

Detailed Solutions and Explanations

n-ચેનલ MOSFET ના ભૌતિક બંધારણનું સાફ લેબલવાળું આકૃતિ દોરો.

આકૃતિ:

મુખ્ય ઘટકો:

- **સોર્સ:** $n+$ ડોપ્ડ વિસ્તાર જે ઈલેક્ટ્રોન પૂરો પાડે છે
- **ડ્રેઇન:** $p+$ ડોપ્ડ વિસ્તાર જે ઈલેક્ટ્રોન એકત્ર કરે છે
- **ગેટ:** ચેનલને નિયંત્રિત કરતું મેટલ ઈલેક્ટ્રોડ
- **ઓક્સાઇડ:** SiO_2 ઇન્વ્યુલેટિંગ સ્તર
- **સબસ્ટ્રેટ:** p -ટાઇપ સિલિકોન બોડી

મેમરી ટ્રીક

“SOGD - સોર્સ, ઓક્સાઇડ, ગેટ, ડ્રેઇન”

એક્સટર્નલ બાયાસ હેઠળ MOS ના ડિપ્લીશન અને ઇન્વર્શનનું એનજી બેન્ડ ડાયાગ્રામ MOS બાયોસિંગ ડાયાગ્રામ સાથે દોરો. ઇન્વર્શન રીજનને વિગતવાર સમજાવો.

MOS બાયાસિંગ સર્કિટ:

```

VG
|
|      Gate
+++++++
| SiO2 |
+{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+
| p{-}type |}
+{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+
    |
    VB

```

એનર્જી બેન્ડ ડાયાગ્રામ:

બાયાસ સ્થિતિ	એનર્જી બેન્ડ વર્તન
ડિપ્લીશન	બેન્ડ ઉપરની તરફ વળે છે, હોલ્સ ખતમ થાય છે
ઇન્વર્શન	મજબૂત બેન્ડ બેન્ડિંગ, ઇલેક્ટ્રોન ચેનલ બને છે

ઇન્વર્શન રીજન વિગતો:

- મજબૂત ઇન્વર્શન: $V_G > V_T$ (થ્રેશોલ્ડ વોલ્ટેજ)
- ઇલેક્ટ્રોન ચેનલ: Si-SiO₂ ઇન્ટરફેસ પર બને છે
- ચેનલ કન્ડક્ટિવિટી: ગેટ વોલ્ટેજ સાથે વધે છે
- થ્રેશોલ્ડ શરત: સરફેસ પોટેન્શિયલ = $2\phi_F$

મેમરી ટ્રીક

“DIVE - ડિપ્લીશન, ઇન્વર્શન, વોલ્ટેજ, ઇલેક્ટ્રોન્સ”

પ્રશ્ન 1(ક) [7 ગુણ]

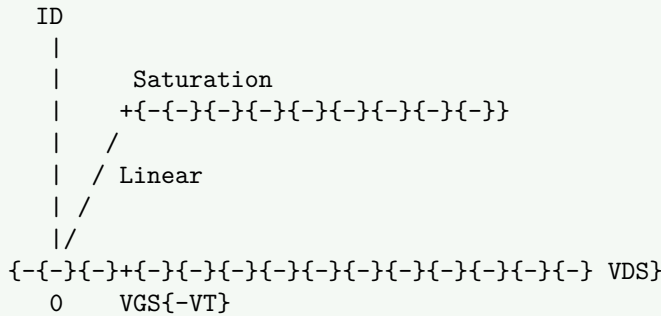
MOSFET ની I-V લાક્ષણિકતા સમજાવો.

જવાબ

I-V લાક્ષણિકતા વિસ્તારો:

વિસ્તાર	શરત	ડ્રેઇન કરંટ
કટઓફ	$V_{GS} < V_T$	$I_D \approx 0$
લિનિયર	$V_{GS} > V_T, V_{DS} < V_{GS} - V_T$	$I_D = \mu_n C_{ox} (W/L) [(V_{GS} - V_T)V_{DS} - V_{DS}^2/2]$
સેચ્યુરેશન	$V_{GS} > V_T, V_{DS} \geq V_{GS} - V_T$	$I_D = (\mu_n C_{ox}/2)(W/L)(V_{GS} - V_T)^2$

લાક્ષણિકતા વક્ર:



મુખ્ય પેરામીટર્સ:

- μ_n : ઇલેક્ટ્રોન મોબિલિટી
- C_{ox} : ગેટ ઓક્સાઇડ કેપેસિટન્સ
- W/L : પહોળાઈ અને લંબાઈનો રેશિયો
- V_T : થ્રેશોલ્ડ વોલ્ટેજ

ઓપરેટિંગ મોડ્સ:

- એન્હાન્સમેન્ટ: પોઝિટિવ V_{GS} સાથે ચેનલ બને છે
- સ્ક્વેર લો: સેચ્યુરેશન વિસ્તાર ચતુર્ભુજ સંબંધ અનુસરે છે

મેમરી ટ્રીક

“CLS - કટઓફ, લિનિયર, સેચ્યુરેશન”

પ્રશ્ન 1(ક) OR [7 ગુણ]

સ્કેલિંગ વ્યાખ્યાયિત કરો. સ્કેલિંગની જરૂરિયાત સમજાવો. સ્કેલિંગની નકારાત્મક અસરોની સૂચિ બનાવો અને સમજાવો.

જવાબ

વ્યાખ્યા: સ્કેલિંગ એટલે પર્ફોર્મન્સ અને ડેન્સિટી સુધારવા માટે MOSFET ના પરિમાણોમાં વ્યવસ્થિત ઘટાડો. સ્કેલિંગની જરૂરિયાત:

ફાયદો	વર્ણન
વધુ ડેન્સિટી	ચિપ વિસ્તાર દીઠ વધુ ટ્રાન્ઝિસ્ટર
ઝડપી સ્પીડ	ઘટેલી ગેટ ડીલે
ઓછી પાવર	ઘટેલી સ્વિચિંગ એનર્જી
કોસ્ટ રિડક્શન	વેફર દીઠ વધુ ચિપ્સ

સ્કેલિંગ પ્રકારો:

પ્રકાર	ગેટ લંબાઈ	સપ્લાય વોલ્ટેજ	ઓક્સાઇડ જાડાઈ
કોન્સ્ટન્ટ વોલ્ટેજ	□□	સ્થિર	□□
કોન્સ્ટન્ટ ફીલ્ડ	□□	□□	□□

નકારાત્મક અસરો:

- શોર્ટ ચેનલ અસરો: થ્રેશોલ્ડ વોલ્ટેજ રોલ-ઓફ
- હોટ કેરિયર અસરો: ડિવાઇસ ડીગ્રેડેશન
- ગેટ લીકેજ: વધેલી ટનલિંગ કરંટ
- પ્રોસેસ વેરિએશન્સ: મેન્યુફેક્ચરિંગ પડકારો
- પાવર ડેન્સિટી: હીટ ડિસિપેશન સમસ્યાઓ

મેમરી ટ્રીક

“SHGPP - શોર્ટ ચેનલ, હોટ કેરિયર, ગેટ લીકેજ, પ્રોસેસ, પાવર”

પ્રશ્ન 2(અ) [3 ગુણ]

CMOS નો ઉપયોગ કરીને $Y' = (AB' + A'B)$ અમલમાં મૂકો.

