

Subject Name (Gujarati)

4353206 -- Summer 2025

Semester 1 Study Material

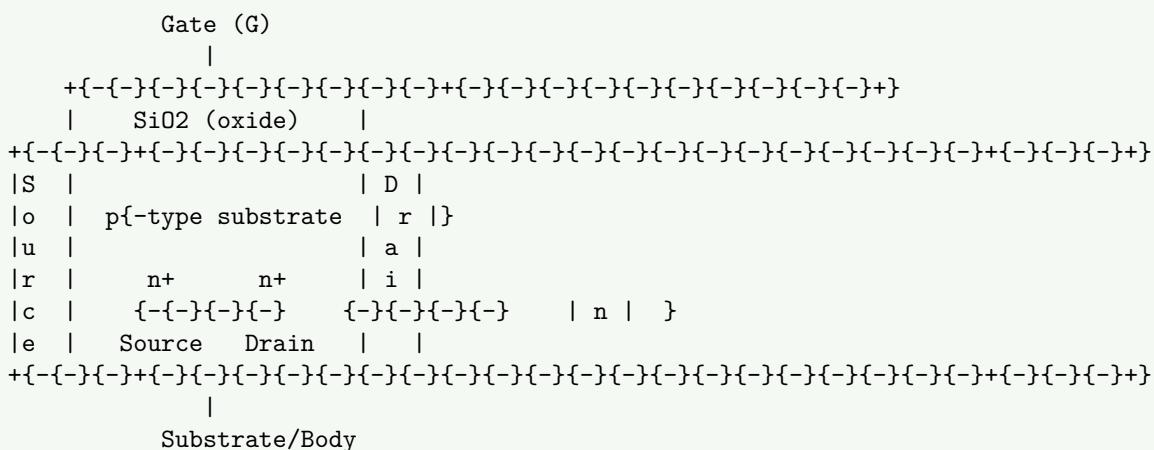
Detailed Solutions and Explanations

પ્રશ્ન 1(અ) [3 ગુણ]

n-ચેનલ MOSFET ના ભૌતિક બંધારણનું સાહેબ વેબલવાળું આકૃતિ દોરો.

જવાબ

આકૃતિ:



મુખ્ય ઘટકો:

- સોર્સ: n+ ડોઝ વિસ્તાર જે ઈલેક્ટોન પૂરો પાડે છે
- ડ્રેઇન: n+ ડોઝ વિસ્તાર જે ઈલેક્ટોન એકત્ર કરે છે
- ગેટ: ચેનલને નિયમિત કરતું મેટલ ઈલેક્ટ્રોડ
- ઓક્સાઇડ: SiO₂ ઇન્સ્યુલેટિંગ સ્ટર
- સબસ્ટ્રેટ: p-ટાઇપ સિલિકન બોડી

મેમરી ટ્રીક

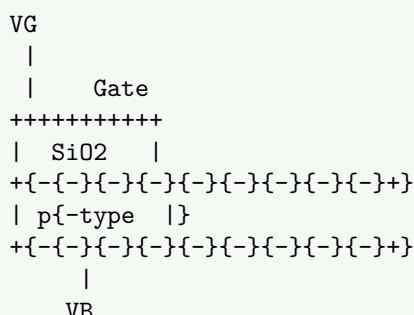
"SOGD - સોર્સ, ઓક્સાઇડ, ગેટ, ડ્રેઇન"

પ્રશ્ન 1(બ) [4 ગુણ]

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

જવાબ

MOS બાયાસિંગ સર્કિટ:



એન્જી બેન્ડ ડાયાગ્રામ:

બાયાસ સ્વિથ્ટિ	અનેર્જી બેન્ડ વર્તન
ડિપ્લીશન ઇન્વર્શન	બેન્ડ ઉપરની તરફ વળે છે, હોલ્સ ખતમ થાય છે મજબુત બેન્ડ બેન્ડિંગ, ઈલેક્ટોન ચેનલ બને છે

ઇન્વર્શન રીજન વિગતો:

- મજબૂત ઇન્વર્શન: VG > VT (થેશોડ વોલ્ટેજ)
 - ઈલેક્ટ્રોન ચેનલ: Si-SiO₂ ઇન્ટરફેસ પર બને છે
 - ચેનલ કંડાન્ડિટિવિટી: ગેટ વોલ્ટેજ સાથે વધે છે
 - થેશોડ શરત: સરકેસ પોટેન્શિયલ = 20F

ਮੇਮਰੀ ਟ੍ਰੀਕ

“DIVE - ડિપ્લિશન, ઇન્વર્શન, વોલ્ટેજ, ઇલેક્ટોન્સ”

પ્રશ્ન 1(ક) [7 ગુણ]

MOSFET नी I-V लाक्षणिकता समजावे.

