

# Subject Name (Gujarati)

4353206 -- Summer 2025

Semester 1 Study Material

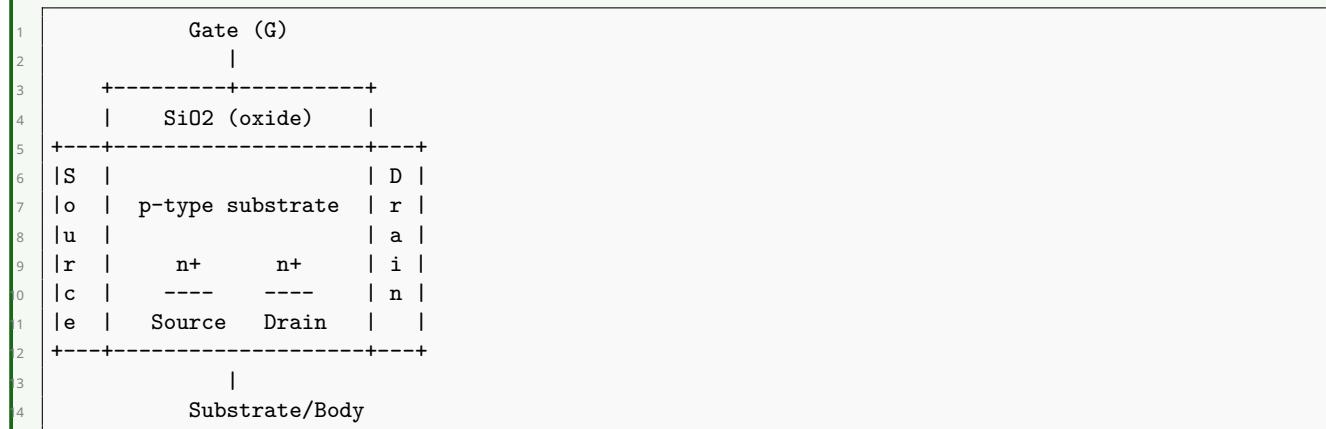
*Detailed Solutions and Explanations*

## પ્રશ્ન 1(અ) [3 ગુણ]

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

### જવાબ

#### આકૃતિ:



#### મુખ્ય ઘટકો:

- સોર્સ: n+ ડોઝ વિસ્તાર જે ઈલેક્ટ્રોન પૂરો પાડે છે
- ડ્રેઇન: n+ ડોઝ વિસ્તાર જે ઈલેક્ટ્રોન અંકત્ર કરે છે
- ગેટ: એનલને નિયંત્રિત કરતું મેટલ ઈલેક્ટ્રોડ
- ઓક્સાઇડ: SiO<sub>2</sub> ઇન્સ્યુલેટિંગ સ્તર
- સબસ્ટ્રેટ: p-ટાઇપ સિલિકન બોડી

### મેમરી ટ્રીક

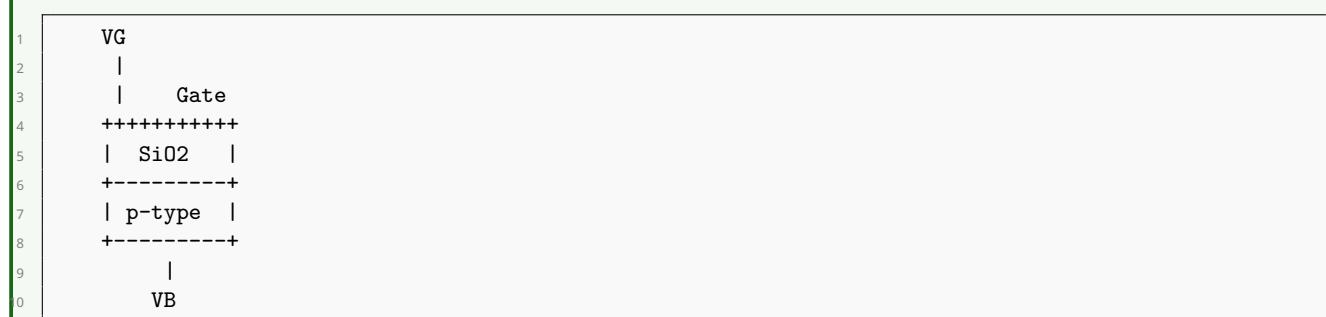
"SOGD - સોર્સ, ઓક્સાઇડ, ગેટ, ડ્રેઇન"

## પ્રશ્ન 1(બ) [4 ગુણ]

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

### જવાબ

#### MOS બાયાસિંગ સર્કિટ:



એનજી બેન્ડ ડાયાગ્રામ:

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

### ઇન્વર્શન રીજન વિગતો:

- મજબૂત ઇન્વર્શન:  $VG > VT$  (થ્રેશોલ્ડ વોલ્ટેજ)
- ઈલેક્ટ્રોન ચેનલ:  $Si-SiO_2$  ઇન્ટરફેસ પર બને છે
- ચેનલ કન્ડક્ટિવિટી: ગેટ વોલ્ટેજ સાથે વધે છે
- થ્રેશોલ્ડ શરત: સરફેસ પોટેન્શિયલ =  $2DF$

### મેમરી ટ્રીક

"DIVE - ડિપ્લીશન, ઇન્વર્શન, વોલ્ટેજ, ઈલેક્ટ્રોન્સ"

### પ્રશ્ન 1(ક) [7 ગુણ]

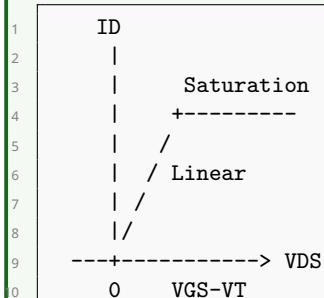
MOSFET ની I-V લાક્ષણિકતા સમજાવો.

