

Q.1a) *Convert (1234.56)10 to octal hexadecimal* 4

Ans. In to octal form-(2322.43)
 In hexadecimal form-(4D2.8F)

b) *Perform the following operation without converting to any other base* 8

Ans. i)(ABC)H-(FEDC)H=(FFFF0124)H
 ii)(234.12)5+(432.34)5=(1222.01)5
 iii)(76)8*(67)8=(6522)8
 iv)(10101011)2/(101)2=(100010)2

c) *Represent (29)10 into Excess 3 code and grey code* 4

Ans. Bcd of 29 is 0011 1001

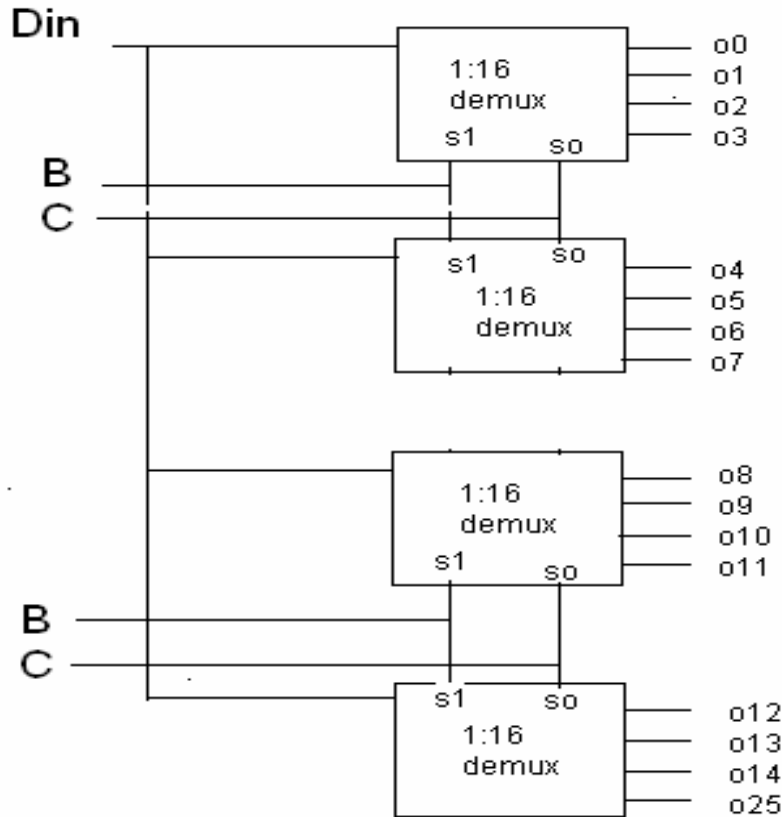
```

0011 1001
+0011 0011
-----
110  1100
    
```

d) *Design 1:16 demux using 1:4 demux*

4

Ans.



Q.2a) *i) subtract using 1's and 2's compliment method*

2

Ans. USING 1'S COMPLIMENT

step1) convert in to binary form

$$73=1001001$$

$$49=0110001$$

step2) 1's compliment of the 49

$$1001110$$

step 3) add 1's compliment into the 73

$$\begin{array}{r}
 1001001 \\
 + \quad 1001110 \\
 \hline
 \underline{10010111}
 \end{array}$$

step 4) add carry into result

$$\begin{array}{r}
 0010111 \\
 + \quad 1 \\
 \hline
 \end{array}$$

$$\begin{array}{r}
 11000 \\
 (73)_{10} - (49)_{10} = (24)_{10} = (11000)_2
 \end{array}$$

USING 2'S COMPLIMENT

step 1) find 2's compliment of the 49

$$\begin{array}{r}
 1001110 \\
 + \quad 1 \\
 \hline
 1000001
 \end{array}$$

step 2) add 2's compliment into 73

$$\begin{array}{r}
 1001001 \\
 + \quad 1001111 \\
 \hline
 \underline{1001100}
 \end{array}$$

discard carry.

$$(73)_{10} - (49)_{10} = (24)_{10} = (11000)_2$$

perform BCD additon for following numbers 56 and 65

BCD of 56 is 0101 0110

BCD of 65 is 0110 0101

BCD addition is

0101 0110

ii)

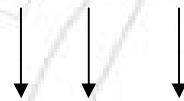
Ans. + 0110 0101

1011 1011

Since number is greater than 6 hence add 6

• 1011
 + 0110 0110

1 0010 0001



1 2 1

(56)+(65)=(121)

iii) perform $(11010)_2 / (101)_2 = 101$

quotient=1

iv) write hamming code for number 0111

b) ***Simplify using Boolean theorem and draw logic diagram for the following***

Ans. i) $A'BC + AB'C + ABC' + ABC$

$$= C(AB' + A'B) + AB(C + C')$$

$$= C(A \text{ XOR } B) + AB$$

II) $A[B + C(AC + AB)']$

$$= AB + AC(AC)'.(AB)'$$

$$= AB$$

III) $(AB)'(B + C) + AB(B + C)'$

$$= (A' + B')(B + C) + AB(B' + C')$$

$$=A'B+A'C+BB'+B'C+ABB'+ABC'$$

$$=A'B+A'C+B'C+ABC'$$

3A) *Minimize the following logic function and realize using NAND gate.*

$$f(A,B,C,D)=EM(1,3,5,8,9,11,15)+(2,13)$$

Ans.

