

Subject Name (Gujarati)

4321102 -- Summer 2024

Semester 1 Study Material

Detailed Solutions and Explanations

પ્રશ્ન 1(અ) [3 માકર્સ]

કન્વર્ટ કરો: $(110101)_2 = (\underline{\hspace{2cm}})_{10} = (\underline{\hspace{2cm}})_8 = (\underline{\hspace{2cm}})_{16}$

જવાબ

સ્ટેપ-બાય-સ્ટેપ કન્વર્ન (110101)₂ :

બાઇનરી $(110101)_2$	ડિસિમલ	ઓક્ટલ	હેક્ચરડિસિમલ
$1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$ $(110101)_2$	$32+16+0+4+0+1 = 53$ $(53)_{10}$	$6 \times 8^1 + 5 \times 8^0 = 48 + 5 = 53$ $(65)_8$	$3 \times 16^1 + 5 \times 16^0 = 48 + 5 = 35$ $(35)_{16}$

મેમરી ટ્રીક

"બાઇનરી ડિજિટ આઉટ હિયર" (BDOH) બાઇનરીકન્વર્ન માટે.

પ્રશ્ન 1(બ) [4 માકર્સ]

**કરો: (i) $(11101101)_2 + (10101000)_2$ (ii) $(11011)_2 * (1010)_2$ **

જવાબ

બાઇનરી સરવાળા અને ગુણાકાર માટે ટેબલ:

(i) બાઇનરી સરવાળો	(ii) બાઇનરી ગુણાકાર
11101101	11011
$+ 10101000$	$\times 1010$
-----	-----
110010101	00000
	11011
	00000
	11011

	11101110

ડિસિમલ વેરિફિકેશન:

- 1. $(11101101)_2 = 237, (10101000)_2 = 168, = 405 = (110010101)_2$
- 1. $(11011)_2 = 27, (1010)_2 = 10, = 270 = (11101110)_2$

મેમરી ટ્રીક

સરવાળા માટે "કેરી અપ મેક્સ સમ" અને ગુણાકાર માટે "શિફ્ટ લેફ્ટ એડ પ્રોડક્ટ".

પ્રશ્ન 1(ક) [7 માકર્સ]

(i) કન્વર્ટ કરો: $(48)_{10} = (\underline{\hspace{2cm}})_2 = (\underline{\hspace{2cm}})_8 = (\underline{\hspace{2cm}})_{16}$

પ્રશ્ન 1(ક) અથવા [7 માંકર્સ]

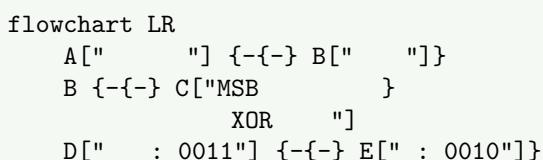
કોડ સમજાવો: ASCII, BCD, Gray

જવાબ

સામાન્ય ડિજિટલ કોડસનું ટેબલ:

કોડ	વર્ણન	ઉદાહરણ
ASCII (American Standard Code for Information Interchange)	128 કેરેક્ટર્સને 2જૂ કરતો 7-બિટ કોડ જેમાં આફ્ઝાબેટ્સ, નંબર્સ અને સ્પેશિયલ સિમ્બોલ્સ શામેલ છે	A = 65 (1000001) ₂
BCD (Binary Coded Decimal)	દરેક ડિસિમલ અંક (0-9) ને 4 બિટ્સનો ઉપયોગ કરીને 2જૂ કરે છે	42 = 0100 0010
Gray Code	બાઇનરી કોડ જેમાં આસપાસના નંબરો માત્ર એક બિટથી અલગ પડે છે	(0,1,3,2) = (00,01,11,10)

ડાયાગ્રામ: ગ્રે કોડ જનરેશન:



મેમરી ટ્રીક

“ઓલવેજ બાઇનરી જનરેટ્સ” - દરેક કોડનો પ્રથમ અક્ષર (ASCII, BCD, Gray).

પ્રશ્ન 2(અ) [3 માંકર્સ]

બુલિયન બીજગણિતનો ઉપયોગ કરીને સરળ બનાવો: $Y = A B + A' B + A' B' + A B'$

જવાબ

સ્ટેપ-બાય-સ્ટેપ સરળીકરણ:

સ્ટેપ	એક્સપ્રેશન	બુલિયન નિયમ
$Y = A B + A' B + A' B' + A B'$	પ્રારંભિક એક્સપ્રેશન	-
$Y = A(B + B') + A'(B + B')$	ફેક્ટરિંગ	ડિસ્ટ્રીબ્યુટિવ લો
$Y = A(1) + A'(1)$	કોમ્પ્લેમેન્ટ લો	$B + B' = 1$
$Y = A + A'$	સરળીકરણ	-
$Y = 1$	કોમ્પ્લેમેન્ટ લો	$A + A' = 1$

મેમરી ટ્રીક

બુલિયન સરળીકરણ સ્ટેપ્સ માટે “ફેક્ટર, સિમ્પ્લિકેશન, ફિનિશ”.

