

VLSI Technology (4361102) - Summer 2024 Solution

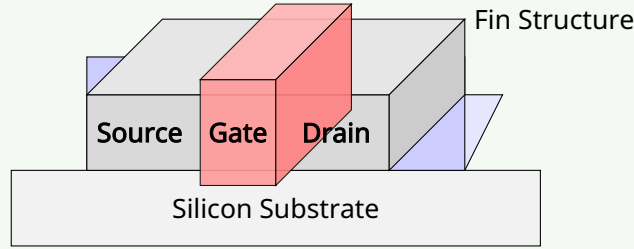
Milav Dabgar

May 16, 2024

પ્રશ્ન 1(a) [3 ગુણ]

FinFET ની રચના દોરો અને તેના ફાયદા લખો.

જવાબ



આકૃતિ 1. FinFET Structure

ફાયદા:

ફાયદો	વર્ણન
બેહતર નિયંત્રણ	ગુણાકાર gates બેહતર channel control આપે છે
ઘટાડેલ લીકેજ	3D રચનાના કારણે ઓછું off-state current
સુધારેલ કામગીરી	વધુ drive current અને ઝડપી switching

કોષ્ટક 1. FinFET Advantages

મેમરી ટ્રીક

“BCR - Better Control Reduces leakage”

પ્રશ્ન 1(b) [4 ગુણ]

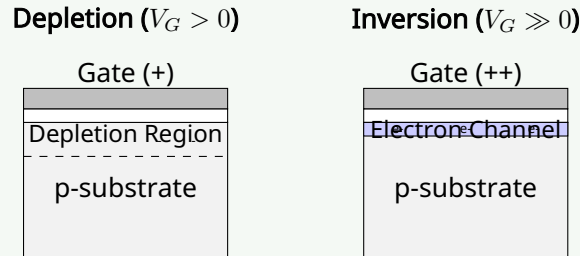
એક્સટર્નલ બાયઝ હેડળ MOS રચનાનું ડેપ્લીશન અને ઇનવર્શન સમજાવો

જવાબ

કોષ્ટક: MOS બાયઝ પરિસ્થિતિઓ

બાયઝ પ્રકાર	ગેટ વોલ્ટેજ	ચેનલ સ્થિતિ	ચાર્જ કેરિયર્સ
ડેપ્લીશન	થોડું પોઝિટિવ	Depleted	Holes દૂર ધકેલાય છે
ઇન્વર્શન	વધુ પોઝિટિવ	Inverted	Electrons આકર્ષાય છે

કોષ્ટક 2. MOS Bias Conditions



આકૃતિ 2. MOS Depletion and Inversion Layers

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

મેમરી ટ્રીક

“DI - Depletion Inverts to conducting channel”

પ્રશ્ન 1(c) [7 ગુણ]

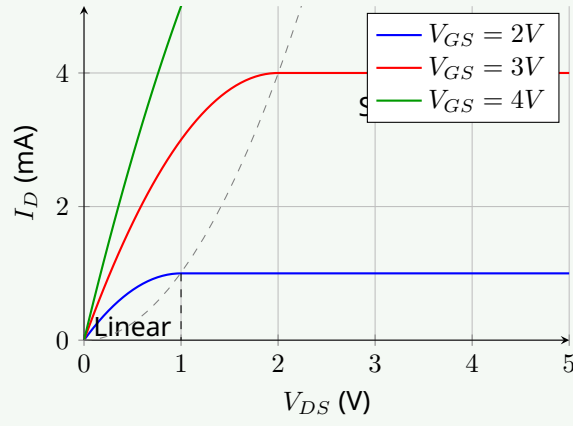
n-ચેનલ MOSFET ને તેની કરંટ-વોલ્ટેજ લાક્ષણિકતાઓની મદદથી સમજાવો.

જવાબ

કોષ્ટક: MOSFET ઓપરેટિંગ વિભાગો

વિભાગ	શરત	ડ્રેઇન કરંટ	લાક્ષણિકતાઓ
કટ-ઓફ	$V_{GS} < V_{TH}$	$I_D \approx 0$	કોઈ conduction નથી
લિનિયર	$V_{DS} < V_{GS} - V_{TH}$	$I_D \propto V_{DS}$	Resistive વર્તન
સેચ્યુરેશન	$V_{DS} \geq V_{GS} - V_{TH}$	$I_D \propto (V_{GS} - V_{TH})^2$	કરંટ V_{DS} પર આધારિત નથી

કોષ્ટક 3. MOSFET Operating Regions



આકૃતિ 3. MOSFET IV Characteristics

મુખ્ય સમીકરણો:

- **Linear:** $I_D = \mu_n C_{ox} \frac{W}{L} [(V_{GS} - V_{TH})V_{DS} - \frac{V_{DS}^2}{2}]$
- **Saturation:** $I_D = \frac{1}{2} \mu_n C_{ox} \frac{W}{L} (V_{GS} - V_{TH})^2$

મેમરી ટ્રીક

“CLS - Cut-off, Linear, Saturation regions”

પ્રશ્ન 1(c 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.
કોષ્ટક: સ્કેલિંગ તુલના

પેરામીટર	Full Voltage Scaling	Constant Voltage Scaling
વોલ્ટેજ	α દ્વારા ઘટાડાય છે	સ્થિર રહે છે
પાવર ડેન્સિટી	સ્થિર	α દ્વારા વધે છે
ઇલેક્ટ્રિક ફિલ્ડ	સ્થિર	α દ્વારા વધે છે
પરફોર્મન્સ	બેહતર	મધ્યમ સુધારો

