

# Digital Electronics (4321102) - Summer 2023 Solution (Gujarati)

Milav Dabgar

August 07, 2023

## પ્રશ્ન 1(a) [3 ગુણ]

બુલિયન એલ્જીબ્રા માટેના ડે-મોર્ગનના નિયમ સમજાવો

જવાબ

ડે-મોર્ગનના નિયમમાં બે કાયદા છે જે AND, OR અને NOT ક્રિયાઓ વચ્ચેના સંબંધને દર્શાવે છે:

કાયદો 1: સરવાળાના પૂરકની કિંમત પૂરકના ગુણાકાર બરાબર હોય છે

$$\overline{A + B} = \overline{A} \cdot \overline{B}$$

કાયદો 2: ગુણાકારના પૂરકની કિંમત પૂરકના સરવાળા બરાબર હોય છે

$$\overline{A \cdot B} = \overline{A} + \overline{B}$$

કોષ્ટક 1. ડે-મોર્ગનના નિયમની ચકાસણી

A	B	A+B	$\overline{A+B}$	$\overline{A}$	$\overline{B}$	$\overline{A} \cdot \overline{B}$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

મેમરી ટ્રીક

"OR પર NOT થાય AND, AND પર NOT થાય OR"

## પ્રશ્ન 1(બ) [4 ગુણ]

નીચેના ડેસિમલ નંબરને બાયનરી અને ઓક્ટલ નંબરમાં ફેરવો (i) 215 (ii) 59

જવાબ

બાયનરી રૂપાંતર:

215 માટે:

- 2 વડે ભાગ કરો:  $215/2 = 107$  શેષ 1
- $107/2 = 53$  શેષ 1
- $53/2 = 26$  શેષ 1
- $26/2 = 13$  શેષ 0
- $13/2 = 6$  શેષ 1
- $6/2 = 3$  શેષ 0

- $3/2 = 1$  શેષ 1
- $1/2 = 0$  શેષ 1
- તેથી,  $(215)_{10} = (11010111)_2$

59 માટે:

- 2 વડે ભાગ કરો:  $59/2 = 29$  શેષ 1
- $29/2 = 14$  શેષ 1
- $14/2 = 7$  શેષ 0
- $7/2 = 3$  શેષ 1
- $3/2 = 1$  શેષ 1
- $1/2 = 0$  શેષ 1
- તેથી,  $(59)_{10} = (111011)_2$

ઓક્ટલ રૂપાંતર:

215 માટે:

- 8 વડે ભાગ કરો:  $215/8 = 26$  શેષ 7
- $26/8 = 3$  શેષ 2
- $3/8 = 0$  શેષ 3
- તેથી,  $(215)_{10} = (327)_8$

59 માટે:

- 8 વડે ભાગ કરો:  $59/8 = 7$  શેષ 3
- $7/8 = 0$  શેષ 7
- તેથી,  $(59)_{10} = (73)_8$

**કોષ્ટક 2.** સંખ્યા રૂપાંતર સારાંશ

ડેસિમલ	બાયનરી	ઓક્ટલ
215	11010111	327
59	111011	73

મેમરી ટ્રીક

"આધાર વડે ભાગો, શેષ નીચેથી ઉપર વાંચો"

## પ્રશ્ન 1(ક)(I) [2 ગુણ]

ડેસિમલ, બાયનરી, ઓક્ટલ અને હેક્ઝાડેસિમલ નંબર સિસ્ટમનો બેઝ લખો

જવાબ

**કોષ્ટક 3.** સંખ્યા પદ્ધતિના આધાર

સંખ્યા પદ્ધતિ	આધાર
ડેસિમલ	10
બાયનરી	2
ઓક્ટલ	8
હેક્ઝાડેસિમલ	16

મેમરી ટ્રીક

"ડે-બા-ઓ-હે: 10-2-8-16"

## પ્રશ્ન 1(ક)(II) [2 ગુણ]

$$(147)_{10} = (\underline{\hspace{2cm}})_2 = (\underline{\hspace{2cm}})_{16}$$

## જવાબ

## ડેસિમલથી બાયનરી રૂપાંતર:

- $147/2 = 73$  શેષ 1
- $73/2 = 36$  શેષ 1
- $36/2 = 18$  શેષ 0
- $18/2 = 9$  શેષ 0
- $9/2 = 4$  શેષ 1
- $4/2 = 2$  શેષ 0
- $2/2 = 1$  શેષ 0
- $1/2 = 0$  શેષ 1
- તેથી,  $(147)_{10} = (10010011)_2$

## ડેસિમલથી હેક્ઝાડેસિમલ રૂપાંતર:

- બાયનરી અંકોને 4ના સમૂહમાં વિભાજિત કરો: 1001 0011
- દરેક સમૂહને હેક્ઝાડેસિમલ રૂપાંતરિત કરો:  $1001 = 9$ ,  $0011 = 3$
- તેથી,  $(147)_{10} = (93)_{16}$

## કોષ્ટક 4. રૂપાંતર પરિણામ

ડેસિમલ	બાયનરી	હેક્ઝાડેસિમલ
147	10010011	93

## મેમરી ટ્રીક

"હેક્ઝાડેસિમલ માટે જમણેથી 4ના સમૂહમાં વિભાજિત કરો"

## પ્રશ્ન 1(ક)(III) [3 ગુણ]

નીચેના બાયનરી કોડનું ગ્રે કોડમાં રૂપાંતર કરો (i) 1011 (ii) 1110

## જવાબ

## બાયનરીથી ગ્રે કોડ રૂપાંતર પ્રક્રિયા:

1. ગ્રે કોડનો MSB (ડાબી બાજુનો બિટ) બાયનરી કોડના MSB જેવો જ હોય છે
2. ગ્રે કોડના અન્ય બિટ્સ બાયનરી કોડના આસપાસના બિટ્સને XOR કરીને મેળવવામાં આવે છે

## 1011 માટે:

- ગ્રે કોડનો MSB = બાયનરી કોડનો MSB = 1
- બીજો બિટ =  $1 \oplus 0 = 1$
- ત્રીજો બિટ =  $0 \oplus 1 = 1$
- ચોથો બિટ =  $1 \oplus 1 = 0$
- તેથી,  $(1011)_2 = (1110)_{gray}$

## 1110 માટે:

- ગ્રે કોડનો MSB = બાયનરી કોડનો MSB = 1
- બીજો બિટ =  $1 \oplus 1 = 0$
- ત્રીજો બિટ =  $1 \oplus 1 = 0$
- ચોથો બિટ =  $1 \oplus 0 = 1$
- તેથી,  $(1110)_2 = (1001)_{gray}$

## કોષ્ટક 5. બાયનરીથી ગ્રે કોડ રૂપાંતર

બાયનરી	રૂપાંતર પદ્ધતિ	ગ્રે કોડ
1011	$1, 1 \oplus 0 = 1, 0 \oplus 1 = 1, 1 \oplus 1 = 0$	1110
1110	$1, 1 \oplus 1 = 0, 1 \oplus 1 = 0, 1 \oplus 0 = 1$	1001

## મેમરી ટ્રીક

"પહેલો રાખો, બાકીના XOR કરો"

## પ્રશ્ન 1 [OR] (I) [2 ગુણ]

BCD અને ASCII નું ફૂલફોર્મ લખો

## જવાબ

કોષ્ટક 6. BCD અને ASCII નું પૂર્ણ નામ

સંક્ષિપ્ત રૂપ	પૂર્ણ નામ
BCD	Binary Coded Decimal
ASCII	American Standard Code for Information Interchange

## મેમરી ટ્રીક

"બાયનરી કોડેડ ડેસિમલ, અમેરિકન સ્ટાન્ડર્ડ કોડ ફોર ઇન્ફોર્મેશન ઇન્ટરચેન્જ"