પ્રશ્ન 2(બ) [4 માંકર્સ]

K-મેપનો ઉપયોગ કરીને નીચેના બુલિયન ફૂક્શન ને સરળ બનાવો: $f(A,B,C,D) = \Sigma m (0,3,4,6,8,11,12)$

જવાબ

K-મેપ સોલ્યુશન:

AB

CD	00	01	11	10
00	1	0	0	1
01	0	0	0	1
11	0	1	0	0
10	0	0	1	0

ગુપિંગ:

- ગુપ 1: $m(0,8) = A'C'D'$
- ગુપ 2: $m(4,12) = BD'$
- ગુપ 3: $m(3,11) = CD$
- ગુપ 4: $m(6) = A'B'CD'$

સરળ કરેલ એક્સપ્રેશન: $f(A,B,C,D) = A'C'D' + BD' + CD + A'B'CD'$

મેમરી ટ્રીક

K-મેપ ગુપિંગ સ્ટ્રેટેજી માટે "ગુપ પાવર્સ ઓફ ટુ".

પ્રશ્ન 2(ક) [7 માક્સી]

NOR ગેટને સ્વર્ચ આફ્ટિઅંડ સાથે યુનિવર્સલ ગેટ તરીકે સમજાવો.

જવાબ

NOR એઝ યુનિવર્સલ ગેટ:

ફૂક્શન	NOR નો ઉપયોગ કરી ઇમ્પિલમેન્ટેશન	કુથ ટેબલ
NOT ગેટ		A
		0
		1
AND ગેટ		A B
		0 0
		0 1
		1 0
		1 1
OR ગેટ		A B
		0 0
		0 1
		1 0
		1 1

ડાયાગ્રામ: NOR ઇમ્પિલમેન્ટેશન:

```
flowchart TD
    A["NOT: A {-1{-} A"}]
    B["AND: A {-{-}1{-}{-}|-}
        | 1{{-}{-}} A•B"]
    C["OR: A {-{-}1{-}{-}|-}
        | |{{-}{-}} A+B"]
    B --> C
```

મેમરી ટ્રીક

NOR ગેટ ઇમ્પિલમેન્ટેશન માટે "NOT AND OR, NOR કરે મોર".

પ્રશ્ન 2(અ) અથવા [3 માક્સી]

બુલિયન સમીકરણ માટે લોજિક સર્કિટ દોરો: $Y = (A + B') . (A' + B') . (B + C)$

જવાબ

લોજિક સર્કિટ ઇમ્પિલમેન્ટેશન:

```

flowchart TD
    A["A"] --{-{->}-- D["OR"]
    B["B{"] --{->-->}-- D
    D --{-{->}-- G["AND"]
    A1["A{"] --{->-->}-- E["OR"]
    B1["B{"] --{->-->}-- E
    E --{-{->}-- G
    B2["B{"] --{-{->}-- F["OR"]
    C["C{"] --{-{->}-- F
    F --{-{->}-- G
    G --{-{->}-- Y["Y"]

```

ટુથ ટેબલ વેરિફિકેશન:

- ટર્મ 1: $(A + B')$
- ટર્મ 2: $(A' + B')$
- ટર્મ 3: $(B + C)$
- આઉટપુટ: $Y = \text{Term1} \cdot \text{Term2} \cdot \text{Term3}$

મેમરી ટ્રીક

જટિલ એક્સપ્રેશન માટે “દરેક ટર્મ અલગથી”.

પ્રશ્ન 2(બ) અથવા [4 માંકર્સ]

ડી-મોર્ગન્સના પ્રમેય લખો અને તેને સાબિત કરો.

જવાબ

ડી-મોર્ગન્સ પ્રમેય અને પૂછુણ:

પ્રમેય	સ્ટેટમેન્ટ	ટુથ ટેબલ દ્વારા પૂછુણ
પ્રમેય 1	$(A \cdot B)' = A' + B'$	A B 0 0 0 1 1 0 1 1
પ્રમેય 2	$(A+B)' = A' \cdot B'$	A B 0 0 0 1 1 0 1 1

ડાયાગ્રામ: ડી-મોર્ગન્સ લો વિગ્યુલાઇઝેશન:

```

flowchart TB
    A["(A•B){ = A+B" ] --{->-->}-- B["      ]
                           AND   OR
                           "]
    C["(A+B){ = A•B" ] --{->-->}-- D["      ]
                           OR   AND
                           "]

```

મેમરી ટ્રીક

ડી-મોર્ગન્સ લો લાગુ કરવા માટે “બાર તોડો, ઓપરેશન બદલો, ઇનપુટ ઇન્વર્ટ કરો”.

પ્રશ્ન 2(ક) અથવા [7 માંકર્સ]

સિમ્બોલ, ટુથ ટેબલ અને સમીકરણની મદદથી તમામ લોજિક ગેટ્સ સમજાવો.