કોષ્ટક 4. Scaling Comparison

ગેરફાયદા:

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

મેમરી ટ્રીક

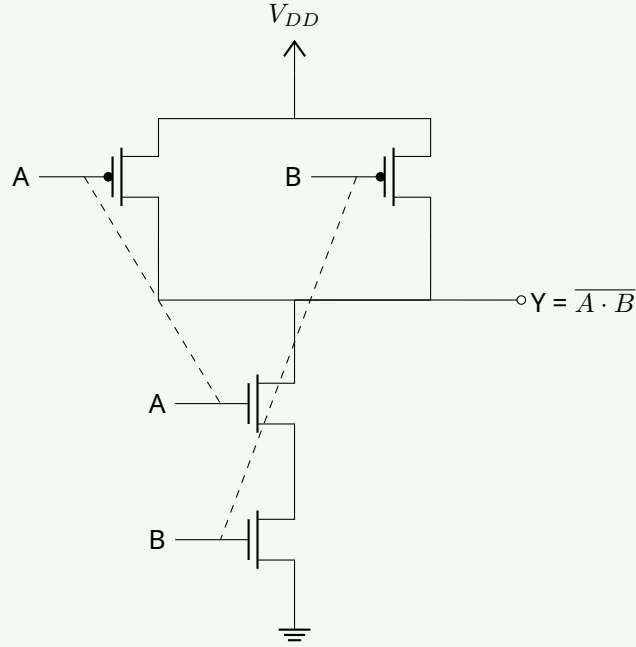
“SHQ - Short channel, Hot carriers, Quantum effects”

પ્રશ્ન 2(a) [3 ગુણ]

CMOS ની મદદથી બે ઇનપુટ NAND ગેટ દોરો.

જવાબ

CMOS NAND gate બે parallel pMOS ટ્રાન્ઝિસ્ટર અને બે series nMOS ટ્રાન્ઝિસ્ટરનો ઉપયોગ કરે છે.



આકૃતિ 4. 2-Input CMOS NAND Gate

કોષ્ટક: NAND સત્ય કોષ્ટક

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

કોષ્ટક 5. NAND Truth Table

મેમરી ટ્રીક

“PP-SS: Parallel PMOS, Series NMOS”

પ્રશ્ન 2(b) [4 ગુણ]

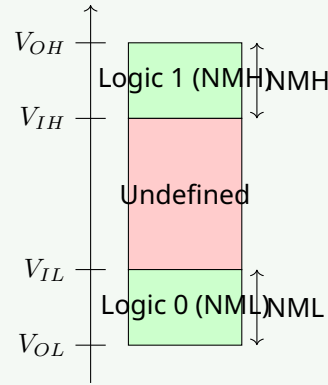
nMOS ઇન્વર્ટર માટે નોઇઝ ઇમ્યુનિટી અને નોઇઝ માર્જિન સમજાવો.

જવાબ

કોષ્ટક: નોઇઝ પેરામીટર્સ

પેરામીટર	વ્યાખ્યા	ફોર્મ્યુલા
NMH	હાઇ નોઇઝ માર્જિન	$V_{OH} - V_{IH}$
NML	લો નોઇઝ માર્જિન	$V_{IL} - V_{OL}$
નોઇઝ ઇમ્યુનિટી	નોઇઝ રિજેક્ટ કરવાની ક્ષમતા	$\text{Min}(\text{NMH}, \text{NML})$

કોષ્ટક 6. Noise Parameters



આકૃતિ 5. Noise Margins

- V_{IL} : મહત્તમ લો ઇનપુટ વોલ્ટેજ.
- V_{IH} : લઘુત્તમ હાઇ ઇનપુટ વોલ્ટેજ.
- સારી નોઇઝ ઇમ્યુનિટી: મોટા નોઇઝ માર્જિન ખોટી switching ને રોકે છે.

મેમરી ટ્રીક

“HILOL - High/Low Input/Output Levels”

પ્રશ્ન 2(c) [7 ગુણ]

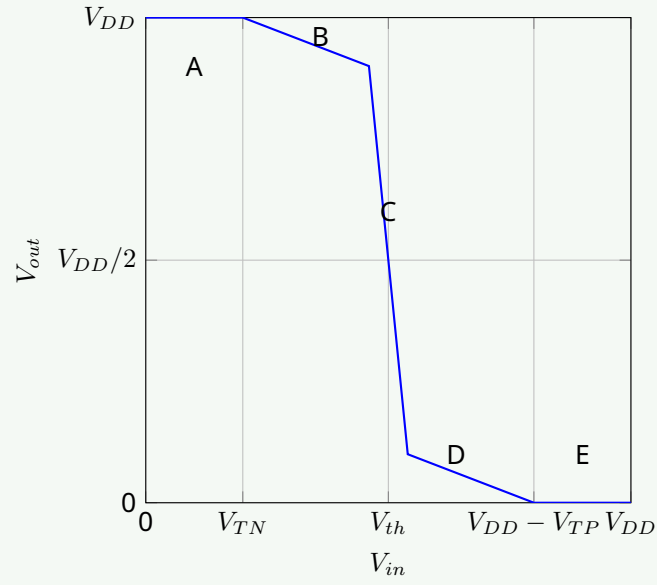
CMOS ઇન્વર્ટરની વોલ્ટેજ ટ્રાન્સફર લાક્ષણિકતાઓ (VTC) સમજાવો.