#### જવાબ

##### I-V લાક્ષણિકતા વિસ્તારો:

વિસ્તાર	શરત	ફોર્મ્યુલા
કટઓફ	$VGS < VT$	$ID \approx 0$
લિનિયર	$VGS > VT, VDS < VGS - VT$	$ID = \mu n Cox(W/L)[(VGS - VT)VDS - VDS^2/2]$
સેચ્યુરેશન	$VGS > VT, VDS \geq VGS - VT$	$ID = (\mu n Cox/2)(W/L)(VGS - VT)^2$

##### લાક્ષણિકતા વક્ત:



##### મુખ્ય પેરામીટર્સ:

- $\mu n$ : ઈલેક્ટ્રોન મોબિલિટી
- $Cox$ : ગેટ ઓક્સાઇડ કેપેસિટન્સ
- $W/L$ : પહોળાઈ અને લંબાઈનો રેશિયો
- $VT$ : થ્રેશોલ્ડ વોલ્ટેજ

##### ઓપરેટિંગ મોડ્સ:

- એન્ડાન્સમેન્ટ: પોઝિટિવ VGS સાથે ચેનલ બને છે
- સ્કવેર 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
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	૦૦	સ્થિર	૦૦
કોન્સ્ટન્ટ ફીલ્ડ	૦૦	૦૦	૦૦

નકારાત્મક અસરો:

- શોર્ટ ચેનલ અસરો: થ્રેશોલ્ડ વોલ્ટેજ રોલ-ઓફ
- હોટ કેરિયર અસરો: ડિવાઇસ ડીગ્રેડેશન
- ગેટ લીકેજ: વધેલો ટનલિંગ કરેટ
- પ્રોસેસ વેરિએશન્સ: મેન્યુક્યુરિટીંગ પડકારો
- પાવર ડેન્સિટી: હીટ ડિસાપેશન સમસ્યાઓ

## મેમરી ટ્રીક

"SHGPP - શોર્ટ ચેનલ, હોટ કેરિયર, ગેટ લીકેજ, પ્રોસેસ, પાવર"

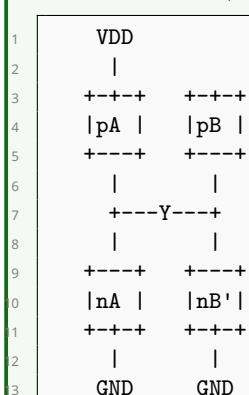
## પ્રશ્ન 2(અ) [૩ ગુણ]

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

## જવાબ

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

CMOS અમલીકરણ:



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 કોમિલમેન્ટરી સ્વિચિંગ જોઈએ"

### પ્રશ્ન 2(બ) [4 ગુણ]

અન્હાન્સમેન્ટ લોડ ઇન્વર્ટરને તેના સર્કિટ ડાયગ્રામ સાથે સમજાવો.

#### જવાબ

##### સર્કિટ ડાયગ્રામ:

```

1   VDD
2   |
3   +---o VG2
4   |
5   +---+ Enhancement
6   |ME | Load
7   +---+
8   |
9   +---o Vout
10  |
11  +---+
12  |MD | Driver
13  +---+
14  |
15  GND
16  |
17  +---o Vin
    
```

##### કન્ફિગરેશન:

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

##### ઓપરેશન:

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

##### દ્રાન્સફર લાક્ષણિકતાઓ:

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

## મેમરી ટ્રીક

"ELI - અન્હાન્સમેન્ટ લોડ ઇન્વર્ટરમાં થ્રેશોલ્ડ સમર્થ્યા"

### પ્રશ્ન 2(ક) [7 ગુણ]

ઇન્વર્ટરની વોલ્ટેજ દ્રાન્સફર કેરેક્ટરિસ્ટિક સમજાવો.

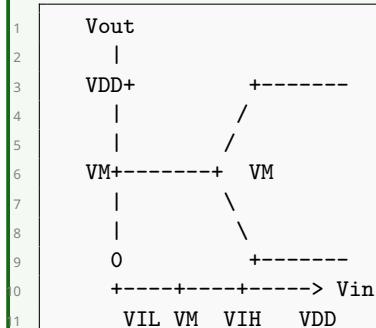
#### જવાબ

##### VTC પેરામીટર્સ:

પેરામીટર	વર્ગન	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	સિવિલિંગ થ્રેશોલ્ડ	VDD/2

### VTC ફક્સ:



#### 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 ઇમ્યુનિટી
- સિમેટ્રિક સિવિલિંગ: VM = VDD/2
- કુલ સિવિંગ: VOH = VDD, VOL = 0

### મેમરી ટ્રીક

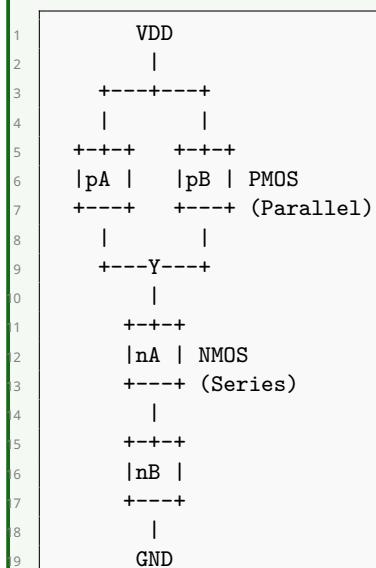
“VTC દર્શાવે છે VOICE - VOH, VOL, ઇનપુટ થ્રેશોલ્ડ, લાક્ષણિકતા, બધું”

### પ્રશ્ન 2(અ) OR [3 ગુણ]

CMOS નો ઉપયોગ કરીને NAND2 ગેટ સમજાવો.

#### જવાબ

#### CMOS NAND2 સક્રિટ:



ટૂથ 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: સીરીજ કનેક્શન (પુલ-ડાઉન)
- આઉટપુટ લો: ફક્ત જ્યારે બંને ઇનપુટ હાઇ

મેમરી ટ્રીક

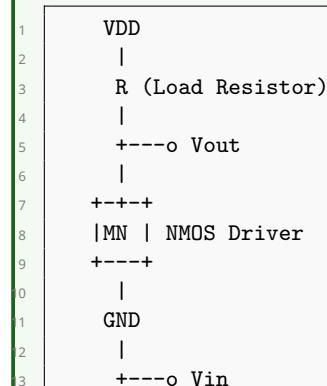
“NAND - નોટ AND, પેરેલલ PMOS, સીરીજ NMOS”

## પ્રશ્ન 2(બ) OR [4 ગુણ]

રેઝિસ્ટિવ લોડ ઇન્વર્ટર સક્રિટના ઓપરેટિંગ મોડ અને VTC સમજાવો.

