

VLSI Technology (4361102) - Summer 2025 Solution

Milav Dabgar

May 12, 2025

પ્રશ્ન 1(a) [3 ગુણ]

State importance of scaling

જવાબ

સ્કેલિંગ semiconductor technology ને આગળ વધારવા અને device performance સુધારવા માટે અત્યંત મહત્વપૂર્ણ છે.

કોષ્ટક 1. Scaling Benefits

સ્કેલિંગ ફાયદા	વર્ણન
Device Size	ઊંચી density માટે transistor dimensions ઘટાડે છે
Speed	ટૂંકી channel length થી ઝડપી switching
Power	પ્રતિ operation ઓછો power consumption
Cost	વધુ chips per wafer, function દીઠ ઓછો cost

- **Technology advancement:** Moore's Law ચાલુ રાખવામાં સક્ષમ બનાવે છે
- **Performance boost:** ઊંચી frequency operation શક્ય બનાવે છે
- **Market competitiveness:** નાના, ઝડપી, સસ્તા products

મેમરી ટ્રીક

"Small Devices Speed Progress Cheaply"

પ્રશ્ન 1(b) [4 ગુણ]

Compare Planar MOSFET and FinFET

જવાબ

FinFET technology નાના nodes પર planar MOSFET ની મર્યાદાઓનો ઉકેલ આપે છે.

કોષ્ટક 2. Planar MOSFET vs FinFET

પેરામીટર	Planar MOSFET	FinFET
Structure	2D flat channel	3D fin-shaped channel
Gate Control	Single gate	Tri-gate/multi-gate
Short Channel Effects	નાના nodes પર ઊંચી	નોંધપાત્ર રીતે ઓછી
Leakage Current	ઊંચી subthreshold leakage	ખૂબ ઓછી leakage

- **Scalability:** FinFET sub-22nm technology nodes શક્ય બનાવે છે
- **Power efficiency:** FinFET વધુ સારો power-performance ratio આપે છે

- **Manufacturing:** FinFET વધુ જટિલ fabrication માંગે છે

મેમરી ટ્રીક

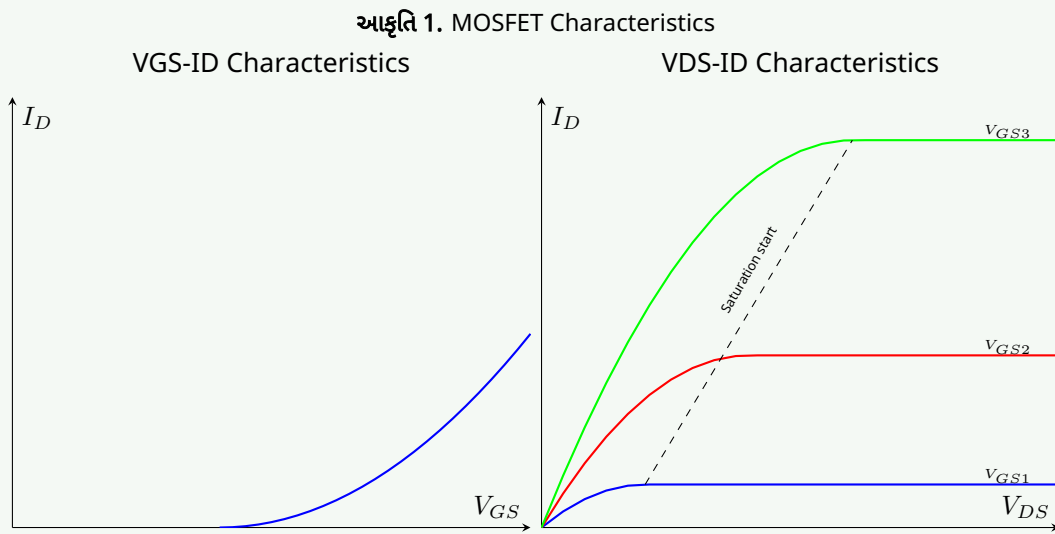
“Fins Control Current Better Than Flat”

પ્રશ્ન 1(c) [7 ગુણ]