જવાબ

કોષ્ટક: VTC વિભાગો

વિભાગ	ઇનપુટ રેન્જ	આઉટપુટ	ટ્રાન્ઝિસ્ટર સ્થિતિઓ
A	0 to V_{TN}	V_{DD}	pMOS ON, nMOS OFF
B	V_{TN} to $V_{DD}/2$	ટ્રાન્ઝિશન	બંને આંશિક રીતે ON
C	$V_{DD}/2$	ટ્રાન્ઝિશન	બંને Saturation
D	$V_{DD}/2$ to $V_{DD} - V_{TP} $	ટ્રાન્ઝિશન	બંને આંશિક રીતે ON
E	$V_{DD} - V_{TP} $ to V_{DD}	0V	pMOS OFF, nMOS ON

કોષ્ટક 7. VTC Regions



આકૃતિ 6. CMOS Inverter VTC

મુખ્ય લક્ષણો:

- તીક્ષ્ણ ટ્રાન્ઝિશન: આદર્શ switching વર્તન.
- હાઇ ગેઇન: ટ્રાન્ઝિશન વિભાગમાં મોટો slope.
- રેઇલ-ટુ-રેઇલ: આઉટપુટ સંપૂર્ણ સપ્લાય રેન્જમાં swing કરે છે (0 to V_{DD}).

મેમરી ટ્રીક

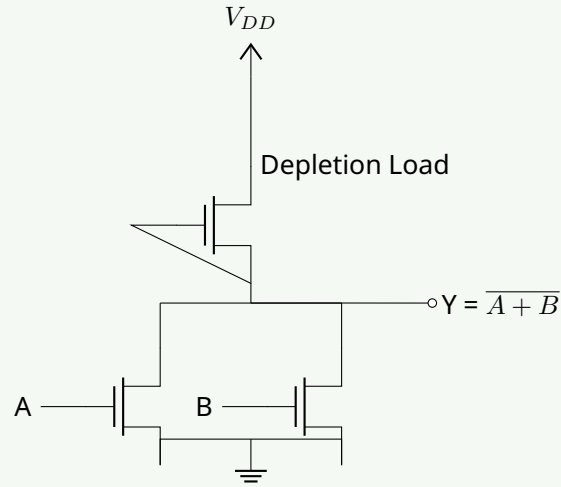
“ASH - A-region, Sharp transition, High gain”

પ્રશ્ન 2(a OR) [3 ગુણ]

ડિપ્લીશન લોડ nMOS નો ઉપયોગ કરીને NOR2 ગેટનો અમલ કરો.

જવાબ

Depletion load NOR2 એ depletion nMOS ને લોડ તરીકે અને બે parallel nMOS ને drivers તરીકે વાપરે છે.



આકૃતિ 7. Depletion Load NOR2

કોષ્ટક: NOR2 સત્ય કોષ્ટક

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

કોષ્ટક 8. NOR2 Truth Table

મેમરી ટ્રીક

“DPN - Depletion load, Parallel NMOS”

પ્રશ્ન 2(b OR) [4 ગુણ]

એન્હાન્સમેન્ટ લોડ ઇન્વર્ટર અને ડિપ્લીશન લોડ ઇન્વર્ટર વચ્ચે તફાવત શોધો.

જવાબ

કોષ્ટક: લોડ ઇન્વર્ટર તુલના

પેરામીટર	એન્હાન્સમેન્ટ લોડ	ડિપ્લીશન લોડ
થ્રેશોલ્ડ વોલ્ટેજ	$V_T > 0$	$V_T < 0$
ગેટ કનેક્શન	$V_{GS} = V_{DS}$	$V_{GS} = 0$
લોજિક હાઇ (V_{OH})	$V_{DD} - V_T$	V_{DD}
પાવર કન્ઝમ્પશન	વધુ	ઓછું
સ્વિચિંગ સ્પીડ	ધીમું	ઝડપી

કોષ્ટક 9. Inverter Load Types Comparison

મેમરી ટ્રીક

“EPDLH - Enhancement Positive, Depletion Lower power, Higher speed”

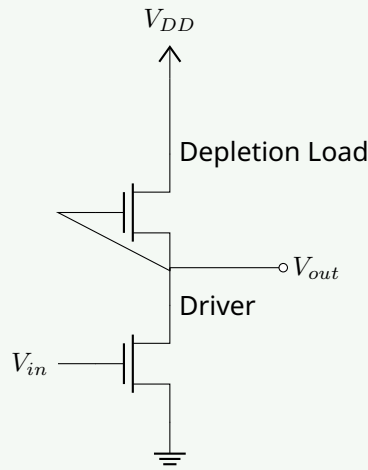
પ્રશ્ન 2(c OR) [7 ગુણ]

ડિપ્લીશન લોડ nMOS ઇન્વર્ટરને તેના VTC સાથે સમજાવો.

જવાબ

સર્કિટ ઓપરેશન:

- લોડ ટ્રાન્ઝિસ્ટર: હંમેશા conducting ($V_{GS} = 0, V_T < 0$).
- ડ્રાઇવર ટ્રાન્ઝિસ્ટર: ઇનપુટ વોલ્ટેજ દ્વારા નિયંત્રિત.
- આઉટપુટ: વોલ્ટેજ ડિવાઇડર એક્શન દ્વારા નક્કી થાય છે.