જવાબ

લોજિક વિશ્લેષણ: $Y' = (AB' + A'B) = A \oplus B (XOR)$

CMOS અમલીકરણ:

```

VDD
|
+{--+{-}+    +{-}+{-}+}
|pA |    |pB |
+{-{-}{-}+    +{-}{-}{-}+}
|          |
+{-{-}{-}Y{-}{-}{-}+}
|          |
+{-{-}{-}+    +{-}{-}{-}+}
|nA |    |nB{|}
+{--+{-}+    +{-}+{-}+}
|          |
GND        GND
    
```

ટ્રુથ ટેબલ:

A	B	AB'	A'B	Y'
0	0	0	0	1

0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

મેમરી ટ્રીક

“XOR ને કોમ્પ્લિમેન્ટરી સ્વિચિંગ જોઈએ”

પ્રશ્ન 2(બ) [4 ગુણ]

એન્હાન્સમેન્ટ લોડ ઇન્વર્ટરને તેના સર્કિટ ડાયાગ્રામ સાથે સમજાવો.

જવાબ

સર્કિટ ડાયાગ્રામ:

```

VDD
|
+{--}{--}o VG2}
|
+{--}{--}+ Enhancement}
|ME | Load
+{--}{--}{--}+}
|
+{--}{--}{--}o Vout}
|
+{--}{--}{--}+}
|MD | Driver
+{--}{--}{--}+}
|
GND
|
+{--}{--}{--}o Vin}

```

કન્ફિગરેશન:

ઘટક	પ્રકાર	કનેક્શન
લોડ (ME)	એન્હાન્સમેન્ટ NMOS	ગેટ VDD સાથે જોડાયેલું
ડ્રાઇવર (MD)	એન્હાન્સમેન્ટ NMOS	ગેટ ઇનપુટ છે

ઓપરેશન:

- લોડ ટ્રાન્ઝિસ્ટર: એક્ટિવ લોડ રેજિસ્ટર તરીકે કામ કરે છે
- હાઇ આઉટપુટ: લોડ ટ્રાન્ઝિસ્ટરના VT દ્વારા મર્યાદિત
- લો આઉટપુટ: ડ્રાઇવરની તાકાત પર આધાર રાખે છે
- ગેરફાયદો: થ્રેશોલ્ડ ડ્રોપ કારણે ખરાબ VOH

ટ્રાન્સફર લાક્ષણિકતાઓ:

- VOH: $VDD - VT$ (બગડેલો હાઇ લેવલ)
- VOL: ગ્રાઉન્ડ પોટેન્શિયલની નજીક
- નોઇઝ માર્જિન: થ્રેશોલ્ડ લોસ કારણે ઘટેલો

મેમરી ટ્રીક

“ELI - એન્હાન્સમેન્ટ લોડ ઇન્વર્ટરમાં થ્રેશોલ્ડ સમસ્યા”

પ્રશ્ન 2(ક) [7 ગુણ]

ઇન્વર્ટરની વોલ્ટેજ ટ્રાન્સફર કેરેક્ટરિસ્ટિક સમજાવો.

VTC પેરામીટર્સ:

પેરામીટર	વર્ણન	આદર્શ કિંમત
VOH	આઉટપુટ હાઇ વોલ્ટેજ	VDD
VOL	આઉટપુટ લો વોલ્ટેજ	0V
VIH	ઇનપુટ હાઇ વોલ્ટેજ	VDD/2
VIL	ઇનપુટ લો વોલ્ટેજ	VDD/2
VM	સ્વિચિંગ થ્રેશોલ્ડ	VDD/2

VTC વક્ર:

$$V_{out} = \frac{VDD + \frac{VM + \{ \dots \}}{0 + \{ \dots \}}}{1 + \frac{VIL}{VM} + \frac{VIH}{VDD} + \dots}$$

નોઇઝ માર્જિન્સ:

- NMH = VOH - VIH (હાઇ નોઇઝ માર્જિન)
- NML = VIL - VOL (લો નોઇઝ માર્જિન)

વિસ્તારો:

- વિસ્તાર 1: ઇનપુટ લો, આઉટપુટ હાઇ
- વિસ્તાર 2: ટ્રાન્ઝિશન વિસ્તાર
- વિસ્તાર 3: ઇનપુટ હાઇ, આઉટપુટ લો

ગુણવત્તા મેટ્રિક્સ:

- તીક્ષ્ણ ટ્રાન્ઝિશન: બહેતર નોઇઝ ઇમ્યુનિટી
- સિમેટ્રિક સ્વિચિંગ: VM = VDD/2
- કુલ સ્વિંગ: VOH = VDD, VOL = 0

મેમરી ટ્રીક

“VTC દર્શાવે છે VOH - VOL, ઇનપુટ થ્રેશોલ્ડ, લાક્ષણિકતા, બધું”

પ્રશ્ન 2(અ) OR [3 ગુણ]

CMOS નો ઉપયોગ કરીને NAND2 ગેટ સમજાવો.

CMOS NAND2 સર્કિટ:

$$VDD$$

$$+ \{ \dots \} + \{ \dots \} + \{ \dots \}$$

$$+ \{ \dots \} + \{ \dots \} + \{ \dots \}$$

$$| pA | | pB | PMOS$$

$$+ \{ \dots \} + \{ \dots \} + \{ \dots \} \text{ (Parallel)}$$

$$+ \{ \dots \} Y \{ \dots \} + \{ \dots \}$$

$$+ \{ \dots \} + \{ \dots \}$$

```

|nA | NMOS
+{--}{--}+ (Series)}
|
+{--}{--}+
|nB |
+{--}{--}+
|
GND

```

ટ્રુથ ટેબલ:

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

ઓપરેશન:

- **PMOS નેટવર્ક:** પેરેલલ કનેક્શન (પુલ-અપ)
- **NMOS નેટવર્ક:** સીરીઝ કનેક્શન (પુલ-ડાઉન)
- **આઉટપુટ લો:** ફક્ત જ્યારે બંને ઇનપુટ હાઇ

મેમરી ટ્રીક

"NAND - નોટ AND, પેરેલલ PMOS, સીરીઝ NMOS"

પ્રશ્ન 2(બ) OR [4 ગુણ]

રેજિસ્ટ્રિવ લોડ ઇન્વર્ટર સર્કિટના ઓપરેટિંગ મોડ અને VTC સમજાવો.