Draw and Explain VDS-ID AND VGS-ID characteristics of N channel MOSFET

જવાબ

N-channel MOSFET characteristics અલગ અલગ operating regions માં device behavior દર્શાવે છે.
Diagram:



કોષ્ટક 3. MOSFET Operating Regions

પ્રદેશ	સ્થિતિ	કરંટ સમીકરણ
Cutoff	$V_{GS} < V_T$	$I_D = 0$
Linear	$V_{DS} < (V_{GS} - V_T)$	$I_D \propto V_{DS}$
Saturation	$V_{DS} \geq (V_{GS} - V_T)$	$I_D \propto (V_{GS} - V_T)^2$

- **Cutoff:** કોઈ પ્રવાહ વહેતો નથી, open switch તરીકે વર્તે છે.
- **Linear/Triode:** પ્રવાહ V_{DS} સાથે રેખીય રીતે વધે છે, resistor તરીકે વર્તે છે.
- **Saturation:** પ્રવાહ અચળ રહે છે, V_{DS} થી સ્વતંત્ર, current source તરીકે વર્તે છે.

મેમરી ટ્રીક

“Threshold Gates Linear Saturation”

OR

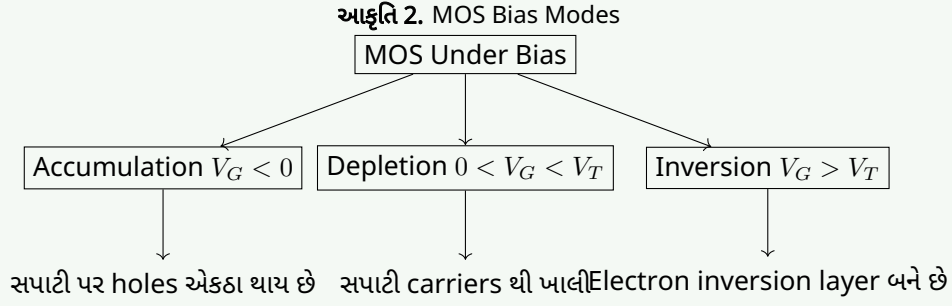
પ્રશ્ન 1(c) [7 ગુણ]

Explain different condition of MOS under external bias

જવાબ

External bias અલગ અલગ charge distributions બનાવે છે જે MOS capacitor behavior ને અસર કરે છે.

Diagram:



કોષ્ટક 4. MOS Operating Modes

બાયસ સ્થિતિ	સપાટીની સ્થિતિ	કેપેસિટન્સ
Accumulation	સપાટી પર majority carriers	ઊંચી (C_{ox})
Depletion	કોઈ mobile carriers નથી	મધ્યમ
Inversion	Minority carriers channel બનાવે છે	ઊંચી (C_{ox})

- ફ્લેટ બેન્ડ વોલ્ટેજ: કોઈ charge separation અસ્તિત્વમાં નથી
- એનર્જી બેન્ડ બેન્ડિંગ: carrier distribution નક્કી કરે છે
- સપાટીનો વિભવ: inversion layer formation નિયંત્રિત કરે છે

મેમરી ટ્રીક

“Accumulate, Deplete, then Invert”

પ્રશ્ન 2(a) [3 ગુણ]

Draw voltage transfer characteristic of ideal inverter

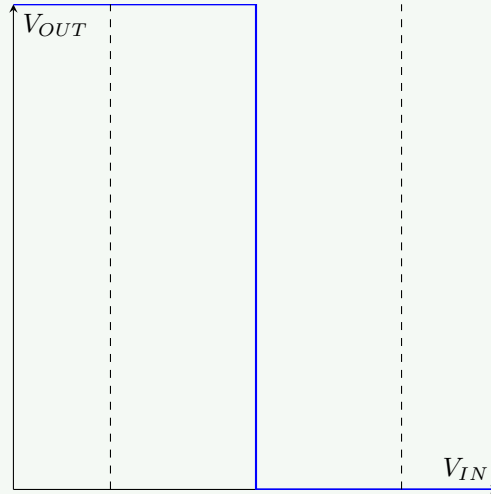
જવાબ

આદર્શ ઇન્વર્ટર infinite gain સાથે logic levels વચ્ચે તીક્ષ્ણ પરિવર્તન આપે છે.