જવાબ

લોજિક ગેટ્સ સમરી:

ગેટ	સિમ્બોલ	ટુથ ટેબલ	સમીકરણ	વર્ણન
AND	A B	Y 0 0 0 1 1 0 1 1	Y = A•B	
OR	A B	Y 0 0 0 1 1 0 1 1	Y = A+B	
NOT	A	Y 0 1	Y = A'	
NAND	A B	Y 0 0 0 1 1 0 1 1	Y = (A•B)'	
NOR	A B	Y 0 0 0 1 1 0 1 1	Y = (A+B)'	
XOR	A B	Y 0 0 0 1 1 0 1 1	Y = A	
XNOR	A B	Y 0 0 0 1 1 0 1 1	Y = (A)'	

મેમરી ટ્રીક

"All Operations Need Necessary eXecution" (દરેક ગેટનો પહેલો અક્ષર - AND, OR, NOT, NAND, NOR, XOR).

પ્રશ્ન 3(અ) [3 માંકર્સ]

સંક્ષિપ્તમાં 4:2 એન્કોડર સમજાવો.

જવાબ

4-to-2 એન્કોડર ઓવરવ્યુ:

ફૂંક્શન	વર્ણન	ટુથ ટેબલ
4:2 એન્કોડર	4 ઇનપુટ લાઇન્સને 2 આઉટપુટ લાઇન્સમાં કન્વર્ટ કરે છે એક સમયે માત્ર એક જ ઇનપુટ એક્ટિવ	I ₀ I ₁ I ₂ I ₃ 1 0 0 0

ડાયાગ્રામ: 4:2 એન્કોડર:

```
flowchart TD
    I0["I_{0}"] --> E["4:2"]
    I1["I_{1}"] --> E
    I2["I_{2}"] --> E
    I3["I_{3}"] --> E
    E --> Y1["Y_{1}"]
    E --> Y0["Y_{0}"]
```

મેમરી ટ્રીક

એન્કોડર ફંક્શન માટે “ઇનપુટ પોઝિશન કિએટ્સ આઉટપુટ”.

પ્રશ્ન 3(બ) [4 માક્સની]

કુલ એડર બ્લોક્સનો ઉપયોગ કરીને 4-બિટ પેરેલલ એડરને સમજાવો.

જવાબ

4-બિટ પેરેલલ એડર:

કોમ્પોનેન્ટ	ફંક્શન
કુલ એડર	3 બિટ્સ (A, B, Carry-in) ને એડ કરે છે અને Sum અને Carry-out આપે છે
પેરેલલ એડર	4 કુલ એડરને કેરી પ્રોપેગેશન સાથે જોડે છે

ડાયાગ્રામ: 4-બિટ પેરેલલ એડર:

```
flowchart LR
    A0["A_{0}"] --> FA0["FA"]
    B0["B_{0}"] --> FA0
    CO["C_{0=0}"] --> FA0
    FA0 --> S0["S_{0}"]
    FA0 --> "C_{1}"
    FA0 --> FA1["FA"]

    A1["A_{1}"] --> FA1
    B1["B_{1}"] --> FA1
    FA1 --> S1["S_{1}"]
    FA1 --> "C_{2}"
    FA1 --> FA2["FA"]

    A2["A_{2}"] --> FA2
    B2["B_{2}"] --> FA2
    FA2 --> S2["S_{2}"]
    FA2 --> "C_{3}"
    FA2 --> FA3["FA"]

    A3["A_{3}"] --> FA3
    B3["B_{3}"] --> FA3
    FA3 --> S3["S_{3}"]
    FA3 --> C4["C_{4}"]
```

મેમરી ટ્રીક

પેરેલલ એડરમાં કેરી પ્રોપેગેશન માટે “કેરી ઓલવેજ પાસેસ રાઇટ”.

પ્રશ્ન 3(ક) [7 માંકર્સ]

ટુથ ટેબલ, સમીકરણ અને સર્કિટ ડાયાગ્રામ સાથે 8:1 મલ્ટિપ્લેક્સરનું વર્ણન કરો.

જવાબ

8:1 મલ્ટિપ્લેક્સર:

કોડોનન્ટ	વર્ણન	ફુલ્ફાળ
8:1 MUX	8 ઇનપુટ્સ, 3 સિલેક્ટ લાઇન્સ, 1 આઉટપુટ વાળો ડેટા સિલેક્ટર	સિલેક્ટ લાઇન્સના આધારે 8 ઇનપુટ્સમાંથી એક પસંદ કરે છે

ટુથ ટેબલ:

સિલેક્ટ લાઇન્સ	આઉટપુટ
$S_2 S_1 S_0$	Y
0 0 0	D_0
0 0 1	D_1
0 1 0	D_2
0 1 1	D_3
1 0 0	D_4
1 0 1	D_5
1 1 0	D_6
1 1 1	D_7

બુલિયન સમીકરણ: $Y = S'_2 \cdot S'_1 \cdot S'_0 \cdot D_0 + S'_2 \cdot S'_1 \cdot S_0 \cdot D_1 + S'_2 \cdot S_1 \cdot S'_0 \cdot D_2 + S'_2 \cdot S_1 \cdot S_0 \cdot D_3 + S_2 \cdot S'_1 \cdot S'_0 \cdot D_4 + S_2 \cdot S'_1 \cdot S_0 \cdot D_5 + S_2 \cdot S_1 \cdot S'_0 \cdot D_6 + S_2 \cdot S_1 \cdot S_0 \cdot D_7$

