

Subject Name (Gujarati)

4361102 -- Summer 2024

Semester 1 Study Material

Detailed Solutions and Explanations

પ્રશ્ન 1(અ) [3 ગુણ]

FinFET ની રૂચના દોરો અને તેના ફાયદા લખો.

જવાબ

Mermaid Diagram (Code)

```
{Shaded}  
{Highlighting} []  
graph LR  
    A[Source] --- B[Gate]  
    B --- C[Drain]  
    D[Fin Structure] --- E[Multiple Gates]  
    F[Silicon Substrate] --- D  
{Highlighting}  
{Shaded}
```

Table 1: FinFET ના ફાયદા

| ફાયદો | વર્ણન |
|-----------------|--|
| બેહતર નિયંત્રણ | ગુણાકાર gates બહેતર channel control આપે છે |
| ઘટાડેલ લીકેજ | 3D રૂચનાના કારણે ઓફ્સ્ટેટ કરીએલ લીકેજ ઘટાડેલ હોય |
| સુધારેલ કામગીરી | વધુ drive current અને ઝડપી switching |

ચાદાશત ટેકનિક: "BCR - Better Control Reduces leakage"

પ્રશ્ન 1(બ) [4 ગુણ]

એક્સટર્નલ બાયજ હેઠળ MOS રૂચનાનું ડેપ્લીશન અને ઇનવર્શન સમજાવો

જવાબ

Table 2: MOS બાયજ પરિસ્થિતિઓ

| બાયજ પ્રકાર | ગેટ વોલ્ટેજ | ચેનલ સ્થિતિ | ચાર્જ કેરિયર્સ |
|-------------|---------------|-------------|---------------------|
| ડેપ્લીશન | થોડું પોઝિટિવ | Depleted | Holes દૂર ધકેલાય છે |
| ઇનવર્શન | વધુ પોઝિટિવ | Inverted | Electrons આકર્ષય છે |

ડાયાગ્રામ:

| | |
|-------------------------------|--------------------------------|
| VG { 0 (Depletion) | VG 0 (Inversion)} |
| + + | |
| Gate Gate | |
| {-{-}{-}{}{-}} | {-}{-}{-}{}{-}} |
| {-} | {-}} |
| Depletion Electron | |
| Region Channel | |
| {-{-}{-}{}{-}{}{-}{}{-}{}{-}} | {-}{-}{-}{}{-}{}{-}{}{-}{}{-}} |
| p{-substrate} | p{-}substrate} |

- ડેપ્લીશન: પોઝિટિવ ગેટ વોલ્ટેજ electric field બનાવે છે જે holes ને દૂર ધક્કેલે છે
- ઇનવર્શન: વધુ વોલ્ટેજ electrons ને આકર્ષે છે અને conducting channel બનાવે છે

ચાદાશત ટેકનિક: "DI - Depletion Inverts to conducting channel"

પ્રશ્ન 1(ક) [7 ગુણ]

n-ચેનલ MOSFET ને તેની કર્ટ-વોલ્ટેજ લાક્ષણિકતાઓની મદદથી સમજાવો.

જવાબ

Table 3: MOSFET ઓપરેટિંગ વિભાગો

| વિભાગ | શરત | ફ્રેન્ન કર્ટ | લાક્ષણિકતાઓ |
|------------|-----------------|-----------------------------|------------------------|
| કટ-ઓફ | VGS < VTH | ID ≈ 0 | કોઈ conduction નથી |
| લિનિયર | VDS < VGS-VTH | ID ∝ VDS | Resistive વર્તન |
| સેચ્યુરેશન | VDS ≥ VGS – VTH | ID ∝ (VGS-VTH) ² | કર્ટ VDS પર આધારિત નથી |

Mermaid Diagram (Code)

```

{Shaded}
{Highlighting} []
graph LR
    A[Gate] {-{-}{-}{} B[n{-}channel]}
    C[Source] {-{-}{-}{} B}
    B {-{-}{-}{} D[Drain]}
    E[p{-}substrate] {-}{-}{-}{} B
{Highlighting}
{Shaded}

```

મુખ્ય સમીકરણો:

- Linear: $ID = nCox(W/L)[(VGS-VTH)VDS - VDS^2/2]$
- Saturation: $ID = (nCox/2)(W/L)(VGS-VTH)^2$
- રચના: ગેટ source અને drain વચ્ચે channel ને નિયંત્રિત કરે છે
- કામગીરી: ગેટ વોલ્ટેજ channel conductivity ને modulate કરે છે
- ઉપયોગો: ડિજિટલ switching અને analog amplification

ચાદાશત ટેકનિક: "CLS - Cut-off, Linear, Saturation regions"

પ્રશ્ન 1(ક OR) [7 ગુણ]

સ્કેલિંગ વ્યાખ્યાયિત કરો. full voltage સ્કેલિંગ સાથે constant voltage સ્કેલિંગની તુલના કરો. સ્કેલિંગના ગેરફાયદા લખો.

