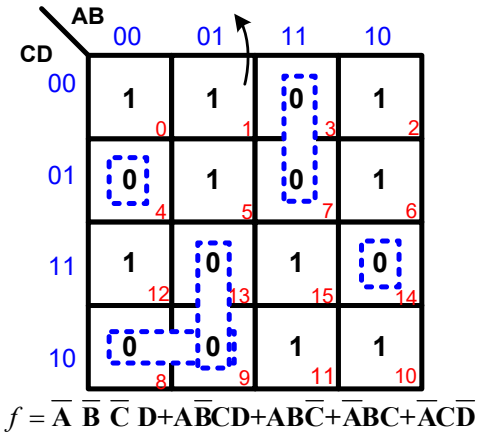


Figure 3

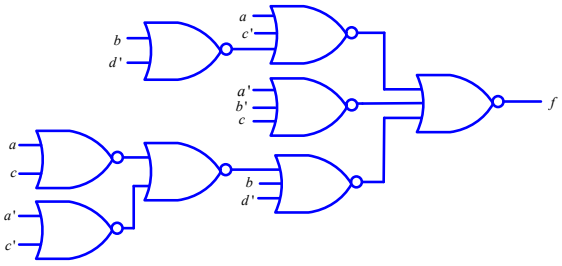


Figure 4

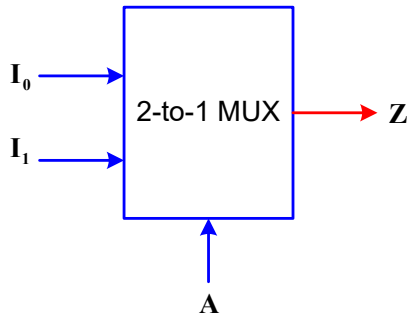
0.2 Multiplexer

The multiplexer, is a combinational logic circuit which has 2^n input lines and has a single output line. designed to switch one of several through to a line by the application of a control signal. Based on the values of the selection lines, one of these data inputs is connected to the output line. Multiplexers are mainly used to send the multiple source data through a common line with certain amount of time.

There are various types of the multiplexer which are as follows:

0.2.1 2:1 Multiplexer

The 2:1 multiplexer, consists of two input lines, I_0 and I_1, and has a one selector line, A with single output line Z. The 2:1 multiplexer block diagram is as shown in Figure 7 and corresponding truth table giving the details of the output based on selection line is as shown in the Table. The corresponding logic circuit diagram is as shown in Figure 6



