

Sequential Ckts - I

Comprehensive Course on Digital Logic Design 2023/2024

NUMBER SYSTEMS

Any number is associated with **Base** (or) **Radix**

$$(734)_{10}$$

$$(734)_{10} =$$

$$(472.15) =$$

A number system with base ‘ b ’ , will have b different digits and they are from 0 to $b - 1$.

$$(421)_4$$

$$(243)_5$$

$$(851)_9$$

Base (b) is always a positive integer .

In general $b \geq 2$

Base	Different digits
2 (Binary)	
8(Octal)	
10 (Decimal)	
16 (Hexadecimal)	

Conversion of Number System

1. Decimal to Any Base

$$[N]_{10} \rightarrow [?]_b$$

2. Any base to Decimal

$$[N]_b \rightarrow [?]_{10}$$

3. one base to another base

$$[N_1]_{b_1} \rightarrow [?]_{b_2}$$

4. Required base = (Given Base)^{integer}

1. Decimal to Other Base

- Integer part, repeated division by the required base .
- Fractional part , repeated multiplication by the required base .

$$Q) (53.75)_{10} = \underline{\hspace{2cm}}_2$$

$$Q) (0.15)_{10} = \underline{\hspace{2cm}}_2$$

Note :

It is possible to obtain the equivalent of integer part but may not possible for fractional part .

$$Q) (53.75)_{10} = \underline{\hspace{2cm}}_4$$

$$Q) (39.5)_{10} = \underline{\hspace{2cm}}_8$$

$$Q) (39.5)_{10} = \underline{\hspace{2cm}}_{16}$$

2. Any base to Decimal

$$(x_2x_1x_0 \cdot x_{-1}x_{-2}x_{-3})_b = (\quad ? \quad)_{10}$$

$$Q) (311.30)_4 = (\underline{\hspace{2cm}} \underline{\hspace{2cm}} \underline{\hspace{2cm}} \underline{\hspace{2cm}})_ {10}$$

Q) Find the minimum decimal equivalent of $(3AB26)_x$

$$Q) (137.4)_8 = \underline{\hspace{2cm}}_{10}$$

$$\text{Q) } (DAD)_{16} = \underline{\hspace{2cm}}_{10}$$

$$\text{Q) } (ECE)_{16} = \quad (- - - -)_{10}$$

$$\text{Q) } (EEE)_{16} = \underline{\hspace{2cm}} \quad (- - - -)_{10}$$

$$Q) \text{ Find } b \text{ if } \sqrt{(41)_b} = (5)_{10}$$

3. One base to another base

$$[N]_{b_1} \rightarrow [?]_{b_2}$$

1. Convert the given number to the decimal system
2. After that convert to required base

$$Q) \ (3)_4 = (?)_8$$

$$Q) \ (7)_8 = (?)_9$$

Q) Find the value of x if $(193)_x = (623)_8$

Q) Find b_1 and b_2 if $(235)_{b_1} = (565)_{10} = (1065)_{b_2}$

Q) The solution to the quadratic equation $x^2 - 11x + 22 = 0$ is $x = 3$ and $x = 6$, what is the base of the system

4. Required base = (*Given Base*)^{integer}

$$Q) (1011010110.010110010)_2 = (-\underline{\hspace{2pt}}\underline{\hspace{2pt}}\underline{\hspace{2pt}}\underline{\hspace{2pt}})_8$$

$$Q) (1011010110.010110010)_2 = (-\underline{\hspace{2mm}} \underline{\hspace{2mm}} \underline{\hspace{2mm}} \underline{\hspace{2mm}})_4$$

$$Q) (1011010110.010110010)_2 = (-\underline{\hspace{2pt}}\underline{\hspace{2pt}}\underline{\hspace{2pt}}\underline{\hspace{2pt}}\underline{\hspace{2pt}})_ {16}$$

$$Q) (2210121012.2011022)_3 = (-\underline{-} \underline{-} \underline{-} \underline{-})_9$$

$$Q) \ (3210332101.2210)_4 = (- - - -)_{16}$$

Q) Find the number of solutions of ‘Y’ exists for $(123)_5 = (X8)_Y$

Q) Find the number of solutions of ‘ x ‘ exists for $(123)_x = (12X)_3$

Q) Find the base of the following system such that given operation is valid

$$24+14 = 41$$

Q) Find the base of the following system such that given operation is valid

$$\frac{66}{6} = 11$$

Q) Find the base of the following system such that given operation is valid

$$\sqrt{121} = 11$$

Q) Find the base of the following system such that given operation is valid

$$\sqrt{41} = 5$$

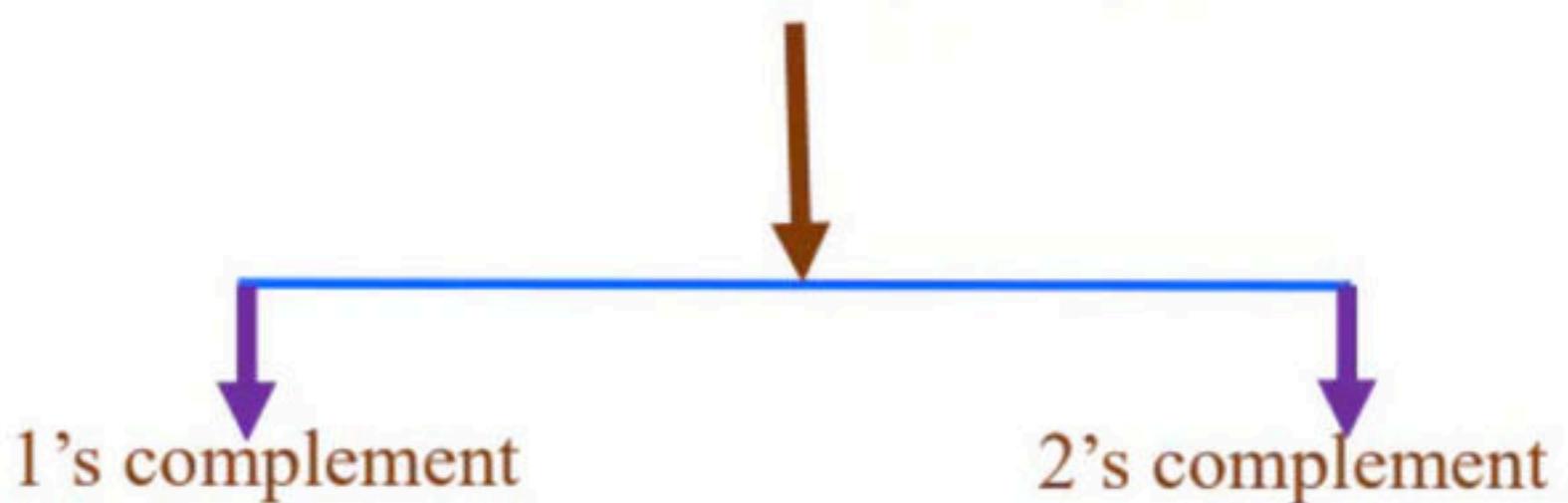
Complement Analysis

$[N]_r$

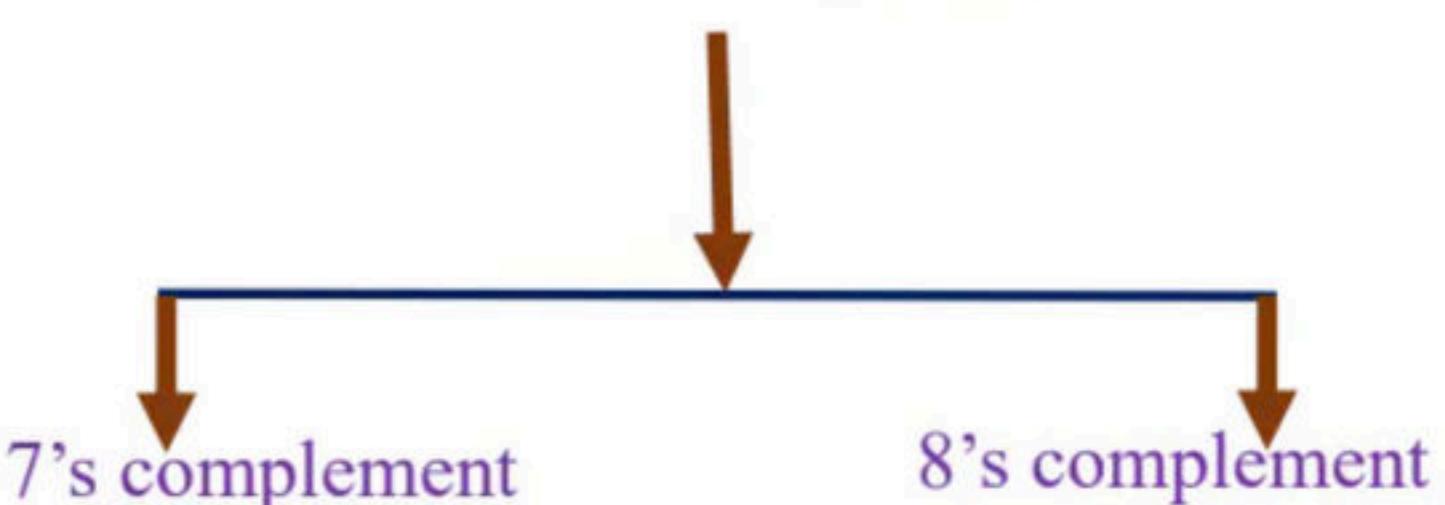
1. r's complement

2. (r-1)'s complement

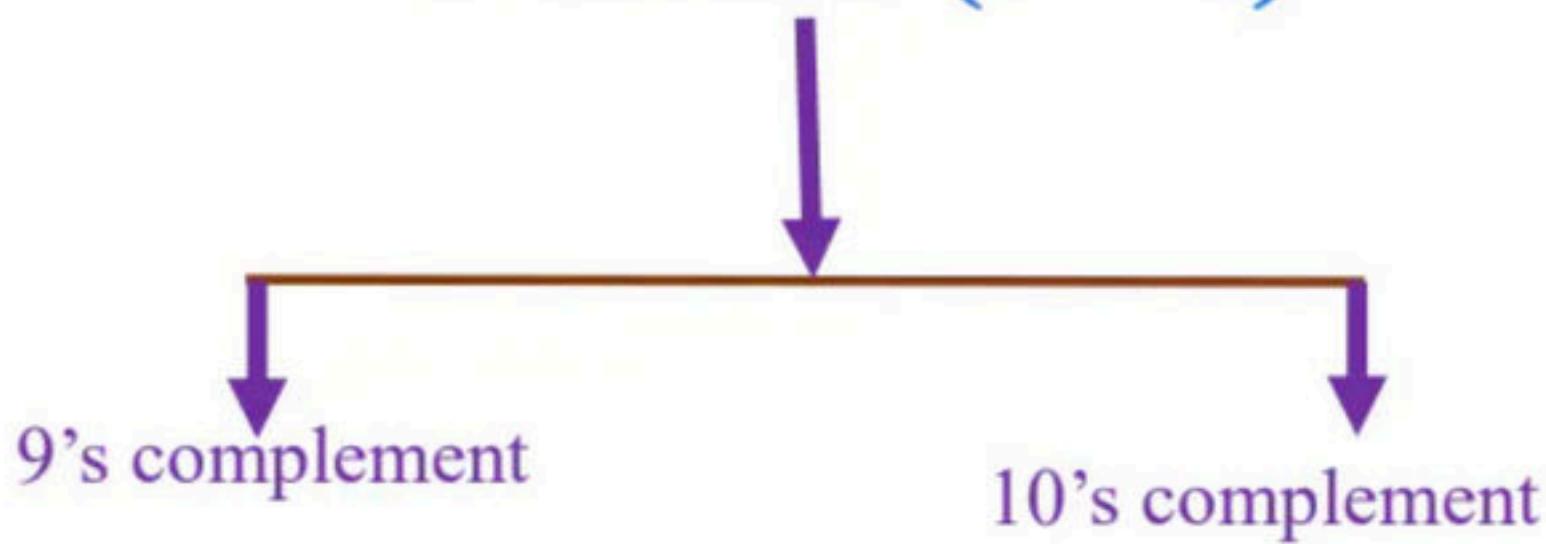
Binary (r=2)



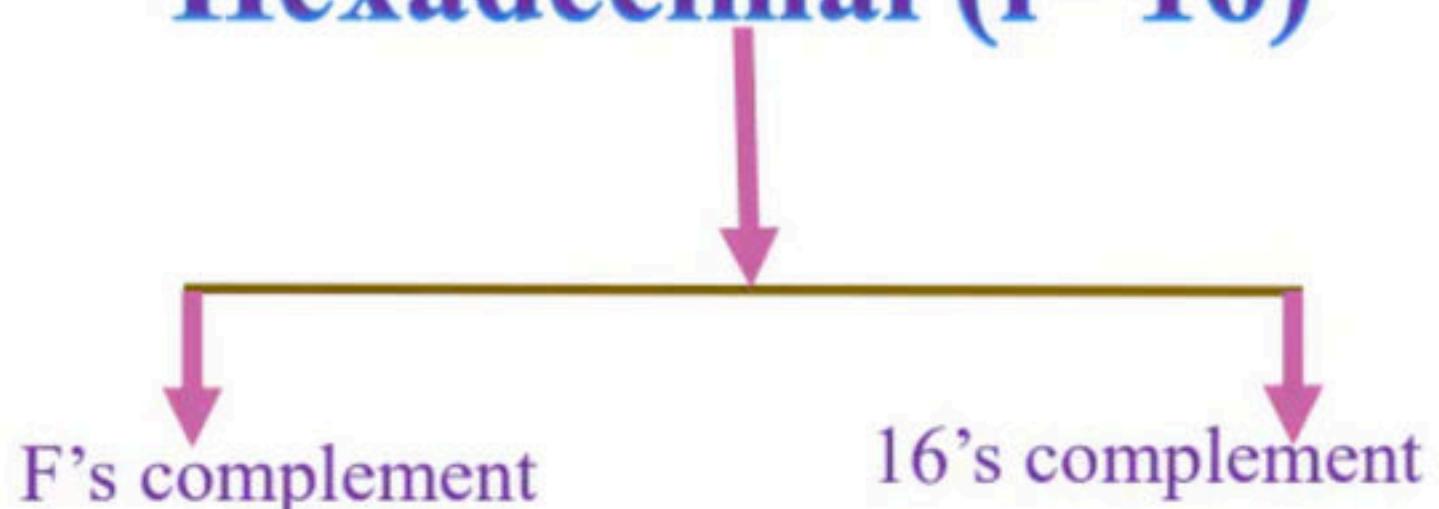
Octal (r=8)



Decimal ($r=10$)



Hexadecimal ($r=16$)



r' s complement

r' s complement of the number (N) = $r^n - N$

r -----> Radix

n -----> number of integer digits

N -----> given number

(r-1) ' s complement

(r-1) ' s complement of the number (N) = $r^n - r^{-m} - N$

r -----> Radix

n -----> number of integer digits

m -----> number of decimal digits

N -----> given number

$(r-1)$'s complement of the number $(N) = r^n - r^{-m} - N$

r 's complement of the number $(N) = (r-1)$'s complement + r^{-m}

If $m = 0$

r 's complement of the number $(N) = (r-1)$'s complement + 1

Q) Find the 10's complement of $(327.452)_{10}$

Q) Find the 9's complement of $(327.452)_{10}$

Q) Find the 10's complement of $(784732179)_{10}$

Q) Find the 2's complement of $(101100)_2$

Q) Find the 2's complement and 1's complement of $(0.0110)_2$

Q) Find the 9's and 10's complement of $(52520)_{10}$

Q) Find the 9's and 10's complement of $(0.3267)_{10}$

Q) Find 1's and 2's complement of $(10100100111)_2$

Q) Find 8's and 9's complement of $(278421)_9$

Q) Find F's and 16's complement of $(792410)_{16}$

Q) Find 1's and 2's complement of $(11000100)_2$

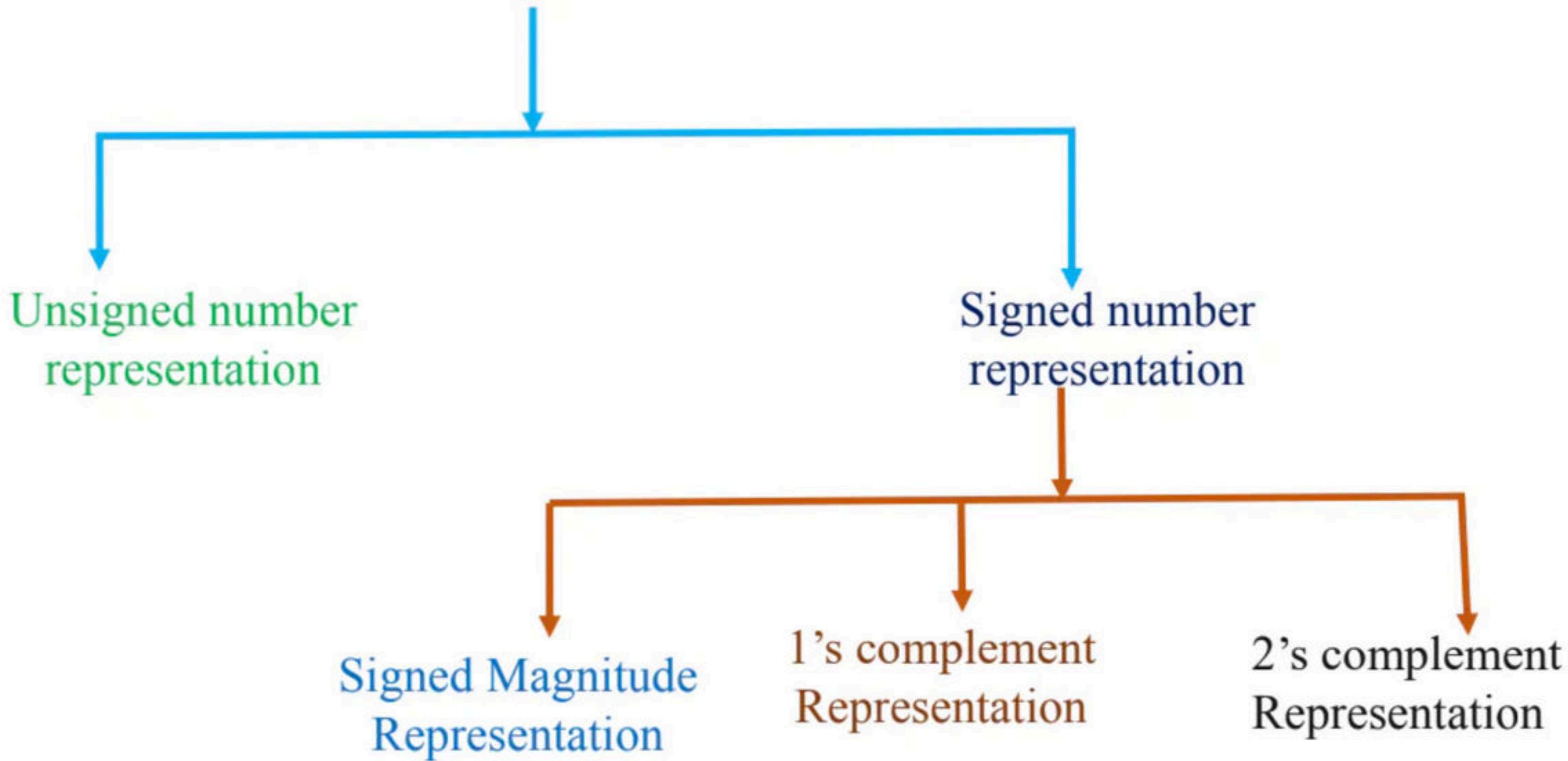
Q) Find 1's and 2's complement of $(11010.11)_2$

Q) Find 8's complement of $(2670)_8$

Q) Find 10's complement of $(7492)_{10}$

Q) Find 16's complement of $(9623)_{16}$

Data Representation



Unsigned Number Representation

- Strictly applicable for positive numbers
- There is no sign bit concept

+ 5 ----->

- 5 ----->

Decimal number	Unsigned number representation (4-bits)
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	

Range with 4 bits =

Range with 5 bits =

Range with n- bits =

Signed Number Representation

- 1.Signed magnitude representation
- 2.1's complement representation
- 3.2's complement representation

Signed Magnitude representation

- Valid for both positive and negative numbers .
- Sign bit concept is used .



Sign bit = 0 , for \oplus Ve number
= 1, for \ominus ve number

$+5 =$

--	--	--	--

$- 5 =$

--	--	--	--

$+5 =$

--	--	--	--	--	--	--	--

$- 5 =$

--	--	--	--	--	--	--	--

Decimal number	Signed Magnitude Representation (4-bits)
+0	
+1	
+2	
+3	
+4	
+5	
+6	
+7	
-0	
-1	
-2	
-3	
-4	
-5	
-6	
-7	

Range with 4 bits =

Range with 5- bits =

Range with n- bits =

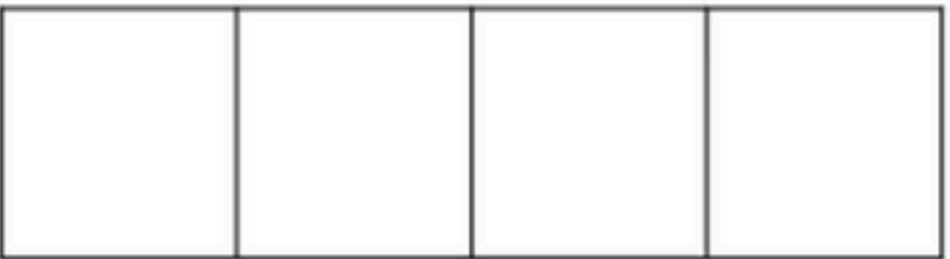
1's Complement Representation

- In this **⊕Ve numbers** are represented as **normal binary number with MSB '0'**

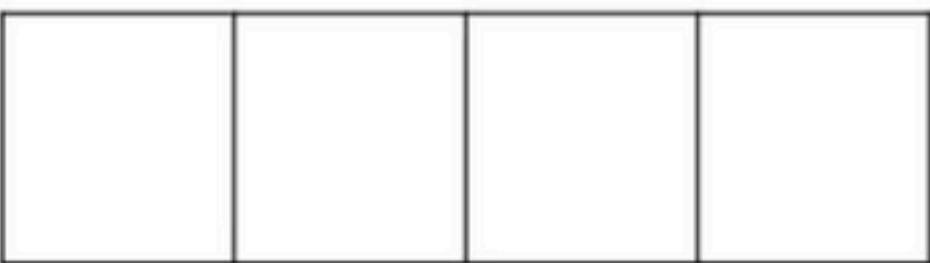
Representation of ⊖ ve number

1. Write the binary equivalent of magnitude
2. Take its 1's complement

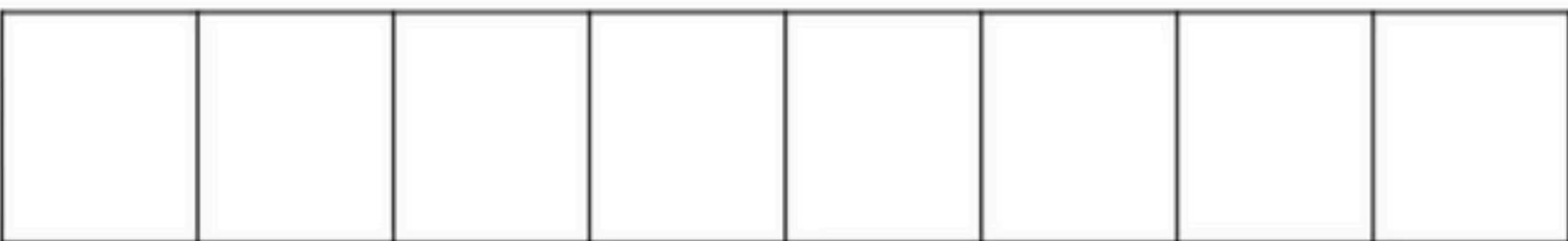
$+6 =$



$- 6 =$



$+6 =$



$- 6 =$



Decimal number	1's complement Representation (4-bits)
+0	
+1	
+2	
+3	
+4	
+5	
+6	
+7	
-0	
-1	
-2	
-3	
-4	
-5	
-6	
-7	

Range with 4 bits =

Range with 5- bits =

Range with n- bits =

2's complement Representation

- In this +Ve numbers are represented as *normal binary number with MSB '0'*

Representation of +ve number

1. Write the binary equivalent of magnitude
2. Take its 2's complement

$+6 =$

--	--	--	--

$- 6 =$

--	--	--	--

$+6 =$

--	--	--	--	--	--	--	--

$- 6 =$

--	--	--	--	--	--	--	--

Decimal number	2's complement Representation
+0	
+1	
+2	
+3	
+4	
+5	
+6	
+7	
-0	
-1	
-2	
-3	
-4	
-5	
-6	
-7	
-8	

Range with 4 bits =

Range with 5- bits =

Range with n- bits =

Q) Find the Decimal equivalent of the **unsigned number representation** given below

a) 01101

b) 11101

Q) Find the Decimal equivalent of the **Signed magnitude representation** given below

a) 01101

b) 11101

Q) Find the Decimal equivalent of the **1's complement representation** given below

a) 01101

b) 11101

Q) Find the Decimal equivalent of the **2's complement representation** given below

a) 01101

b) 11101

Q) 1's complement representation of the numbers given below , find the decimal equivalents

a) 01001

b) 001001

c) 0001001

d) 11001

e) 111001

f) 1111001

Q) 2's complement representation of the numbers given below , find the decimal equivalents

a) 01001

b) 001001

c) 0001001

d) 11001

e) 111001

f) 1111001

Q) Find the **2's complement representation** of the following

$$-2 =$$

$$-4 =$$

$$-8 =$$

$$-16 =$$

$$-2^n =$$

Note :

The minimum number of bits required for - 2^n , using 2's complement representation = -----

Q) A number in 4-bit 2's complement is $x_3x_2x_1x_0$, this number when stored using 8-bits will be

- a) 0000 $x_3x_2x_1x_0$
- b) 1111 $x_3x_2x_1x_0$
- c) $x_3x_3x_3x_3x_3x_2x_1x_0$
- d) $\overline{x_3} \ \overline{x_2} \ \overline{x_1} \ \overline{x_0} \ x_3x_2x_1x_0$

**Q. How many one's are present in the binary representation of
 $(8 \times 4096) + (4 \times 256) + (9 \times 16) + 5$**

- (a) 6
- (b) 5
- (c) 3
- (d) 4

Binary Subtraction using 1's complement

1. Represent the given numbers in the 1's complement form.
2. Add the two numbers.
3. If carry is generated ,then the result is positive and in the true form , add carry to the LSB to get the final answer .
4. If carry is not generated , then the result is negative , and in the 1's complement form . To get final answer take 1's complement of the result.

Q) Perform the following operation for the given numbers using 1's complement form .

a) $8 - 4$

b) $4 - 8$

Binary Subtraction using 2's complement

1. Represent the given numbers in the 2's complement form
2. Add the two numbers
3. If carry is generated ,ignore it .
4. If MSB is 0, then the result is positive and in the true form .
5. If MSB is 1, then the result is negative and is in 2's complement form .
(whether there is a carry (or) no carry does not matter)

Q. Perform the following operation for the given numbers using 2's complement form .

$$46 - 14$$

Q. Perform the following operation for the given numbers using 2's complement form .

$$-75 + 26$$

Q) Simplify the following using 2's complement form

$$9 + 4$$

Q) Simplify the following using 2's complement form

9- 4

Q) Simplify the following using 2's complement form

$$-9 + 4$$

Q) Simplify the following using 2's complement form

$$-9 - 4$$

Q) Simplify the following using 2's complement form

$$9 + 8$$

Q) Simplify the following using 2's complement form

$$-9 - 8$$

Overflow

Over flow occurs in signed arithmetic operations if two same sign numbers are added and result exceeds with given number of bits.

Over flow can be detected by using 2- methods

1. by using carry bits

2. by using sign bit

1. By using carry bits

C_{in} ----- *carry into MSB*

C_{out} ----- *carry out from MSB*

if $C_{in} \oplus C_{out} =$

$C_{in} \oplus C_{out} =$

2. By using sign bit

X -----> Sign bit of 1st number

Y -----> Sign bit of 2nd number

Z-----> Sign bit of Resultant

Over flow =

NOTE :

to avoid the overflow , increase the number of bits .

Q) Let x be the sign bit of N_1 , y be the sign bit of N_2 , and z be the sign bit of $N_1 + N_2$, then the condition for overflow .

- a) $x \neq y \neq z$
- b) $x \neq y = z$
- c) $x = y \neq z$
- d) $x = y = z$

Q. Let R1 and R2 be two 4-bit registers that store number in 2's complement form , for operation $R1 + R2$, which of the following values of R1 and R2 gives overflow.

- a) R1= 1100 , R2 = 1010
- b) R1= 1001 , R2 = 1111
- c) R1= 1011 , R2 = 1110
- d) R1= 0011 , R2 = 0100

Q) Two numbers represented in signed 2's complement form as $P = 11101101$, $Q = 11100110$, if Q is subtracted from P , then the value obtained in signed 2's complement form is

- a) 100000111
- b) 00000111
- c) 11111001
- d) 111111001

BINARY CODES

Numeric Codes

- 1.BCD Code
- 2.Excess-3 Code
- 3.Gray Code
- 4.Self-complementing code

1. BCD (Binary Coded Decimal) Code :

In this code each decimal number is represented by a separate group of 4- bits.

(2 3 4 5)₁₀=

- It uses only 0 to 9
- 0 to 9 are valid BCD Code
- 10, 11, 12, 13, 14, 15 are invalid BCD Code
- Coding method is very simple but it requires more number of bits .

Eg. of BCD Codes

8 4 2 1
2 4 2 1
3 3 2 1
4 2 2 1
5 2 1 1
5 3 1 1
5 4 2 1
6 3 1 1
7 4 2 1
7 4 2 1
8 4 2 1

BCD Addition

1. Express the given numbers in BCD form
2. Add the corresponding digits of the decimal numbers of each group .
3. If there is no carry and the sum term is valid code , no correction is needed
4. If there is a carry out of one group to next group , (or) if the sum term is an invalid BCD code , then add 6_{10} (0110) to the sum term of that group and the resulting carry is added to the next group .

Q) Perform the following using BCD addition

$$25+13$$

Q) Perform the following using BCD addition

$$679.6 + 536.8$$

Q. When two BCD numbers are added, under what conditions a correction factor of 6 is added to a 4-bit nibble

- a) When the nibble value is one of 1010, 1011, 1100, 1101, 1110, or 111
- b) When there is a carry out of the nibble to the next higher significant nibble
- c) When a final carry is generated
- d) When the nibble value is one of 0001, 0010, 0100, 1000,

BCD Subtraction

1. Express the given numbers in BCD form
2. Subtract the corresponding digits of the decimal numbers of each group .
3. If there is no barrow no correction is needed.
4. If there is a barrow from the next group ,or if the difference term is an invalid BCD code then 6_{10} (0110) is subtracted from the difference term of that group .

Q) Perform the following using BCD subtraction

38-15

Q) Perform the following using BCD subtraction

$$206.7 - 147.8$$

EXCESS-3 CODE

The EX-3 code can be derived from the natural BCD code by adding 3 to each coded number.

Valid EX -3 :

Invalid EX-3 :

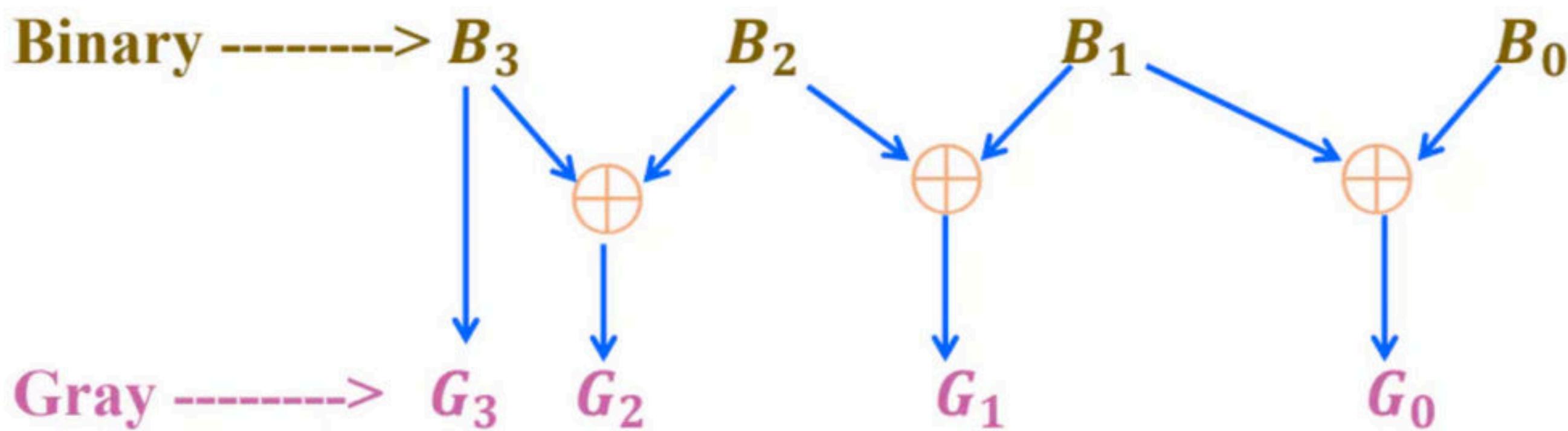
Gray Code

Gray code is a non-weighted code, successive decimal numbers are differ by only one bit .

- Non- weighted code
- Unit distance code
- Cyclic code
- Reflective code
- Minimum distance code

Decimal	1- bit Gray code	2- bit Gray code	3- bit Gray code
0			
1			
2			
3			
4			
5			
6			
7			

Binary to Gray Code



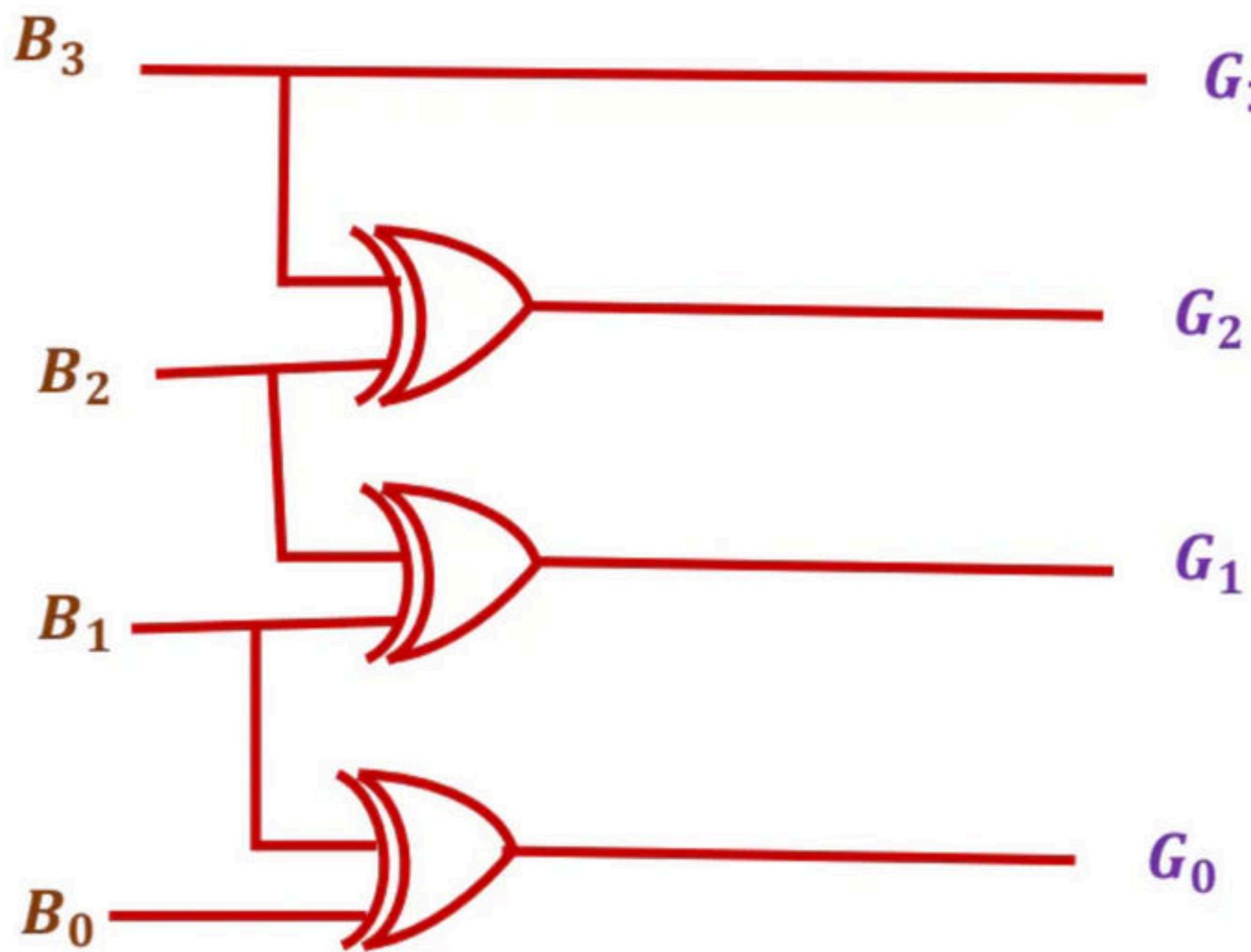
Q) Find the Gray code of the following

1 1 0 0 1 0

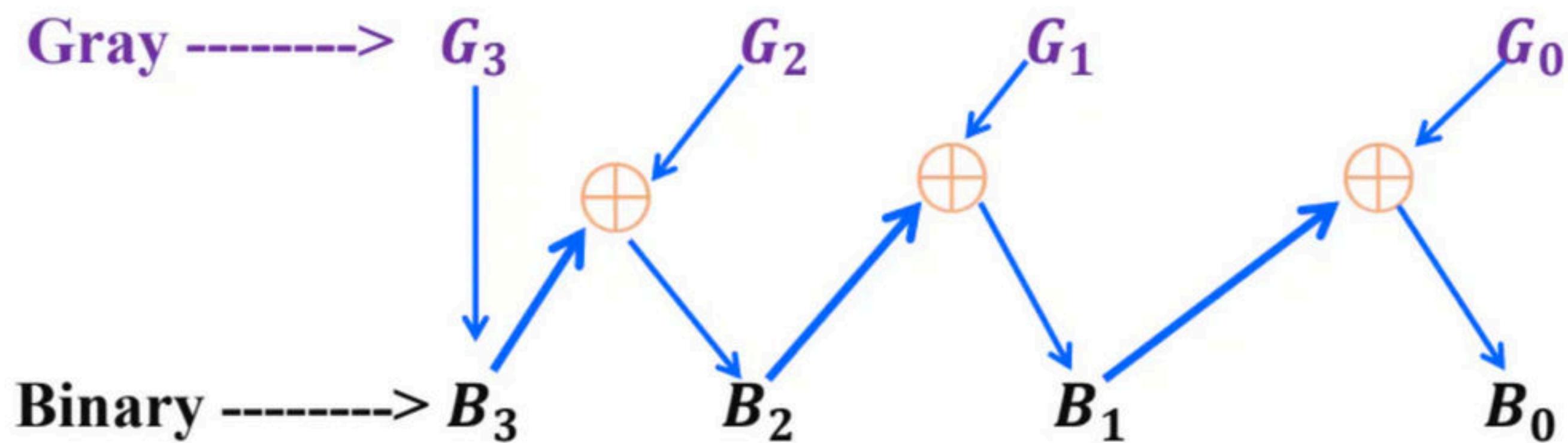
Q) Find the Gray code of the following

1 1 1 0 0 1 1 0

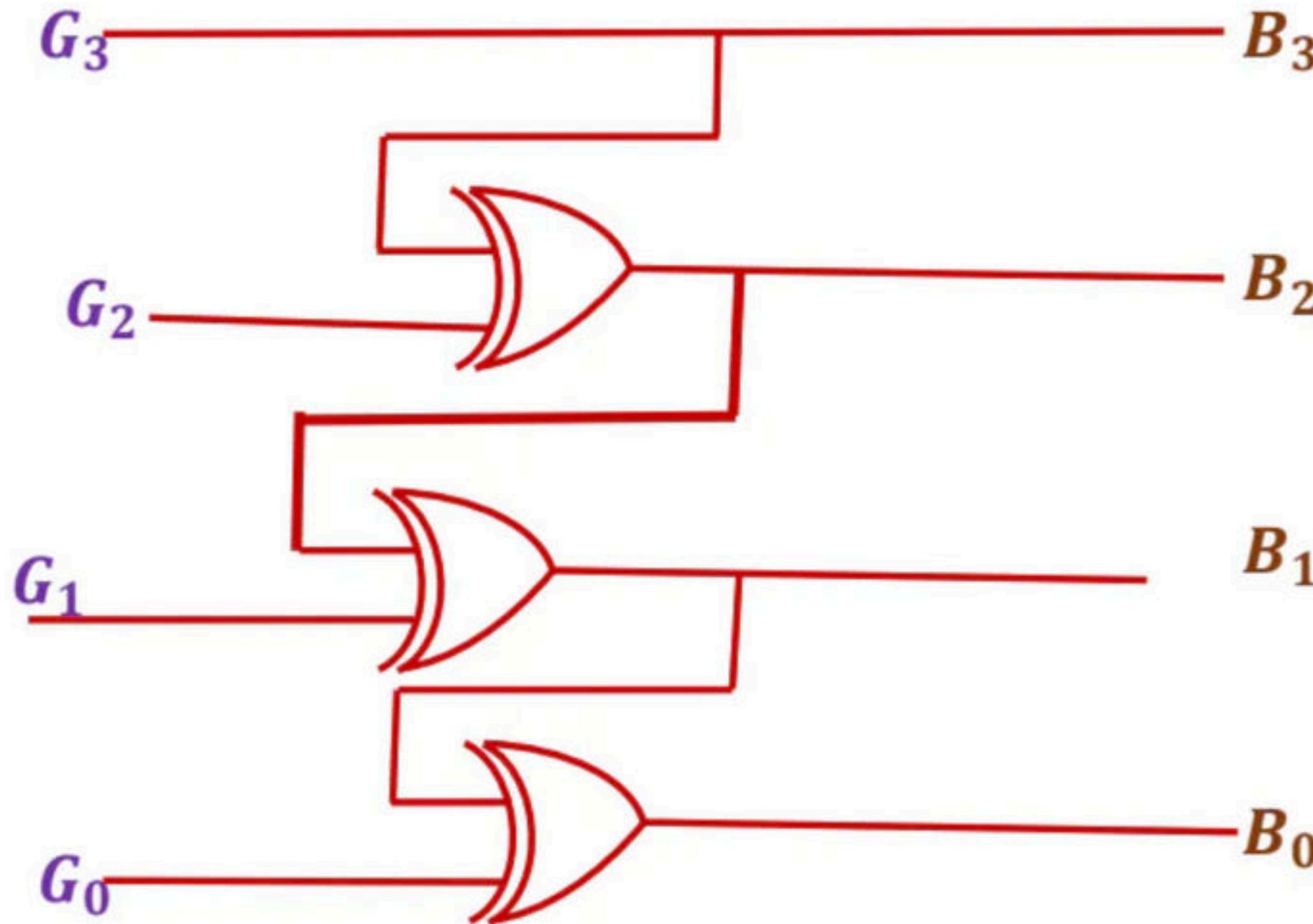
Logic gate



Gray to Binary Code



Gray Code to Binary Code



Q) Find the binary code of the following

1 0 1 0 1 1

Q) Find the binary code of the following

1 1 1 0 0 1 1 0

Self Complementing Codes

- A code is said to be self complementing, if the 1' complement of a number N is equal to the 9's complement of the number
- For a code to be self complementing, the sum of all its weights must be 9 .

Eg. of Self Complementing Codes

2	4	2	1
5	2	1	1
4	3	1	1
3	3	2	1
XS-3			

PARITY BIT

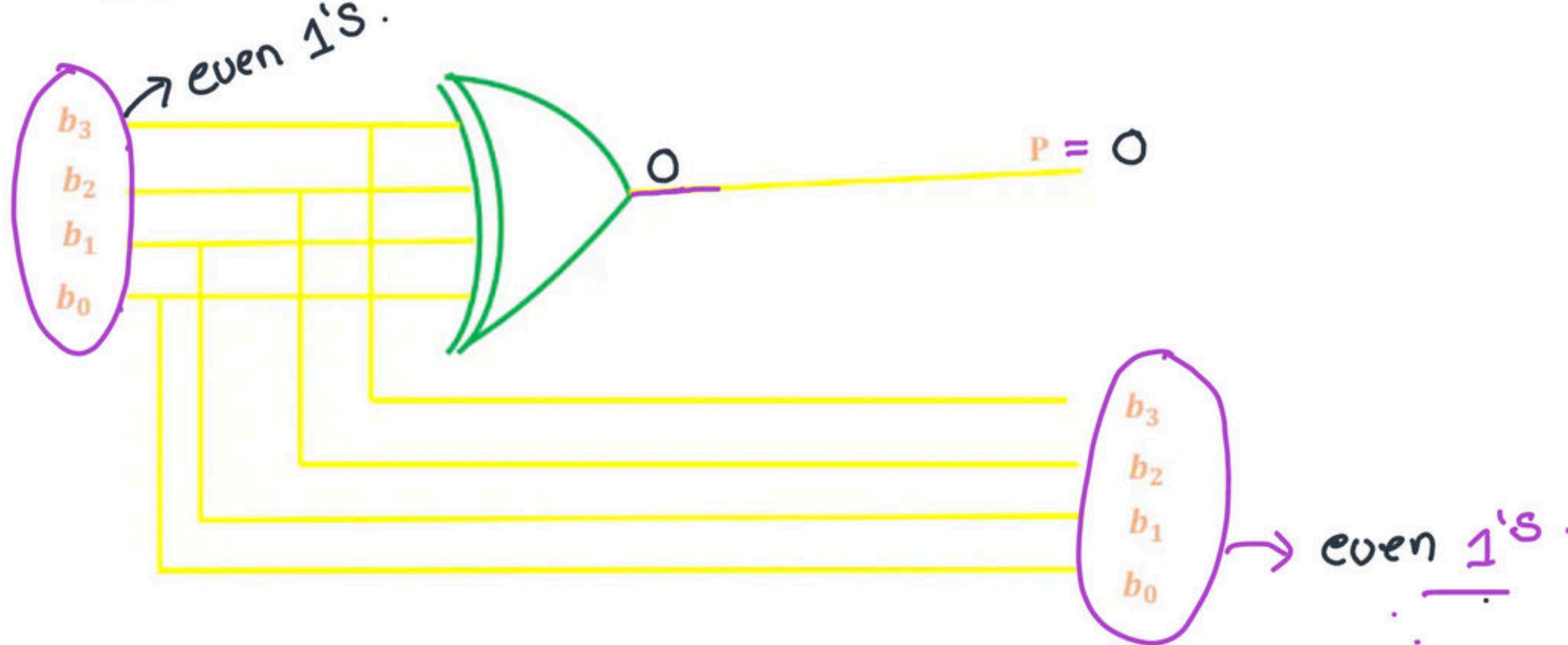
A parity bit is used for the purpose of detecting errors during transmission of binary information . A parity bit is an extra bit included with a binary message to make the number of 1's either odd or even. The message including the parity bit is transmitted and then checked at the receiving end for errors. The circuit that generates the parity bit in the transmitter is called a parity generator and the circuit that checks the parity in the receiver is called a parity checker .

Even parity

In case of even parity , the added parity bit will make the total number of 1's is an even number .