Diagram:

આકૃતિ 3. Ideal Inverter VTC

Ideal Inverter VTC



- તીક્ષ્ણ પરિવર્તન: switching point પર infinite slope
- નોઈઝ માર્જિન: $NMH = V_{OH} - V_{IH}$, $NML = V_{IL} - V_{OL}$
- સંપૂર્ણ લોજિક લેવલ: $V_{OH} = V_{DD}$, $V_{OL} = 0V$

મેમરી ટ્રીક

“Sharp Switch, Perfect Levels”

પ્રશ્ન 2(b) [4 ગુણ]

Explain noise immunity and noise margin

જવાબ

નોઈઝ ઇમ્યુનિટી circuit ની અનચાહેલા signal variations ને નકારવાની ક્ષમતા માપે છે.

કોષ્ટક 5. Noise Parameters

પેરામીટર	વ્યાખ્યા	ફોર્મ્યુલા
NMH	હાઈ-લેવલ નોઈઝ માર્જિન	$V_{OH} - V_{IH}$
NML	લો-લેવલ નોઈઝ માર્જિન	$V_{IL} - V_{OL}$
નોઈઝ ઇમ્યુનિટી	નોઈઝ નકારવાની ક્ષમતા	$\min(NMH, NML)$

- લોજિક થ્રેશોલ્ડ લેવલ: V_{IH} (input high), V_{IL} (input low)
- આઉટપુટ લેવલ: V_{OH} (output high), V_{OL} (output low)
- વધુ સારી ઇમ્યુનિટી: મોટા નોઈઝ માર્જિન વધુ સારી સુરક્ષા આપે છે
- ડિઝાઇન લક્ષ્ય: મજબૂત operation માટે નોઈઝ માર્જિન વધારવા

મેમરી ટ્રીક

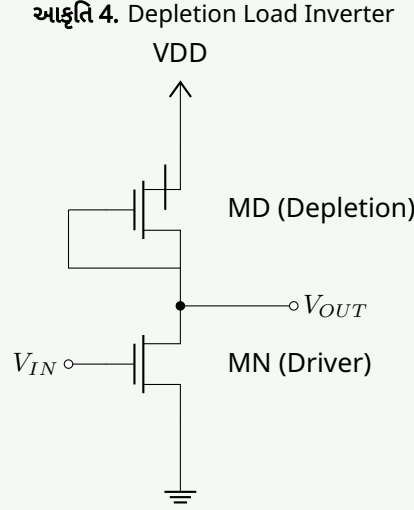
“Margins Protect Against Noise”

પ્રશ્ન 2(c) [7 ગુણ]

Describe inverter circuit with saturated and linear depletion load nMOS inverter

જવાબ

Depletion load nMOS ઇન્વર્ટર active load resistor તરીકે depletion transistor વાપરે છે.
Diagram:



કોષ્ટક 6. Load Operation Modes

લોડ પ્રકાર	ગેટ કનેક્શન	ઓપરેશન
Saturated Load	$V_G = V_D$	હંમેશા saturation માં
Linear Load	$V_G = V_{DD}$	Linear region માં કામ કરી શકે છે

- ડિપ્લીશન ડિવાઇસ: $V_{GS} = 0$ સાથે વહન કરે છે, current source તરીકે કામ કરે છે
- લોડ લાઇન વિશ્લેષણ: operating point intersection નક્કી કરે છે
- પાવર કન્ઝ્યુમ્પશન: હંમેશા વહન કરે છે, ઊંચો static power
- સ્વિચિંગ સ્પીડ: pull-up કરતાં pull-down ઝડપી

મેમરી ટ્રીક

“Depletion Loads Drive Outputs”