## પ્રશ્ન 1 [OR] (II) [2 ગુણ]

નીચેના બાયનરી નંબરના 1's અને 2's કોમ્પ્લિમેન્ટ શોધો (i) 1010 (ii) 1011

## જવાબ

1's કોમ્પ્લિમેન્ટ: બધા બિટ્સ ઉલટાવો (0 ને 1 અને 1 ને 0 માં બદલો)

2's કોમ્પ્લિમેન્ટ: 1's કોમ્પ્લિમેન્ટ લો અને 1 ઉમેરો

1010 માટે:

- 1's કોમ્પ્લિમેન્ટ: 0101
- 2's કોમ્પ્લિમેન્ટ:  $0101 + 1 = 0110$

1011 માટે:

- 1's કોમ્પ્લિમેન્ટ: 0100
- 2's કોમ્પ્લિમેન્ટ:  $0100 + 1 = 0101$

કોષ્ટક 7. કોમ્પ્લિમેન્ટ પરિણામો

બાયનરી	1's કોમ્પ્લિમેન્ટ	2's કોમ્પ્લિમેન્ટ
1010	0101	0110
1011	0100	0101

## મેમરી ટ્રીક

"1's માટે બધા બિટ ઉલટાવો, 2's માટે એક ઉમેરો"

## પ્રશ્ન 1 [OR] (III) [3 ગુણ]

2's કોમ્પ્લિમેન્ટ મેથડથી બાદબાકી કરો (i)  $(110110)_2 - (101010)_2$

### જવાબ

2's કોમ્પ્લિમેન્ટ પદ્ધતિથી બાદબાકી માટે:

1. બાદ થનાર સંખ્યાનો 2's કોમ્પ્લિમેન્ટ શોધો
2. તેને મૂળ સંખ્યામાં ઉમેરો
3. બિટ વિડ્થની બહારના કેરીને છોડી દો

**બાદબાકી:**  $(110110)_2 - (101010)_2$

**પગલું 1:** 101010 નો 2's કોમ્પ્લિમેન્ટ શોધો

- 101010 નો 1's કોમ્પ્લિમેન્ટ = 010101
- 2's કોમ્પ્લિમેન્ટ = 010101 + 1 = 010110

**પગલું 2:** 110110 + 010110 ઉમેરો

```

1 1 1 1 1
1 1 0 1 1 0
+ 0 1 0 1 1 0
-----
0 0 1 1 0 0

```

**પગલું 3:** પરિણામ 001100 =  $(12)_{10}$

### કોષ્ટક 8. બાદબાકી પ્રક્રિયા

પગલું	ક્રિયા	પરિણામ
1	101010 નો 2's કોમ્પ્લિમેન્ટ	010110
2	110110 + 010110 ઉમેરો	001100
3	અંતિમ પરિણામ (ડેસિમલ)	12

### મેમરી ટ્રીક

"બાદનારનો કોમ્પ્લિમેન્ટ લો, ઉમેરો, કેરી ભૂલી જાઓ"

## પ્રશ્ન 2(અ) [3 ગુણ]

NAND ગેટનો જ ઉપયોગ કરી AND, OR અને NOT ગેટની લૉજિક સર્કિટ બનાવો

### જવાબ

**AND ગેટ NAND ગેટથી:**

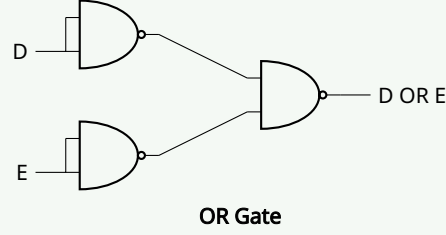
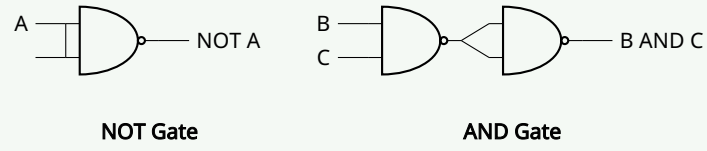
- AND ગેટ = NAND ગેટ પછી NOT ગેટ (NAND ગેટ)

**OR ગેટ NAND ગેટથી:**

- OR ગેટ = બંને ઇનપુટને NOT (NAND ગેટ) લાગુ કરો, પછી તે પરિણામોને NAND કરો

**NOT ગેટ NAND ગેટથી:**

- NOT ગેટ = NAND ગેટ જેમાં બંને ઇનપુટ જોડાયેલા હોય



આકૃતિ 1. Universal Gates implementation

**મેમરી ટ્રીક**

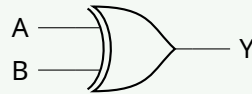
"NOT માટે એક NAND, AND માટે બે NAND, OR માટે ત્રણ NAND"

**પ્રશ્ન 2(બ) [4 ગુણ]**

નીચેના લૉજિક ગેટનો લૉજિક સિમ્બોલ, ટ્રુથ ટેબલ અને સમીકરણ લખો/દોરો (i) XOR ગેટ (ii) OR ગેટ

**જવાબ**

XOR ગેટ:  
લૉજિક સિમ્બોલ:

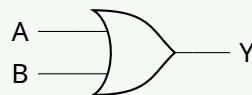


ટ્રુથ ટેબલ:

A	B	Y ( $A \oplus B$ )
0	0	0
0	1	1
1	0	1
1	1	0

બુલિયન સમીકરણ:  $Y = A \oplus B = A'B + AB'$

OR ગેટ:  
લૉજિક સિમ્બોલ:



ટ્રુથ ટેબલ:

A	B	Y (A+B)
0	0	0
0	1	1
1	0	1
1	1	1

બુલિયન સમીકરણ:  $Y = A + B$

### મેમરી ટ્રીક

"XOR: એકસકલુસિવ OR - એક અથવા બીજું પણ બંને નહીં; OR: એક અથવા બીજું અથવા બંને"

## પ્રશ્ન 2(ક)(I) [3 ગુણ]

બુલિયન સમીકરણ  $Y = A + B[AC + (B + \overline{C})D]$  ને algebraic મેથડથી સરળ બનાવો

### જવાબ

પગલાંવાર સરળીકરણ:

$$Y = A + B[AC + (B + \overline{C})D]$$

$$Y = A + B[AC + BD + \overline{C}D]$$

$$Y = A + BAC + BBD + B\overline{C}D$$

$$Y = A + ABC + BD + B\overline{C}D$$

(કારણ કે  $BB = B$ )

$$Y = A + AC + BD + B\overline{C}D$$

(અવશોષણ:  $A + AB = A$ )

$$Y = A + BD + B\overline{C}D$$

(અવશોષણ:  $A + AC = A$ )

$$Y = A + BD(1 + \overline{C})$$

$$Y = A + BD$$

(કારણ કે  $1 + X = 1$ )

અંતિમ સમીકરણ:  $Y = A + BD$

### કોષ્ટક 9. સરળીકરણ પગલાં

પગલું	સમીકરણ	લાગુ પડેલ નિયમ
1	$A + B[AC + (B + \overline{C})D]$	મૂળ
2	$A + B[AC + BD + \overline{C}D]$	વિતરણ
3	$A + BAC + BBD + B\overline{C}D$	વિતરણ
4	$A + ABC + BD + B\overline{C}D$	આઇડેમ્પોટન્સ ( $BB = B$ )
5	$A + AC + BD + B\overline{C}D$	અવશોષણ
6	$A + BD + B\overline{C}D$	અવશોષણ ( $A + AC = A$ )
7	$A + BD(1 + \overline{C})$	ફેક્ટરિંગ
8	$A + BD$	પૂરક નિયમ

### મેમરી ટ્રીક

"આઇડેમ્પોટન્સ, અવશોષણ, અને પૂરક પેટર્ન માટે હંમેશા તપાસો"