3- bit message	Message with even parity	
	message	Parity
0 0 0	0 0 0	0
0 0 1	0 0 1	1
0 1 0	0 1 0	1
0 1 1	0 1 1	0
1 0 0	1 0 0	1
1 0 1	1 0 1	0
1 1 0	1 1 0	0
1 1 1	1 1 1	1

Even Parity Generator



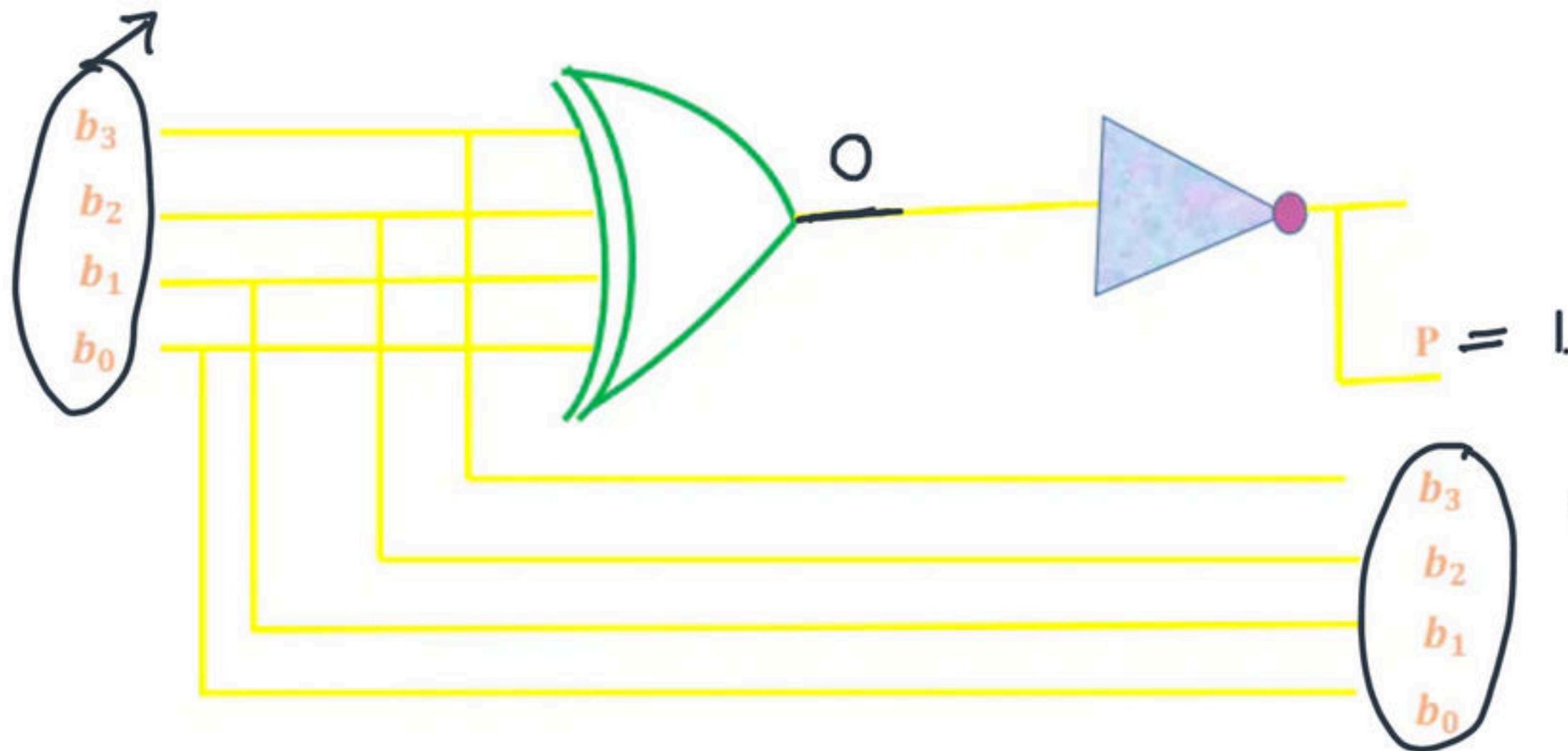
Odd parity

In case of odd parity , the added parity bit will make the total number of 1's is an odd number .

3- bit message	Message with odd parity	
	message	Parity
000	000	1
001	001	0
010	010	0
011	011	1
100	100	0
101	101	1
110	110	1
111	111	0

odd Parity Generators.

even 1's.



even Parity

(A)

1010

.

(S)

1100

Hamming Code

For the detection and correction of 1-bit errors

data bits (n) = 4.

No. of parity bits (P).

$$2^P \geq n + P + 1$$

$$\underline{n=2}$$

$$P=1$$

$$2^1 \geq 2+1+1 \times$$

$$\underline{P=2}$$

$$2^2 \geq 2+2+1 \times$$

$$\underline{P=3}$$

$$2^3 \geq 2+3+1 \checkmark$$

$n=4$

$$2^p \geq n+p+1.$$

$p=3$

$$2^3 \geq 4+3+1 \checkmark$$

p = 3

Selection of Parity bits

The hamming code uses the number of redundant bits (parity bits) depending on the number of information bits in the message.

$$2^P \geq n + P + 1$$

$$\underline{n=4}$$

$$\text{No. of parity bits (P)} = 3$$

$$\underline{n=5}.$$

$$p=4$$

$$2^4 \geq 5 + 4 + 1$$

$$\boxed{P=4} \checkmark$$

Location of Parity bits

$$\frac{2^m}{}$$

No. of Parity bits (P) = 3.

$$m=0$$

$$2^0 = 1$$

$$m=1$$

$$2^1 = 2$$

P_1 P_2 P_4

No. of Parity bits (P) = 5.

P_1 $\underline{P_2}$ P_4 P_8 P_{16}

No. of data bits (n) = 4. $a b c d$

No. of parity bits (p) = 3. $P_1 \ P_2 \ P_4$

7 111	6 110	5 101	4 100	3 011	2 010	1 001
a .	b	c	P_4	d	P_2	P_1

even parity

odd parity

$P_1 \rightarrow P_1 \text{ dca}$

$P_2 \rightarrow P_2 \text{ dba}$

$P_4 \rightarrow P_4 \text{ cba}$

$$n = 5$$

$$n + p = 9$$

$$p = 4$$

$P_1 \quad P_2 \quad P_4 \quad P_8$

a b c d e
←

9 1001	8 1000	7 0111	6 0110	5 0101	4 0100	3 0011	2 0010	1 0001
a	P_8	b	c	d	P_4	e	P_2	P_1

$$P_1 \rightarrow P_1 e d b a$$

$$P_8 \rightarrow P_8 a$$

$$P_2 \rightarrow P_2 e c b$$

$$P_4 \rightarrow P_4 d c b$$

Q) data bits = 11001.

develop the hamming code.

$$\begin{array}{l|l} n=5 & n+p=9 \\ p=4 & \\ 2^p \geq p+n+1 & \end{array}$$

9 1001.	8 1000	7 0111	6 0110	5 0101	4 0100	3 0011	2 0010	1 0001
1	P_8	1	0	0	P_4	1	P_2	P_1

$$P_1 \rightarrow P_1 1011$$

$$P_2 \rightarrow P_2 101$$

$$P_4 \rightarrow P_4 001$$

$$P_8 \rightarrow P_8 1.$$

Even Parity

$$P_1 = 1$$

$$P_2 = 0$$

$$P_4 = 1$$

$$P_8 = 1$$

Odd Parity

$$P_1 = 0$$

$$P_2 = 1$$

$$P_4 = 0$$

$$P_8 = 0$$

even Parity hamming code.

1	1	1	0	0	1	1	0	1
---	---	---	---	---	---	---	---	---

odd Parity hamming code.

1	0	1	0	0	0	0	1	1	0
---	---	---	---	---	---	---	---	---	---

Assigning the values to Parity bits

- Each parity bit check the corresponding bit locations and assign the bit values as 1 or 0 so as to make the number of 1 s as even for even parity and odd for odd parity .

Bit location	7	6	5	4	3	2	1
Bit designation							
Binary representation							
Data bits							
Parity bits							

Q. Encode a binary word 11001 into the even parity hamming code.

Q. Let us assume the even parity hamming code , if the received code is 1100100 , verify whether the received data is correct or not , if not correct find the correct data .

$$n + p = 7$$

$$n = 4 \quad p = 3.$$

$$P_1 \rightarrow P_1 | 01 \Rightarrow P_1 = 0$$

7	6	5	4	3	2	1
0111	0110	0101	0100	0011	0010	0001
1	1	0	P_4	1	P_2	P_1
1	1	0	0	1	0	1

$$P_2 \rightarrow P_2 | 11 \Rightarrow P_2 = 1$$

$$P_4 \rightarrow P_4 | 01 \Rightarrow P_4 = 0$$

$$\begin{matrix} P_4 & P_2 & P_1 \\ 0 & 1 & 0 \end{matrix}$$

Q) even parity hamming code

Received data = 111001101
 $n+p=9 \Rightarrow n=5 \quad p=4$

9	8	7	6	5	4	3	2	1
1001	1000	0111	0110	0101	0100	0011	0010	0001
1	P ₈ ✓	1	0	0	1 ✓	1	0 ✓	1 ✓

$$P_1 | 0111 \rightarrow P_1 = 1$$

$$P_8 | \rightarrow P_8 = 1 \checkmark$$

$$P_2 | 01 \rightarrow P_2 = 0$$

$$P_4 | 001 \rightarrow P_4 = 1$$

a) even parity hamming code

Received data = 11000 1101

$$n=5 \quad p=4$$

9 1001	8 1000	7 0111	6 0110	5 0101	4 0100	3 0011	2 0010	1 0001
l.	P ₈	0	0	0	1 P ₄	1	0 P ₂	1 P ₁

$$P_1 + 0\ 0\ 1 \rightarrow P_1 = 0$$

$$P_8 = 1$$

$$P_2$$

$$P_4$$

$$P_2 = 1$$

$$P_4 = 0$$

$$\begin{matrix} P_8 & P_4 & P_2 & P_1 \\ | & 0 & 1 & 0 \end{matrix}$$

Q. P, Q and R are the decimal integers corresponding to the 4-bit binary number 1100 considering in signed magnitude, 1's complement and 2's complement representations, respectively. The 6-bit 2's complement representation of $(P+Q+R)$ is.....

- a) 111101
- ~~b) 110101~~
- c) 110010
- d) 111001

$$\begin{array}{r} \text{1100} \\ \hline P = -[100] = -4 \\ Q = -[0011] = -3 \\ R = -[0100] = -4 \end{array}$$

$$\begin{array}{r} | P + Q + R = -11 \\ | \\ | \quad 11 \rightarrow 001011 \\ | \\ | \quad 110101 \\ | \\ | \hline \end{array}$$

Q. Two numbers are chosen independently (with replacement) and uniformly at random from the set {1,2,3,.....13} . The probability that their 4-bit unsigned binary representations have the same most significant bit -----

1 0001

2 0010

3 0011

4 0100

5 0101

6 0110

7 0111

8 1000

9 1001

10 1010

11 1011

12 1100

13. 1101

$$P = \frac{7}{13} \cdot \frac{7}{13} + \frac{6}{13} \cdot \frac{6}{13}$$

$$P = 0.5$$

Q. Which of the following represents ' $E3_{16}$ '?

(a) $(CE)_{16} + (A2)_{16}$

(c) $(2BC)_{16} - (1DE)_{16}$

(b) $(1BC)_{16} - (DE)_{16}$

(d) ~~$(200)_{16} - (11D)_{16}$~~

$$\begin{array}{r} C \ E \\ A \ 2 \\ \hline (1 \ 7 \ 0)_{16} \end{array}$$

b) $\begin{array}{r} | \ B \ C \\ D \ E \\ \hline D \ E \end{array}$

c) $\begin{array}{r} 2 \ BC \\ 1 \ DE \\ \hline 0 \ DE \end{array}$

~~200~~

$$\begin{array}{r} 1 \ 1 \ D \\ \hline E \ 3 \end{array}$$

~~2000~~

$$\begin{array}{r} 1789 \\ \hline 0211 \end{array}$$

10

$$\begin{array}{r} 12 \\ 16 \\ \hline 28-14 \end{array}$$

$10+16=26$

$$\begin{array}{r} 26 \\ 13 \\ \hline 13 \end{array}$$

Q. A new Binary Coded Pentary (BCP) number systems is proposed in which every digit of a base -5 number is represented by its corresponding 3-bit binary code. For example, the base-5 number 24 will be represented by its BCP code 010100 . In this numbering system, the BCP code 100010011001 corresponds to the following number in base -5 system

- a)423
- b)1324
- c)2201
- d)4231

Q. Consider the addition of numbers with different bases

$$(X)_7 + (Y)_8 + (W)_{10} + (Z)_5 = (K)_9$$

If X=36 , Y = 67 ,W=98 and K =241 then Z is

$$(19)_{10} = (34)_5.$$

- a)34
 - b)15
 - c)68
 - d)25

$$(36)_7 = 3(7) + 6 = (27)_{10}$$

$$(67)_8 = 6(8) + 7 = (55)_{10}$$

$$\omega = (98)_{10}$$

$$(241)_9 = 2(81) + 4(9) + 1 = 199.$$

$$(E)_5 = 199 - 27 - 55 - 98 = 19.$$

$$5 \overline{)19}$$

Q. Let x_1 be the maximum value that can be represented in signed 2's complement form using 6 bits.

Let x_2 be the minimum value that can be represented in signed 1's complement form using 5 bits.

Let x_3 be the minimum value that can be represented in signed 2's complement form using 7 bits.

Then find the value of $x_1 + \frac{x_2}{2} + 2x_3$

- (a) -180.5
(c) 108.5

- ~~(b)~~ -104.5
(d) 130.5

$$x_1 = 31$$

$$x_2 = -15$$

$$x_3 = -64$$

2's complement form

$$\text{Range} = \underline{-\left(2^{n-1}\right)} \text{ to } \underline{+\left(2^{n-1}-1\right)}$$

$$1\text{'s comp. Range} = \underline{-\left(2^{n-1}-1\right)} \text{ to } \underline{+\left(2^{n-1}-1\right)}$$

$$x_1 + \frac{x_2}{2} + 2x_3 = 31 - 7.5 - 128 \\ = -104.5$$

Q. The 4-bit binary number 1110 represents

- (a) $(-1)_{10}$ in signed magnitude system and $(-2)_{10}$ in signed 2's complement system
- (b) $(-1)_{10}$ in signed 1's complement system and $(-2)_{10}$ in signed 2's complement system
- (c) $(-2)_{10}$ in signed magnitude system and $(-1)_{10}$ in signed 2's complement system
- (d) $(-6)_{10}$ in signed magnitude system and $(-1)_{10}$ in signed 2's complement system

Q. A number N of base r is represented as $(N)_r$ let $(10_{16})^3 = (X)_{10}^2$

- a)64
- b)15
- c)22
- d)19

Combinational Logic Circuit

Combinational Logic Circuit

- The present output depends on present input only
- In combinational circuits feedback and clock is not present

- HA
- HS
- FA
- FS
- Parallel Adder
- Carry look ahead Adder
- Binary Multiplier
- Magnitude Comparators
- Multiplexer
- Demultiplexers
- Decoder
- Encoder
- Priority Encoder
- Code converters

Half Adder

For the addition of two single bits



A	B	Sum	Carry

Logic Circuit

Half Adder using NAND Gates

Half Adder using NOR Gates

Half Subtractor (A-B)

For the subtraction of two single bits



A	B	Difference	Borrow

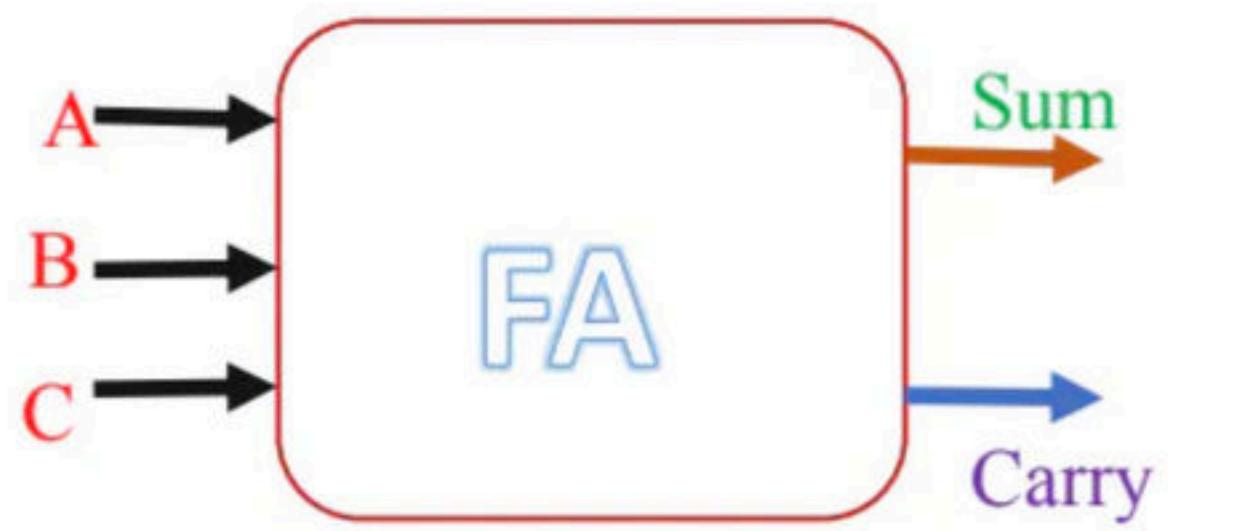
Logic Circuit

Half Adder / Half Subtractor

Half Subtractor using NAND Gates

Half Subtractor using NOR Gates

Full Adder



Full Adder with two Half Adders

Full Adder with two Half Adders

Full Adder using NAND Gates

Full Adder using NOR Gates

Full Subtractor(A - B-C)



Logic Circuit

Full Subtractor with two Half Subtractors

Full Subtractor with two Half Subtractors

Full Subtractor using NAND Gates

Full Subtractor using NOR Gates

FS : A- B- C

FS : B- C- A

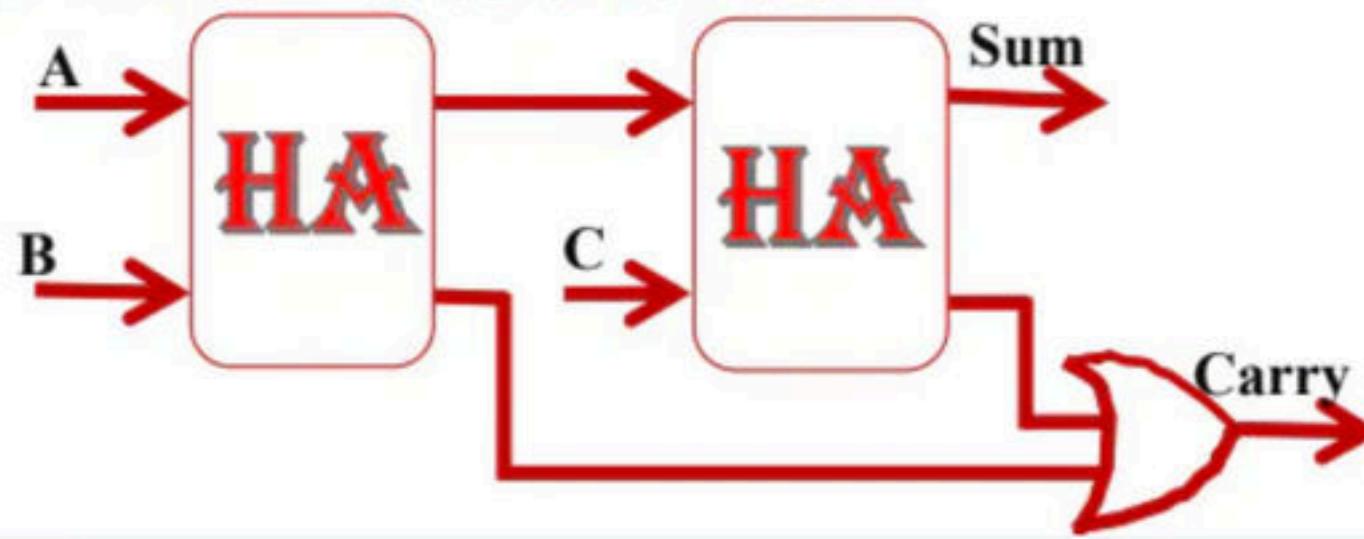
FS : C- A- B

HA

1. Logical expression for Sum =
2. Logical expression for Carry =
3. Minimum number of NAND Gates =
4. Minimum number of NOR Gates =

FA

1. Logical expression for Sum =
2. Logical expression for Carry =
3. Minimum number of NAND Gates =
4. Minimum number of NOR Gates =

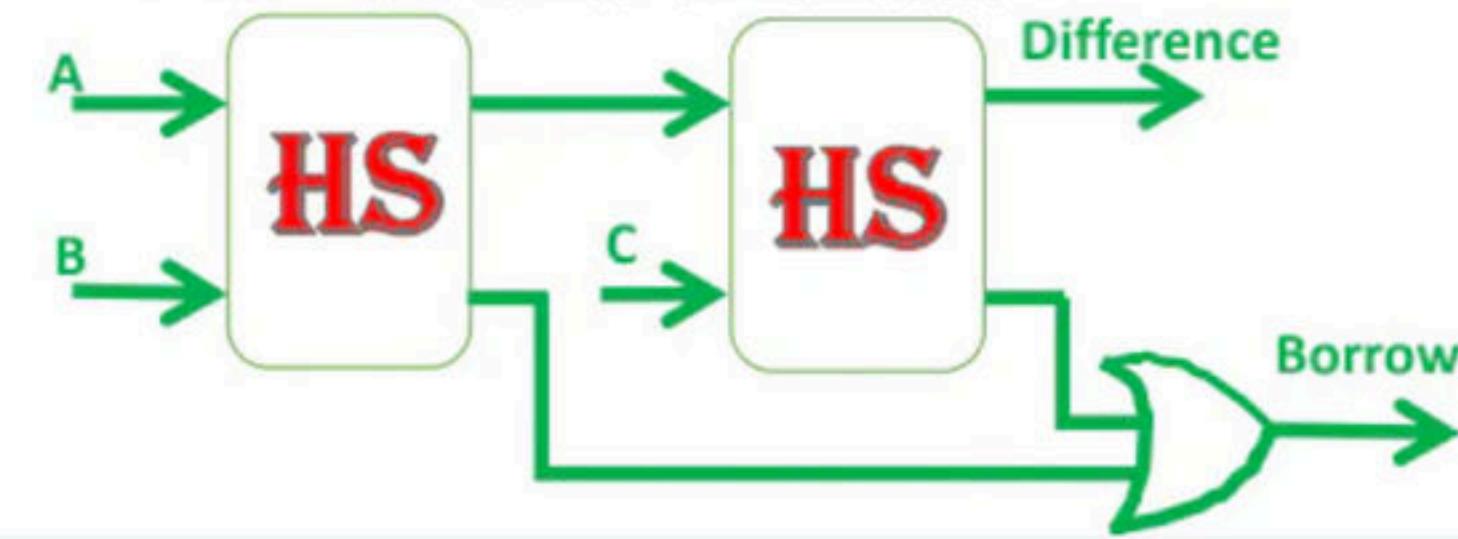


HS

1. Logical expression for Difference =
2. Logical expression for Borrow=
3. Minimum number of NAND Gates =
4. Minimum number of NOR Gates =

FS

1. Logical expression for Difference=
2. Logical expression for Borrow =
3. Minimum number of NAND Gates =
4. Minimum number of NOR Gates =

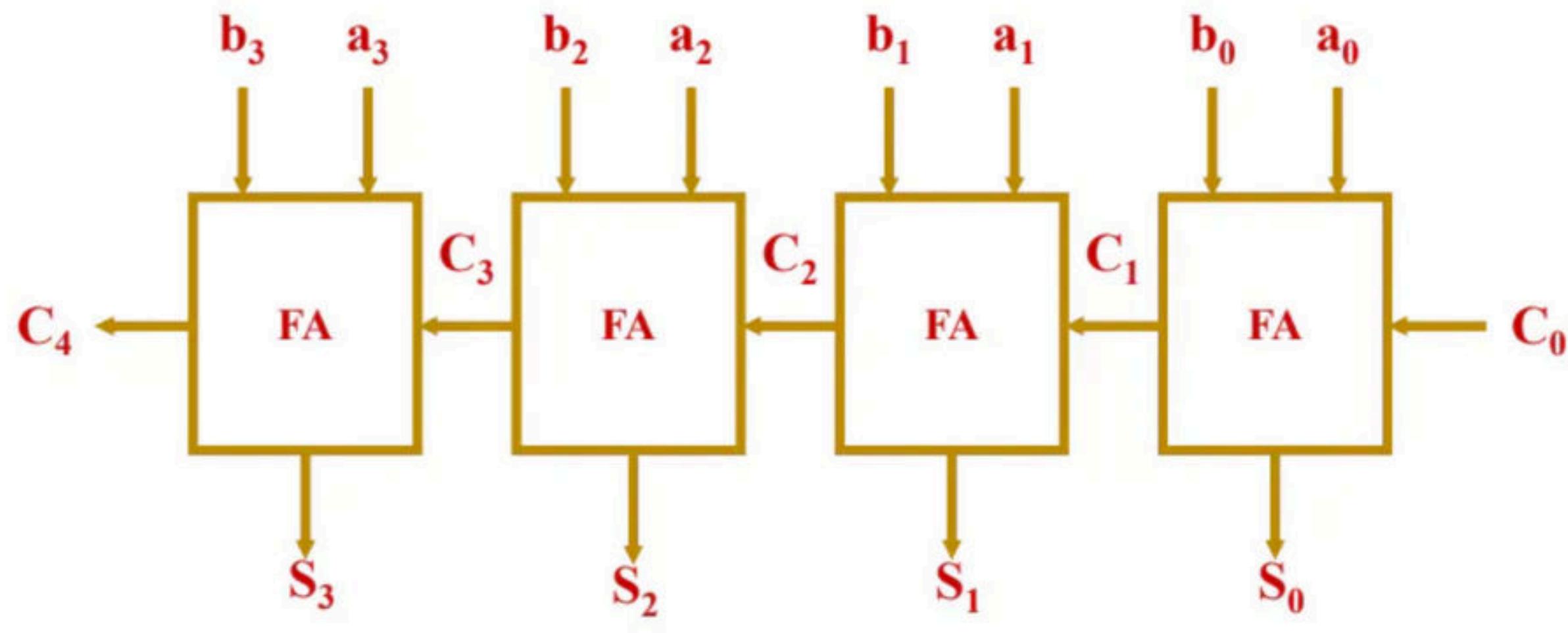


Ripple Carry Adder (Parallel Adder)

A $\longrightarrow a_3 \quad a_2 \quad a_1 \quad a_0$

B $\longrightarrow b_3 \quad b_2 \quad b_1 \quad b_0$





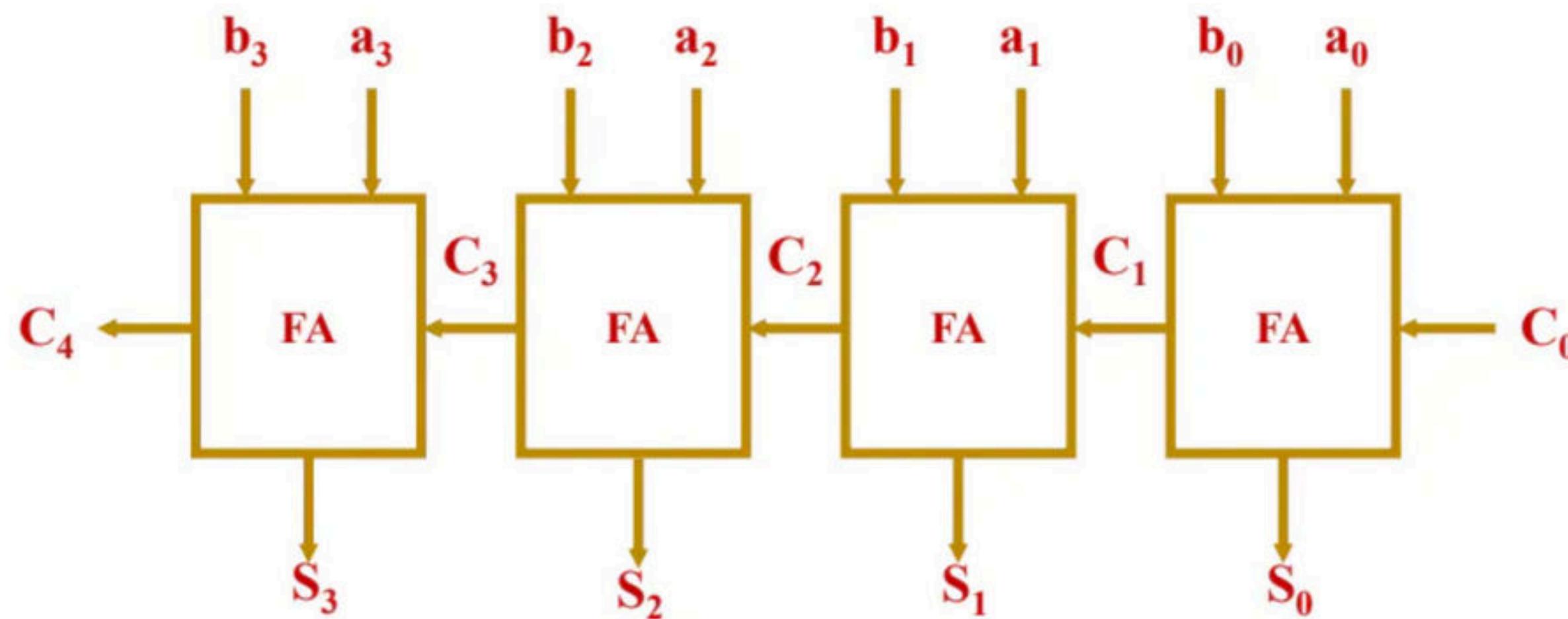
Note :

To implement 4-bit parallel adder

To implement n-bit parallel adder

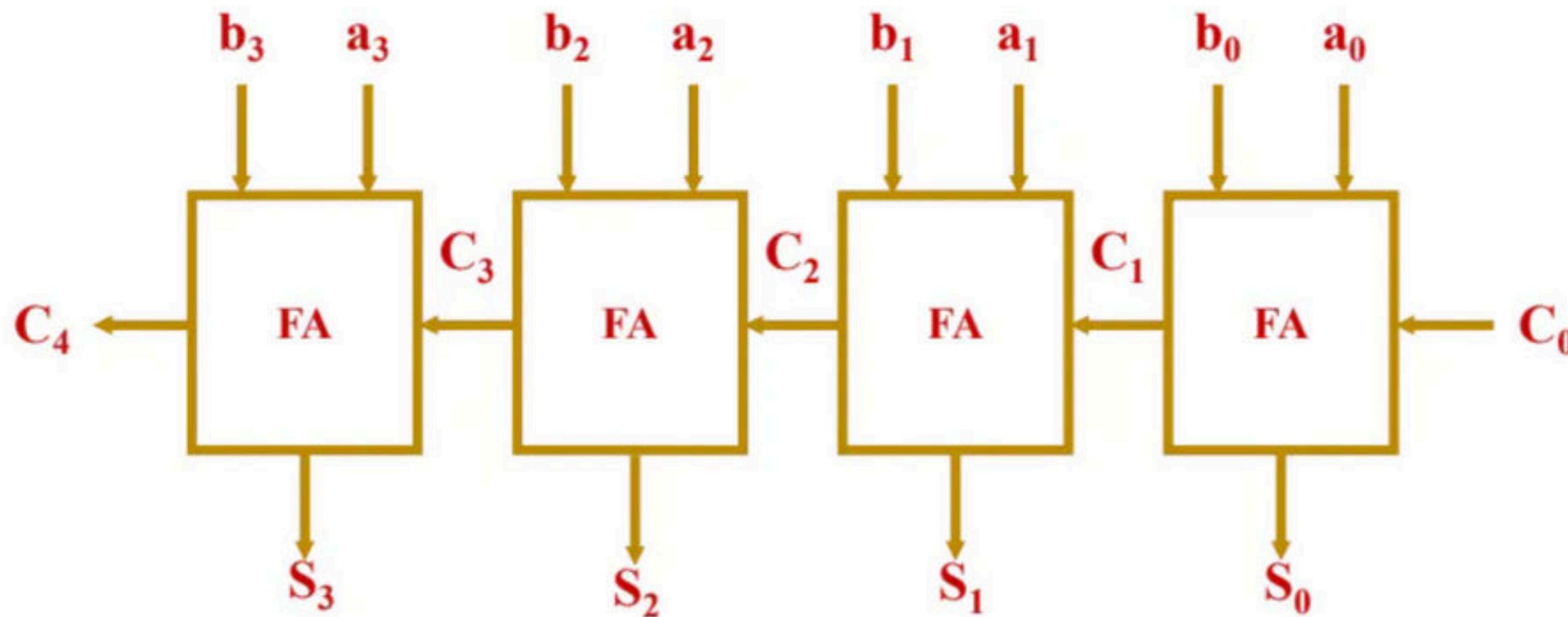
Delay Analysis

Case : 1 $(tpd)_{sum} > (tpd)_{carry}$



Delay Analysis

Case : 2 $(tpd)_{sum} < (tpd)_{carry}$



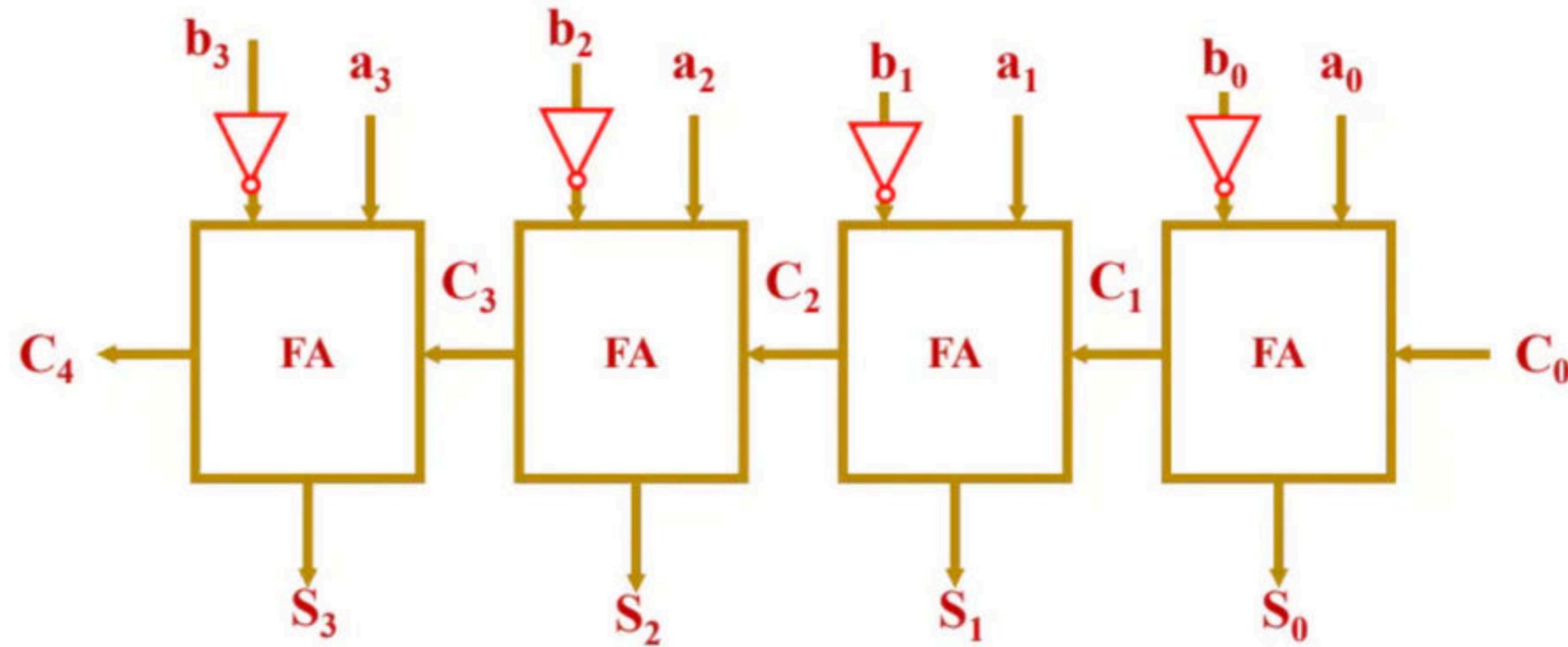
In general for n- bit Parallel Adder

Delay =

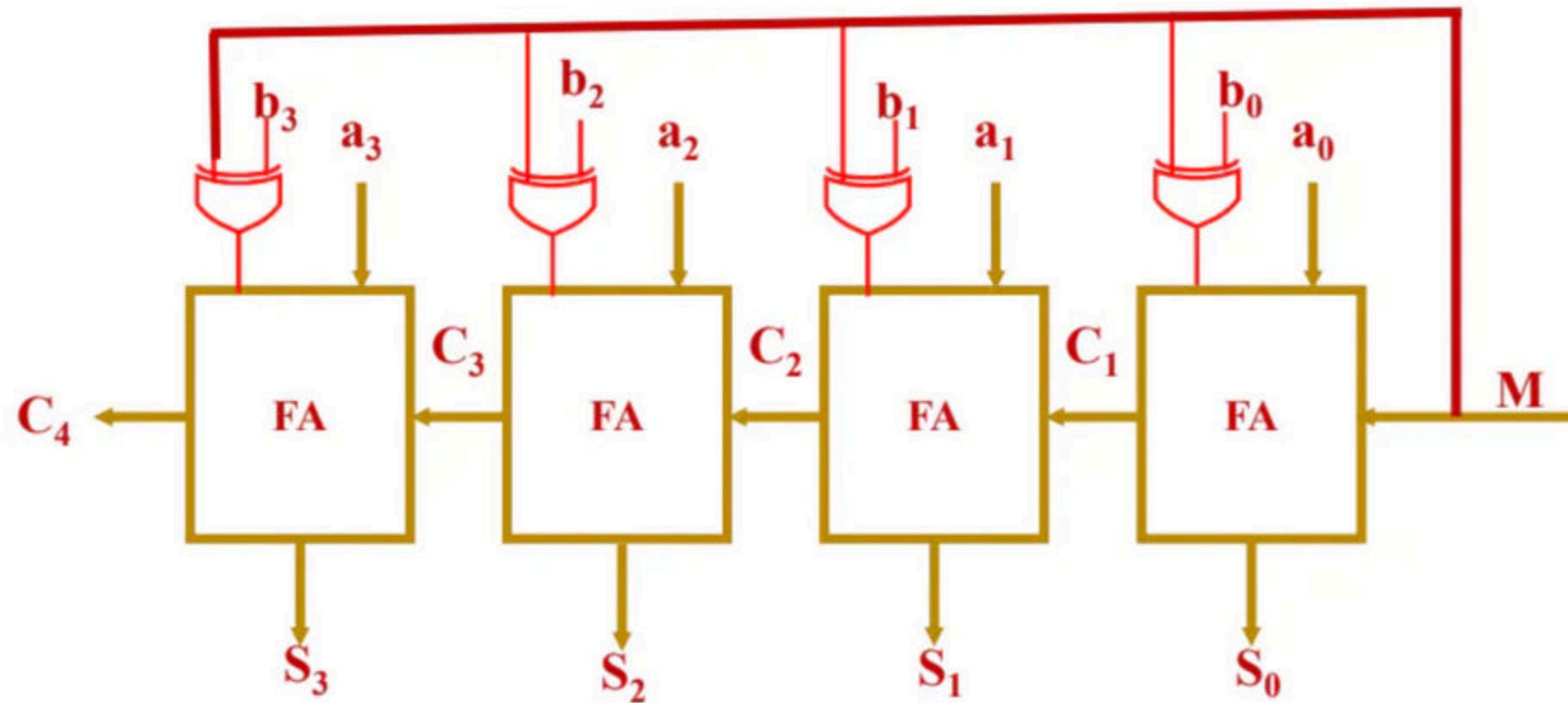
Q) A 16-bit RCA is realized using 16 identical FAs , if the (tpd)carry = 12ns ,
(tpd)sum= 15ns , then the overall delay is ----- ns

Q) A 4-bit RCA is implemented using 4-FAs , if the propagation delay of XOR – Gate is twice the delay of AND/OR Gate , then the overall delay of 4- bit RCA if the delay of AND/OR Gate is $1.2 \mu\text{sec}$

Parallel Subtractor

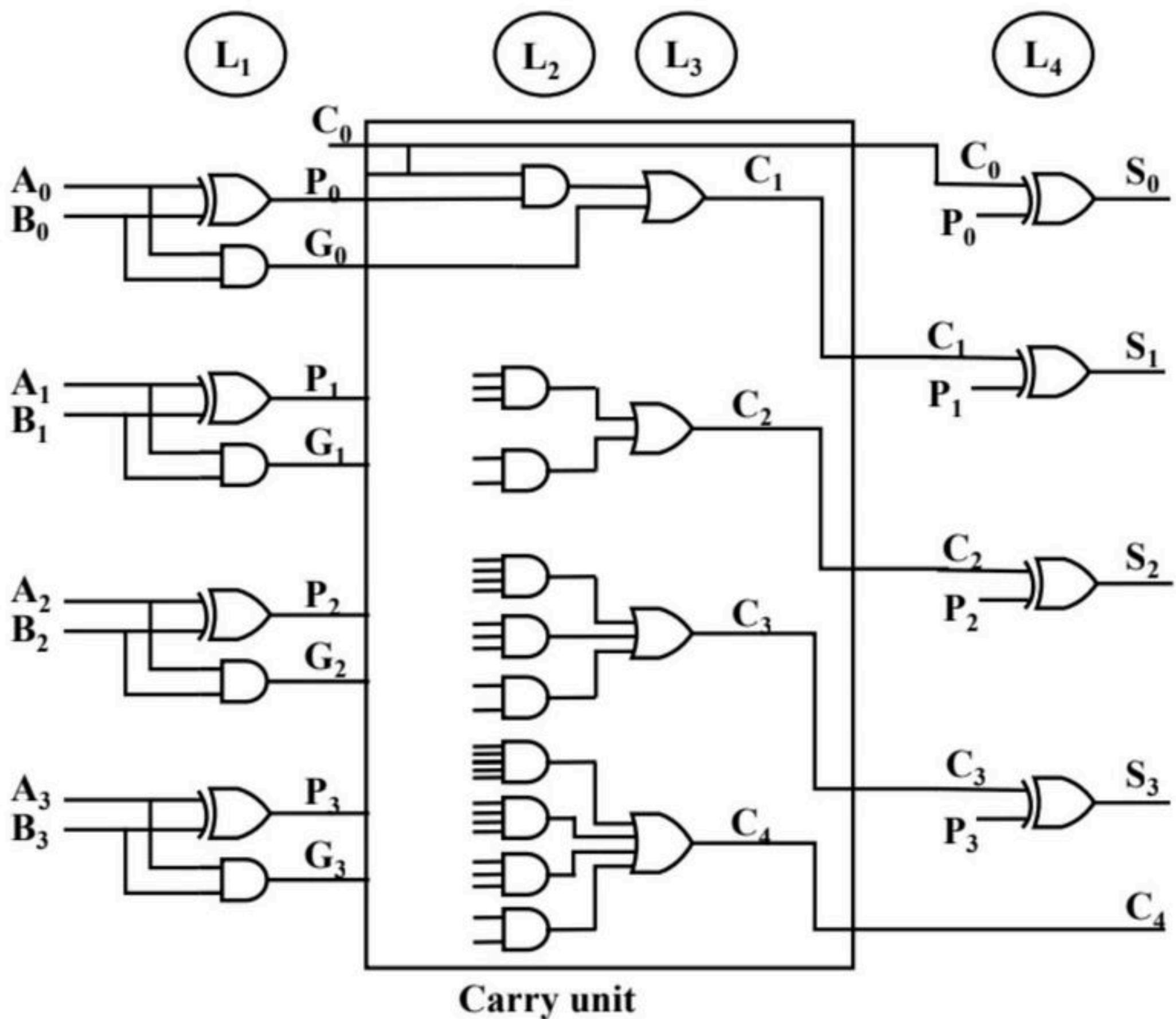


Parallel Adder/ Subtractor



Look Ahead Carry Adder

- In this adder ,the carry dependency of Ripple Carry Adder (RCA) is eliminated
- This is the fastest adder among all
- This adder have the maximum complexity



Hardware Requirements

L 1 :

L 2 :

L 3 :

L 4 :

Total number of gates for carry =

Total number of gates for sum =

Delay Analysis

Carry =

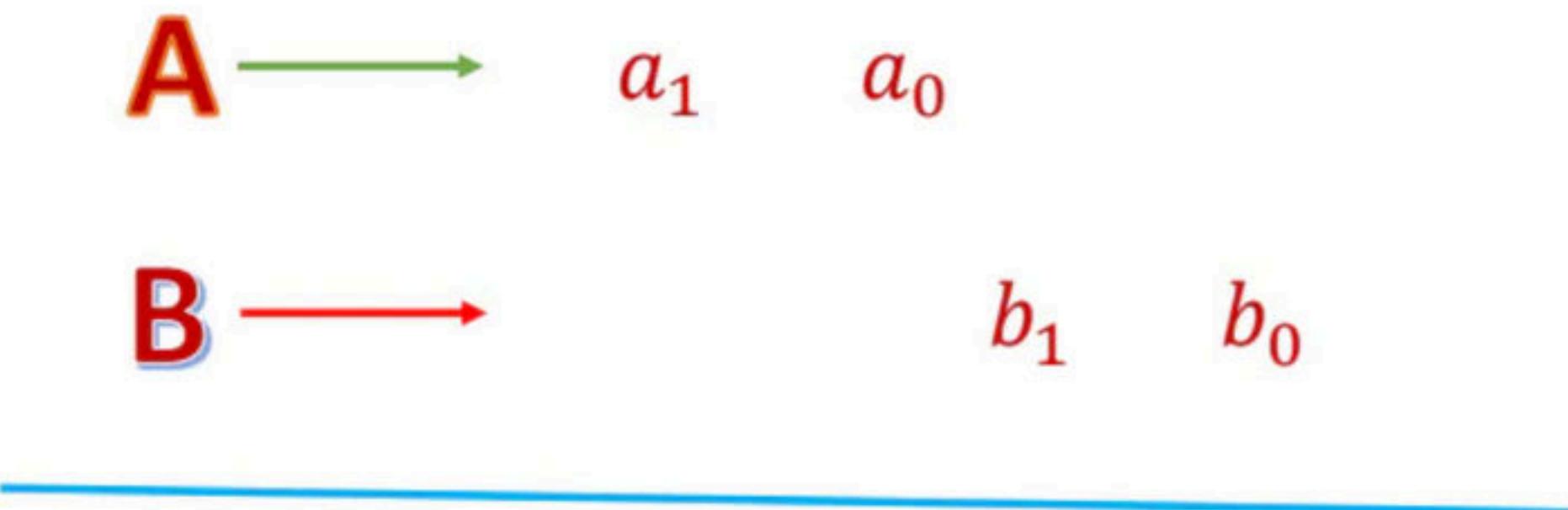
Sum =

Q) The minimum number of gates required for the implementation 4- bit look ahead carry adder are -----

Q) During the implementation of carry look ahead adder , if carry generator (G_i) and carry propagator (P_i) are available, then the minimum number of gates required are

Q) In 4- bit look ahead carry adder is implemented with the following gates NOT , AND, OR , NAND , NOR calculate the minimum time required to generate sum if each gate has 1 unit

Binary Multiplier



A 

a_2 a_1 a_0

B 

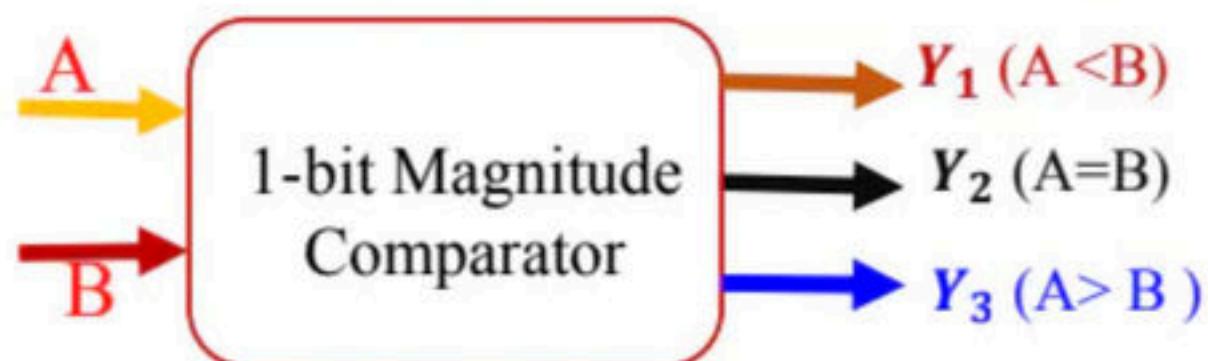
b_1 b_0



Magnitude Comparator

To compare the magnitude of two binary numbers .