આકૃતિ 8. Depletion Load Inverter

કોષ્ટક: ઓપરેટિંગ પોઇન્ટ્સ

ઇનપુટ સ્થિતિ	ડ્રાઇવર	લોડ	આઉટપુટ
$V_{in} = 0$	OFF	ON (Linear)	V_{DD}
$V_{in} = V_{DD}$	ON (Linear)	ON (Sat)	$\approx 0V$

કોષ્ટક 10. Inverter Operating Points

VTC લાક્ષણિકતાઓ:

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

મેમરી ટ્રીક

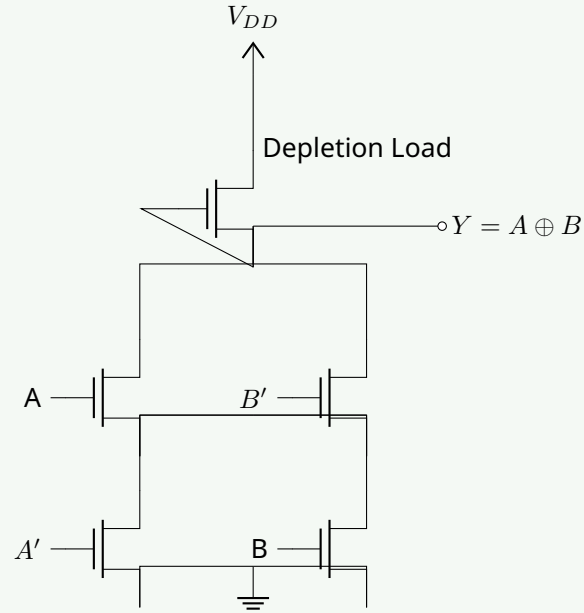
“DLB - Depletion Load gives Better high output”

પ્રશ્ન 3(a) [3 ગુણ]

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

જવાબ

XOR ($Y = A \oplus B$) અમલ કરવા માટે, આપણે XNOR નું pull-down network બનાવીએ છીએ.



આકૃતિ 9. Depletion Load NMOS EX-OR Gate

કોષ્ટક: XOR સત્ય કોષ્ટક

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

કોષ્ટક 11. XOR Truth Table

મેમરી ટ્રીક

“XOR - eXclusive OR, different inputs give 1”

પ્રશ્ન 3(b) [4 ગુણ]

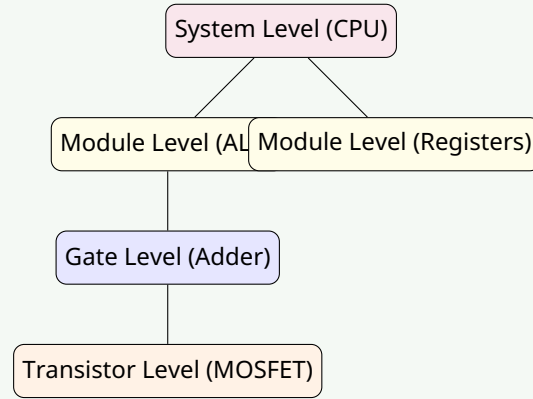
ડિઝાઇન હાઇરાકીને ઉદાહરણ સાથે સમજાવો.

જવાબ

કોષ્ટક: હાઇરાકી લેવલ્સ

લેવલ	કમ્પોનન્ટ	ઉદાહરણ
સિસ્ટમ	સંપૂર્ણ ચિપ	માઇક્રોપ્રોસેસર
મોડ્યુલ	ફંક્શનલ બ્લોક્સ	ALU, મેમરી
ગેટ	લોજિક ગેટ્સ	NAND, NOR
ટ્રાન્ઝિસ્ટર	વ્યક્તિગત ડિવાઇસેસ	MOSFET

કોષ્ટક 12. Design Hierarchy Levels



આકૃતિ 10. Design Hierarchy Tree

ફાયદા:

- મોડ્યુલારિટી: સ્વતંત્ર ડિઝાઇન અને ટેસ્ટિંગ.
- પુનઃઉપયોગ: સામાન્ય બ્લોક્સ (જેમ કે adders) ઘણી વખત વપરાય છે.
- જાળવણીયોગ્યતા: સરળ debugging અને modification.

મેમરી ટ્રીક

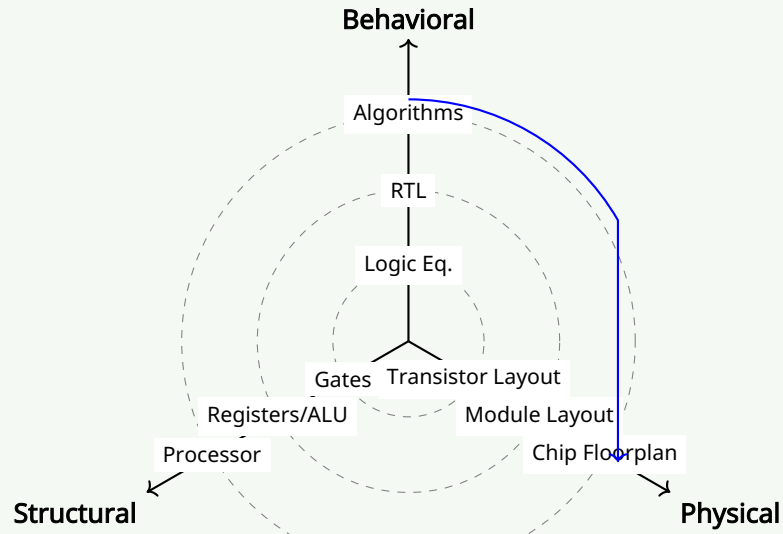
"SMG-T: System, Module, Gate, Transistor levels"

પ્રશ્ન 3(c) [7 ગુણ]

Y ચાર્ટ ડિઝાઇન ફલો દોરો અને સમજાવો.

જવાબ

Y-ચાર્ટ VLSI ડિઝાઇનના ત્રણ ડોમેન્સ દર્શાવે છે: Behavioral, Structural, અને Physical.