જવાબ

સર્કિટ કન્ફિગરેશન:

```

VDD
|
R (Load Resistor)
|
+{--}{--}o Vout}
|
+{--}{--}+
|MN | NMOS Driver
+{--}{--}+
|
GND
|
+{--}{--}o Vin}

```

ઓપરેટિંગ મોડ્સ:

ઇનપુટ સ્થિતિ	NMOS સ્થિતિ	આઉટપુટ
V_{in} = 0	OFF	VOH = VDD
V_{in} = VDD	ON	VOL = R·ID/(R+RDS)

VTC લાક્ષણિકતાઓ:

- V_{OH} : ઉત્તમ (V_{DD})
- V_{OL} : R અને R_{DS} રેશિયો પર આધાર રાખે છે
- પાવર વપરાશ: ઇનપુટ હાઇ હોય ત્યારે સ્ટેટિક કરંટ
- ટ્રાન્ઝિશન: રેજિસ્ટિવ લોડ કારણે ધીમું

ડિઝાઇન ટ્રેડ-ઓફ્સ:

- મોટો R: બહેતર V_{OL} , ધીમું સ્વિચિંગ
- નાનો R: ઝડપી સ્વિચિંગ, વધુ પાવર
- એરિયા: રેજિસ્ટર નોંધપાત્ર જગ્યા લે છે

મેમરી ટ્રીક

“RLI - રેજિસ્ટિવ લોડમાં અનિવાર્ય પાવર વપરાશ”

પ્રશ્ન 2(ક) OR [7 ગુણ]

CMOS ઇન્વર્ટર દોરો અને VTC સાથે તેની કામગીરી સમજાવો.

જવાબ

CMOS ઇન્વર્ટર સર્કિટ:

```
VDD
|
+{-+{-}+}
|MP | PMOS
+{-{-}{-}+}
|
+{-{-}{-}o Vout}
|
+{-+{-}+}
|MN | NMOS
+{-{-}{-}+}
|
GND
|
+{-{-}{-}o Vin}
```

ઓપરેશન વિસ્તારો:

Vin રેન્જ	PMOS	NMOS	Vout	વિસ્તાર
0 થી V_{TN}	ON	OFF	V_{DD}	1
V_{TN} થી $V_{DD}-V_{TP}$	VTP	**	ON	ON
$V_{DD}-V_{TP}$ થી V_{DD}	VTP	થી V_{DD} **	OFF	ON

VTC विश्लेषणः

$$\begin{array}{ccccccc}
 \text{Vout} & & & & & & \\
 | & & & & & & \\
 \text{VDD+} & & & & & & \\
 | \{ \} & & & & & & \\
 | \{ \} & & & & & & \\
 | \{ _ _ _ _ \} & & & & & & \\
 \text{VM+} & & \{ _ _ _ _ \} & & & & \\
 | & & & \{ _ _ _ _ \} & & & \\
 | & & & \{ \} & & & \\
 0 & & & + \{ - \{ - \} \{ - \} \} & & & \\
 + \{ - \{ - \} \{ - \} \{ - \} + \{ - \} \{ - \} \{ - \} \{ - \} + \{ - \} \{ - \} \{ - \} \{ - \} + \{ - \} \{ - \} \{ - \} & & & & & & \text{Vin} \} \\
 \text{VTN} & \text{VM} & \text{VTP} & \text{VDD} & & &
 \end{array}$$

મુખ્ય લક્ષણો:

- **ઝીરો સ્ટેટિક પાવર:** કોઈ DC કરંટ પાથ નથી
- **કુલ સ્વિંગ:** $V_{OH} = V_{DD}$, $V_{OL} = 0V$
- **હાઇ નોઇઝ માર્જિન-સ:** $NMH = NML \approx 0.4V_{DD}$
- **તીક્ષ્ણ ટ્રાન્ઝિશન:** ટ્રાન્ઝિશન વિસ્તારમાં હાઇ ગેઇન

ડિઝાઇન વિચારણાઓ:

- **रेशियो:** सिमेट्रिक स्विचिंग माटे N/P
- **थ्रेशोल्ड भेयिंग:** $V_{TN} \approx |V_{TP}|$

મેમરી ટ્રીક

“CMOS માં ઝીરો સ્ટેટિક પાવર અને કુલ સ્વિંગ”

પ્રશ્ન 3(અ) [3 ગુણ]

ડિપ્લીશન લોડનો ઉપયોગ કરીને $Y = (A\Box + B\Box)C\Box + D\Box + E\Box$ અમલમાં મૂકો.

ଝରାଘ

લોજિક સરળીકરણ: $Y = (A+B)C+D+E = AC+B C+D+E$
ડિપ્લીશન લોડ અમલીકરણ:

```

VDD
|
+{--+{-}+ VGS=0}
|MD | Depletion
+{-{-}{-}+ Load}
|
+{-{-}{-}o Y}
|
+{-{-}{-}+{-}{-}{-}+{-}{-}{-}+{-}{-}{-}+}
| | | | |
+{-+{-}{-}+{-}+{-}+{-}+{-}+{-}+{-}+{-}+}
|A{| |B| |C| |D| |E| Pull{-}down}
+{-{-}+ +{-}{-}+ +{-}{-}+ +{-}{-}+ +{-}{-}+ Network}
| | | | |
GND GND GND GND GND

```

પ્રુલ-ડાઉન નેટવર્ક:

- **સીરીઝ:** A□□□ પાથ અને B□□□ પાથ
- **પેરેલલ:** બધા પાથ પેરેલલમાં જોડાયેલા
- **અમલીકરણ:** યોગ્ય ટ્રાન્ઝિસ્ટર સાઈઝિંગ જરૂરી

મેમરી ટ્રીક

“ડિપ્લીશન લોડ પેરેલલ પુલ-ડાઉન પાથ સાથે”

પ્રશ્ન 3(બ) [4 ગુણ]

FPGA પર ટૂંકી નોંધ લખો.