જવાબ

સક્રિટ કન્ફિગરેશન:



ઓપરેટિંગ મોડ્સ:

ઇનપુટ સ્થિતિ	NMOS સ્થિતિ	આઉટપુટ
$V_{in} = 0$	OFF	$V_{OH} = V_{DD}$
$V_{in} = V_{DD}$	ON	$V_{OL} = R \cdot I_D / (R + R_{DS})$

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 કરેં
- ટ્રાન્ઝિશન: રેઝિસ્ટિવ લોડ કારણે ધીમું

ડિઝાઇન ટ્રૈક-ઓફસ:

- મોટો  $R$ : બહેતર  $V_{OL}$ , ધીમું સ્વચ્છિંગ
- નાનો  $R$ : જડપી સ્વચ્છિંગ, વધુ પાવર
- એરિયા: રેઝિસ્ટર નોંધપાત્ર જગ્યા લે છે

મેમરી ટ્રીક

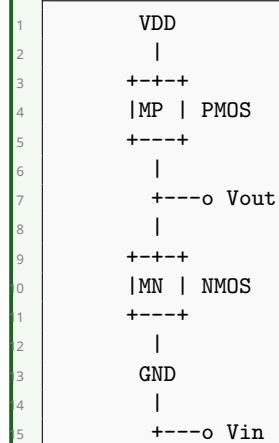
“RLI - રેઝિસ્ટિવ લોડમાં અનિવાર્ય પાવર વપરાશ”

## પ્રશ્ન 2(ક) OR [7 ગુણ]

CMOS ઇનવર્ટર દોરો અને VTC સાથે તેની કામગીરી સમજાવો.

### જવાબ

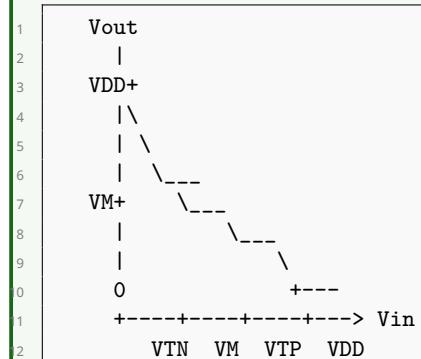
#### CMOS ઇનવર્ટર સર્કિટ:



#### ઓપરેશન વિસ્તારો:

Vin રેઝ	PMOS	NMOS	Vout	વિસ્તાર
0 થી VTN	ON	OFF	VDD	1
**VTN થી VDD-	VTP	**	ON	ON
**VDD-	VTP	થી VDD**	OFF	ON

#### VTC વિશ્લેષણ:



#### મુખ્ય લક્ષણો:

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

#### ડિઝાઇન વિચારણાઓ:

- એ રેશિયો: સિમેટ્રિક સ્વિંગ માટે  $\frac{1}{2}N - \frac{1}{2}P$
- થ્રેશોલ્ડ મેથિંગ:  $VTN \approx |VTP|$

### મેમરી ટ્રીક

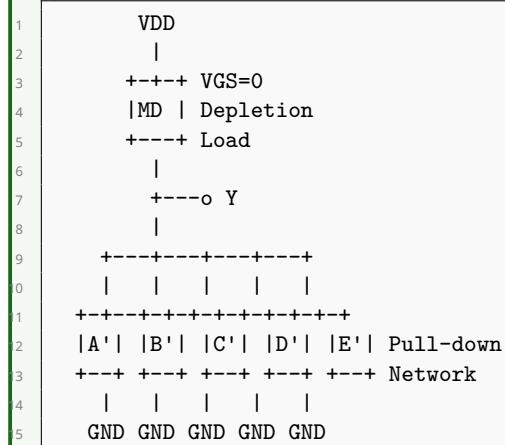
“CMOS માં જીરો સ્ટેટિક પાવર અને કુલ સ્વિંગ”

## પ્રશ્ન 3(અ) [3 ગુણ]

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

## જવાબ

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



### પુલ-ડાઉન નેટવર્ક:

- સીરીઝ:  $A \oplus C$  પાથ અને  $B \oplus C$  પાથ
- પેરેલલ: બધા પાથ પેરેલલમાં જોડાયેલા
- અમલીકરણ: યોગ્ય ટ્રાન્ઝિસ્ટર સાઇઙ્ચિંગ 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
ઇન્ટરક્નેક્ટ	રાઉટિંગ રિસોર્સ
સ્વિચ મેટ્રિક્સ	કનેક્શન પોઇન્ટ્સ

### પ્રોગ્રામિંગ ટેકનોલોજીઓ:

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

### એપ્લિકેશન્સ:

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

### ASIC સામે ફાયદા:

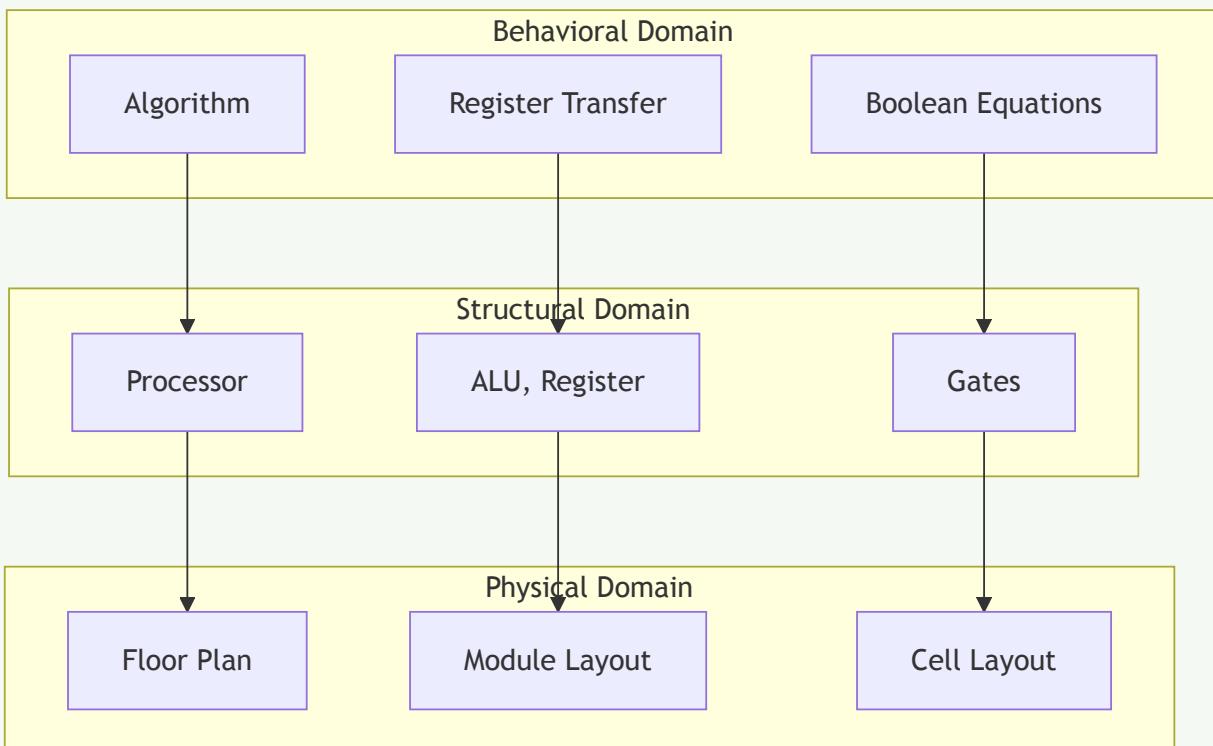
- લવચીકતા: રીકન્ફિગરેબલ ડિજાઇન
- ટાઇમ-ડુ-માર્કેટ: ઝડપી વિકાસ
- કોરસ્ટ: નાના વોલ્યુમ માટે ઓછો
- જોખમ: ઘટેલો ડિજાઇન જોખમ