AB \ CD

	1	1	X
	1		
	X	1	
1	1	1	

$$Y=A'C'D+ABC+AD+AB'C'+AB'D$$

b) *Simplify using Quinn McCluskey method realize equation using universal gate.* 10

$$F(A,B,C,D)=PI M(0,2,3,6,7,8,9,12)$$

Ans. .first convert into SOP form

$$f(A,B,C,D)=EM(1,4,5,10,11,13,14,15)$$

M1	0001	M1,m5	0-01
M4		M4m5	010-
	0100	M5-m13	-101
M5		M10-m11	101-
	0101	M11-m15	1-11
M10		M13-m14	11-1
	1010		
M11	1011		
M13	1101		
M14	1110		
M15	1111		

	M1	M4	M5	M10	M11	M13	M14	M15
A'C'D	X		X					
A'BC'		X	X					
BC'D			X			X		
AB'C				X	X			

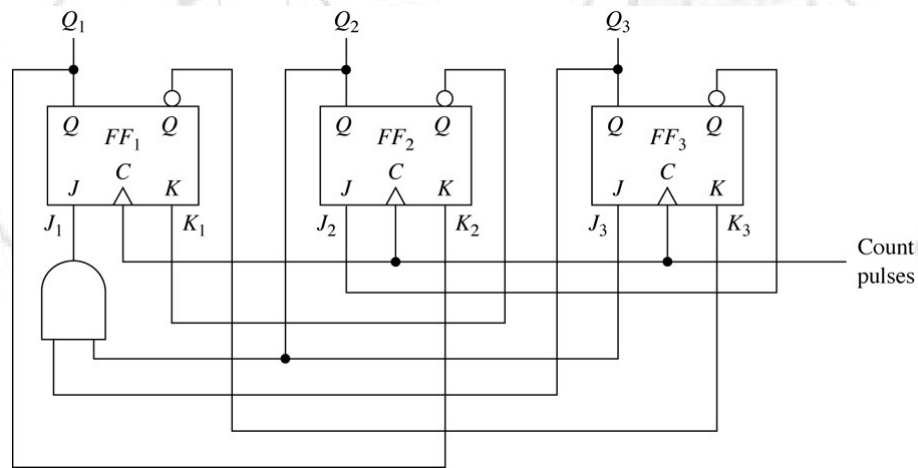
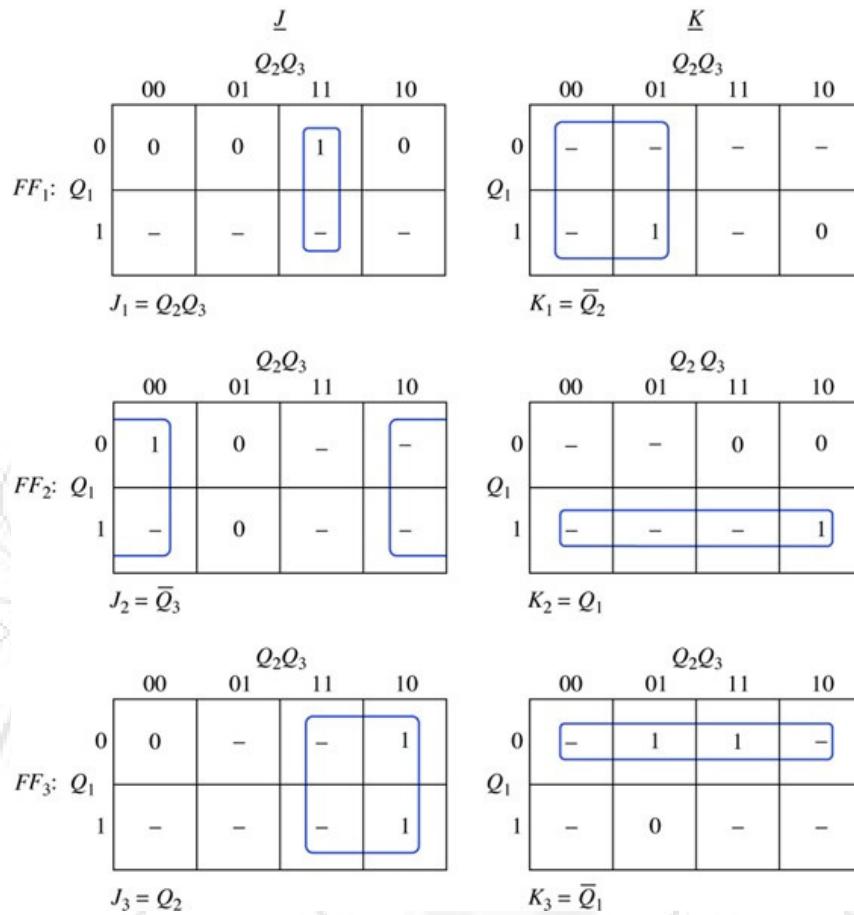
ACD					X			X
ABD						X	X	

$$Y = A'C'D + A'BC' + AB'C + ACD + ABD$$

Q.4) design MOD 6up synchronous up counter.