જવાબ

FPGA વ્યાખ્યા: ફીલ્ડ પ્રોગ્રામેબલ ગેટ એરે - રીકન્ફિગરેબલ ઇન્ટિગ્રેટેડ સર્કિટ.
આર્કિટેક્ચર ઘટકો:

ઘટક	કાર્ય
CLB	કન્ફિગરેબલ લોજિક બ્લોક
IOB	ઇનપુટ/આઉટપુટ બ્લોક
ઇન્ટરકનેક્ટ	રાઉટિંગ રિસોર્સ
સ્વિચ મેટ્રિક્સ	કનેક્શન પોઇન્ટ્સ

પ્રોગ્રામિંગ ટેકનોલોજીઝ:

- SRAM-આધારિત: વોલેટાઇલ, ઝડપી રીકન્ફિગરેશન
- એન્ટીફ્યુઝ: નોન-વોલેટાઇલ, એક વખતનું પ્રોગ્રામેબલ
- ફ્લેશ-આધારિત: નોન-વોલેટાઇલ, રીપ્રોગ્રામેબલ

એપ્લિકેશન્સ:

- પ્રોટોટાઇપિંગ: ડિજિટલ સિસ્ટમ ડેવલપમેન્ટ
- DSP: સિગ્નલ પ્રોસેસિંગ એપ્લિકેશન્સ
- કંટ્રોલ સિસ્ટમ્સ: ઇન્ડસ્ટ્રિયલ ઓટોમેશન
- કોમ્યુનિકેશન્સ: પ્રોટોકોલ અમલીકરણ

ASIC સામે ફાયદા:

- લવચીકતા: રીકન્ફિગરેબલ ડિઝાઇન
- ટાઇમ-ટુ-માર્કેટ: ઝડપી વિકાસ
- કોસ્ટ: નાના વોલ્યુમ માટે ઓછો
- જોખમ: ઘટેલો ડિઝાઇન જોખમ

મેમરી ટ્રીક

“FPGA - લવચીક પ્રોગ્રામિંગ ફાયદા આપે છે”

પ્રશ્ન 3(ક) [7 ગુણ]

Y ચાર્ટ ડિઝાઇન ફ્લો દોરો અને સમજાવો.

જવાબ

Y-ચાર્ટ ડાયાગ્રામ:

```
graph TB
    subgraph "Behavioral Domain"
        B1[Algorithm]
        B2[Register Transfer]
        B3[Boolean Equations]
    end
    end

    subgraph "Structural Domain"
        S1[Processor]
        S2[ALU, Register]
        S3[Gates]
    end
    end
```

```

subgraph "Physical Domain"
  P1[Floor Plan]
  P2[Module Layout]
  P3[Cell Layout]
end

B1 {-{-} S1}
B2 {-{-} S2}
B3 {-{-} S3}
S1 {-{-} P1}
S2 {-{-} P2}
S3 {-{-} P3}

```

ડિઝાઇન ડોમેઇન-સ:

ડોમેઇન	લેવલ	વર્ણન
બિહેવિયરલ	એલ્ગોરિધમ $\rightarrow RT \rightarrow$	સિસ્ટમ શું કરે છે
સ્ટ્રક્ચરલ	પ્રોસેસર $\rightarrow ALU \rightarrow$	સિસ્ટમ કેવી રીતે બનાવેલ છે
ફિઝિકલ	ફ્લોર પ્લાન $\rightarrow \rightarrow$	ભૌતિક અમલીકરણ

ડિઝાઇન ફ્લો પ્રક્રિયા:

- ટોપ-ડાઉન: બિહેવિયરલથી શરૂ કરી ફિઝિકલ તરફ જાયો
- બોટમ-અપ: ઘટકોથી ઉપરની તરફ બનાવો
- મિક્સ્ડ એપ્રોચ: બંનેની સંયુક્ત પદ્ધતિ

એબ્સ્ટ્રેક્શન લેવલ્સ:

- સિસ્ટમ લેવલ: સૌથી વધુ એબ્સ્ટ્રેક્શન
- RT લેવલ: રજિસ્ટર ટ્રાન્સફર ઓપરેશન-સ
- ગેટ લેવલ: બુલિયન લોજિક અમલીકરણ
- લેઆઉટ લેવલ: ભૌતિક જ્યોમેટ્રી

ડિઝાઇન વેરિફિકેશન:

- હોરિઝોન્ટલ: સમાન લેવલે ડોમેઇન-સ વચ્ચે
- વર્ટિકલ: સમાન ડોમેઇનમાં લેવલ્સ વચ્ચે

મેમરી ટ્રીક

“Y-ચાર્ટ: બિહેવિયરલ, સ્ટ્રક્ચરલ, ફિઝિકલ - BSP ડોમેઇન-સ”

પ્રશ્ન 3(અ) OR [3 ગુણ]

ડિપ્લીશન લોડનો ઉપયોગ કરીને NOR2 ગેટ સમજાવો.

જવાબ

ડિપ્લીશન લોડ NOR2 સર્કિટ:

```

VDD
|
+{-+{-}+ VGS=0}
|MD | Depletion
+{-{-}{-}+ Load}
|
+{-{-}{-}o Y}
|
+{-{-}{-}+{-}{-}{-}+}
|
+{-+{-}+ +{-}{-}+}
|nA | |nB | NMOS
+{-{-}{-}+ +{-}{-}{-}+ (Parallel)}
|
GND      GND

```

ટ્રુથ ટેબલ:

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

ઓપરેશન:

- બંને ઇનપુટ લો: બંને NMOS OFF, Y = VDD
- કોઈપણ ઇનપુટ હાઇ: સંબંધિત NMOS ON, Y = VOL
- લોડ ટ્રાન્ઝિસ્ટર: પુલ-અપ કરેટ પૂરો પાડે છે

મેમરી ટ્રીક

“ડિપ્લીશન સાથે NOR - પેરેલલ NMOS પુલ-ડાઉન”

પ્રશ્ન 3(બ) OR [4 ગુણ]

કુલ કસ્ટમ અને સેમી કસ્ટમ ડિઝાઇન શૈલીઓની તુલના કરો.