OR

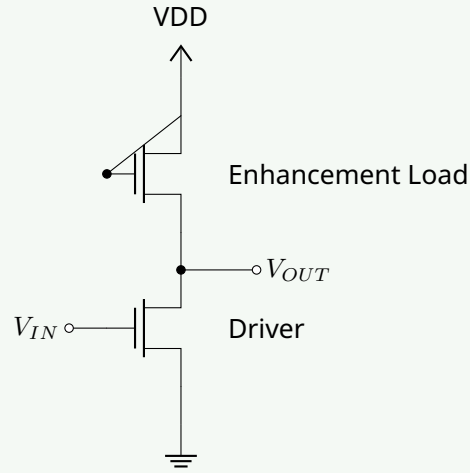
પ્રશ્ન 2(a) [3 ગુણ]

Draw and explain enhancement load inverter

જવાબ

Enhancement load ઇન્વર્ટર ખાસ biasing સાથે enhancement MOSFET ને load તરીકે વાપરે છે.
Diagram:

આકૃતિ 5. Enhancement Load Inverter



- બૂટસ્ટ્રેપ કનેક્શન: લોડ માટે gate ને drain સાથે જોડાયેલ
- મર્યાદિત આઉટપુટ હાઈ: $V_{OUT(max)} = V_{DD} - V_T$
- થ્રેશોલ્ડ નુકસાન: Enhancement load વોલ્ટેજ ડ્રોપ કરાવે છે

મેમરી ટ્રીક

“Enhancement Loses Threshold”

પ્રશ્ન 2(b) [4 ગુણ]

List the advantages of CMOS inverter

જવાબ

CMOS technology NMOS ઇન્વર્ટર કરતાં શ્રેષ્ઠ performance આપે છે.

કોષ્ટક 7. CMOS Advantages

ફાયદો	લાભ
શૂન્ય સ્ટેટિક પાવર	steady state માં કોઈ current path નથી
રેલ-ટુ-રેલ આઉટપુટ	સંપૂર્ણ V_{DD} અને 0V આઉટપુટ લેવલ
ઊંચી નોઈઝ ઇમ્યુનિટી	મોટા નોઈઝ માર્જિન
સમપ્રમાણ સ્વિચિંગ	બરાબર rise અને fall times

- પાવર એફિશિયન્સી: માત્ર switching દરમિયાન dynamic power
- સ્કેલેબિલિટી: બધા technology nodes પર સારી રીતે કામ કરે છે
- ફેન-આઉટ ક્ષમતા: અનેક inputs ડ્રાઇવ કરી શકે છે
- તાપમાન સ્થિરતા: performance તાપમાન પર ઓછી સંવેદનશીલ

મેમરી ટ્રીક

“CMOS Saves Power Perfectly”

પ્રશ્ન 2(c) [7 ગુણ]

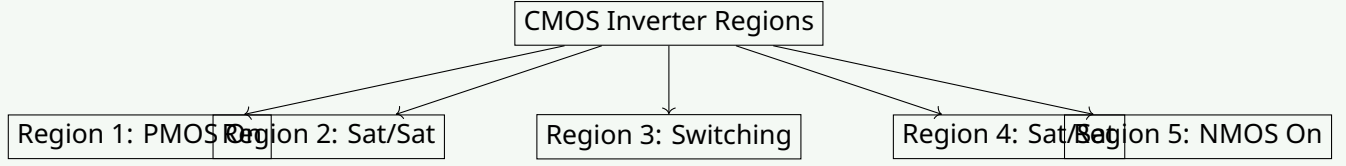
Draw and Explain operating mode of region for CMOS Inverter

જવાબ

CMOS ઇન્વર્ટર operation input voltage ના આધારે પાંચ અલગ અલગ regions સમાવે છે.