Ans.

Presentstate			Next state			Flip-flop inputs					
Q	Q2	Q3	Q1+	Q2+	Q3+	J1	K1	J2	K2	J3	K3
1											
0	0	0	0	1	0	0	-	1	-	0	-
0	1	0	0	1	1	0	-	-	0	1	-
0	1	1	1	1	0	1	-	-	0	-	1
1	1	0	1	0	1	-	0	-	1	1	-
1	0	1	0	0	1	-	1	0	-	-	0
0	0	1	0	0	0	0	-	0	-	-	1



Q4b) what is shift register explain 4 bit bidirectional shift register

Ans. The binary information in a register can be moved from stage to stage within the register or into or out of the register upon application of clock pulses. This type of bit movement or shifting is essential for certain arithmetic and logic operations used in microprocessors. This gives rise to group of registers called shift registers.

There are five types. They are,

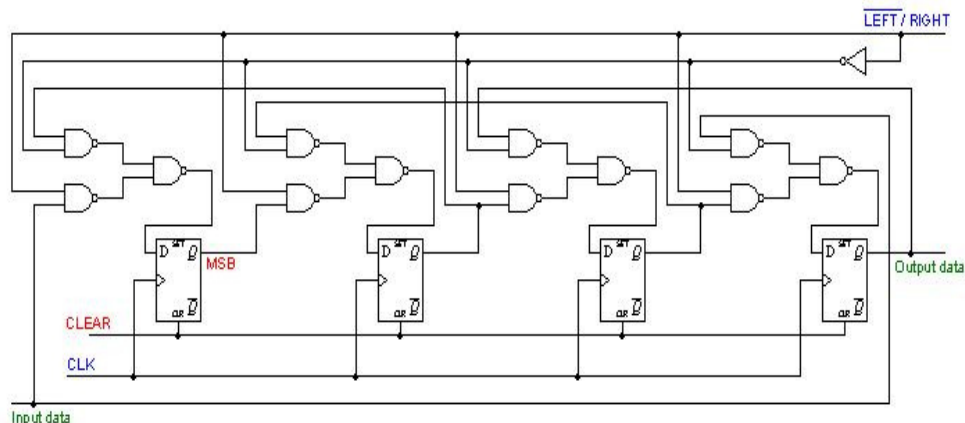
- Serial In Serial Out Shift Register
- Serial In Parallel Out Shift Register
- Parallel In Serial Out Shift Register
- Parallel In Parallel Out Shift Register
- Bidirectional Shift Register

The registers discussed so far involved only right shift operations. Each right shift operation has the effect of successively dividing the binary number by two. If the operation is reversed (left shift), this has the effect of multiplying the number by two. With suitable gating arrangement a serial shift register can perform both operations.

A bidirectional or reversible shift register is one in which the data can be shift either left or right. A four-bit bidirectional shift register using D flip-flops is shown below.

Here a set of NAND gates are configured as OR gates to select data inputs from the right or left adjacent bistable, as selected by the LEFT/RIGHT control line.

The animation below performs right shift four times, then left shift four times. Notice the order of the four output bits are not the same as the order of the original four input bits. They are actually reversed!