જવાબ

વ્યાખ્યા: સ્કેલિંગ એ ડેન્સિટી અને performance વધારવા માટે device dimensions ઘટાડવાની પ્રક્રિયા છે.

Table 4: સ્કેલિંગ તુલના

| પેરામીટર | Full Voltage Scaling | Constant Voltage Scaling |
|------------------|----------------------|--------------------------|
| વોલ્ટેજ | □ દ્વારા ઘટાડાય છે | સ્થિર રહે છે |
| પાવર ડેન્સિટી | સ્થિર | □ દ્વારા વધે છે |
| ઇલેક્ટ્રિક ફિલ્ડ | સ્થિર | □ દ્વારા વધે છે |
| પરફોર્માન્સ | બેહતર | મધ્યમ સુધારો |

ગેરફાયદા:

- શોર્ડ ચેનલ ઇફેક્ટ્સ: ચેનલ લેન્થ modulation વધે છે
- હોટ કેરિયર ઇફેક્ટ્સ: વધુ electric fields devices ને નુકસાન કરે છે
- કવોન્ટમ ઇફેક્ટ્સ: ટનલિંગ currents નોંધપાત્ર રીતે વધે છે

યાદાશ્ટ ટેકનિક: "SHQ - Short channel, Hot carriers, Quantum effects"

પ્રશ્ન 2(અ) [3 ગુણ]

CMOS ની મદદથી બે ઇનપુટ NAND ગેટ દોરો.

જવાબ

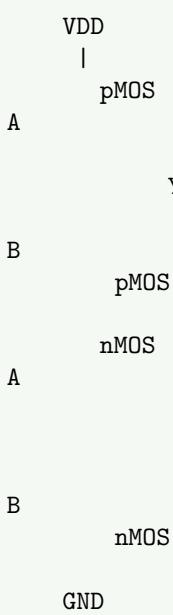


Table 5: NAND સત્ય કોષ્ક

| A | B | Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

યાદાશ્ટ ટેકનિક: "PP-SS: Parallel PMOS, Series NMOS"

પ્રશ્ન 2(બ) [4 ગુણ]

nMOS ઇન્વર્ટર માટે નોઇજ ઇમ્પુનિટી અને નોઇજ માર્જિન સમજાવો.

જવાબ

Table 6: નોઇજ પેરામીટર્સ

| પેરામીટર | વ્યાખ્યા | ફોર્મ્યુલા |
|----------------|----------------------------|---------------|
| NMH | હાઇ નોઇજ માર્જિન | VOH - VIH |
| NML | લો નોઇજ માર્જિન | VIL - VOL |
| નોઇજ ઈમ્યુનિટી | નોઇજ રિજેક્ટ કરવાની ક્ષમતા | Min(NMH, NML) |

Mermaid Diagram (Code)

```

{Shaded}
{Highlighting} []
graph TD
    A[VIL] {-{-}{}} B[VIH]
    C[VOL] {-{-}{}} D[VOH]
    E[NML] {-{-}{}} F[NMH]
{Highlighting}
{Shaded}



- VIL: મહત્તમ લો ઈનપુટ વોલ્ટેજ
- VIH: લઘૃતમ હાઇ ઈનપુટ વોલ્ટેજ
- સારી નોઇજ ઈમ્યુનિટી: મોટા નોઇજ માર્જિન ખોટી switching ને રોકે છે


ચાદશત ટેકનિક: "HIOL - High/Low Input/Output Levels"

```

પ્રશ્ન 2(ક) [7 ગુણ]

CMOS ઈન્વર્ટરની વોલ્ટેજ ટ્રાન્સફર લાક્ષણિકતાઓ (VTC) સમજાવો.

જવાબ

Table 7: VTC વિભાગો

| વિભાગ | ઈનપુટ રેન્જ | આઉટપુટ | ટ્રાન્ઝિસ્ટર સ્થિતિઓ |
|-------|--------------------|------------|----------------------|
| A | 0 to VTN | VDD | pMOS ON, nMOS OFF |
| B | VTN to VDD/2 | ટ્રાન્ઝિશન | બંને આંશિક રીતે ON |
| C | VDD/2 to VDD- VTP | ટ્રાન્ઝિશન | બંને આંશિક રીતે ON |
| D | VDD- VTP to VDD | 0V | pMOS OFF, nMOS ON |

Mermaid Diagram (Code)

```

{Shaded}
{Highlighting} []
graph TD
    A[VIN = 0] {-{-}{}} B[VOUT = VDD]
    C[VIN = VDD/2] {-{-}{}} D[VOUT = VDD/2]
    E[VIN = VDD] {-{-}{}} F[VOUT = 0]
{Highlighting}
{Shaded}

```

મુખ્ય લક્ષણો:

- તીક્ષ્ણ ટ્રાન્ઝિશન: આદર્શ switching વર્તન
- હાઇ ગેઇન: ટ્રાન્ઝિશન વિભાગમાં મોટો slope
- રેઇલ-ટુ-રેઇલ: આઉટપુટ સંપૂર્ણ સપ્લાય રેન્જમાં swing કરે છે

ચાદશત ટેકનિક: "ASH - A-region, Sharp transition, High gain"

પ્રશ્ન 2(અ OR) [3 ગુણ]

ડિપ્લીશન લોડ nMOS નો ઉપયોગ કરીને NOR2 ગેટનો અમલ કરો.