## પ્રશ્ન 2(ક)(II) [4 ગુણ]

બુલિયન સમીકરણ  $F(A,B,C) = \sum m(0, 2, 3, 4, 5, 6)$  ને Karnaugh Map ની મદદથી સરળ બનાવો

**જવાબ**

$F(A,B,C) = \sum m(0, 2, 3, 4, 5, 6)$  માટે K-map બનાવો:

		BC			
		00	01	11	10
A	0	1	0	0	1
	1	1	1	0	1

1 ની ગ્રુપિંગ કરો:

- ગ્રુપ 1:  $m(0,4)$  -  $B'C'$  સાથે સંબંધિત
- ગ્રુપ 2:  $m(2,6)$  -  $BC'$  સાથે સંબંધિત
- ગ્રુપ 3:  $m(4,5)$  -  $AB'$  સાથે સંબંધિત

સરળ સમીકરણ:  $F(A, B, C) = B'C' + BC' + AB'$

વધુ સરળ કરીએ:  $F = C'(B' + B) + AB' = C' + AB'$

**મેમરી ટ્રીક**

"2ની પાવરમાં આસપાસના 1 ને ગ્રુપ કરો"

## પ્રશ્ન 2 [OR] (અ) [3 ગુણ]

NOR ગેટનો જ ઉપયોગ કરી AND, OR અને NOT ગેટની લૉજિક સર્કિટ બનાવો

**જવાબ**

NOT ગેટ NOR ગેટથી:

- NOT ગેટ = NOR ગેટ જેમાં બંને ઇનપુટ જોડાયેલા હોય

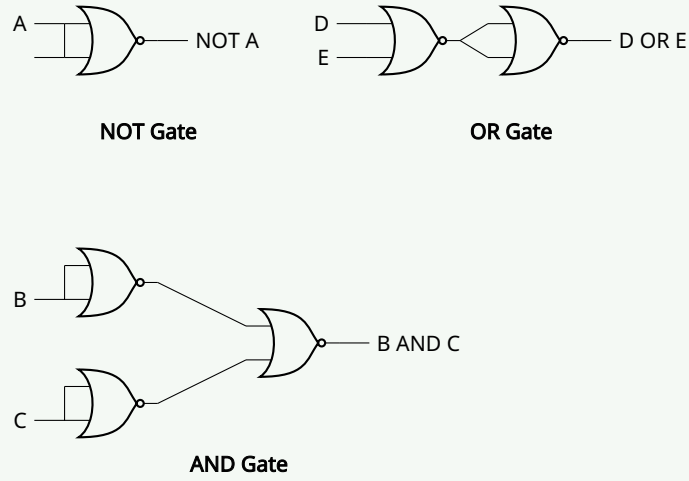
AND ગેટ NOR ગેટથી:

- AND ગેટ = બંને ઇનપુટને NOT (NOR ગેટ) લાગુ કરો, પછી તે પરિણામોને ફરીથી NOR કરો

OR ગેટ NOR ગેટથી:

- OR ગેટ = NOR ગેટ પછી NOT ગેટ (NOR ગેટ)





આકૃતિ 2. Universal Gates implementation (NOR)

## મેમરી ટ્રીક

"NOT માટે એક NOR, દરેકને NOT કરીને NOR કરો AND માટે, બે વાર NOR કરો OR માટે"

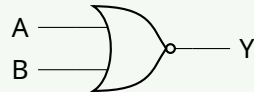
## પ્રશ્ન 2 [OR] (બ) [4 ગુણ]

નીચેના લૉજિક ગેટનો લૉજિક સિમ્બોલ, ટ્રુથ ટેબલ અને સમીકરણ લખો/દોરો (i) NOR ગેટ (ii) AND ગેટ

## જવાબ

NOR ગેટ:

લૉજિક સિમ્બોલ:



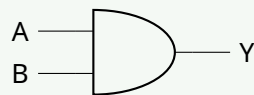
ટ્રુથ ટેબલ:

A	B	Y (A+B)'
0	0	1
0	1	0
1	0	0
1	1	0

બુલિયન સમીકરણ:  $Y = (A + B)' = A'B'$

AND ગેટ:

લૉજિક સિમ્બોલ:



ટ્રુથ ટેબલ:

A	B	Y (A·B)
0	0	0
0	1	0
1	0	0
1	1	1

બુલિયન સમીકરણ:  $Y = A \cdot B$

મેમરી ટ્રીક

"NOR: NOT OR - ન તો એક કે ન તો બીજું; AND: બંને 1 હોવા જ જોઈએ"

પ્રશ્ન 2 [OR] (ક) [7 ગુણ]

ઉપરની લોજિક સર્કિટ માટે બુલિયન સમીકરણ લખો. આ સમીકરણને સરળ બનાવો અને આ સરળ સમીકરણની લોજિક સર્કિટ AND-OR-Invert મેથડથી દોરો

જવાબ

પગલું 1: સર્કિટમાંથી બુલિયન સમીકરણ લખો:  $Q = (A + B) \cdot (B + C \cdot ((B + C)'))$

$Q = (A + B) \cdot (B + C \cdot (B' \cdot C'))$

$Q = (A + B) \cdot (B + C \cdot B' \cdot C')$

પગલું 2: સમીકરણને સરળ બનાવો:

- નોંધ કરો કે  $C \cdot C' = 0$
- તેથી,  $C \cdot B' \cdot C' = 0$
- એટલે  $Q = (A + B) \cdot (B + 0) = (A + B) \cdot B = A \cdot B + B \cdot B = A \cdot B + B = B + A \cdot B = B(1 + A) = B$

પગલું 3: અંતિમ સરળ સમીકરણ:  $Q = B$

પગલું 4: AND-OR-Invert દ્વારા  $Q = B$  નું અમલીકરણ:

- આ ફક્ત ઇનપુટ B થી આઉટપુટ Q સુધીનો એક તાર છે



કોષ્ટક 10. સરળીકરણ પગલાં

પગલું	સમીકરણ	સરળીકરણ
1	$(A + B) \cdot (B + C \cdot ((B + C)'))$	મૂળ સમીકરણ
2	$(A + B) \cdot (B + C \cdot B' \cdot C')$	ડી મોર્ગનનો નિયમ લાગુ કરવો
3	$(A + B) \cdot (B + 0)$	$C \cdot C' = 0$
4	$(A + B) \cdot B$	સરળીકરણ
5	$A \cdot B + B \cdot B$	વિતરણ ગુણધર્મ
6	$A \cdot B + B$	આઇડેમ્પોટન્ટ ગુણધર્મ ( $B \cdot B = B$ )
7	$B(1 + A)$	ફેક્ટરિંગ
8	$B$	$1 + A = 1$

મેમરી ટ્રીક

"જ્યારે પૂરક ચલ ગુણાકાર કરે, તેઓ શૂન્ય થાય"

### પ્રશ્ન 3(અ) [3 ગુણ]

કોમ્બીનેશનલ સર્કિટની વ્યાખ્યા લખો. કોમ્બીનેશનલ સર્કિટના બે ઉદાહરણ લખો

#### જવાબ

**કોમ્બીનેશનલ સર્કિટ:** એક ડિજિટલ સર્કિટ જેનું આઉટપુટ માત્ર વર્તમાન ઇનપુટ મૂલ્યો પર આધારિત હોય છે અને અગાઉના ઇનપુટ અથવા સ્થિતિઓ પર નહીં. કોમ્બીનેશનલ સર્કિટમાં કોઈ મેમરી અથવા ફ્લિપ-ફ્લોપ હોતા નથી.