Q.5a)impliment the following boolean function using 4:1 mux

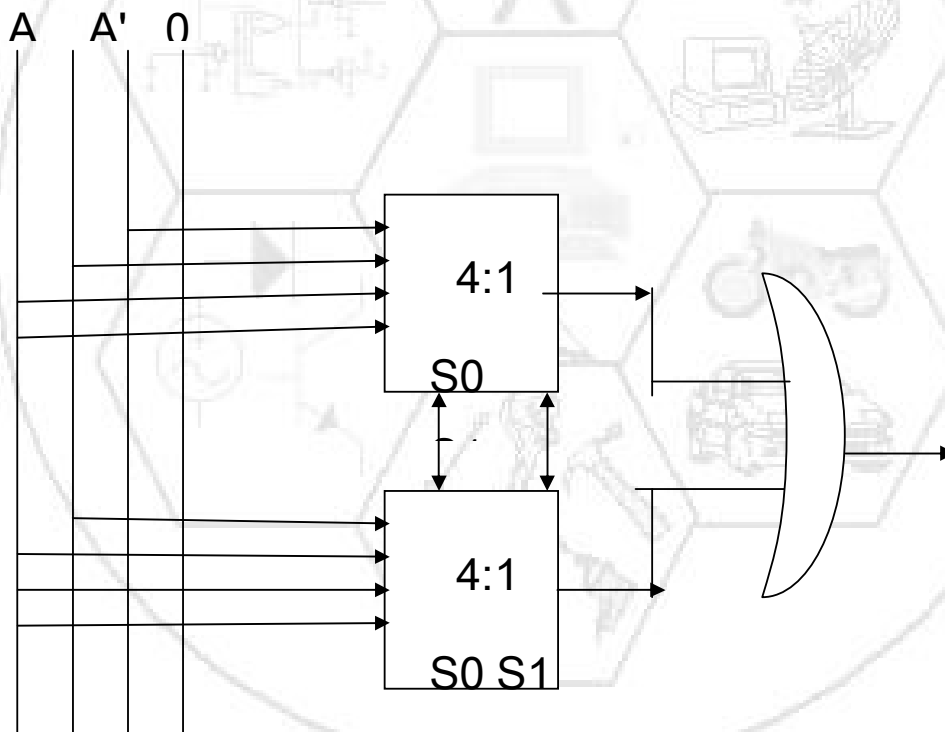
$F(A,B,C,D)=\sum m(0,2,3,6,7,8,9,12,13)$

converting in to min term

$F(A,B,C,D)=\sum m(1,4,5,10,11,14,15)$

Ans.

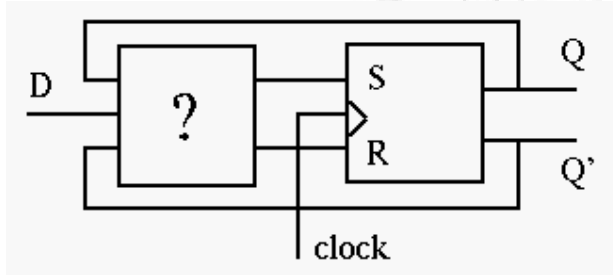
A'	<u>0</u>	1	<u>2</u>	<u>3</u>	4	5	<u>6</u>	<u>7</u>
A	<u>8</u>	<u>9</u>	10	11	<u>12</u>	<u>13</u>	14	15
O/P	0	A'	A	A	A'	A	A	A



B) convert SR flip flop to T and D flip flop

Ans.

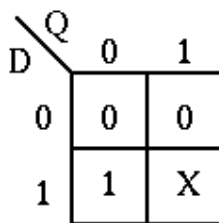
Convert a RS-FF to a D-FF:



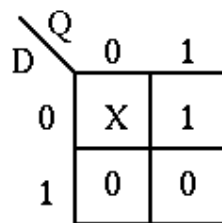
We need to design the circuit to generate the triggering signals S and R as functions of D and Q . Consider the excitation table:

Q_t	Q_{t+1}	D	S	R
0	0	0	0	x
0	1	1	1	0
1	0	0	0	1
1	1	1	x	0

The desired signal S and R can be obtained as functions of T and current FF state from the Karnaugh maps:

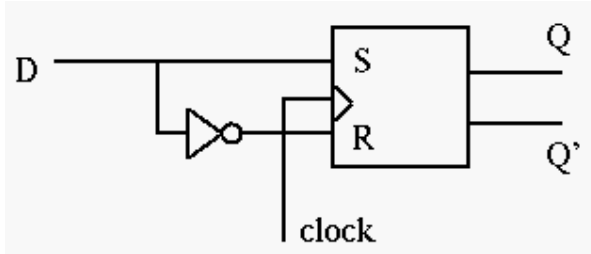


$S=D$

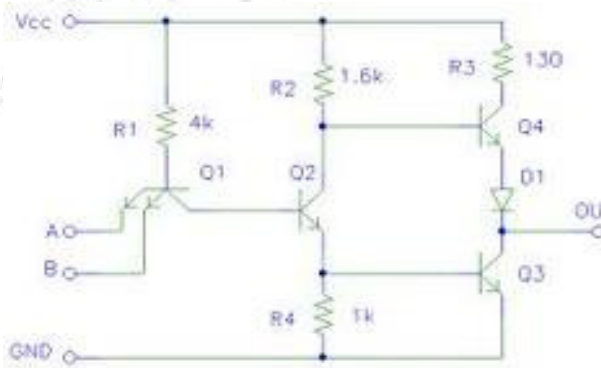


$R=D'$

$$S = D, \quad R = D'$$



Q.6A) draw 2 input TTL NAND gate and explain its working

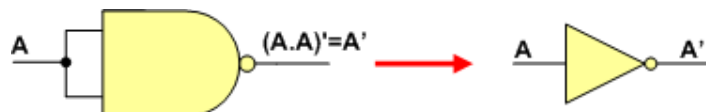


Q.6b) PROVE THAT NAND AND NOR GATES ARE UNIVERSAL GATES

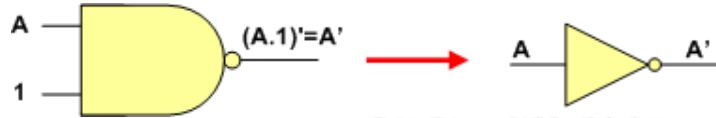
Ans.