જવાબ

VDD
|
Depletion Load
(VGS = 0)

Y
nMOS

A

B

nMOS

GND

Table 8: NOR2 સત્ય કોષ્ટક

| A | B | Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

ચાદાશત ટેકનિક: "DPN - Depletion load, Parallel NMOS"

પ્રશ્ન 2(બ OR) [4 ગુણ]

એન્હાન્સમેન્ટ લોડ ઇન્વર્ટર અને ડિપ્લીશન લોડ ઇન્વર્ટર વચ્ચે તફાવત શોધો.

જવાબ

Table 9: લોડ ઇન્વર્ટર તુલના

| પેરામીટર | એન્હાન્સમેન્ટ લોડ | ડિપ્લીશન લોડ |
|-----------------|-------------------|--------------|
| શ્રેષ્ઠ વોલ્ટેજ | $VT > 0$ | $VT < 0$ |
| ગેટ કનેક્શન | $VGS = VDS$ | $VGS = 0$ |
| લોજિક હાઇ | $VDD - VT$ | VDD |
| પાવર કન્યાંશન | વધુ | ઓછું |
| સ્વિંગ સ્પીડ | ધીમું | ઝડપી |

- એન્હાન્સમેન્ટ: conduction માટે પોઝિટિવ ગેટ વોલ્ટેજની જરૂર
- ડિપ્લીશન: જીરો ગેટ વોલ્ટેજ સાથે conduct કરે છે
- પરફોર્માન્સ: ડિપ્લીશન લોડ બેહતર લાક્ષણિકતાઓ આપે છે

ચાદાશત ટેકનિક: "EPDLH - Enhancement Positive, Depletion Lower power, Higher speed"

પ્રશ્ન 2(ક OR) [7 ગુણ]

ડિપ્લીશન લોડ nMOS ઇન્વર્ટરને તેના VTC સાથે સમજાવો.

જવાબ

સક્રિપ્ટ ઓપરેશન:

- લોડ ટ્રાન્ઝિસ્ટર: હંમેશા કાઉન્ટિંગ ($VGS = 0, VT < 0$)
- ડ્રાઇવર ટ્રાન્ઝિસ્ટર: ઇનપુટ વોલ્ટેજ દ્વારા નિયન્ત્રિત
- આઉટપુટ: વોલ્ટેજ ડિવાઇડર એક્શન દ્વારા નક્કી થાય છે

Mermaid Diagram (Code)

```
{Shaded}
{Highlighting} []
graph LR
    A[VIN Low] --> B[Driver OFF]
    B --> C[VOUT = VDD]
    D[VIN High] --> E[Driver ON]
    E --> F[VOUT OV]
{Highlighting}
{Shaded}
```

Table 10: ઓપરેટિંગ પોઇન્ટ્સ

| ઇનપુટ સ્થિતિ | ડ્રાઇવર | લોડ | આઉટપુટ |
|--------------|---------|-----|--------------|
| $VIN = 0$ | OFF | ON | VDD |
| $VIN = VDD$ | ON | ON | $\approx 0V$ |

VTC લાક્ષણિકતાઓ:

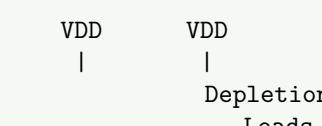
- V_{OH} : VDD (અન્હાન્સમેન્ટ લોડ કરતાં બેહતર)
- V_{OL} : ડિપ્લીશન લોડ લાક્ષણિકતાઓના કારણે ઓછું
- ટ્રાન્ઝિશન: સ્થિતિઓ વચ્ચે તીક્ષ્ણ switching

ચાદાશત ટેકનિક: "DLB - Depletion Load gives Better high output"

પ્રશ્ન 3(અ) [3 ગુણ]

ડિપ્લીશન લોડ nMOS નો ઉપયોગ કરીને EX-OR નો અમલ કરો.

જવાબ



A B

nMOS

A{ B}

Y

GND

Table 11: XOR સત્ય કોષ્ટક

| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |

$$\begin{array}{ccc} 1 & 0 & 1 \\ 1 & 1 & 0 \end{array}$$

અમલીકરણ: $Y = A = A'B + AB'$

યાદાશંકા ટેકનિક: "XOR - eXclusive OR, અલગ inputs 1 આપે છે"

પ્રશ્ન 3(બ) [4 ગુણ]

ડિજાઇન હાઇરાર્કોને ઉદાહરણ સાથે સમજાવો.