આકૃતિ 11. Gajski-Kuhn Y-Chart

કોષ્ટક: Y-ચાર્ટ ડોમેન્સ

ડોમેન	વર્ણન	ઉદાહરણો
બિહેવિયરલ	સિસ્ટમ શું કરે છે	Algorithms, RTL
સ્ટ્રક્ચરલ	તે કેવી રીતે ગોઠવાયેલું છે	Architecture, Gates
ફિઝિકલ	કમ્પોનન્ટ્સ ક્યાં મૂકાયેલા છે	Floorplan, Layout

કોષ્ટક 13. Y-Chart Domains

ડિઝાઇન ફ્લો:

- ટોપ-ડાઉન: Behavioral → Structural → Physical.
- બોટમ-અપ: Physical constraints ઉપરના લેવેલ્સને પ્રભાવિત કરે છે.

મેમરી ટ્રીક

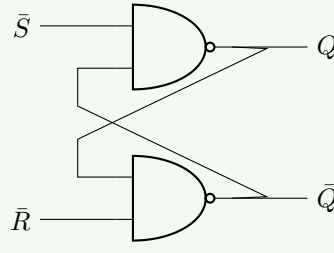
“BSP - Behavioral, Structural, Physical domains”

પ્રશ્ન 3(a OR) [3 ગુણ]

CMOS નો ઉપયોગ કરીને NAND2 - SR લેચનો અમલ કરો

જવાબ

NAND SR Latch બે cross-coupled NAND gates નો ઉપયોગ કરે છે.



આકૃતિ 12. CMOS NAND SR Latch

કોષ્ટક: SR લેચ ઓપરેશન

S	R	Q	Q'	સ્થિતિ
0	0	1	1	અમાન્ય (Invalid)
0	1	1	0	સેટ (Set)
1	0	0	1	રીસેટ (Reset)
1	1	Q	Q'	હોલ્ડ (Hold)

કોષ્ટક 14. NAND SR Latch Truth Table (Active Low Inputs)

મેમરી ટ્રીક

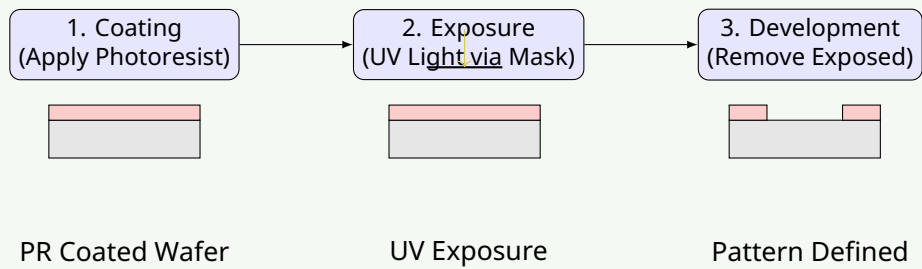
"SR-HRI: Set, Reset, Hold, Invalid states"

પ્રશ્ન 3(b OR) [4 ગુણ]

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

જવાબ

પદ્ધતિ: લિથોગ્રાફી (Photolithography) એ માસ્ક પરની ભૌમિતિક રચનાઓને સિલિકોન વેફરની સપાટી પર ટ્રાન્સફર કરવા માટે વપરાય છે.



આકૃતિ 13. Photolithography Process

પ્રક્રિયાના પગલાં:

- **Coating:** ફોટોરેસિસ્ટનું પાતળું સ્તર લગાવવું.
- **Exposure:** માસ્ક દ્વારા UV light થી રેસિસ્ટને expose કરવું.
- **Development:** Exposed (positive) અથવા unexposed (negative) રેસિસ્ટને દૂર કરવું.

મેમરી ટ્રીક

"CED - Coating, Exposure, Development"

પ્રશ્ન 3(c OR) [7 ગુણ]

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

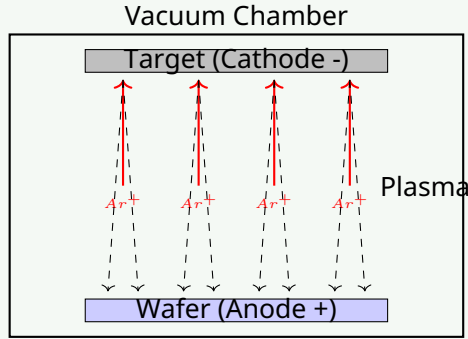
જવાબ

કોષ્ટક: મેટલ ડિપોઝિશન પદ્ધતિઓ

પદ્ધતિ	ટેકનિક	ઉપયોગ
PVD	Sputtering, Evaporation	એલ્યુમિનિયમ, કોપર
CVD	CVD, PECVD	ટંગસ્ટન, ટાઇટેનિયમ
Electroplating	Electrochemical	કોપર interconnects

કોષ્ટક 15. Metal Deposition Methods

Sputtering Process: Sputtering માં, પ્લાઝ્માંથી આયનો ટાર્ગેટ મટેરિયલ તરફ પ્રવેગિત થાય છે, જેનાથી એટમ્સ બહાર નીકળે છે અને વેક્યુમ પર જમા થાય છે.



આકૃતિ 14. Sputtering System

ફાયદા:

- સમાન જાડાઈ: ઉત્તમ step coverage.
- ઓછું તાપમાન: ડિવાઇસ integrity જાળવે છે.
- વિવિધતા: Alloys અને compounds નું ડિપોઝિશન શક્ય.