NAND Gate is a Universal Gate:

To prove that any Boolean function can be implemented using only NAND gates, we will show that the AND, OR, and NOT operations can be performed using only these gates. Implementing an Inverter Using only NAND Gate The figure shows two ways in which a NAND gate can be used as an inverter (NOT gate). All NAND input pins connect to the input signal A gives an output A'.



- One NAND input pin is connected to the input signal A while all other input pins are connected to logic 1. The output will be A'.



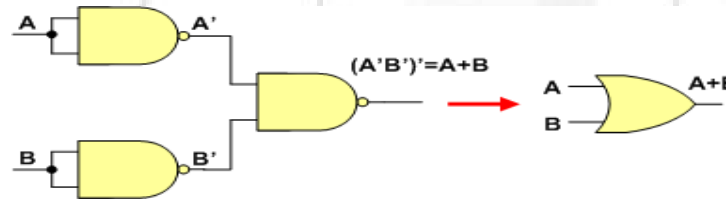
Implementing AND Using only NAND Gates.

An AND gate can be replaced by NAND gates as shown in the figure (The AND is replaced by a NAND gate with its output complemented by a NAND gate inverter).



Implementing OR Using only NAND Gates

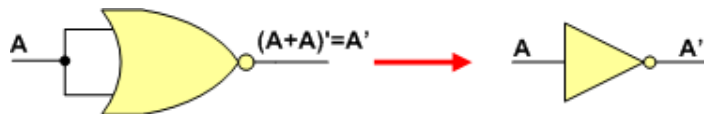
An OR gate can be replaced by NAND gates as shown in the figure (The OR gate is replaced by a NAND gate with all its inputs complemented by NAND gate inverters).



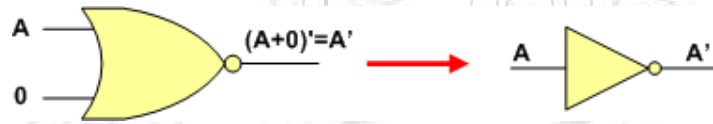
Thus, the NAND gate is a universal gate since it can implement the AND, OR and NOT functions.

Implementing an Inverter Using only NOR Gate

The figure shows two ways in which a NOR gate can be used as an inverter (NOT gate).



- All NOR input pins connect to the input signal A gives an output A' .
- One NOR input pin is connected to the input signal A while all other input pins are
- connected to logic 0. The output will be A' .



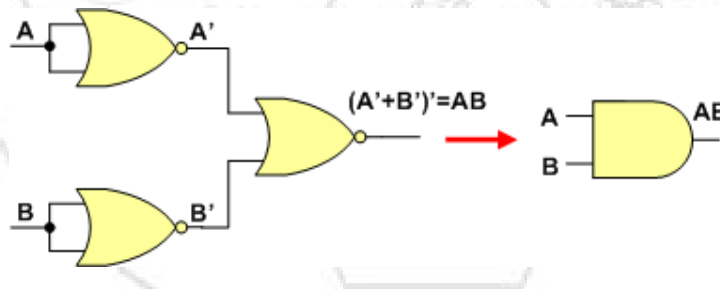
Implementing OR Using only NOR Gates

An OR gate can be replaced by NOR gates as shown in the figure (The OR is replaced by a NOR gate with its output complemented by a NOR gate inverter)



Implementing AND Using only NOR Gates

An AND gate can be replaced by NOR gates as shown in the figure (The AND gate is replaced by a NOR gate with all its inputs complemented by NOR gate inverters)