જવાબ

Table 12: હાઇરાર્કો લેવલ્સ

| લેવલ | કમ્પોનેન્ટ | ઉદાહરણ |
|--------------|--------------------|---------------|
| સિસ્ટમ | સંપૂર્ણ ચિપ | માઇક્રોપોસેસર |
| મોડ્યુલ | ફંક્શનલ બ્લોક્સ | ALU, મેમરી |
| ગેટ | લોજિક ગેટ્સ | NAND, NOR |
| ટ્રાન્ઝિસ્ટર | વ્યક્તિગત ડિવાઇસેસ | MOSFET |

Mermaid Diagram (Code)

```
{Shaded}
{Highlighting} []
graph LR
    A[ ] --- B[ ]
    B --- C[ ]
    C --- D[ ]
    E[CPU] --- F[ALU]
    F --- G[Adder]
    G --- H[MOSFET]
{Highlighting}
{Shaded}
```

ફાયદા:

- મોડ્યુલારિટી: સ્વતંત્ર ડિજાઇન અને ટેસ્ટિંગ
- પુનર્નિર્માણ: સામાન્ય બ્લોક્સ ઘણી વખત વપરાય છે
- જાળવણીધોળ્યતા: સરળ debugging અને modification

યાદાશંકા ટેકનિક: "SMG-T: System, Module, Gate, Transistor levels"

પ્રશ્ન 3(ક) [7 ગુણ]

Y ચાર્ટ ડિજાઇન ફ્લો દોરો અને સમજાવો.

જવાબ

Mermaid Diagram (Code)

```
{Shaded}
{Highlighting} []
graph LR
    A[ ] --- D[ ]
    B[ ] --- E[ ]
    C[ ] --- F[ ]
    D --- G[ ]
    E --- H[ ]
```

```

F {-{-}{}} I[   ]
G {-{-}{}} J[RTL]
H {-{-}{}} K[    ]
I {-{-}{}} L[      ]
{Highlighting}
{Shaded}

```

Table 13: Y-ચાર્ટ ડોમેન્સ

| ડોમેન | વર્ણન | ઉદાહરણો |
|-------------|------------------------------|--------------------|
| બિહેવિયરલ | સિસ્ટમ શું કરે છે | એલગોરિધમ્સ, RTL |
| સ્ક્ર્યુલરલ | તે કેવી રીતે ગોઠવાયેલું છે | આર્કિટેક્ચર, ગેટ્સ |
| ફિઝિકલ | કમ્પોનન્ટ્સ ક્રાંતિકાયેલા છે | ફલોરપ્લાન, લેઆઉટ |

ડિઝાઇન ફ્લો:

- ટોપ-ડાઉન: બિહેવિયરલ \rightarrow \rightarrow
- બોટમ-અપ: ફિઝિકલ constraints ઉપરના લેવલ્સને પ્રભાવિત કરે છે
- પુનરાવર્તી: ઓપ્ટિમાઇઝેશન માટે બહુવિધ passes

યાદાશંકાની ટેકનિક: "BSP - Behavioral, Structural, Physical domains"

પ્રશ્ન 3(અ OR) [3 ગુણ]

CMOS નો ઉપયોગ કરીને NAND2 - SR લેચનો અમલ કરો

જવાબ

S Q

NAND

NAND

R

Q{}

Table 14: SR લેચ ઓપરેશન

| S | R | Q | Q' | સ્થિતિ |
|---|---|---|----|--------|
| 0 | 0 | Q | Q' | હોલ્ડ |
| 0 | 1 | 0 | 1 | રીસેટ |
| 1 | 0 | 1 | 0 | સેટ |
| 1 | 1 | 1 | 1 | અમાન્ય |

યાદાશંકાની ટેકનિક: "SR-HRI: Set, Reset, Hold, Invalid states"

પ્રશ્ન 3(બ OR) [4 ગુણ]

સિલિકોન વેફર પર પેટર્ન અથવા માસ્ક ટ્રાન્સફર કરવા માટે કઈ પદ્ધતિનો ઉપયોગ થાય છે? તેને સ્વરચ્છ આકૃતિઓ સાથે સમજાવો.