ଜ୍ଵାବୁ

I-V लाक्षणिकता विस्तारी:

વિસ્તાર	શરત	ફેન કરંટ
કટાઓક	$VGS < VT$	$ID \approx 0$
લિનિયર	$VGS > VT, VDS < VGS - VT$	$ID = \frac{1}{2}nCox(W/L)[(VGS-VT)VDS - VDS^2/2]$
સેચ્યુરેશન	$VGS > VT, VDS \geq VGS - VT$	$ID = (\frac{1}{2}nCox/2)(W/L)(VGS-VT)^2$

लाक्षणिकता वक्तः

```

ID
|
|
|     Saturation
| +-{--}{-}{-}{-}{-}{-}{-}{-}{-}
|
| /
| / Linear
| /
|/
{--}{-}{-}+{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-} VDS
0      VGS{-VT}

```

મુખ્ય પેરામીટર્સ:

- **En:** ઇલેક્ટ્રોન મોબિલિટી
 - **Cox:** ગેટ ઓક્સાઇડ કેપેસિટન્સ
 - **W/L:** પહોળાઈ અને લંબાઈનો રેશિયો
 - **VT:** થ્રેશોડ વોલ્ટેજ

ઓપરેટિંગ મોડસ:

- એન્હાન્સમેન્ટ: પોઝિટિવ VGS સાથે ચેનલ બને છે
 - સ્કવર લો: સેર્વયુરેશન વિસ્તાર ચતુર્ભુજ સંબંધ અનુસરે છે

ਮੇਮਰੀ ਡੀਕ

“CLS - કટાઓક, લિનિયર, સેર્વયરેશન”

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 પરિમાળોમાં વ્યવસ્થિત ઘટાડો. સ્કેલિંગની જરૂરિયાત:

ફાયદો	વર્ણન
વધુ ડેન્સિટી	ચિપ વિસ્તાર દીઠ વધુ ટ્રાન્ઝિસ્ટર
જડપી સ્પીડ	ઘટેલી ગેટ ડીલે
ઓછી પાવર	ઘટેલી સ્વચ્છિંગ એનજી
કોસ્ટ રિડક્ષન	વેફર દીઠ વધુ ચિપ્સ

સ્કેલિંગ પ્રકારો:

પ્રકાર	ગેટ લંબાઈ	સપ્લાય વોલ્ટેજ	ઓક્સાઇડ જાડાઈ
કોન્સ્ટન્ટ વોલ્ટેજ	દોષ	સ્થિર	દોષ
કોન્સ્ટન્ટ ફીડ્ડ	દોષ	દોષ	દોષ

નકારાત્મક અસરો:

- શૉર્ટ ચેનલ અસરો: થ્રેશોલ્ડ વોલ્ટેજ રોલ-ઓફ
- હોટ કેરિયર અસરો: ડિવાઇસ ડીગ્રેડેશન
- ગેટ લીકેજ: વધેલો ટનલિંગ કરંટ
- પ્રોસેસ વેરિએશન્સ: મેન્યુક્કચરિંગ પડકારો
- પાવર ડિન્સિટી: હોટ ડિસિપેશન સમસ્યાઓ

મેમરી ટ્રીક

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

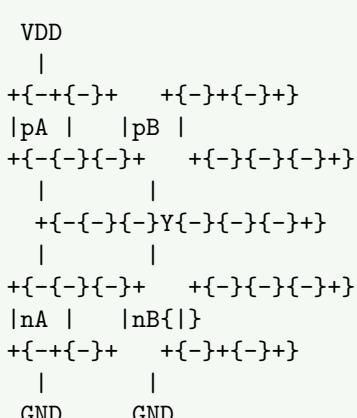
પ્રશ્ન 2(અ) [3 ગુણ]

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

જવાબ

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

CMOS અમલીકરણાં:



ટુથ 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 કોમ્પિલેન્ટરી સ્વિચિંગ જોઈએ”

પ્રશ્ન 2(બ) [4 ગુણ]

અન્હાન્સમેન્ટ લોડ ઇન્વર્ટરને તેના સર્કિટ ડાયાગ્રામ સાથે સમજાવો.

જવાબ

સર્કિટ ડાયાગ્રામ:

```

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

```

કન્ફિગ્યુરેશન:

ઘટક	પ્રકાર	કનેક્શન
લોડ (ME)	અન્હાન્સમેન્ટ NMOS	ગેટ VDD સાથે જોડાયેલું
ડ્રાઇવર (MD)	અન્હાન્સમેન્ટ NMOS	ગેટ ઇન્પુટ 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 નાળુક
- નોઇજ માર્જિન: થ્રેશોલ્ડ લોસ કારણે ઘટેલો

મેમરી ટ્રીક

“ELI - અન્હાન્સમેન્ટ લોડ ઇન્વર્ટરમાં થ્રેશોલ્ડ સમસ્યા”

પ્રશ્ન 2(ક) [7 ગુણ]

ઇન્વર્ટરની વોલ્ટેજ ટ્રાન્સફર કેરેક્ટરિસ્ટિક સમજાવો.