Input	Output
A	Z
0	I_0
1	I_1

Figure 5

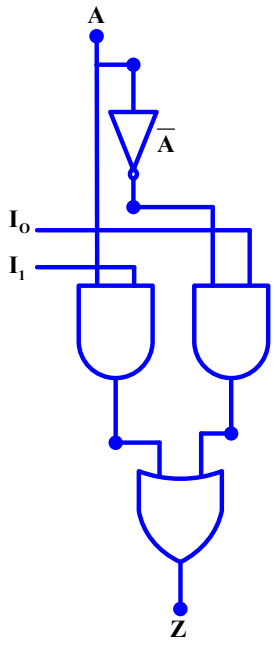


Figure 6

0.2.2 4:1 Multiplexer

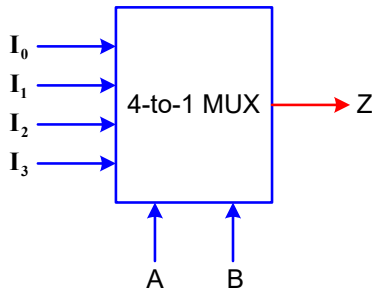


Figure 7

Inputs		Output
A	B	Z
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

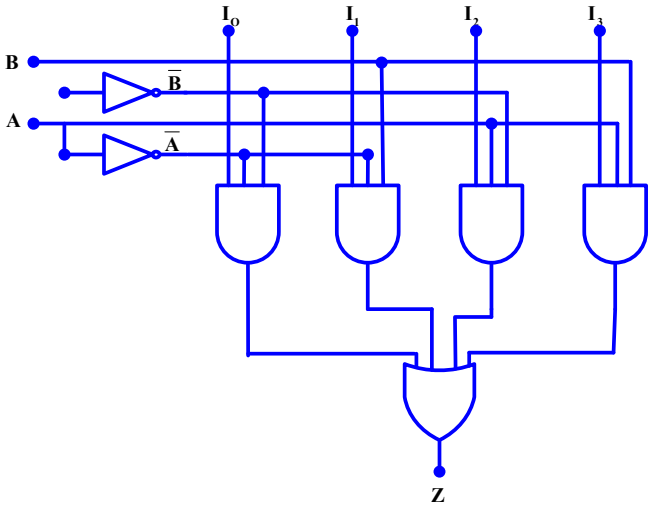


Figure 8

0.2.3 8:1 Multiplexer

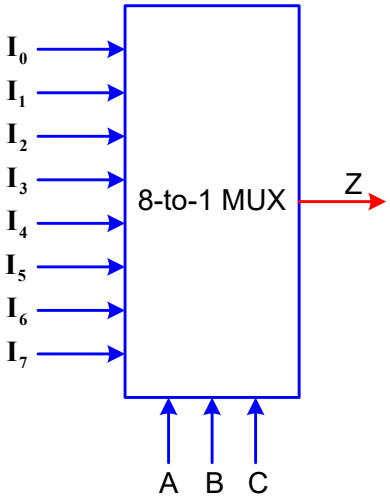


Figure 9

Inputs			Output
A	B	C	Z
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7

0.2.4 16:1 Multiplexer using 8:1 MUX

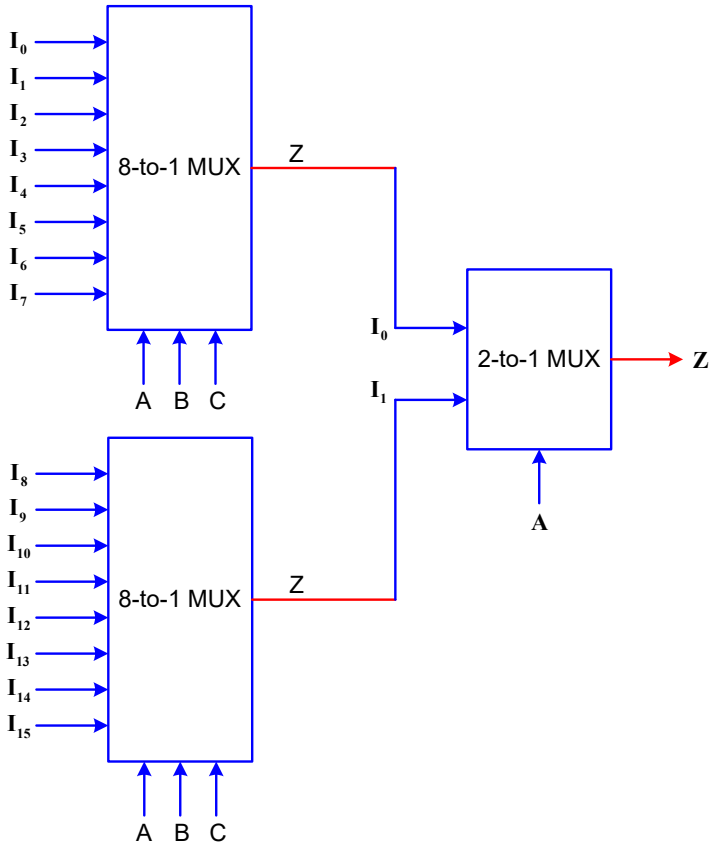


Figure 10

0.2.5 16:1 Multiplexer using 4:1 MUX

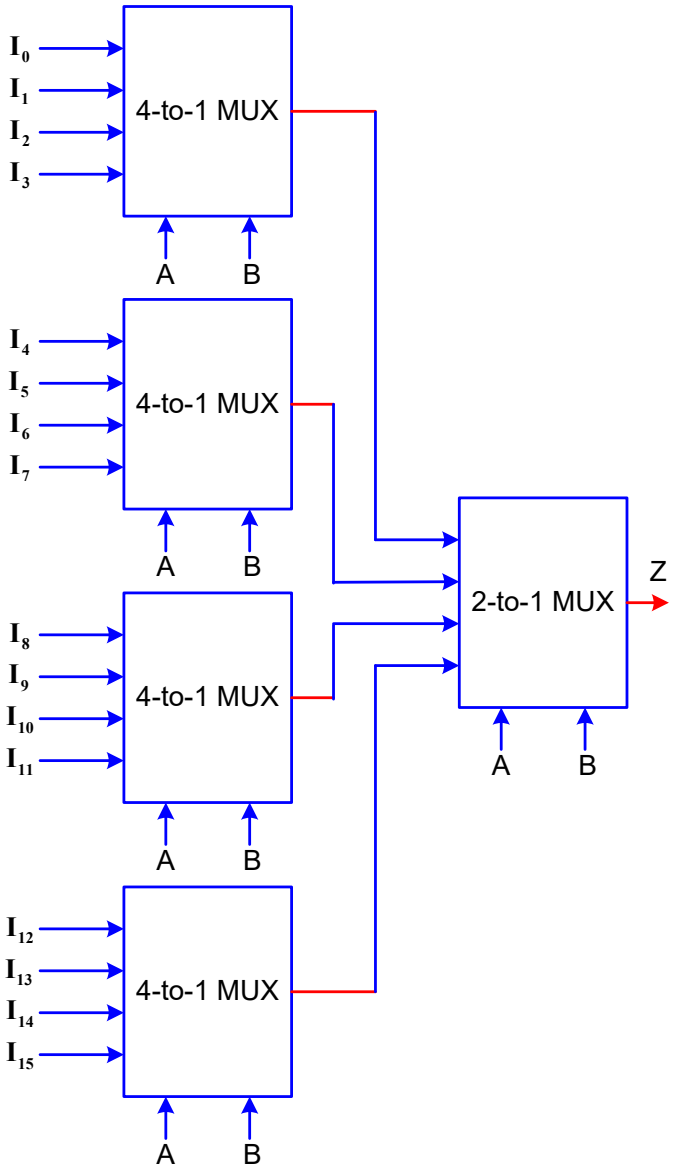


Figure 11

Q1. Realize the following function using 8:1 Mux

$$f(x, y, x) = \sum m(1, 4, 5, 7)$$

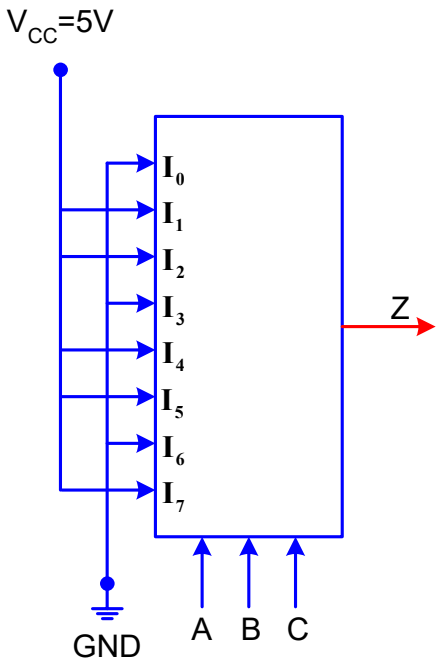


Figure 12

Q1. Realize the following function using 8:1 Mux and 4:1 Mux

$$f(x, y, x) = \sum m(0, 2, 3, 5)$$

Decimal	Inputs			Output
	A	B	C	Z
0	0	0	0	1
1	0	0	1	0
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	A	B	C	0

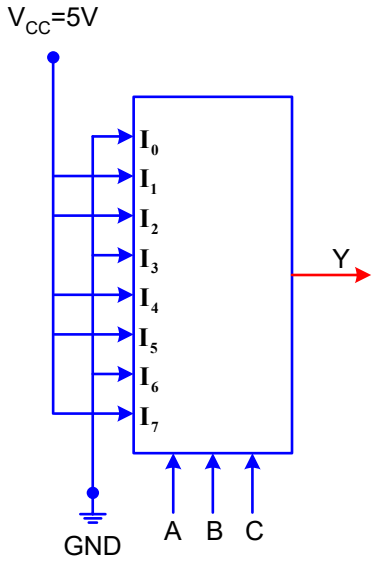


Figure 13

$$
\begin{aligned}
y &= \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,B\,\overline{C} + \overline{A}\,B\,C + A\,\overline{B}\,C \\
&= \overline{A}\,\overline{B}\,(\overline{C}) + \overline{A}\,B\,(C + \overline{C}) + A\,\overline{B}\,C \\
&= \overline{A}\,\overline{B}\,(\overline{C}) + \overline{A}\,B\,(1) + A\,\overline{B}\,C + AB(0)
\end{aligned}
$$