Diagram:

આકૃતિ 6. CMOS Operation Regions



કોષ્ટક 8. CMOS Operating Regions

પ્રદેશ	NMOS સ્થિતિ	PMOS સ્થિતિ	આઉટપુટ
1	OFF	Linear	$V_{OH} \approx V_{DD}$
2	Saturation	Saturation	Transition
3	Saturation	Saturation	$V_{DD}/2$
4	Saturation	Saturation	Transition
5	Linear	OFF	$V_{OL} \approx 0V$

- સ્વિચિંગ થ્રેશોલ્ડ: VTC region 3 પર $V_{DD}/2$ ને પાર કરે છે
- કરંટ ફ્લો: માત્ર transition regions 2,3,4 દરમિયાન
- નોઈઝ માર્જિન: Regions 1 અને 5 ઇમ્યુનિટી આપે છે
- ગેઇન: Region 3 માં મહત્તમ (switching point)

પ્રશ્ન 3(a) [3 ગુણ]

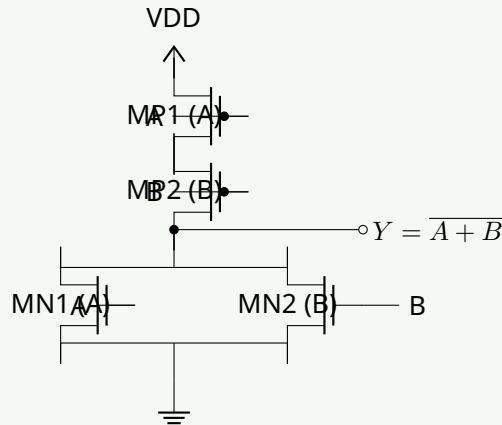
Draw two input NOR gate using CMOS

જવાબ

CMOS NOR ગેટ complementary networks વાપરીને De Morgan's law અમલમાં મૂકે છે.

Diagram:

આકૃતિ 7. CMOS NOR2 Gate



- પુલ-અપ નેટવર્ક: સીરીઝ PMOS transistors (હાઈ આઉટપુટ માટે A અને B બંને લો)
- પુલ-ડાઉન નેટવર્ક: પેરેલલ NMOS transistors (લો આઉટપુટ માટે A અથવા B હાઈ)
- લોજિક ફંક્શન: $Y = (A + B)' = A' \cdot B'$

મેમરી ટ્રીક

"Series PMOS, Parallel NMOS"

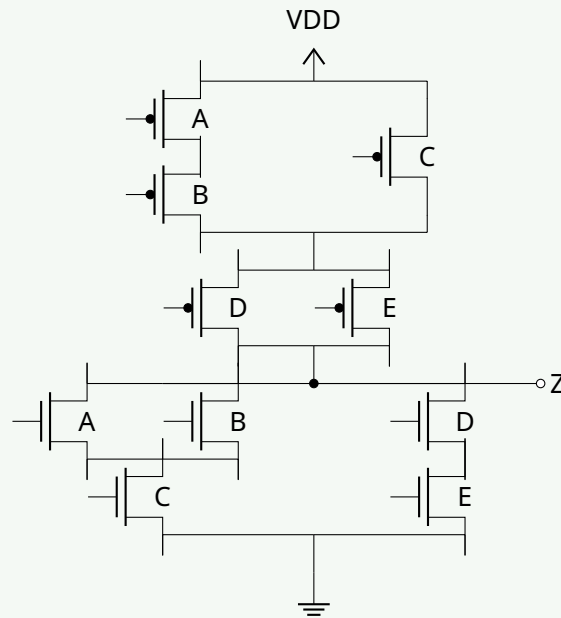
પ્રશ્ન 3(b) [4 ગુણ]

Implement Boolean function $Z = [(A+B)C+DE]'$ using CMOS

જવાબ

જટિલ CMOS લોજિક કાર્યક્ષમ અમલીકરણ માટે AOI (AND-OR-Invert) સ્ટ્રક્ચર વાપરે છે.

Diagram:

આકૃતિ 8. CMOS Implementation of $Z = [(A + B)C + DE]'$ 

- AOI સ્ટ્રક્ચર: કાર્યક્ષમ single-stage અમલીકરણ
- ડ્યુઅલ નેટવર્ક: complementary pull-up અને pull-down
- લોજિક ઓપ્ટિમાઇઝેશન: અલગ ગેટ કરતાં ઓછા transistors

મેમરી ટ્રીક

"AOI Inverts Complex Logic Efficiently"

પ્રશ્ન 3(c) [7 ગુણ]