જવાબ

તુલના ટેબલ:

પેરામીટર	કુલ કસ્ટમ	સેમી કસ્ટમ
ડિઝાઇન ટાઇમ	લાંબો (6-18 મહિના)	ટૂંકો (2-6 મહિના)
પર્ફોર્મન્સ	શ્રેષ્ઠ	સારું
એરિયા	લઘુત્તમ	મધ્યમ
પાવર	ઓપ્ટિમાઇઝડ	સ્વીકાર્ય
કોસ્ટ	હાઇ NRE	લોઅર NRE
લવચીકતા	મહત્તમ	મર્યાદિત
જોખમ	વધુ	ઓછું

કુલ કસ્ટમ લાક્ષણિકતાઓ:

- દરેક ટ્રાન્ઝિસ્ટર: મેન્યુઅલી ડિઝાઇન અને પ્લેસ કરેલું
- લેઆઉટ ઓપ્ટિમાઇઝેશન: મહત્તમ ડેન્સિટી હાંસલ
- એપ્લિકેશન્સ: હાઇ-વોલ્યુમ, પર્ફોર્મન્સ-ક્રિટિકલ

સેમી કસ્ટમ પ્રકારો:

- ગેટ એરે: પૂર્વ-વ્યાખ્યાયિત ટ્રાન્ઝિસ્ટર એરે
- સ્ટેન્ડર્ડ સેલ: પૂર્વ-ડિઝાઇન કરેલા સેલ્સની લાઇબ્રેરી
- FPGA: ફીલ્ડ પ્રોગ્રામેબલ લોજિક

ડિઝાઇન ફ્લો તુલના:

- કુલ કસ્ટમ: સ્પેસિફિકેશન → → →
- સેમી કસ્ટમ: સ્પેસિફિકેશન → HDL → → &

મેમરી ટ્રીક

“કુલ કસ્ટમ - મહત્તમ નિયંત્રણ, સેમી કસ્ટમ - સ્પીડ સમજૂતી”

પ્રશ્ન 3(ક) OR [7 ગુણ]

ASIC ડિઝાઇન ફ્લો વિગતવાર દોરો અને સમજાવો.

ASIC ડિઝાઇન ફ્લો:

flowchart TD

```

A[          ] {-{-} B[          ]]
B {-{-} C[RTL          ]}
C {-{-} D[          ]}
D {-{-} E[          ]}
E {-{-} F[ {-}          ]}
F {-{-} G[          ]}
G {-{-} H[          ]}
H {-{-} I[          ]}
I {-{-} J[          ]}
J {-{-} K[          ]}
K {-{-} L[          ]}
L {-{-} M[ {-}          ]}

```

ડિઝાઇન સ્ટેજો:

સ્ટેજ	વર્ણન	ટૂલ્સ/પદ્ધતિઓ
RTL ડિઝાઇન	હાર્ડવેર વર્ણન	વેરિલોગ/VHDL
સિન્થેસિસ	RTL ને ગેટ્સમાં કન્વર્ટ	લોજિક સિન્થેસિસ ટૂલ્સ
ફ્લોર પ્લાનિંગ	ચિપ એરિયા વિતરણ	ફ્લોર પ્લાનિંગ ટૂલ્સ
પ્લેસમેન્ટ	ગેટ્સ/બ્લોક્સ સ્થાન	પ્લેસમેન્ટ એલ્ગોરિધમ
રાઉટિંગ	પ્લેસ કરેલા એલિમેન્ટ્સ જોડો	રાઉટિંગ એલ્ગોરિધમ

वेरिफिकेशन स्टेप्स:

- ફંક્શનલ: RTL સિમ્યુલેશન અને વેરિફિકેશન
- ગેટ-લેવલ: પોસ્ટ-સિન્થેસિસ સિમ્યુલેશન
- ફિક્સલ: DRC, LVS, એન્ટેના ચેક્સ
- ટાઇમિંગ: સેટઅપ/હોલ્ડ વાયોલેશન માટે STA

ડિઝાઇન કન્સ્ટ્રેઇન્ટ્સ:

- **ટાઇમિંગ:** કલોક ફીક્વન્સી જરૂરિયાતો
- **એરિયા:** સિલિકોન એરિયા મર્યાદાઓ
- **પાવર:** પાવર વપરાશ લક્ષ્યો
- **ટેસ્ટ:** ટેસ્ટેબિલિટી માટે ડિઝાઇન

સાઇન-ઓફ ચેક્સ:

- **DRC:** ડિઝાઇન રૂલ ચેક
- **LVS:** લેઆઉટ વર્સીસ સ્કીમેટિક
- **STA:** સ્ટેટિક ટાઇમિંગ એનાલિસિસ
- **પાવર:** પાવર ઇન્ટેગ્રિટી એનાલિસિસ

મેમરી ટ્રીક

“ASIC ફાઇ: RTL \rightarrow \rightarrow \rightarrow ”

પ્રશ્ન 4(અ) [3 ગુણ]

CMOS સાથે લોજિક ફંક્શન $G = (A(D+E)+BC)$ ની અમલમાં મૂકો

લોજિક વિશ્લેષણ: $G = (A(D+E)+BC) \square = (AD+AE+BC) \square$

CMOS અમલીકરણ:

$$\begin{array}{ccccccc}
 & & \text{VDD} & & & & \\
 & & | & & & & \\
 +\{-\{-\}\{-\}\{-\}\{-\}+\{-\}\{-\}\{-\}\{-\}+\{-\}\{-\}\{-\}\{-\}\{-\}+\} & & & & & & \\
 | & & | & & | & & | \\
 +\{-+\{-\}+\}+ & +\{-\}+\{-\}+ & +\{-\}+\{-\}+ & +\{-\}+\{-\}+ & & &
 \end{array}$$

```

|pA | |pD | |pA | |pB | PMOS
+{-{-}{-}{-}+ +{-}{-}{-}{-}+ +{-}{-}{-}{-}+ +{-}{-}{-}{-}+ (    )}
|      |      |      |
+{-{-}{-}{-}{-}{-}+      +{-}{-}{-}{-}{-}{-}+}
|      |
+{-{-}{-}{-}{-}{-}{-}G{-}{-}{-}{-}{-}{-}+}
|
+{-{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}+}
|      |      |
+{-+{-}{-}+ +{-}{-}{-}{-}+ +{-}{-}{-}{-}+}
|nA |      |nA |      |nB | NMOS
+{-{-}{-}{-}{-}+ +{-}{-}{-}{-}{-}+ +{-}{-}{-}{-}{-}+ (    )}
|      |      |
+{-+{-}{-}+ +{-}{-}{-}{-}+      |}
|nD |      |nE |      |
+{-{-}{-}{-}{-}+ +{-}{-}{-}{-}{-}+      |}
|      |      |
GND      GND      +{-+{-}{-}+}
|nC |
+{-{-}{-}{-}{-}+}
|
GND

```

નેટવર્ક કન્ફિગરેશન:

- PMOS: કોમ્પ્લિમેન્ટનું સીરીઝ અમલીકરણ
- NMOS: મૂળ ફંક્શનનું પેરેલલ અમલીકરણ

મેમરી ટ્રીક

“કોમ્પ્લેક્સ CMOS - PMOS સીરીઝ, NMOS પેરેલલ”

પ્રશ્ન 4(બ) [4 ગુણ]

3 બિટ પેરિટી ચેકર માટે વેરિલોગ કોડ લખો.