જવાબ

પદ્ધતિ: લિથોગ્રાફી - પ્રકાશ એક્સપોઝર વાપરીને પેટર્ન ટ્રાન્સફર

Mermaid Diagram (Code)

```

{Shaded}
{Highlighting} []
graph LR
    A[UV] --> B[ ]
    B --> C[ ]
    C --> D[ ]
    D --> E[ ]
{Highlighting}
{Shaded}

```

પ્રક્રિયાના પગલાં:

| પગલું | ક્રિયા | પરિણામ |
|------------|--------------------------|------------------|
| કોટિંગ | ફોટોરેસિસ્ટ લગાવો | સમાન સ્તર |
| એક્સપોઝર | માસ્ક દ્વારા UV | રાસાયણિક ફેરફાર |
| ડેવલપમેન્ટ | એક્સપોઝર રેસિસ્ટ દૂર કરો | પેટર્ન ટ્રાન્સફર |

ઉપયોગો: ગેટ્સ, interconnects, contact holes બનાવવા

ચાદાશત ટેકનિક: "CED - Coating, Exposure, Development"

પ્રશ્ન 3(ક OR) [7 ગુણ]

MOSFET ફેલિકેશનમાં મેટલ deposit કરવા માટે કઈ પદ્ધતિઓનો ઉપયોગ થાય છે? યોગ્ય ડાયાગ્રામ સાથે ડિપોઝિશનને વિગતવાર સમજાવો.

જવાબ

Table 15: મેટલ ડિપોઝિશન પદ્ધતિઓ

| પદ્ધતિ | ટેકનિક | ઉપયોગ |
|------------------------|-------------------------|--------------------|
| ફિલ્ડિંગ વેપર ડિપોઝિશન | Sputtering, Evaporation | એલ્યુમિનિયમ, કોપર |
| કેમિકલ વેપર ડિપોઝિશન | CVD, PECVD | ટાઇટનિયમ |
| ઇલેક્ટ્રોલોટીંગ | ઇલેક્ટ્રોકેમિકલ | કોપર interconnects |

Mermaid Diagram (Code)

```

{Shaded}
{Highlighting} []
graph LR
    A[ ] --- B[ ]
    B --- C[ ]
    C --- D[ ]
    E[ ] --- D
{Highlighting}
{Shaded}

```

સ્પર્ટિંગ પ્રક્રિયા:

- આયન બોમ્બાર્ડમેન્ટ: આગોન આયન-સ ટાર્ગેટ મટેરિયલને અથડાવે છે
- એટમ ઇજેક્શન: ટાર્ગેટ એટમ્સ બહાર નીકળે છે
- ડિપોઝિશન: એટમ્સ વેફર સપાટી પર સ્થિર થાય છે
- નિયંત્રણ: દબાણ અને પાવર દર નક્કી કરે છે

કાયદા:

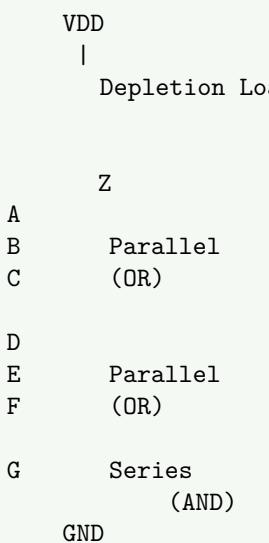
- સમાન જાડાઈ: ઉત્તમ સ્ટેપ કવરેજ
- ઓફ્સ્ટ તાપમાન: ડિવાઇસ integrity જાળવે છે
- વિવિધતા: બહુવિધ મટેરિયલ્સ શક્ય

ચાદશાત ટેકનિક: "IBE-DC: Ion Bombardment Ejects atoms for Deposition Control"

પ્રશ્ન 4(અ) [3 ગુણ]

ડિલીશન nMOS લોડ સાથે $Z = ((A+B+C) \cdot (D+E+F))' \cdot G'$ અમલમાં મૂકો.

જવાબ



લોજિક અમલીકરણ:

- પ્રથમ સ્તર: $(A+B+C)$ અને $(D+E+F)$ OR ફંક્શન્સ
- બીજું સ્તર: G સાથે AND
- આઉટપુટ: nMOS રચનાના કારણે ઉલટાવેલું પરિણામ

ચાદશાત ટેકનિક: "POI - Parallel OR, Inversion at output"

પ્રશ્ન 4(બ) [4 ગુણ]

VERILOG માં વપરાતી ડિજાઇન શૈલીઓની સૂચિ બનાવો અને સમજાવો.

જવાબ

Table 16: વેરિલોગ ડિજાઇન શૈલીઓ

| શૈલી | વર્ણન | ઉપયોગનો કેસ | ઉદાહરણ |
|------------|-------------------------|----------------------|--------------------|
| બિહેવિયરલ | એલગોરિધમ વર્ણન | ઉચ્ચ-સ્તરીય મોડેલિંગ | always blocks |
| ડેટાફ્લો | બૂલિયન expressions | કમ્પનેશનલ લોજિક | assign statements |
| સ્ટ્રક્ચરલ | કમ્પોનન્ટ instantiation | હાઇરાર્કિકલ ડિજાઇન | module connections |
| ગેટ-લેવલ | પ્રિમિટિવ ગેટ્સ | લો-લેવલ ડિજાઇન | and, or, not gates |

લાખાણિકતાઓ:

- બિહેવિયરલ: સર્કિટ શું કરે છે તેનું વર્ણન
- સ્ટ્રક્ચરલ: કમ્પોનન્ટ્સ કેવી રીતે જોડાય છે તે બતાવે છે
- મિશ્રિત: જાટિલ ડિજાઇન માટે બહુવિધ શૈલીઓ જોડે છે

યાદાશંકા ટેકનિક: "BDSG - Behavioral, Dataflow, Structural, Gate-level"

પ્રશ્ન 4(ક) [7 ગુણ]

CMOS નો ઉપયોગ કરીને NAND2 SR લેચનો અમલ કરો અને CMOS નો ઉપયોગ કરીને NOR2 SR લેચનો પણ અમલ કરો.