મેમરી ટ્રીક

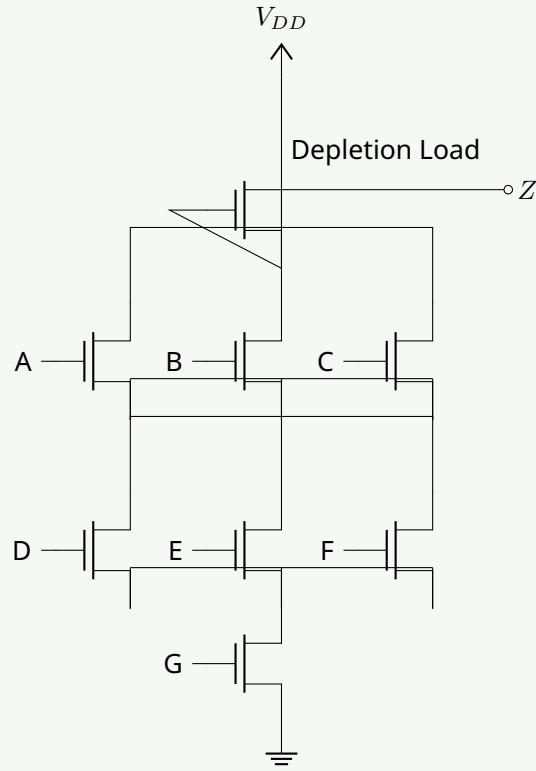
"IBE-DC: Ion Bombardment Ejects atoms for Deposition Control"

પ્રશ્ન 4(a) [3 ગુણ]

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

જવાબ

લોજિક ફંક્શન $Z = \overline{(A + B + C) \cdot (D + E + F) \cdot G}$ ને depletion load અને pull-down network દ્વારા અમલમાં મુકાય છે. 1. Parallel inputs A, B, C (OR) 2. Parallel inputs D, E, F (OR) 3. Single input G આ બધા સીરિઝમાં જોડાયેલા છે (AND operation).



આકૃતિ 15. Logic Implementation of Z

મેમરી ટ્રીક

“POI - Parallel OR, Inversion at output”

પ્રશ્ન 4(b) [4 ગુણ]

VERILOG માં વપરાતી ડિઝાઇન શૈલીઓની સૂચિ બનાવો અને સમજાવો.

જવાબ

કોષ્ટક: વેરિલોગ ડિઝાઇન શૈલીઓ

શૈલી	વર્ણન	ઉપયોગનો કેસ	ઉદાહરણ
બિહેવિયરલ	એલ્ગોરિધમ વર્ણન	ઉચ્ચ-સ્તરીય મોડેલિંગ	always blocks
ડેટાફ્લો	બૂલિયન expressions	કમ્પિનેશનલ લોજિક	assign statements
સ્ટ્રક્ચરલ	કમ્પોનન્ટ instantiation	હાઇરાર્કિકલ ડિઝાઇન	module connections
ગેટ-લેવલ	પ્રિમિટિવ ગેટ્સ	લો-લેવલ ડિઝાઇન	and, or gates

કોષ્ટક 16. Verilog Design Styles

લાક્ષણિકતાઓ:

- બિહેવિયરલ: સર્કિટ શું કરે છે (functionality) તેનું વર્ણન.
- સ્ટ્રક્ચરલ: કમ્પોનન્ટ્સ કેવી રીતે જોડાય છે (netlist) તે બતાવે છે.

મેમરી ટ્રીક

``BDSG - Behavioral, Dataflow, Structural, Gate-level"

પ્રશ્ન 4(c) [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 Latch (Verilog):

```

1 module nand_sr_latch(
2   input S, R,
3   output Q, Q_bar
4 );
5   nand(Q, S, Q_bar);
6   nand(Q_bar, R, Q);
7 endmodule

```

NOR2 SR Latch (Verilog):

```

1 module nor_sr_latch(
2   input S, R,
3   output Q, Q_bar
4 );
5   nor(Q_bar, R, Q);
6   nor(Q, S, Q_bar);
7 endmodule

```

તફાવત:

- **NAND:** Low Inputs સાથે Set/Reset (Active Low).
- **NOR:** High Inputs સાથે Set/Reset (Active High).

મેમરી ટ્રીક

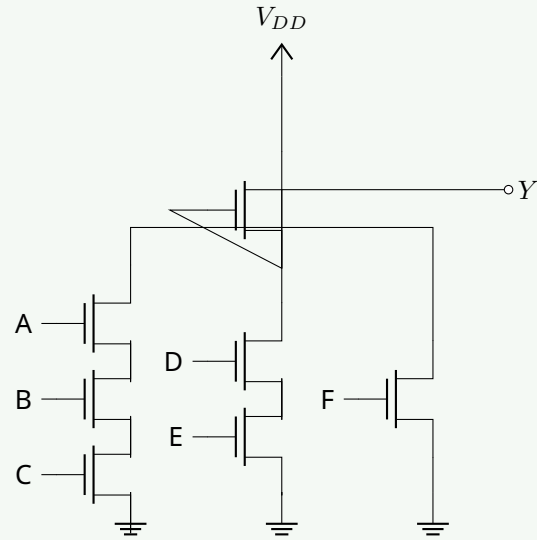
``NAND-Low, NOR-High active"

પ્રશ્ન 4(a OR) [3 ગુણ]

Y= (ABC + DE + F)' ને ડિપ્લીશન nMOS લોડ સાથે અમલમાં મૂકો.

જવાબ

લોજિક ફંક્શન $Y = \overline{(ABC) + (DE) + F}$ એ Sum of Products (ORing of AND terms) દર્શાવે છે. Pull-down network માં ત્રણ parallel branches હશે: 1. A, B, C series માં (AND) 2. D, E series માં (AND) 3. F single



આકૃતિ 16. Logic Implementation of Y

મેમરી ટ્રીક

``SSS-I: Series-Series-Single with Inversion"

પ્રશ્ન 4(b OR) [4 ગુણ]

કુલ એડરને અમલમાં મૂકવા માટે વેરિલોગ કોડ લખો.