1-bit magnitude comparator



A	B	Y_1 (A < B)	Y_2 (A = B)	Y_3 (A > B)

Y_1 (A < B) =

Y_2 (A = B) =

Y_3 (A > B) =

Logic circuit

1-bit Magnitude Comparator

Total number of input combinations =

Lesser than combinations =

Greater than combinations =

Equal combinations =

2-bit Magnitude Comparator



For 2-bit Magnitude Comparator

Total number of input combinations =

Lesser than combinations =

Greater than combinations =

Equal combinations =

For 2-bit Magnitude Comparator

$Y_1 (A < B) =$

$Y_2 (A = B) =$

$Y_3 (A > B) =$

For 3- bit Magnitude Comparator

$Y_1 (A < B) =$

$Y_2 (A = B) =$

$Y_3 (A > B) =$

For 4-bit Magnitude Comparator

$Y_1 (A < B) =$

$Y_2 (A = B) =$

$Y_3 (A > B) =$

For n-bit Magnitude Comparator

Total number of input combinations =

Lesser than combinations =

Greater than combinations =

Equal combinations =

Q) Find the number of 1- bit comparators , AND gates and OR gates required to implement 2- bit Comparator

Q) Find the number of 1- bit comparators , AND gates and OR gates required to implement 4- bit Comparator

Parity bit

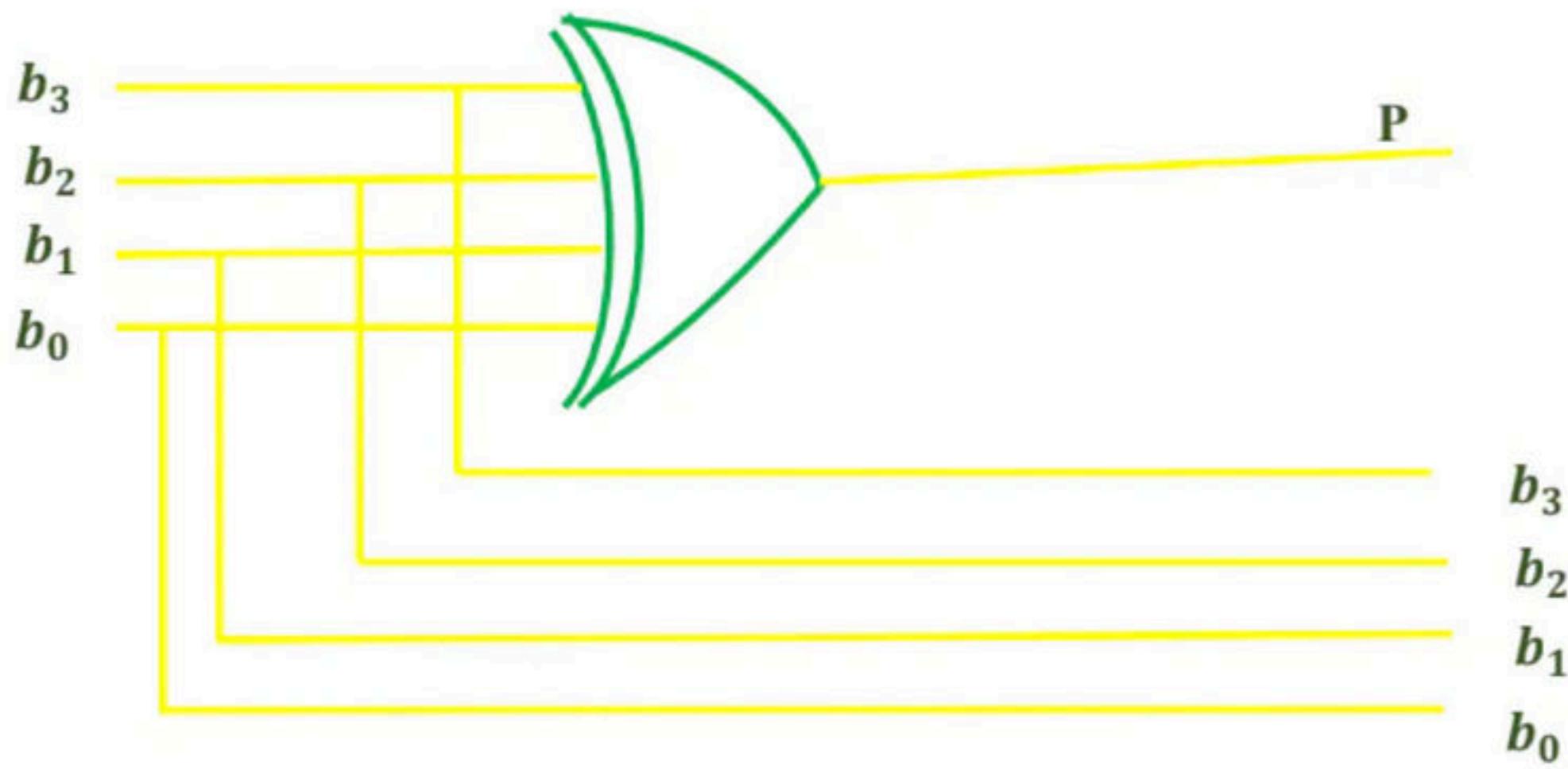
A parity bit is used for the purpose of detecting errors during transmission of binary information . A parity bit is an extra bit included with a binary message to make the number of 1's either odd or even. The message including the parity bit is transmitted and then checked at the receiving end for errors. The circuit that generates the parity bit in the transmitter is called a parity generator and the circuit that checks the parity in the receiver is called a parity checker .

Even parity

In case of even parity , the added parity bit will make the total number of 1's an even number .

3- bit message	Message with even parity	
	message	Parity

Even parity generator

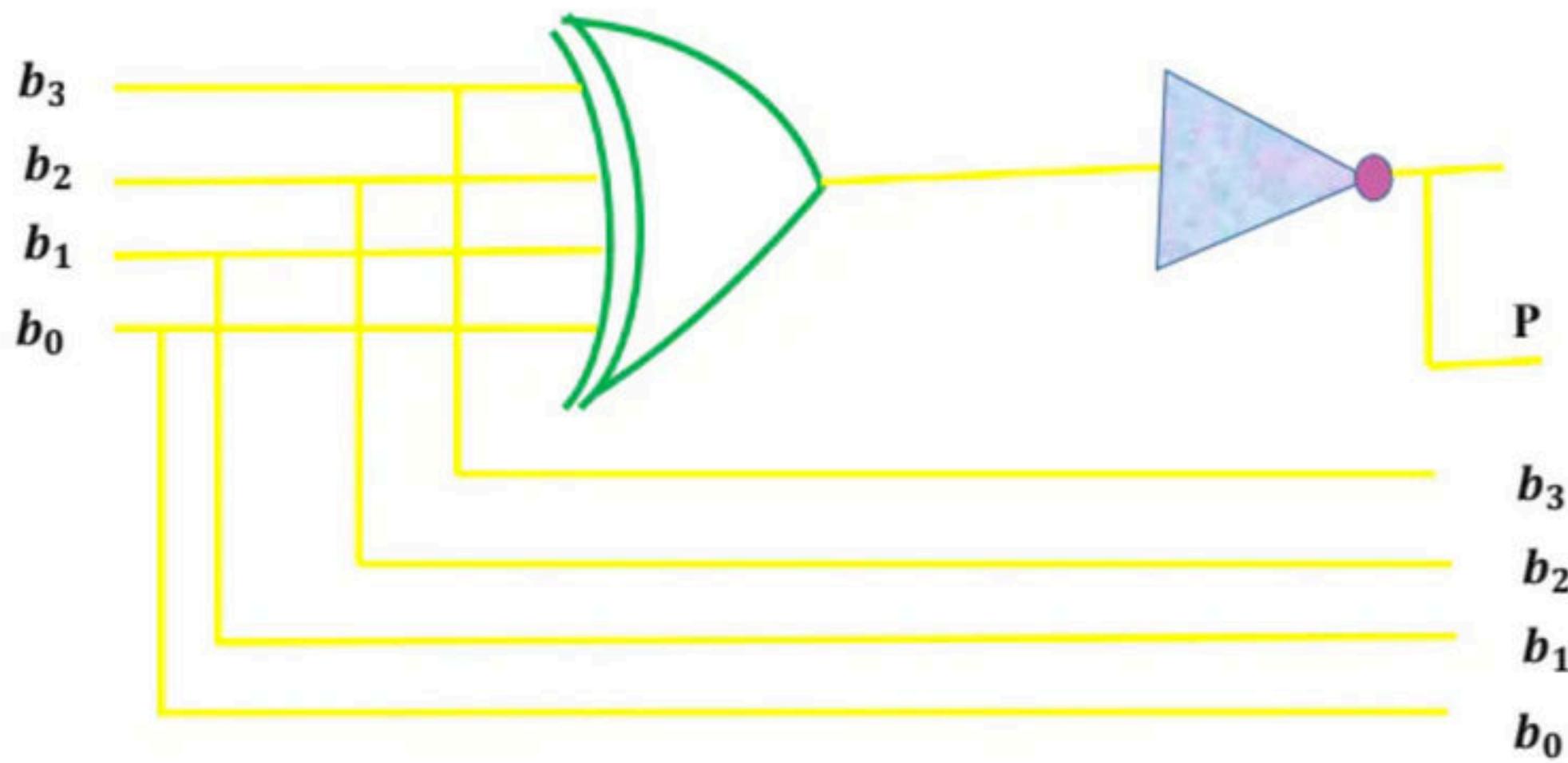


Odd parity

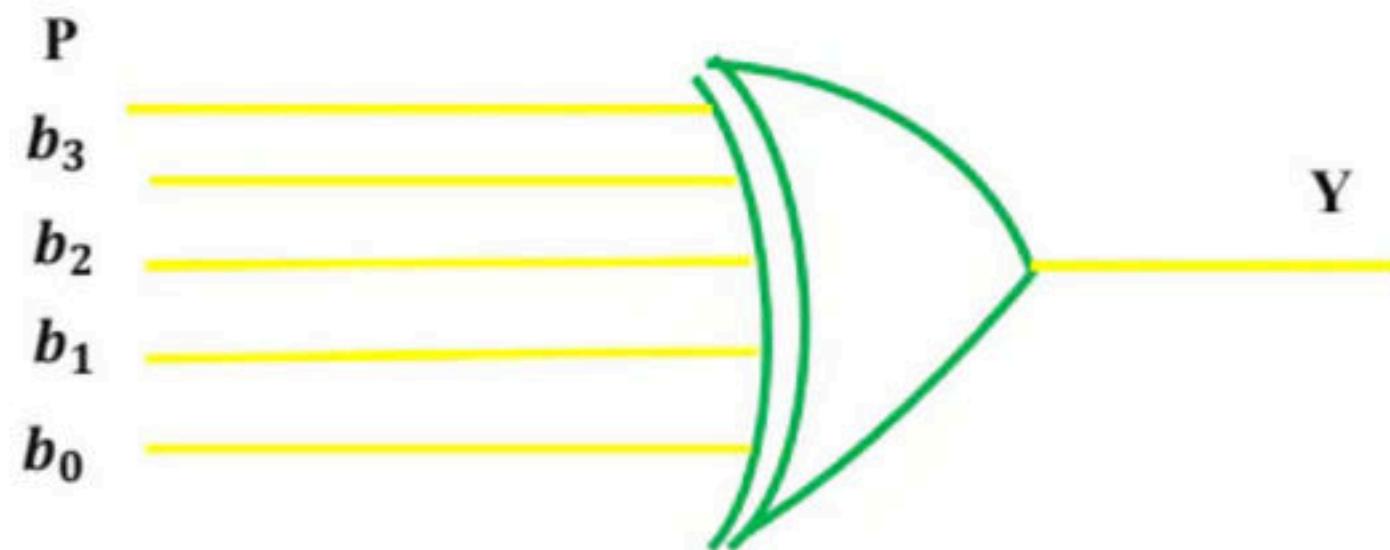
In case of odd parity , the added parity bit will make the total number of 1's an odd number .

3- bit message	Message with odd parity	
	message	Parity

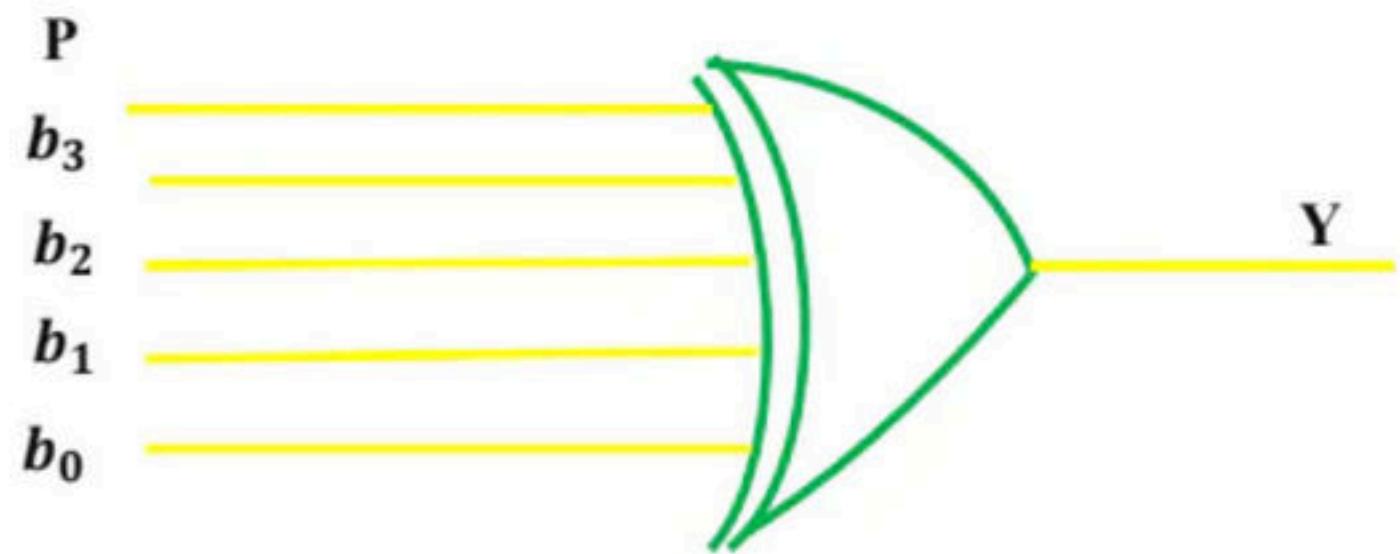
Odd parity generator



Parity Checker



Parity Checker



Hazards

Hazards are unwanted switching transients that may occurs due to unequal propagation delay of the paths , such a transient is also called a glitch or spurious spike .

Hazards

Static Hazards

Static '1
Hazards Static '0
Hazards

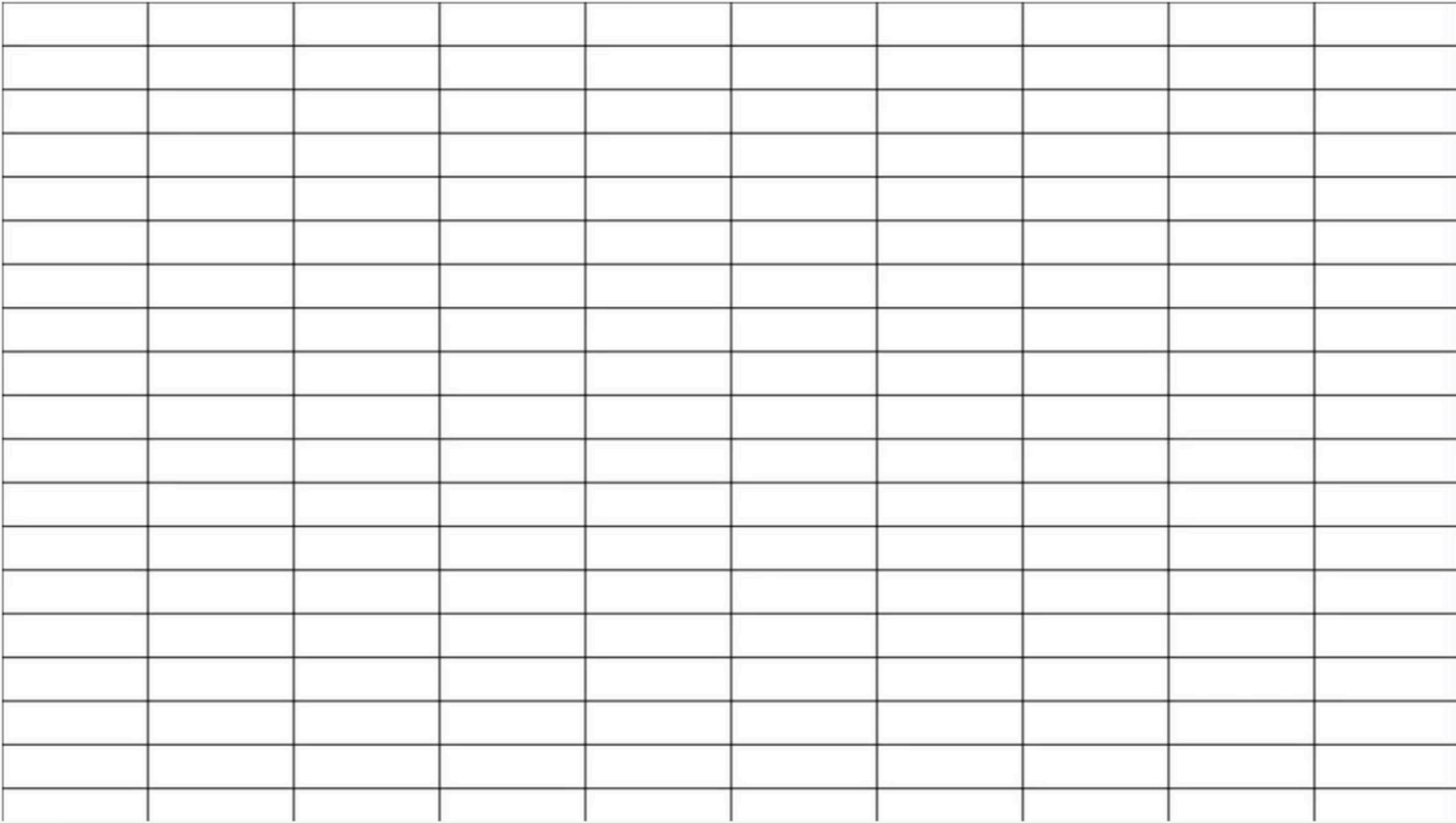
Dynamic Hazards

Occurs in multilevel circuits

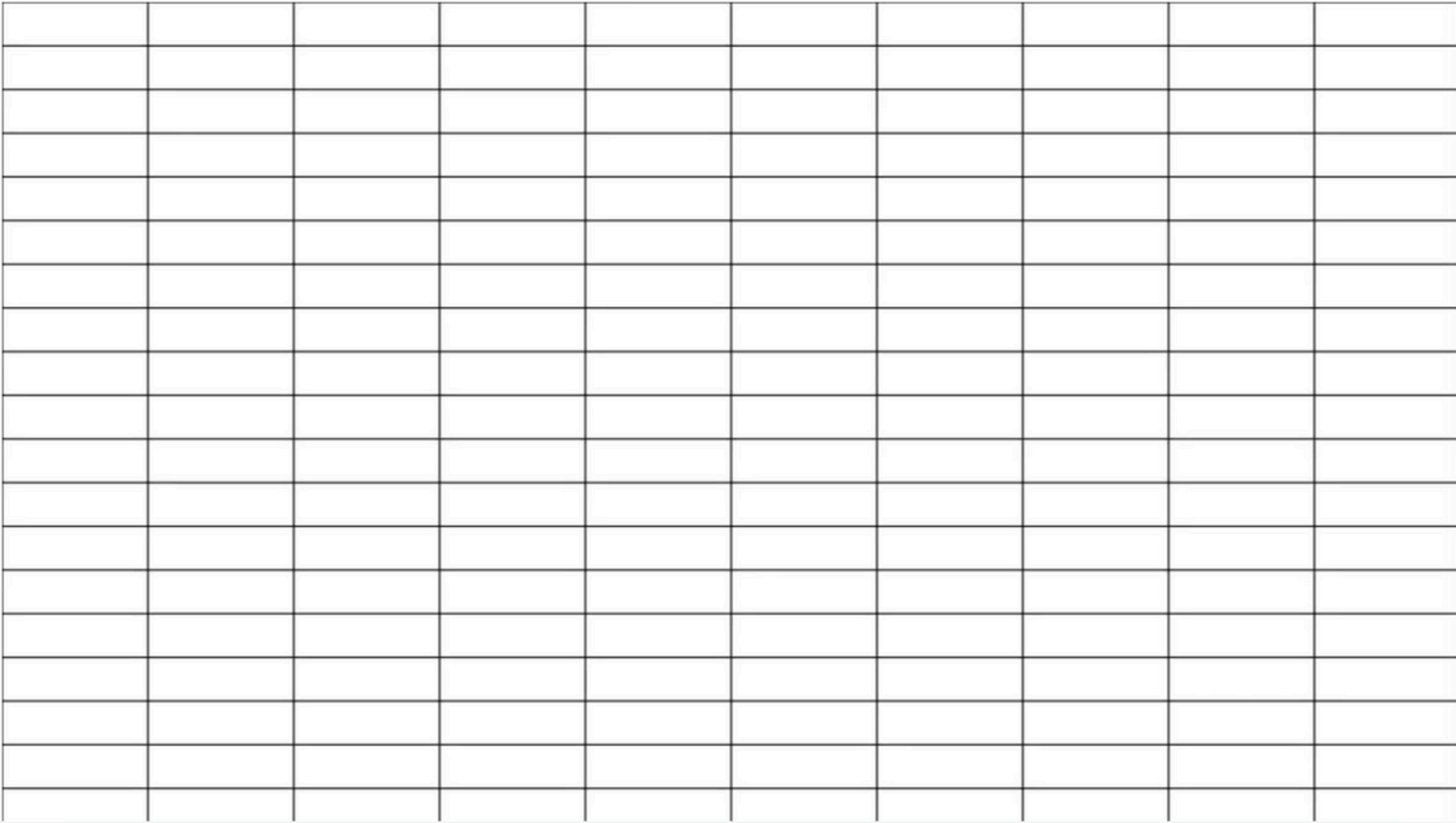
Essential Hazards

Occurs in Sequential circuits

$$Q) F(A, B, C) = \Sigma(3, 4, 5, 7)$$



$$Q) F(A,B,C) = \prod M(0,1,2,6)$$



- Static hazards can be eliminated by adding the redundant term

Multiplexer (MUX)

- Data selector
- Many to one
- Universal logic gate
- Parallel to serial converter

The general structure of a Mux

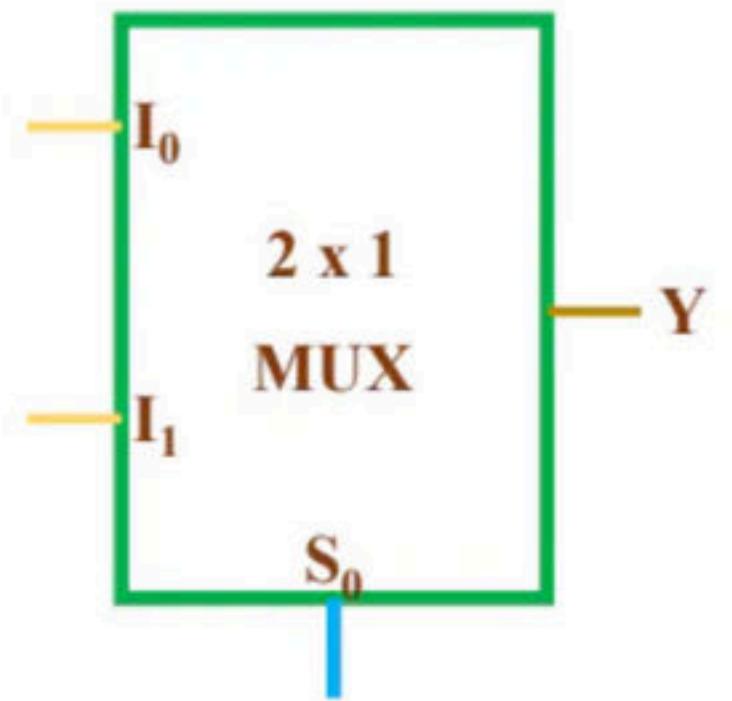
$2^n \times 1$

2^n -----> number of data inputs

n -----> number of select inputs

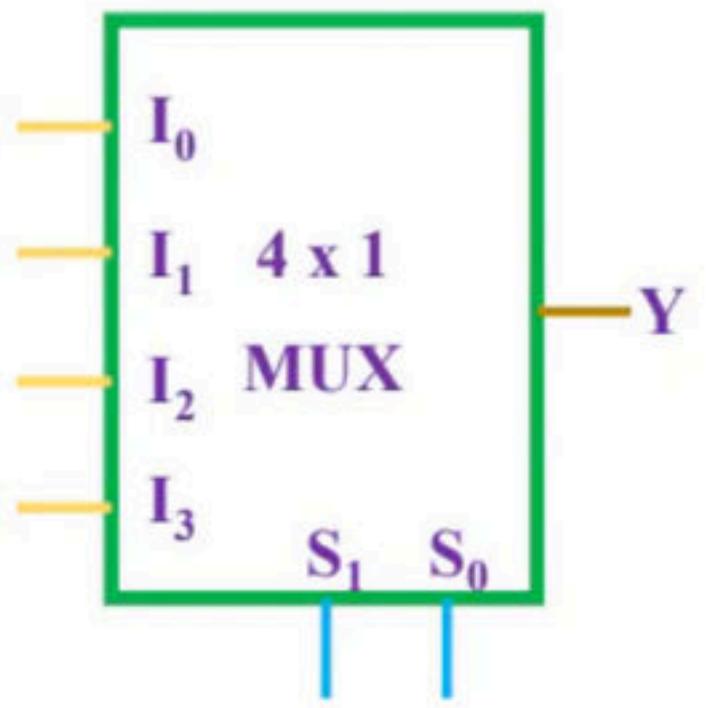
1 -----> number of outputs

2×1 MUX



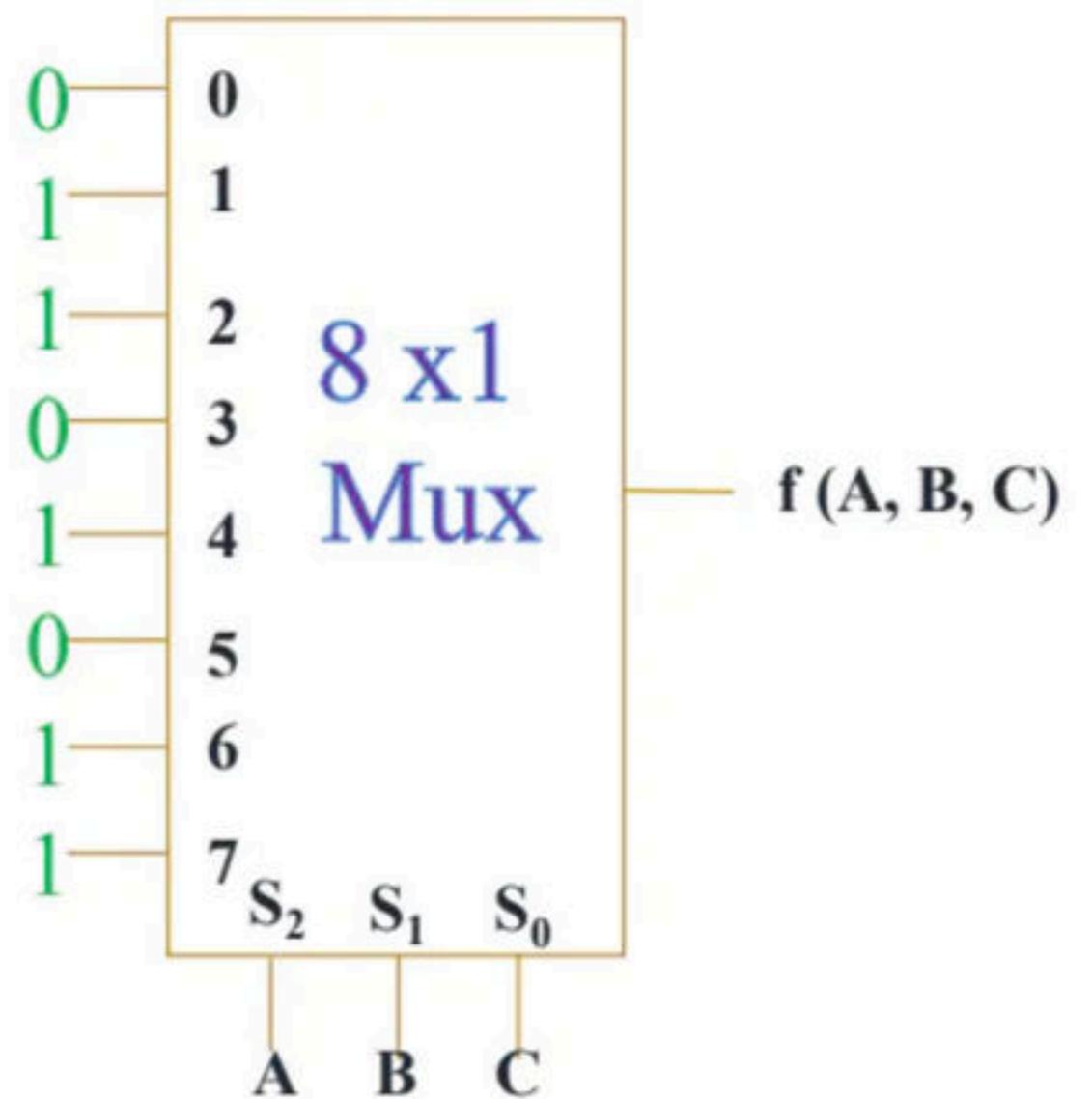
Logic circuit

4×1 MUX



S_1	S_0	Y

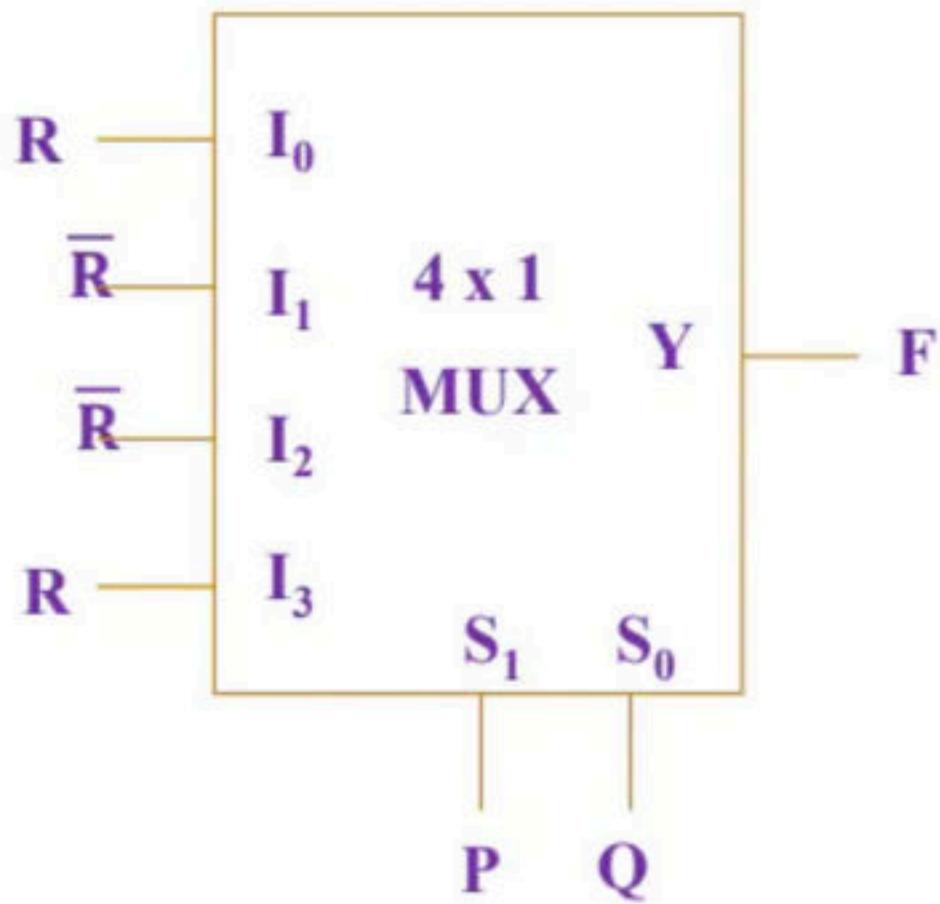
Q) Find the minterms



Q. The output F of the multiplexer circuit shown below expressed in terms of the inputs P, Q and R is

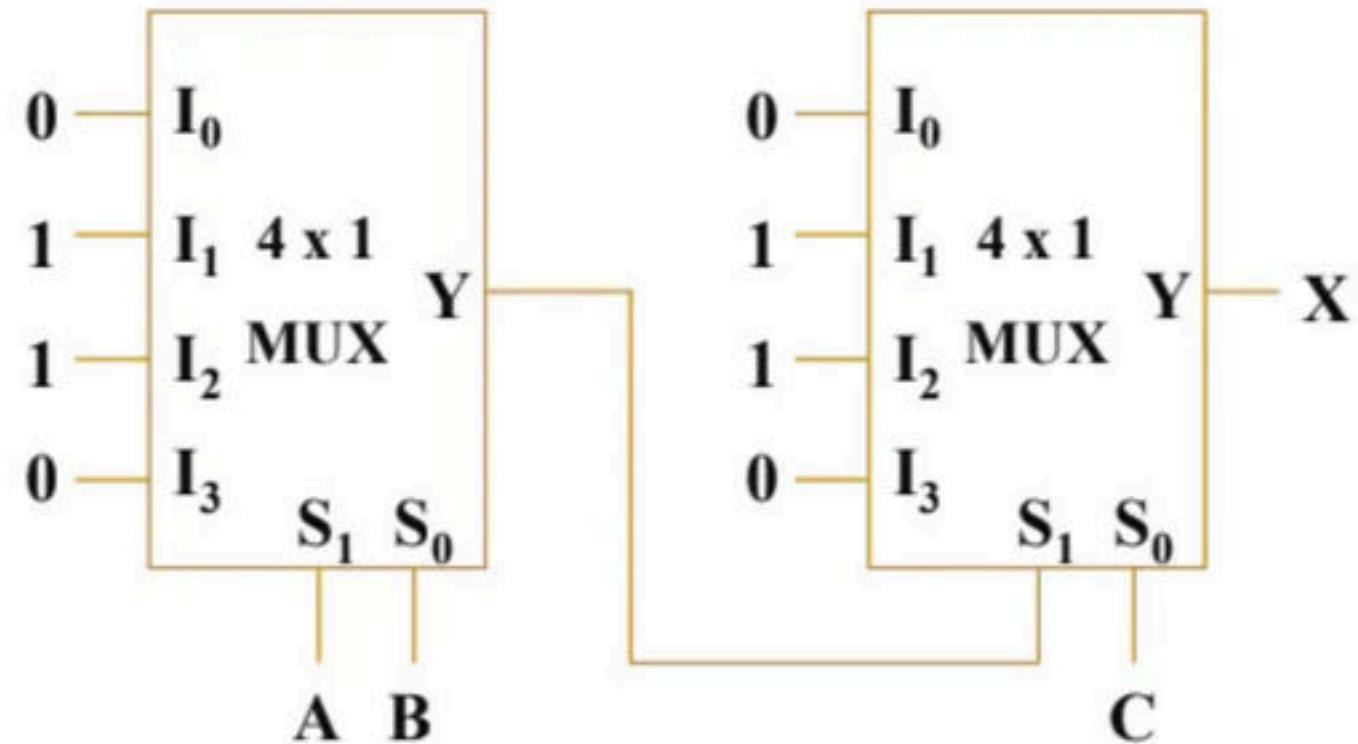
- (a) $F = P \oplus Q \oplus R$
- (c) $F = (P \oplus Q)R$

- (b) $F = PQ + QR + RP$
- (d) $F = (P \oplus Q)\bar{R}$



Q. in the following circuit, X is given by

- (a) $X = A\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}\bar{B}C + ABC$
- (b) $X = \bar{A}BC + A\bar{B}C + AB\bar{C} + \bar{A}\bar{B}\bar{C}$
- (c) $X = AB + BC + AC$
- (d) $X = \bar{A}\bar{B} + \bar{B}\bar{C} + \bar{A}\bar{C}$



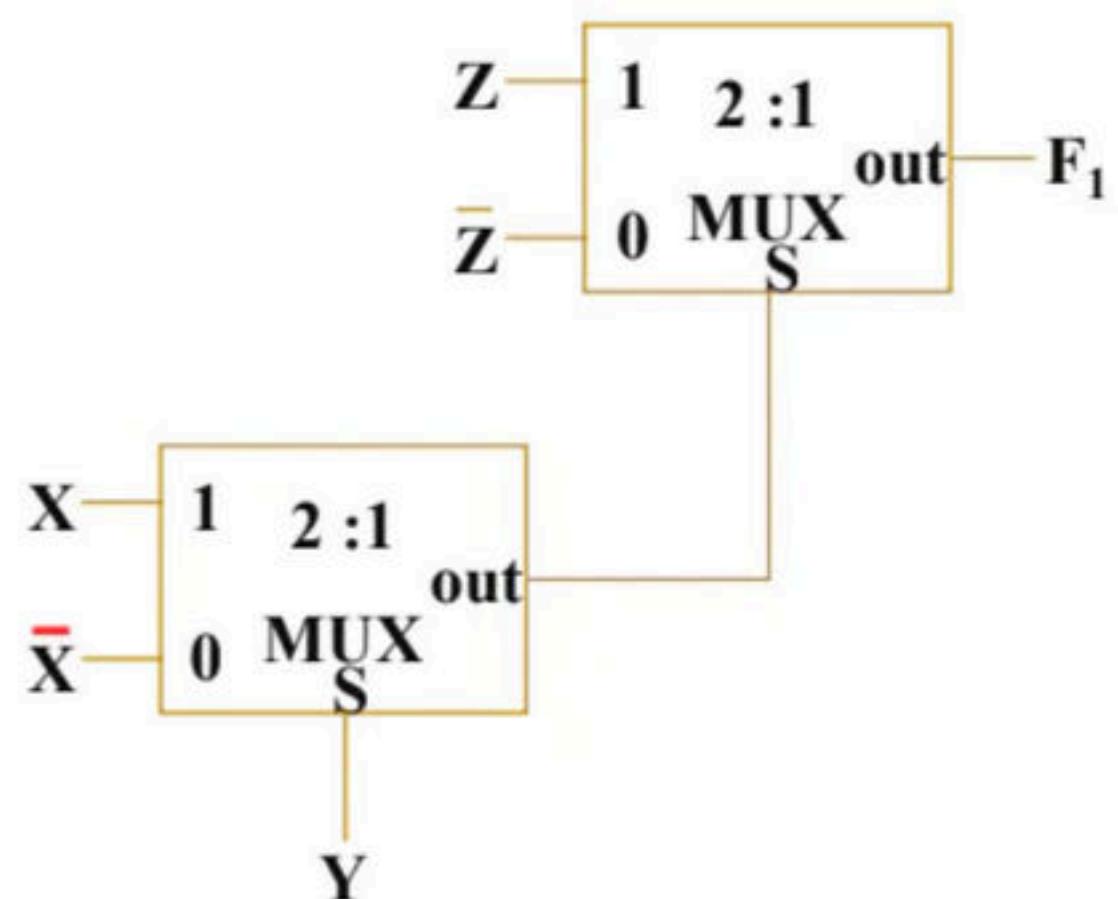
Q. A MUX circuit shown in the figure below implements a logic function F_1 . The correct expression for F_1 is.

(a) $(\overline{X} \oplus \overline{Y}) \oplus Z$

(c) $(X \oplus Y) \oplus \bar{Z}$

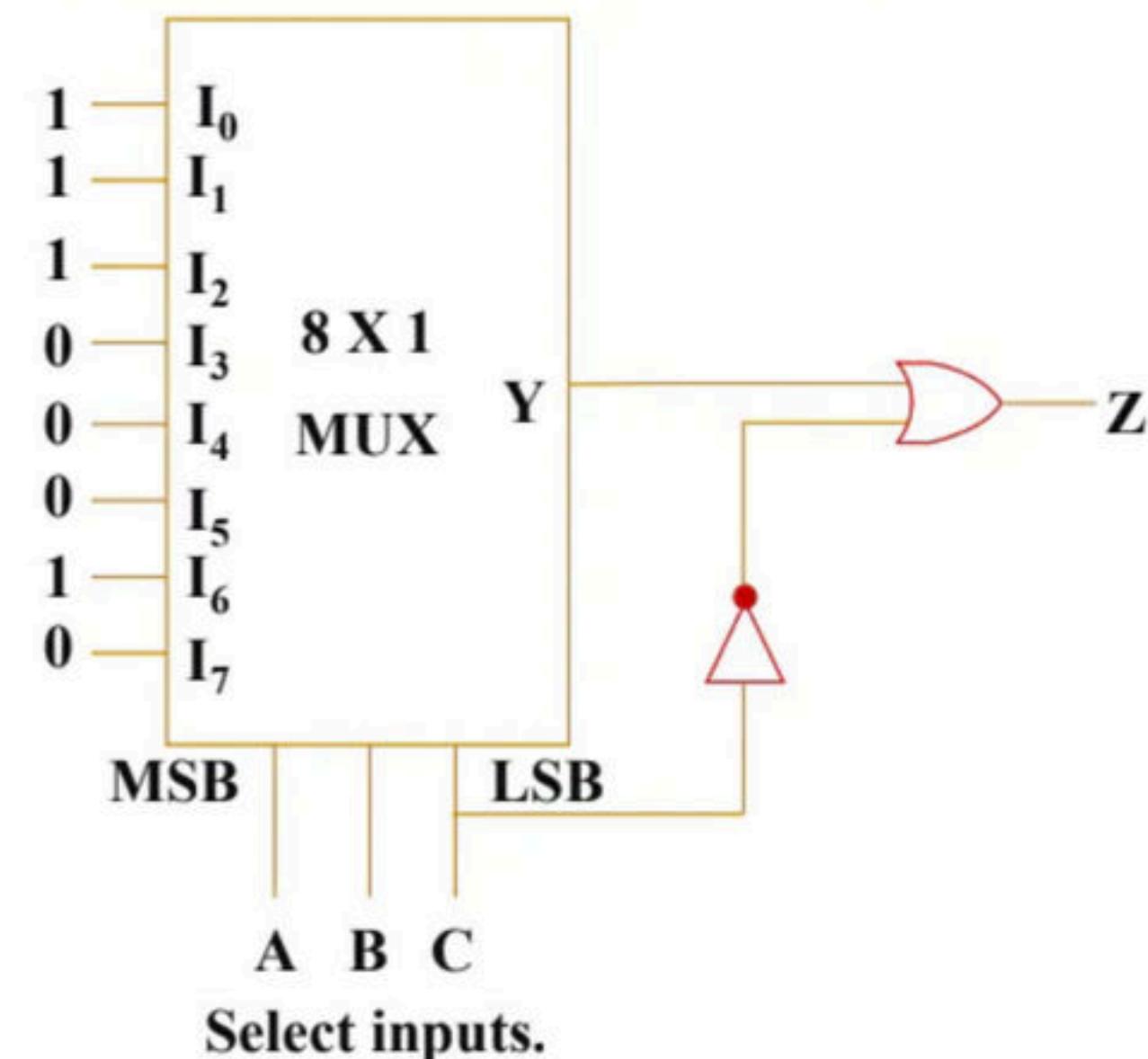
(b) $\overline{(\overline{X} \oplus Y) \oplus Z}$

(d) $(X \oplus Y) \oplus Z$



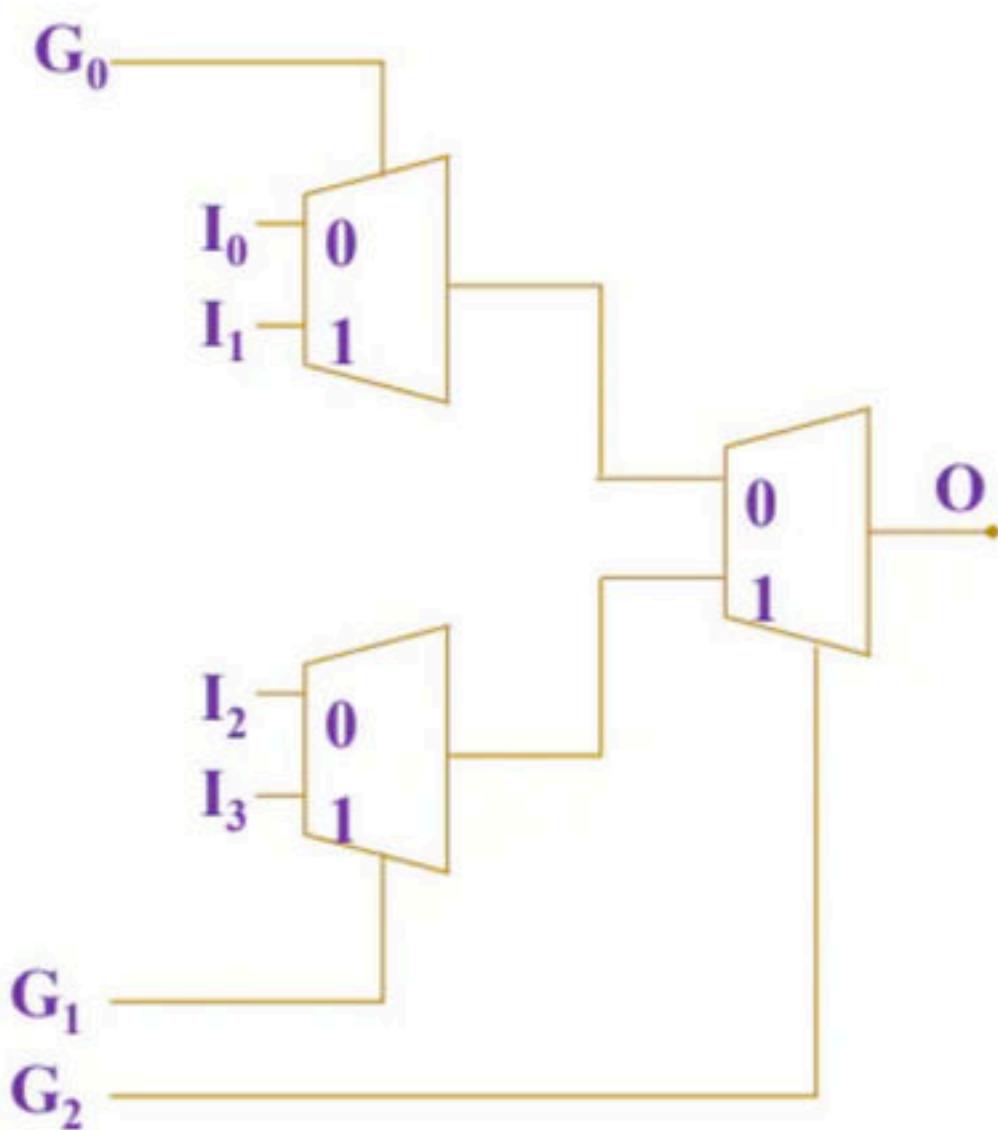
Q. A combinational circuit using an 8×1 multiplexer as shown in the figure. The minimized expression for the output (Z) is

- (a) $C(\bar{A} + \bar{B})$
- (b) $C(A + B)$
- (c) $\bar{C} + \bar{A}\bar{B}$
- (d) $\bar{C} + AB$



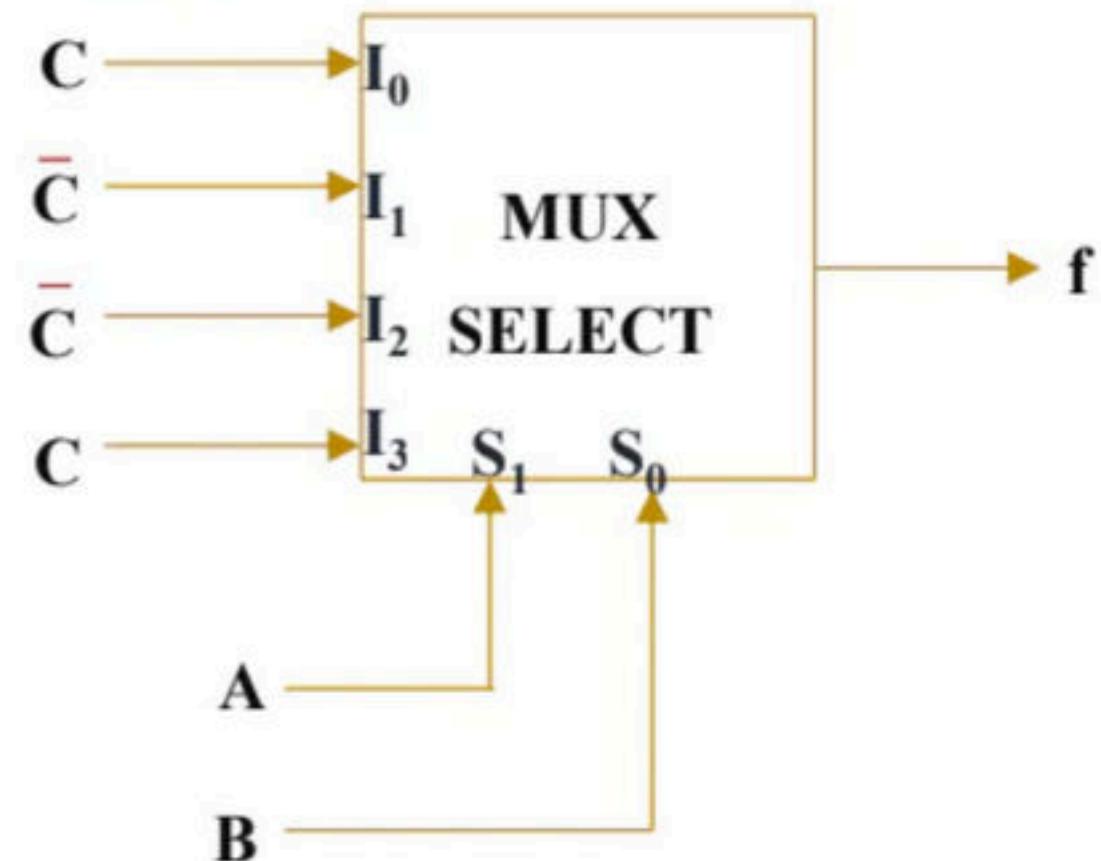
Q. The cell of a Field Programmable Gate Array is shown in the figure. It has three 2 to 1 multiplexers with their select lines G_0 , G_1 , G_2 and 4 digital signal input lines I_0 , I_1 , I_2 and I_3 . The logical function that relates the output O to the select and signal input lines is.