જવાબ

વેરિલોગ કોડ:

```

module parity\_checker\_3bit(
    input [2:0] data\_in,
    output parity\_even,
    output parity\_odd
);

//
assign parity\_even = ^{data\_in};

//
assign parity\_odd = {(^){data\_in}};

//
/*
assign parity\_even = data\_in[0] ^{ data\_in[1] ^{ data\_in[2]};}
assign parity\_odd = {(data\_in[0] ^{ data\_in[1] ^{ data\_in[2]}});}
*/

endmodule

```

ટૂથ ટેબલ:

મેમરી ટ્રીક

“કોમ્પ્લેક્સ ફંક્શનને સ્ટેજ અમલીકરણ જોઈએ”

પ્રશ્ન 4(અ) OR [3 ગુણ]

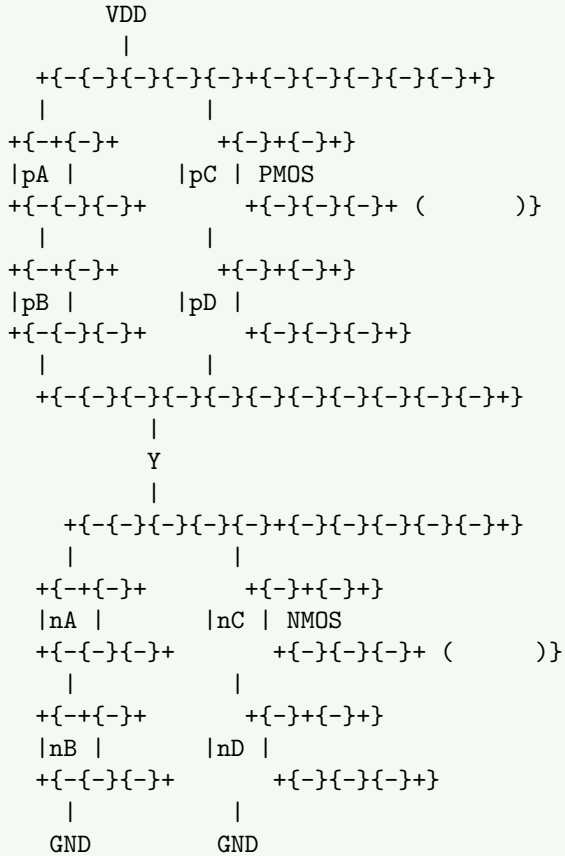
ઉદાહરણ સાથે AOI લોજિક સમજાવો.

જવાબ

AOI વ્યાખ્યા: AND-OR-Invert લોજિક આ પ્રકારના ફંક્શન અમલીકરણ કરે છે: $Y = (AB + CD + \dots)$

ઉદાહરણ: $Y = (AB + CD)$

AOI અમલીકરણ:



ફાયદા:

- સિંગલ સ્ટેજ: ડાયરેક્ટ અમલીકરણ
- ઝડપી: બહુવિધ લેવલ્સ દ્વારા પ્રોપેગેશન નહીં
- એરિયા એફિશિયન્ટ: અલગ ગેટ્સ કરતાં ઓછા ટ્રાન્ઝિસ્ટર

એપ્લિકેશન્સ:

- કોમ્પ્લેક્સ ગેટ્સ: મલ્ટિ-ઇનપુટ ફંક્શન
- સ્પીડ-ક્રિટિકલ પાથ: ઘટેલી ડીલે

મેમરી ટ્રીક

“AOI - AND-OR-Invert એક સ્ટેજમાં”

પ્રશ્ન 4(બ) OR [4 ગુણ]

4-બિટ સીરિયલ IN પેરેલલ આઉટ શિફ્ટ રજિસ્ટર માટે વેરિલોગ કોડ લખો.

વેરિલોગ કોડ:

```

module sipo\_4bit(
    input clk,
    input reset,
    input serial\_in,
    output reg [3:0] parallel\_out
);

always @(posedge clk or posedge reset) begin
    if (reset) begin
        parallel\_out {=} 4{b0000};
    end else begin
        //          LSB
        parallel\_out {=} \{parallel\_out[2:0], serial\_in\};
    end
end

endmodule

```

ટેસ્ટબેન્ચ ઉદાહરણ:

```

module tb\_sipo\_4bit;
    reg clk, reset, serial\_in;
    wire [3:0] parallel\_out;

    sipo\_4bit dut(.clk(clk), .reset(reset),
                  .serial\_in(serial\_in),
                  .parallel\_out(parallel\_out));

    initial begin
        clk = 0;
        forever \#5 clk = {~clk};
    end

    initial begin
        reset = 1; serial\_in = 0;
        \#10 reset = 0;
        \#10 serial\_in = 1; // LSB
        \#10 serial\_in = 0;
        \#10 serial\_in = 1;
        \#10 serial\_in = 1; // MSB
        \#20 $finish;
    end
endmodule

```

ઓપરેશન ટાઇમલાઇન:

ક્લોક	Serial_in	Parallel_out
1	1	0001
2	0	0010
3	1	0101
4	1	1011

મેમરી ટ્રીક

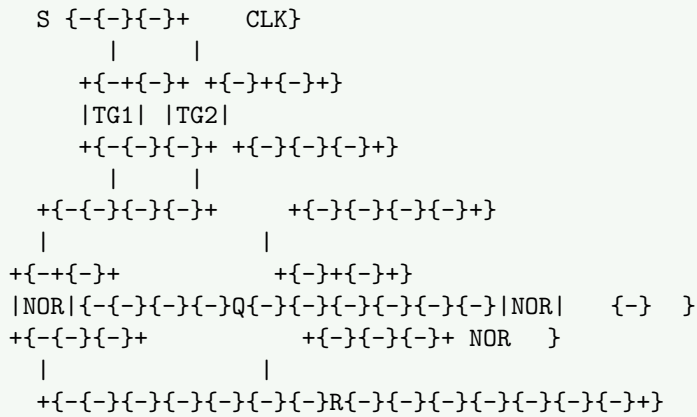
“SIPO - સીરિયલ ઇન, પેરેલલ આઉટ શિફ્ટ લેફ્ટ સાથે”

પ્રશ્ન 4(ક) OR [7 ગુણ]

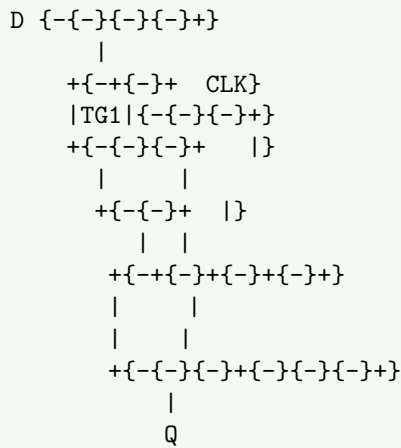
CMOS નો ઉપયોગ કરીને કલોક્સ NOR2 SR લેચ અને D-લેચ અમલીકરણ કરો.