#### મુખ્ય લક્ષણો:

- આઉટપુટ ફક્ત વર્તમાન ઇનપુટ પર આધારિત હોય છે
- કોઈ મેમરી એલિમેન્ટ નથી
- કોઈ ફ્લિપ-ફ્લોપ પાથ નથી

#### કોમ્બીનેશનલ સર્કિટના ઉદાહરણો:

- મલ્ટિપ્લેક્સર (MUX)
- ડિકોડર
- એડર/સબટ્રેક્ટર
- એનકોડર
- કમ્પેરેટર

**કોષ્ટક 11.** કોમ્બીનેશનલ vs સિક્વેન્શિયલ સર્કિટ

લક્ષણ	કોમ્બીનેશનલ સર્કિટ	સિક્વેન્શિયલ સર્કિટ
મેમરી	ના	હા
ફ્લિપ-ફ્લોપ	ના	સામાન્ય રીતે
આઉટપુટ આધારિત	માત્ર વર્તમાન ઇનપુટ	વર્તમાન અને અગાઉના ઇનપુટ
ઉદાહરણો	મલ્ટિપ્લેક્સર, એડર	ફ્લિપ-ફ્લોપ, કાઉન્ટર

#### મેમરી ટ્રીક

"કોમ્બીનેશનલ સર્કિટ: વર્તમાન આવે, વર્તમાન જાય - કોઈ યાદ નહીં"

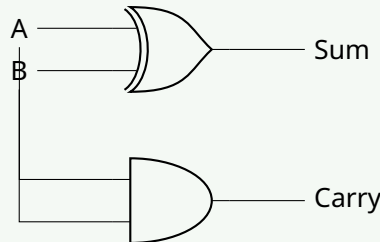
### પ્રશ્ન 3(બ) [4 ગુણ]

લોજિક સર્કિટ અને ટ્રુથ ટેબલની મદદથી હાફ એડર સમજાવો

#### જવાબ

**હાફ એડર:** એક કોમ્બીનેશનલ સર્કિટ જે બે બાયનરી અંકો ઉમેરે છે અને સમ અને કેરી આઉટપુટ ઉત્પન્ન કરે છે.

#### લોજિક સર્કિટ:



#### ટ્રુથ ટેબલ:

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

બુલિયન સમીકરણ:

- $\text{Sum} = A \oplus B = A'B + AB'$
- $\text{Carry} = A \cdot B$

મર્યાદાઓ:

- ત્રણ બાયનરી અંકો ઉમેરી શકતા નથી
- અગાઉના તબક્કામાંથી કેરી ઇનપુટ સમાવી શકતા નથી

મેમરી ટ્રીક

"XOR સમને માટે, AND કેરીને માટે"

## પ્રશ્ન 3(ક)(I) [3 ગુણ]

મલ્ટિપ્લેક્સર ટૂંકમાં સમજાવો

જવાબ

**મલ્ટિપ્લેક્સર (MUX):** એક કોમ્બીનેશનલ સર્કિટ જે સિલેક્ટ લાઇન્સના આધારે અનેક ઇનપુટ સિગ્નલ્સમાંથી એકને પસંદ કરે છે અને તેને એક આઉટપુટ લાઇન પર મોકલે છે.

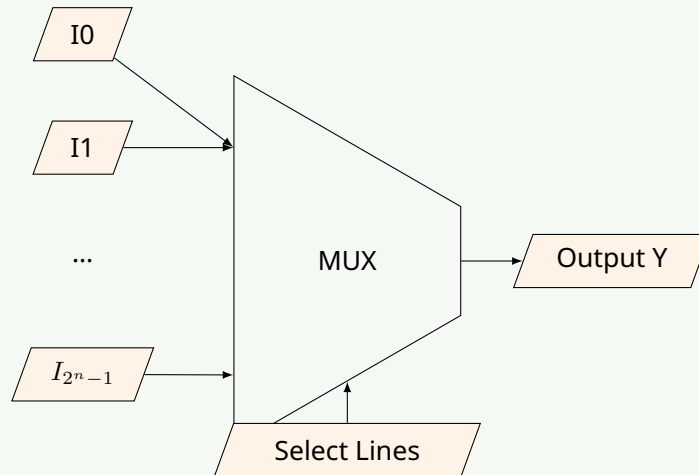
**મુખ્ય લક્ષણો:**

- ડિજિટલ સ્વિચ તરીકે કાર્ય કરે છે
- $2^n$  ડેટા ઇનપુટ,  $n$  સિલેક્ટ લાઇન, અને 1 આઉટપુટ ધરાવે છે
- સિલેક્ટ લાઇન્સ નક્કી કરે છે કે કયું ઇનપુટ આઉટપુટથી જોડાયેલું છે

**સામાન્ય મલ્ટિપ્લેક્સર:**

- 2:1 MUX (1 સિલેક્ટ લાઇન)
- 4:1 MUX (2 સિલેક્ટ લાઇન)
- 8:1 MUX (3 સિલેક્ટ લાઇન)

**મૂળભૂત રચના:**



**ઉપયોગો:**

- ડેટા રાઉટિંગ
- ડેટા પસંદગી
- પેરેલલથી સીરિયલ રૂપાંતર

- બુલિયન ફંક્શનનું અમલીકરણ

### મેમરી ટ્રીક

"ઘણા ઇન, સિલેક્શન પસંદ કરે, એક આઉટ"

## પ્રશ્ન ૩(ક)(II) [4 ગુણ]

8:1 મલ્ટિપ્લેક્સર ડિઝાઇન કરો. તેનું ટ્રુથ ટેબલ લખો અને લોજિક સર્કિટ દોરો

### જવાબ

8:1 મલ્ટિપ્લેક્સર ડિઝાઇન:

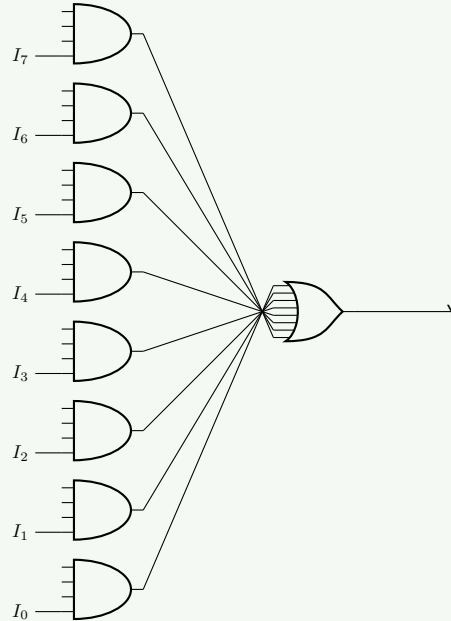
- 8 ડેટા ઇનપુટ ( $I_0$  થી  $I_7$ )
- 3 સિલેક્ટ લાઇન ( $S_2, S_1, S_0$ )
- 1 આઉટપુટ ( $Y$ )

ટ્રુથ ટેબલ:

$S_2$	$S_1$	$S_0$	આઉટપુટ $Y$
0	0	0	$I_0$
0	0	1	$I_1$
0	1	0	$I_2$
0	1	1	$I_3$
1	0	0	$I_4$
1	0	1	$I_5$
1	1	0	$I_6$
1	1	1	$I_7$

બુલિયન સમીકરણ:  $Y = S_2'S_1'S_0'I_0 + S_2'S_1'S_0I_1 + S_2'S_1S_0'I_2 + S_2'S_1S_0I_3 + S_2S_1'S_0'I_4 + S_2S_1'S_0I_5 + S_2S_1S_0'I_6 + S_2S_1S_0I_7$

લોજિક સર્કિટ:



## મેમરી ટ્રીક

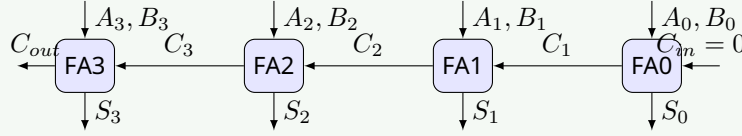
"આઠ ઇનપુટ, ત્રણ સિલેક્ટ, ડિકોડ કરો અને આઉટપુટ મેળવવા OR કરો"

## પ્રશ્ન 3 [OR] (અ) [3 ગુણ]

4-bit બાયનરી પેરેલલ એડરનો બ્લોક ડાયાગ્રામ દોરો

## જવાબ

4-bit બાયનરી પેરેલલ એડર: બે 4-bit બાયનરી નંબર ઉમેરતી અને 4-bit સરવાળો અને એક કેરી આઉટપુટ ઉત્પન્ન કરતી સર્કિટ.