ડાયાગ્રામ: 8:1 MUX:

```
flowchart TD
    D0["D_{0}"] --> MUX1["8:1 MUX"]
    D1["D_{1}"] --> MUX1
    D2["D_{2}"] --> MUX1
    D3["D_{3}"] --> MUX1
    D4["D_{4}"] --> MUX1
    D5["D_{5}"] --> MUX1
    D6["D_{6}"] --> MUX1
    D7["D_{7}"] --> MUX1
    S0["S_{0}"] --> MUX2["8:1 MUX"]
    S1["S_{1}"] --> MUX2
    S2["S_{2}"] --> MUX2
    MUX1 --> Y["Y"]
    MUX2 --> Y
```

મેમરી ટ્રીક

મલ્ટિપ્લેક્સર ઓપરેશન માટે “સિલેક્ટ ડિસાઇડ્સ ડેટા આઉટપુટ”.

પ્રશ્ન 3(અ) અથવા [3 માંકર્સ]

હાફ સબટ્રૈક્ટરની લોજિક સર્કિટ દોરો અને તેનું કાર્ય સમજાવો.

જવાબ

હાફ સબટ્રૈક્ટર:

ફુલ્ફાળ	વર્ણન	ટુથ ટેબલ
હાફ સબટ્રૈક્ટર	બે બિટ્સને બાદ કરે છે અને ડિફરન્સ અને બોરો આપે છે	A B 0 0

0 1
1 0
1 1

લોજિક સર્કિટ:

```
flowchart TD
    A["A"] --{-{-}} XOR[""]
    B["B"] --{-{-}} XOR
    XOR --{-{-}} D["D = A"]
    A1["A{"} --{-{-}} AND["•"]
    B1["B{"} --{-{-}} AND
    AND --{-{-}} Bout["Bout = A•B"]
```

સમીકરણો:

- ડિફરન્સ (D) = $A \oplus B$
- બોરો આઉટ (Bout) = $A' \cdot B$

મેમરી ટ્રીક

હાફ સબટ્રોક્ટર ઓપરેશન માટે "ડિફરન્ટ બિટ્સ બોરો".

પ્રશ્ન 3(બ) અથવા [4 માંકર્સ]

ટુથ ટેબલ અને સર્કિટ ડાયાગ્રામ સાથે 3:8 ડીકોડર સમજાવો.

જવાબ

3:8 ડીકોડર:

ફુલ્ફલ	વર્ણન	ટુથ ટેબલ (આંશિક)
3:8 ડીકોડર	3-બિટ બાઇનરી ઇનપુટને 8 આઉટપુટ લાઇન્સમાં કન્વર્ટ કરે છે	$A_2 A_1 A_0$
	એક સમયે માત્ર એક જ આઉટપુટ એક્ટિવ	0 0 0
		0 0 1
		...
		1 1 1

સક્રિપ્ટ ડાયાગ્રામ:

```

flowchart TD
    A0["A_{0}"] --> Dec["3:8"]
    A1["A_{1}"] --> Dec
    A2["A_{2}"] --> Dec
    Dec --> Y0["Y_{0}"]
    Dec --> Y1["Y_{1}"]
    Dec --> Y2["Y_{2}"]
    Dec --> Y3["Y_{3}"]
    Dec --> Y4["Y_{4}"]
    Dec --> Y5["Y_{5}"]
    Dec --> Y6["Y_{6}"]
    Dec --> Y7["Y_{7}"]

```

સમીક્ષરણો:

- $Y_0 = A'_2 \cdot A'_1 \cdot A'_0$
- $Y_1 = A'_2 \cdot A'_1 \cdot A_0$
- ...
- $Y_7 = A_2 \cdot A_1 \cdot A_0$

મેમરી ટ્રીક

ડિકોડર ઓપરેશન માટે "બાઇનરી ઇનપુટ એક્સિટવેટ્સ આઉટપુટ".

પ્રશ્ન 3(ક) અથવા [7 માંકર્સ]

ટ્રૂ ટેબલ, સમીક્ષરણ અને સક્રિપ્ટ ડાયાગ્રામ સાથે ગ્રે થી બાઇનરી કોડ કન્વર્ટર સમજાવો.

જવાબ

ગ્રે ટુ બાઇનરી કન્વર્ટર:

ફંક્શન	વર્ણન	ટેબલ: ગ્રે ટુ બાઇનરી
ગ્રે ટુ બાઇનરી	ગ્રે કોડને બાઇનરી કોડમાં કન્વર્ટ કરે છે	ગ્રે
	બાઇનરીનો MSB ગ્રેના MSBને સમાન	0000
	દરેક બાઇનરી બિટ, હાલના ગ્રે બિટ	0001
	અને અગાઉના બાઇનરી બિટનો XOR	
	છે	0011
		0010
		0110
		...