જવાબ

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 એફ્સ:

```

Vout
|
VDD+      +{--}{-}{-}{-}{-}{-}
|           /
|           /
VM+{--}{-}{-}{-}{-}{-}+ VM
|           {}
|           {}
0          +{--}{-}{-}{-}{-}{-}
+{--}{-}{-}{-}{-}{-}+{--}{-}{-}{-}{-}+{--}{-}{-}{-}{-} Vin
VIL VM VIH VDD

```

નોઇજ માર્જિન્સ:

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

વિસ્તારો:

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

ગુણવત્તા મેટ્રિક્સ:

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

મેમરી ટ્રીક

"VTC દર્શાવે છે VOICE - VOH, VOL, ઇનપુટ થ્રેશોલ્ડ, લાક્ષણિકતા, બધું"

પ્રશ્ન 2(અ) OR [3 ગુણ]

CMOS નો ઉપયોગ કરીને NAND2 ગેટ સમજાવો.

જવાબ

CMOS NAND2 સક્રિએટ:

```

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

```

```

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

```

ટ્રૂથ 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]

રાન્ડિસ્ટ્રિવ લોડ ઇન્વાર્ટર 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
Vin = 0	OFF	VOH = VDD
Vin = VDD	ON	VOL = R·ID/(R+RDS)

VTC લાક્ષણિકતાઓ:

- VOH: ઉત્તમ (VDD)
- VOL: R અને RDS રેશિયો પર આધાર રાખે છે
- પાવર વપરાશ: ઇનપુટ હાઇ લોય ત્યારે સ્ટેટિક કરેંગ
- ટ્રાન્ઝિશન: રેજિસ્ટ્રિટ લોડ કારણે ધીમું

ડિગ્યુન્ડ્રોડ-ઓફ્સ:

- મોટો R: બહુતર VOL, ધીમું સ્વિચિંગ
- નાનો R: જડપી સ્વિચિંગ, વધુ પાવર
- એરિયા: રેજિસ્ટર નોંધપાત્ર જગ્યા લે છે

મેમરી ટ્રીક

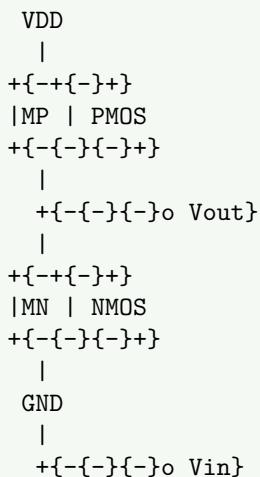
"RLI - રેજિસ્ટ્રિટ લોડમાં અનિવાર્ય પાવર વપરાશ"

પ્રશ્ન 2(ક) OR [7 ગુણ]

CMOS ઇન્વર્ટર દોરો અને VTC સાથે તેની કામગીરી સમજાવો.

જવાબ

CMOS ઇન્વર્ટર સક્રિંટ:



ઓપરેશન વિસ્તારો:

Vin રેન્જ	PMOS	NMOS	Vout	વિસ્તાર
0 થી VTN	ON	OFF	VDD	1
**VTN થી VDD-	VT _P	**	ON	ON
VDD-	VT _P	થી VDD	OFF	ON

VTC વિશ્લેષણ:

```

Vout
|
VDD+
| {}
| {}
| {} {\_ \_ \_}
VM+ {\_ \_ \_} {\_ \_ \_}
| {} {\_ \_ \_}
| {} 0 +{-{-}{-}}
+{-{-}{-}{-}+{-}{-}{-}+{-}{-}{-}+{-}{-}{-}+{-}{-}{-} Vin}
VTN VM VTP VDD

```

મુખ્ય લક્ષણો:

- ઝીરો સ્ટેટિક પાવર: કોઈ DC કરંટ પાથ નથી
- કુલ સ્વિંગ: $VOH = VDD$, $VOL = 0V$
- હાઇ નોઇડ માર્જિન્સ: $NMH = NML \approx 0.4VDD$
- તીક્ષ્ણ ટ્રાન્ઝિશન: ટ્રાન્ઝિશન વિસ્તારમાં હાઇ ગેઇન

ડિઝાઇન વિચારણાઓ:

- એરેશિયો: સિમેટ્રિક સ્વિચિંગ માટે EN/EP
- થ્રેશોફ્ટ મેર્ચિંગ: $VTN \approx |VTP|$

મેમરી ટ્રીક

"CMOS માં ઝીરો સ્ટેટિક પાવર અને કુલ સ્વિંગ"

પ્રશ્ન 3(અ) [3 ગુણ]

ડિલીશન લોડનો ઉપયોગ કરીને $Y = (A\bar{D} + B\bar{D})C\bar{D} + D\bar{D} + E\bar{D}$ અમલમાં મૂકો.