Inputs		Output
A	B	Z
0	0	$I_0 = \overline{C}$
0	1	$I_1 = 1$
1	0	$I_2 = C$
1	1	$I_3 = 0$

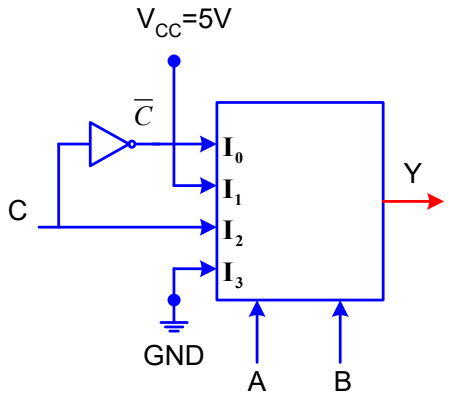


Figure 14

2020-Jan Realize the following function using 8:1 Mux

$$f(A, B, C, D) = \sum m(1, 2, 5, 6, 9, 12)$$

Solution:

	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
$\overline{A}$	0	(1)	(2)	3	4	(5)	(6)	7
A	8	(9)	10	11	(12)	13	14	15
	0	1	$\overline{A}$	0	A	$\overline{A}$	$\overline{A}$	0

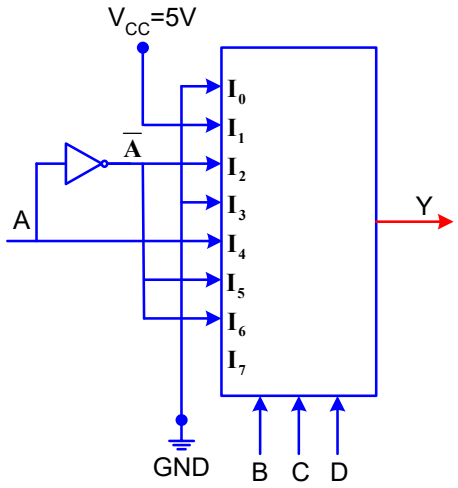


Figure 15

Q3. Implement following function using 8:1 MUX and logic gates

$$f(A, B, C, D) = \sum m(0, 3, 4, 7, 8, 9, 13, 14)$$

Solution:

	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
$\overline{A}$	(0)	1	2	(3)	(4)	5	6	(7)
A	(8)	(9)	10	11	12	(13)	(14)	15
	1	A	0	$\overline{A}$	$\overline{A}$	A	A	$\overline{A}$

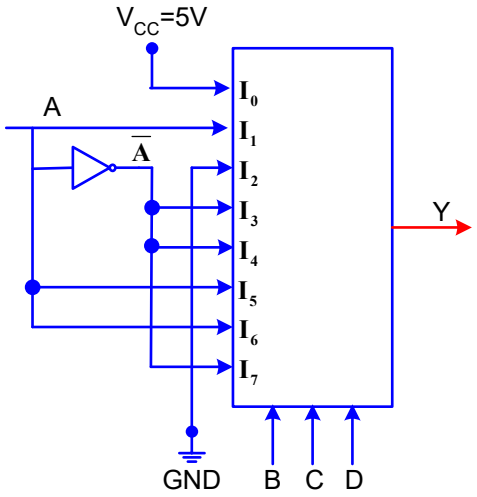


Figure 16

Q4. How do you implement following function using 8:1 MUX

$$f(A,B,C,D) = \sum m(0,2,6,10,12,13) + \sum d(3,8,14)$$

Solution:

Without don't care

	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
$\overline{A}$	(0)	1	(2)	3	4	5	(6)	7
A	8	9	(10)	(11)	(12)	(13)	14	15
	$\overline{A}$	0	1	A	A	A	$\overline{A}$	0

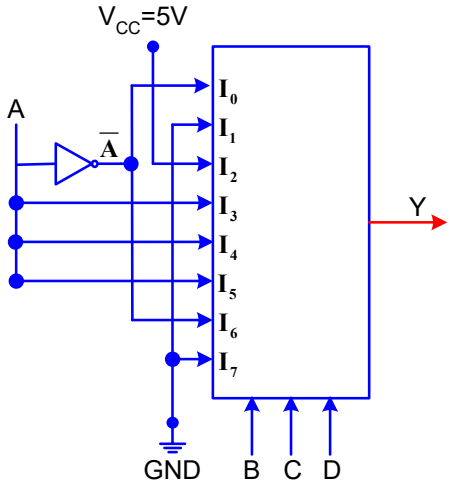


Figure 17

With don't care

	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
$\overline{A}$	(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

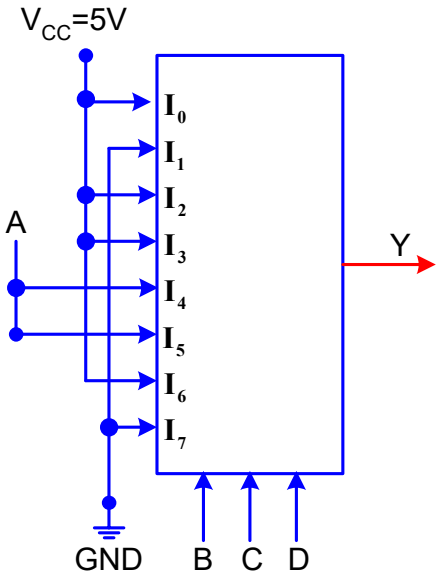


Figure 18

Q6. Implement the following function using 4:1 MUX

$$f(A,B,C,D) = \sum m(0,1,3,4,8,9,15)$$

Solution:

	I_0	I_1	I_2	I_3
$\overline{A}\ \overline{B}$	(0)	(1)	2	$\overline{3}$
$\overline{A}\ B$	(4)	5	6	7
$A\ \overline{B}$	(8)	(9)	10	11
$A\ B$	12	13	14	(15)

$$\begin{aligned} I_0 &= \overline{A}\ \overline{B} + \overline{A}\ B + A\ \overline{B} \\ &= \overline{A}(\overline{B}+B) + A\ \overline{B} = \overline{A} + A\ \overline{B} \\ I_1 &= \overline{A}\ \overline{B} + A\overline{B} \\ &= \overline{B}(\overline{A}+A) = \overline{B} \\ I_2 &= 0 \\ I_3 &= \overline{A}\ \overline{B} + AB = \overline{A \oplus B} \end{aligned}$$

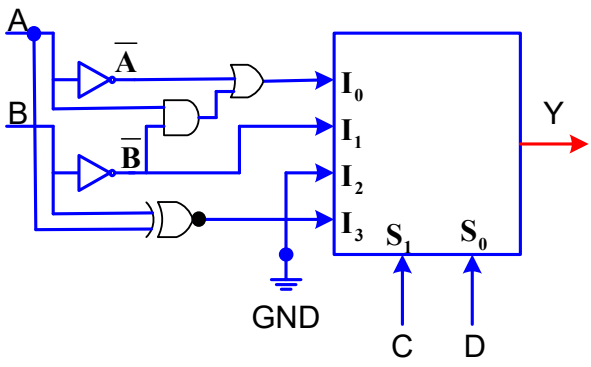


Figure 19

Q7. Implement the following function using 4:1 MUX

$$f(A,B,C) = \sum m(1,3,5,6)$$

Solution:

	I_0	I_1	I_2	I_3
$\overline{A}$	0	(1)	2	(3)
A	4	(5)	(6)	7
	0	1	A	(A)

$$I_0 = 0$$
$$I_1 = 1$$
$$I_2 = A$$
$$I_3 = \overline{A}$$

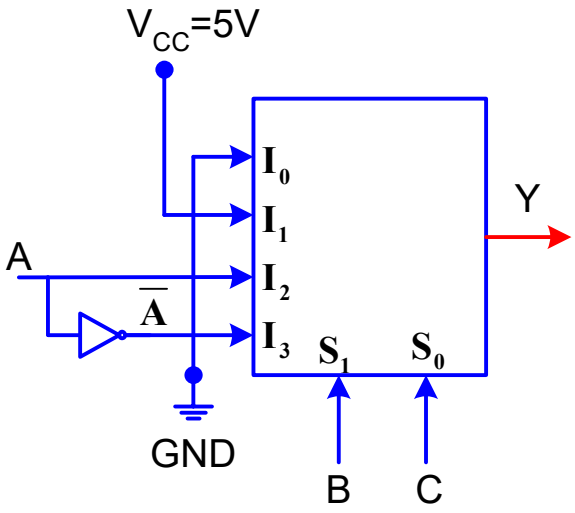


Figure 20

2020-Jan. Implement $f(A,B,C,D) = \sum m(0,2,6,10,12,13)$ **using**

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

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

Solution:

	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
$\overline{D}$	(0)	1	(2)	3	4	5	(6)	7
D	8	9	(10)	11	(12)	(13)	14	15
	$\overline{D}$	0	1	0	D	D	$\overline{D}$	0

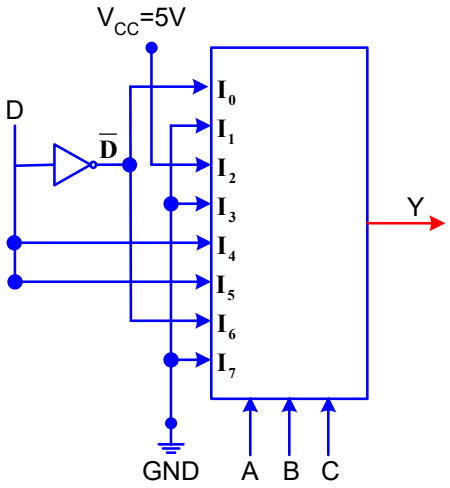


Figure 21

With don't care

	I_0	I_1	I_2	I_3
$\overline{C}\ \overline{D}$	(0)	1	(2)	3
$\overline{C}\ D$	4	5	(6)	7
$C\ \overline{D}$	8	9	(10)	11
$C\ D$	(12)	(13)	14	15

$$
\begin{aligned}
I_0 &= \overline{C}\ \overline{D} + CD = \overline{C \oplus D} \\
I_1 &= C\ D \\
I_2 &= \overline{C}\ \overline{D} + \overline{C}D + C\overline{D} \\
&= \overline{C}(\overline{D} + D) + C\overline{D} = C + C\overline{D} \\
&= = C(1 + D) = C \\
I_3 &= 0
\end{aligned}
$$

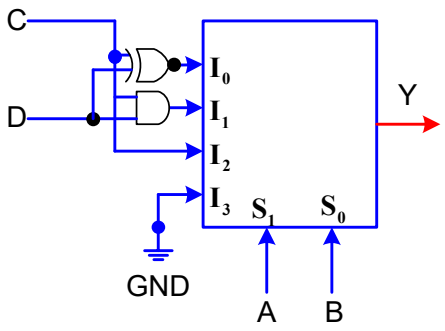


Figure 22

Q Model question Paper. Implement the following function using 4:1 multiplexer $f(A, B, C, D) = \sum m(0, 1, 2, 4, 6, 9, 12, 14)$

Solution:

	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
$\overline{A}$	(0)	(1)	(2)	3	(4)	5	(6)	7
A	8	(9)	10	11	(12)	13	(14)	15
	$\overline{A}$	1	$\overline{A}$	0	1	0	1	0