- (a) $\bar{G}_0\bar{G}_1I_2 + \bar{G}_0\bar{G}_1I_3 + \bar{G}_2\bar{G}_1I_0 + \bar{G}_2\bar{G}_1I_1$
- (b) $\bar{G}_0I_2 + \bar{G}_0G_1 + \bar{G}_2I_0 + \bar{G}_2\bar{G}_1I_1 + G_0$
- (c) $\bar{G}_0\bar{G}_2I_0 + G_0\bar{G}_2I_1 + G_2\bar{G}_1I_2 + G_2G_1I_3$
- (d) $G_2G_1\bar{I}_2 + \bar{G}_2\bar{G}_1\bar{I}_3 + G_2\bar{G}_0I_0 + G_0\bar{G}_2I_1$



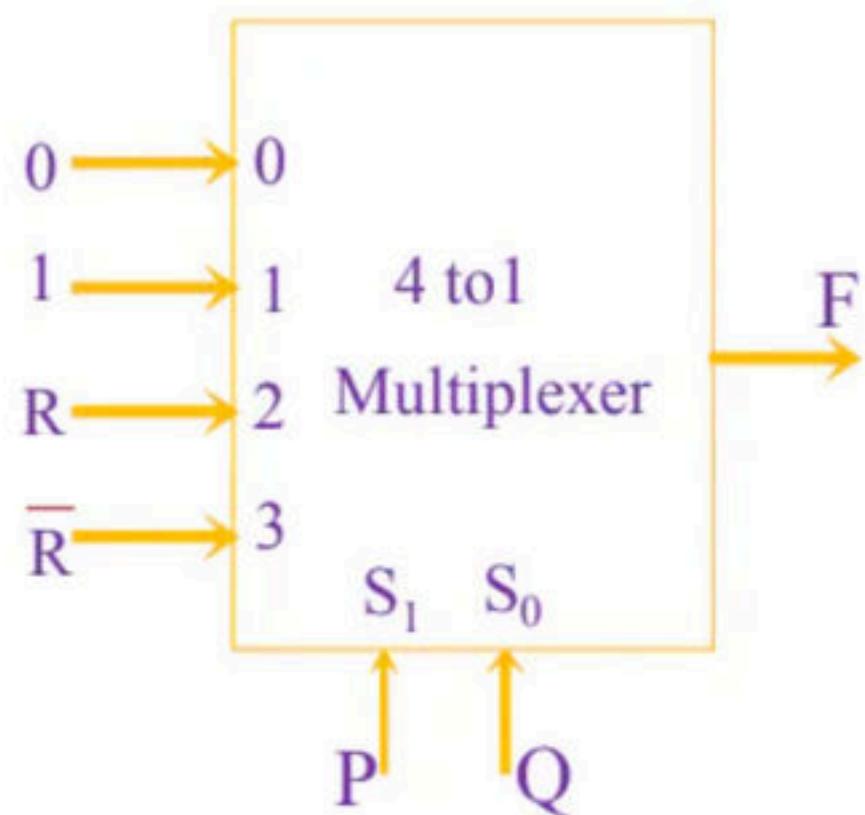
Q. The output 'F' of the multiplexer circuit shown in the figure will be

- (a) $AB + B\bar{C} + \bar{C}A + \bar{B}\bar{C}$
- (b) $A \oplus B \oplus C$
- (c) $A \oplus B$
- (d) $\overline{ABC} + \overline{ABC} + AB\bar{C} + ABC$



Q. Consider the 4-to-1 multiplexer with two lines S_1 and S_0 given below. The minimal sum of products form of the Boolean expression for the output F of the Multiplexer is

- (A) $\bar{P}Q + Q\bar{R} + P\bar{Q}R$
- (B) $\bar{P}Q + \bar{P}Q\bar{R} + PQ\bar{R} + P\bar{Q}R$
- (C) $\bar{P}QR + \bar{P}Q\bar{R} + Q\bar{R} + P\bar{Q}R$
- (D) $PQ\bar{R}$



Q. Consider the following combinational function block involving four Boolean variables x, y, a, b where x, a, b are inputs and y is the output.

$f(x, y, a, b)$

{

if (x is 1) $y = a$;

else $y = b$;

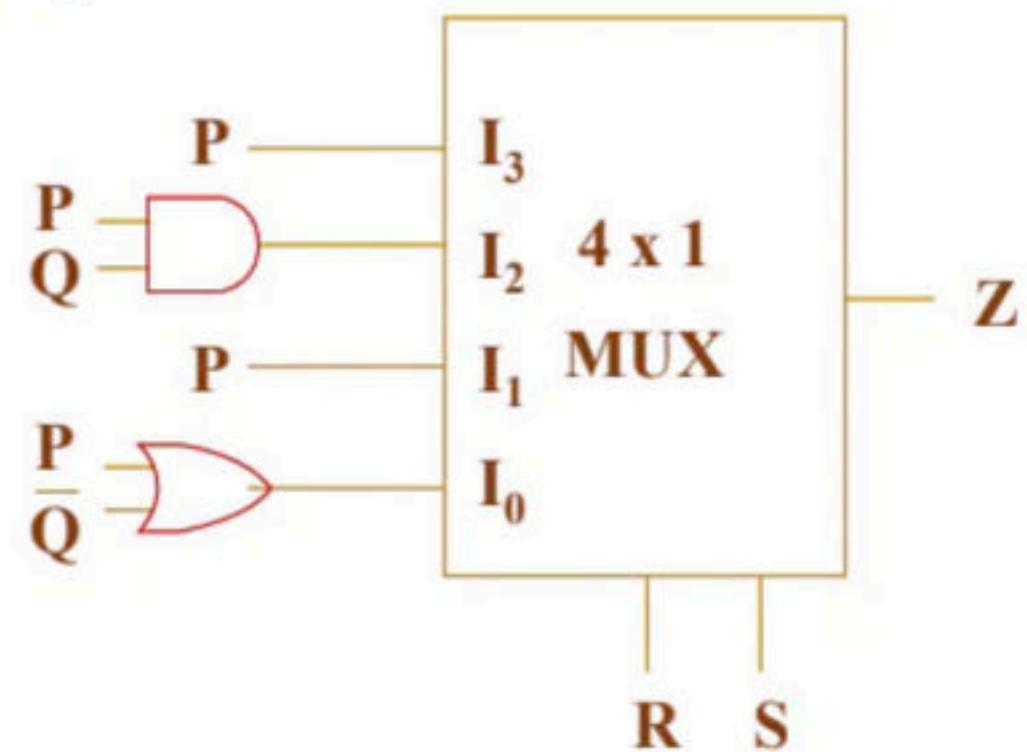
}

Which one of the following digital logic blocks is the most suitable for implementing this function?

- (A) Full adder
- (B) Priority encoder
- (C) Multiplexor
- (D) Flip-flop

Q. For the circuit shown in the following figure $I_0 - I_3$ are inputs to the 4:1 multiplexer R(MSB) and S are control bits. The output Z can be represented by

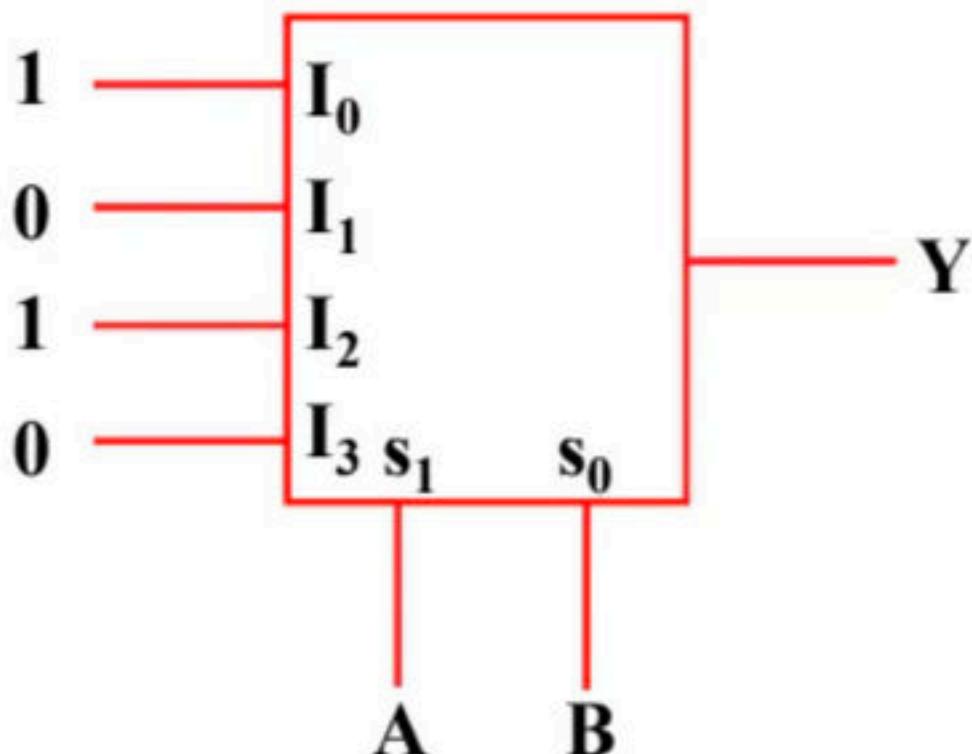
- (a) $PQ + P\bar{Q}S + \bar{Q}\bar{R}\bar{S}$
- (b) $P\bar{Q} + PQ\bar{R} + \bar{P}\bar{Q}\bar{S}$
- (c) $P\bar{Q}\bar{R} + \bar{P}QR + PQRS + \bar{Q}\bar{R}\bar{S}$
- (d) $PQR + PQRS + P\bar{Q}\bar{R}S + \bar{Q}\bar{R}\bar{S}$



Q. The logical expressions of the output of a 4×1 multiplexer shown below is

- (a) $A + \bar{B}$
- (c) \bar{A}

- (b) \bar{B}
- (d) B



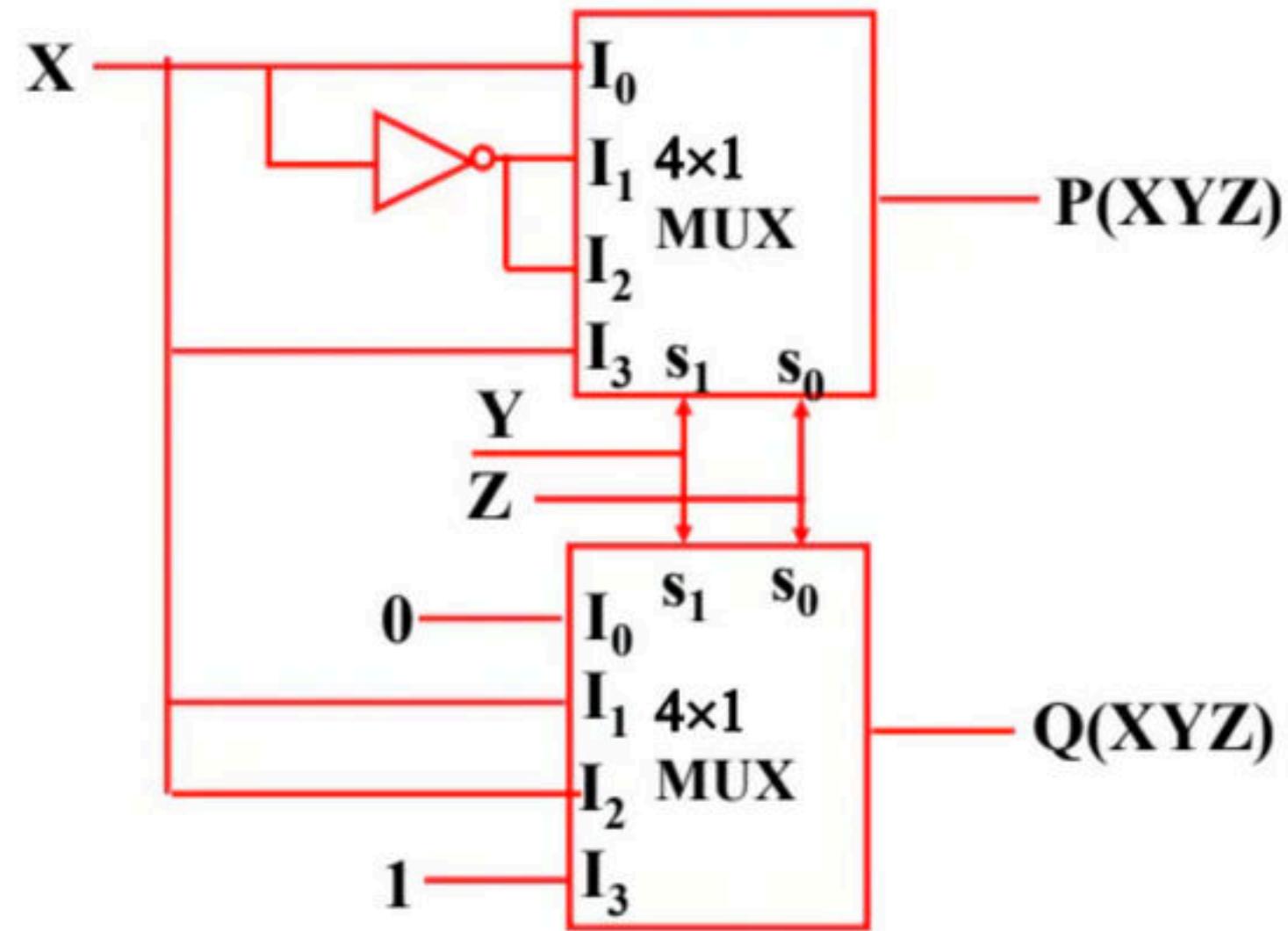
Q. The multiplexer circuit function as

(a) Full subtractor

(c) Two output comparator

(b) Full adder

(d) Half adder



Q. If the logic expression of the outputs in the circuit shown in figure A and B are same, then select the correct combination of signals to be connected to the inputs of multiplexer

- | | | | | |
|-----|-------|-----------|-----------|-------|
| | I_0 | I_1 | I_2 | I_3 |
| (a) | C | 0 | \bar{C} | 1 |
| (b) | C | C | \bar{C} | C |
| (c) | C | \bar{C} | \bar{C} | C |
| (d) | 1 | C | \bar{C} | 1 |

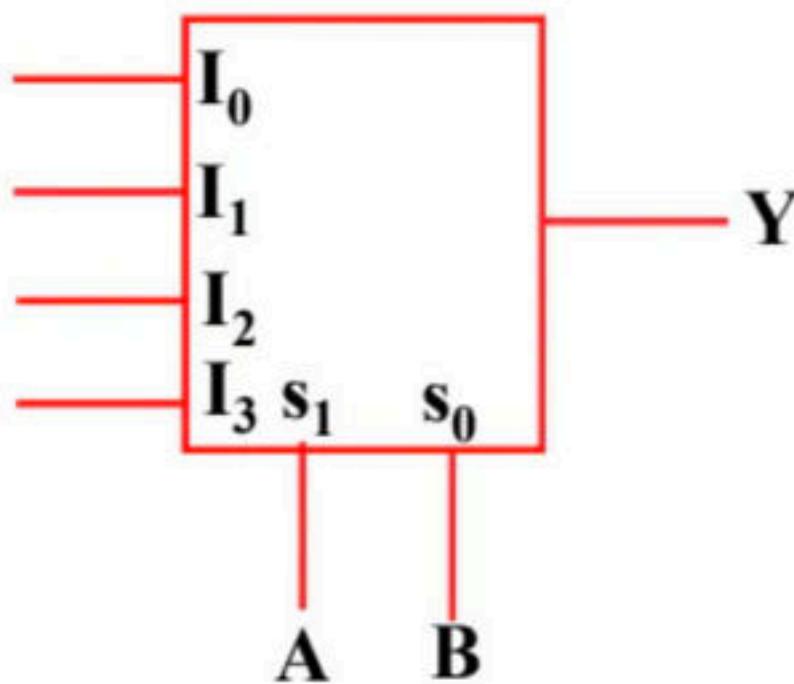


Figure A

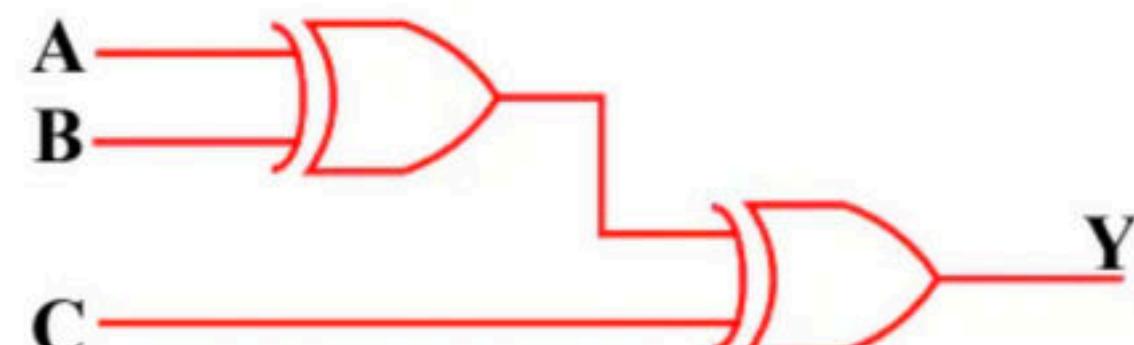


Figure B

Q. A combinational circuit using 4×1 mux is shown in figure

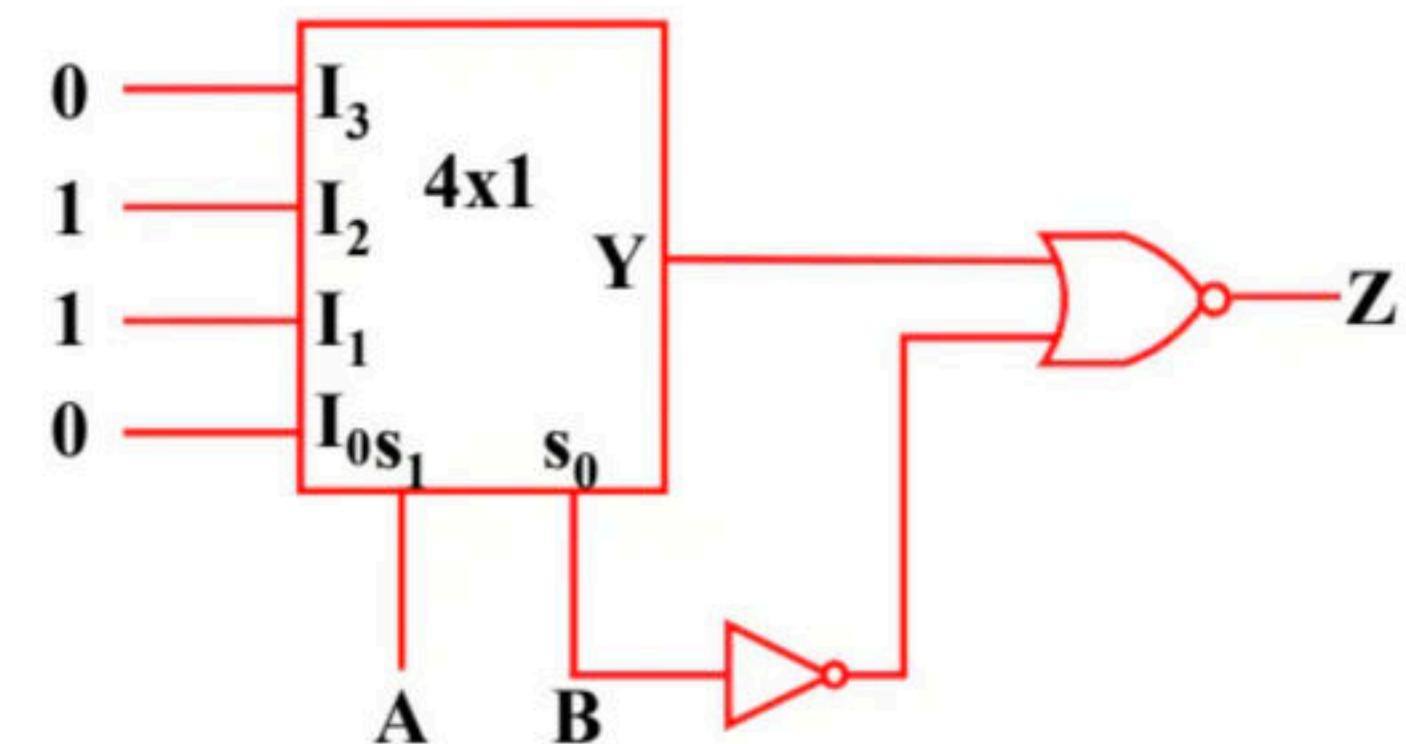
The output Z is

(a) $A + B$

(b) $\overline{A} \oplus B$

(c) AB

(d) $(\overline{A} + \overline{B})$



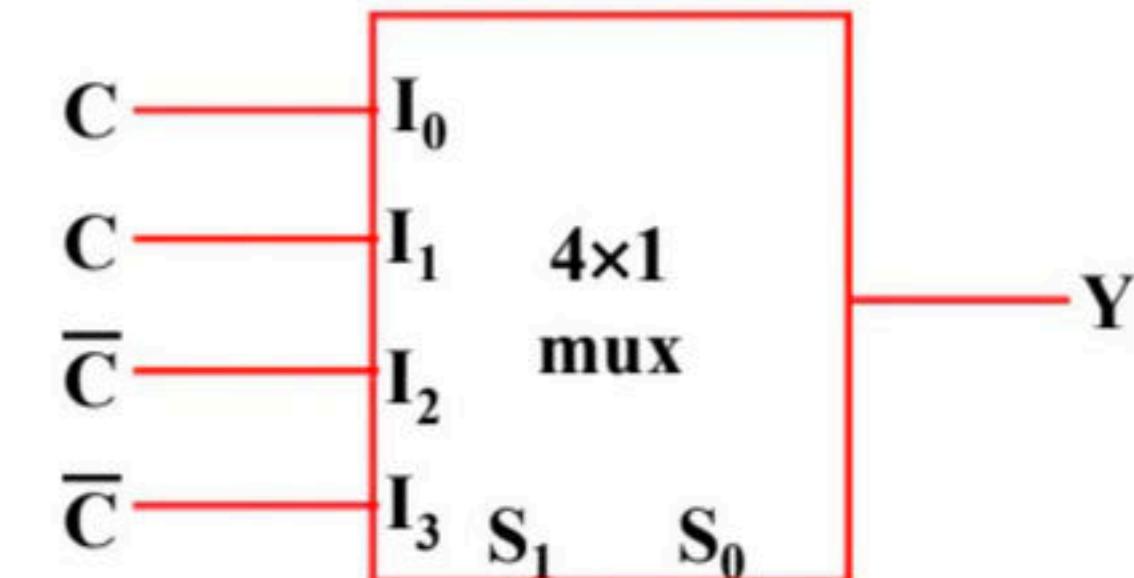
Q. The expression for Y is

(a) $A \oplus B \oplus C$

(b) $(A \oplus B)C + AB\bar{C}$

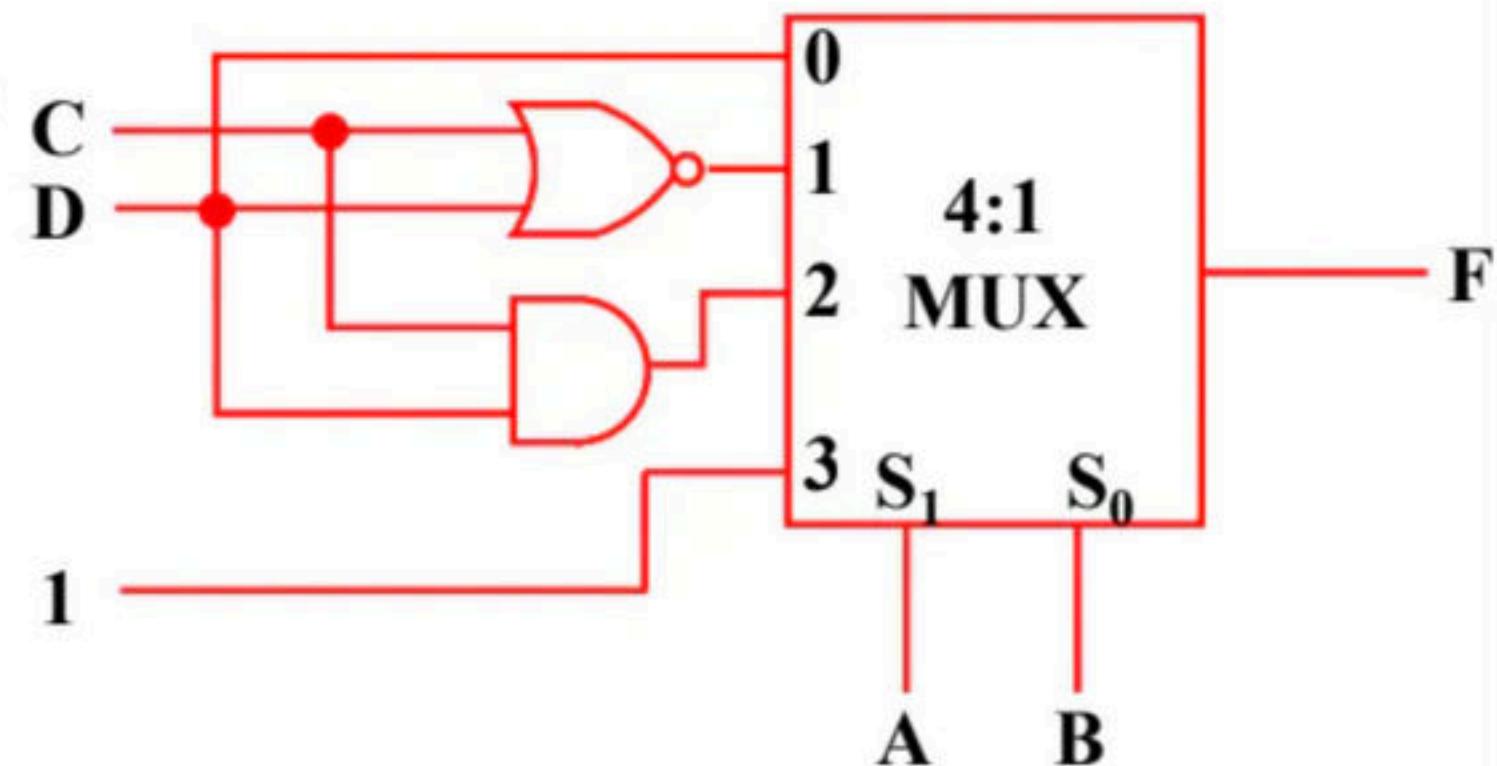
(c) $AB + A \oplus B$

(d) $ABC + (A \oplus B)\bar{C}$



Q. The Boolean function realized by the following circuit is

- (a) $F(A, B, C, D) = \sum(1, 3, 4, 11, 12, 13, 14, 15)$
- (b) $F(A, B, C, D) = \sum(0, 2, 4, 5, 9, 10, 11)$
- (c) $F(A, B, C, D) = \sum(1, 8, 14, 15)$
- (d) $F(A, B, C, D) = \sum(0, 2, 6, 8, 14, 15)$



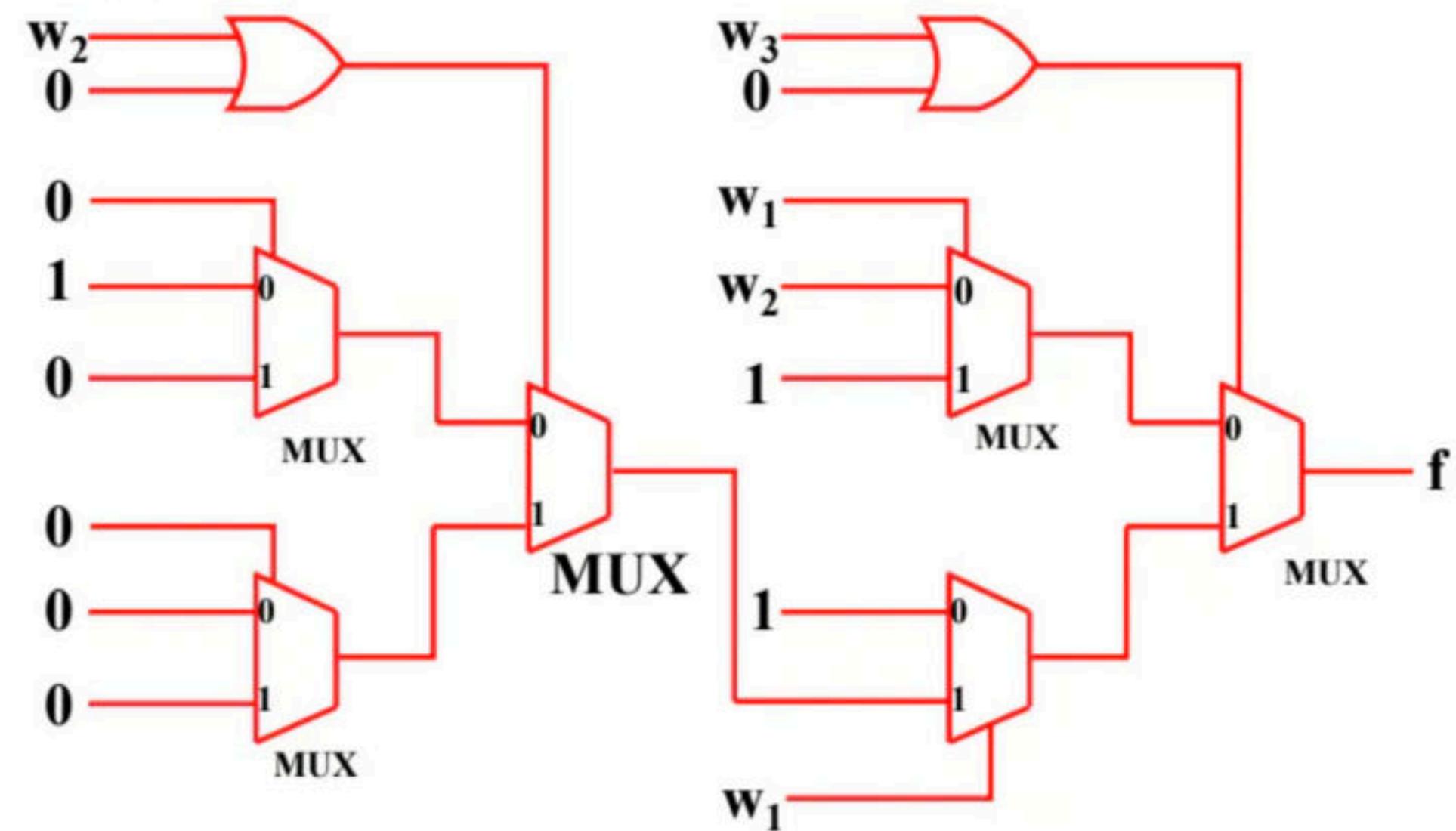
Q. The logic circuit shown below implements

(a) $(w_1 \oplus w_2) + (w_2 \oplus w_3)$

(b) $(w_2 \oplus w_1) + (w_1 \oplus w_3)$

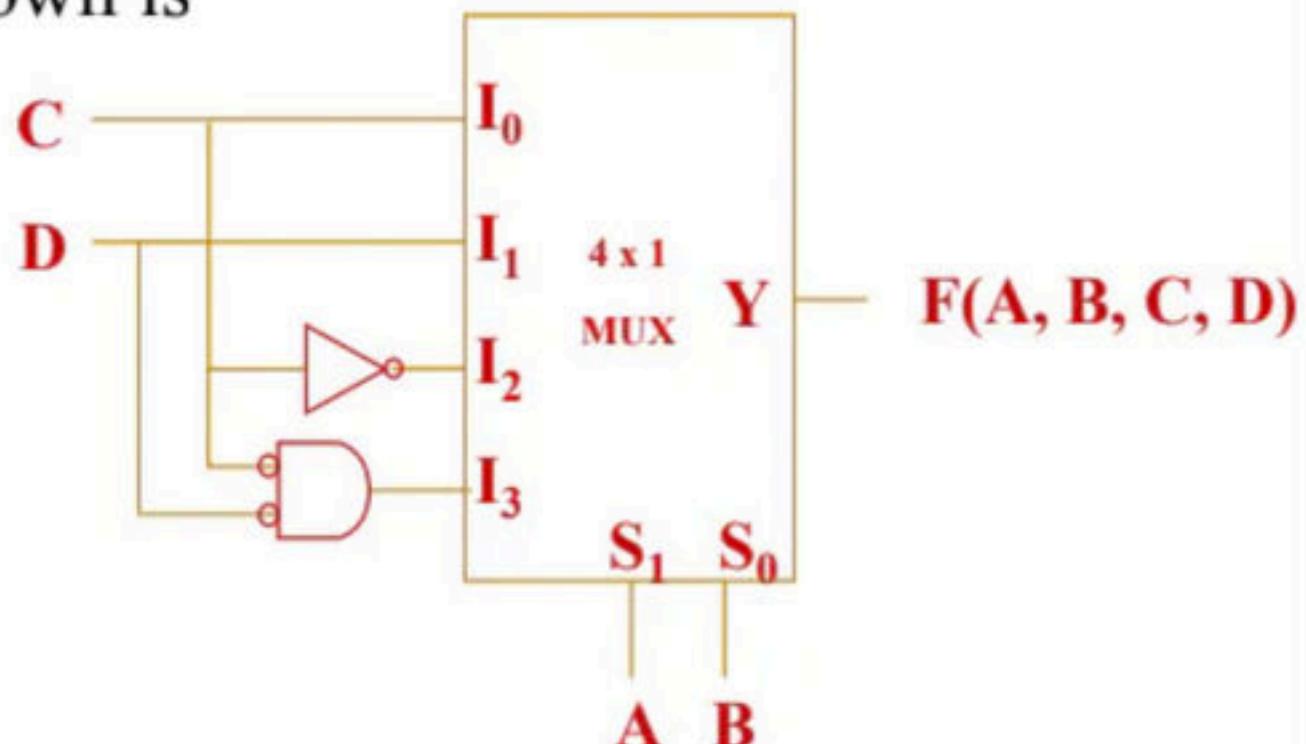
(c) $(w_2 \oplus w_3) + (w_1 \oplus w_3)$

(d) $(w_1 \oplus w_2 \oplus w_3)$



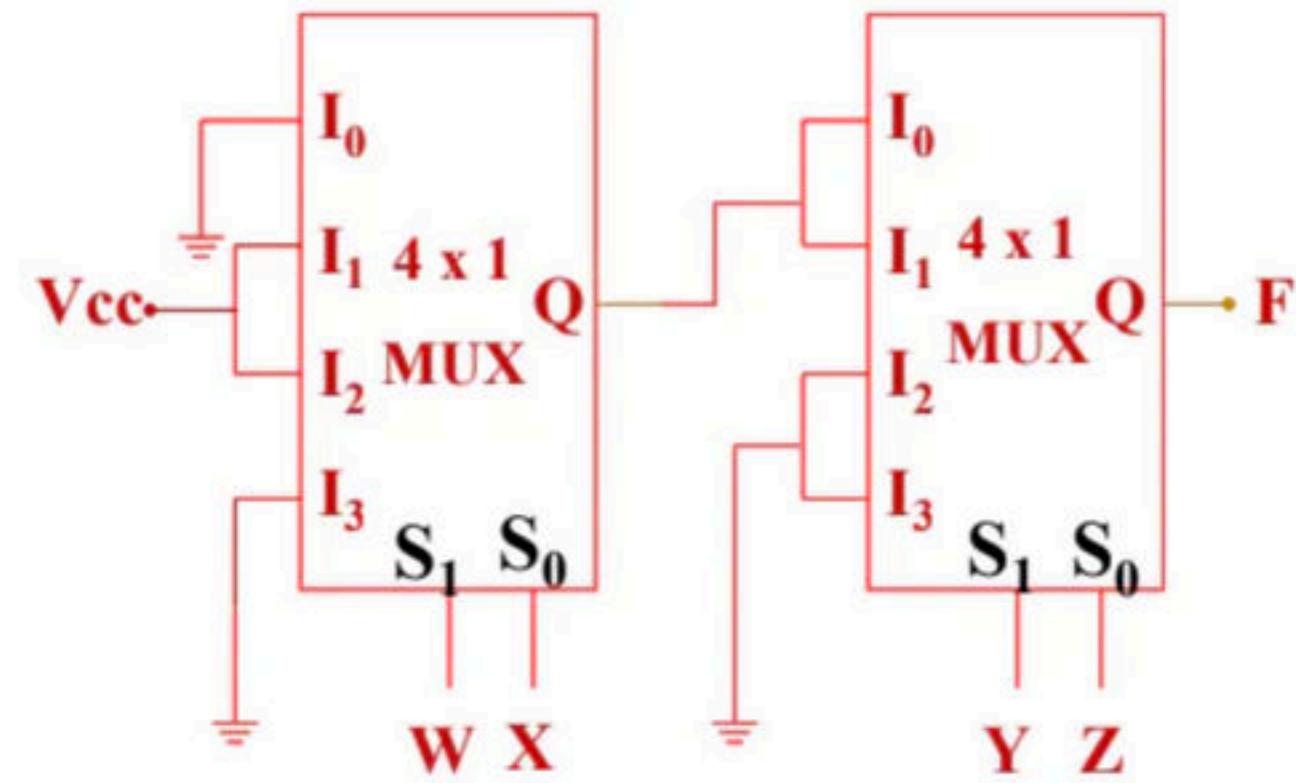
Q. The Boolean function realized by the logic circuit shown is

- (a) $F = \sum m(0,1,3,5,9,10,14)$
- (b) $F = \sum m(2,3,5,7,8,12,13)$
- (c) $F = \sum m(1,2,4,5,11,14,15)$
- (d) $F = \sum m(2,3,5,7,8,9,12)$

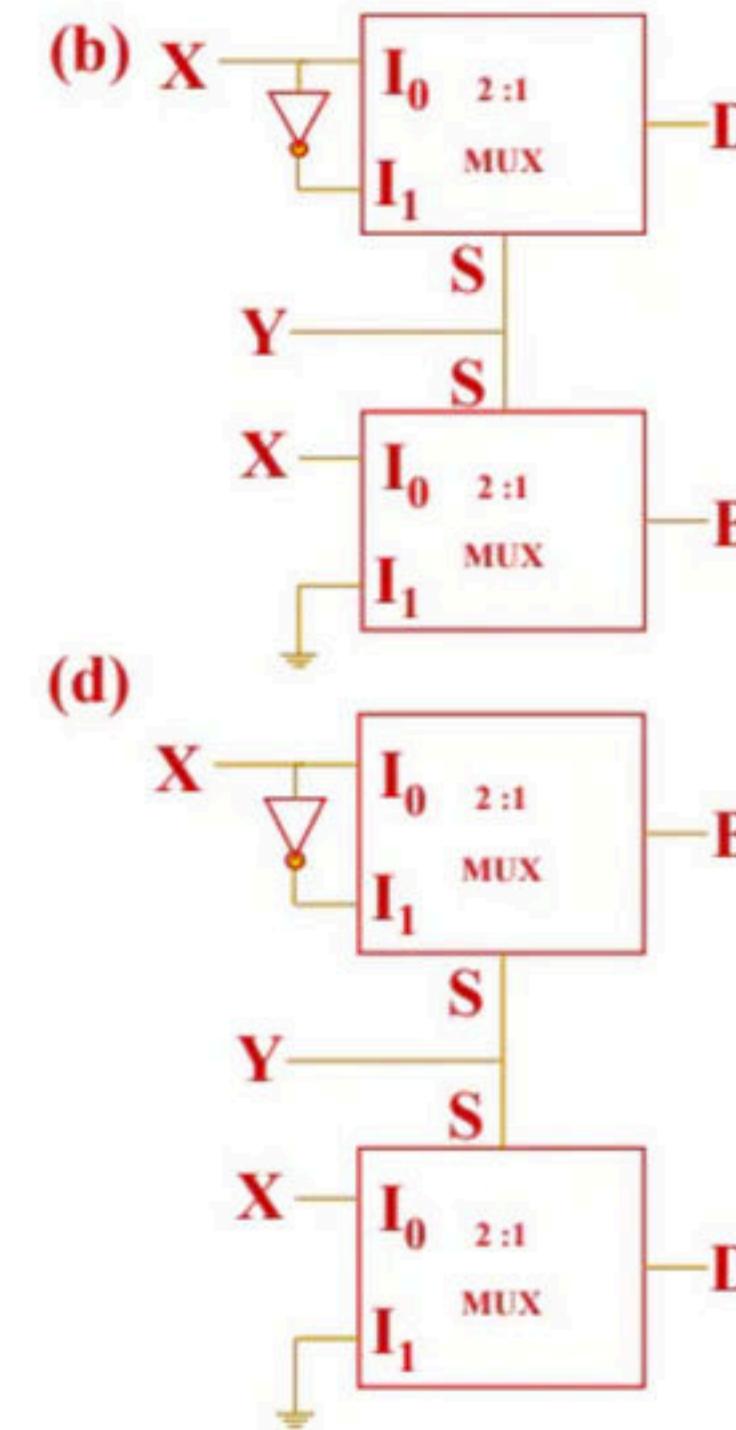
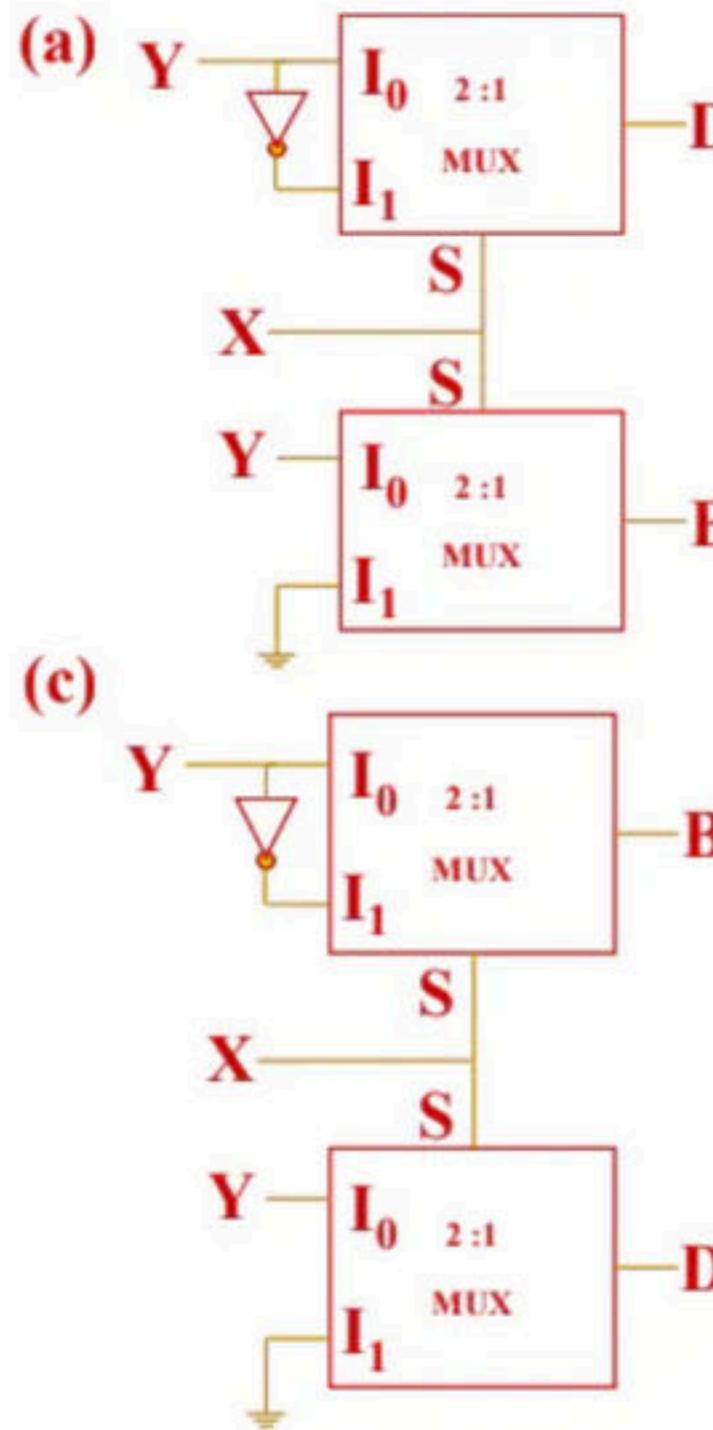


Q. In the circuit shown, W and Y are MSBs of the control inputs. The output F is given by

- (a) $F = W\bar{X} + \bar{W}X + \bar{Y}\bar{Z}$
- (b) $F = W\bar{X} + \bar{W}X + \bar{Y}Z$
- (c) $F = W\bar{X}\bar{Y} + \bar{W}XY$
- (d) $F = (\bar{W} + \bar{X}) + \bar{Y}Z$



Q. If X and Y are inputs and the Difference ($D = X - Y$) and the Borrow (B) are the outputs, which one of the following diagrams implements a half subtractor?



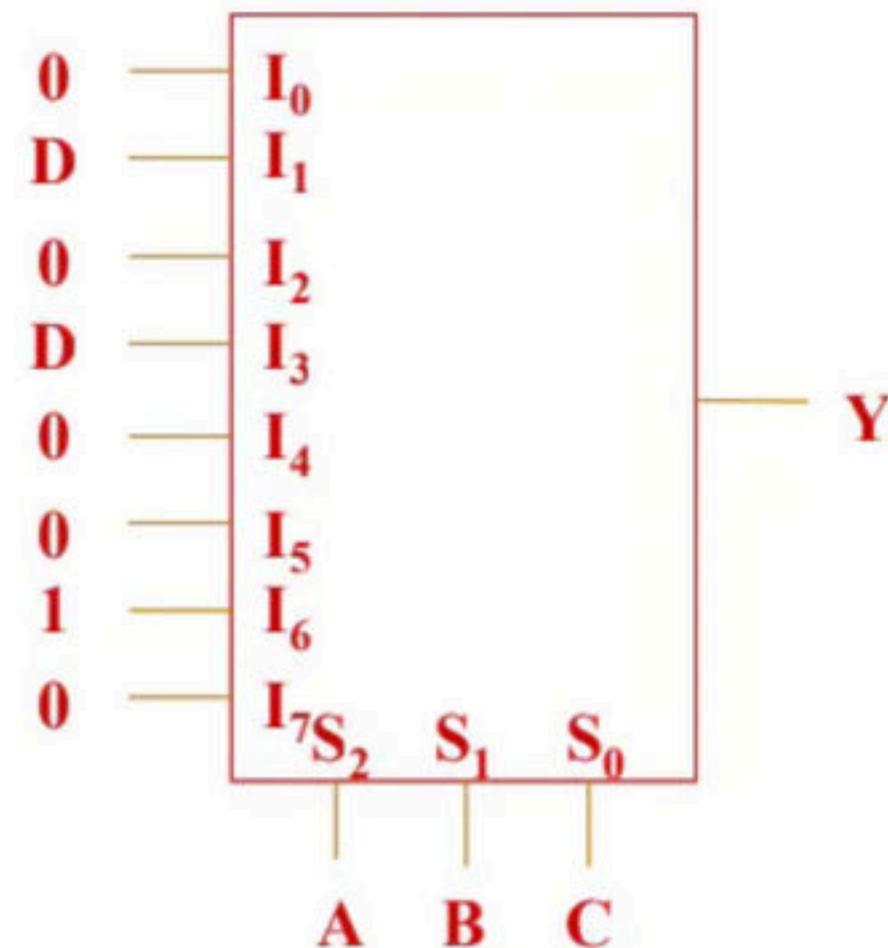
q. An 8-to-1 multiplexer is used to implement logical function Y as shown in the figure. The output Y is given by.