જવાબ

લોજિક સરળીકરણ: $Y = (A\bar{D} + B\bar{D})C\bar{D} + D\bar{D} + E\bar{D} = A\bar{D}C\bar{D} + B\bar{D}C\bar{D} + D\bar{D} + E\bar{D}$
ડિલીશન લોડ અમલીકરણ:

```

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

```

પુલ-ડાઉન નેટવર્ક:

- સીરીઝ: $A\bar{D}C\bar{D}$ પાથ અને $B\bar{D}C\bar{D}$ પાથ
- પેરેલલ: બધા પાથ પેરેલલમાં જોડાયેલા
- અમલીકરણ: યોગ્ય ટ્રાન્ઝિસ્ટર સાઇલિંગ જરૂરી

મેમરી ટ્રીક

"દિપ્લીશન 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
ઇન્ટરક્નેક્ટ	રાઉટિંગ રિસોર્સ
સ્વચ્છ મેટ્રિક્સ	કનેક્શન પોઇન્ટ્સ

પ્રોગ્રામિંગ ટેકનોલોજીઓ:

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

એપ્લિકેશન્સ:

- પ્રોટોટાઇપિંગ: ડિજિટલ સિસ્ટમ ડેવલપમેન્ટ
- DSP: સિગ્નલ પ્રોસેસિંગ એપ્લિકેશન્સ
- કંટ્રોલ સિસ્ટમ્સ: ઇન્ડસ્ટ્રીયલ ઓટોમેશન
- કોમ્પ્યુનિકેશન્સ: પ્રોટોકોલ અમલીકરણ

ASIC સામે ફાયદા:

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

મેમરી ટ્રીક

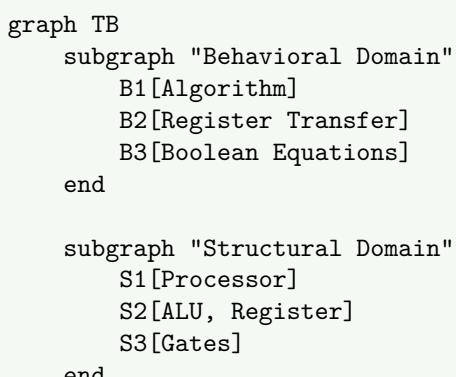
"FPGA - લવચીક પ્રોગ્રામિંગ ફાયદા આપે છે"

પ્રશ્ન 3(ક) [7 ગુણ]

Y ચાર્ટ ડિઝાઇન ફલો દોરો અને સમજાવો.

જવાબ

Y-ચાર્ટ ડાયાગ્રામ:



```

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 ડોમેન્સ:

ડોમેન	લેવલ	વર્ણન
બિહેવિયરલ	એલગોરિધમ $\rightarrow RT \rightarrow$	સિસ્ટમ શું કરે છે
સ્ટ્રક્ચરલ	પ્રોસેસર $\rightarrow ALU \rightarrow$	સિસ્ટમ કેવી રીતે બનાવેલ છે
ફિજિકલ	ફ્લોર પ્લાન $\rightarrow \rightarrow$	ભौતિક અમલીકરણ

ડિઝાઇન ફલો પ્રક્રિયા:

- ટોપ-ડાઉન: બિહેવિયરલથી શરૂ કરી ફિજિકલ તરફ જાઓ
- બોટમ-અપ: ઘટકોથી ઉપરની તરફ બનાવો
- મિક્સડ એપ્રોચ: બનેની સંયુક્ત પદ્ધતિ

એબસ્ટ્રેક્શન લેવલ્સ:

- સિસ્ટમ લેવલ: સૌથી વધુ એબસ્ટ્રેક્શન
- RT લેવલ: રજિસ્ટર ટ્રાન્સફર ઓપરેશન્સ
- ગેટ લેવલ: બુલિયન લોજિક અમલીકરણ
- લેન્યાઉટ લેવલ: ભૌતિક જ્યોમેટ્રી

ડિઝાઇન વેરિફિકેશન:

- હોરિડોન્ટલ: સમાન લેવલ ડોમેન્સ વચ્ચે
- વર્ટિકલ: સમાન ડોમેનમાં લેવલ્સ વચ્ચે

મેમરી ટ્રીક

"Y-ચાર્ટ: બિહેવિયરલ, સ્ટ્રક્ચરલ, ફિજિકલ - BSP ડોમેન્સ"

પ્રશ્ન 3(અ) OR [3 ગુણ]

ડિસ્લિશન લોડનો ઉપયોગ કરીને NOR2 ગેટ સમજાવો.

જવાબ

ડિસ્લિશન લોડ NOR2 સર્કિટ:

```

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

```

ટૂથ 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 કરંટ પૂરો પાડે છે

મેમરી ટ્રીક

"ડિલ્ફીશન 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)
પફોર્માન્સ	શ્રોષ	સારં
એરિયા	લઘુત્તમ	મધ્યમ
પાવર	ઓપ્ટિમાઇઝડ	સ્વીકાર્ય
કોર્સ્ટ	હાઇ NRE	લોઓર NRE
લવચીકરણ	મહત્તમ	મધ્યદિત
જોખમ	વધુ	ઓછું

કુલ કસ્ટમ લાક્ષણિકતાઓ:

- દરેક ટ્રાન્ઝિસ્ટર: મેન્યુઅલી ડિઝાઇન અને પ્લેસ કરેલું
- લેઆઉટ ઓપ્ટિમાઇઝેશન: મહત્તમ ડેન્સિટી હાંસલ
- એલિક્શાસ: હાઇ-વોલ્યુમ, પફોર્માન્સ-કિટિકલ

સેમી કસ્ટમ પ્રકારો:

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

ડિઝાઇન ફ્લો તુલના:

- કુલ કસ્ટમ: સ્પેસિફિકેશન $\rightarrow \rightarrow \rightarrow$
- સેમી કસ્ટમ: સ્પેસિફિકેશન $\rightarrow HDL \rightarrow \rightarrow &$

મેમરી ટ્રીક

"કુલ કસ્ટમ - મહત્તમ નિયંત્રણ, સેમી કસ્ટમ - સ્પીડ સમજૂતી"

પ્રશ્ન 3(ક) OR [7 ગુણ]

ASIC ડિઝાઇન ફ્લો વિગતવાર દોરો અને સમજાવો.

જવાબ

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	તૂલ્સ/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 એલગોરિધમ
રાઉટિંગ	પ્લેસ કરેલા એલિમેન્ટ્સ જોડો	રાઉટિંગ એલગોરિધમ

વેરિફિકેશન સ્ટેપ્સ:

- ફુંક્શનલ: RTL સિમ્યુલેશન અને વેરિફિકેશન
- ગેટ-લેવલ: પોસ્ટ-સિન્થેસિસ સિમ્યુલેશન
- ફિલીક્લ: DRC, LVS, એન્ટેના ચેક્સ
- ટાઇમિંગ: સેટઅપ/હાઉલ વાયોલેશન માટે STA

ડિઝાઇન કન્સ્ટ્રીટ્યુન્ટ્સ:

- ટાઇમિંગ: કલોક ફીકવનસી જરૂરિયાતો
- એરિયા: સિલિકોન એરિયા મર્યાદાઓ
- પાવર: પાવર વપરાશ લક્ષ્યો
- ટેસ્ટ: ટેસ્ટબિલિટી માટે ડિઝાઇન

સાઇન-ઓફ ચેક્સ:

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

મેમરી ટ્રીક

“ASIC ફ્લો: RTL → → → ”

પ્રશ્ન 4(અ) [3 ગુણ]

CMOS સાથે લોજિક ફુંક્શન $G = (A(D+E)+BC) \oplus$ અમલમાં મૂકો

જવાબ

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

CMOS અમલીકરણ:

$$\begin{array}{c}
 \text{VDD} \\
 | \\
 +\{-\{-\}\{-\}\{-\}\{-\}+\{-\}\{-\}\{-\}\{-\}\{-\}+\{-\}\{-\}\{-\}\{-\}\{-\}+ \\
 | \quad | \quad | \quad | \\
 +\{-+\{-\}+ \quad +\{-\}+\{-\}+ \quad +\{-\}+\{-\}+ \quad +\{-\}+\{-\}+
 \end{array}$$