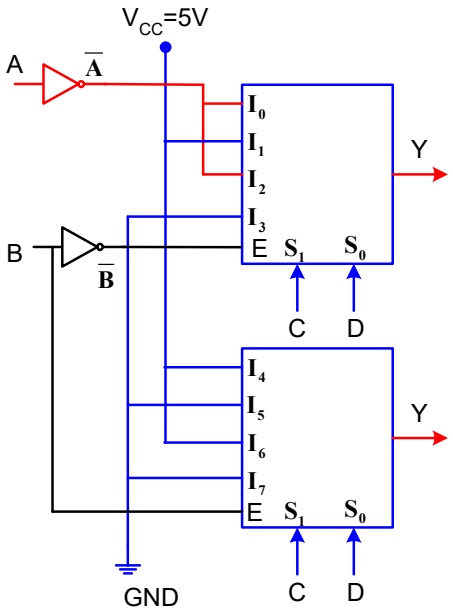


Figure 23

	I_0	I_1	I_2	I_3
$\overline{A}\ \overline{B}$	(0)	(1)	(2)	3
$\overline{A}\ B$	(4)	5	(6)	7
$A\ \overline{B}$	8	(9)	10	11
$A\ B$	(12)	13	(14)	15

$$
\begin{aligned}
I_0 &= \overline{A}\ \overline{B} + \overline{A}\ B + AB \\
&= \overline{A}(\overline{B} + B) + AB = \overline{A} + AB \\
I_1 &= \overline{A}\ \overline{B} + A\ \overline{B} = \overline{B}(\overline{A} + A) \\
&= \overline{B} \\
I_2 &= \overline{A}\ \overline{B} + \overline{A}\ B + AB \\
&= \overline{A}(\overline{B} + B) + AB = \overline{A} + AB \\
I_3 &= 0
\end{aligned}
$$

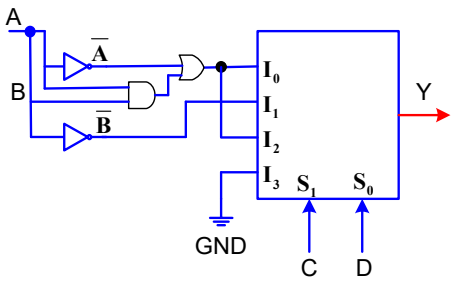


Figure 24

Q Model question Paper. Implement the following boolean function using 8:1 multiplexer $f(A,B,C,D) = \prod m(1,2,5,9,12)$

Solution:

	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
$\overline{A}$	(0)	1	2	(3)	(4)	5	6	(7)
A	8	9	(10)	(11)	12	(13)	(14)	(15)
	$\overline{A}$	0	A	1	$\overline{A}$	A	A	1

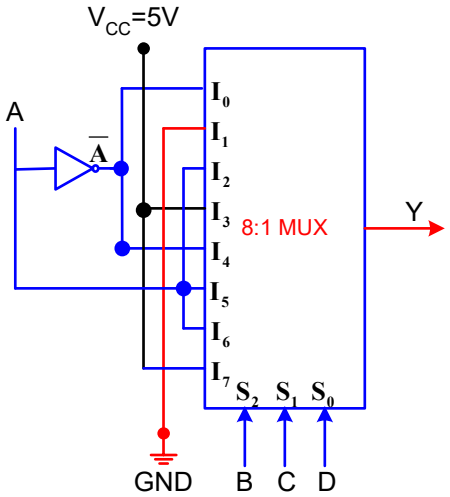


Figure 25

2017-JAN Model question Paper. Implement the following boolean function using 8:1 multiplexer $f(A,B,C,D) = \sum m(0,1,5,6,8,10,12,15)$

Solution:

	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
$\overline{A}$	(0)	(1)	2	3	4	(5)	(6)	7
A	(8)	9	(10)	11	(12)	13	14	(15)
	1	$\overline{A}$	A	0	$\overline{A}$	A	A	A

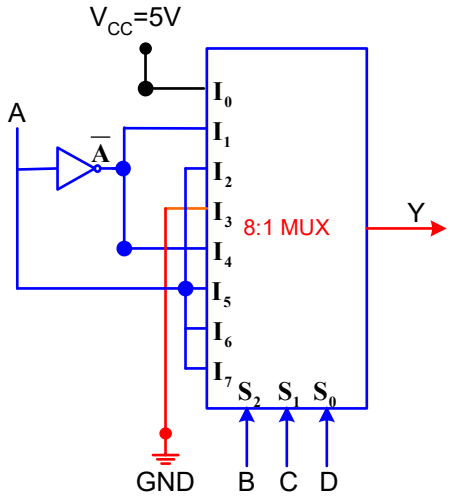


Figure 26

2020 Sept Construct 32:1 multiplexer using 8:1 MUX and decoder

Solution:

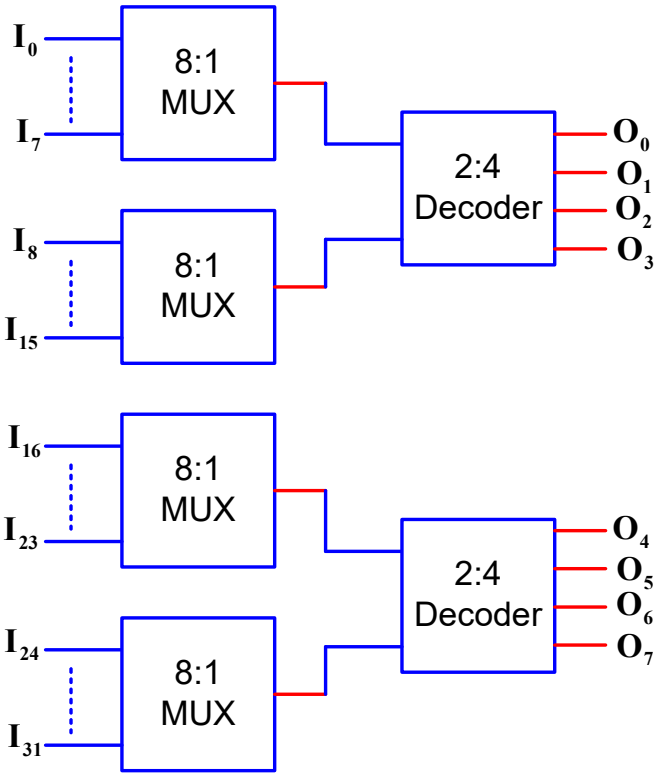


Figure 27

Design a 32:1 multiplexer using 16:1 MUX and 2:1 MUX

Solution:

Five variables are required to implement 32:1 multiplexer. Consider a five variables as A, B, C, D, E. To implement this A, B, C, D variables are used for 32:1 multiplexer and E is for 2:1 multiplexer. The details of the implementation is as shown in Figure 28