## પ્રશ્ન 3(ક) [7 ગુણ]

Y ચાર્ટ ડિઝાઇન ફ્લો દોરો અને સમજાવો.

## જવાબ

## Y-ચાર્ટ ડાયાગ્રામ:



## ડિઝાઇન ડોમેઇન્સ:

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 $\rightarrow$ $\rightarrow$	ભौતિક અમલીકરણ

## ડિઝાઇન ફ્લો પ્રક્રિયા:

- ટોપ-ડાઉન: બિહેવિયરલથી શરૂ કરી ફિઝિકલ તરફ જાઓ
- બોટમ-અપ: ઘટકોથી ઉપરની તરફ બનાવો
- મિક્રોડ એપ્રોથ્યુ: બનેની સંયુક્ત પદ્ધતિ

## ઓફસ્ટ્રેક્શન લેવલ્સ:

- સિસ્ટમ લેવલ: સૌથી વધુ ઓફસ્ટ્રેક્શન
- RT લેવલ: રજિસ્ટર ટ્રાન્સફર ઓપરેશન્સ
- ગેટ લેવલ: યુલિયન લોજિક અમલીકરણ
- લેઆઉટ લેવલ: ભૌતિક જ્યોમેટ્રી

## ડિઝાઇન વેરિફિકેશન:

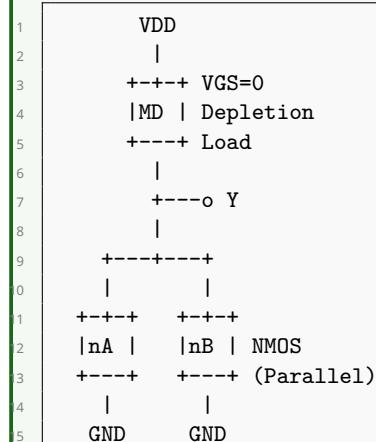
- હોરિઝોન્ટલ: સમાન લેવલે ડોમેઇન્સ વચ્ચે
- વર્ટિકલ: સમાન ડોમેઇનમાં લેવલ્સ વચ્ચે

## પ્રશ્ન 3(અ) OR [3 ગુણ]

ડિલીશન લોડનો ઉપયોગ કરીને NOR2 ગેટ સમજાવો.

### જવાબ

#### ડિલીશન લોડ NOR2 સર્કિટ:



#### ટુથ 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

"ડિલીશન 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)
પફોર્મન્સ	શ્રેષ્ઠ	સારું
એરિયા	લઘુત્તમ	મધ્યમ
પાવર	ઓફિન્માઇઝ	સ્વીકાર્ય
કોસ્ટ	હાઇ NRE	લોઅર NRE
લવચીક્રિટા	મહત્તમ	મધ્યાર્ડિટ
જોખમ	વધુ	ઓછું

### કુલ કસ્ટમ લાક્ષણિકતાઓ:

- દરેક ટ્રાન્ઝિસ્ટર: મેન્યુઅલી ડિજાઇન અને પ્લેસ કરેલું
- લેઆઉટ ઓરિયાઇઝનાન: મહત્વમાન ડેન્સિટી હાસલ
- એપ્લિકેશન્સ: હાઇ-વોલ્યુમ, પર્ફોર્મન્સ-કિટિકલ

### સેમી કસ્ટમ પ્રકારો:

- ગેટ એરે: પૂર્વ-વ્યાખ્યાપિત ટ્રાન્ઝિસ્ટર એરે
- સ્ટેન્ડડ સેલ: પૂર્વ-ડિજાઇન કરેલા સેલ્વસની લાઇબ્રેરી
- FPGA: ફીલ્ડ પ્રોગ્રામેબલ લોજિક

### ડિજાઇન ફલો તુલના:

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

### મેમરી ટ્રીક

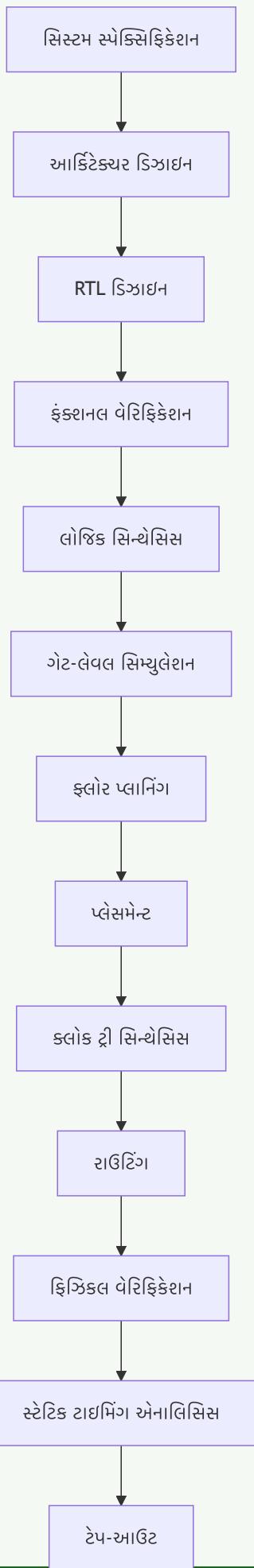
“કુલ કસ્ટમ - મહત્વમાન નિયંત્રણ, સેમી કસ્ટમ - સ્પીડ સમજૂતી”

### પ્રશ્ન 3(ક) OR [7 ગુણ]

ASIC ડિજાઇન ફલો વિગતવાર દીરો અને સમજાવો.