(a) $Y = A\bar{B}C + A\bar{C}D$

(c) $Y = AB\bar{C} + \bar{A}CD$

(b) $Y = \bar{A}BC + A\bar{B}D$

(d) $Y = \bar{A}\bar{B}D + A\bar{B}C$



Q) Design a logic circuit $f(A, B, C) = \sum m(0, 1, 3, 6, 7)$ using suitable MUX

Q) Design a logic circuit $f(A, B, C) = \sum m(0, 1, 3, 6, 7)$ using 4×1 MUX

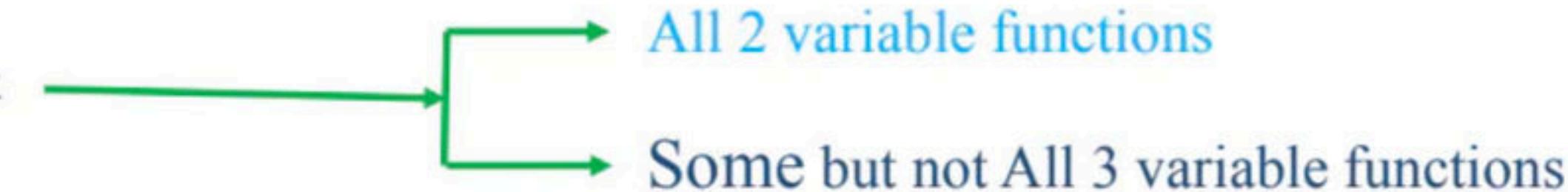
- a) AB as select lines
- b) BC as select lines
- c) AC as select lines

BC as select lines

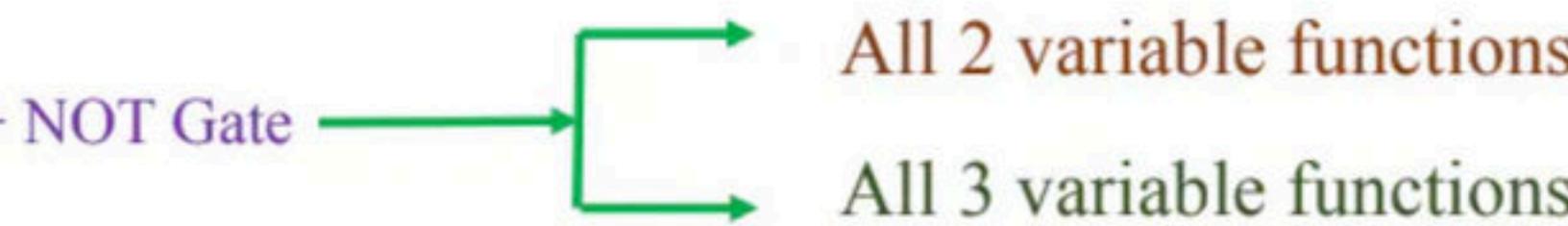
AC as select lines

Note :

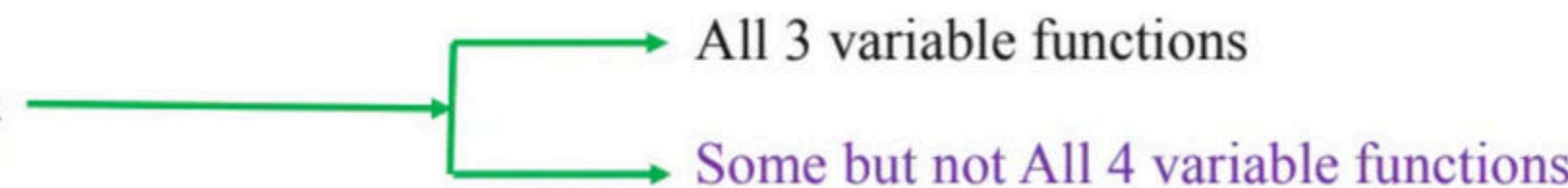
1. By using one 4×1 Mux



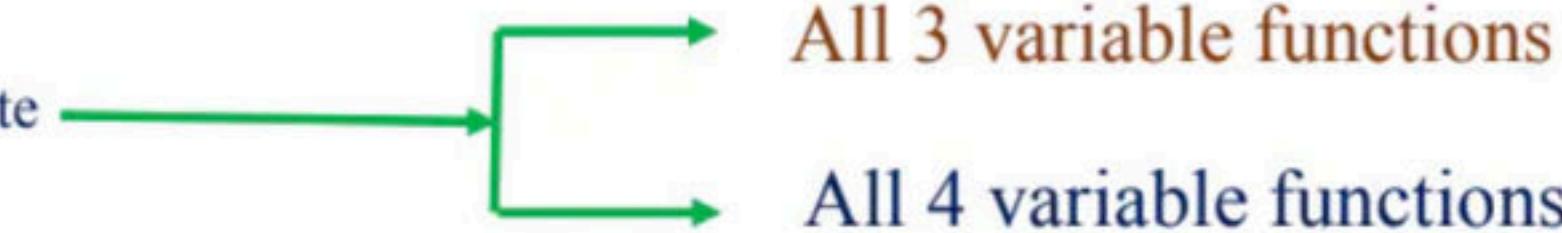
2. By using one 4×1 Mux + NOT Gate



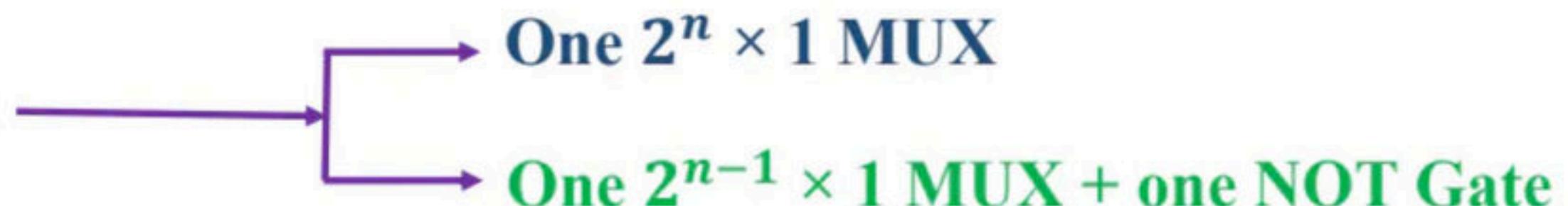
3. By using one 8×1 Mux



4. By using one 8×1 Mux + NOT Gate



5. n-variable function



Q) Suppose only one mux and one inverter are allowed to be used to implement Boolean function of n- variables , what is the minimum size of the mux needed

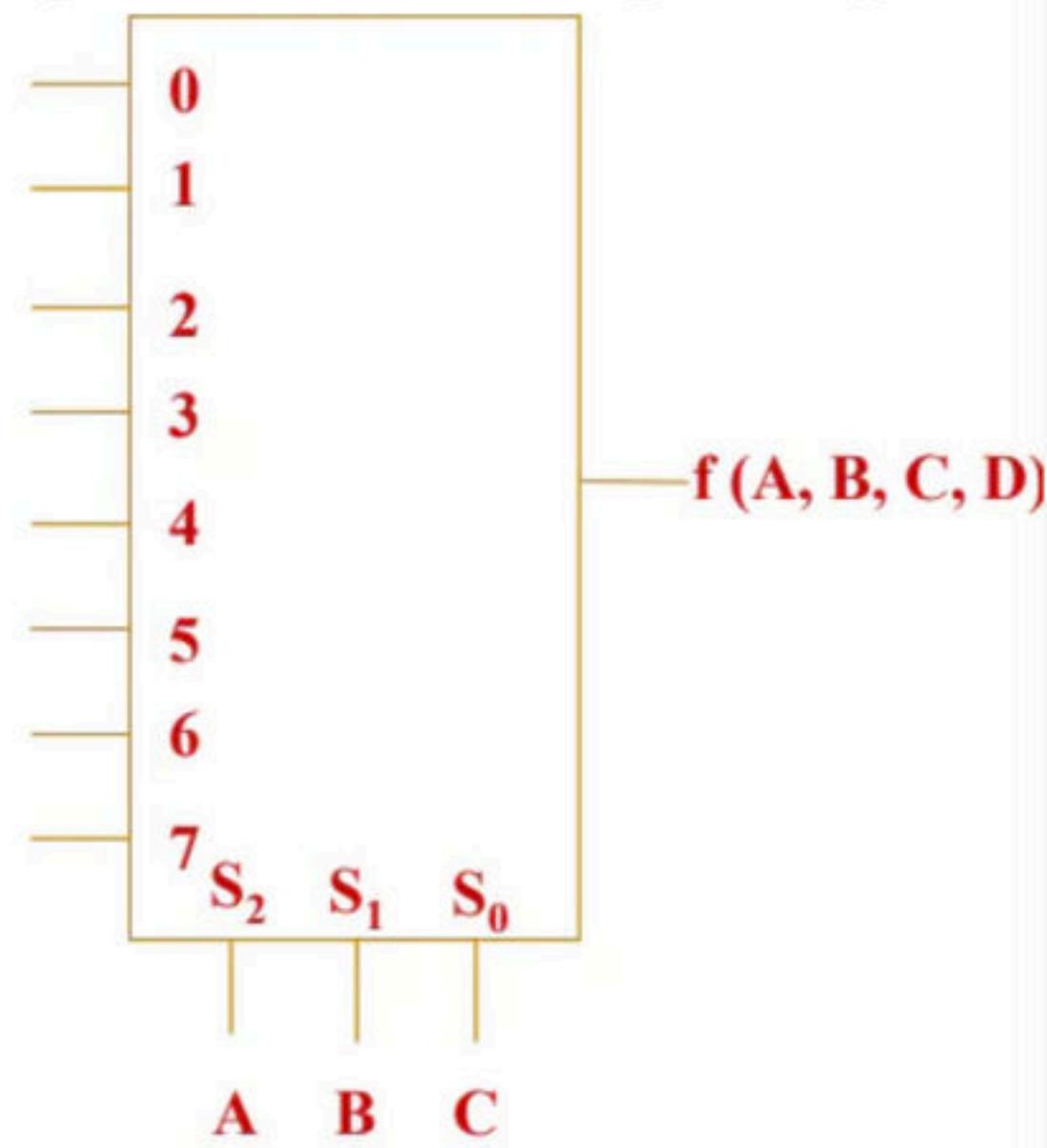
- a) $2^n \times 1$ MUX**
- b) $2^{n+1} \times 1$ MUX**
- c) $2^{n-1} \times 1$ MUX**
- d) $2^{n-2} \times 1$ MUX**

Q) Without using any additional circuitry an 8×1 mux can be used to obtain

- a) Some but not all Boolean functions of 3 variables
- b) All functions of 3 variable & none of 4- variables
- c) All function's of 4 variables
- d) All functions of 3 variables and some but not all functions of 4 variables

Q. A Boolean function $f(A, B, C, D) = \pi(1, 5, 12, 15)$ is to be implemented using an 8×1 multiplexer (A is MSB). The inputs ABC are connected to the select inputs $S_2 S_1 S_0$ of the multiplexer, respectively. Which one of the following options gives the correct inputs to pins 0,1,2,3,4,5,6,7 in order?

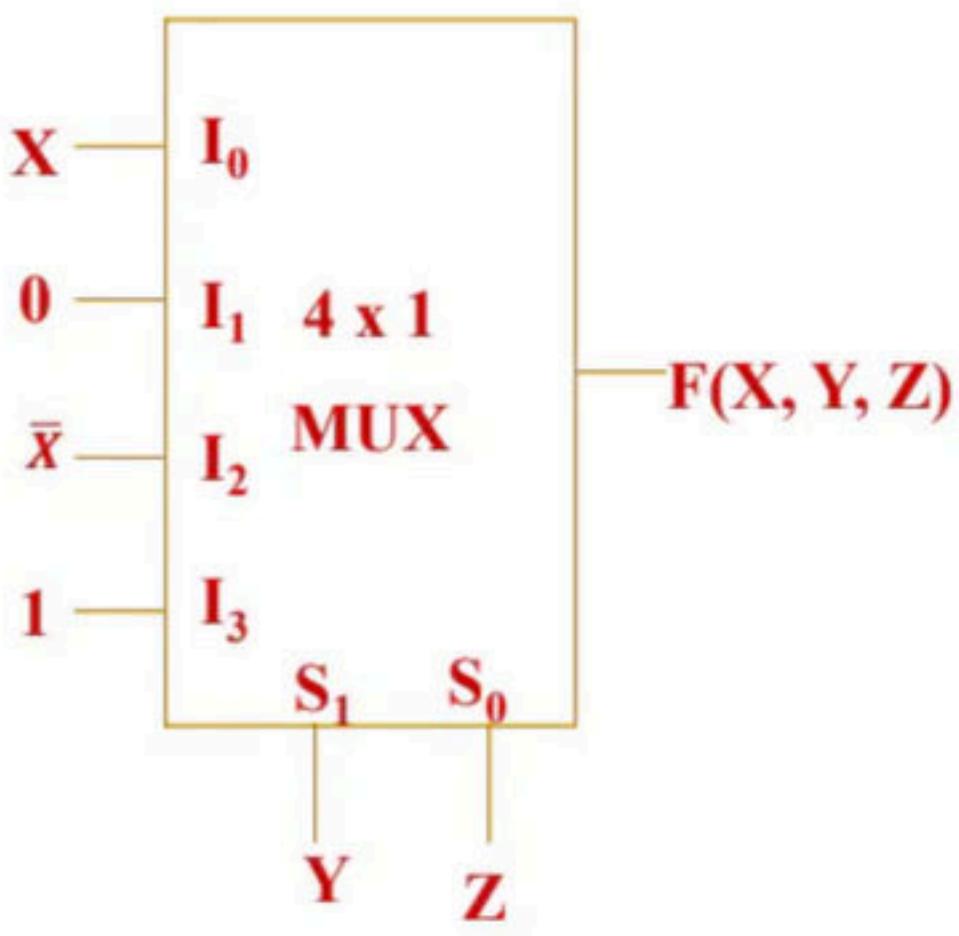
- (a) D, 0, D, 0, 0, 0, \bar{D} , D
- (b) \bar{D} , 1, \bar{D} , 1, 1, 1, D, \bar{D}
- (c) D, 1, D, 1, 1, 1, \bar{D} , D
- (d) \bar{D} , 0, \bar{D} , 0, 0, 0, D, \bar{D}



Q. A 4 to 1 multiplexer to realize a Boolean function $F(X, Y, Z)$ is shown in the figure below. The inputs Y and Z are connected to the selectors of the MUX (Y is more significant). The canonical sum-of-product expression for $F(X, Y, Z)$ is

- (a) $\sum m(2,3,4,7)$
- (c) $\sum m(0,2,4,6)$

- (b) $\sum m(1,3,5,7)$
- (d) $\sum m(2,3,5,6)$

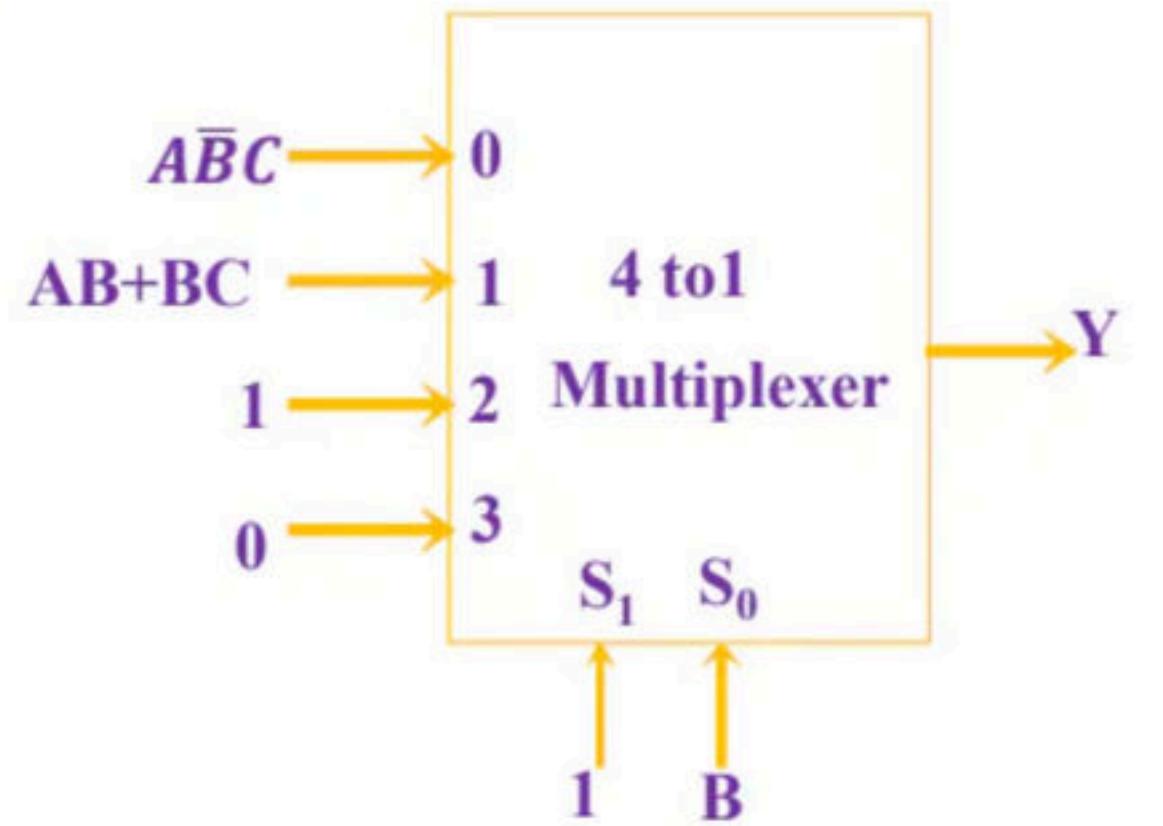


Q) Design a logic circuit $F(A, B, C) = \sum m(0, 3, 6, 7)$ using 2×1 MUX by using A as select line

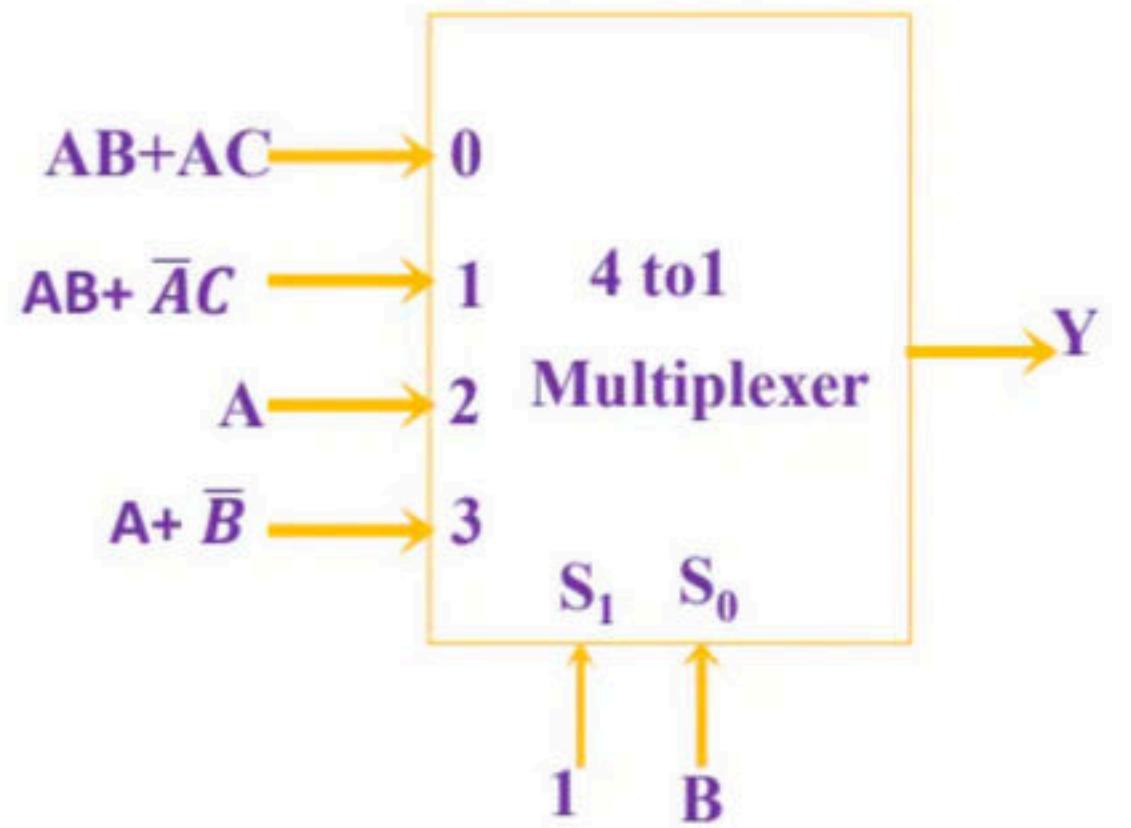
Q) Design a logic circuit $F(A, B, C) = \sum m(0, 3, 6, 7)$ using 2×1 MUX by using B as select line

MUX as Universal Gate

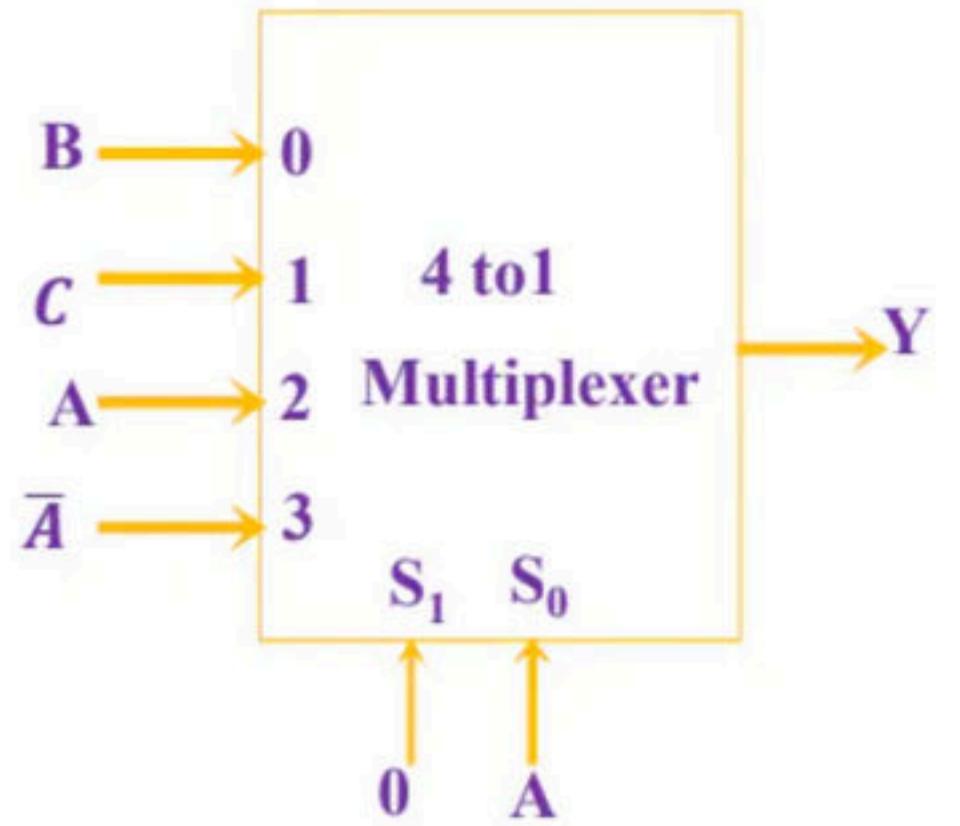
Q) Find the logic expression



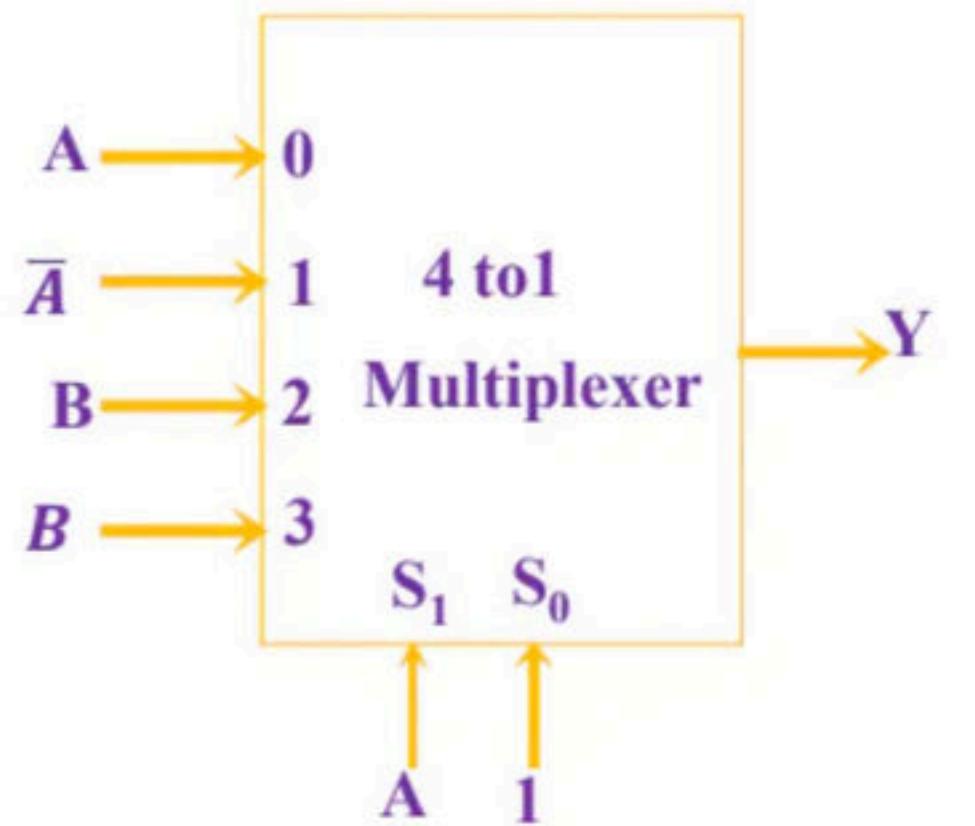
Q) Find the logic expression



Q) Find the logic expression



Q) Find the logic expression



Implementation of Higher order MUX using lower order MUX

Q) Design 4×1 MUX using 2×1 MUX

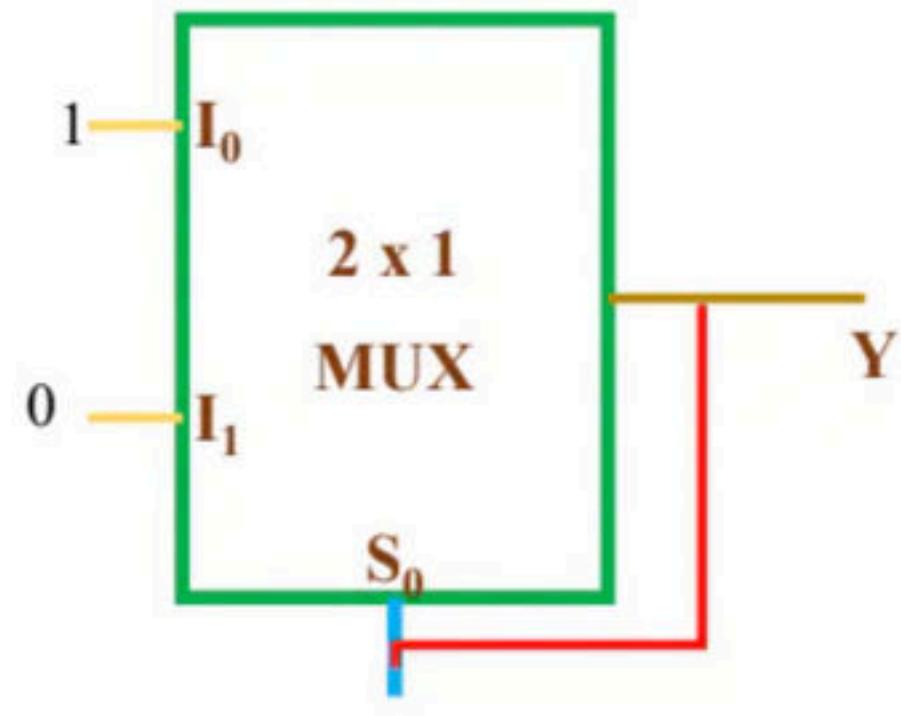
Q) Design 8×1 MUX using 2×1 MUX

Q) Design 32×1 MUX using 4×1 MUX

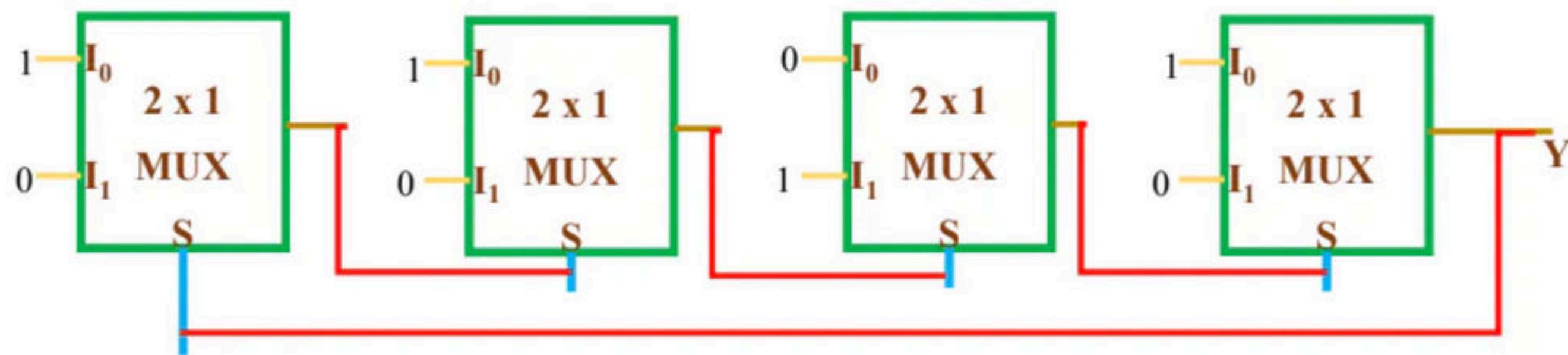
Q) Design 8×1 MUX using 4×1 MUX

Delay Analysis of MUX

Q) Draw the output waveform of the circuit , if the delay of the MUX is tpd

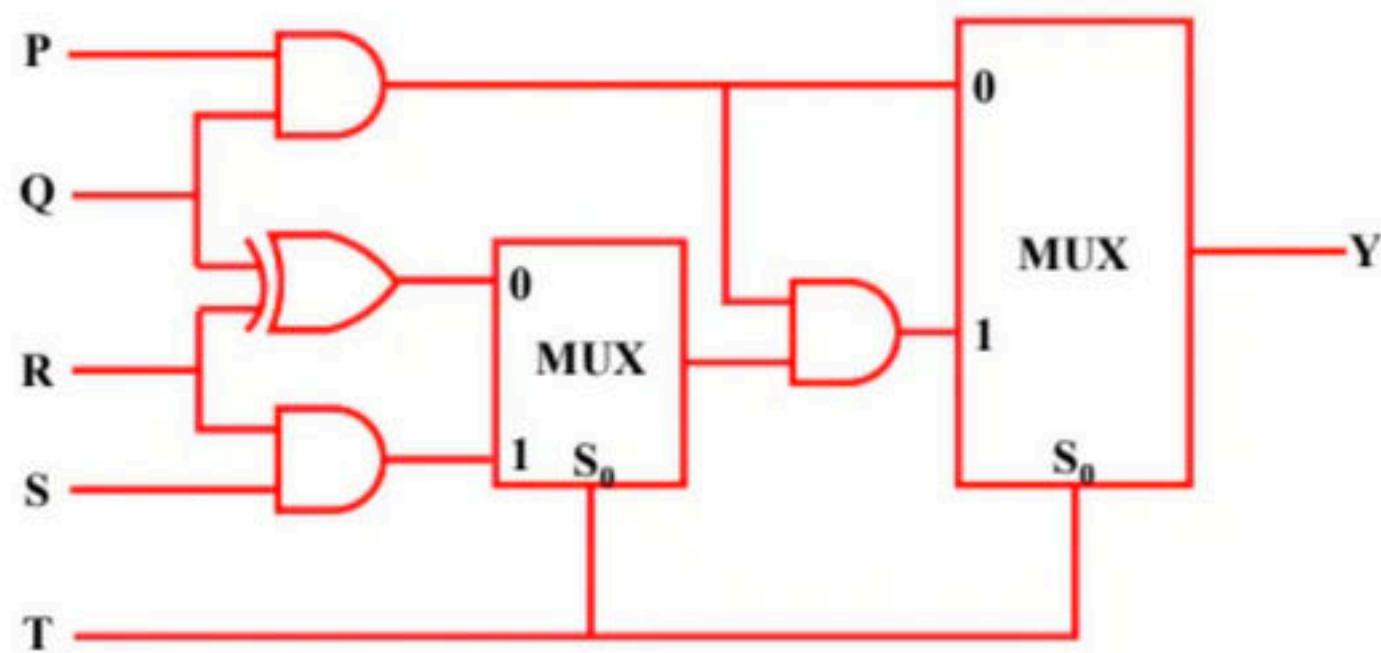


Q) Find the delay of the output Y , if the delay of each mux is 1ns



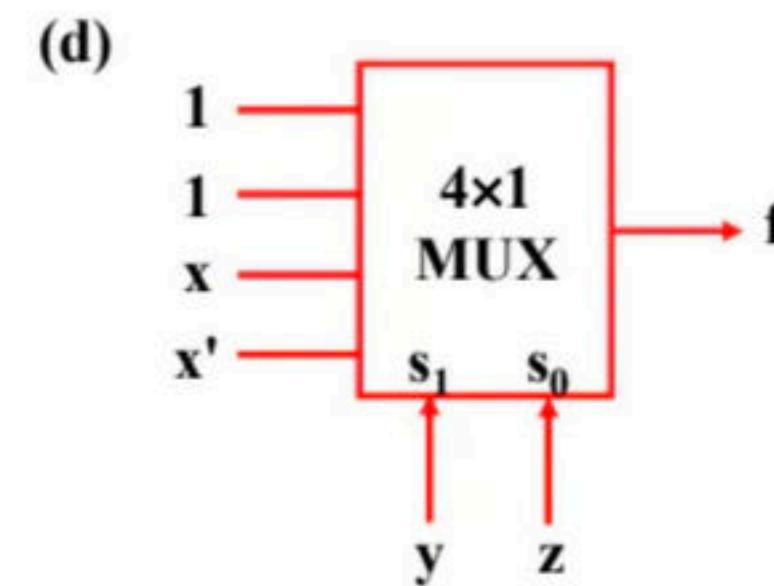
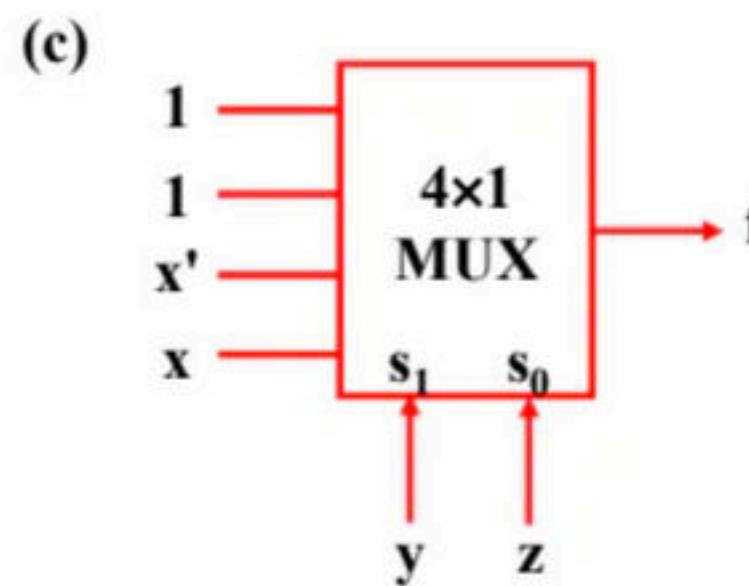
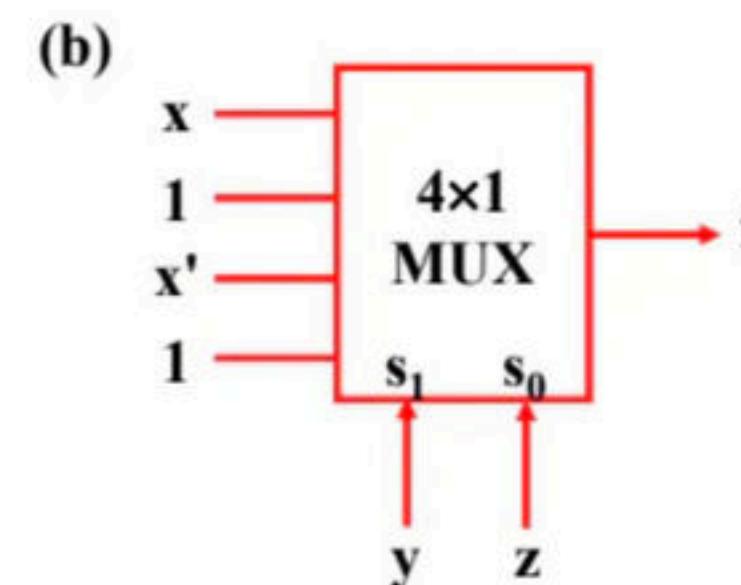
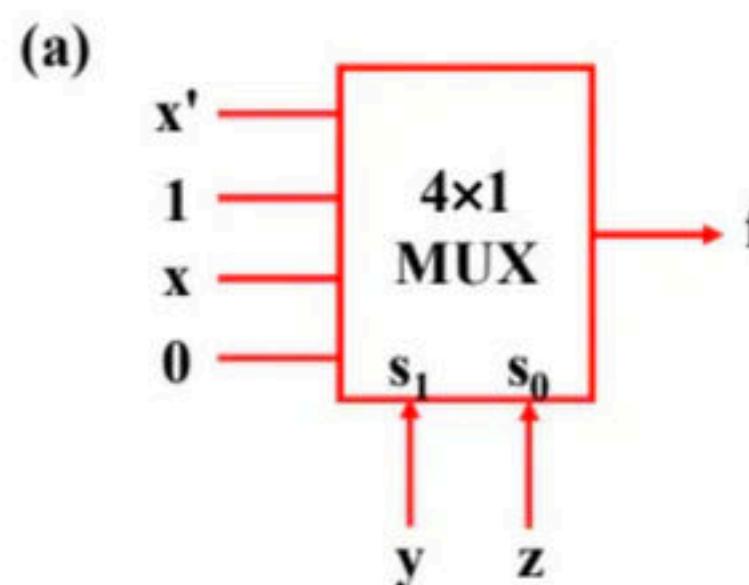
Q. The propagation delays of the XOR gate, AND gate and multiplexer (MUX) in the circuit shown in the figure are 4 ns, 2 ns and 1 ns, respectively. If all the inputs P, Q, R, S and T are applied simultaneously and held constant, the maximum propagation delay of the circuit is

- (a) 3 ns (b) 5 ns
(c) 6 ns (d) 7 ns



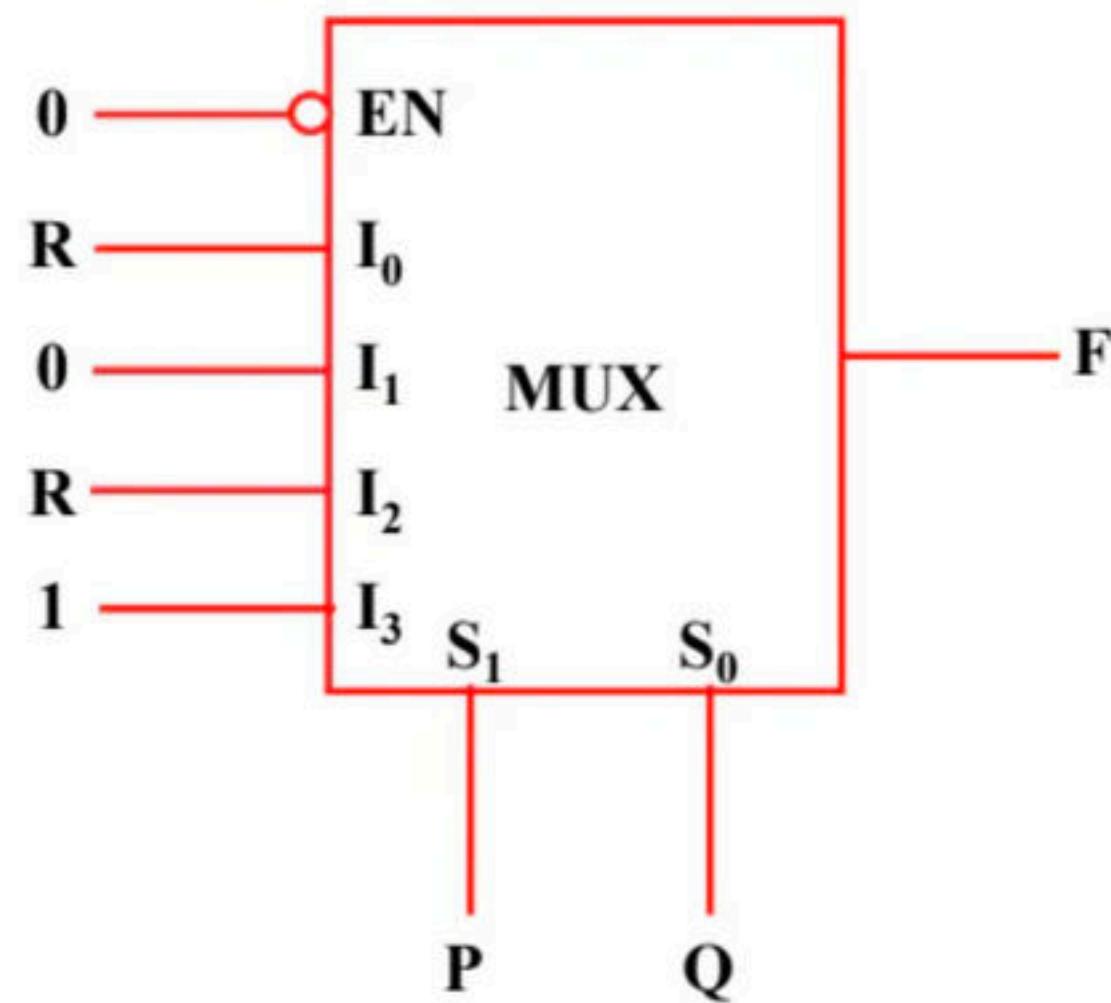
Q. Which one of the following circuits implements the Boolean function given below?

$$f(x, y, z) = m_0 + m_1 + m_3 + m_4 + m_5 + m_6, \text{ where } m_i \text{ is the } i^{\text{th}} \text{ minterm.}$$



Q. The figure below shows a multiplexer where S_1 and S_0 are the select lines. I to I_0 are the input data lines, EN is the enable line, and $F(P, Q, R)$ is the output. F is

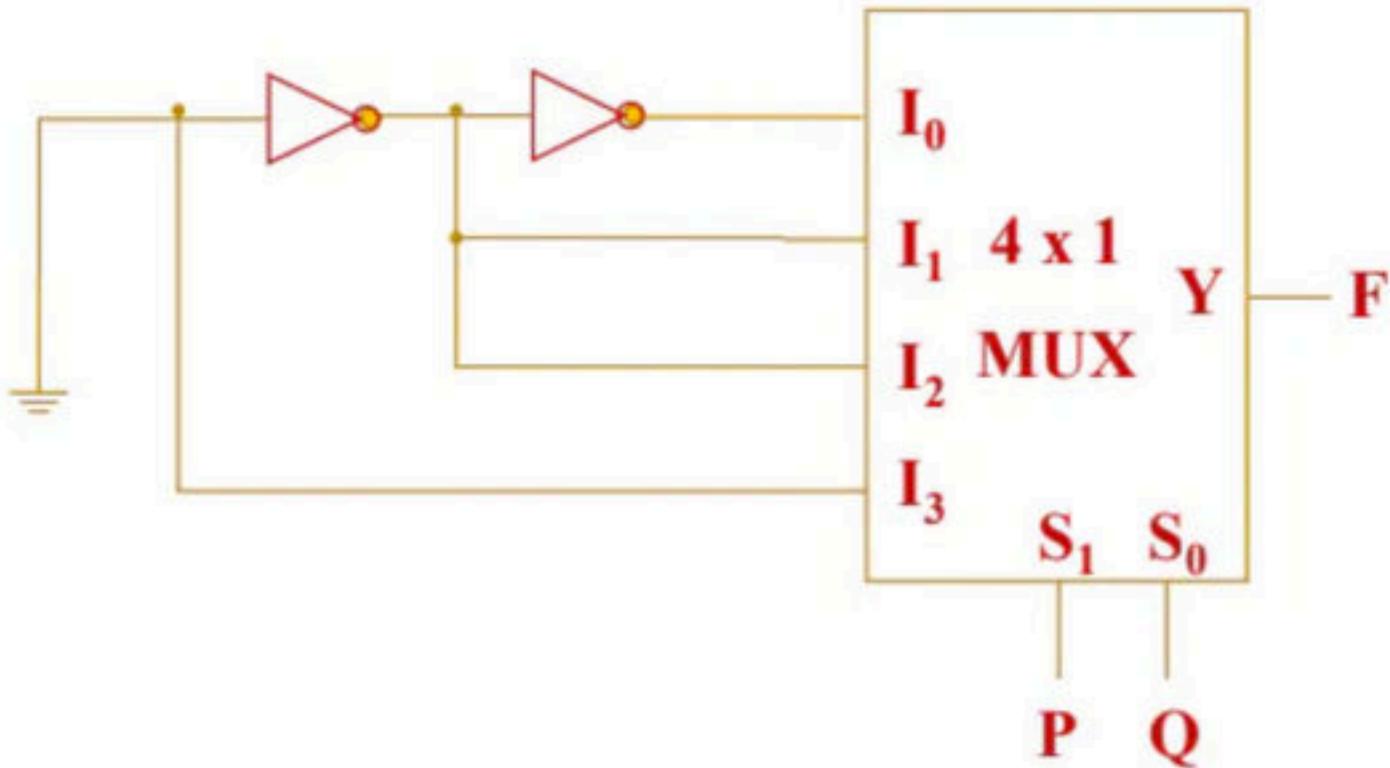
- (a) $\bar{Q} + PR$.
- (b) $P + Q\bar{R}$.
- (c) $PQ + \bar{Q}R$.
- (d) $P\bar{Q}R + \bar{P}Q$.



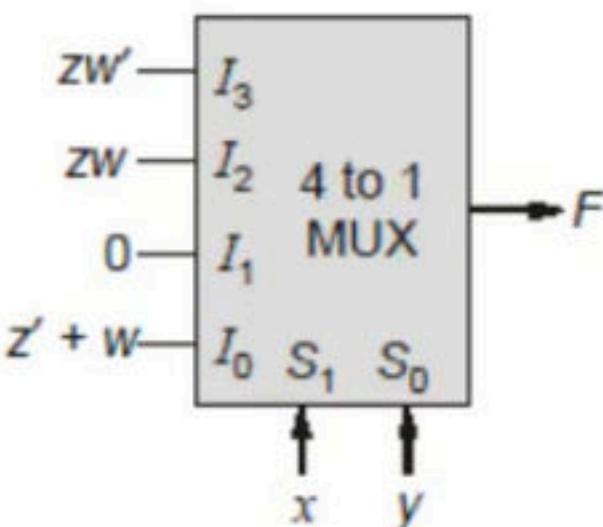
Q. The logic function implemented by the circuit below is (ground implies logic 0)

- (a) $F = \text{AND}(P, Q)$
- (c) $F = \text{XNOR}(P, Q)$

- (b) $F = \text{XOR}(P, Q)$
- (d) $F = \text{OR}(P, Q)$



A 4×1 multiplexer with two selector lines is used to realize a Boolean function, F having four Boolean variables X, Y, Z and W as shown below. S_0 and S_1 denote the least significant bit (LSB) and most significant bit (MSB) of the selector lines of the multiplexer respectively. I_0, I_1, I_2, I_3 are the input lines of the multiplexer.



The canonical sum of product representation of F is

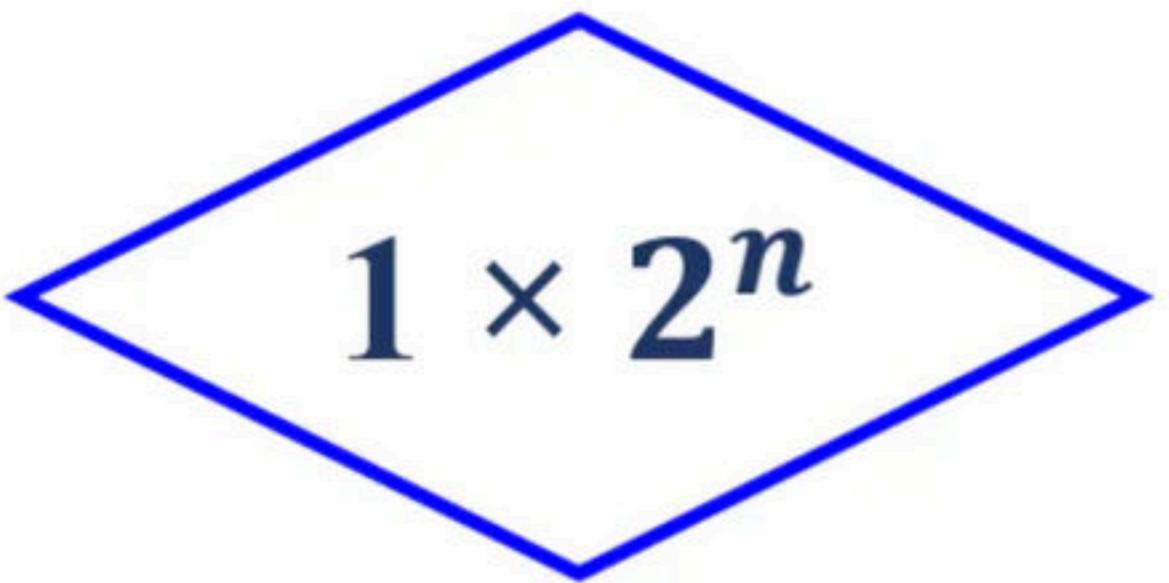
- (a) $F(X, Y, Z, W) = \Sigma m(0, 1, 3, 14, 15)$
- (b) $F(X, Y, Z, W) = \Sigma m(0, 1, 3, 11, 14)$
- (c) $F(X, Y, Z, W) = \Sigma m(2, 5, 9, 11, 14)$
- (d) $F(X, Y, Z, W) = \Sigma m(1, 3, 7, 9, 15)$

Demultiplexer

A demultiplexer is a circuit that receives information on a single line and transmits to one of the 2^n possible output lines , according to the selection lines .