## ઘટકો:

- ચાર ફુલ એડર (FA) કેસ્કેડમાં જોડાયેલા
- દરેક FA સંબંધિત બિટ્સ અને અગાઉના તબક્કાની કેરી ઉમેરે છે
- પ્રારંભિક કેરી-ઇન (Cin) સામાન્ય રીતે 0 હોય છે

## મેમરી ટ્રીક

"ચાર FA જોડાયેલા, કેરીઓ વચ્ચેથી પસાર થાય છે"

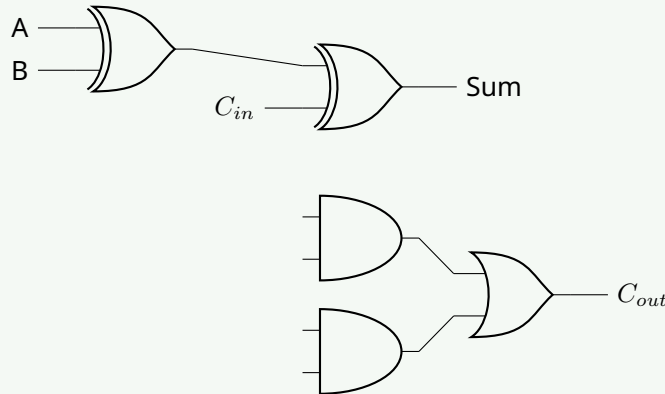
## પ્રશ્ન 3 [OR] (બ) [4 ગુણ]

લોજિક સર્કિટ અને ટ્રુથ ટેબલની મદદથી ફૂલ એડર સમજાવો

## જવાબ

ફૂલ એડર: એક કોમ્બીનેશનલ સર્કિટ જે ત્રણ બાયનરી અંક (બે ઇનપુટ અને એક કેરી-ઇન) ઉમેરે છે અને સરવાળો અને કેરી આઉટપુટ ઉત્પન્ન કરે છે.

લોજિક સર્કિટ:



## ટ્રુથ ટેબલ:

A	B	$C_{in}$	Sum	$C_{out}$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

બુલિયન સમીકરણ:

- $Sum = A \oplus B \oplus C_{in}$
- $C_{out} = AB + C_{in}(A \oplus B)$

મેમરી ટ્રીક

"ત્રણેય XOR કરો સમ માટે, ANDsને OR કરો કેરી માટે"

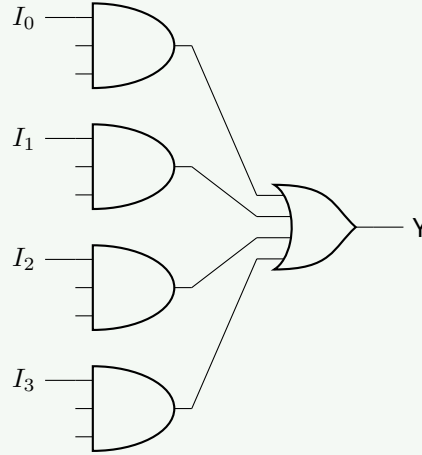
### પ્રશ્ન 3 [OR] (ક) (I) [3 ગુણ]

લોજિક સર્કિટ અને ટ્રુથ ટેબલની મદદથી 4:1 મલ્ટિપ્લેક્સર સમજાવો

જવાબ

4:1 મલ્ટિપ્લેક્સર: એક ડિજિટલ સ્વિચ જે બે સિલેક્ટ લાઇન-સના આધારે ચાર ઇનપુટ લાઇન-સમાંથી એકને પસંદ કરે છે અને તેને આઉટપુટથી જોડે છે.

લોજિક સર્કિટ:



ટ્રુથ ટેબલ:

$S_1$	$S_0$	આઉટપુટ Y
0	0	$I_0$
0	1	$I_1$
1	0	$I_2$
1	1	$I_3$

બુલિયન સમીકરણ:  $Y = S_1' S_0' I_0 + S_1' S_0 I_1 + S_1 S_0' I_2 + S_1 S_0 I_3$

## મેમરી ટ્રીક

"બે સિલેક્ટ લાઇન ચાર ઇનપુટમાંથી એક પસંદ કરે છે"

## પ્રશ્ન 3 [OR] (ક) (II) [4 ગુણ]

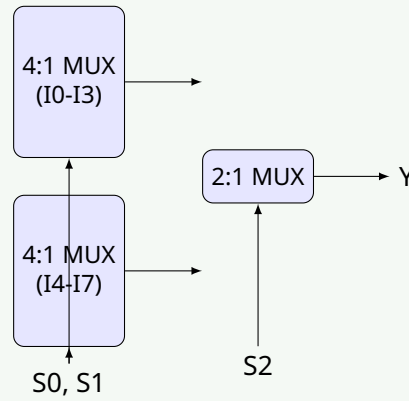
બે 4:1 મલ્ટિપ્લેક્સરનો ઉપયોગ કરીને 8:1 મલ્ટિપ્લેક્સર ડિઝાઇન કરો.

## જવાબ

ડિઝાઇન અભિગમ: 8:1 MUX બનાવવા માટે બે 4:1 MUX અને એક 2:1 MUX વાપરો.

1. પ્રથમ 4:1 MUX ઇનપુટ  $I_0 - I_3$  સંભાળે છે, સિલેક્ટ લાઇન  $S_0, S_1$  નો ઉપયોગ કરીને
2. બીજો 4:1 MUX ઇનપુટ  $I_4 - I_7$  સંભાળે છે, સિલેક્ટ લાઇન  $S_0, S_1$  નો ઉપયોગ કરીને
3. 2:1 MUX બે 4:1 MUXના આઉટપુટ વચ્ચે  $S_2$  નો ઉપયોગ કરીને પસંદગી કરે છે

બ્લોક ડાયાગ્રામ:



ટ્રુથ ટેબલ:

$S_2$	$S_1$	$S_0$	આઉટપુટ Y
0	0	0	$I_0$
0	0	1	$I_1$
0	1	0	$I_2$
0	1	1	$I_3$
1	0	0	$I_4$
1	0	1	$I_5$
1	1	0	$I_6$
1	1	1	$I_7$

## મેમરી ટ્રીક

" $S_0, S_1$  દરેક 4:1 MUXમાંથી પસંદ કરે છે,  $S_2$  તેમની વચ્ચે પસંદ કરે છે"

## પ્રશ્ન 4(અ) [3 ગુણ]

સિક્વન્સીયલ સર્કિટની વ્યાખ્યા લખો. તેના બે ઉદાહરણ લખો



## જવાબ

**સિકવન્સીયલ સર્કિટ:** એક ડિજિટલ સર્કિટ જેનું આઉટપુટ માત્ર વર્તમાન ઇનપુટ પર જ નહીં પણ ઇનપુટના ભૂતકાળના ક્રમ (ઇતિહાસ/અગાઉની સ્થિતિ) પર પણ આધારિત હોય છે.