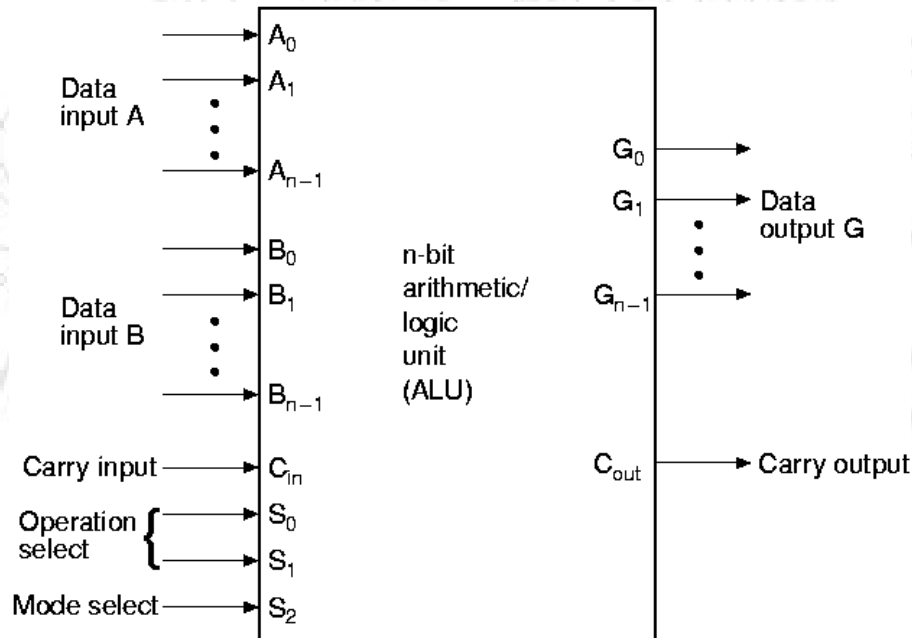


Thus, the NOR gate is a universal gate since it can implement the AND, OR and NOT functions.

7 write note on

a)ALU

Ans.



-ALU is very widely used and popular combinational circuit.

-it is capable of performing the arithmetic and logical operation

-ALU is heary of microprocessore

-Essential element of the Central Processing Unit Arithmetic and logic functions on binary words

- n-bit data inputs A and B

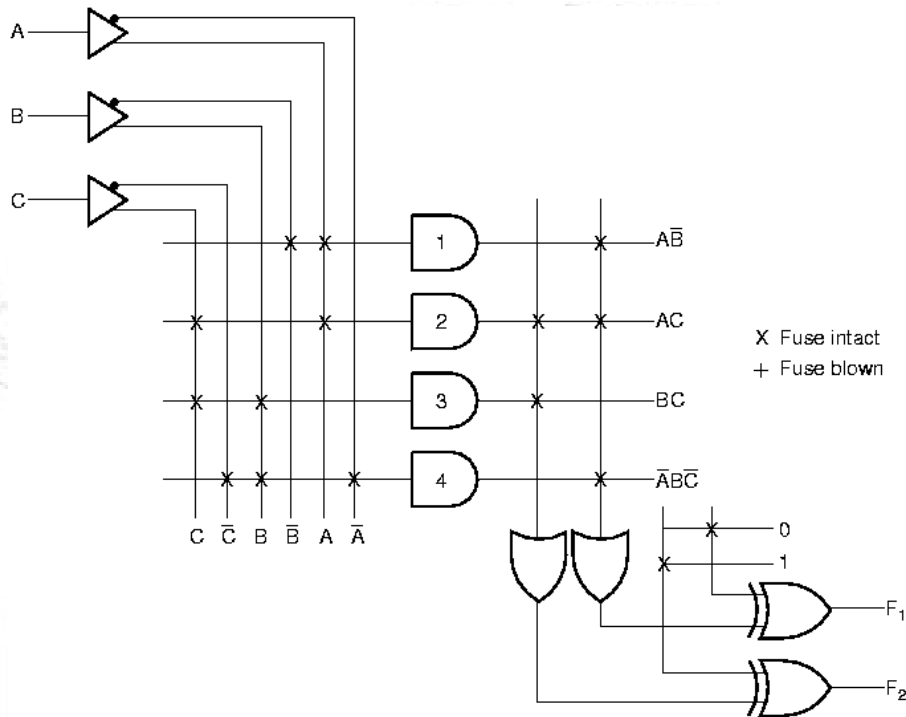
- n-bit data output $G = f(A, B)$

- Selection inputs S0, S1 select a function f

- Selection input S2 select an operating mode (arithmetic or logic)

b) PLA and PAL

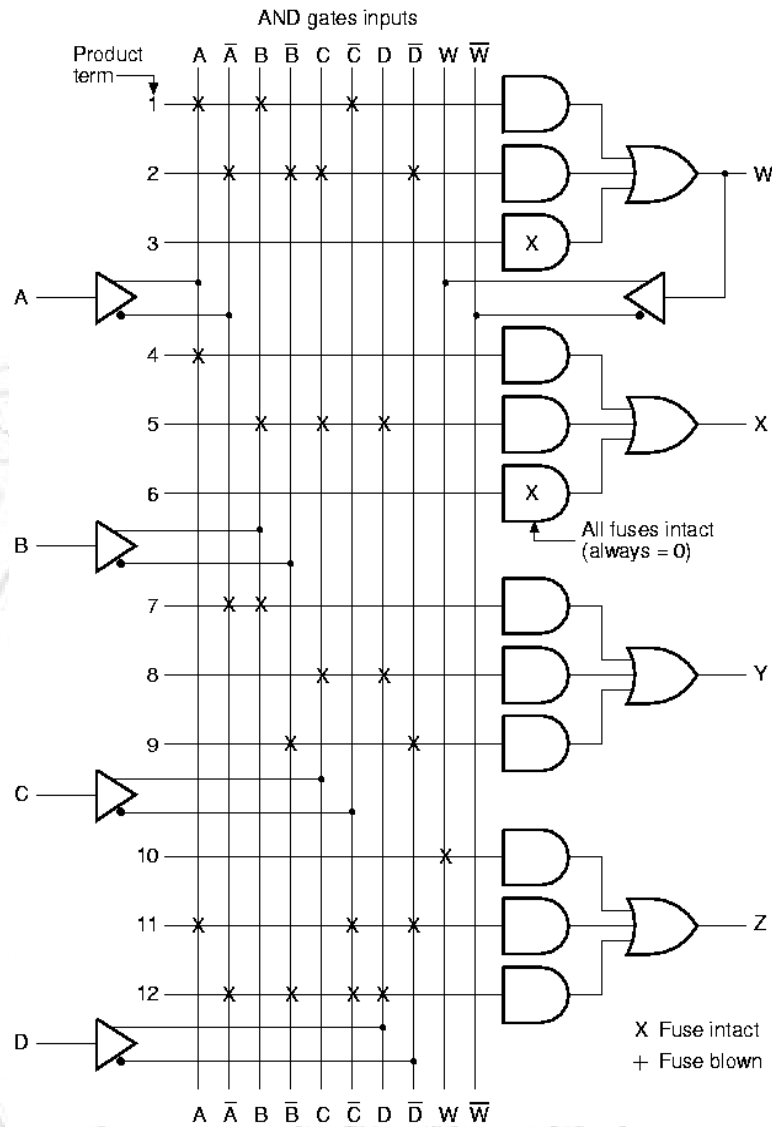
Ans.