જવાબ

NAND2 SR લેચ:

```
module nand_sr_latch(
    input S, R,
    output Q, Q_bar
);
    nand(Q, S, Q_bar);
    nand(Q_bar, R, Q);
endmodule
```

NOR2 SR લેચ:

```
module nor_sr_latch(
    input S, R,
    output Q, Q_bar
);
    nor(Q_bar, R, Q);
    nor(Q, S, Q_bar);
endmodule
```

Table 17: લેચ તુલના

| પ્રકાર | સર્કિય સ્તર | સેટ ઓપરેશન | રીસેટ ઓપરેશન |
|--------|-------------|------------|--------------|
| NAND | લો (0) | S=0, R=1 | S=1, R=0 |
| NOR | હાઇ (1) | S=1, R=0 | S=0, R=1 |

મુખ્ય તફાવતો:

- NAND: લો ઇનપુટ્સ સાથે Set/Reset
- NOR: હાઇ ઇનપુટ્સ સાથે Set/Reset
- ફીડબેક: કોસ-કપાડ ગેટ્સ સ્થિત જાળવે છે

યાદાશંકા ટેકનિક: "NAND-Low, NOR-High active"

પ્રશ્ન 4(અ) OR) [3 ગુણ]

$Y = (ABC + DE + F)'$ ને ડિપ્લિશન નMOS લોડ સાથે અમલમાં મુકો.

જવાબ

VDD
|
Depletion Load

Y
A Series (AND)
C
D
E Series (AND)
F Single

GND

અમલીકરણ લોજિક:

- ABC: સીરિઝ કનેક્શન (AND ફૂક્શન)
- DE: સીરિઝ કનેક્શન (AND ફૂક્શન)

- F: સિંગલ ટ્રાન્ઝિસ્ટર
- પરિણામ: ઇનવર્શનના કારણે $Y = (ABC + DE + F)'$

યાદાશત ટેકનિક: "SSS-I: Series-Series-Single with Inversion"

પ્રશ્ન 4(બ OR) [4 ગુણ]

કુલ એડરને અમલમાં મૂકવા માટે વેરિલોગ કોડ લખો.

જવાબ

```
module full_adder(
    input a, b, cin,
    output sum, cout
);
    assign sum = a ^ b ^ cin;
    assign cout = (a & b) | (cin & (a ^ b));
endmodule
```

Table 18: કુલ એડર સત્ય કોષ્ટક

| A | B | Cin | Sum | Cout |
|---|---|-----|-----|------|
| 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 |

લોજિક ફૂક્શન્સ:

- સમ: ટ્રિપલ XOR ઓપરેશન
- કેરી: ઇનપુટ્સનું મેજોરિટી ફૂક્શન

યાદાશત ટેકનિક: "XOR-Sum, Majority-Carry"

પ્રશ્ન 4(ક) OR) [7 ગુણ]

ડિલીશન લોડનો ઉપયોગ કરીને $Y = (S1'S0'I0 + S1'S0 I1 + S1 S0' I2 + S1 S2 I3)$ લાગુ કરો

જવાબ

નોંધ: છેલ્લા ટર્મમાં $S2$ એ $S0$ હોવું જોઈએ.

```
// 4:1
module mux\_4to1(
    input [1:0] sel, // S1, S0
    input [3:0] data, // I3, I2, I1, I0
    output Y
);
assign
    Y = (sel == 2{b00}) ? data[0] :
        (sel == 2{b01}) ? data[1] :
        (sel == 2{b10}) ? data[2] :
            data[3];
endmodule
```

Table 19: માલિટિપ્લિક્સર સિલેક્શન

| S1 | S0 | પસંદ કરેલ ઇનપુટ | આઉટપુટ |
|----|----|-----------------|----------|
| 0 | 0 | I0 | $Y = I0$ |
| 0 | 1 | I1 | $Y = I1$ |
| 1 | 0 | I2 | $Y = I2$ |
| 1 | 1 | I3 | $Y = I3$ |

સર્કિટ અમલીકરણ:

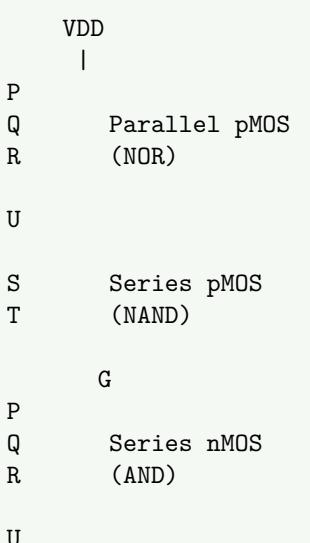
- ડીકોડર: $S1, S0$ select signals જનરેટ કરે છે
- AND ગેટ્સ: દરેક ઇનપુટ સંબંધિત select સાથે ANDed
- OR ગેટ: બધા AND આઉટપુટ્સને જોડે છે

ચાદાશત ટેકનિક: "DAO - Decoder, AND gates, OR combination"

પ્રશ્ન 5(અ) [3 ગુણ]

CMOS નો ઉપયોગ કરીને લોજિક ફક્કશન $G = (PQR + U(S+T))'$ નો અમલ કરો

જવાબ



S Parallel nMOS

T (OR)

GND

અમલીકરણ:

- pMOS: OR માટે Parallel, AND માટે Series (ઉલ્લિય લોજિક)
- nMOS: AND માટે Series, OR માટે Parallel (સામાન્ય લોજિક)
- પરિણામ: ડી મોર્ગન્નો નિયમ આપોઆપ લાગુ થાય છે

ચાદાશત ટેકનિક: "PSSP - Parallel Series Series Parallel"

પ્રશ્ન 5(બ) [4 ગુણ]

વેરિલોગનો ઉપયોગ કરીને 8×1 .

જવાબ

```
module mux\_8to1(
    input [2:0] sel,      // 3{-bit select}
    input [7:0] data,     // 8 data inputs
    output reg Y         // Output
);
    always @(*) begin
        case(sel)
            3{b000}: Y = data[0];
            3{b001}: Y = data[1];
            3{b010}: Y = data[2];
            3{b011}: Y = data[3];
            3{b100}: Y = data[4];
            3{b101}: Y = data[5];
            3{b110}: Y = data[6];
            3{b111}: Y = data[7];
        endcase
    end
endmodule
```

Table 20: 8:1 MUX સિલેક્શન

| S2 | S1 | S0 | આઉટપુટ |
|----|----|----|---------|
| 0 | 0 | 0 | data[0] |
| 0 | 0 | 1 | data[1] |
| 0 | 1 | 0 | data[2] |
| 0 | 1 | 1 | data[3] |
| 1 | 0 | 0 | data[4] |
| 1 | 0 | 1 | data[5] |
| 1 | 1 | 0 | data[6] |
| 1 | 1 | 1 | data[7] |

ચાદાશત ટેકનિક: "Case-Always: always block માં case statement વાપરો"

પ્રશ્ન 5(ક) [7 ગુણ]

વેરિલોગમાં સ્ટ્રક્ચરલ મોડેલિંગ શૈલીનો ઉપયોગ કરીને 4 બીટ ફુલ એડરને લાગુ કરો.

```

module full\_adder\_4bit(
    input [3:0] a, b,
    input cin,
    output [3:0] sum,
    output cout
);
    wire c1, c2, c3;

    full\_adder fa0(.a(a[0]), .b(b[0]), .cin(cin),
                    .sum(sum[0]), .cout(c1));
    full\_adder fa1(.a(a[1]), .b(b[1]), .cin(c1),
                    .sum(sum[1]), .cout(c2));
    full\_adder fa2(.a(a[2]), .b(b[2]), .cin(c2),
                    .sum(sum[2]), .cout(c3));
    full\_adder fa3(.a(a[3]), .b(b[3]), .cin(c3),
                    .sum(sum[3]), .cout(cout));
endmodule

module full\_adder(
    input a, b, cin,
    output sum, cout
);
    assign sum = a \^{} b \^{} cin;
    assign cout = (a \& b) | (cin \& (a \^{} b));
endmodule

```

સ્રોકચરલ લક્ષણો:

- મોડ્યુલ instantiation: ચાર 1-બીટ ફૂલ એડર્સ
- કેરી ચેઇન: સ્ટેજો વચ્ચે carries કનેક્ટ કરે છે
- હાઇરાર્કલ ડિઝાઇન: બેસિક ફૂલ એડર મોડ્યુલનો પુન:ઉપયોગ

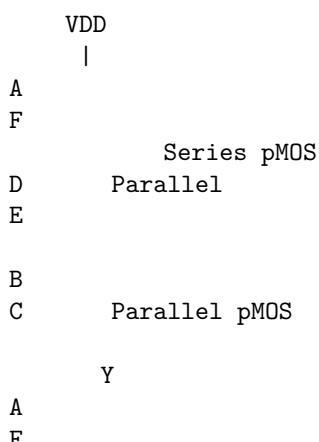
Table 21: રિપલ કેરી એડિશન

| સ્ટેજ | ઇનપુટ્સ | કેરી ઇન | સમ | કેરી આઉટ |
|-------|------------|---------|------|----------|
| FA0 | A[0], B[0] | Cin | S[0] | C1 |
| FA1 | A[1], B[1] | C1 | S[1] | C2 |
| FA2 | A[2], B[2] | C2 | S[2] | C3 |
| FA3 | A[3], B[3] | C3 | S[3] | Cout |