**મુખ્ય લક્ષણો:**

- મેમરી એલિમેન્ટ્સ (ફ્લિપ-ફ્લોપ) ધરાવે છે
- આઉટપુટ વર્તમાન ઇનપુટ અને અગાઉની સ્થિતિઓ બંને પર આધારિત છે
- સામાન્ય રીતે ફ્લિપ-ફ્લોપ પાથ સમાવે છે
- સિંક્રોનાઇઝેશન માટે ક્લોક સિગ્નલની જરૂર પડે છે (સિંક્રોનસ સર્કિટ માટે)

**સિકવન્સીયલ સર્કિટના ઉદાહરણો:**

- ફ્લિપ-ફ્લોપ (SR, JK, D, T)
- રજિસ્ટર (શિફ્ટ રજિસ્ટર)
- કાઉન્ટર (બાયનરી, ડેકેડ, રિંગ કાઉન્ટર)
- સ્ટેટ મશીન
- મેમરી યુનિટ

**કોષ્ટક 12. સિકવન્સીયલ vs કોમ્બીનેશનલ સર્કિટ**

લક્ષણ	સિકવન્સીયલ સર્કિટ	કોમ્બીનેશનલ સર્કિટ
મેમરી	હા	ના
ફ્લિપ-ફ્લોપ	સામાન્ય રીતે	ના
આઉટપુટ આધારિત	વર્તમાન અને અગાઉના ઇનપુટ	માત્ર વર્તમાન ઇનપુટ
ક્લોક જરૂરી	સામાન્ય રીતે	ના
ઉદાહરણો	ફ્લિપ-ફ્લોપ, કાઉન્ટર	મલ્ટિપ્લેક્સર, એડર

## મેમરી ટ્રીક

"સિકવન્સીયલ ઇતિહાસ યાદ રાખે છે, કોમ્બીનેશનલ માત્ર વર્તમાન જાણે છે"

## પ્રશ્ન 4(બ) [4 ગુણ]

## ડિકેડ કાઉન્ટર ડિઝાઇન કરો

## જવાબ

**ડિકેડ કાઉન્ટર:** એક સિકવન્સીયલ સર્કિટ જે 0 થી 9 (ડેસિમલ) સુધી ગણે છે અને પછી 0 પર રીસેટ થાય છે.

**JK ફ્લિપ-ફ્લોપનો ઉપયોગ કરી ડિઝાઇન:**

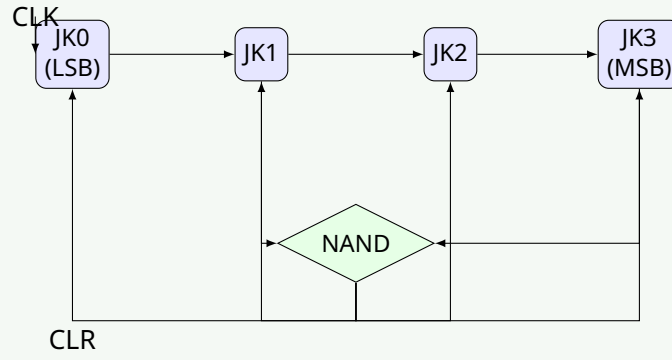
- 4 બિટ બાયનરી નંબર રજૂ કરવા માટે 4 JK ફ્લિપ-ફ્લોપ ( $Q_3, Q_2, Q_1, Q_0$ ) જરૂરી છે
- 0000 થી 1001 (0-9 ડેસિમલ) સુધી ગણે છે પછી રીસેટ થાય છે

**J-K ઇનપુટ સમીકરણ:**

- $J_0 = K_0 = 1$  (દરેક ક્લોક પર ટોગલ)
- $J_1 = K_1 = Q_0 \cdot \overline{Q_3}$
- $J_2 = K_2 = Q_1 \cdot Q_0$
- $J_3 = K_3 = Q_2 \cdot Q_1 \cdot Q_0 + Q_3 \cdot Q_0$

**રીસેટ સ્થિતિ:** જ્યારે  $Q_3 \cdot Q_1 = 1$  (સ્થિતિ 1010), બધા ફ્લિપ-ફ્લોપ રીસેટ કરો

**બ્લોક ડાયાગ્રામ (Asynchronous/Ripple):**



### મેમરી ટ્રીક

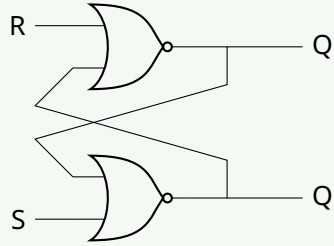
"BCD ગણો, 9 પછી રીસેટ"

## પ્રશ્ન 4(ક)(I) [3 ગુણ]

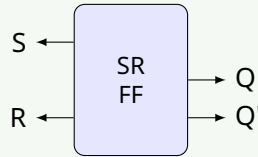
NOR ગેટની મદદથી S-R ફ્લિપ-ફ્લોપ સમજાવો. તેનો લૉજિક સિમ્બોલ દોરો અને ટ્રુથ ટેબલ લખો.

### જવાબ

**NOR ગેટથી S-R ફ્લિપ-ફ્લોપ:** બે ક્રોસ-કપલ્ડ NOR ગેટમાંથી બનેલું એક મૂળભૂત ફ્લિપ-ફ્લોપ જે એક બિટની માહિતી સંગ્રહિત કરી શકે છે. લૉજિક સર્કિટ:



લૉજિક સિમ્બોલ:



ટ્રુથ ટેબલ:

S	R	Q (આગામી)	Q' (આગામી)	ઓપરેશન
0	0	Q (અગાઉની)	Q' (અગાઉની)	મેમરી (કોઈ ફેરફાર નહીં)
0	1	0	1	રીસેટ
1	0	1	0	સેટ
1	1	0	0	અમાન્ય (ટાળો)

### મેમરી ટ્રીક

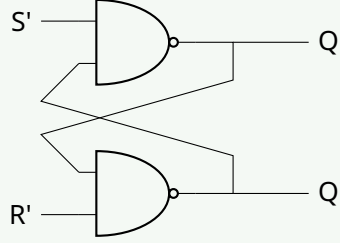
"S થી 1 સેટ થાય, R થી 0 રીસેટ થાય, બંને એકસાથે અમાન્ય સ્થિતિ આપે"

## પ્રશ્ન 4(ક)(II) [4 ગુણ]

NAND ગેટની મદદથી S-R ફ્લિપ-ફ્લોપ સમજાવો. S-R ફ્લિપ-ફ્લોપની મર્યાદા લખો

**જવાબ**

NAND ગેટથી S-R ફ્લિપ-ફ્લોપ: બે ક્રોસ-કપલ્ડ NAND ગેટમાંથી બનેલું એક મૂળભૂત ફ્લિપ-ફ્લોપ. લૉજિક સર્કિટ:



SR ફ્લિપ-ફ્લોપની મર્યાદાઓ:

1. અમાન્ય સ્થિતિ: જ્યારે  $S=1$ ,  $R=1$  (NOR માટે) અથવા  $S=0$ ,  $R=0$  (NAND માટે), આઉટપુટ અનિશ્ચિત રહે છે
2. રેસ કન્ડિશન: જ્યારે ઇનપુટ એકસાથે બદલાય છે, ત્યારે અંતિમ સ્થિતિ અનિશ્ચિત હોઈ શકે છે
3. કલોકિંગ મેકેનિઝમ નથી: અન્ય ડિજિટલ ઘટકો સાથે સિંક્રોનાઇઝ થઈ શકતું નથી
4. એજ-ટ્રિગર્ડ નથી: ટૂંકા પલ્સને વિશ્વસનીય રીતે પ્રતિક્રિયા આપી શકતું નથી
5. અનિચ્છનીય ટોગલિંગ: નોઇઝ કે ગ્લિચને પ્રતિક્રિયા આપી શકે છે

**કોષ્ટક 13.** NAND vs NOR SR ફ્લિપ-ફ્લોપ

લક્ષણ	NAND SR ફ્લિપ-ફ્લોપ	NOR SR ફ્લિપ-ફ્લોપ
સક્રિય ઇનપુટ	લો (0)	હાઇ (1)
નિષ્ક્રિય ઇનપુટ	હાઇ (1)	લો (0)
અમાન્ય સ્થિતિ	$S=0$ , $R=0$	$S=1$ , $R=1$