#### જવાબ

ASIC ડિજાઇન ફલો:



## 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

## વેરિફિકેશન સ્ટેપ્સ:

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

## ડિઝાઇન કન્સ્ટ્રીટ્યુન્ટ્સ:

- ટાઇમિંગ: ક્લોક ફીકવન્સી 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 ફ્લો: RTL → → → ”

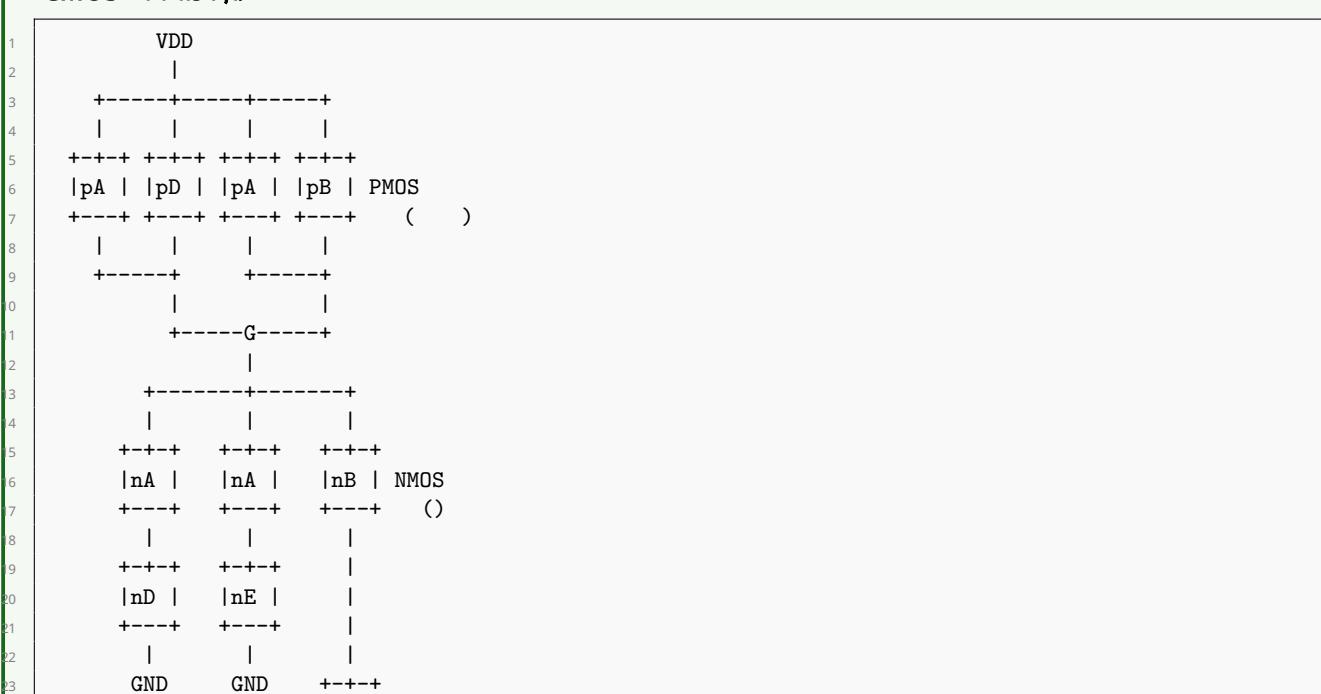
## પ્રશ્ન 4(અ) [3 ગુણ]

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

## જવાબ

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

CMOS અમલીકરણ:



```

24      | nC |
25      +---+
26      |
27      GND

```

#### નેટવર્ક કન્ફિગરેશન:

- PMOS: કોમિલમેન્ટનું સીરીજ અમલીકરણ
- NMOS: મૂળ ફંક્શનનું પેરેલલ અમલીકરણ

#### મેમરી ટ્રીક

“કોમલેક્સ CMOS - PMOS સીરીજ, NMOS પેરેલલ”

### પ્રશ્ન 4(બ) [4 ગુણ]

3 બિટ પેરિટી ચેકર માટે વેરિલોગ કોડ લખો.

#### જવાબ

##### વેરિલોગ કોડ:

```

1 module parity_checker_3bit(
2     input [2:0] data_in,
3     output parity_even,
4     output parity_odd
5 );
6
7 // assign parity_even = ^data_in;
8
9 // assign parity_odd = ~(^data_in);
10
11 /*
12 assign parity_even = data_in[0] ^ data_in[1] ^ data_in[2];
13 assign parity_odd = ~(data_in[0] ^ data_in[1] ^ data_in[2]);
14 */
15
16 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 રિઝફશન:  $\sim \text{data\_in}$  ઇવન પેરિટી આપે છે
- કોમિલમેન્ટ:  $\sim(\sim \text{data\_in})$  ઓડ પેરિટી આપે છે

#### મેમરી ટ્રીક

“પેરિટી ચેક: બધા બિટ્સનું XOR”

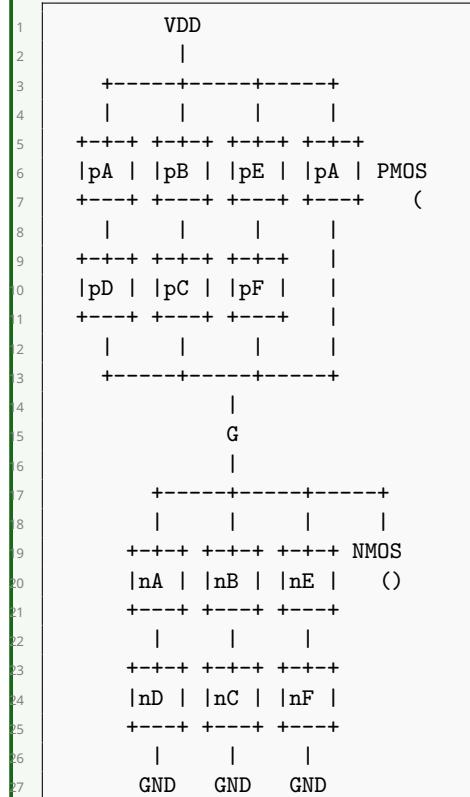
## પ્રશ્ન 4(ક) [7 ગુણ]

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 સર્કિટ:



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

આ એક કોમ્પ્લેક્સ ફંક્શન છે જેને બહુવિધ સ્ટેજની જરૂર છે:

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

### મેમરી ટ્રીક

"કોમ્પ્લેક્સ ફંક્શનને સ્ટેજડ અમલીકરણ જોઈએ"

## પ્રશ્ન 4(અ) OR [3 ગુણ]

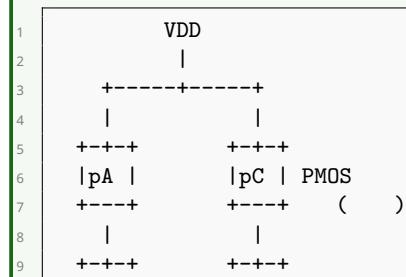
ઉદાહરણ સાથે AOI લોજિક સમજાવો.

### જવાબ

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

ઉદાહરણ:  $Y = (AB + CD) \oplus$

AOI અમલીકરણ:



```

0 | pB |      | pD |
1 +---+      +---+
2 |           |
3 +-----+
4 |
5   Y
6 |
7 +-----+
8 |           |
9 +---+      +---+
10 | nA |      | nC | NMOS
11 +---+      +---+ (   )
12 |           |
13 +---+      +---+
14 | nB |      | nD |
15 +---+      +---+
16 |           |
17 GND        GND

```

#### ફાયદા:

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

#### એપ્લિકેશન્સ:

- કોમ્પ્લેક્સ ગેટ્સ: મલ્ટિ-ઇનપુટ ફંક્શન્સ
- સ્પીડ-કિટિકલ પાથ: ઘટેલી ડિલે

## 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.

### જવાબ

#### વેરિલોગ કોડ:

```

1 module sipo_4bit(
2     input clk,
3     input reset,
4     input serial_in,
5     output reg [3:0] parallel_out
6 );
7
8     always @ (posedge clk or posedge reset) begin
9         if (reset) begin
10             parallel_out <= 4'b0000;
11         end else begin
12             // LSB
13             parallel_out <= {parallel_out[2:0], serial_in};
14         end
15     end
16
17 endmodule

```

#### ટેસ્ટબેન્ચ ઉદાહરણ:

```

1 module tb_sipo_4bit;
2     reg clk, reset, serial_in;
3     wire [3:0] parallel_out;
4
5     sipo_4bit dut(.clk(clk), .reset(reset),
6                     .serial_in(serial_in),

```

```

7           .parallel_out(parallel_out));
8
9     initial begin
10    clk = 0;
11    forever #5 clk = ~clk;
12  end
13
14  initial begin
15    reset = 1; serial_in = 0;
16    #10 reset = 0;
17    #10 serial_in = 1; // LSB
18    #10 serial_in = 0;
19    #10 serial_in = 1;
20    #10 serial_in = 1; // MSB
21    #20 $finish;
22  end
23 endmodule

```

ઓપરેશન ટાઈમલાઇન:

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

### મેમરી ટ્રીક

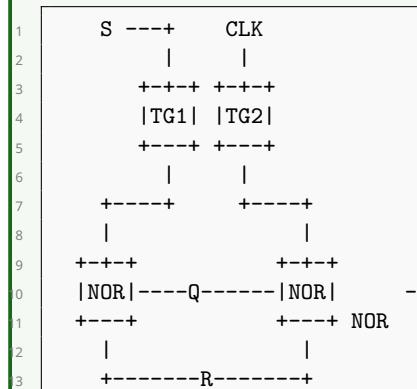
“SISO - સીરિયલ ઇન, પેરેલલ આઉટ શિફ્ટ લેફ્ટ સાથે”

## પ્રશ્ન 4(ક) OR [7 ગુણ]

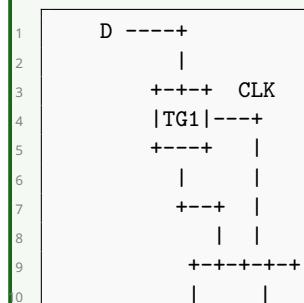
CMOS નો ઉપયોગ કરીને કલોક NOR2 SR લેચ અને D-લેચ અમલીકરણ કરો.

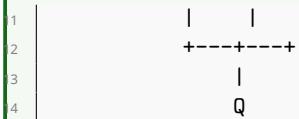
### જવાબ

#### કલોક NOR2 SR લેચ:

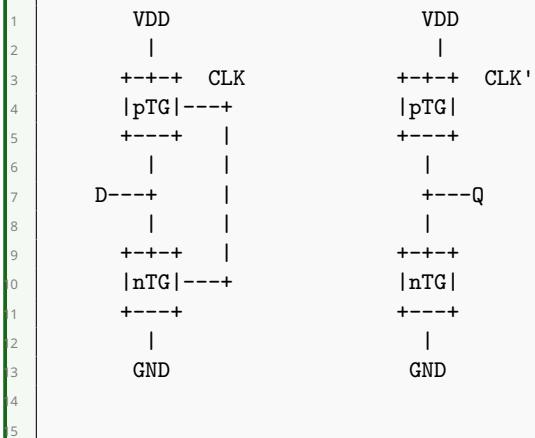


#### D-લેચ અમલીકરણ:





### CMOS D-લેચ સર્કિટ:



### ઓપરેશન:

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

### SR લેચ માર્ટ્ટર ટુથ ટેબલ:

S	R	CLK	Q	$Q'$
0	0	1	હોલ્ડ	હોલ્ડ
0	1	1	0	1
1	0	1	1	0
1	1	1	અવૈધ	અવૈધ

### મેમરી ટ્રીક

“કલોક લેચ ટાઇમિંગ કંટ્રોલ માટે ટ્રાન્સમિશન ગેટ ઉપયોગ કરે છે”

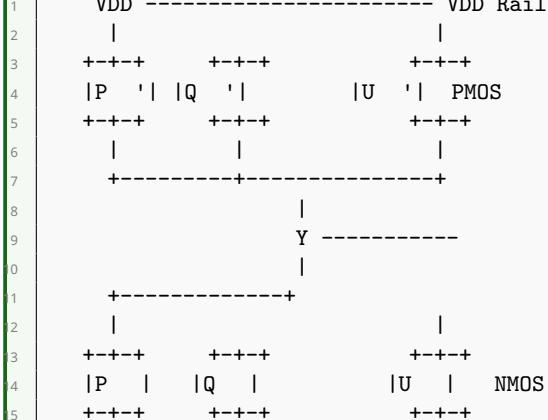
### પ્રશ્ન 5(અ) [3 ગુણ]

યુલર પાથ એપ્રોથને દ્યાનમાં લેતા CMOS નો ઉપયોગ કરીને  $Y = (PQ + U)'$  માટે સર્કિટ ડાયાગ્રામ દારો.

### જવાબ

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

### સર્કિટ ડાયાગ્રામ:



```

6   |       |
7   +-----+
8   |       |
9 GND -----+-----+--- GND Rail
10
11 :
12 - : P-diffusion (PMOS)
13 - : N-diffusion (NMOS)
14 - : Polysilicon  ()
15 - :

```

#### યુલર પાથ:

1. PMOS:  $P' \rightarrow Q'(), U'$

1. NMOS:  $P \rightarrow Q(), U$

1. ઓપ્ટિમલ રાઉટિંગ: કોસાઓવર મિનિમાઇઝ કરે છે

#### લેચાઉટ વિચારણાઓ:

- દિફ્યુઝન બેક્સ: બહેતર પર્ફોર્મન્સ માટે મિનિમાઇઝ
- કોન્ટેક્ટ પ્લેસમેન્ટ: યોગ્ય VDD/GND કનેક્શન્સ
- મેટલ રાઉટિંગ: DRC વાયોલેશન ટાળો

#### મેમરી ટ્રીક

"સ્ટિક ડાયાગ્રામ યુલર પાથ ઓપ્ટિમાઇઝેશન સાથે ફિઝિકલ લેચાઉટ દર્શાવે છે"

### પ્રશ્ન 5(બ) [4 ગુણ]

વેરિલોગનો ઉપયોગ કરીને  $8 \times 1$

#### જવાબ

##### વેરિલોગ કોડ:

```

1 module mux_8x1(
2     input [7:0] data_in,      // 8
3     input [2:0] select,      // 3-
4     output reg data_out      //
5 );
6
7 always @(*) begin
8     case (select)
9         3'b000: data_out = data_in[0];
10        3'b001: data_out = data_in[1];
11        3'b010: data_out = data_in[2];
12        3'b011: data_out = data_in[3];
13        3'b100: data_out = data_in[4];
14        3'b101: data_out = data_in[5];
15        3'b110: data_out = data_in[6];
16        3'b111: data_out = data_in[7];
17         default: data_out = 1'b0;
18     endcase
19 end
20
21 endmodule

```

##### વૈકલ્પિક અમલીકરણ:

```

1 module mux_8x1_dataflow(
2     input [7:0] data_in,
3     input [2:0] select,
4     output data_out
5 );
6
7 assign data_out = data_in[select];
8
9 endmodule

```

## ટ્રથ 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]

## ટેસ્ટબેન્ક:

```
1 module tb_mux_8x1;
2   reg [7:0] data_in;
3   reg [2:0] select;
4   wire data_out;
5
6   mux_8x1 dut(.data_in(data_in), .select(select), .data_out(data_out));
7
8   initial begin
9     data_in = 8'b10110100;
10    for (int i = 0; i < 8; i++) begin
11      select = i;
12      #10;
13      $display("Select=%d, Output=%b", select, data_out);
14    end
15  end
16 endmodule
```

## મેમરી ટ્રીક

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

## પ્રશ્ન 5(ક) [7 ગુણ]

વરિલોગમાં બિહેવિયરલ મોડેલિંગ સ્ટાઇલનો ઉપયોગ કરીને ફુલ એડર અમલમાં મૂકો.

## જવાબ

### વરિલોગ કોડ:

```
1 module full_adder_behavioral(
2   input A,
3   input B,
4   input Cin,
5   output reg Sum,
6   output reg Cout
7 );
8
9 //      always
10 always @(*) begin
11   case ({A, B, Cin})
12     3'b000: begin Sum = 1'b0; Cout = 1'b0; end
13     3'b001: begin Sum = 1'b1; Cout = 1'b0; end
14     3'b010: begin Sum = 1'b1; Cout = 1'b0; end
15     3'b011: begin Sum = 1'b0; Cout = 1'b1; end
16     3'b100: begin Sum = 1'b1; Cout = 1'b0; end
17     3'b101: begin Sum = 1'b0; Cout = 1'b1; end
18     3'b110: begin Sum = 1'b0; Cout = 1'b1; end
```

```

9      3'b111: begin Sum = 1'b1; Cout = 1'b1; end
10     default: begin Sum = 1'b0; Cout = 1'b0; end
11   endcase
12 end
13
14 endmodule

```

### ટેકલ્પિક બિહેવિયરલ સ્ટાઇલ:

```

1 module full_adder_behavioral_alt(
2   input A, B, Cin,
3   output reg Sum, Cout
4 );
5
6   always @(*) begin
7     {Cout, Sum} = A + B + Cin;
8   end
9
10 endmodule

```

### ટૂથ 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:

```

1 module tb_full_adder;
2   reg A, B, Cin;
3   wire Sum, Cout;
4
5   full_adder_behavioral dut(.A(A), .B(B), .Cin(Cin),
6                             .Sum(Sum), .Cout(Cout));
7
8   initial begin
9     $monitor("A=%b
0
1
B=%b Cin=%b | Sum=%b Cout=%b",
2
3           A, B, Cin, Sum, Cout);
4
5     {A, B, Cin} = 3'b000; #10;
6     {A, B, Cin} = 3'b001; #10;
7     {A, B, Cin} = 3'b010; #10;
8     {A, B, Cin} = 3'b011; #10;
9     {A, B, Cin} = 3'b100; #10;
10    {A, B, Cin} = 3'b101; #10;
11    {A, B, Cin} = 3'b110; #10;
12    {A, B, Cin} = 3'b111; #10;
13
14    $finish;
15  end
16 endmodule

```

### બિહેવિયરલ લક્ષણો:

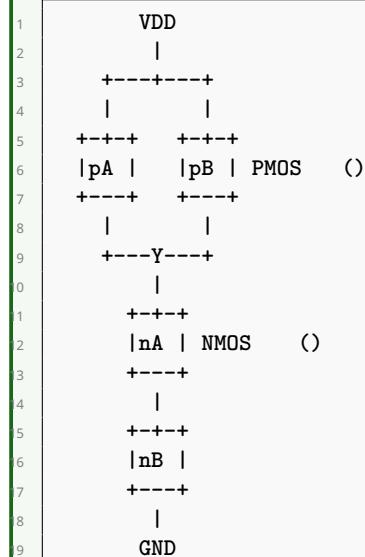
- Always બ્લોક: સ્ટ્રક્ચર નહીં, બિહેવિયર વર્ણિત છે
- Case સ્ટેટમેન્ટ: ટૂથ ટેબલ અમલીકરણા
- ઓટોમેટિક સિન્થેસિસ: ટૂલ્સ ઓપ્ટિમાઇઝ સર્કિટ જનરેટ કરે છે

## પ્રશ્ન 5(અ) OR [3 ગુણ]

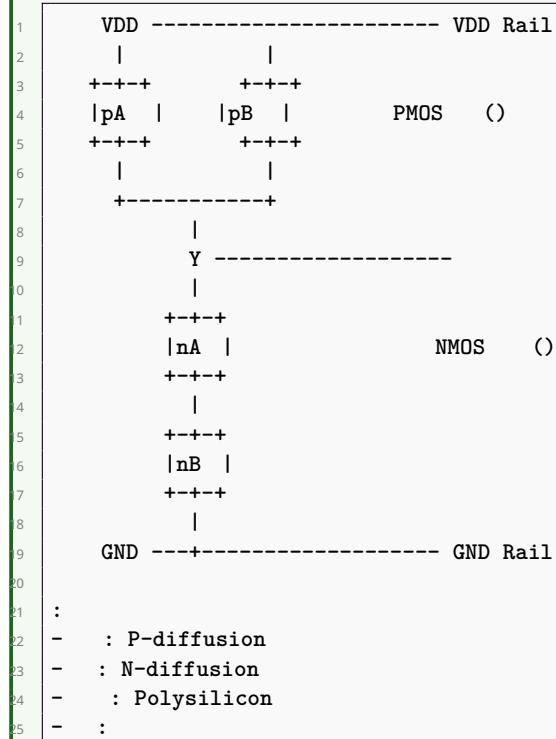
NOR2 ગેટ CMOS સર્કિટને તેના સ્ટિક ડાયાગ્રામ સાથે અમલમાં મૂકો.

### જવાબ

#### CMOS NOR2 સર્કિટ:



#### સ્ટિક ડાયાગ્રામ:



#### લેઆઉટ નિયમો:

- PMOS: પુલ-અપ માટે પેરેલલ કનેક્શન
- NMOS: પુલ-ડાઉન માટે સીરીઝ કનેક્શન
- કોન્ટેક્ટ્સ: યોગ્ય VDD/GND કનેક્શન્સ
- સ્પેસિંગ: લઘુત્તમ ડિઝાઇન નિયમો પૂરા કરવા

## પ્રશ્ન 5(બ) OR [4 ગુણ]

વેરિલોગનો ઉપયોગ કરીને 4 બિટ અપ કાઉન્ટર અમલમાં મૂકો

## જવાબ

વેરિલોગ કોડ:

```

1 module counter_4bit_up(
2     input clk,
3     input reset,
4     input enable,
5     output reg [3:0] count
6 );
7
8     always @(posedge clk or posedge reset) begin
9         if (reset) begin
10             count <= 4'b0000;
11         end else if (enable) begin
12             if (count == 4'b1111) begin
13                 count <= 4'b0000; ///
14             end else begin
15                 count <= count + 1;
16             end
17         end
18         // enable
19     end
20
21 endmodule

```

ઓવરફ્લો સાથે વિસ્તૃત વર્ણન:

```

1 module counter_4bit_enhanced(
2     input clk,
3     input reset,
4     input enable,
5     output reg [3:0] count,
6     output overflow
7 );
8
9     always @(posedge clk or posedge reset) begin
10        if (reset) begin
11            count <= 4'b0000;
12        end else if (enable) begin
13            count <= count + 1; ///
14        end
15    end
16
17    assign overflow = (count == 4'b1111) & enable;
18
19 endmodule

```

કાઉન્ટ સિક્વન્સ:

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

## ટેસ્ટબેન્ક:

```
1 module tb_counter_4bit;
2   reg clk, reset, enable;
3   wire [3:0] count;
4
5   counter_4bit_up dut(.clk(clk), .reset(reset),
6                         .enable(enable), .count(count));
7
8   //
9   initial begin
10     clk = 0;
11     forever #5 clk = ~clk;
12   end
13
14   //
15   initial begin
16     reset = 1; enable = 0;
17     #10 reset = 0; enable = 1;
18     #200 enable = 0; //
19     #20 enable = 1; //
20     #100 $finish;
21   end
22
23   //
24   always @(posedge clk) begin
25     $display("Time=%t Count=%d", $time, count);
26   end
27 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:

```
1 module decoder_3x8_behavioral(
2   input [2:0] address, // 3-
3   input enable, // 
4   output reg [7:0] decode_out // 8-
5 );
6
7   always @(*) begin
8     if (enable) begin
9       case (address)
10         3'b000: decode_out = 8'b00000001; // Y0
11         3'b001: decode_out = 8'b00000010; // Y1
12         3'b010: decode_out = 8'b00000100; // Y2
13         3'b011: decode_out = 8'b00001000; // Y3
14         3'b100: decode_out = 8'b00010000; // Y4
15         3'b101: decode_out = 8'b00100000; // Y5
16         3'b110: decode_out = 8'b01000000; // Y6
17         3'b111: decode_out = 8'b10000000; // Y7
18       default: decode_out = 8'b00000000;
19     endcase
20   end else begin
21     decode_out = 8'b00000000; //
22   end
23 end
```

```
23 end
24
25 endmodule
```

### વૈકલ્પિક અમલીકરણ:

```
1 module decoder_3x8_shift(
2     input [2:0] address,
3     input enable,
4     output [7:0] decode_out
5 );
6
7 assign decode_out = enable ? (8'b00000001 << address) : 8'b00000000;
8
9 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:

```
1 module tb_decoder_3x8;
2     reg [2:0] address;
3     reg enable;
4     wire [7:0] decode_out;
5
6     decoder_3x8_behavioral dut(.address(address), .enable(enable),
7                               .decode_out(decode_out));
8
9     initial begin
10        $monitor("Enable=%b Address=%b | Output=%b",
11                  enable, address, decode_out);
12
13        // Enable = 0
14        enable = 0;
15        for (int i = 0; i < 8; i++) begin
16            address = i;
17            #10;
18        end
19
20        // Enable = 1
21        enable = 1;
22        for (int i = 0; i < 8; i++) begin
23            address = i;
24            #10;
25        end
26
27        $finish;
28    end
29 endmodule
```

## એપ્લિકેશન્સ:

- મેમોરી એડ્રેસિંગ: 8 મેમોરી લોકેશનમાંથી એક પસંદ કરો
- ડિવાઇસ સિલેક્શન: 8 પેરિફેરલ ડિવાઇસમાંથી એક ઇનેબલ કરો
- ડીમાલ્ટિલાઇસિંગ: 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 ડીકોડિંગ: બધા શક્ય ઇનપુટ કોમ્બિનેશન હેન્ડલ

## મેમરી ટ્રીક

"3:8 ડીકોડર - 3 ઇનપુટ્સ 8 આઉટપુટમાંથી 1 પસંદ કરે છે"