સર્કિટ ડાયાગ્રામ:

```

flowchart LR
    G3["G_{3}"] --> B3["B_{3}"]
    G3 --> XOR1[""]
    G2["G_{2}"] --> XOR1
    XOR1 --> B2["B_{2}"]
    XOR1 --> XOR2[""]
    G1["G_{1}"] --> XOR2
    XOR2 --> B1["B_{1}"]
    XOR2 --> XOR3[""]
    G0["G_{0}"] --> XOR3
    XOR3 --> B0["B_{0}"]

```

સમીકરણો:

- $B_3 = G_3$
- $B_2 = G_3 \oplus G_2$
- $B_1 = B_2 \oplus G_1$
- $B_0 = B_1 \oplus G_0$

મેમરી ટ્રીક

ગ્રે ટુ બાઇનરી કન્વર્જન માટે "MSB સ્ટેઝ, રેસ્ટ XOR".

પ્રશ્ન 4(અ) [3 માંકર્સ]

દુધ ટેબલ અને સર્કિટ ડાયાગ્રામ સાથે D ફિલ્પ-ફ્લોપ સમજાવો.

જવાબ

D ફિલ્પ-ફ્લોપ:

ફુક્શન	વર્ણન	દુધ ટેબલ
D ફિલ્પ-ફ્લોપ	ડેટા/ડિલે ફિલ્પ-ફ્લોપ કલોક એજ પર Q, D ને ફોલો કરે છે	CLK <input type="checkbox"/> <input type="checkbox"/>

સર્કિટ ડાયાગ્રામ:

```

flowchart LR
    D["D"] --> FF["D   -   "]
    CLK["  "] --> FF
    FF --> Q["Q"]
    FF --> Qnot["Q"]

```

ક્રેકટરિસ્ટિક સમીકરણ:

- $Q(\text{next}) = D$

મેમરી ટ્રીક

D ફિલ્પ-ફ્લોપ ઓપરેશન માટે "ડેટા ડિલેજ વન કલોક".

પ્રશ્ન 4(બ) [4 માંકર્સ]

માસ્ટર સ્લેવ JK ફિલ્પ ફ્લોપનું કાર્ય સમજાવો.

જવાબ

માસ્ટર-સ્લેવ JK ફિલ્પ-ફ્લોપ:

કોમ્પોનન્ટ	ઓપરેશન	ટુથ ટેબલ
માસ્ટર	CLK = 1 હોય ત્યારે ઇનપુટ્સને સેમ્પલ કરે છે	J K
સ્લેવ	CLK = 0 હોય ત્યારે માસ્ટર આઉટપુટને ટ્રાન્સફર કરે છે	0 0 0 1 1 0 1 1

ડાયાગ્રામ: માસ્ટર-સ્લેવ JK:

```
flowchart LR
    J["J"] --{-} Master["    JK"]
    K["K"] --{-} Master
    CLK["    "] --{-} Master
    CLK{ } --{-} Slave["    JK"]
    Master --{-} Slave
    Slave --{-} Q["Q"]
    Slave --{-} Q["Q"]
```

કાર્યપદ્ધતિ:

- માસ્ટર સ્ટેજ: કલોક હાઇ હોય ત્યારે ઇનપુટ કેપ્ચર કરે છે
- સ્લેવ સ્ટેજ: કલોક લો હોય ત્યારે આઉટપુટ અપડેટ કરે છે
- રેસ કન્ડિશન અટકાવે છે ઇનપુટ કેપ્ચર અને આઉટપુટ અપડેટને અલગ કરીને

મેમરી ટ્રીક

માસ્ટર-સ્લેવ ઓપરેશન માટે "માસ્ટર સેમ્પલ્સ, સ્લેવ ટ્રાન્સફર્સ".

પ્રશ્ન 4(ક) [7 માકર્સ]

બ્લોક ડાયાગ્રામની મદદથી શિફ્ટ રજિસ્ટરનું વર્ગીકરણ કરો અને તેમાંના કોઈપણ એકને વિગતવાર સમજાવો.

જવાબ

શિફ્ટ રજિસ્ટર વર્ગીકરણ:

પ્રકાર	વર્ણન	ફૂકશન
SISO	સિરિયલ ઇન સિરિયલ આઉટ	ડેટા સિરિયલી, બિટ દર બિટ, એન્ટર થાય છે અને એક્ઝિટ થાય છે
SIPO	સિરિયલ ઇન પેરેલલ આઉટ	ડેટા સિરિયલી એન્ટર થાય છે, પેરેલલમાં એક્ઝિટ થાય છે
PISO	પેરેલલ ઇન સિરિયલ આઉટ	ડેટા પેરેલલમાં એન્ટર થાય છે, સિરિયલી એક્ઝિટ થાય છે
PIPO	પેરેલલ ઇન પેરેલલ આઉટ	ડેટા પેરેલલમાં એન્ટર થાય છે અને પેરેલલમાં એક્ઝિટ થાય છે

SIPO શિક્ષટ રજિસ્ટર વિગતવાર:

```

flowchart LR
    Din[""] --> FF1["FF_{1}"]
    FF1 --> FF2["FF_{2}"]
    FF2 --> FF3["FF_{3}"]
    FF3 --> FF4["FF_{4}"]
    CLK[""] --> FF1
    CLK --> FF2
    CLK --> FF3
    CLK --> FF4
    FF1 --> Q0["Q_{0}"]
    FF2 --> Q1["Q_{1}"]
    FF3 --> Q2["Q_{2}"]
    FF4 --> Q3["Q_{3}"]