જવાબ

```

1 module full_adder(
2     input a, b, cin,
3     output sum, cout
4 );
5     assign sum = a ^ b ^ cin;
6     assign cout = (a & b) | (cin & (a ^ b));
7 endmodule

```

લોજિક ફંક્શન-સ:

- Sum: Triple XOR operation ($A \oplus B \oplus C_{in}$)
- Carry: Majority function of inputs.

મેમરી ટ્રીક

``XOR-Sum, Majority-Carry"

પ્રશ્ન 4(c OR) [7 ગુણ]

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

જવાબ

4:1 Multiplexer Verilog Code (Assuming S2 corresponds to S0):

```

1 // 4:1 Multiplexer implementation
2 module mux_4to1(
3     input [1:0] sel, // S1, S0
4     input [3:0] data, // I3, I2, I1, I0
5     output Y
6 );
7     assign Y = (sel == 2'b00) ? data[0] :
8         (sel == 2'b01) ? data[1] :
9         (sel == 2'b10) ? data[2] :
10            data[3];
11 endmodule

```

કોષ્ટક: મલ્ટિપ્લેક્સર સિલેક્શન

S1	S0	પસંદ કરેલ ઇનપુટ	આઉટપુટ
0	0	I0	$Y = I_0$
0	1	I1	$Y = I_1$
1	0	I2	$Y = I_2$
1	1	I3	$Y = I_3$

કોષ્ટક 17. Multiplexer Selection

મેમરી ટ્રીક

“DAO - Decoder, AND gates, OR combination”

પ્રશ્ન 5(a) [3 ગુણ]

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

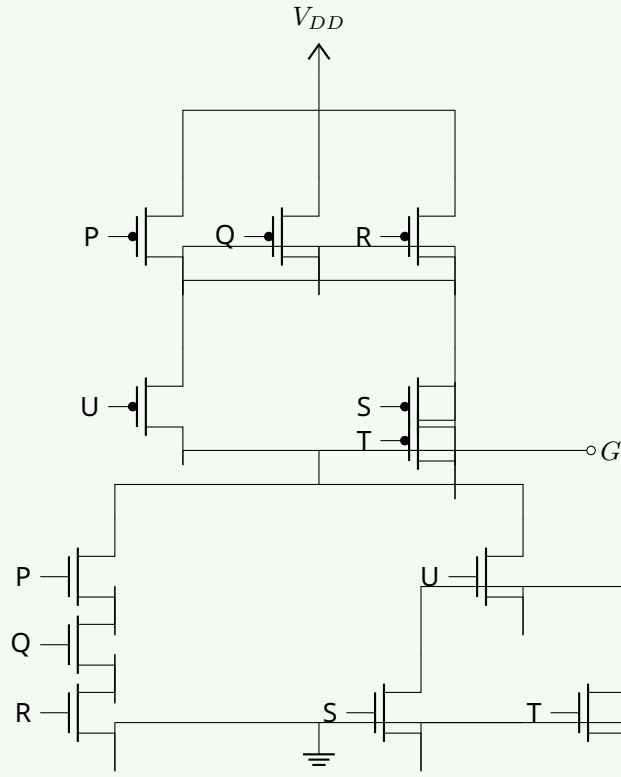
જવાબ

ફંક્શન $G = \overline{(PQR) + U(S+T)}$ માટે CMOS અમલીકરણ: **Pull-Down Network (PDN):** $(PQR) + U(S+T)$ નું અમલીકરણ (nMOS).

- P, Q, R Series માં (AND).
- S, T Parallel માં (OR).
- U એ $(S||T)$ સાથે Series માં.
- બ્લોક $(P - Q - R)$ એ $(U - (S||T))$ સાથે Parallel માં.

Pull-Up Network (PUN): Dual નું અમલીકરણ (pMOS). W

- P, Q, R Parallel માં.
- S, T Series માં.
- U એ $(S - T)$ સાથે Parallel માં.
- બ્લોક $(P||Q||R)$ એ $(U||(S - T))$ સાથે Series માં.



આકૃતિ 17. CMOS Implementation of Logic G

મેમરી ટ્રીક

``PSSP - Parallel Series Series Parallel''

પ્રશ્ન 5(b) [4 ગુણ]

વેરિલોગનો ઉપયોગ કરીને 8×1 મલ્ટિપ્લેક્સર અમલમાં મૂકો.

જવાબ

```

1 module mux_8to1(
2   input [2:0] sel, // 3-bit select
3   input [7:0] data, // 8 data inputs
4   output reg Y // Output
5 );
6 always @(*) begin
7   case(sel)
8     3'b000: Y = data[0];
9     3'b001: Y = data[1];
10    3'b010: Y = data[2];
11    3'b011: Y = data[3];
12    3'b100: Y = data[4];
13    3'b101: Y = data[5];
14    3'b110: Y = data[6];
15    3'b111: Y = data[7];
16  endcase
17 end
18 endmodule

```

કોષ્ટક: 8:1 MUX સિલેક્શન

S2	S1	S0	Output
0	0	0	data[0]
0	0	1	data[1]
1	0	0	data[4]
1	1	1	data[7]

કોષ્ટક 18. 8:1 MUX Truth Table (Partial)

મેમરી ટ્રીક

“Case-Always: Use case statement in always block”

પ્રશ્ન 5(c) [7 ગુણ]

વેરિલોગમાં સ્ટ્રક્ચરલ મોડેલિંગ શૈલીનો ઉપયોગ કરીને 4 બીટ કુલ એડરને લાગૂ કરો.