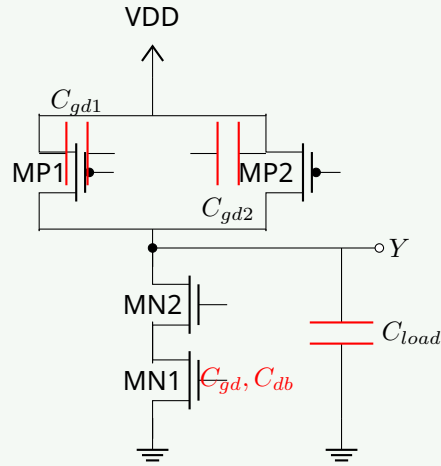
Draw and explain CMOS NAND2 gate with the parasitic device capacitances

જવાબ

CMOS ગેટમાં પેરાસિટિક કેપેસિટન્સ switching speed અને power consumption ને અસર કરે છે.

Diagram:

આકૃતિ 9. CMOS NAND2 with Parasitics



કોષ્ટક 9. Parasitic Capacitances

કેપેસિટન્સ	સ્થાન	અસર
Cgs	Gate-Source	Input capacitance
Cgd	Gate-Drain	Miller effect
Cdb	Drain-Bulk	Output loading
Csb	Source-Bulk	Source loading

- સ્વિચિંગ વિલંબ: પેરાસિટિક કેપેસિટન્સ transitions ધીમા કરે છે
- પાવર કન્ઝ્યુમ્પશન: પેરાસિટિક caps ચાર્જ/ડિસ્ચાર્જ કરવા
- મિલર ઇફેક્ટ: Cgd feedback બનાવે છે, switching ધીમું કરે છે
- લેઆઉટ ઓપ્ટિમાઇઝેશન: પેરાસિટિક કેપેસિટન્સ ઓછા કરવા

મેમરી ટ્રીક

“Parasitics Slow Gates Down”

OR

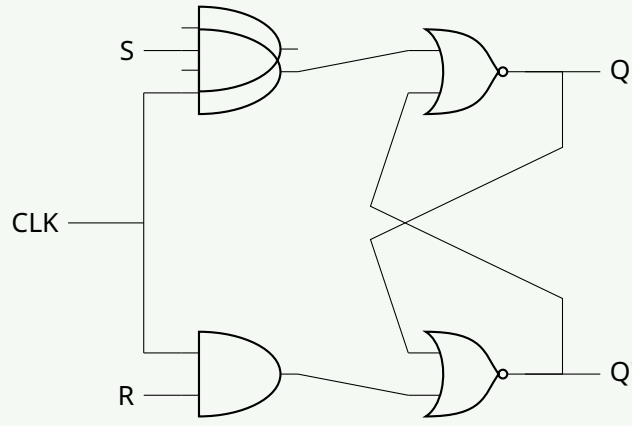
પ્રશ્ન 3(a) [3 ગુણ]

Draw and explain NOR based Clocked SR latch using CMOS

જવાબ

Clocked SR latch synchronous operation માટે clock enable સાથે NOR gates વાપરે છે.
Diagram:

આકૃતિ 10. Clocked SR Latch



આકૃતિ 11. Clocked SR Latch

- કલોક કંટ્રોલ: S અને R માત્ર CLK = 1 હોય ત્યારે જ અસરકારક
- ટ્રાન્સપેરન્ટ મોડ: clock સક્રિય હોય ત્યારે આઉટપુટ input ને અનુસરે છે
- હોલ્ડ મોડ: clock નિષ્ક્રિય હોય ત્યારે આઉટપુટ સ્થિતિ જાળવે છે
- મૂળભૂત બિલ્ડિંગ બ્લોક: flip-flops માટે પાયા

મેમરી ટ્રીક

“Clock Controls Transparent Latching”

પ્રશ્ન 3(b) [4 ગુણ]