જવાબ

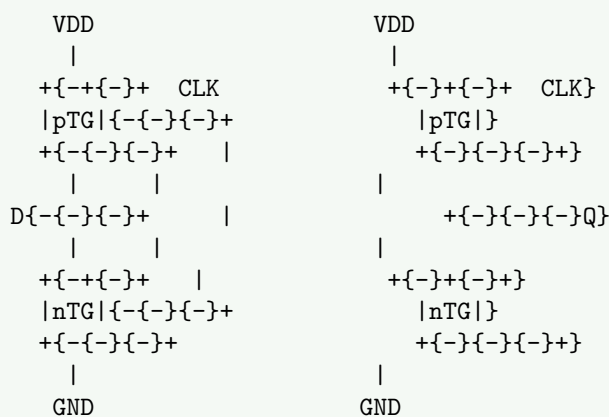
કલોક્સ NOR2 SR લેચ:



D-લેચ અમલીકરણ:



CMOS D-લેચ સર્કિટ:



ઓપરેશન:

- CLK = 1: માસ્ટર ટ્રાન્સપેરન્ટ, સ્લેવ હોલ્ડ
- CLK = 0: માસ્ટર હોલ્ડ, સ્લેવ ટ્રાન્સપેરન્ટ
- ડેટા ટ્રાન્સફર: કલોક એજ પર

SR લેચ માટે ટ્રુથ ટેબલ:

પ્રશ્ન 5(બ) [4 ગુણ]

વેરિલોગનો ઉપયોગ કરીને 8×1

જવાબ

વેરિલોગ કોડ:

```
module mux\_8x1(  
    input [7:0] data\_in,      // 8  
    input [2:0] select,      // 3{-  
    output reg data\_out      //  
);  
  
always @(*) begin  
    case (select)  
        3{b000}: data\_out = data\_in[0];  
        3{b001}: data\_out = data\_in[1];  
        3{b010}: data\_out = data\_in[2];  
        3{b011}: data\_out = data\_in[3];  
        3{b100}: data\_out = data\_in[4];  
        3{b101}: data\_out = data\_in[5];  
        3{b110}: data\_out = data\_in[6];  
        3{b111}: data\_out = data\_in[7];  
        default: data\_out = 1{b0};  
    endcase  
end  
  
endmodule
```

વૈકલ્પિક અમલીકરણ:

```
module mux\_8x1\_dataflow(  
    input [7:0] data\_in,  
    input [2:0] select,  
    output data\_out  
);  
  
assign data\_out = data\_in[select];  
  
endmodule
```

ટ્રુથ ટેબલ:

Select[2:0]	આઉટપુટ
000	data_in[0]
001	data_in[1]
010	data_in[2]
011	data_in[3]
100	data_in[4]
101	data_in[5]
110	data_in[6]
111	data_in[7]

ટેસ્ટબેન્ચ:

```
module tb\_mux\_8x1;
    reg [7:0] data\_in;
    reg [2:0] select;
    wire data\_out;

    mux\_8x1 dut(.data\_in(data\_in), .select(select), .data\_out(data\_out));

    initial begin
        data\_in = 8{b10110100};
        for (int i = 0; i { } 8; i++) begin
            select = i;
            \#10;
            $display("Select=\\%d, Output=\\%b", select, data\_out);
        end
    end
endmodule
```

મેમરી ટ્રીક

“MUX સિલેક્ટ લાઇન્સના આધારે ઘણા ઇનપુટ્સમાંથી એક પસંદ કરે છે”

પ્રશ્ન 5(ક) [7 ગુણ]

વેરિલોગમાં બિહેવિયરલ મોડેલિંગ સ્ટાઇલનો ઉપયોગ કરીને કુલ એડર અમલમાં મૂકો.

જવાબ

વેરિલોગ કોડ:

```
module full\_adder\_behavioral(
    input A,
    input B,
    input Cin,
    output reg Sum,
    output reg Cout
);

//          always
always @(*) begin
    case ({A, B, Cin})
        3{b000}: begin Sum = 1{b0}; Cout = 1{b0}; end
        3{b001}: begin Sum = 1{b1}; Cout = 1{b0}; end
        3{b010}: begin Sum = 1{b1}; Cout = 1{b0}; end
        3{b011}: begin Sum = 1{b0}; Cout = 1{b1}; end
        3{b100}: begin Sum = 1{b1}; Cout = 1{b0}; end
        3{b101}: begin Sum = 1{b0}; Cout = 1{b1}; end
        3{b110}: begin Sum = 1{b0}; Cout = 1{b1}; end
        3{b111}: begin Sum = 1{b1}; Cout = 1{b1}; end
    endcase
end
```

```

        default: begin Sum = 1{b0}; Cout = 1{b0}; end
    endcase
end

endmodule

```

વૈકલ્પિક બિહેવિયરલ સ્ટાઇલ:

```

module full\_adder\_behavioral\_alt(
    input A, B, Cin,
    output reg Sum, Cout
);

always @(*) begin
    \{Cout, Sum\} = A + B + Cin;
end

endmodule

```

ટ્રુથ ટેબલ:

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

ટેસ્ટબેન્ચ:

```

module tb\_full\_adder;
    reg A, B, Cin;
    wire Sum, Cout;

    full\_adder\_behavioral dut(.A(A), .B(B), .Cin(Cin),
                                .Sum(Sum), .Cout(Cout));

    initial begin
$monitor("A=%b
B=%b Cin=%b | Sum=%b Cout=%b",

        A, B, Cin, Sum, Cout);

        \{A, B, Cin\} = 3{b000}; \#10;
        \{A, B, Cin\} = 3{b001}; \#10;
        \{A, B, Cin\} = 3{b010}; \#10;
        \{A, B, Cin\} = 3{b011}; \#10;
        \{A, B, Cin\} = 3{b100}; \#10;
        \{A, B, Cin\} = 3{b101}; \#10;
        \{A, B, Cin\} = 3{b110}; \#10;
        \{A, B, Cin\} = 3{b111}; \#10;

        $finish;
    end
endmodule

```

બિહેવિયરલ લક્ષણો:

- Always બ્લોક: સ્ટુકચર નહીં, બિહેવિયર વર્ણવે છે
- Case સ્ટેટમેન્ટ: ટૂથ ટેબલ અમલીકરણ
- ઓટોમેટિક સિન્થેસિસ: ટુલ્સ ઓપ્ટિમાઇઝડ સર્કિટ જનરેટ કરે છે

મેમરી ટ્રીક

“બિહેવિયરલ મોડેલિંગ સર્કિટ કેવી રીતે નહીં, શું કરે છે તે વર્ણવે છે”

પ્રશ્ન 5(અ) OR [3 ગુણ]

NOR2 ગેટ CMOS સર્કિટને તેના સ્ટિક ડાયાગ્રામ સાથે અમલમાં મૂકો.