Behave like a ROM but has different structure

- Uses ANDs array instead of decoder to produce product terms of inputs
- Has programmable connections before ANDs, between ANDs and ORs, after ORs. That is $2nk + km + m$ fuses
- More flexible than ROM but more difficult to program
- Logic expressions for content information to be stored in PLA must be obtained first, then minimized, and finally programmed into the PLA using a PLA program table
- PLA program table specifies product terms and sum terms of information that will be stored in

PAL



Similar to PLA

- Only the connection inputs to ANDs are programmable
- Easier to program than but not as flexible as PLA
- There are feedback connections
- Logic expressions for content information to be stored in PAL must be obtained first, then minimized, and finally programmed into the PAL using a PAL program table
- PAL program table specifies only product terms of information that will be stored in PAL

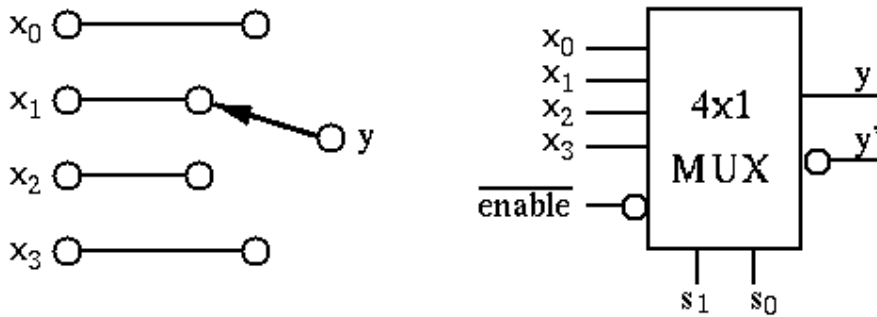
C) multiplexer and demultiplexer

Ans.

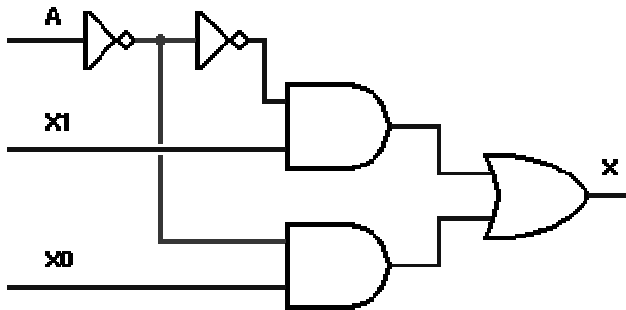
o Multiplexers (MUX)

The *multiplexer* this is a digital circuit with multiple signal inputs, one of which is selected by separate address inputs to be sent to the single output. It's not easy to describe without the logic diagram, but is easy to understand when the diagram is available.

An MUX has N inputs and one output. Under the control of $n = \log_2 N$ selection signals, one of the inputs is passed on to the output.



A two-input multiplexer is shown below.

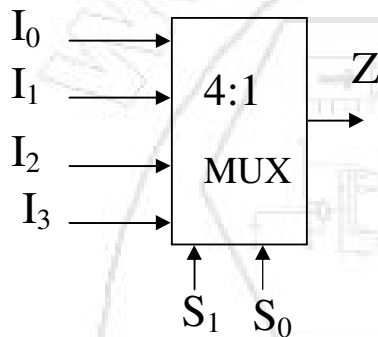
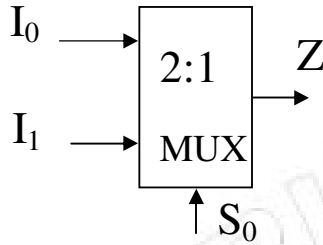


The multiplexer circuit is typically used to combine two or more digital signals onto a single line, by placing them there at different times. Technically, this is known as *time-division multiplexing*.