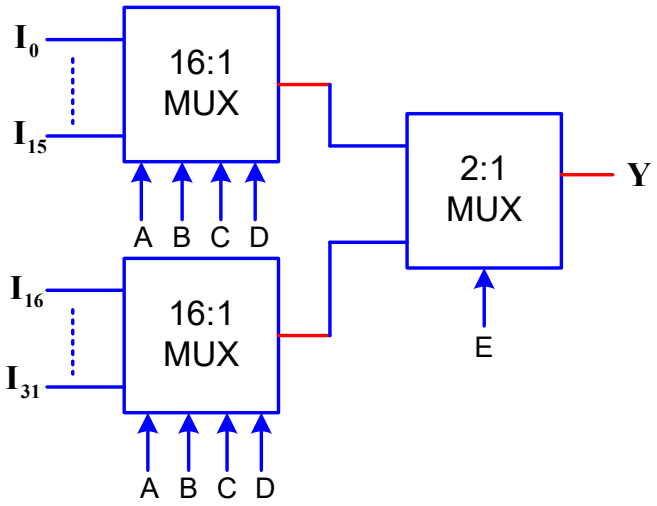


Figure 28

Design a 1:32 demultiplexer using 1:16 demultiplexer.

Solution:

Five variables are required to implement 1:32 demultiplexer. Consider a five variables as A, B, C, D, E. To implement this A, B, C, D variables are used for 1:32 demultiplexer and E and $\overline{E}$ to select anyone demux . The details of the implementation is as shown in Figure 29

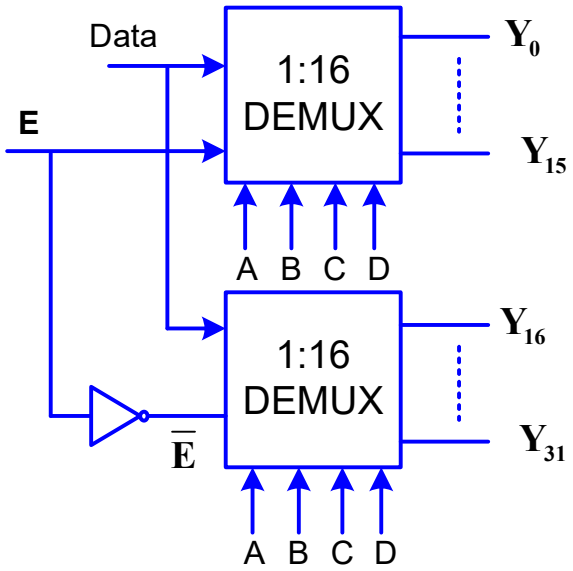


Figure 29

0.3 Decoders

2-4 Line Decoder

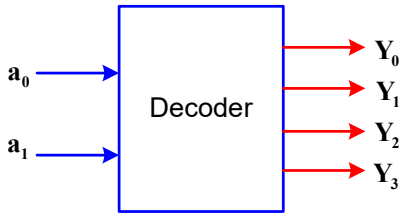


Figure 30

Truth Table

Inputs		Outputs			
a_1	a_0	Y_0	Y_1	Y_2	Y_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

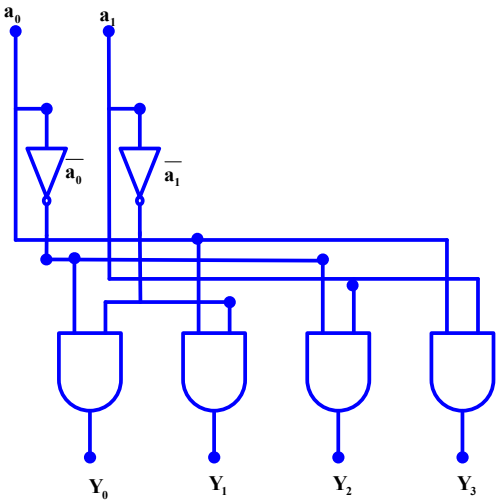


Figure 31

3-8 Line Decoder

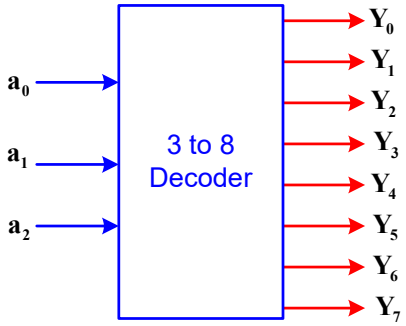


Figure 32

Inputs			Outputs							
a_2	a_1	a_0	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
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

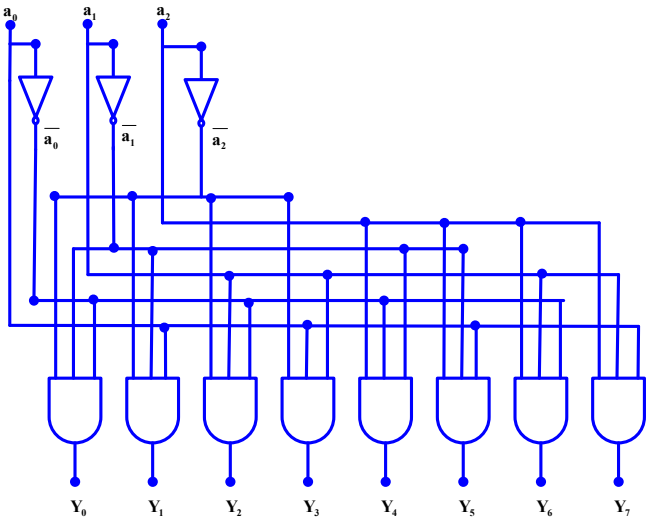


Figure 33

Implement the following Boolean expressions using 3-8 Line Decoder

$$
\begin{aligned}
F_1(A,B,C) &= \sum m(0,4,6) \\
F_2(A,B,C) &= \sum m(0,5) \\
F_3(A,B,C) &= \sum m(1,2,3,7)
\end{aligned}
$$

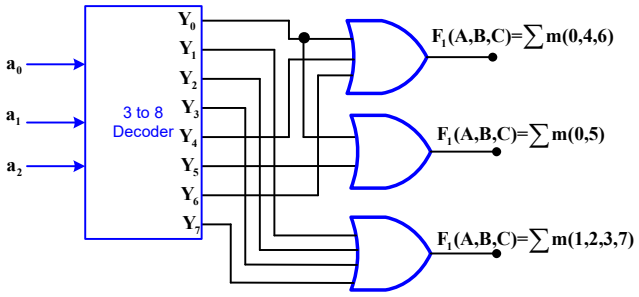


Figure 34

Implement the following Boolean expressions using 3-8 Line Decoder

$$
\begin{aligned}
F_1(A,B,C) &= \sum m(0,4,6,7) \\
F_2(A,B,C) &= \sum m(1,4,5)
\end{aligned}
$$