```

SIPO શિક્ષટ રજિસ્ટરનું કાર્ય:

- સિરિયલ ડેટા ડેટા ઇન પર, પ્રતિ કલોક સાયકલ એક બિટ, પ્રવેશે છે
- દરેક ફિલ્પ-ફલોપ કલોક પલ્સ પર તેની સામગ્રીને આગળના ફિલ્પ-ફલોપમાં પાસ કરે છે
- 4 કલોક સાયકલ્સ પછી, 4-બિટ ડેટા બધા ફિલ્પ-ફલોપ્સમાં સ્ટોર થાય છે
- પેરેલલ આઉટપુટ Q0-Q3 પરથી એક સાથે ઉપલબ્ધ થાય છે

SIPO માટે ટાઇમિંગ ડાયાગ્રામ:

Clock	_ _ _ _ _
Data	_ _ _ _ _ _ _
Q0	_ _ _ _ _ _ _
Q1	_ _ _ _ _ _ _ _
Q2	_ _ _ _ _ _ _ _ _
Q3	_ _ _ _ _ _ _ _ _

મેમરી ટ્રીક

SIPO ઓપરેશન માટે “સિરિયલ ઇનપુટ્સ પેરેલલ આઉટપુટ્સ”.

પ્રશ્ન 4(અ) અથવા [3 માકર્સ]

ટુથ ટેબલ અને સર્કિટ ડાયાગ્રામ સાથે SR ફિલ્પ-ફલોપ સમજાવો.

જવાબ

SR ફિલ્પ-ફલોપ:

ફૂકશન	વર્ણન	ટુથ ટેબલ
SR ફિલ્પ-ફલોપ	સેટ-રિસેટ ફિલ્પ-ફલોપ	S R
	બેઝિક મેમરી એલિમેન્ટ	0 0
		0 1
		1 0
		1 1

સર્કિટ ડાયાગ્રામ:

```

flowchart LR
    S["S"] --> NOR1["NOR1[1]"]
    QN["Q{ }"] --> NOR1
    NOR1 --> Q["Q{ }"]
    R["R{ }"] --> NOR2["NOR2[1]"]
    Q --> NOR2
    NOR2 --> QN

```

મેમરી ટ્રીક

SR ફિલપ-ફલોપ ઓપરેશન માટે "સેટ ટુ 1, રિસેટ ટુ 0".

પ્રશ્ન 4(બ) અથવા [4 માંકર્સ]

ટૂથ ટેબલ અને સર્કિટ ડાયાગ્રામ સાથે JK ફિલપ ફલોપ સમજાવો.

જવાબ

JK ફિલપ-ફલોપ:

ફક્શન	વર્ણન	ટૂથ ટેબલ
JK ફિલપ-ફલોપ	ઇમ્પ્રુંડ SR ફિલપ-ફલોપ અમાન્ય કન્ડિશન હલ કરે છે	J K
		0 0
		0 1
		1 0
		1 1

સર્કિટ ડાયાગ્રામ:

```
flowchart LR
    J["J"] --> AND1["AND1"]
    AND1 --> Qn["Q{ }"]
    Qn --> AND1
    AND1 --> OR1["OR"]
    OR1 --> K["K"]
    K --> AND2["AND2"]
    AND2 --> Q["Q{ }"]
    Q --> AND2
    AND2 --> OR2["OR"]
    OR2 --> FF["FF[D FF]"]
    FF --> CLK["CLK[ ]"]
    CLK --> FF
    FF --> Q
    FF --> Qn
```

કેરેક્ટારેસિક સમીકરણ:

$$Q(\text{next}) = J \cdot Q' + K' \cdot Q$$

મેમરી ટ્રીક

JK ફિલપ-ફલોપ સ્ટેટ્સ માટે "જમ્પ-કીપ-ટોગલ" (J=1

K=0: 1

પર જમ્પ,

J=0

K=0: સ્ટેટ જાળવવો,

J=1

K=1: ટોગલ).

પ્રશ્ન 4(ક) અથવા [7 માંકર્સ]

ટૂથ ટેબલ અને સર્કિટ ડાયાગ્રામ સાથે 4-બિટ અસિંકોન્સ અપ કાઉન્ટરનું વર્ણન કરો.