Input A is the addressing input, which controls which of the two data inputs, X0 or X1, will be transmitted to the output. If the A input switches back and forth at a frequency more than double the frequency of either digital signal, both signals will be accurately reproduced, and can be separated again by a *demultiplexer* circuit synchronized to the multiplexer.

This is not as difficult as it may seem at first glance; the telephone network combines multiple audio signals onto a single pair of wires using exactly this technique, and is readily

able to separate many telephone conversations so that everyone's voice goes only to the intended recipient. With the growth of the Internet and the World Wide Web, most people have heard about T1 telephone lines. A T1 line can transmit up to 24 individual telephone conversations by multiplexing them in this manner.



The same can be said about a 4:1 MUX:

Input I_i is selected ($Z=I_i$)

if S_1S_0 combination represents the number i in binary.

A 2:1 MUX selects input I_i if $S_0 = I_i$

[If $S_0 = 0, Z = I_0$

$S_0 = 1, Z = I_1$]

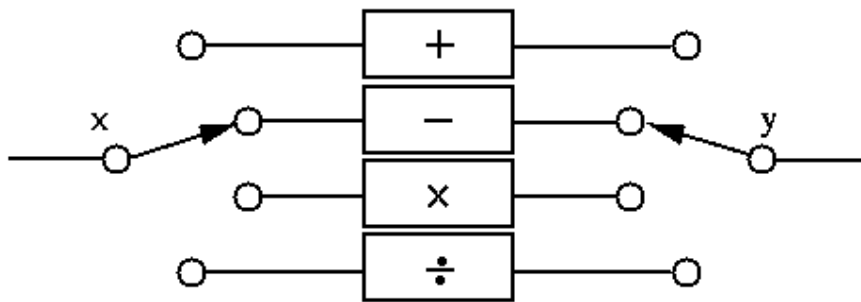
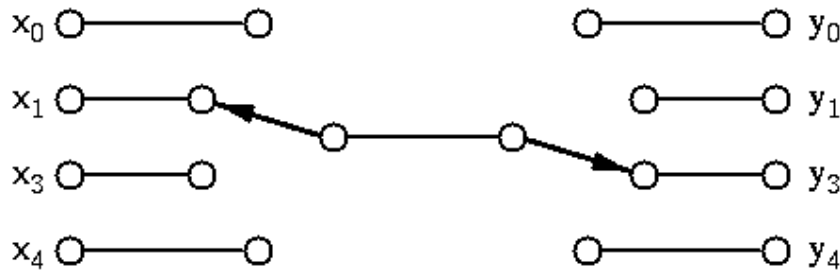
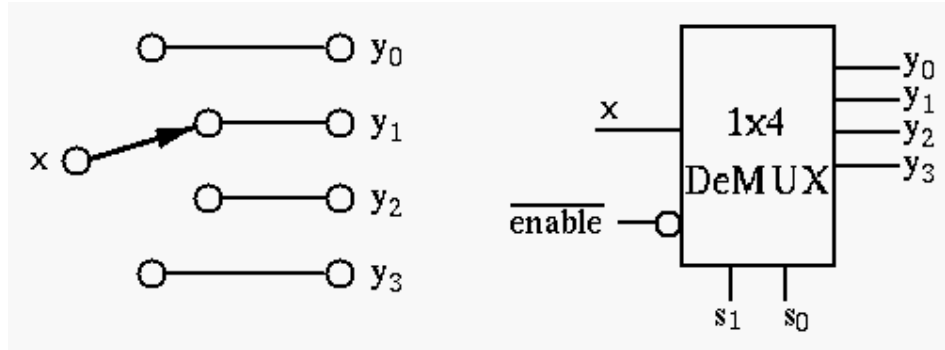
S_1	S_0	Z
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

$$Z = \bar{S}_1\bar{S}_0I_0 + \bar{S}_1S_0I_1 + S_1\bar{S}_0I_2 + S_1S_0I_3$$

○ *DeMultiplexer (DeMUX)*

$$n = \log_2 N$$

A DeMUX has a single input but N outputs. Under the control of *selection* signals, the input is passed to one of the outputs.



1x2 DeMUX:

s	x	y1	y0
0	0	0	0
0	1	0	1
1	0	0	0
1	1	1	0

$$y_0 = s'x, \quad y_1 = sx$$

1x4 DeMUX:

s_1	s_0	x	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

d) *race around condition in JK flip flop*

Ans.

In JK flip-flop output is fed back to the input. Therefore change in the output results change in the input. Due to this in the positive half of the clock pulse if both J and K are high then output toggles continuously. This condition is called 'race around condition'.

References –

-moris mano

-technical publicatins