```

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

```

નેટવર્ક કન્ફિગરેશન:

- PMOS: કોમ્પ્લેમેન્ટનું સીરીજ અમલીકરણ
- NMOS: મૂળ ફુકશનનું પેરેલલ અમલીકરણ

મેમરી ટ્રીક

"કોમ્પ્લેક્સ CMOS - PMOS સીરીજ, 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[0];
//                     \^{}data\_in[1];
//                     \^{}data\_in[2]);
//                     \^{}data\_in[0];
//                     \^{}data\_in[1];
//                     \^{}data\_in[2])};

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

ટુથ ટેબલ:

ઇનપુટ [2:0]	1 નું સંખ્યા	ઇવન પેરિટી	ઓડ પેરિટી
000	0	0	1
001	1	1	0
010	1	1	0
011	2	0	1
100	1	1	0
101	2	0	1
110	2	0	1
111	3	1	0

મુખ્ય લક્ષણો:

- XOR રિફશન: $\sim \text{data_in}$ ઇવન પેરિટી આપે છે
- કોમ્પલેક્સ: $\sim (\sim \text{data_in})$ ઓડ પેરિટી આપે છે

મેમરી ટ્રીક

"પેરિટી ચેક: બધા બિટ્સનું XOR"

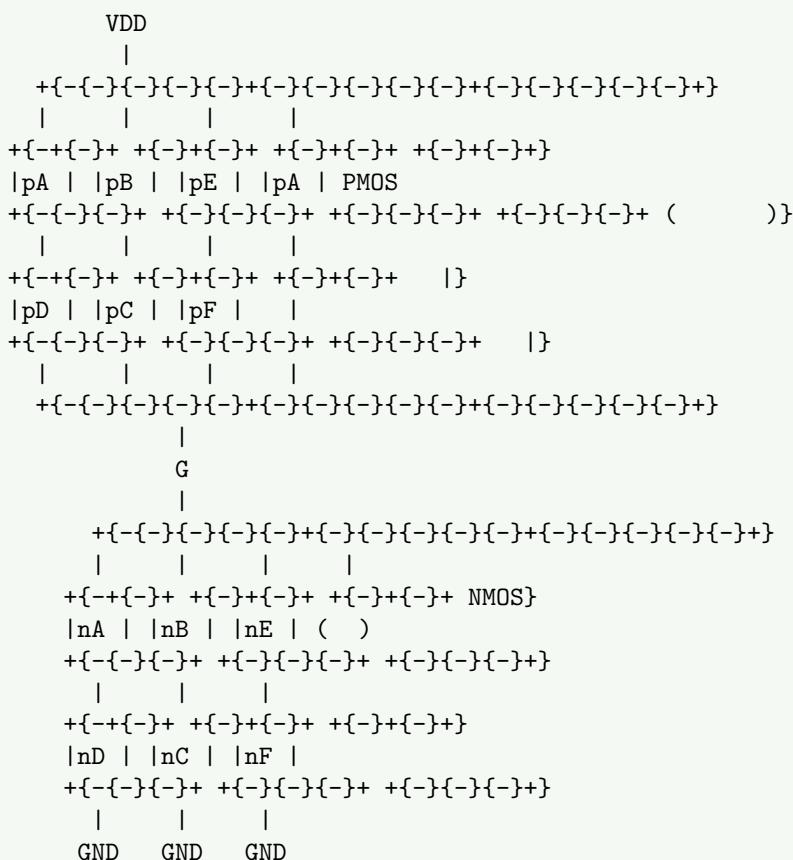
પ્રશ્ન 4(ક) [7 ગુણ]

અમલીકરણ: 1) $G = (AD + BC + EF)$ CMOS નો ઉપયોગ કરીને [3 ગુણ] 2) $Y' = (ABCD + EF(G+H)+ J)$ CMOS નો ઉપયોગ કરીને [4 ગુણ]

જવાબ

ભાગ 1: $G = (AD + BC + EF)$ [3 ગુણ]

CMOS સક્રિયા:



ભાગ 2: $Y' = (ABCD + EF(G+H)+ J)$ [4 ગુણ]

આ એક કોમ્પલેક્સ ફંક્શન છે જેને બહુવિધ સ્ટેજની જરૂર છે:

સ્ટેજ 1: $(G+H)$ અમલીકરણ સ્ટેજ 2: $EF(G+H)$ અમલીકરણ સ્ટેજ 3: બધા ટર્મ્સ કોમ્પાઇન આવા કોમ્પલેક્સ ફંક્શન માટે ટ્રાન્સમિશન ગેટ્સ અને બહુવિધ સ્ટેજનો ઉપયોગ વધુ પ્રેક્ટિકલ છે.

મેમરી ટ્રીક

“કોમ્પ્લેક્સ ફંક્શન્સને સ્ટેજડ અમલીકરણ જોઈએ”

પ્રશ્ન 4(અ) OR [3 ગુણ]

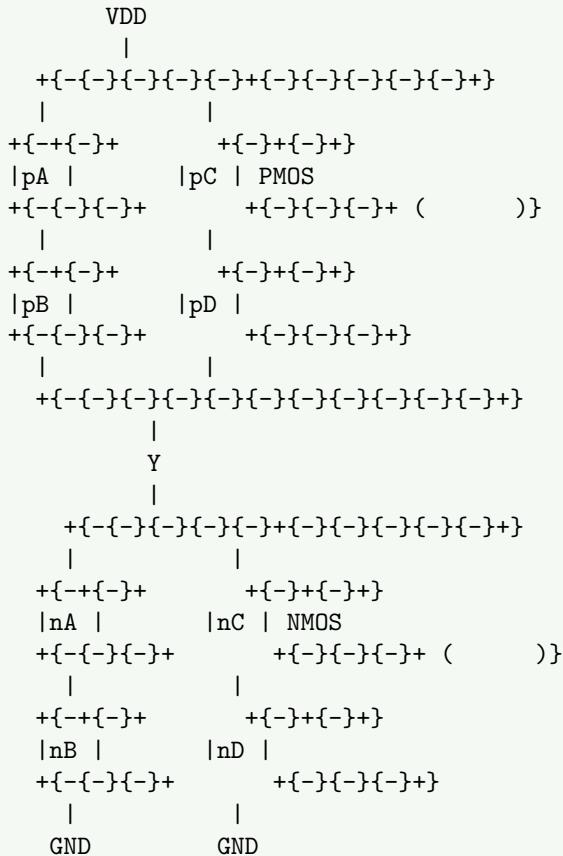
ઉદાહરણ સાથે AOI લોજિક સમજાવો.

જવાબ

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

ઉદાહરણ: $Y = (AB + CD) \bar{}$

AOI અમલીકરણ:



ફાયદા:

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

એપ્લિકેશન્સ:

- કોમ્પ્લેક્સ ગેટ્સ: મલ્ટિ-ઇનપુટ ફંક્શન્સ
- સ્પીડ-કિટિકલ 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:

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

મેમરી ટ્રીક

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

પ્રશ્ન 4(ક) OR [7 ગુણ]

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

જવાબ

કલોક NOR2 SR લેચ:

```

S { -{ - }{ - }+      CLK}
   |       |
 +{ -+{ - }+  +{ - }+{ - }+}
 |TG1| |TG2|
 +{ -{ - }{ - }+  +{ - }{ - }{ - }+}
   |       |
 +{ -{ - }{ - }{ - }+  +{ - }{ - }{ - }{ - }+}
   |       |
 +{ -+{ - }+      +{ - }+{ - }+}
 |NOR| { -{ - }{ - }{ - }Q{ - }{ - }{ - }{ - }{ - }|NOR| { - } }
 +{ -{ - }{ - }+  +{ - }{ - }{ - }+ NOR
   |       |
 +{ -{ - }{ - }{ - }{ - }{ - }R{ - }{ - }{ - }{ - }{ - }{ - }{ - }+}

```

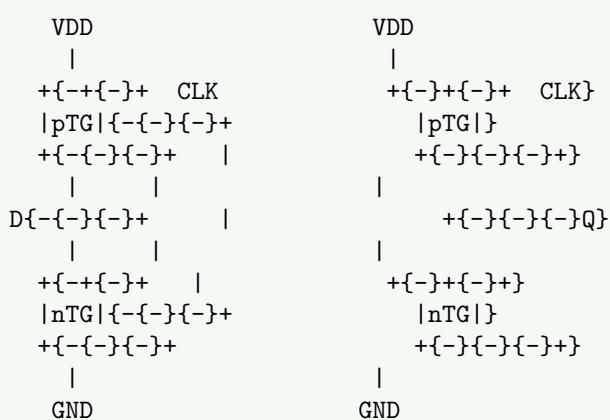
D-લેચ અમલીકરણ:

```

D { -{ - }{ - }{ - }+}
   |
 +{ -+{ - }+  CLK}
 |TG1| { -{ - }{ - }+}
 +{ -{ - }{ - }+  |}
   |
 +{ -{ - }+  |}
   |
 +{ -+{ - }+{ - }+{ - }+}
   |
 +{ -{ - }{ - }+{ - }{ - }{ - }+}
   |
 Q

```

CMOS D-લેચ સર્કિટ:



ઓપરેશન:

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

SR લેચ માર્ગ ટ્રથ ટેબલ:

S	R	CLK	Q	Q'
0	0	1	0	0
0	1	1	0	1
1	0	1	1	0
1	1	1	1	1

ਮੇਮਰੀ ਟ੍ਰੀਕ

“કલોકડ લેય ટાઇમિંગ કંટ્રોલ માટે ટ્રાન્સમિશન ગેટ ઉપયોગ કરે છે”

પ્રશ્ન 5(અ) [૩ ગુણ]

યુલર પાથ એપ્રોચને દ્યાનમાં લેતા CMOS નો ઉપયોગ કરીને $Y = (PQ + U)^T$ માટે સ્ટિક ડાયાગ્રામ દોરો.

ଜ୍ଵାବୁ

લોજિક વિશ્લેષણ: $Y = (PQ + U)'$ માટે જરૂરી છે PMOS: $(PQ)' \cdot U' = (P' + Q') \cdot U'$ NMOS: $PQ + U$ સ્ટિક ડાયાગ્રામ:

यूलर पाठः

1. **PMOS:** $P' \rightarrow Q'(), U'$
 1. **NMOS:** $P \rightarrow Q(), U$
 1. એપોર્ટિવ ગાર્ડિંગ: દોઅંગેઠા પ્રિન્ટિયાંડાં દેખી છે

1. આસ્ટ્રમલ્લ રાઉન્ડબોર્ડ

- ડિસ્કુન બ્લેક્સ: બહેતર પર્ફોર્મન્સ માટે મિનિમાઇઝ
 - કોન્ટ્રક્ટ પ્લેસમેન્ટ: યોગ્ય VDD/GND કનેક્શન્સ
 - મેટલ રાઉટિંગ: DRC વાયોલેશન ટાળો

ਮੇਮਰੀ ਟ੍ਰਿਕ

“સ્ટિક ડાયાગ્રામ યબર પાથ ઓપ્ટિમાઇઝેશન સાથે કિંજિકલ લેઆઉટ દર્શાવે છે”

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
```