- **One input to many output**
- **Data distributor**
- **One to many circuit**

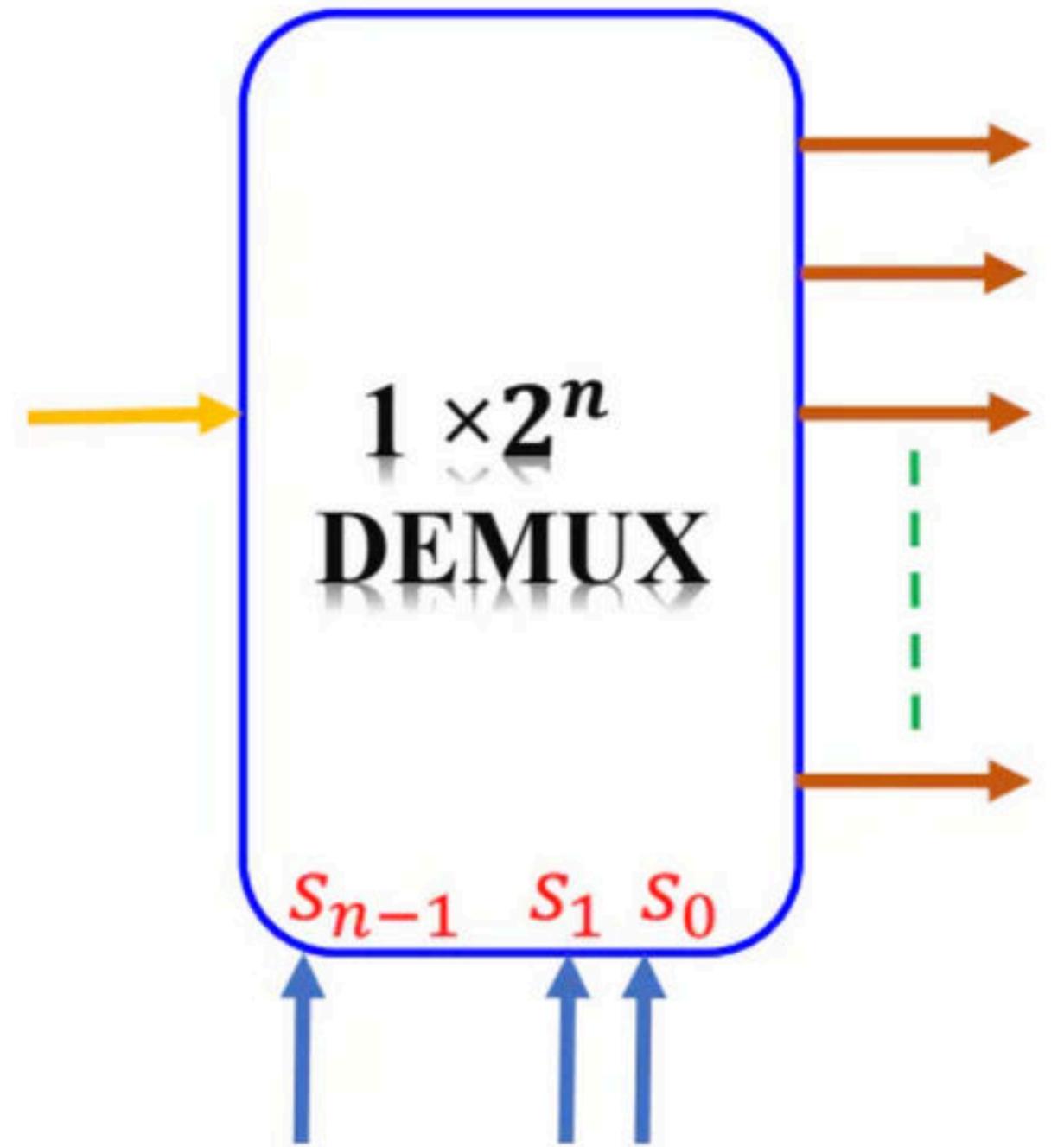
General structure



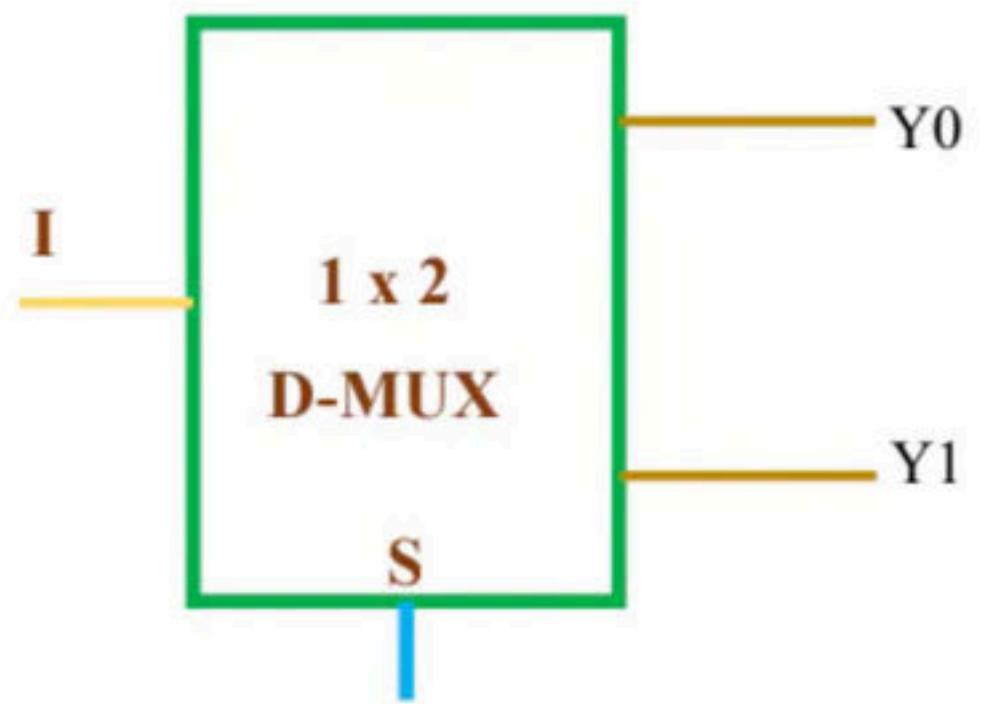
n -----> number of select lines

2^n -----> number of output lines

1 -----> number of inputs



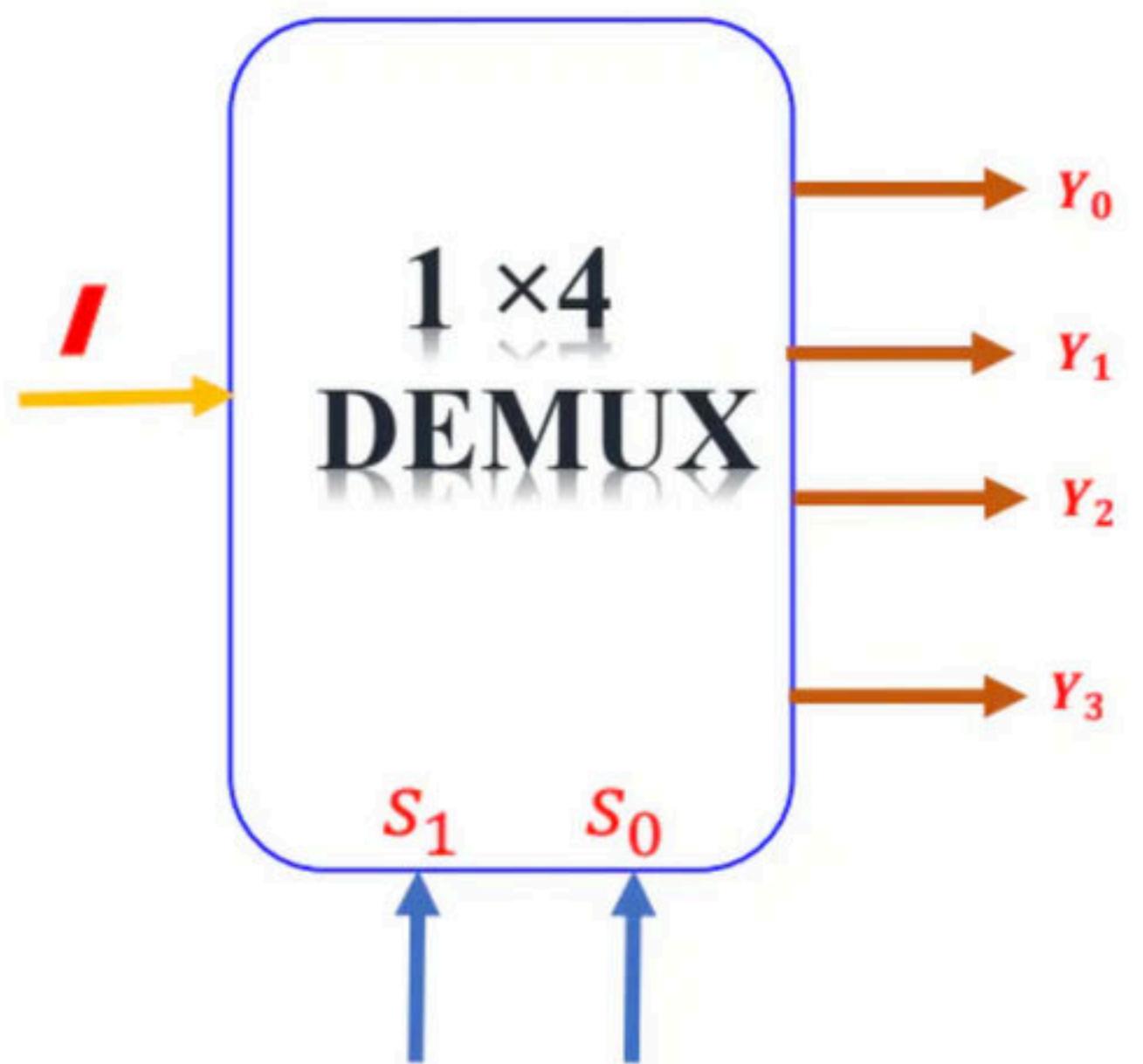
1×2 DEMUX



S	Y0	Y1

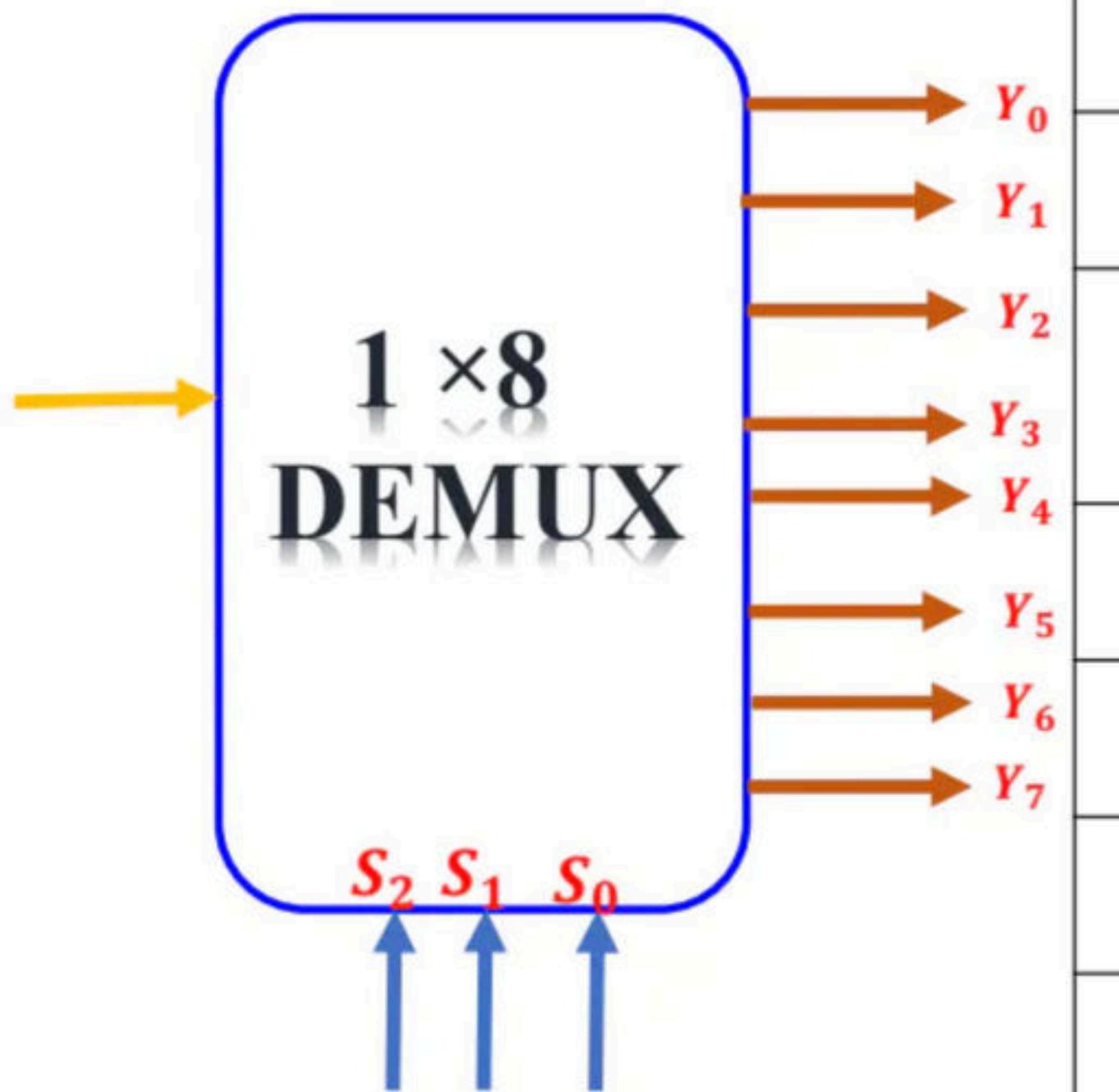
Logic circuit

1×4 Demultiplexer



S_1	S_0	Y_3	Y_2	Y_1	Y_0

1 ×8 DEMUX

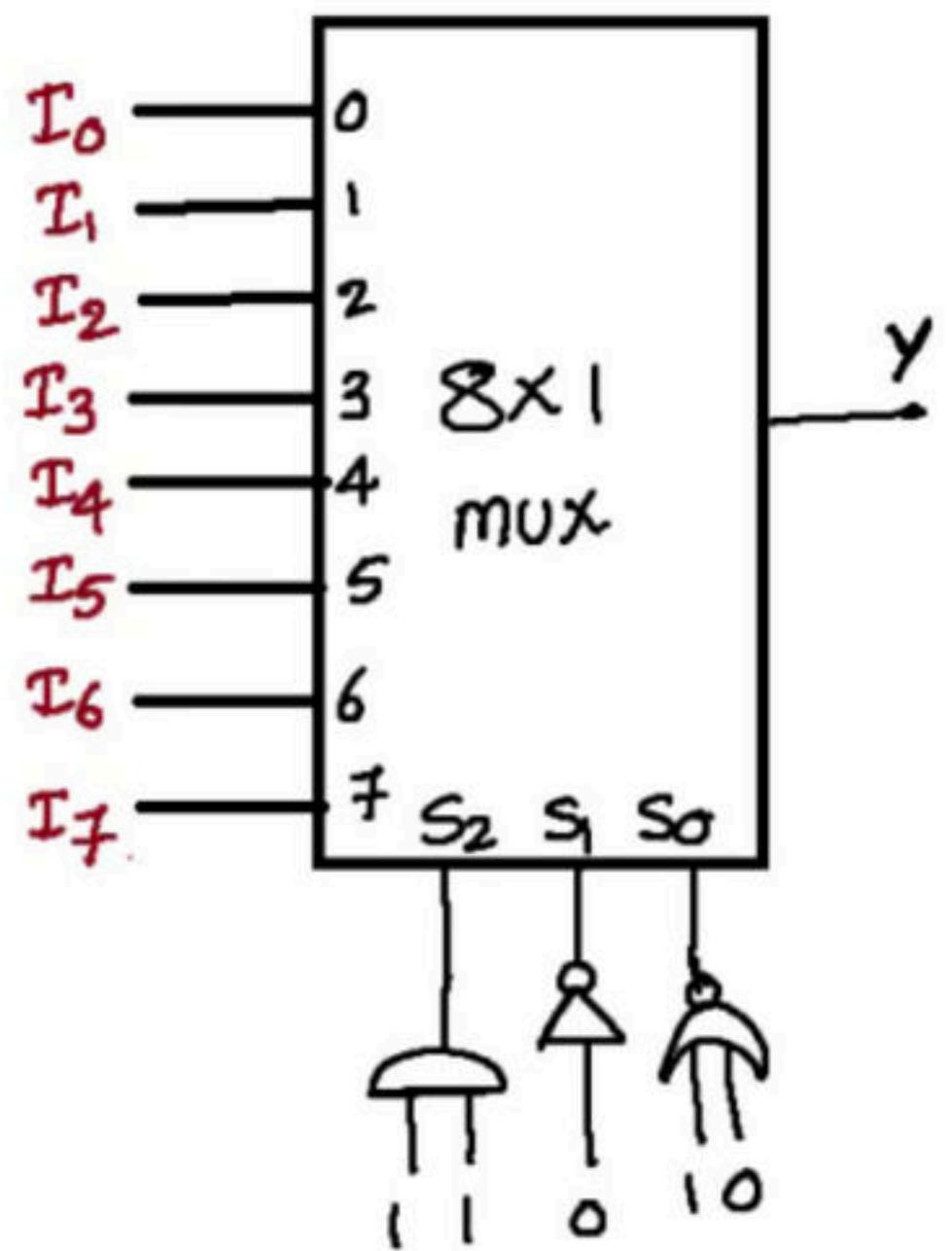


Q) Implement HA using 1×4 DEMUX

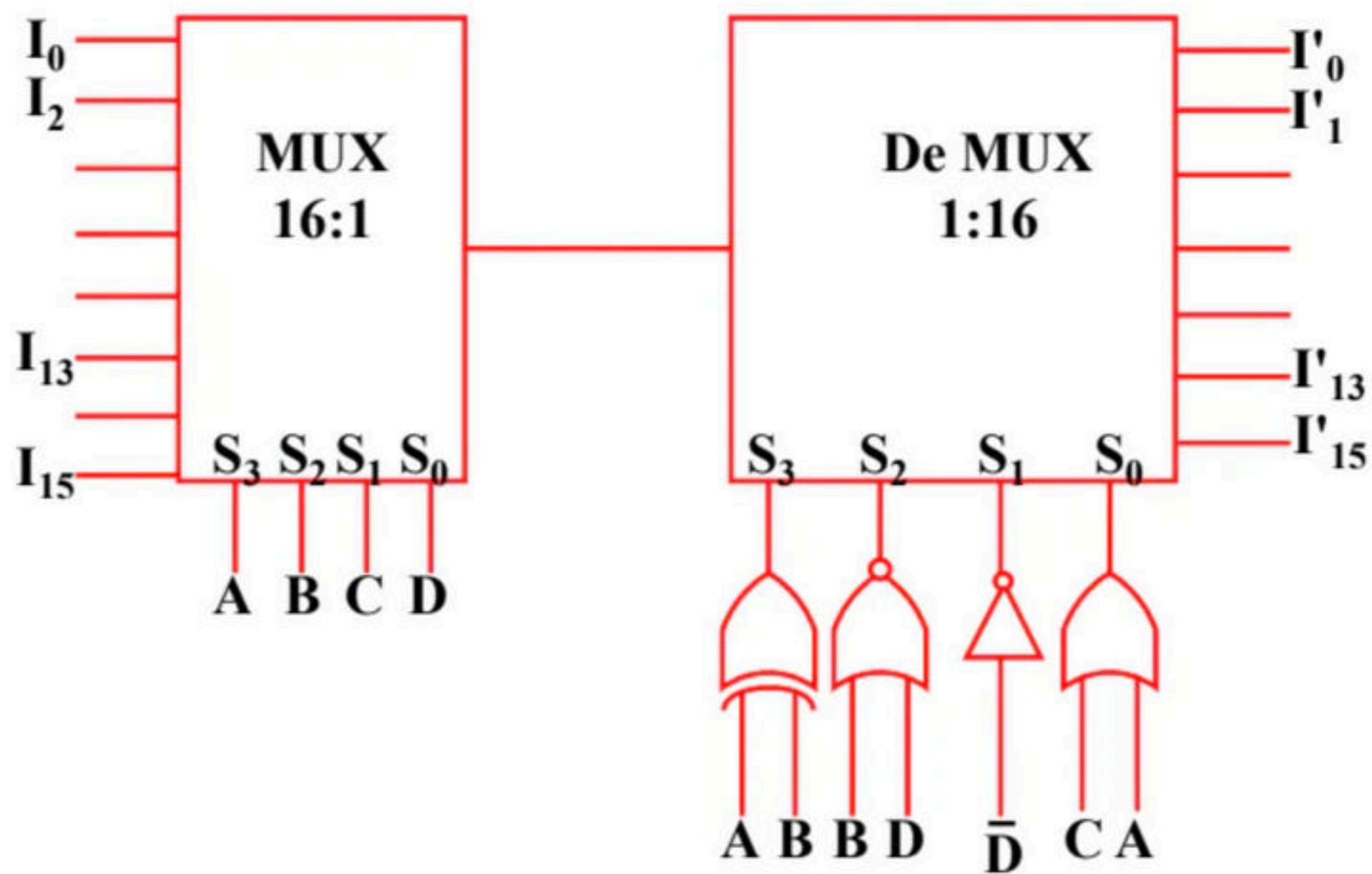
Q) Implement HS using 1×4 DEMUX

Q) Implement FA using 1×8 DEMUX

Q) The output of the mux (Y) is



Q. Consider the logical circuit given below , Input at line I_{13} in 16×1 MUX corresponds to output at line I'_n of 1×16 De-MUX. The value of ‘n’ is _____.



Implementation of higher order Demux using lower order
Demux

Q) Implement 1×4 Demux using 1×2 Demux

Q) Implement 1×16 Demux using 1×2 Demux

Q) Implement 1×8 Demux using 1×4 Demux

Decoder

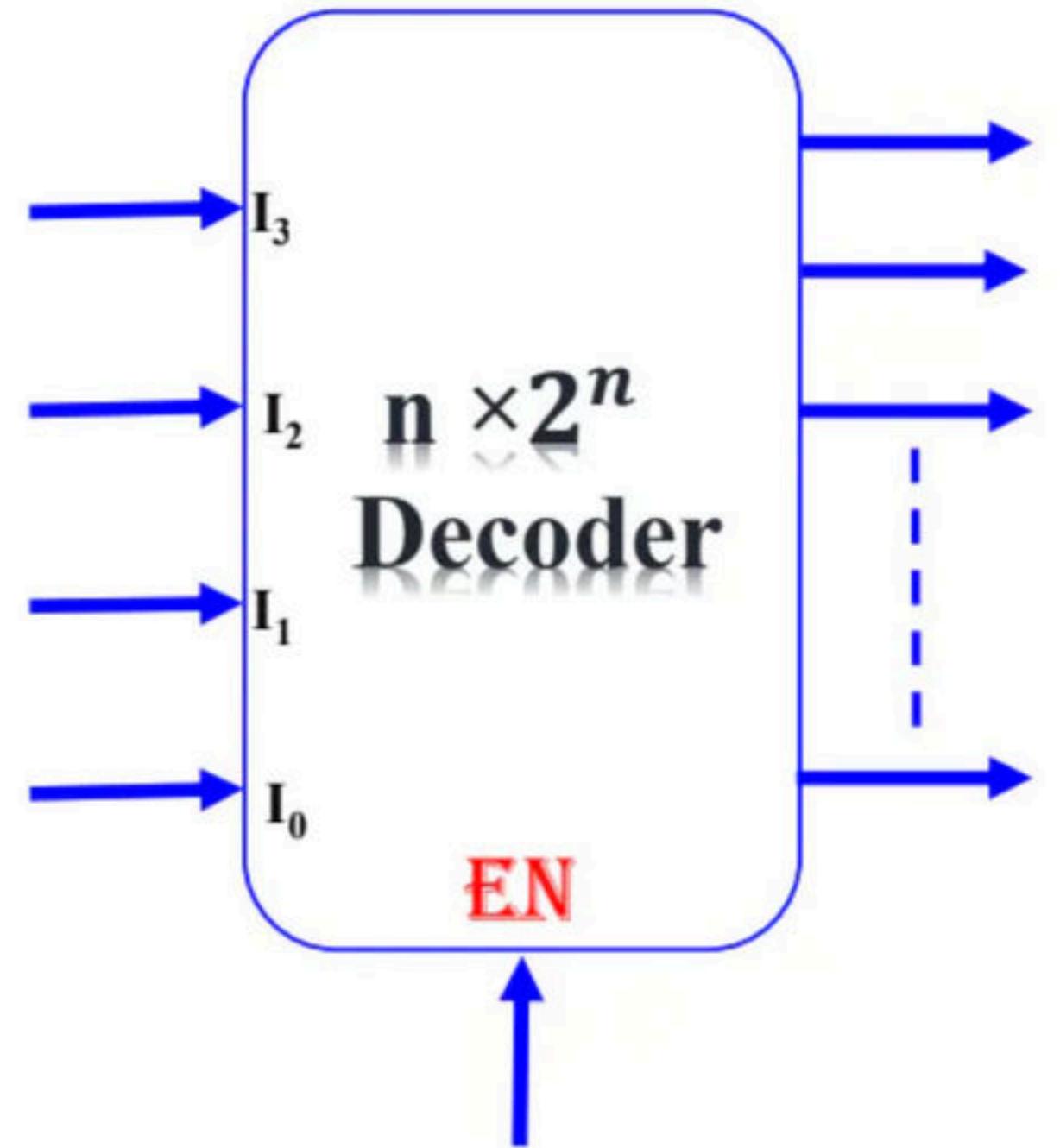
Decoder is a multi input ,multi output logic circuit which converts coded input into coded output , where the input and output codes are different .

General structure

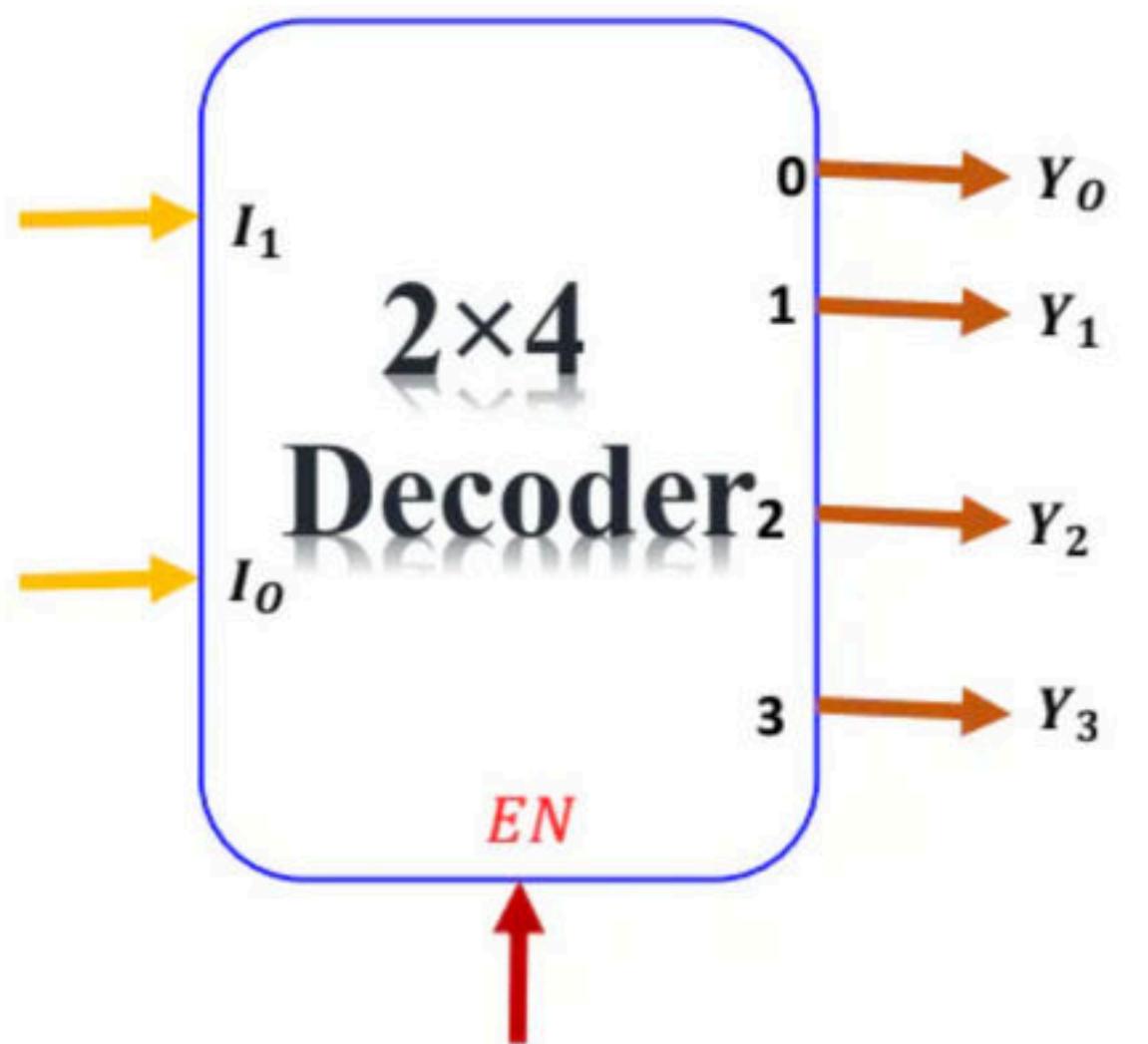
$$n \times 2^n$$

n -----> number of inputs

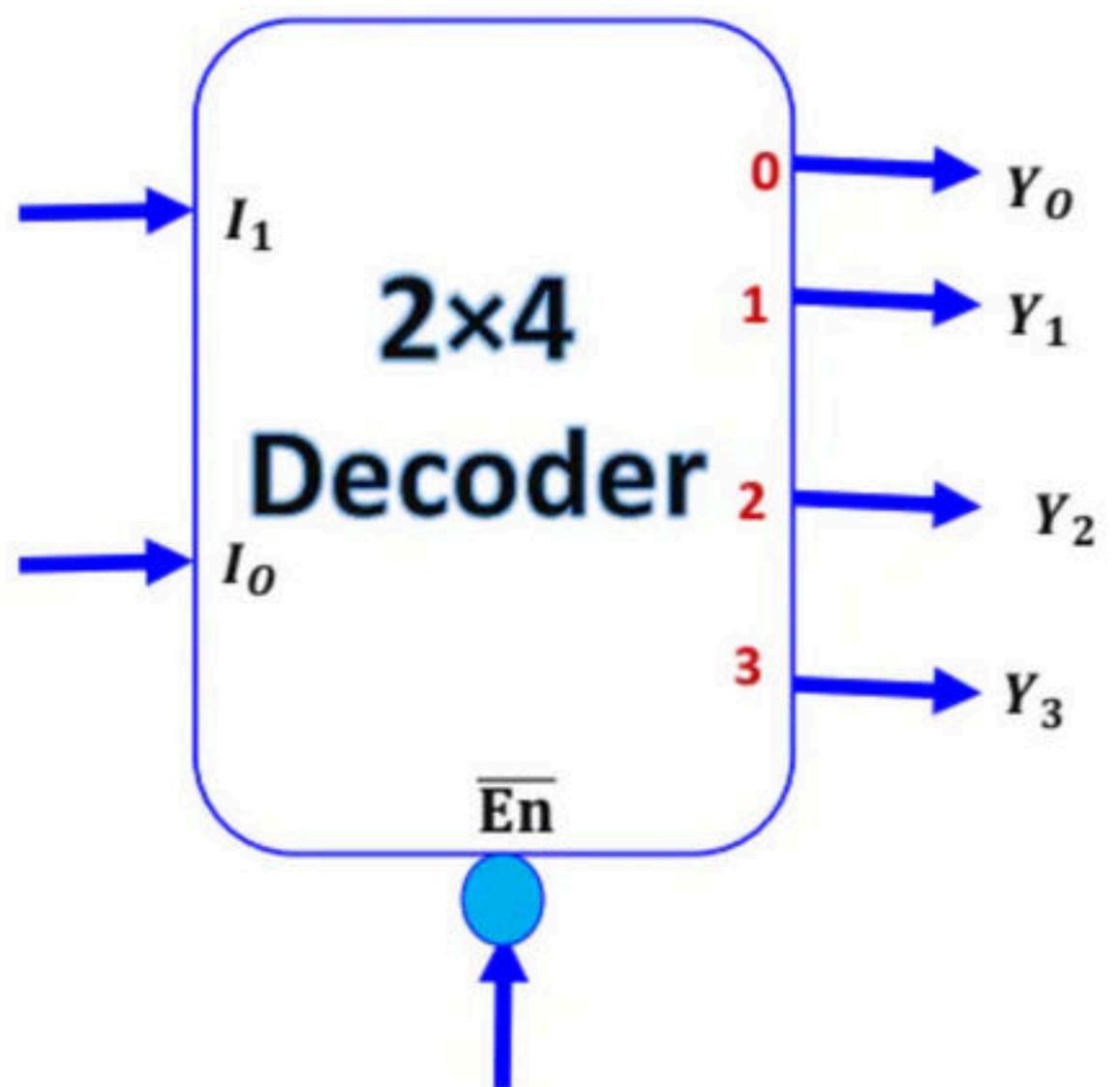
2^n -----> number of outputs



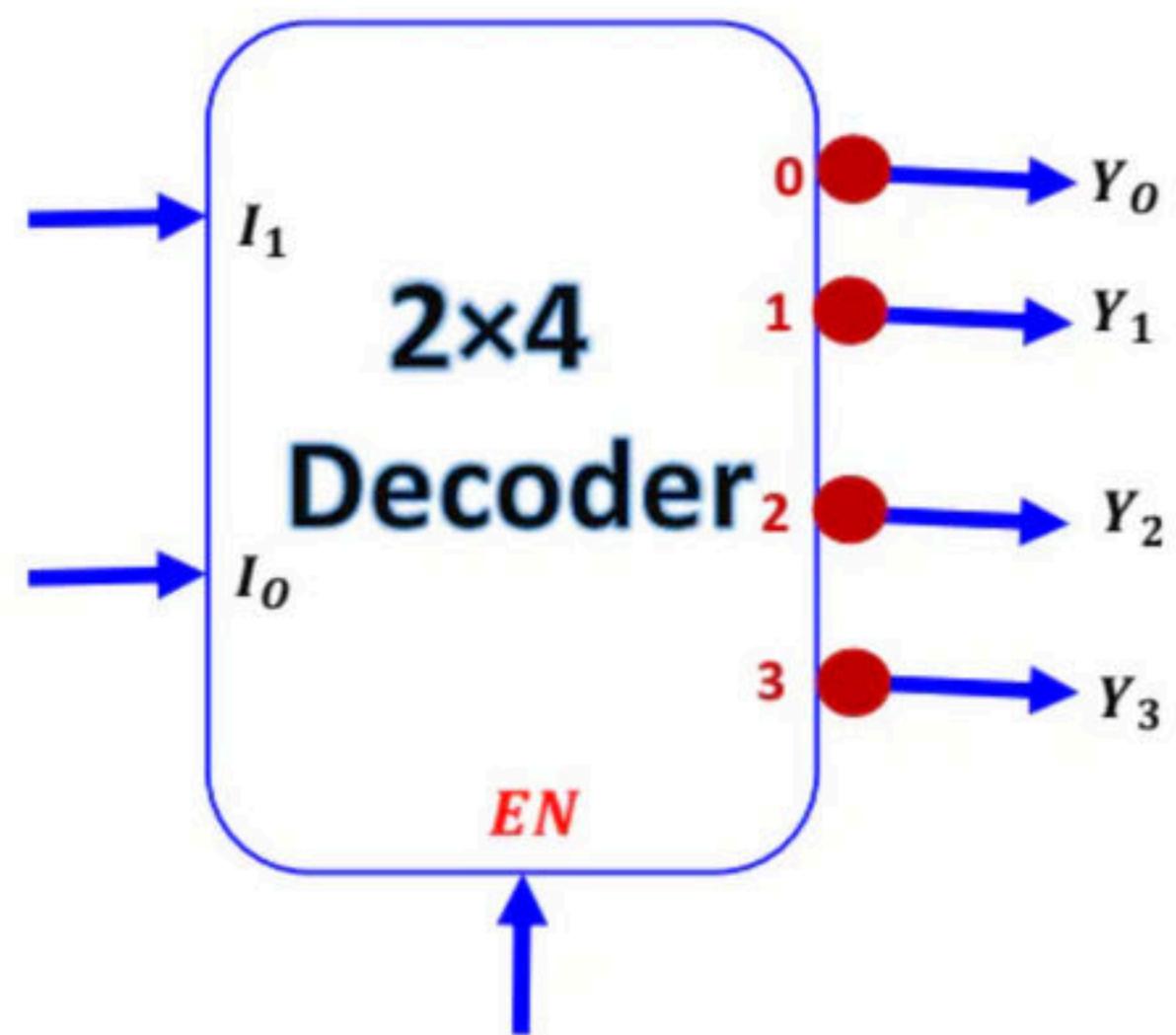
Active High Decoder



Active High Decoder

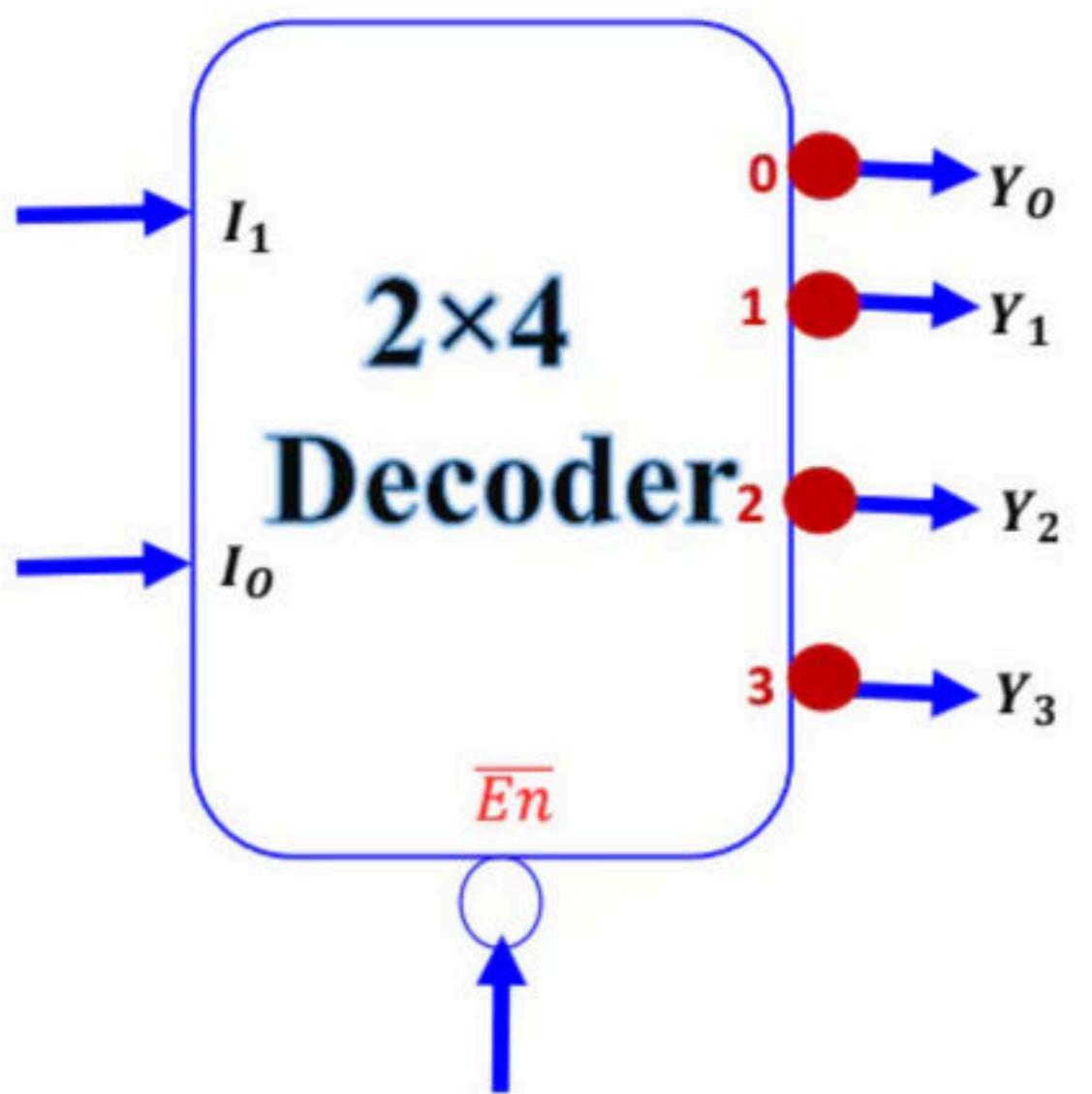


Active Low Decoder

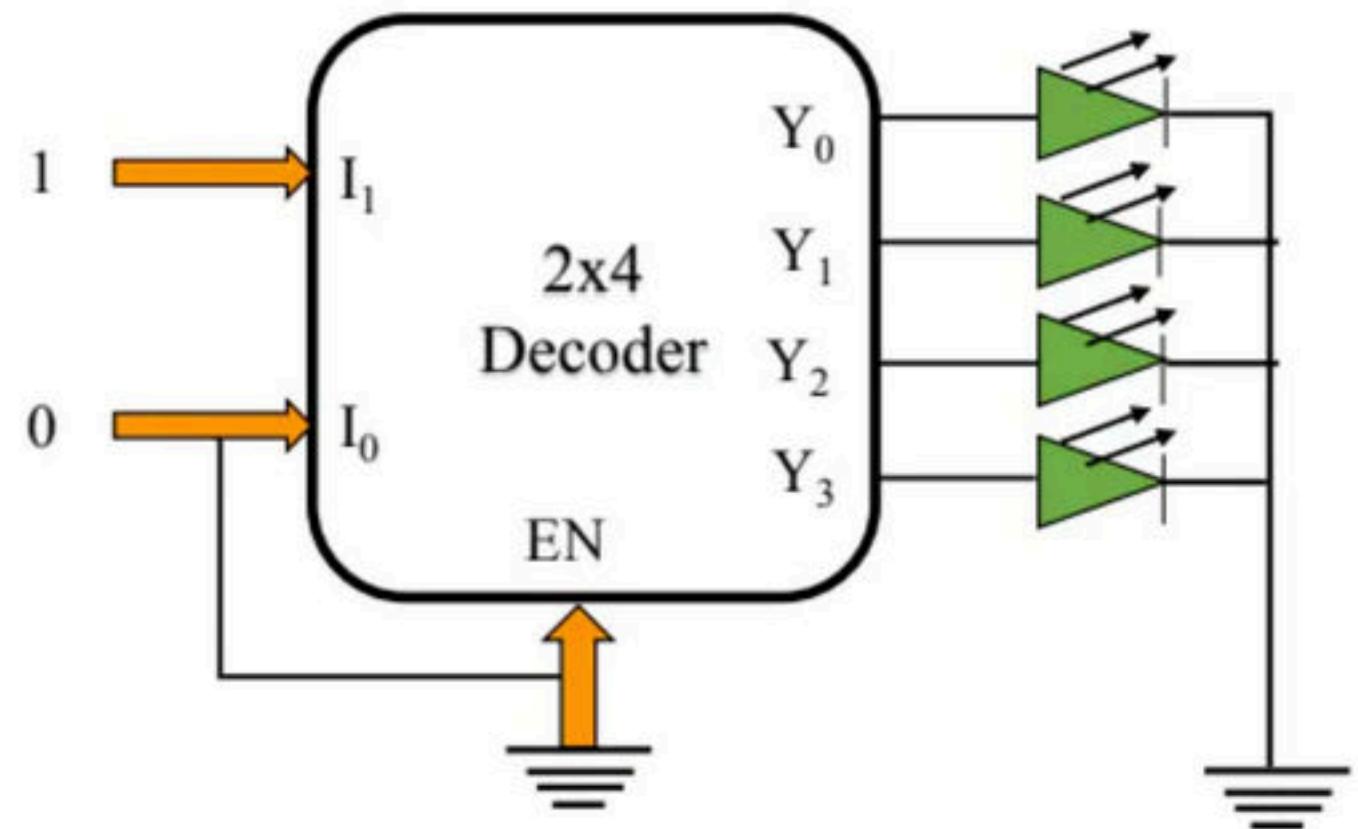


En	A	B	Y_3	Y_2	Y_1	Y_0

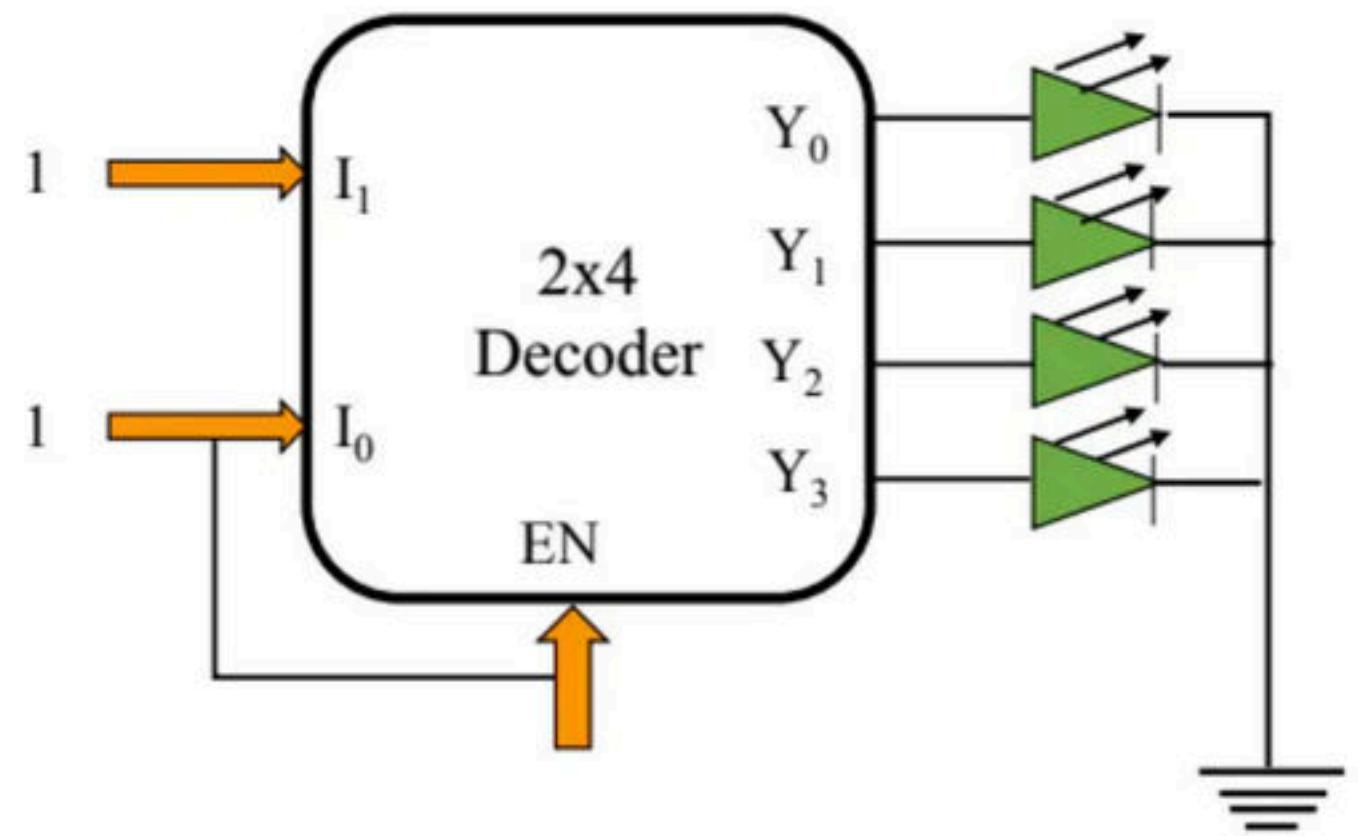
Active Low Decoder



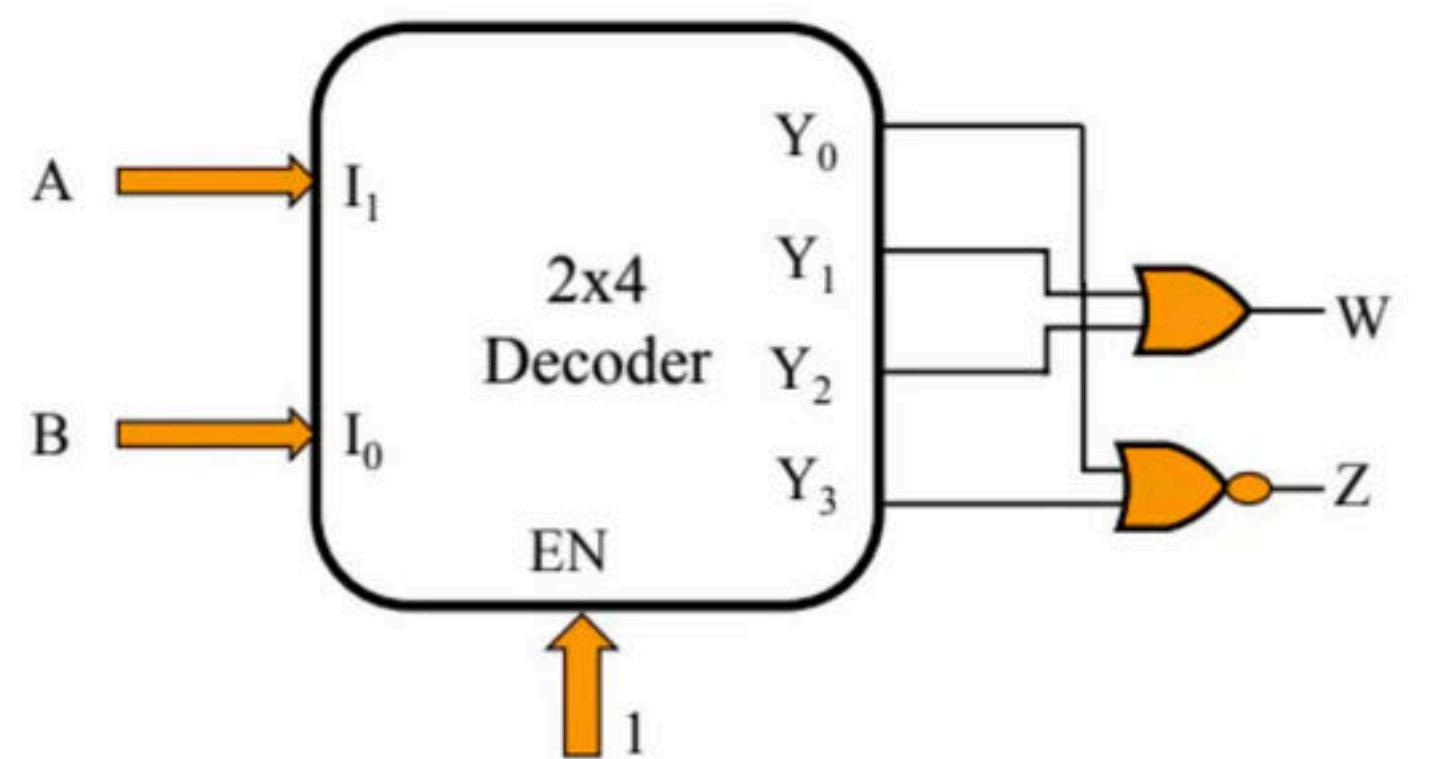
Q) Which of the following LED will glows