જવાબ

Verilog Code (Structural):

```

1  module full_adder(
2      input a, b, cin,
3      output sum, cout
4  );
5      assign sum = a ^ b ^ cin;
6      assign cout = (a & b) | (cin & (a ^ b));
7  endmodule
8
9  module full_adder_4bit(
10     input [3:0] a, b,
11     input cin,
12     output [3:0] sum,
13     output cout
14 );
15     wire c1, c2, c3;
16
17     // Instantiating 4 Full Adders
18     full_adder fa0(.a(a[0]), .b(b[0]), .cin(cin),
19                   .sum(sum[0]), .cout(c1));
20     full_adder fa1(.a(a[1]), .b(b[1]), .cin(c1),
21                   .sum(sum[1]), .cout(c2));
22     full_adder fa2(.a(a[2]), .b(b[2]), .cin(c2),
23                   .sum(sum[2]), .cout(c3));
24     full_adder fa3(.a(a[3]), .b(b[3]), .cin(c3),
25                   .sum(sum[3]), .cout(cout));
26 endmodule

```

કોષ્ટક: 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

કોષ્ટક 19. Ripple Carry Structure

મેમરી ટ્રીક

"RCC - Ripple Carry Chain connection"

પ્રશ્ન 5(a OR) [3 ગુણ]

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

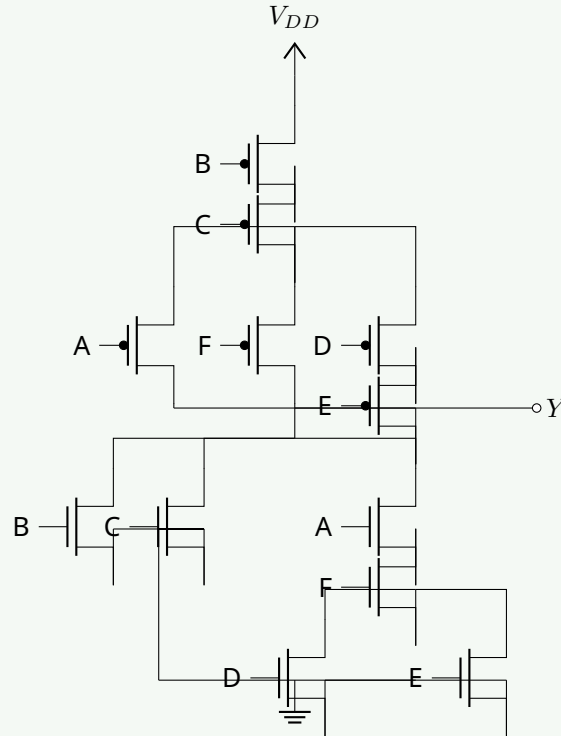
જવાબ

ફંક્શન: $Y = \overline{(A \cdot F \cdot (D + E)) + (B + C)}$. PDN (Implementation of Function):

- Block 1: B, C Parallel માં (OR).
- Block 2: D, E Parallel માં (OR) → Series માં A, F સાથે (AND).
- Top level: Block 1 || Block 2.

PUN (Implementation of Dual):

- Block 1: B, C Series માં.
- Block 2: D, E Series માં → Parallel માં A, F સાથે.
- Top level: Block 1 - Block 2 (Series).



આકૃતિ 18. CMOS Implementation of Y

મેમરી ટ્રીક

``PNAI - PMOS Network Applies Inversion"

પ્રશ્ન 5(b OR) [4 ગુણ]

વેરિલોગનો ઉપયોગ કરીને 4 બીટ અપ કાઉન્ટર અમલમાં મૂકવું

જવાબ

```

1 module counter_4bit_up(
2   input clk, reset,
3   output reg [3:0] count
4 );
5   always @(posedge clk or posedge reset) begin
6     if (reset)
7       count <= 4'b0000;
8     else
9       count <= count + 1;
10  end
11 endmodule

```

કોષ્ટક: કાઉન્ટર સિક્વન્સ

Clock	Reset	Count	Next Count
↑	1	X	0000
↑	0	0000	0001
↑	0	1111	0000

કોષ્ટક 20. Counter Sequence

મેમરી ટ્રીક

``SRA - Synchronous Reset with Auto rollover"

પ્રશ્ન 5(c OR) [7 ગુણ]

વેરિલોગમાં બિહેવિયરલ મોડેલિંગ સ્ટાઈલનો ઉપયોગ કરીને 3:8 ડીકોડરનો અમલ કરો

જવાબ

```

1 module decoder_3to8(
2   input [2:0] select,
3   input enable,
4   output reg [7:0] out
5 );
6   always @(*) begin
7     if (enable) begin
8       case(select)
9         3'b000: out = 8'b00000001;
10        3'b001: out = 8'b00000010;
11        3'b010: out = 8'b00000100;

```

```
12      3'b011: out = 8'b00001000;  
13      3'b100: out = 8'b00010000;  
14      3'b101: out = 8'b00100000;  
15      3'b110: out = 8'b01000000;  
16      3'b111: out = 8'b10000000;  
17      default: out = 8'b00000000;  
18      endcase  
19      end else begin  
20          out = 8'b00000000;  
21      end  
22      end  
23      endmodule
```

ઉપયોગો:

- મેમરી એડ્રેસિંગ: Chip select generation.
- ડેટા રાઉટિંગ: Channel selection.

મેમરી ટ્રીક

``BEOH - Behavioral Enable One-Hot decoder"