જવાબ

4-બિટ અસિંકોન્સ અપ કાઉન્ટર:

ફક્શન	વર્ણન	કાઉન્ટ સિકવન્સ
અસિંકોન્સ કાઉન્ટર	રિપલ કાઉન્ટર પણ કહેવાય છે કલોક માત્ર પહેલા FF ને ડ્રાઇવ કરે છે	0000 → 0001 → 0010 → 0011 0100 → 0101 → 0110 → 0111

દરેક FF અગાઉના FF આઉટપુટ દ્વારા ટ્રિગર થાય છે

1000 → 1001 → 1010 → 1011

1100 → 1101 → 1110 → 1111

સર્કિટ ડાયગ્રામ:

```
flowchart LR
    CLK[" "] --> JK1["JK FF_{0}"]
    JK1 --> K1["K=1"]
    K1 --> JK1
    JK1 --> Q0["Q_{0}"]
    JK1 --> J2["J=1"]
    J2 --> K2["K=1"]
    K2 --> JK2
    JK2 --> Q1["Q_{1}"]
    JK2 --> J3["J=1"]
    J3 --> K3["K=1"]
    K3 --> JK3
    JK3 --> Q2["Q_{2}"]
    JK3 --> J4["J=1"]
    J4 --> K4["K=1"]
    K4 --> JK4
    JK4 --> Q3["Q_{3}"]
```

કાર્યપદ્ધતિ:

- પહેલો FF દરેક કલોક પદ્સ પર ટોગલ થાય છે
- બીજો FF જ્યારે પહેલો FF 1 થી 0 પર જાય છે ત્યારે ટોગલ થાય છે
- ત્રીજો FF જ્યારે બીજો FF 1 થી 0 પર જાય છે ત્યારે ટોગલ થાય છે
- ચોથો FF જ્યારે ત્રીજો FF 1 થી 0 પર જાય છે ત્યારે ટોગલ થાય છે

મેમરી ટ્રીક

અસિંકોન્સ કાઉન્ટર ઓપરેશન માટે "રિપલ કેરીજ પ્રોપેરેશન ડિલે".

પ્રશ્ન 5(અ) [3 માકર્સ]

નીચેની લોજિક ફેમિલીઝની તુલના કરો: TTL, CMOS, ECL

જવાબ

લોજિક ફેમિલી કાર્યરીતાન:

પેરામીટર	TTL	CMOS	ECL
ટેકનોલોજી	બાયપોલર ટ્રાન્జિસ્ટર્સ	MOSFETs	બાયપોલર ટ્રાન્જિસ્ટર્સ
પાવર કન્યાશન	મધ્યમ	ખૂબ ઓછો	ઉચ્ચ
સ્પીડ	મધ્યમ	નીચી-મધ્યમ	ખૂબ ઉચ્ચ
નોઈજ ઇમ્પુનિટી	મધ્યમ	ઉચ્ચ	નીચી
ફેન-આઉટ	10	50+	25
સપ્લાય વોલ્ટેજ	5V	3-15V	-5.2V

મેમરી ટ્રીક

લોજિક ફેમિલીઝની તુલના માટે "ટેકનોલોજી કન્ટ્રોલ મેની ઇલેક્ટ્રિકલ કેરેક્ટરિસ્ટિક્સ".

પ્રશ્ન 5(બ) [4 માકર્સ]

કોમ્પ્લિનેશનલ અને સિક્વેન્શિયલ લોજિક સર્કિટ્સની સરખામણી કરો.

જવાબ

કોમ્પ્યુનેશનલ VS સિક્વેન્શિયલ સર્કિટ્સ:

પેરામીટર	કોમ્પ્યુનેશનલ સર્કિટ્સ	સિક્વેન્શિયલ સર્કિટ્સ
આઉટપુટ આધારિત છે	માત્ર વર્તમાન ઇનપુટ્સ પર	વર્તમાન ઇનપુટ્સ અને અગાઉની સ્ટેટ પર
મેમોરી	કોઈ મેમોરી નથી	મેમોરી એલિમેન્ટ્સ ધરાવે છે
ફીડબેક	કોઈ ફીડબેક પાથ નથી	ફીડબેક પાથ્સ ધરાવે છે
ઉદાહરણો	ઓર્ડર્સ, MUX, ડિકોડર્સ	ફિલ્પ-ફલોપ્સ, કાઉન્ટર્સ, રજિસ્ટર્સ
કલોક	કલોકની જરૂર નથી	ઘણી વાર કલોકની જરૂર પડ છે
ડિઝાઇન એપ્રોચ	ટૂથ ટેબલ્સ, K-મેપ્સ	સ્ટેટ ડાયાગ્રામ્સ, ટેબલ્સ

ડાયાગ્રામ: ક્ર્યેરિક્ઝન:

flowchart TB

```
A[" "] --{-{-}} B[" " = f( )"] 
C[" "] --{-{-}} D[" " = f( , )"]
```

મેમોરી ટ્રીક

કોમ્પ્યુનેશનલ અને સિક્વેન્શિયલ સર્કિટ્સ વચ્ચે તફાવત કરવા માટે “કરંટ ઓન્લી વિસ્તાર મેમોરી સ્ટેટ્સ”.