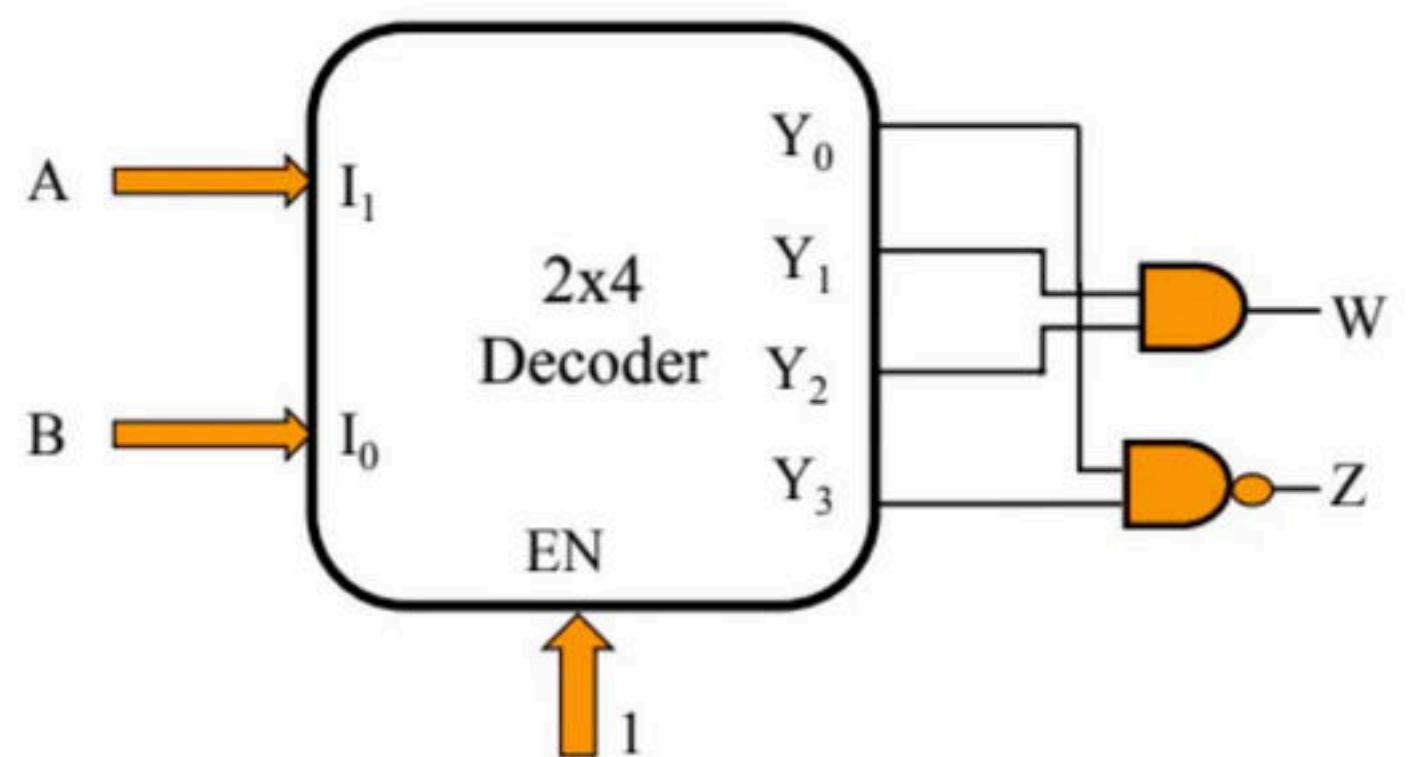
Q) Which of the following LED will glows



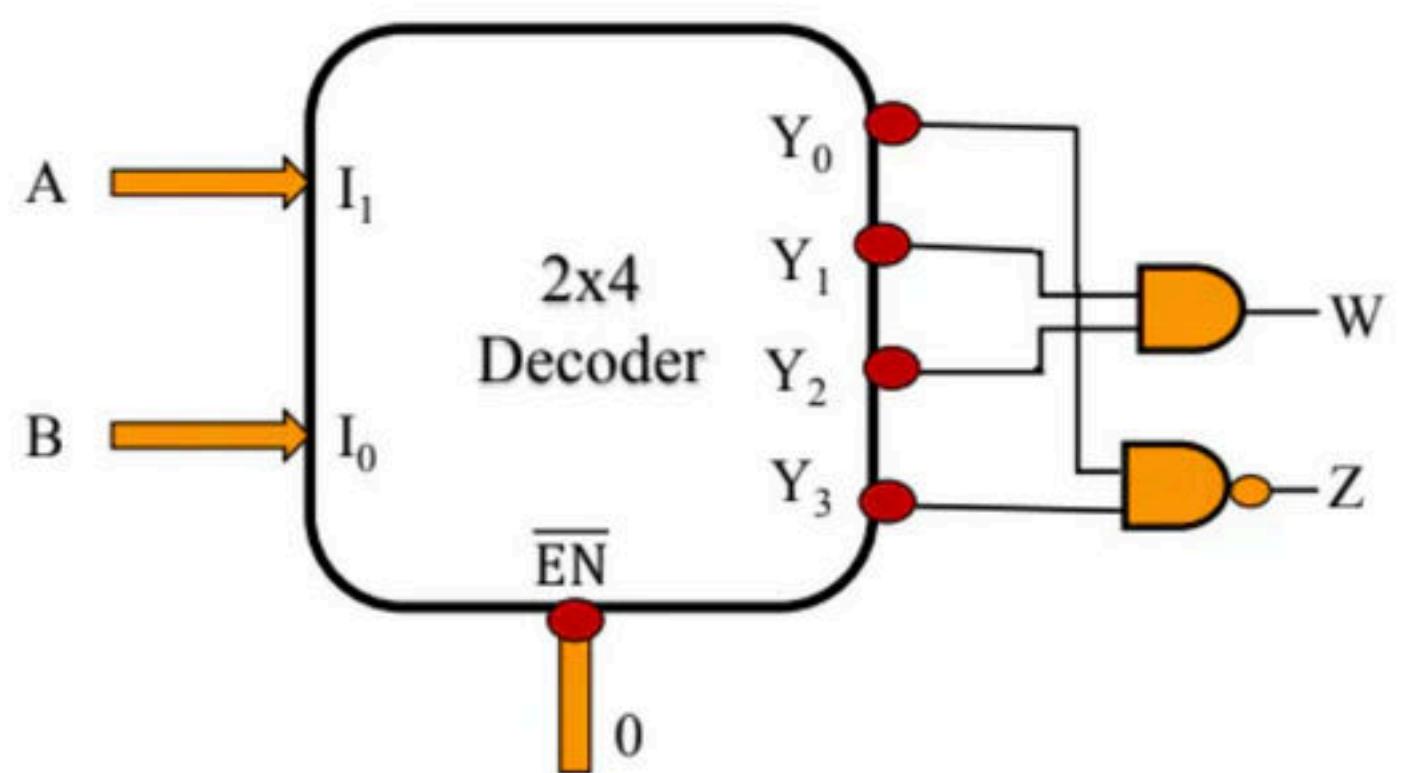
Q) Find the logic expression of W and Z



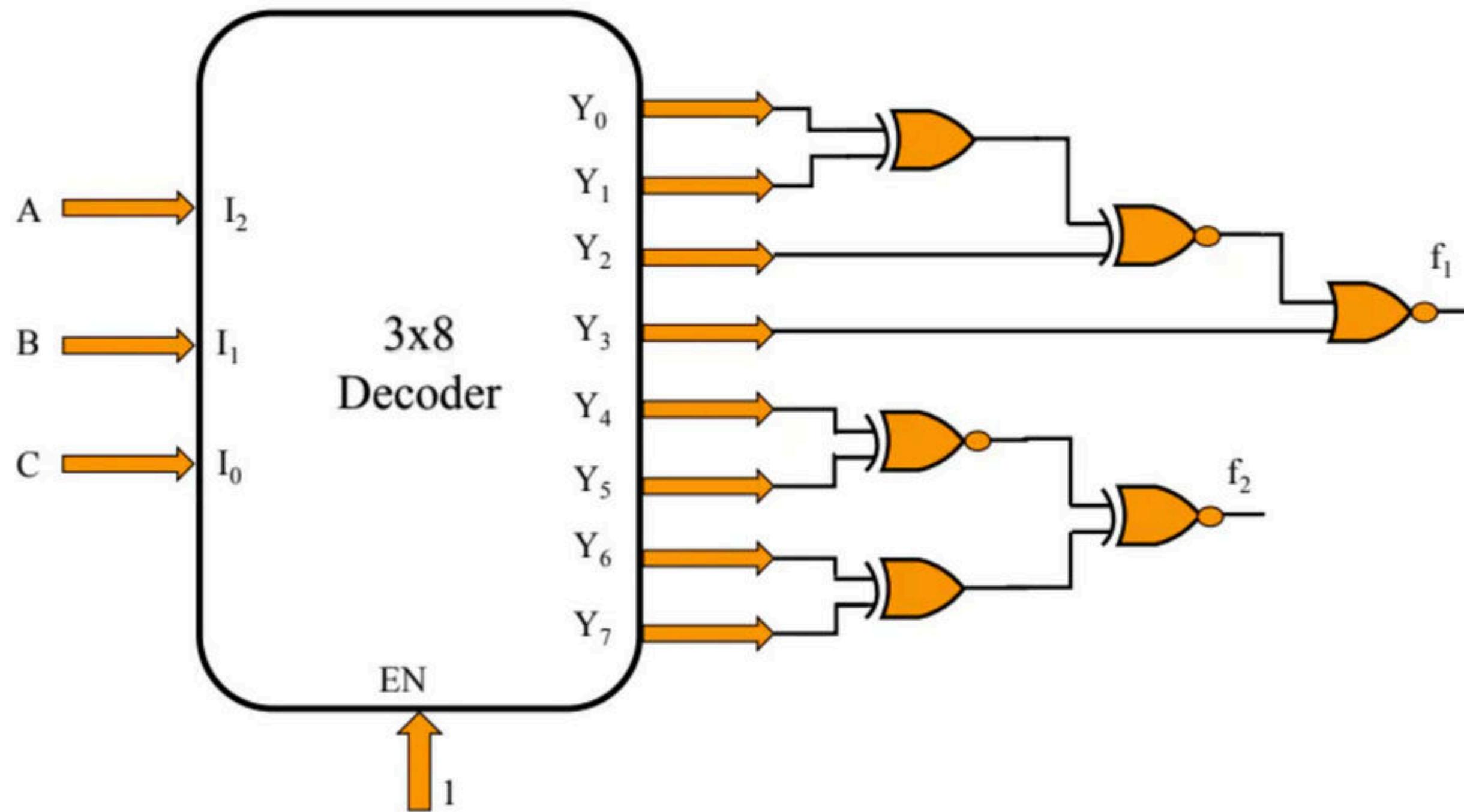
Q) Find the logic expression of W and Z



Q) Find the logic expression of W and Z



Q) The logic expression of F1 and F2

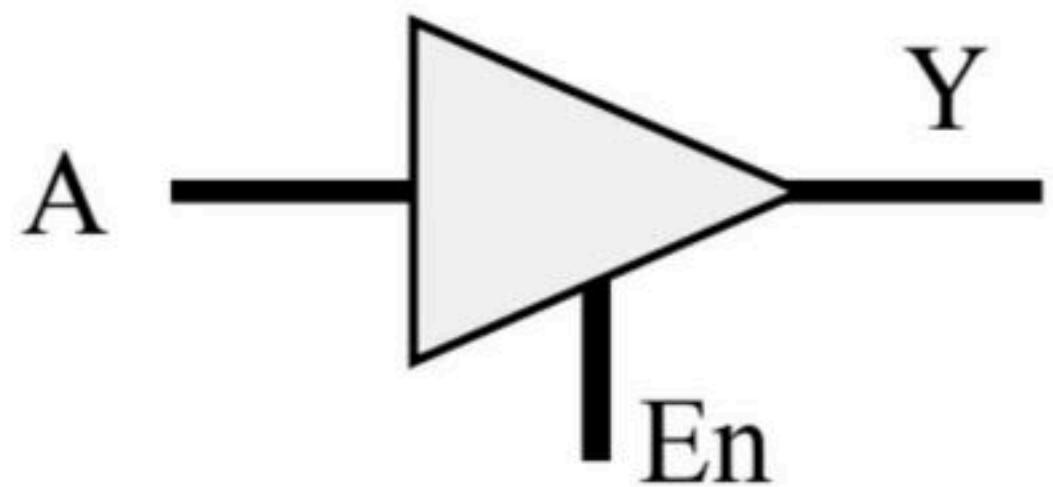


Q) Implement HA using 2×4 decoder

Q) Implement HS using 2×4 decoder

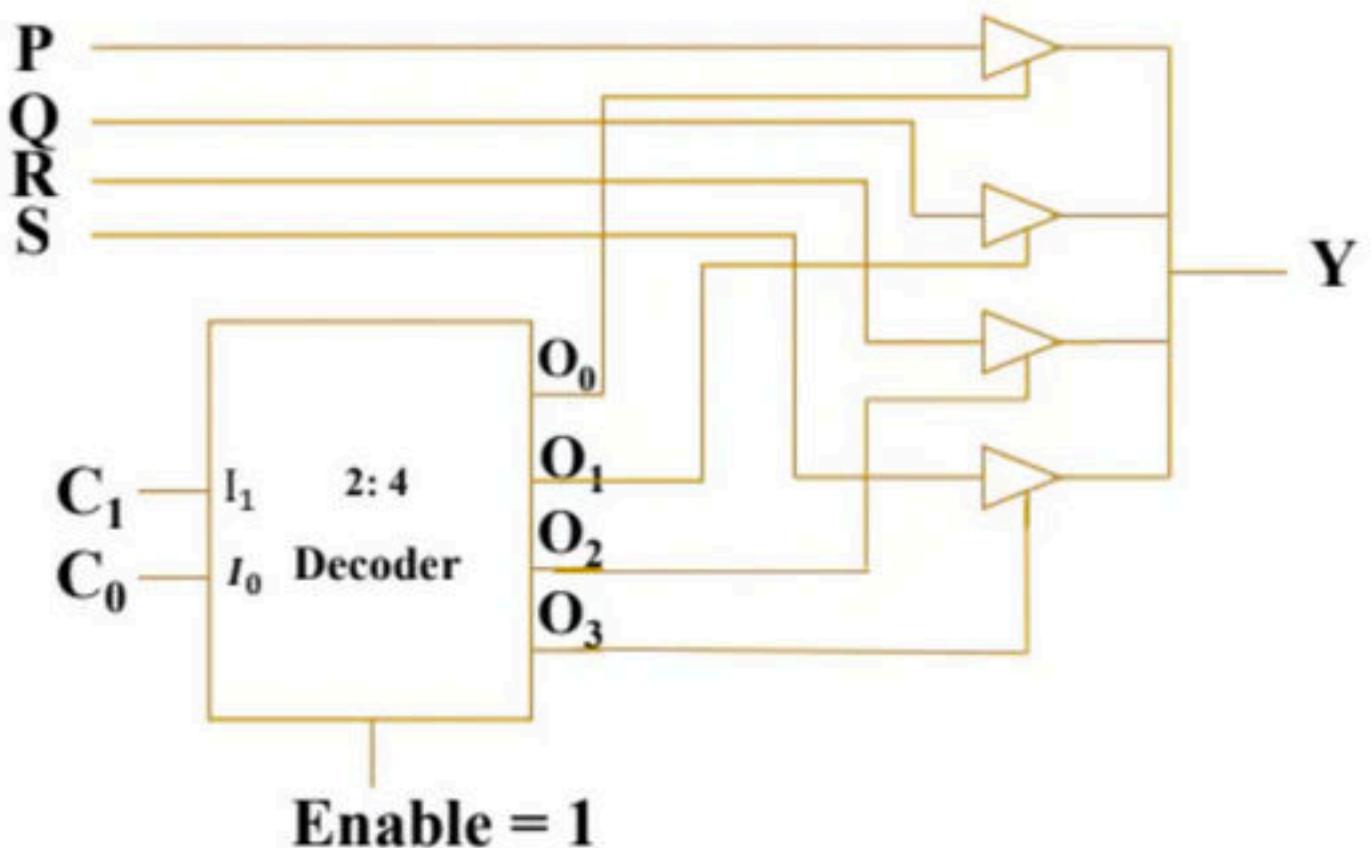
Q) implement $F(A, B, C) = A + BC$, using decoder

Tri-state Buffer



Q. The functionality implemented by the circuit below is.

- (a) 2-to-1 multiplexer
- (b) 4-to-1 multiplexer
- (c) 7-to-1 multiplexer
- (d) 6-to-1 multiplexer



Tristate buffer

Q. A logic circuit consists of two 2×4 decoder as shown below, The output of decoder are given below

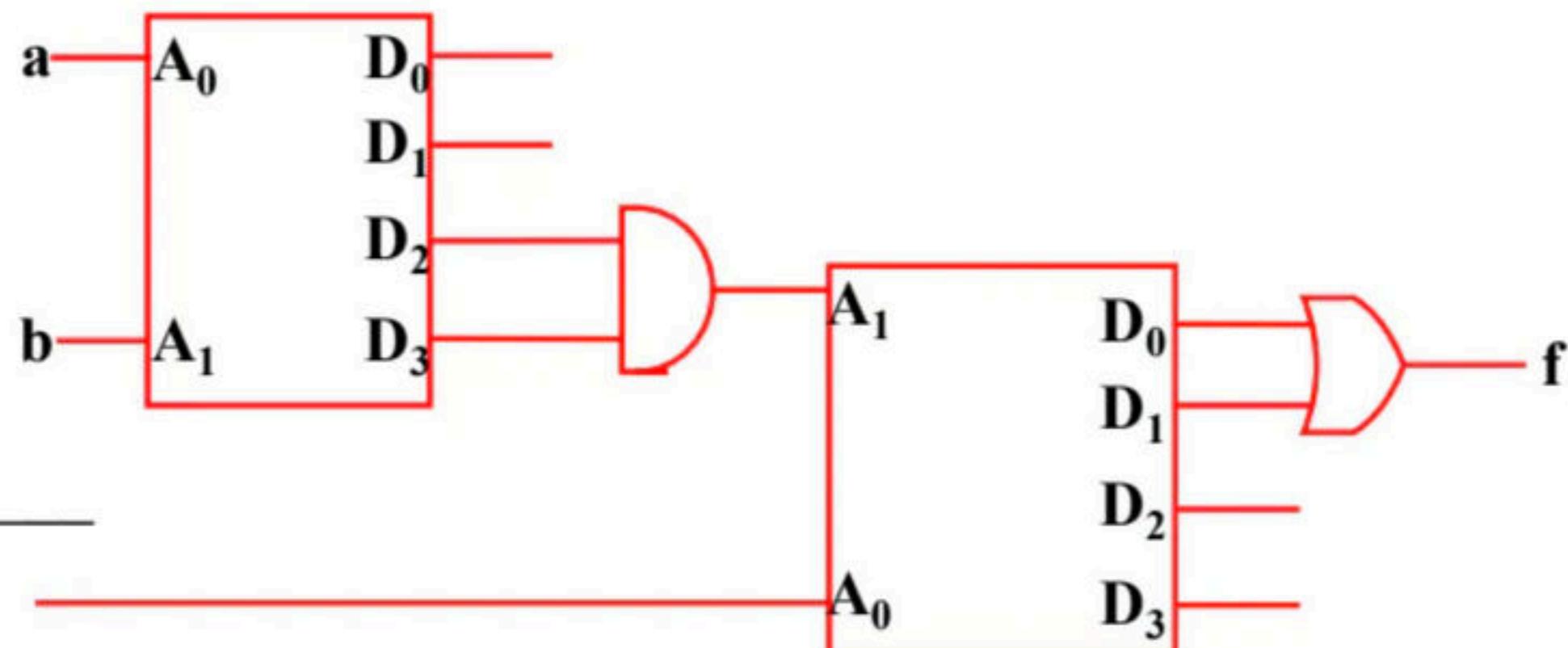
$$D_0 = 1 \text{ when } A_0 = 0, A_1 = 0$$

$$D_1 = 1 \text{ when } A_0 = 1, A_1 = 0$$

$$D_2 = 1 \text{ when } A_0 = 0, A_1 = 1$$

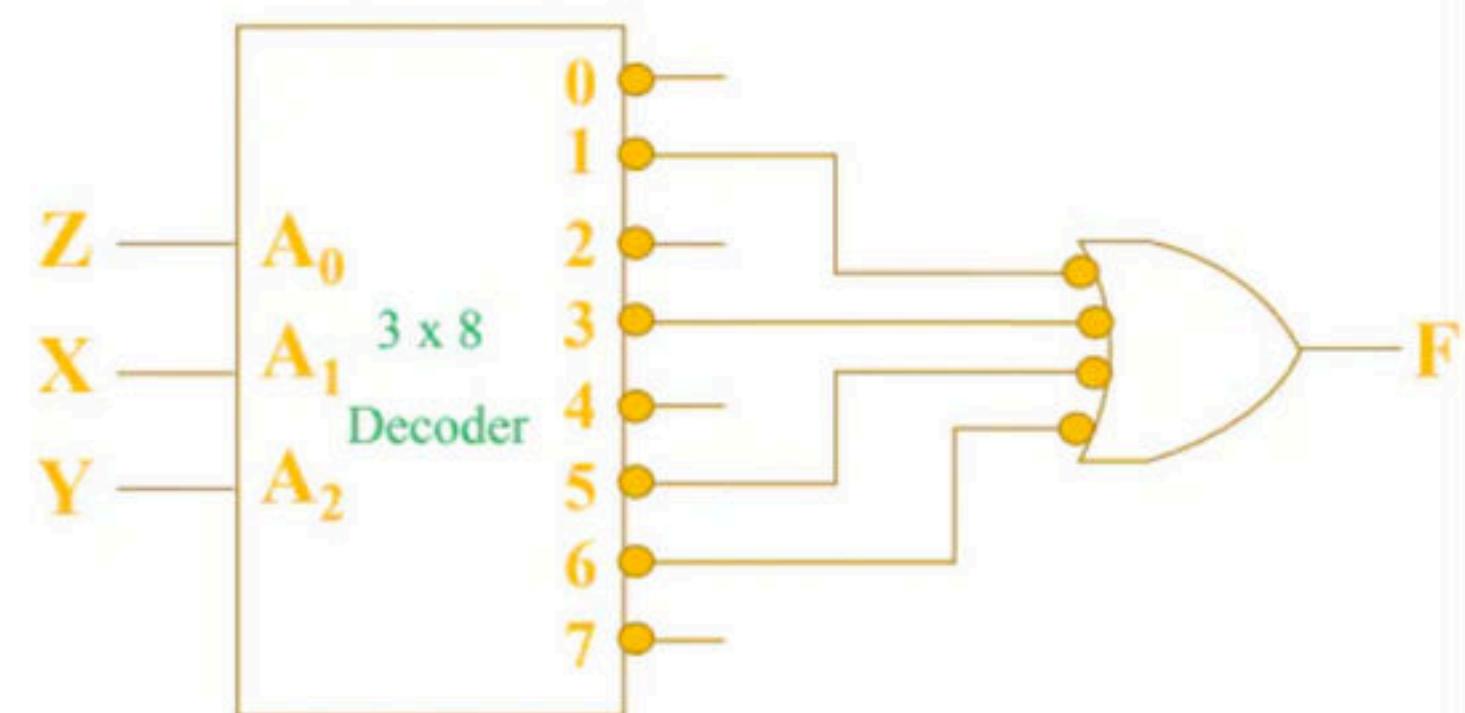
$$D_3 = 1 \text{ when } A_0 = 1, A_1 = 1$$

The value of $f(a, b, c)$ will be _____



Q. A 3 line to 8-line decoder, with active low outputs, is used to implement a 3-variable Boolean function as shown in the figure. The simplified form of Boolean function F (X,Y,Z) implemented in ‘Product of Sum’ form will be.

- (a) $(X + Z) \cdot (\bar{X} + \bar{Y} + \bar{Z}) \cdot (Y + Z)$
- (b) $(\bar{X} + \bar{Y}) \cdot (X + Y + Z) \cdot (\bar{Y} + \bar{Z})$
- (c) $(\bar{X} + \bar{Y} + Z) \cdot (\bar{X} + Y + Z) \cdot (X + \bar{Y} + Z) \cdot (X + Y + \bar{Z})$
- (d) $(\bar{X} + \bar{Y} + \bar{Z}) \cdot (\bar{X} + Y + \bar{Z}) \cdot (X + Y + Z) \cdot (X + \bar{Y} + \bar{Z})$



Conversation of
Demultiplexer <-----> Decoder



Inputs \longleftrightarrow Enable

Select lines \longleftrightarrow Inputs

Decoder is a special case of Demux , in which the select lines of Demux are treated as input's to the decoder and input of Demux is treated as Enable input of the Decoder

Implementation of higher order Decoders using lower order Decoders

Q) Implement 4×16 decoder using 2×4 decoder

Q) Implement 3×8 decoder using 2×4 decoder

Q) Implement 4×16 decoder using 3×8 decoder

Encoder

Encoder is a combinational circuit , which is used to convert

1. Octal to binary (8×3 encoder)
2. Decimal to Binary (10×4 encoder)
3. Hexadecimal to Binary (16×4 encoder)

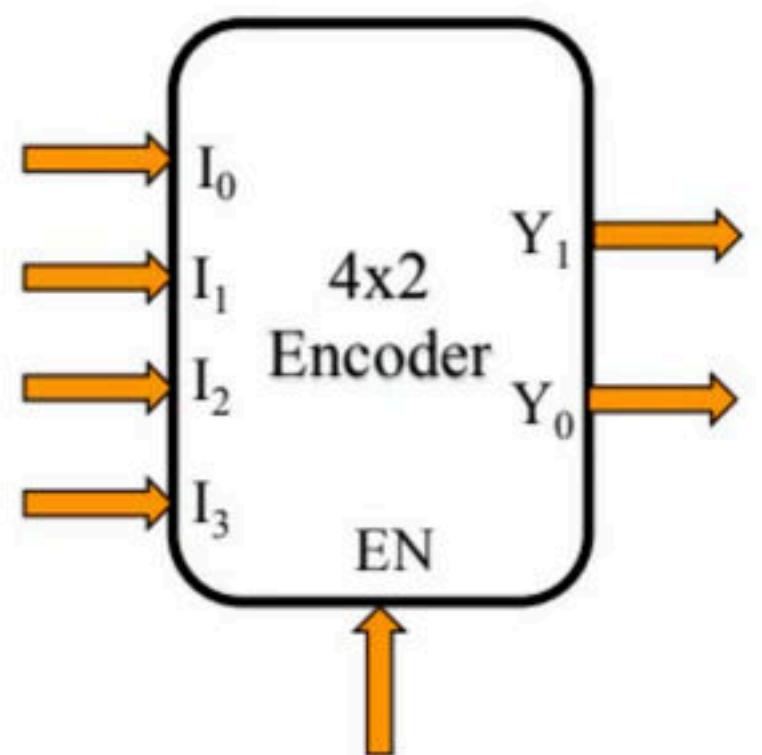
General structure

$2^n \times n$

n -----> number of outputs

2^n -----> number of inputs

4 X 2 Encoder

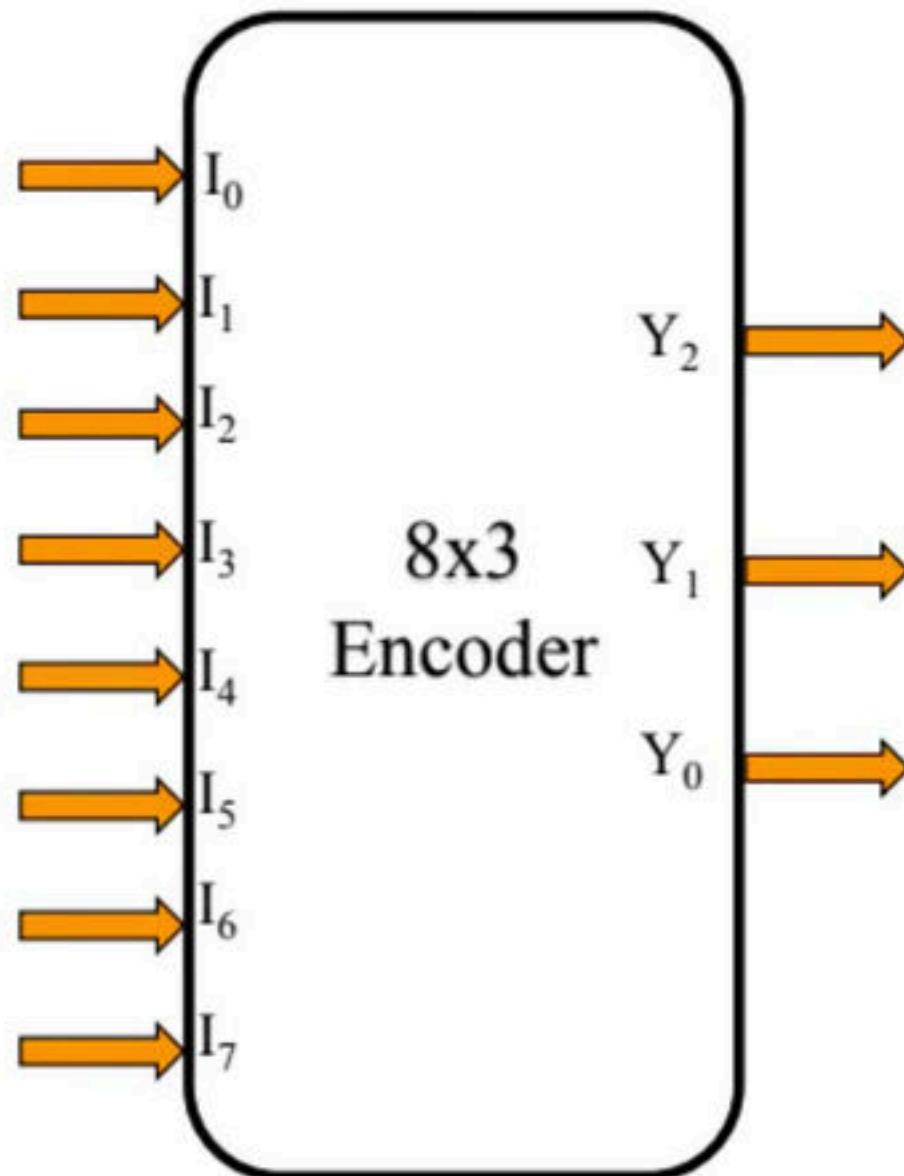


I_3	I_2	I_1	I_0	Y_1	Y_0	Valid

Drawbacks of Encoder

- For an Encoder at a time only one among the all inputs is high , remaining inputs should be zero
- If multiple inputs are simultaneously high, then the output is not valid, to avoid this restriction we will go for priority encoder.

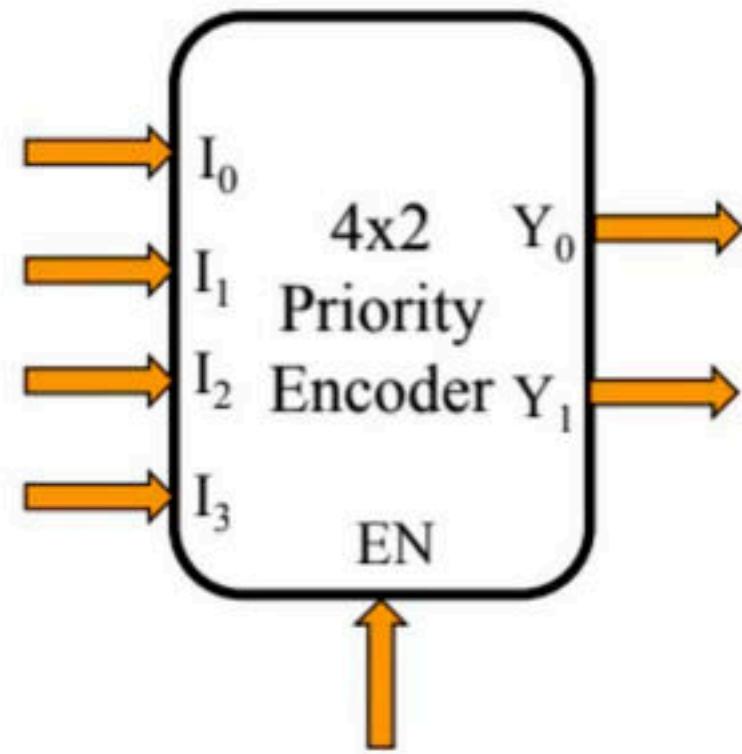
8 X 3 Encoder



Priority Encoder

Priority encoder assign priority to every input and whenever higher priority input is one , then other inputs are not consider

Priority Encoder



I3	I2	I1	I0	Y_1	Y_0	Valid

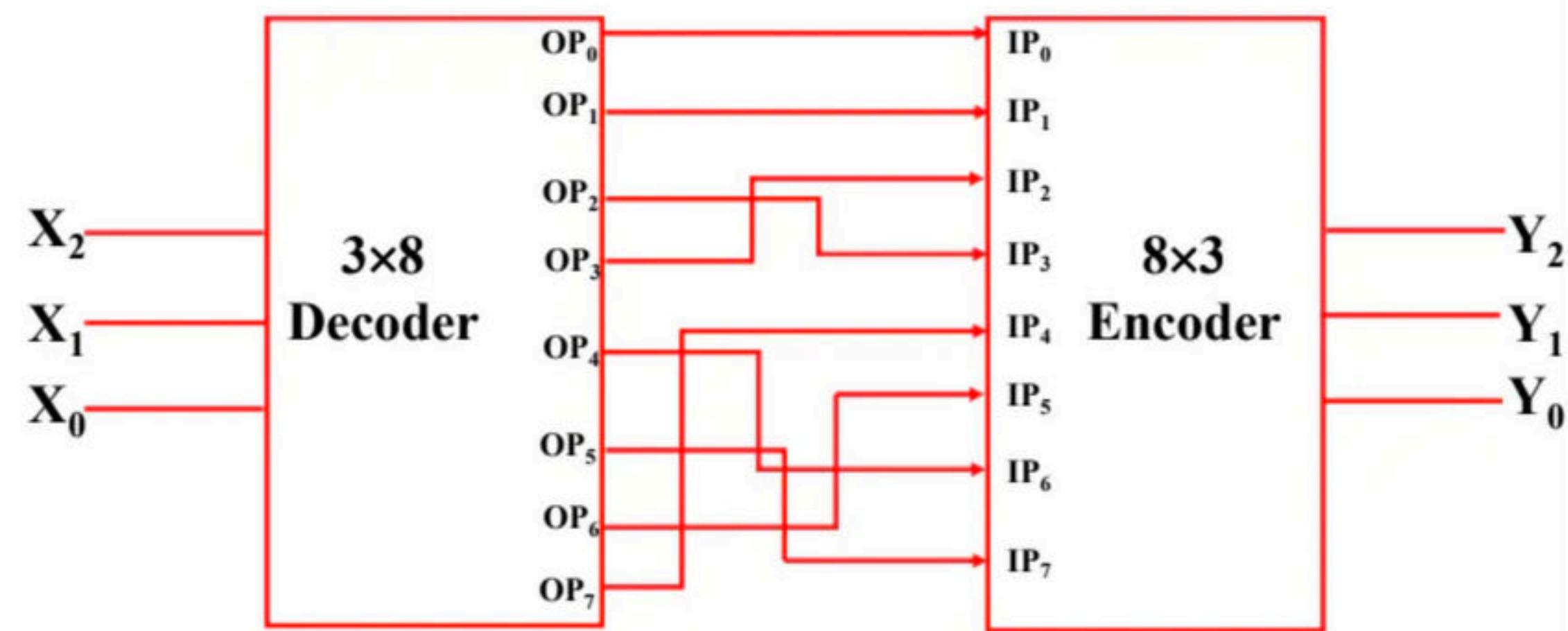
Q. Identify the circuit below

(a) Gray to binary converter

(b) Binary to excess 3 converter

(c) Binary to gray converter

(d) Excess-3 to binary converter



Code Converter

Q) Design a circuit for BCD to 7– segment display decoder

Q) Design a circuit for Binary to BCD

Q) Design a circuit for BCD to EX-3 code

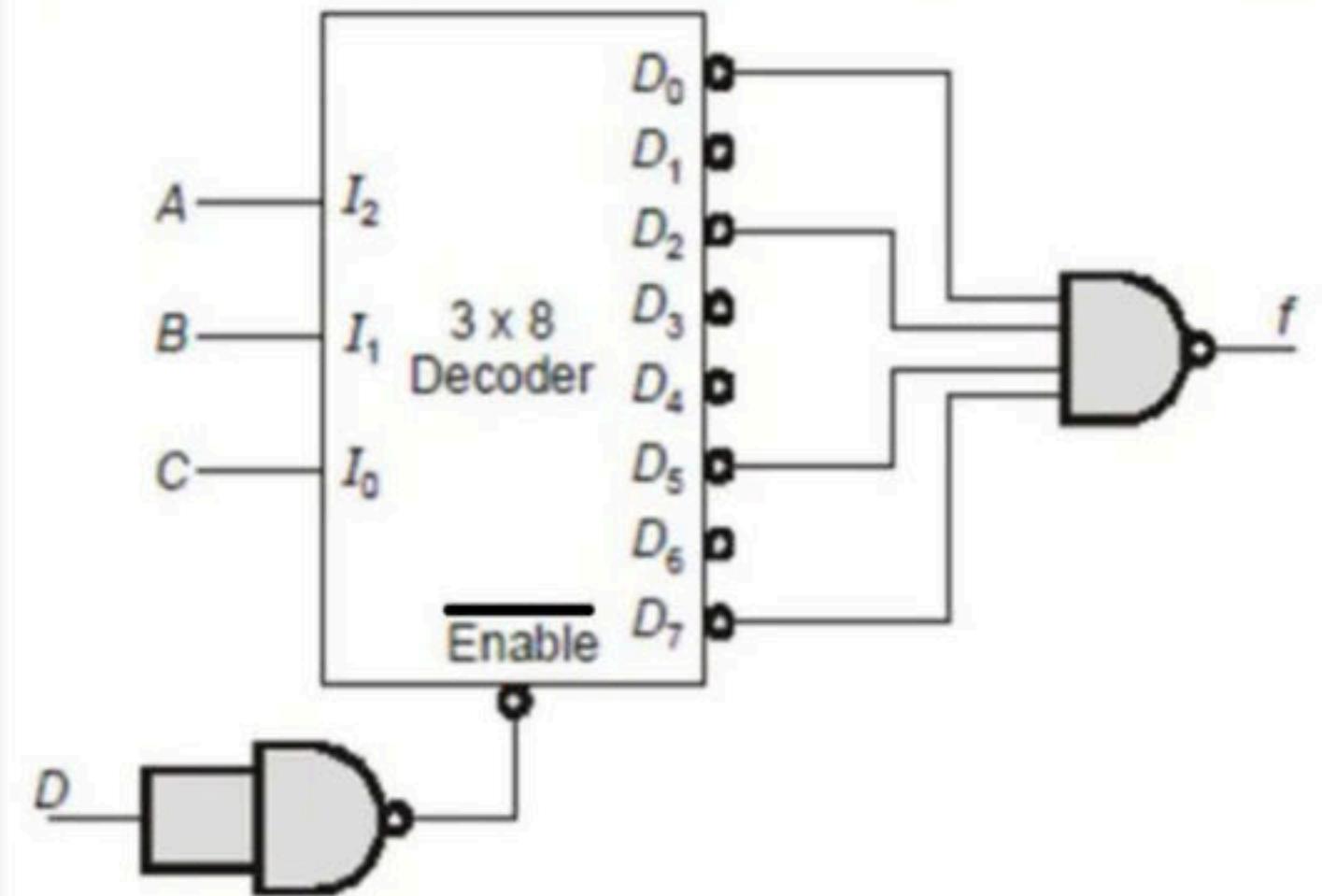
An n -bit carry look ahead adder is designed using only Ex-OR, AND, OR gates. The propagation delay of each Ex-OR gate is 20 ns and that of each AND, OR gates is t_0 ns. If the total propagation delay of the adder circuit is 60 ns, then the value of t_0 will be

(given that $t_0 \leq 20$ ns)

- a. 10
- b. 15
- c. 20
- d. depends on ' n ' value

A one bit full adder takes 75 nsec to produce sum and 50 nsec to produce carry. A 4 bit parallel adder is designed using this type of full adder. The maximum rate of additions per second can be provided by 4 bit parallel adder is $A \times 10^6$ additions/sec. The value of A is _____

The logic function $f(A, B, C, D)$ implemented by the circuit shown below is



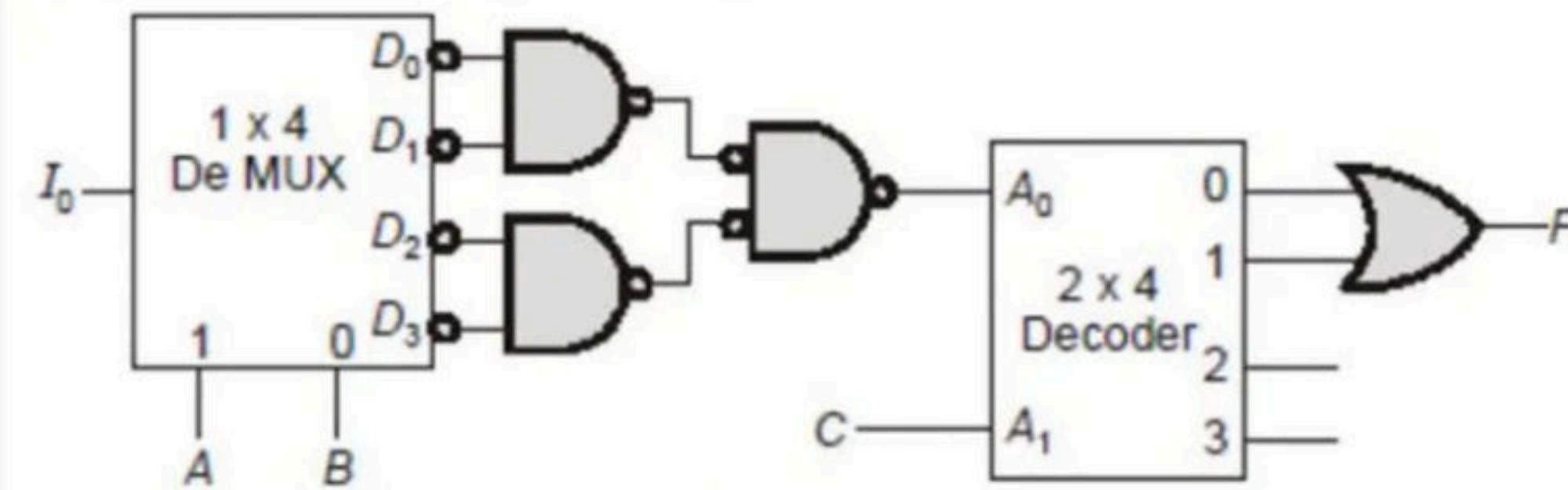
a. $\bar{D}(A \oplus C)$

b. $\bar{D}(A \odot C)$

c. $\bar{D}(A \oplus B)$

d. $D(A \odot C)$

Consider the logic circuit given below



The minimized expression for F is

- a. \bar{C}
- b. I_0
- c. C
- d. \bar{I}_0

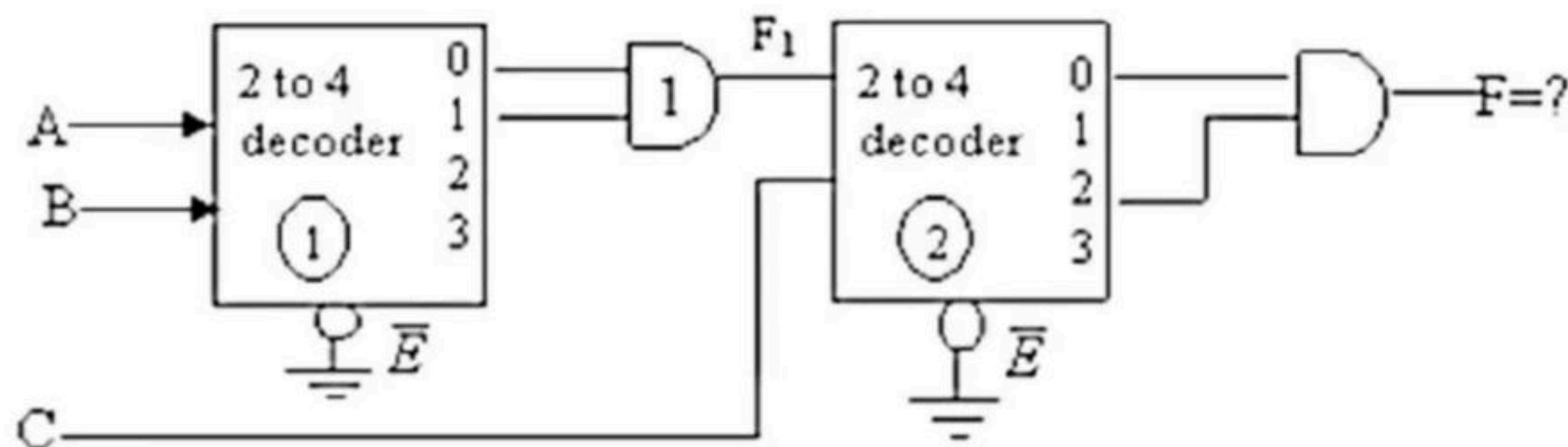
Find the output of the following circuit