Implement Boolean function $Z = [AB + C(D + E)]'$ using CMOS

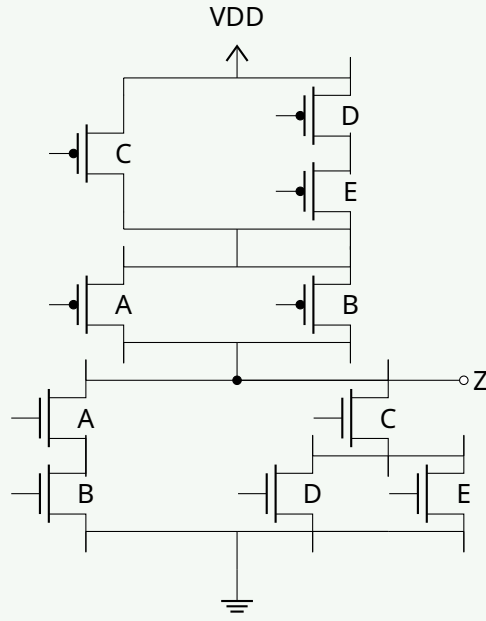
જવાબ

આ ફંક્શન AOI લોજિક સ્ટ્રક્ચર વાપરીને inverted sum-of-products અમલમાં મૂકે છે.

Logic Analysis: $Z = [AB + C(D + E)]' = [AB + CD + CE]'$

Diagram:

આકૃતિ 12. CMOS Implementation of $Z = [AB + C(D + E)]'$



કોષ્ટક 10. Logic Terms

ટર્મ	ઇનપુટ્સ	ફંક્શન
ટર્મ 1	A, B	AB
ટર્મ 2	C, D	CD
ટર્મ 3	C, E	CE
આઉટપુટ	બધા terms	$(AB + CD + CE)'$

- AOI અમલીકરણ: single stage, કાર્યક્ષમ ડિઝાઇન
- ટ્રાન્ઝિસ્ટર કાઉન્ટ: અલગ ગેટ અમલીકરણ કરતાં ઓછા
- પર્ફોર્મન્સ: ઝડપી switching, ઓછો power

મેમરી ટ્રીક

“Three AND Terms Feed One NOR”

પ્રશ્ન 3(c) [7 ગુણ]

Differentiate AOI and OAI logic with suitable example

જવાબ

AOI અને OAI કાર્યક્ષમ CMOS અમલીકરણ માટે પૂરક લોજિક પરિવારો છે.

કોષ્ટક 11. AOI vs OAI

પેરામીટર	AOI (AND-OR-Invert)	OAI (OR-AND-Invert)
લોજિક ફંક્શન	Sum of Products (SOP) inverted	Product of Sums (POS) inverted
સમીકરણ	$Y = (AB + CD)'$	$Y = ((A + B)(C + D))'$
PDN સ્ટ્રક્ચર	Parallel branches of series NMOS	Series branches of parallel NMOS
PUN સ્ટ્રક્ચર	Series branches of parallel PMOS	Parallel branches of series PMOS
ઝડપ	Typically faster for SOP	Typically faster for POS

Example:

- **AOI:** $Y = (AB + CD)'$
 - NMOS: Series A-B in parallel with Series C-D
 - PMOS: Parallel A,B in series with Parallel C,D
- **OAI:** $Y = ((A + B)(C + D))'$
 - PMOS: Parallel A,B in series with Parallel C,D
 - NMOS: Series A-B in parallel with Series C-D
- **ડિઝાઇન પસંદગી:** બૂલિયન ફંક્શન ફોર્મ આધારે પસંદ કરો
- **ઓપ્ટિમાઇઝેશન:** ટ્રાન્ઝિસ્ટર કાઉન્ટ અને વિલંબ ઓછો કરે છે
- **ફેક્તલ:** AOI અને OAI De Morgan duals છે

મેમરી ટ્રીક

“AOI ANDs then ORs, OAI ORs then ANDs”

પ્રશ્ન 4

પ્રશ્ન 4(a) [3 ગુણ]

Define: 1) Regularity 2) Modularity 3) Locality

જવાબ

VLSI જટિલતાને સંચાલિત કરવા અને સફળ અમલીકરણ સુનિશ્ચિત કરવા માટે ડિઝાઇન હાયરાર્કી સિદ્ધાંતો આવશ્યક છે.