ટૈકલ્યિક અમલીકરણ:

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

assign data\_out = data\_in[select];

endmodule
```

ટૂથ 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 લાઇનના આધારે ઘણા ઇનપુટ્સમાંથી એક પસંદ કરે છે"

પ્રશ્ન 5(ક) [7 ગુણ]

વેરિલોગમાં બિહેવિયર 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

```

વैકल्पिक બિહેવિયરલ સ્ટાઇલ:

```

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

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

endmodule

```

ટુથ 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
```

બિહેવિયર લક્ષણો:

- Always બ્લોક: સ્ટ્રક્ચર નહીં, બિહેવિયર વર્ણવે છે
- Case સ્ટેટમેન્ટ: ટુથ ટેબલ અમલીકરણ
- ઓટોમેટિક સિન્થેસિસ: ટૂલ્સ ઓપ્ટિમાઇઝ સર્કિટ જનરેટ કરે છે

મેમરી ટ્રીક

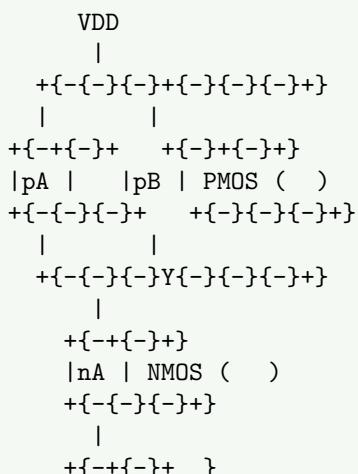
“બિહેવિયર મોડેલિંગ સર્કિટ કેવી રીતે નહીં, શું કરે છે તે વર્ણવે છે”

પ્રશ્ન 5(અ) OR [3 ગુણ]

NOR2 ગે �CMOS સર્કિટને તેના સ્થિક ડાયાગ્રામ સાથે અમલમાં મૂકો.

જવાબ

CMOS NOR2 સર્કિટ:



```

| nB |
+{-{-} {-}+}
|
GND

स्ट्रिक डायाग्राम:

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

:
{-
  : P{-}diffusion
{-
  : N{-}diffusion  }
{-
  : Polysilicon  }
{-
  }

लेआउट नियमों:
• PMOS: पुल-अप माटे पेरेलब कनेक्शन
• NMOS: पुल-डाउन माटे सीरीज कनेक्शन
• को-ट्रैक्ट्रस: योग्य VDD/GND कनेक्शन्स
• स्पेसिंग: लघुतम डिजाइन नियमों पूरा करवा

```

मेमरी ट्रीक

"NOR गेट: पेरेलब PMOS, सीरीज NMOS"

प्रश्न 5(ब) OR [4 गुण]

वरिलोगनो उपयोग करीने 4 बिट अप काउन्टर अभियां मूको

जवाब

वरिलोग कोड:

```

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

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

```

```

    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 વર્ણન:

```

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 સિક્વેન્ચ્સ:

કલોક	Count[3:0]	દર્શાંશ
1	0000	0
2	0001	1
3	0010	2
...
15	1110	14
16	1111	15
17	0000	0 (રોલઅડર)

ટેસ્ટબેન્ક:

```
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]

દરિલોગમાં બિલેવિયરલ મોડેલિંગ સ્ટાઇલનો ઉપયોગ કરીને 3:8 ડિકોડર અમલમાં મૂકો.

જવાબ

દરિલોગ 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{b00000100}; // Y2
            3{b011}: decode\_out = 8{b00001000}; // Y3
            3{b100}: decode\_out = 8{b00010000}; // Y4
            3{b101}: decode\_out = 8{b00100000}; // Y5
            3{b110}: decode\_out = 8{b01000000}; // Y6
            3{b111}: decode\_out = 8{b10000000}; // Y7
            default: decode\_out = 8{b00000000};
        endcase
    end
endmodule
```

```

    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

```

ટુથ 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

ટેસ્ટબેન્ક:

```
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
- ડીમલ્ટિપ્લિકેશન: સિંગલ ઇનપુટને પસંદ કરેલા આઉટપુટ પર રાઉટ કરો

ડિઝાઇન લક્ષણો:

- વન-હોટ એન્કોડિંગ: એક સમયે ફક્ત એક આઉટપુટ હાઈ
- ઇનેબલ કંટ્રોલ: ગલોબલ ઇનેબલ/ડિસેબલ ફંક્શનાલિટી
- ફૂલ ટીકોડિંગ: બધા શક્ય ઇનપુટ કોમ્બિનેશન હેન્ડલ

મેમરી ટ્રીક

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