**મેમરી ટ્રીક**

"NAND: ઇનપુટ એક્ટિવ-લો, NOR: ઇનપુટ એક્ટિવ-હાઇ; બંનેમાં એક અમાન્ય સ્થિતિ છે"

## પ્રશ્ન 4 [OR] (અ) [3 ગુણ]

ફ્લિપ-ફ્લોપની વ્યાખ્યા લખો. ફ્લિપ-ફ્લોપના પ્રકાર લખો

**જવાબ**

**ફ્લિપ-ફ્લોપ:** એક મૂળભૂત સિક્વન્સીયલ ડિજિટલ સર્કિટ જે એક બિટની માહિતી સંગ્રહિત કરી શકે છે અને બે સ્થાયી સ્થિતિઓ (0 અથવા 1) ધરાવે છે. તે ડિજિટલ સિસ્ટમમાં મૂળભૂત મેમરી એલિમેન્ટ તરીકે કામ કરે છે.

**મુખ્ય લક્ષણો:**

- બાયસ્ટેબલ મલ્ટિવાયબ્રેટર (બે સ્થાયી સ્થિતિઓ)
- જ્યાં સુધી બદલવાનો નિર્દેશ ન અપાય ત્યાં સુધી પોતાની સ્થિતિ અનિશ્ચિત સમય સુધી જાળવી રાખી શકે છે
- રજિસ્ટર, કાઉન્ટર અને મેમરી સર્કિટ માટે મૂળભૂત બિલ્ડિંગ બ્લોક બને છે

**ફ્લિપ-ફ્લોપના પ્રકાર:**

ફ્લિપ-ફ્લોપ પ્રકાર	વર્ણન
SR (સેટ-રીસેટ)	સૌથી મૂળભૂત ફ્લિપ-ફ્લોપ જેમાં સેટ અને રીસેટ ઇનપુટ હોય છે
JK	SR ફ્લિપ-ફ્લોપની સુધારેલી આવૃત્તિ જે અમાન્ય સ્થિતિ દૂર કરે છે
D (ડેટા)	ઇનપુટ D પરનો મૂલ્ય સંગ્રહિત કરે છે, ડેટા સ્ટોરેજ માટે વપરાય છે
T (ટોગલ)	ટ્રિગર થયે સ્થિતિ બદલે છે, કાઉન્ટર માટે ઉપયોગી
માસ્ટર-સ્લેવ	રેસ કન્ડિશન અટકાવતું બે-તબક્કાનું ફ્લિપ-ફ્લોપ

### મેમરી ટ્રીક

"એક સિંગલ સ્ટેટ સ્ટોરેજ: SR, JK, D, T"

## પ્રશ્ન 4 [OR] (બ) [4 ગુણ]

3-bit રિંગ કાઉન્ટર ડિઝાઇન કરો

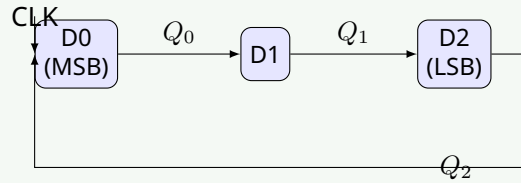
### જવાબ

**રિંગ કાઉન્ટર:** એક સર્ક્યુલર શિફ્ટ રજિસ્ટર જેમાં ફક્ત એક બિટ સેટ (1) હોય છે અને બાકી બધા રીસેટ (0) હોય છે. એકમાત્ર સેટ બિટ કલોક થતાં રજિસ્ટરમાં "ફરે" છે.

**D ફ્લિપ-ફ્લોપનો ઉપયોગ કરી ડિઝાઇન:**

- 3-bit કાઉન્ટર માટે 3 D ફ્લિપ-ફ્લોપ જરૂરી છે
- પ્રારંભિક સ્થિતિ: 100, પછી 010, 001, અને પાછા 100 પર જાય છે

**બ્લોક ડાયાગ્રામ:**



### મેમરી ટ્રીક

"એક હોટ બિટ વર્તુળમાં ફરે છે"

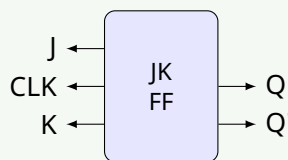
## પ્રશ્ન 4 [OR] (ક)(I) [3 ગુણ]

લોજિક સિમ્બોલ અને ટ્રુથ ટેબલની મદદથી J-K ફ્લિપ-ફ્લોપ સમજાવો

### જવાબ

**J-K ફ્લિપ-ફ્લોપ:** SR ફ્લિપ-ફ્લોપની સુધારેલી આવૃત્તિ જે અમાન્ય સ્થિતિ દૂર કરે છે અને બધા ઇનપુટ સંયોજનોમાં સચોટ વર્તન દર્શાવે છે.

**લોજિક સિમ્બોલ:**



**ટ્રુથ ટેબલ:**

J	K	Q (આગામી)	ઓપરેશન
0	0	Q (અગાઉની)	કોઈ ફેરફાર નહીં
0	1	0	રીસેટ
1	0	1	સેટ
1	1	Q' (અગાઉની)	ટોગલ

### મેમરી ટ્રીક

"J સેટ કરે, K રીસેટ કરે, બંને ટોગલ કરે, કોઈ નહીં યાદ રાખે"

## પ્રશ્ન 4 [OR] (ક)(II) [4 ગુણ]

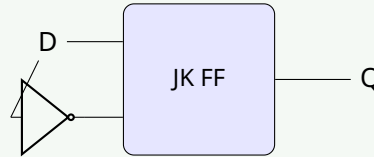
J-K ફ્લિપ-ફ્લોપનો ઉપયોગ કરી D ફ્લિપ-ફ્લોપ અને T ફ્લિપ-ફ્લોપની લૉજિક સર્કિટ દોરો

### જવાબ

J-K ફ્લિપ-ફ્લોપનો ઉપયોગ કરી D ફ્લિપ-ફ્લોપ:

- D ઇનપુટને J સાથે જોડો
- D' (NOT D)ને K સાથે જોડો

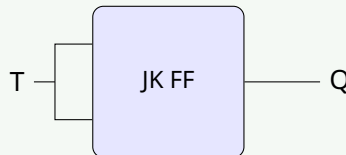
લૉજિક સર્કિટ:



J-K ફ્લિપ-ફ્લોપનો ઉપયોગ કરી T ફ્લિપ-ફ્લોપ:

- T ઇનપુટને J અને K બંને સાથે જોડો

લૉજિક સર્કિટ:



### મેમરી ટ્રીક

"D સીધું અનુસરે, T સાચું હોય ત્યારે ટોગલ થાય"

## પ્રશ્ન 5(અ) [3 ગુણ]

RAM અને ROMની સરખામણી કરો

### જવાબ

RAM (Random Access Memory) vs ROM (Read-Only Memory):

કોષ્ટક 14. RAM vs ROM સરખામણી

લક્ષણ	RAM	ROM
<b>**પૂર્ણ નામ**</b>	Random Access Memory	Read-Only Memory
<b>**ડેટા નિભાવણી**</b>	અસ્થાયી (પાવર બંધ થતાં ડેટા ગુમાવે)	સ્થાયી (પાવર વિના પણ ડેટા જળવાય)
<b>**વાંચન/લેખન ક્ષમતા**</b>	વાંચન અને લેખન બંને	મુખ્યત્વે માત્ર વાંચન
<b>**ગતિ**</b>	વધુ ઝડપી	ધીમી
<b>**બિટ દીઠ ખર્ચ**</b>	વધુ	ઓછો
<b>**ઉપયોગો**</b>	અસ્થાયી ડેટા સ્ટોરેજ	બૂટ સૂચનાઓ, ફર્મવેર

### મેમરી ટ્રીક

"RAM વાંચે અને સુધારે (પણ ભૂલી જાય), ROM શટડાઉન પર યાદ રાખે (પણ નિશ્ચિત)"

## પ્રશ્ન 5(બ) [4 ગુણ]

સિરિયલ ઇન સિરિયલ આઉટ શિફ્ટ રજિસ્ટર સમજાવો

### જવાબ

**સિરિયલ ઇન સિરિયલ આઉટ (SISO) શિફ્ટ રજિસ્ટર:** એક સિક્વન્સિયલ સર્કિટ જે ઇનપુટ અને આઉટપુટ બંને પર ડેટાને એક સમયે એક બિટ શિફ્ટ કરે છે.