યાદાશત ટેકનિક: "RCC - Ripple Carry Chain connection"

પ્રશ્ન 5(અ OR) [3 ગુણ]

CMOS નો ઉપયોગ કરીને લોજિક ફંક્શન $Y = ((AF(D + E)) + (B + C))'$ ને અમલમાં મૂકો.



```

    Series nMOS
D      Parallel
E

B
C      Parallel nMOS

GND

```

લોજિક વિભાજન:

- આંતરિક અભિવ્યક્તિ: $AF(D + E) = A \text{ AND } F \text{ AND } (D \text{ OR } E)$
- બાહ્ય અભિવ્યક્તિ: $(B + C) = B \text{ OR } C$
- અંતિમ: $Y = (AF(D + E) + (B + C))'$

CMOS અમલીકરણ:

- PMOS નેટવર્ક: ફુંક્શનનું complement અમલ કરે છે
- NMOS નેટવર્ક: મૂળ ફુંક્શન અમલ કરે છે
- પરિણામ: કુદરતી inversion Y આપે છે

ચાદાશત ટેકનિક: "PNAI - PMOS Network Applies Inversion"

પ્રશ્ન 5(બ) OR) [4 ગુણ]

વેરિલોગનો ઉપયોગ કરીને 4 બીટ અપ કાઉન્ટર અમલમાં મૂકવું

જવાબ

```

module counter\_4bit\_up(
    input clk, reset,
    output reg [3:0] count
);
    always @(posedge clk or posedge reset) begin
        if (reset)
            count {=} 4{b0000};
        else
            count {=} count + 1;
    end
endmodule

```

Table 22: કાઉન્ટર સિકવન્સ

| કલોક | રીસેટ | કાઉન્ટ | નેકર્ટ કાઉન્ટ |
|------|-------|--------|---------------|
| □ | 1 | X | 0000 |
| □ | 0 | 0000 | 0001 |
| □ | 0 | 0001 | 0010 |
| □ | 0 | ... | ... |
| □ | 0 | 1111 | 0000 |

લક્ષણો:

- સિક્લોનસ રીસેટ: કલોક એજ પર રીસેટ
- ઓટો રોલાઓવર: $1111 \rightarrow 0000$
- 4-બીટ રેઝન: 0 થી 15 સુધી ગણે છે

ચાદાશત ટેકનિક: "SRA - Synchronous Reset with Auto rollover"

પ્રશ્ન 5(ક) OR) [7 ગુણ]

વેરિલોગમાં બિહેવિયરલ મોડેલિંગ સ્ટાર્ટલનો ઉપયોગ કરીને 3:8 ડિકોડરનો અમલ કરો

```

module decoder\_3to8(
    input [2:0] select,
    input enable,
    output reg [7:0] out
);
    always @(*) begin
        if (enable) begin
            case(select)
                3{b000}: out = 8{b00000001};
                3{b001}: out = 8{b00000010};
                3{b010}: out = 8{b00000100};
                3{b011}: out = 8{b00001000};
                3{b100}: out = 8{b00010000};
                3{b101}: out = 8{b00100000};
                3{b110}: out = 8{b01000000};
                3{b111}: out = 8{b10000000};
                default: out = 8{b00000000};
            endcase
        end else begin
            out = 8{b00000000};
        end
    end
endmodule

```

Table 23: 3:8 ડીકોડર સત્ય કોષ્ટક

| Enable | A2 | A1 | A0 | Y7 | Y6 | Y5 | Y4 | Y3 | Y2 | Y1 | Y0 |
|--------|----|----|----|----|----|----|----|----|----|----|----|
| 0 | X | X | X | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

મુખ્ય લક્ષણો:

- બિહેવિયરલ મોડેલિંગ: always ભ્લોક અને case statement વાપરે છે
- Enable કંટ્રોલ: enable = 0 હોય ત્યારે બધા આઉટપુટ્સ disabled
- વન-હોટ આઉટપુટ: એક સમયે માત્ર એક આઉટપુટ active
- 3-બીટ ઇનપુટ: 8 આઉટપુટ્સમાંથી એક પરંદ કરે છે

ઉપયોગો:

- મેમરી એડ્રેસિંગ: ચિપ select જનરેશન
- ડેટા રાઉટિંગ: ચેનલ સિલેક્શન
- કંટ્રોલ લોજિક: સ્ટેટ મશીન આઉટપુટ્સ

ચાદાશત ટેકનિક: "BEOH - Behavioral Enable One-Hot decoder"