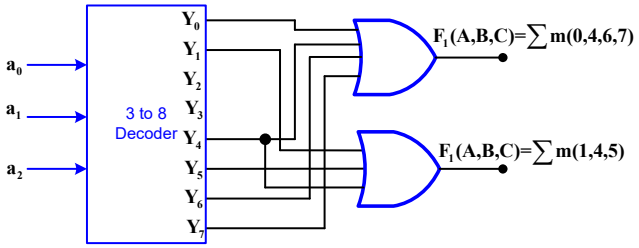


Figure 35

0.3.1 Decoder with Enable Input

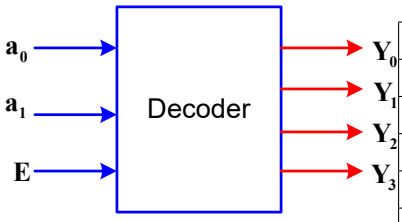


Figure 36

Truth Table

Inputs			Outputs			
E	a_1	a_0	Y_0	Y_1	Y_2	Y_3
0	X	X	0	0	0	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	0	1

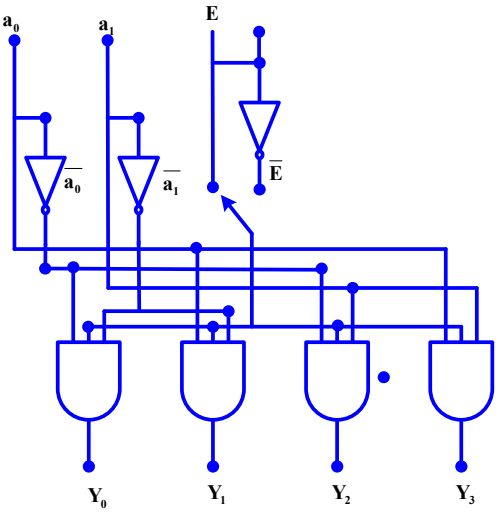


Figure 37

Q1. Implement full adder using 3:1 decoder

Truth Table

A	B	Cin	Sum	Carry	Y
0	0	0	0	0	Y_0
0	0	1	1	0	Y_1
0	1	0	1	0	Y_2
0	1	1	0	1	Y_3
1	0	0	1	0	Y_4
1	0	1	0	1	Y_5
1	1	0	0	1	Y_6
1	1	1	1	1	Y_7

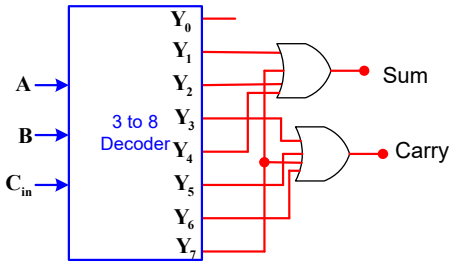


Figure 38

$$Sum = Y_1 + Y_2 + Y_4 + Y_7$$
$$Carry = Y_3 + Y_5 + Y_6 + Y_7$$

0.4 Programmable Logic Arrays (PLA):

A programmable logic array (PLA) is a type of logic device that can be programmed to implement various kinds of combinational logic circuits. It's a type of PLD, which has both programmable AND array and programmable OR array. PLA is used for the implementation of various combinational circuits using a buffer, AND gate and OR gate. The structure of PLA a is as shown in Figure 39

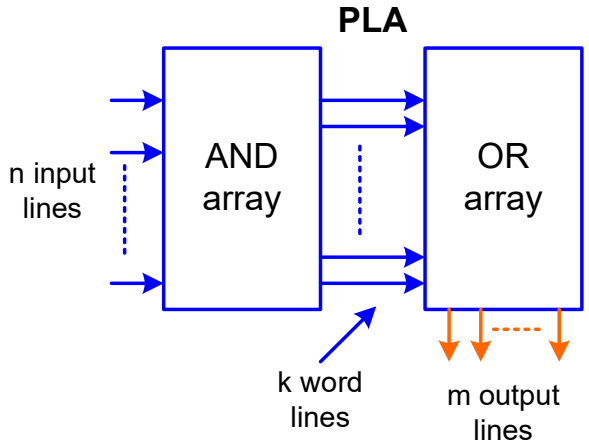
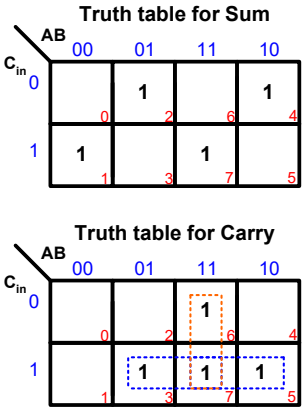


Figure 39

Q1. Design a full adder using PLA.

A	B	C_{in}	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



$$Sum = \sum_m (1,2,4,7)$$

$$Carry = \sum_m (3,5,6,7)$$

Figure 40

$$\begin{aligned} Sum &= \overline{A}\,\overline{B}C + \overline{A}\,B\overline{C} + A\,\overline{B}\,\overline{C} + ABC \\ &= \overline{A}(\overline{B}C + B\overline{C}) + A(\overline{B}\,\overline{C} + BC) \\ &= \overline{A}(B \oplus C) + A(\overline{B \oplus C}) \\ &= A \oplus B \oplus C \end{aligned}$$

$$Carry = AB + BC + AC$$

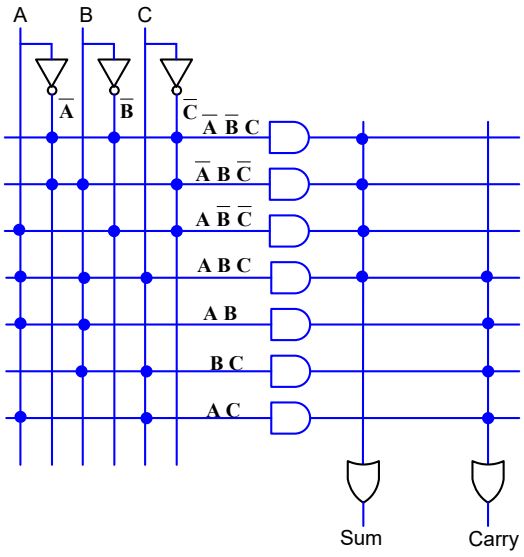
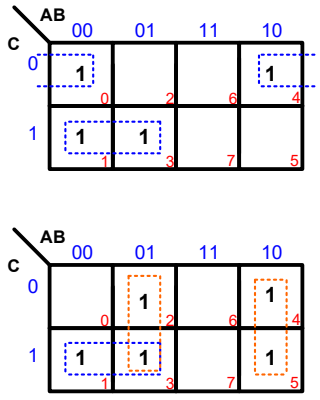