**કાર્યપદ્ધતિ:**

- ડેટા સિરિયલી એક બિટ એક વખતે દાખલ થાય છે
- દરેક કલોક પલ્સ પર દરેક બિટ રજિસ્ટરમાંથી શિફ્ટ થાય છે
- ડેટા સિરિયલી એક બિટ એક વખતે બહાર નીકળે છે

**બ્લોક ડાયાગ્રામ:**



### મેમરી ટ્રીક

"બિટ્સ લાઇનમાં પ્રવેશે, શ્રેણીમાં આગળ વધે, ક્રમમાં બહાર નીકળે"

## પ્રશ્ન 5(ક) [7 ગુણ]

લોજિક ફેમિલિઝ પર ટૂંક નોંધ લખો

### જવાબ

**લોજિક ફેમિલિઝ:** સમાન ઇલેક્ટ્રિકલ લક્ષણો, ફેબ્રિકેશન ટેકનોલોજી અને લોજિક અમલીકરણ સાથેના ડિજિટલ ઇન્ટિગ્રેટેડ સર્કિટના સમૂહો. મુખ્ય લોજિક ફેમિલિઝ:

1. TTL (ટ્રાન્ઝિસ્ટર-ટ્રાન્ઝિસ્ટર લોજિક):

- બાયપોલર જંક્શન ટ્રાન્ઝિસ્ટર પર આધારિત
- Supply voltage: 5V

2. CMOS (કોમ્પ્લિમેન્ટરી મેટલ-ઓક્સાઇડ-સેમિકન્ડક્ટર):

- MOSFETs (P-ટાઇપ અને N-ટાઇપ) પર આધારિત
- ખૂબ ઓછો પાવર વપરાશ

3. ECL (ઇમિટર-કપલ્ડ લોજિક):

- અત્યંત ઊંચી ઝડપ (સૌથી ઝડપી લોજિક ફેમિલી)

## કોષ્ટક 15. સરખામણી કોષ્ટક

પેરામીટર	TTL	CMOS	ECL
ઝડપ	મધ્યમ	ઓછી થી ઊંચી	ખૂબ ઊંચી
પાવર વપરાશ	મધ્યમ	ખૂબ ઓછો	ઊંચો
નોઇઝ ઇમ્યુનિટી	ઊંચી	ખૂબ ઊંચી	ઓછી
ફેન-આઉટ	10	50+	25

## મેમરી ટ્રીક

"TTL ટ્રાન્ઝિસ્ટર ટેકનોલોજી, CMOS કરંટ ઓછો વાપરે છે, ECL એક્સટ્રીમ ઝડપે કામ કરે છે"

## પ્રશ્ન 5 [OR] (અ) [3 ગુણ]

## SRAM અને DRAMની સરખામણી કરો

## જવાબ

SRAM (સ્ટેટિક RAM) vs DRAM (ડાયનેમિક RAM):

## કોષ્ટક 16. SRAM vs DRAM સરખામણી

લક્ષણ	SRAM	DRAM
**પૂર્ણ નામ**	Static Random Access Memory	Dynamic Random Access Memory
**સ્ટોરેજ એલિમેન્ટ**	ફ્લિપ-ફ્લોપ	કેપેસિટર
**રિફ્રેશિંગ**	જરૂરી નથી	સમયાંતરે જરૂરી (ms)
**ઝડપ**	વધુ ઝડપી	ધીમી
**ડેન્સિટી**	ઓછી (મોટો સેલ સાઇઝ)	ઊંચી (નાનો સેલ સાઇઝ)
**બિટ દીઠ ખર્ચ**	વધુ	ઓછો
**ઉપયોગો**	કેશ મેમરી	મુખ્ય મેમરી (RAM)

## મેમરી ટ્રીક

"સ્ટેટિક સ્થિર રહે છે છ ટ્રાન્ઝિસ્ટર સાથે, ડાયનેમિક ડ્રેઇન થાય અને નિયમિત રિફ્રેશ જોઈએ"

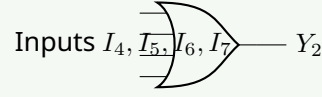
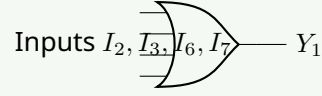
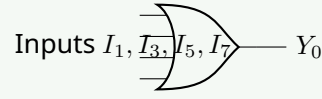
## પ્રશ્ન 5 [OR] (બ) [4 ગુણ]

## 8:3 એનકોડર સમજાવો

## જવાબ

**8:3 એનકોડર:** એક કોમ્બીનેશનલ સર્કિટ જે 8 ઇનપુટ લાઇન-સને 3 આઉટપુટ લાઇન-સમાં રૂપાંતરિત કરે છે, મૂળભૂત રીતે સક્રિય ઇનપુટ લાઇનને તેની બાયનરી પોઝિશનમાં રૂપાંતરિત કરે છે.

**લોજિક સર્કિટ:**



બુલિયન સમીકરણ:

- $Y_0 = I_1 + I_3 + I_5 + I_7$
- $Y_1 = I_2 + I_3 + I_6 + I_7$
- $Y_2 = I_4 + I_5 + I_6 + I_7$

મેમરી ટ્રીક

"આઠ ઇનપુટ તેમના સ્થાન ત્રણ બિટમાં બને"

## પ્રશ્ન 5 [OR] (ક) [7 ગુણ]

લોજિક ફેમિલિઝ માટે નીચેની વ્યાખ્યાઓ લખો (i) ફેન-ઇન (ii) ફેન-આઉટ (iii) નોઇસ માર્જિન (iv) પ્રોપેગેશન ડિલે (v) પાવર ડિસિપેશન

જવાબ

લોજિક ફેમિલિઝના મુખ્ય પેરામીટર:

1. ફેન-ઇન: લોજિક ગેટ સ્વીકારી શકે તેવા ઇનપુટની મહત્તમ સંખ્યા
2. ફેન-આઉટ: એક ગેટ આઉટપુટ દ્વારા વિશ્વસનીય રીતે ડ્રાઇવ થઈ શકતા સમાન ગેટની મહત્તમ સંખ્યા
3. નોઇઝ માર્જિન: અનિચ્છનીય ઇલેક્ટ્રિકલ નોઇઝ/સિગ્નલ સહન કરવાની ક્ષમતા
4. પ્રોપેગેશન ડિલે: ઇનપુટ ચેન્જ અને તેના તરત પછીના આઉટપુટ ચેન્જ વચ્ચેનો સમય વિલંબ
5. પાવર ડિસિપેશન: ગેટ દ્વારા વપરાતી પાવરની માત્રા

મેમરી ટ્રીક

"પાંચ ફેક્ટર: ફેન-ઇન ઇનપુટ ગણો, ફેન-આઉટ ગેટ ચલાવે, નોઇઝ માર્જિન દખલ સામે લડે, પ્રોપેગેશન ડિલે ઝડપ માપે, પાવર ડિસિપેશન ગરમી ઉત્પન્ન કરે"