ଉଦାହରଣ

CMOS NOR2 સર્કિટ:

```

      VDD
      |
+{-{-}{-}{-}+{-}{-}{-}{-}+}
      |           |
+{-+{-}{-}+   +{-}{-}+{-}{-}+}
|pA |         |pB | PMOS ( )
+{-{-}{-}{-}+   +{-}{-}{-}{-}+}
      |           |
+{-{-}{-}{-}Y{-}{-}{-}{-}+}
      |
+{-+{-}{-}+}
|nA | NMOS ( )
+{-{-}{-}{-}+}
      |
+{-+{-}{-}+ }

```



```

end else if (enable) begin
    if (count == 4{b1111}) begin
        count {=} 4{b0000}; //
    end else begin
        count {=} count + 1;
    end
end
// enable
end

```

endmodule

ઓવરફ્લો સાથે વિસ્તૃત વર્ઝન:

```

module counter\_4bit\_enhanced(
    input clk,
    input reset,
    input enable,
    output reg [3:0] count,
    output overflow
);

always @(posedge clk or posedge reset) begin
    if (reset) begin
        count {=} 4{b0000};
    end else if (enable) begin
        count {=} count + 1; //
    end
end

assign overflow = (count == 4{b1111}) & enable;

endmodule

```

કાઉન્ટ સિક્વન્સ:

ક્લોક	Count[3:0]	દર્શાવેલ
1	0000	0
2	0001	1
3	0010	2
...
15	1110	14
16	1111	15
17	0000	0 (રોલઓવર)

ટેસ્ટબેન્ચ:

```
module tb\_counter\_4bit;
    reg clk, reset, enable;
    wire [3:0] count;

    counter\_4bit\_up dut(.clk(clk), .reset(reset),
        .enable(enable), .count(count));

    //
    initial begin
        clk = 0;
        forever \#5 clk = {}clk;
    end

    //
    initial begin
        reset = 1; enable = 0;
        \#10 reset = 0; enable = 1;
        \#200 enable = 0; //
        \#20 enable = 1; //
        \#100 $finish;
    end

    //
    always @(posedge clk) begin
        $display("Time=%t Count=%d", $time, count);
    end
endmodule
```

મેમરી ટ્રીક

“અપ કાઉન્ટર: ઇનેબલ હોય ત્યારે દરેક કલોક પર વધારો”

પ્રશ્ન 5(ક) OR [7 ગુણ]

વેરિલોગમાં બિહેવિયરલ મોડેલિંગ સ્ટાઇલનો ઉપયોગ કરીને 3:8 ડિકોડર અમલમાં મૂકો.

જવાબ

વેરિલોગ કોડ:

```
module decoder\_3x8\_behavioral(
    input [2:0] address, // 3{- }
    input enable, //
    output reg [7:0] decode\_out // 8{- }
);

always @(*) begin
    if (enable) begin
        case (address)
            3{b000}: decode\_out = 8{b00000001}; // Y0
            3{b001}: decode\_out = 8{b00000010}; // Y1
            3{b010}: decode\_out = 8{b000000100}; // Y2
            3{b011}: decode\_out = 8{b000001000}; // Y3
            3{b100}: decode\_out = 8{b000010000}; // Y4
            3{b101}: decode\_out = 8{b000100000}; // Y5
            3{b110}: decode\_out = 8{b001000000}; // Y6
            3{b111}: decode\_out = 8{b010000000}; // Y7
            default: decode\_out = 8{b000000000};
        endcase
    end
end
```

```

    end else begin
        decode\_out = 8{b00000000}; //
    end
end

endmodule

```

વૈકલ્પિક અમલીકરણ:

```

module decoder\_3x8\_shift(
    input [2:0] address,
    input enable,
    output [7:0] decode\_out
);

assign decode\_out = enable ? (8{b00000001} {} address) : 8{b00000000};

endmodule

```

ટ્રુથ ટેબલ:

Enable	Address[2:0]	decode_out[7:0]
0	XXX	00000000
1	000	00000001
1	001	00000010
1	010	00000100
1	011	00001000
1	100	00010000
1	101	00100000
1	110	01000000
1	111	10000000

ટેસ્ટબેન્ચ:

```
module tb\_decoder\_3x8;
    reg [2:0] address;
    reg enable;
    wire [7:0] decode\_out;

    decoder\_3x8\_behavioral dut(.address(address), .enable(enable),
                                .decode\_out(decode\_out));

    initial begin
        $monitor("Enable=\\%b Address=\\%b | Output=\\%b",
                enable, address, decode\_out);

        // Enable = 0
        enable = 0;
        for (int i = 0; i { } 8; i++) begin
            address = i;
            \\#10;
        end

        // Enable = 1
        enable = 1;
        for (int i = 0; i { } 8; i++) begin
            address = i;
            \\#10;
        end

        $finish;
    end
endmodule
```

એપ્લિકેશન્સ:

- મેમોરી એડ્રેસિંગ: 8 મેમોરી લોકેશનમાંથી એક પસંદ કરો
- ડિવાઇસ સિલેક્શન: 8 પેરિફેરલ ડિવાઇસમાંથી એક ઇનેબલ કરો
- ડીમલ્ટિપ્લેક્સિંગ: સિંગલ ઇનપુટને પસંદ કરેલા આઉટપુટ પર રાઉટ કરો

ડિઝાઇન લક્ષણો:

- વન-હોટ એન્કોડિંગ: એક સમયે ફક્ત એક આઉટપુટ હાઇ
- ઇનેબલ કંટ્રોલ: ગ્લોબલ ઇનેબલ/ડિસેબલ ફંક્શનાલિટી
- ફુલ ડીકોડિંગ: બધા શક્ય ઇનપુટ કોમ્બિનેશન હેન્ડલ

મેમરી ટ્રીક

“3:8 ડીકોડર - 3 ઇનપુટ્સ 8 આઉટપુટમાંથી 1 પસંદ કરે છે”