Figure 41

Q2. Implement the following using PLA.

$$f_1(A,B,C) = \sum(0,1,3,4)$$
$$f_2(A,B,C) = \sum(1,2,3,4,5)$$



$$f_1 = \overline{B}\,\overline{C} + \overline{A}\,C$$
$$f_2 = \overline{A}C + \overline{A}\,B + A\,\overline{B}$$

Figure 42

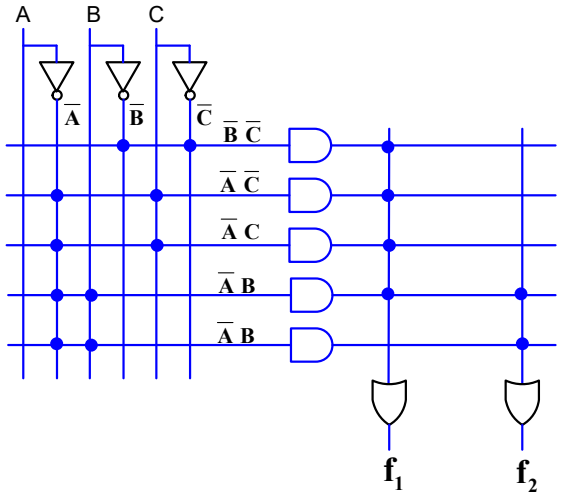


Figure 43

Q1. Design a PLA to recognize each of the 10 decimal digits represxented in binary form and to correctly drive a 7-segment display.

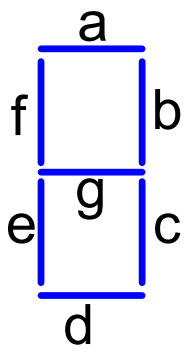


Figure 44

Display	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	0	0	1	1

The details of the seven segment is as shown in Figure 45

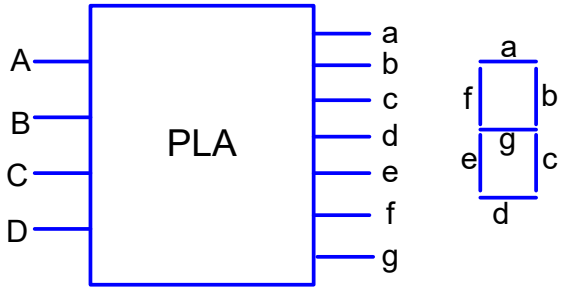


Figure 45

Consider a seven segment display which is as shown in Figure which consists of 7 segments as a,b,c,d,e,f,g. To display digit 0 the segments a,b,c,d,e,f should be ON and its corresponding ABCD=LLLL, similarly to display digit 3 the segments a,b,c,d,g should be ON and its corresponding ABCD=LLHH,. The details of the AND OR logic circuit is as shown in Figure 46

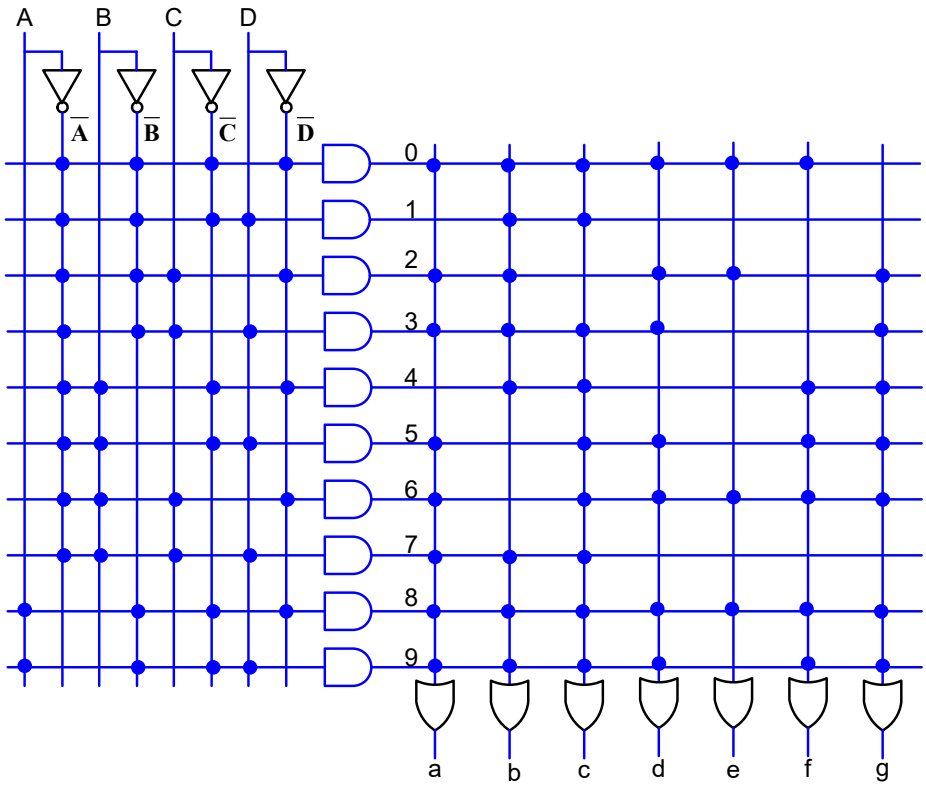

A
B
C
D
$\overline{A}$
$\overline{B}$
$\overline{C}$
$\overline{D}$
0
1
2
3
4
5
6
7
8
9
a
b
c
d
e
f
g

Figure 46