(a) 1

(b) C

(c) \bar{C}

(d) 0



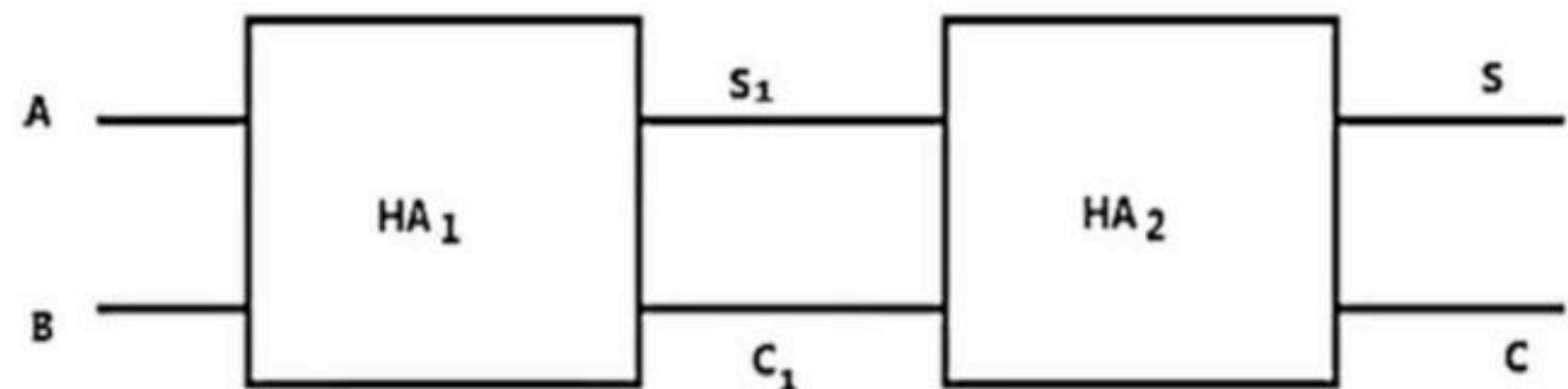
Two Half Adders are connected in cascade as shown in figure below. The output "S" and "C" are

(a) $S = A \oplus B, C = AB$

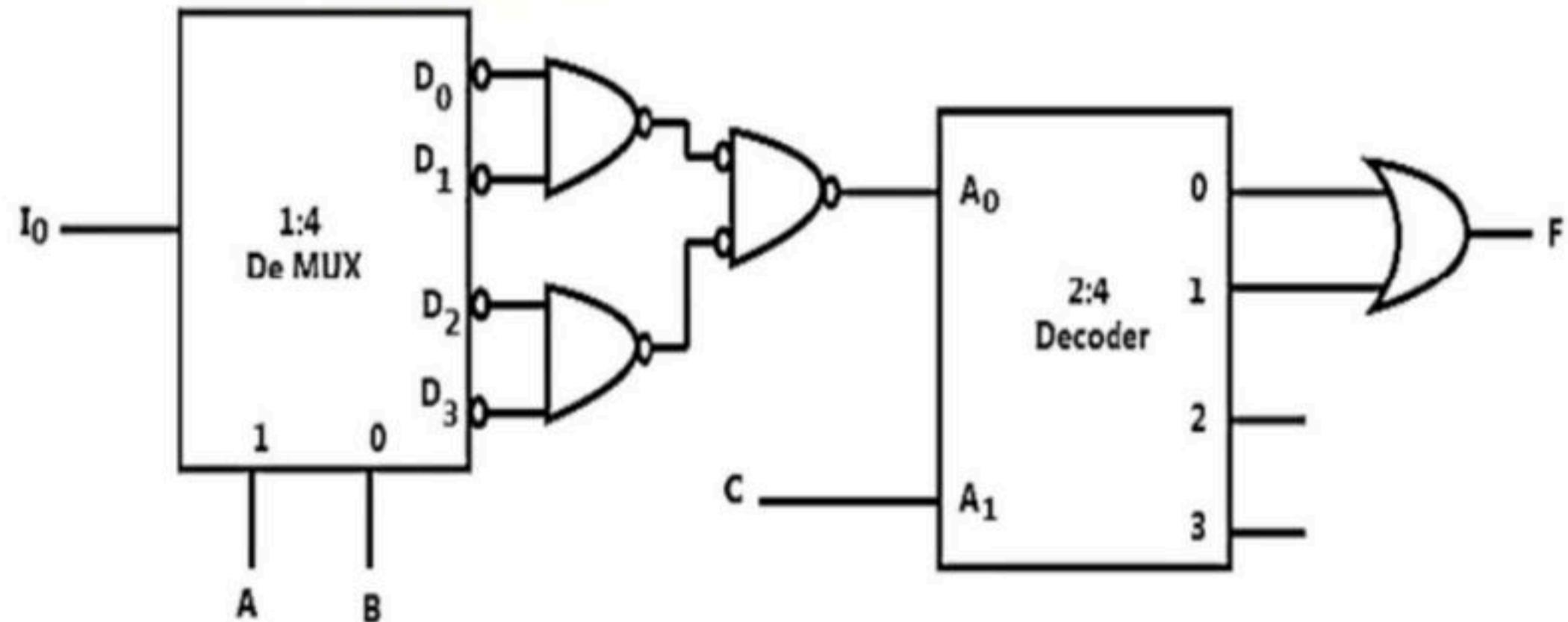
(b) $S = A \Theta B, C = 0$

(c) $S = A + B, C = 0$

(d) $S = AB, C = 0$



Consider the logic circuit given below

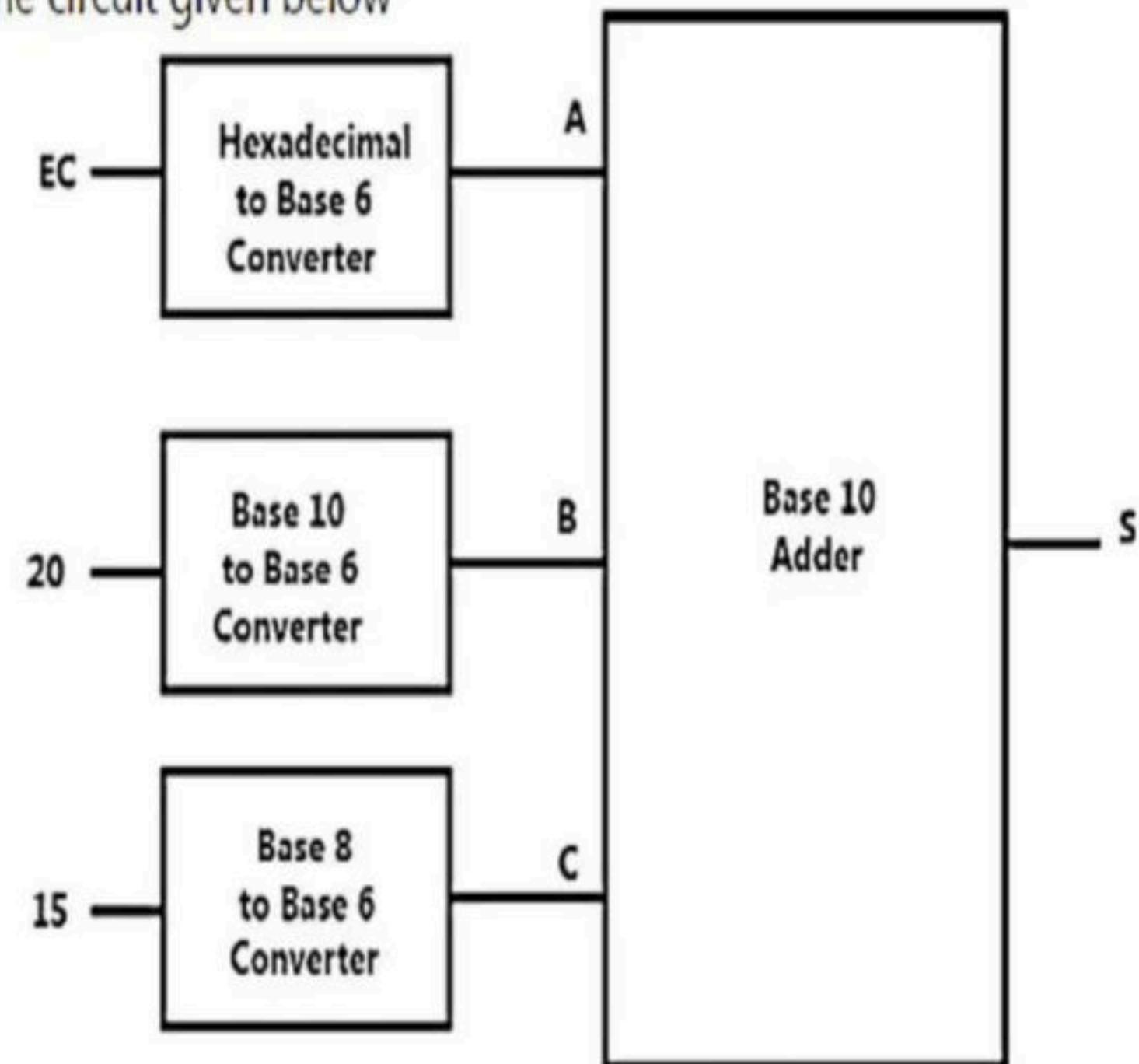


The minimized expression for F is

- (a) \bar{C}
- (c) C

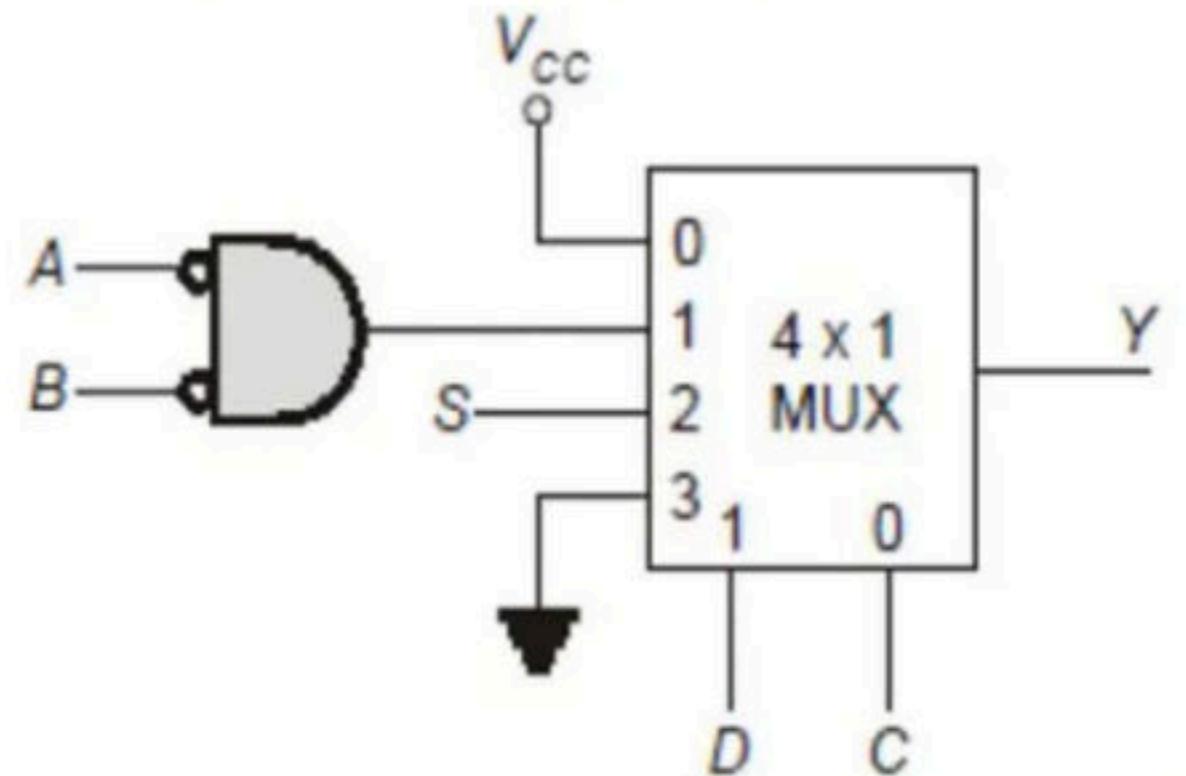
- (b) I_0
- (d) \bar{I}_0

Consider the circuit given below



The output of each converter is given to adder which adds them considering decimal number. The output of adder is S. The value of S is ____.

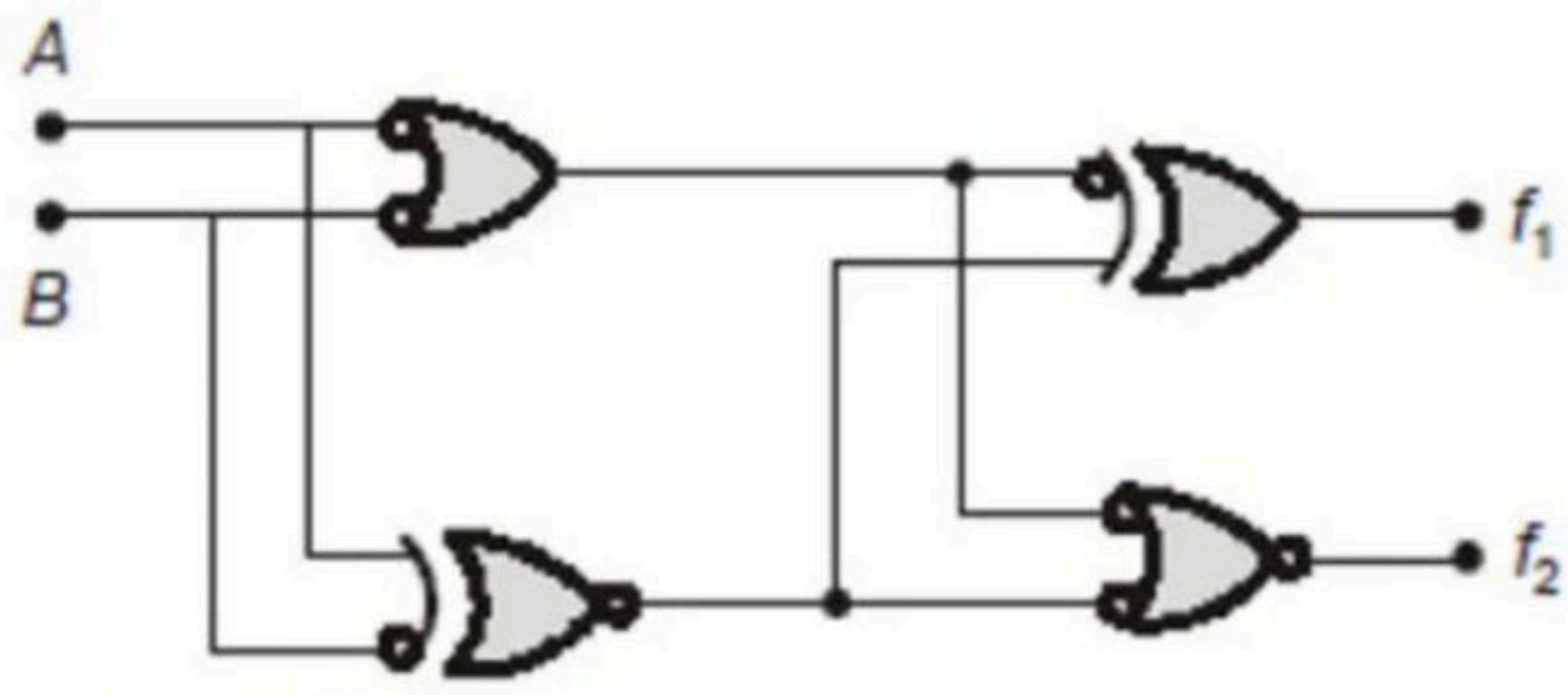
Consider the circuit given below



Which of the following statements is true for Y.

- a. $Y = \bar{C}\bar{D} + \bar{D}C(\bar{A} + \bar{B}) + \bar{C}DS$
- b. $Y = CD + D\bar{C}(\bar{A} + \bar{B}) + CDS$
- c. $Y = \bar{C}\bar{D} + (\bar{D} + C)(\bar{A} + \bar{B}) + \bar{C} + \bar{D} + \bar{S}$
- d. $Y = \bar{C}\bar{D} + (D + \bar{C})(\bar{A} + \bar{B}) + \bar{C} + \bar{D} + \bar{S}$

Consider the digital circuit shown below



It represents

- a. Half adder followed by half subtractor
- b. Half subtractor followed by half adder
- c. Half adder followed by a half adder
- d. A full adder

The minimum number of NOR gates required to realize the half adder circuit is

-----.

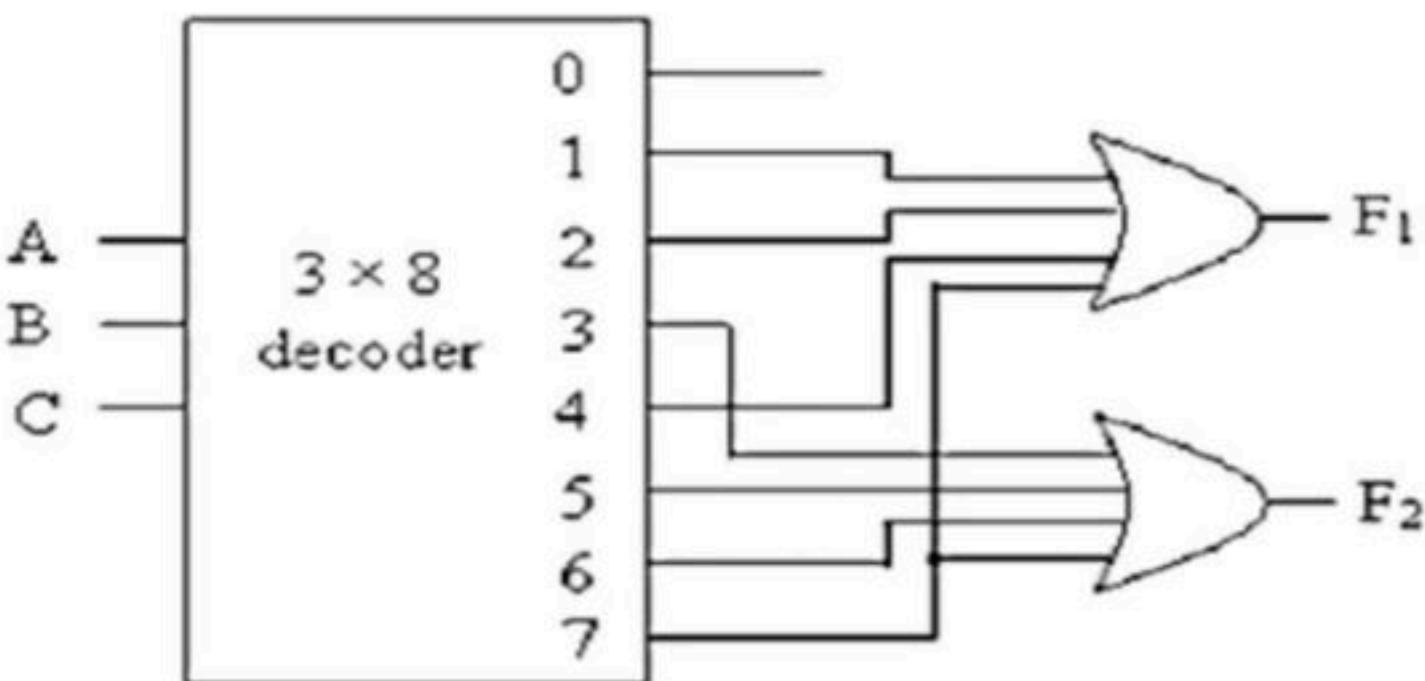
The output Y of a 2 bit comparator is logic 1 whenever the 2 bit input A is greater than the 2-bit input B . The number of combinations for which the output is logic 1 is _____.

How many 1-bit comparators, 2-input AND gates, 2-input OR gates required to design a 2-bit comparator.

- (a) 2, 3, 2
- (b) 2, 2, 3
- (c) 2, 3, 3
- (d) 2, 2, 2

What is the name of given circuit?

- (a) Full Subtractor
- (b) Full Adder
- (c) 3-bit even parity generator
- (d) 3-bit odd parity generator



In a 2-bit magnitude comparator circuit ($A = A_1A_0$, $B = B_1B_0$), the expression for $A > B$ & $A < B$ is

(a) $A > B = \bar{A}_1 \bar{B}_1 + (A_1 \oplus B_1) A_0 \bar{B}_0$

$A < B = \bar{A}_1 \bar{B}_1 + (A_1 \oplus B_1) \bar{A}_0 B_0$

(c) $A > B = A_1 B_1 + (A_1 \oplus B_1) A_0 \bar{B}_0$

$A < B = A_1 B_1 + (A_1 \oplus B_1) \bar{A}_0 B_0$

(b) $A > B = A_1 \bar{B}_1 + (A_1 \oplus B_1) A_0 \bar{B}_0$

$A < B = \bar{A}_1 B_1 + (A_1 \oplus B_1) \bar{A}_0 B_0$

(d) $A > B = A_1 B_1 + (A_1 \oplus B_1) A_0 \bar{B}_0$

$A < B = A_1 B_1 + (A_1 \oplus B_1) \bar{A}_0 B_0$

An 8×1 multiplexer has inputs A, B and C connected to the selection input S_2 , S_1 , and S_0 , respectively. The data inputs I_0 through I_7 are as follows:

$$I_1 = I_2 = I_7 = 0; I_3 = I_5 = 1; I_0 = I_4 = D; \text{ and } I_6 = \bar{D};$$

The Boolean function that the multiplexer implements is

(a) $Y = \sum m(1, 6, 7, 9, 10, 11, 13)$

(c) $Y = \sum m(4, 5, 7, 8, 9, 11, 15)$

(b) $Y = \sum m(1, 6, 7, 9, 10, 11, 12)$

(d) $Y = \sum m(0, 1, 3, 4, 5, 7, 9, 11)$

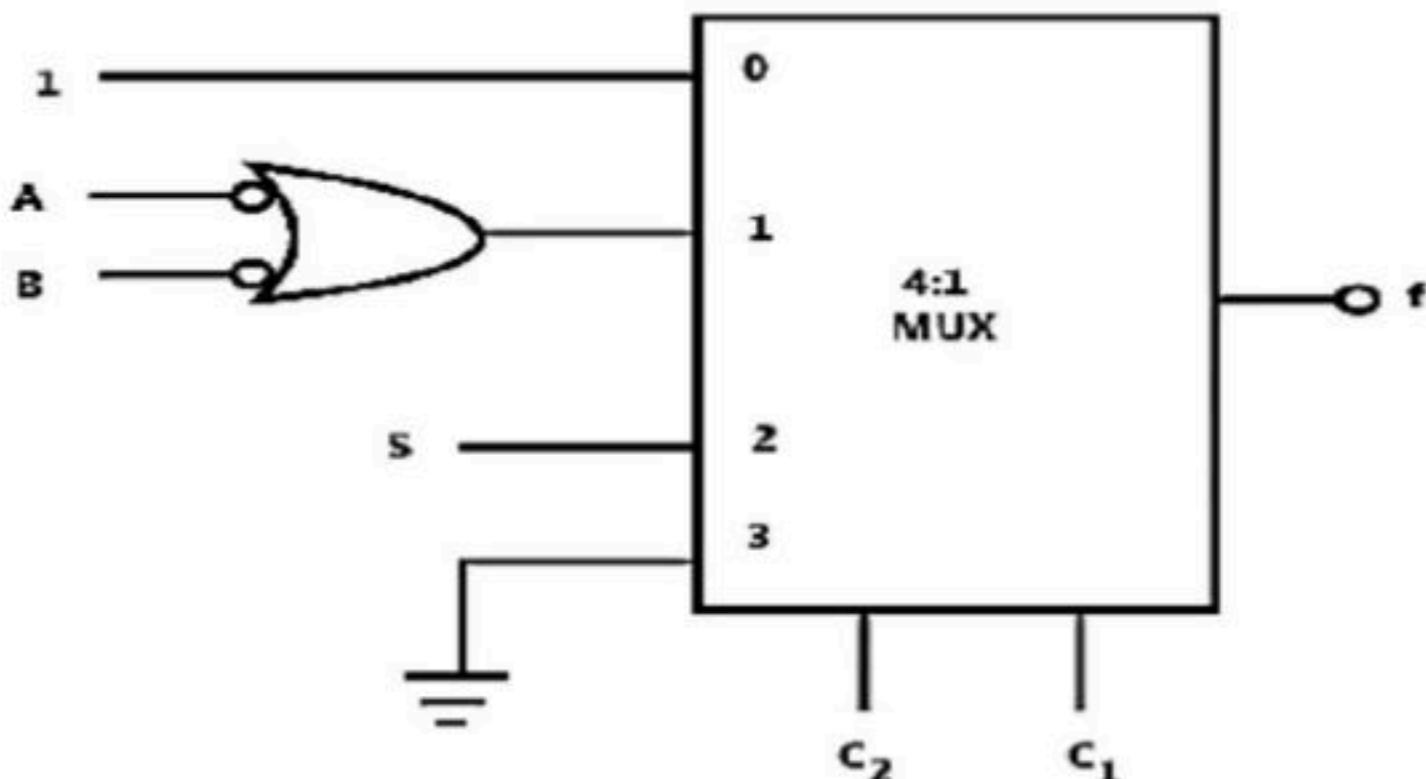
In the following MUX, find the output f.

(a) $C_2 \cdot \bar{C}_1 S + \bar{C}_2 C_1 (\bar{A} + \bar{B})$

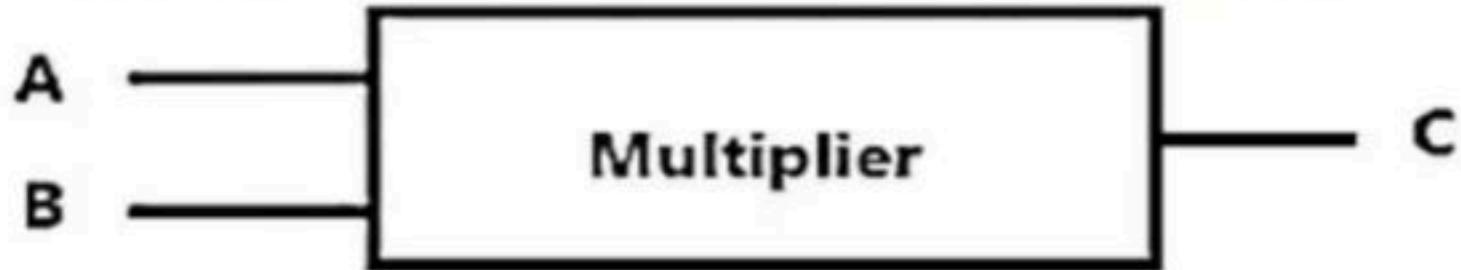
(b) $\bar{C}_2 \bar{C}_1 + C_2 C_1 + \bar{C}_2 \bar{C}_1 S + \bar{C}_2 C_1 \bar{A} \bar{B}$

(c) $\bar{A} \bar{B} + S$

(d) $\bar{C}_2 \bar{C}_1 + C_2 \bar{C}_1 S + \bar{C}_2 C_1 (\bar{A} \bar{B})$



Consider a 3-bit number A and 2 bit number B are given to a multiplier. The output of multiplier is realized using AND gate and one bit full adders. If minimum number of AND gates required are X and one bit full adders required are Y, then $X + Y = \underline{\hspace{2cm}}$.



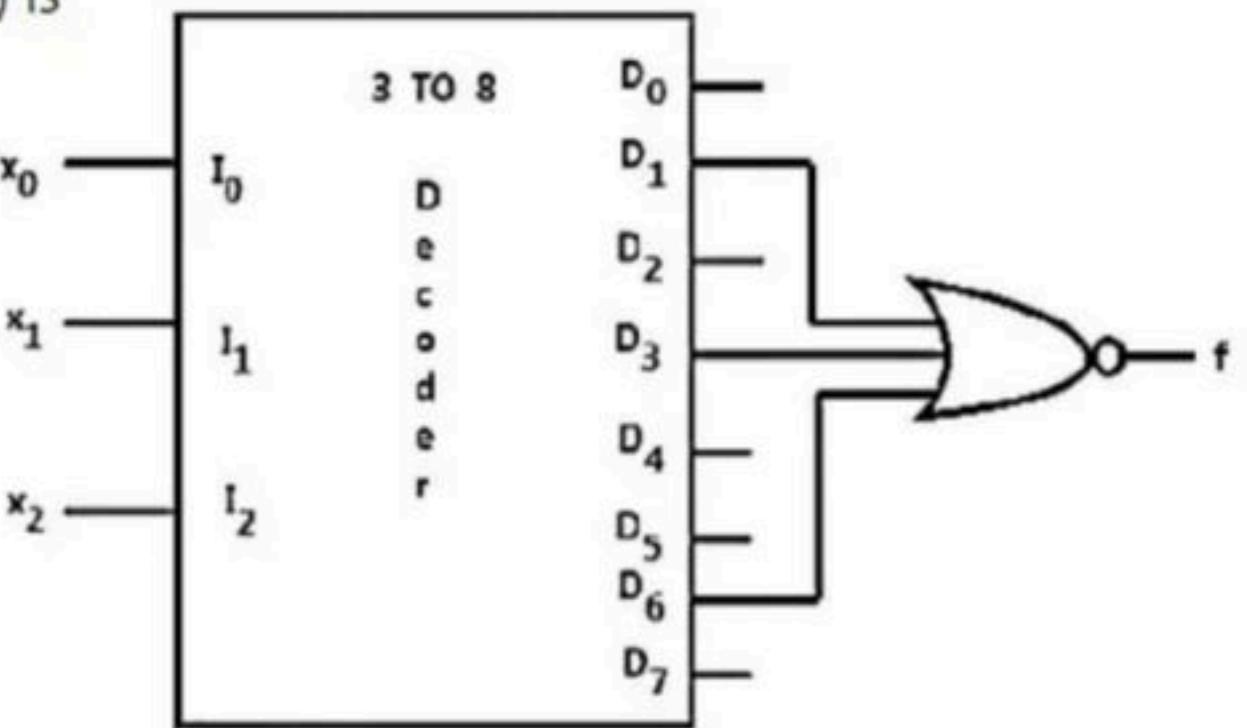
In the following circuit the function $f(x_2, x_1, x_0)$ is

(a) $\prod M(0, 2, 4, 5)$

(b) $\sum m(0, 2, 4, 5, 7)$

(c) $\sum m(1, 3, 6)$

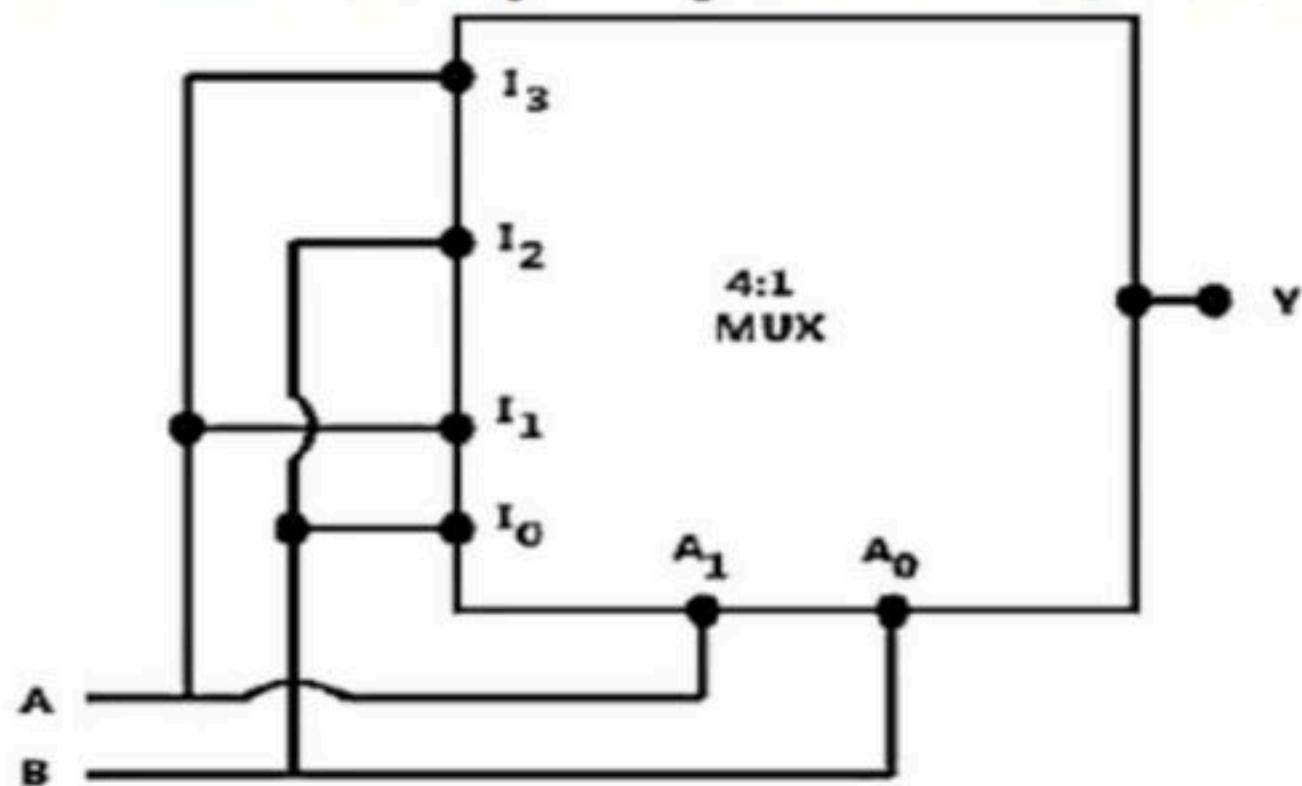
(d) $\sum M(1, 3, 6)$



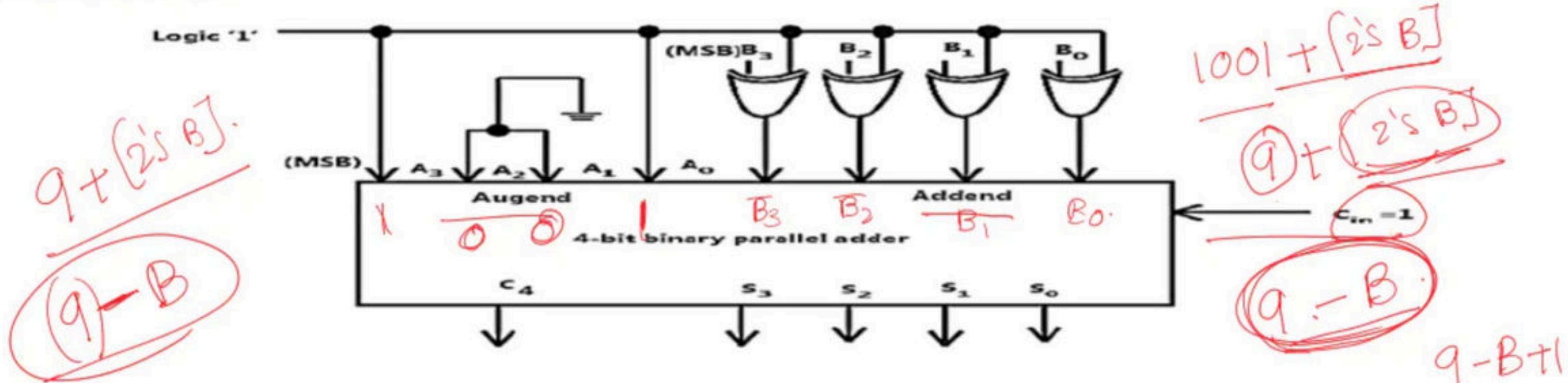
A combinational logic circuit has three inputs A, B and C and one output Y. The output $Y = 1$ when at least two inputs are 1. Otherwise, $Y = 0$. In its minimized SOP realization, the maximum number of two input terms is _____.

A gate having two inputs (A , B) and one output (Y) is implemented using 4 : 1 MUX as shown in figure below. A_1 (MSB) and A_0 are the control bits and I_0 to I_3 are the inputs to the MUX. The gate is

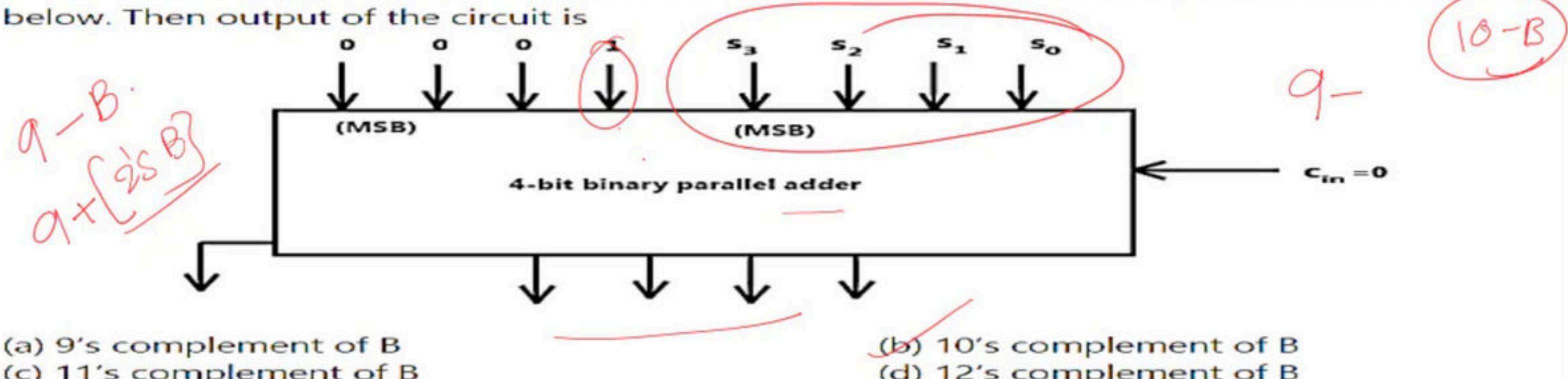
- (a) AND
- (b) NOR
- (c) OR
- (d) EX-OR



Consider the digital circuit shown below. A single digit decimal number(B) is converted into its 4 bit binary equivalent($B_3B_2B_1B_0$) and then applied to the addend bits of the adder as shown below:



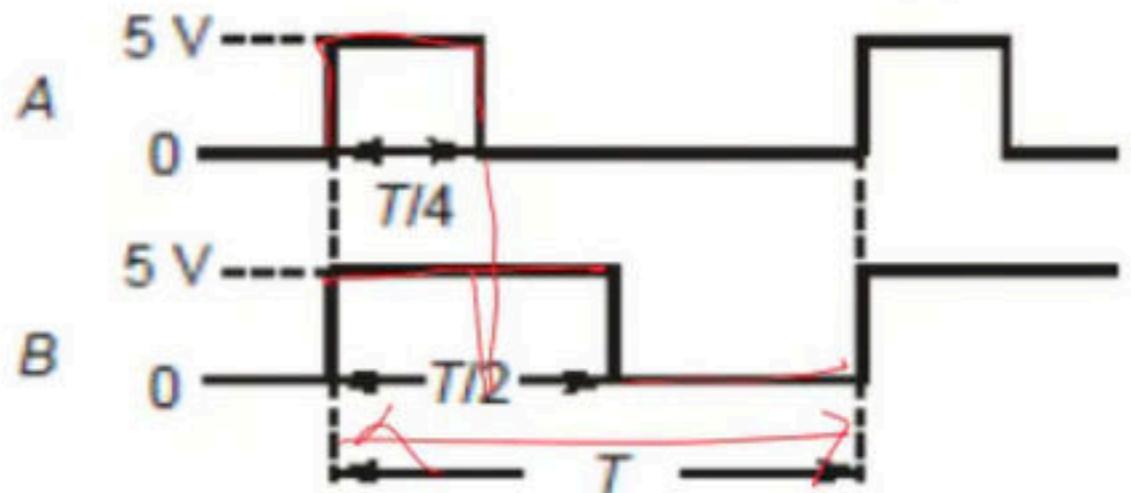
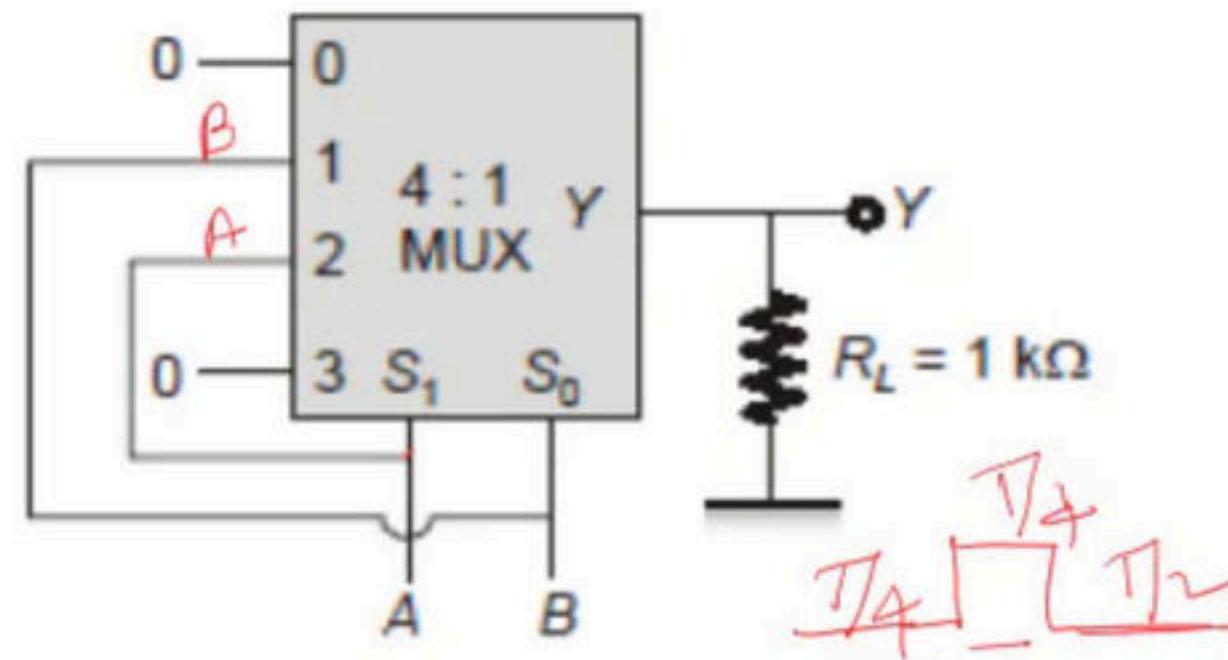
If $C_4 = 1$ and $S_3S_2S_1S_0$ are given to the addend bits of the 4-bit binary parallel adder as shown below. Then output of the circuit is



- (a) 9's complement of B
- (c) 11's complement of B

- (b) 10's complement of B
- (d) 12's complement of B

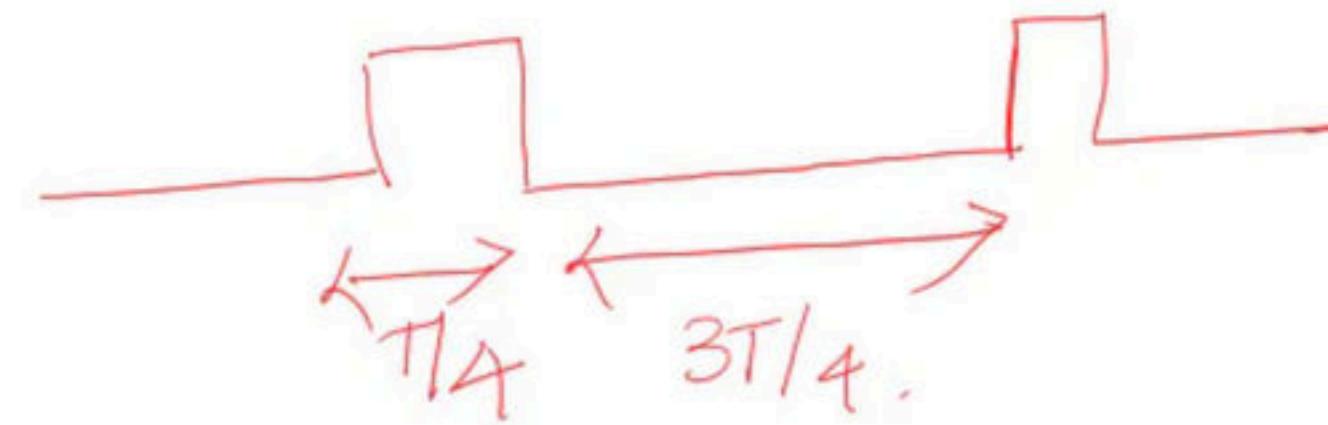
Consider the 4 : 1 MUX based circuit as shown in the figure. A and B are two periodic signals with duty cycles 25% and 50% respectively as given in the figure. If +5 V and 0 V are used to represent logic-1 and logic-0 respectively, then the average power dissipated by the resistor R_L will be _____ mW.



$$Y = \overline{AB}B + \overline{A}\overline{B}A$$

$$Y = \overline{AB} + \overline{A}\overline{B}$$

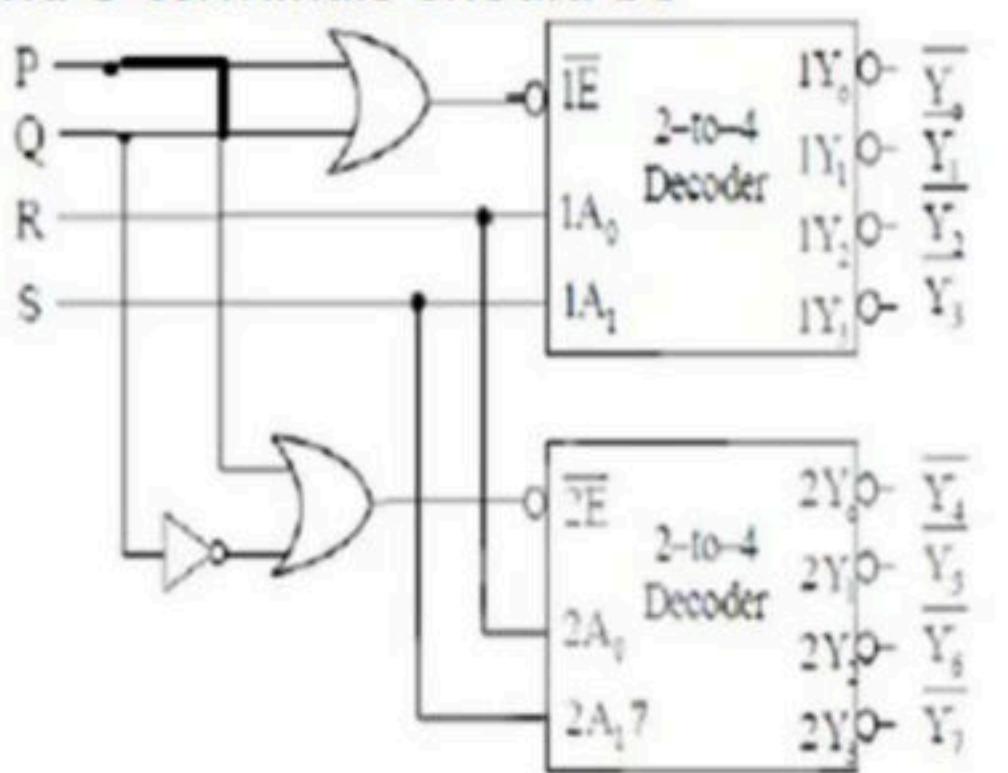
$$Y = A \oplus B =$$



$$D = \frac{T/4}{T} = \frac{1}{4}$$

$$D = \frac{T/4}{T}$$

A 1-to-8 demultiplexer with data input D_{in} , address inputs S_0, S_1, S_2 (with S_0 as the LSB) and \bar{Y}_0 to \bar{Y}_7 as the eight demultiplexed output, is to be designed using two 2-to-4 decoders (with enable input \bar{E} and address input A_0 and A_1) as shown in the figure. D_{in}, S_0, S_1 and S_2 are to be connected to P, Q, R and S, but not necessarily in this order. The respective input connections to P, Q, R and S terminals should be



- A. $\underline{S_2}, \bar{D}_{in}, S_0, S_1$
- B. $S_1, \bar{D}_{in}, S_0, S_2$
- C. $\bar{D}_{in}, S_0, S_1, S_2$
- D. $\bar{D}_{in}, S_2, S_1, S_0$