પ્રશ્ન 5(ક) [7 માકર્સ]

વ્યાખ્યાયિત કરો: ફેન ઇન, ફેન આઉટ, નોઇજ માર્જિન, પ્રોપેશન ડિલે, પાવર ડિસીપેશન, ફિગર ઓફ મેરિટ, રેમ

જવાબ

ડિજિટલ ઇલેક્ટ્રોનિક્સ કી ડેફીનિશન્સ:

ટર્મ	વ્યાખ્યા	ટિપ્પિકલ વેલ્યુઝ
ફેન-ઇન	લોજિક ગેટ જેટલા ઇનપુટ્સ હેન્ડલ કરી શકે તેની મહત્વમાં સંખ્યા	TTL: 2-8, CMOS: 100+
ફેન-આઉટ	સિંગલ આઉટપુટ દ્વારા જેટલા ગેટ ઇનપુટ્સ દ્રાઇવ કરી શકાય તેની મહત્વમાં સંખ્યા	TTL: 10, CMOS: 50
નોઇજ માર્જિન	અરેર થાય તે પહેલાં ઉમેરી શકાય તેવો મહત્વમાં નોઇજ વોલ્ટેજ	TTL: 0.4V, CMOS: 1.5V
પ્રોપેશન ડિલે	ઇનપુટમાં બદલાવથી આઉટપુટમાં બદલાવ થવામાં લાગતો સમય	TTL: 10ns, CMOS: 20ns
પાવર ડિસીપેશન	ઓપરેશન દરમિયાન ગેટ દ્વારા વપરાતી શક્તિ	TTL: 10mW, CMOS: 0.1mW
ફિગર ઓફ મેરિટ	સ્પીડ અને પાવરનો ગુણાકાર (ઓછો વધુ સારો)	TTL: 100pJ, CMOS: 2pJ
RAM	રેન્ડમ એક્સેસ મેમોરી - ટેમ્પરરી સ્ટોરેજ ડિવાઇસ	પ્રકાર: SRAM, DRAM

ડાયાગ્રામ: ડિજિટલ પેરામીટર રિલેશનશિપ્સ:

flowchart LR

```
A[" "] --{-{-}} "[-{-}B[" "]] 
C[" "] --{-{-}} "[-{-}D[" "]] 
B{--{-}x"--{-}E[" "]} 
D{--{-}x"--{-}E}
```

મેમોરી ટ્રીક

પેરામીટર ટમર્સ યાદ રાખવા માટે “ફાસ્ટ પાવર નીડ્સ પ્રોપર ફિગર રેટિંગ્સ”.

પ્રશ્ન 5(અ) અથવા [3 માક્સ્સ]

ડિજિટલ ICના ઇ-વેસ્ટ મેનેજમેન્ટના પગલાં અને જરૂરિયાતનું વર્ણન કરો.

જવાબ

ડિજિટલ ICs માટે ઇ-વેસ્ટ મેનેજમેન્ટ:

સ્ટેપ	વર્ણન	મહત્વ
કલેક્શન	ઇલેક્ટ્રોનિક વેરટનું અલગ કલેક્શન	અયોગ્ય ડિસ્પોઝલને રોકે છે
સેચેનેશન	ICને અન્ય કોમ્પોનન્ટ્સથી અલગ કરવું	ટાર્ગેટ રિસાયક્લિંગ શક્ય બનાવે છે
ડિસમેન્ટલિંગ	હાનિકારક ભાગોને ફૂર કરવા	પર્યાવરણીય નુકસાન ઘટાડે છે
રિકવરી	મૂલ્યવાન મટીરિયલ્સ (ગોલ્ડ, સિલિકોન) એક્સટ્રેક્ટ કરવા	સંસાધનો બચાવે છે
સેફ ડિસ્પોઝલ	નોન-રિસાયક્લેબલ પાર્ટ્સનો યોગ્ય નિકાલ	પ્રદૂષણ અટકાવે છે

ઇ-વેસ્ટ મેનેજમેન્ટની જરૂરિયાત:

- હાનિકારક મટીરિયલ્સ: IC લેડ, મકર્ચરી, કેડમિયમ ધરાવે છે
- રિસોર્સ કન્જર્વેશન: કિમતી ધાતુઓ અને દુર્લભ સામગ્રી પુનઃપ્રાપ્ત કરે છે
- પર્યાવરણ સંરક્ષણ: જમીન અને પાણીના પ્રદૂષણને રોકે છે
- હેલ્થ સેફ્ટી: તેરી પદાર્થોના સાંપર્કને ઘટાડે છે

મેમરી ટ્રીક

ઇ-વેસ્ટ મેનેજમેન્ટ સ્ટેપ્સ માટે ``કલેક્શન સ્ટાર્ટ્સ ડિસમેન્ટલિંગ રિકવરી સેફલી''.

પ્રશ્ન 5(બ) અથવા [4 માક્સ્સ]

સાંક્રાન્તિક ડાયાગ્રામ સાથે રીંગ કાઉન્ટરનું કામ સમજાવો.

જવાબ

રીંગ કાઉન્ટર:

ફુંક્શન	વર્ણન	કાઉન્ટ સિક્વિન્સ
રીંગ કાઉન્ટર	સિંગલ 1 સાથે સક્ર્યુલર શિફ્ટ રજિસ્ટર કોઈપણ સમયે માત્ર એક જ ફિલિપ-ફ્લોપ સેટ થયેલ હોય છે N સ્ટેપ્સ માટે N ફિલિપ-ફ્લોપ્સ	1000 → 0100 → 0010 → 0001 → 1000