કોષ્ટક 12. Design Principles

સિદ્ધાંત	વ્યાખ્યા	ફાયદો
Regularity	સમાન સ્ટ્રક્ચર્સનો વારંવાર ઉપયોગ	સરળ લેઆઉટ, ટેસ્ટિંગ
Modularity	ડિઝાઇનને નાના બ્લોક્સમાં વહેંચવું	સ્વતંત્ર ડિઝાઇન, પુનઃઉપયોગ
Locality	મોટે ભાગે સ્થાનિક interconnections	ઓછી routing જટિલતા

- **ડિઝાઇન કાર્યક્ષમતા:** સિદ્ધાંતો ડિઝાઇન સમય અને પ્રયાસ ઘટાડે છે
- **વેરિફિકેશન:** મોડ્યુલર અભિગમ ટેસ્ટિંગ સરળ બનાવે છે
- **સ્કેલેબિલિટી:** મોટા, વધુ જટિલ ડિઝાઇન્સ શક્ય બનાવે છે

મેમરી ટ્રીક

“Regular Modules Stay Local”

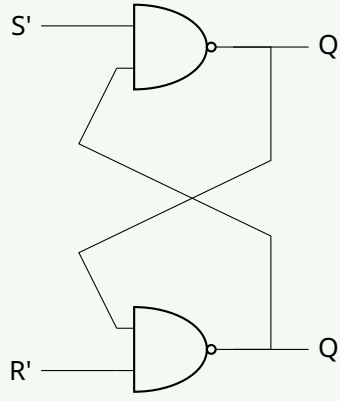
પ્રશ્ન 4(b) [4 ગુણ]

Implement SR latch (NAND gate) using CMOS inverter

જવાબ

NAND ગેટ વાપરીને SR latch active-low inputs સાથે set-reset functionality આપે છે.
Diagram:

આકૃતિ 13. NAND SR Latch



કોષ્ટક 13. NAND SR Latch Truth Table

S'	R'	Q	Q'	સ્થિતિ
0	1	1	0	Set
1	0	0	1	Reset
1	1	Q	Q'	Hold
0	0	1	1	અમાન્ય

- ક્રોસ-કપ્લડ સ્ટ્રક્ચર: મેમરી ફંક્શન આપે છે
- એક્ટિવ-લો ઇનપુટ્સ: $S' = 0$ સેટ કરે છે, $R' = 0$ રીસેટ કરે છે
- પ્રતિબંધિત સ્થિતિ: બંને ઇનપુટ્સ એકસાથે લો

મેમરી ટ્રીક

“Cross-Coupled NANDS Remember State”

પ્રશ્ન 4(c) [7 ગુણ]

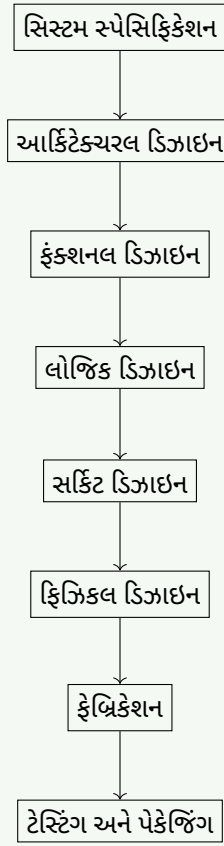
Explain VLSI design flow

જવાબ

VLSI ડિઝાઇન ફ્લો specification થી fabrication સુધીના વ્યવસ્થિત પગલાંઓ અનુસરે છે.

Diagram:

આકૃતિ 14. VLSI Design Flow



કોષ્ટક 14. Design Levels

લેવલ	પ્રવૃત્તિઓ	આઉટપુટ
સિસ્ટમ	આવશ્યકતા વિશ્લેષણ	સ્પેસિફિકેશન્સ
આર્કિટેક્ચર	બ્લોક-લેવલ ડિઝાઇન	સિસ્ટમ આર્કિટેક્ચર
લોજિક	બૂલિયન ઓપ્ટિમાઇઝેશન	ગેટ નેટલિસ્ટ
સર્કિટ	ટ્રાન્ઝિસ્ટર સાઇઝિંગ	સર્કિટ નેટલિસ્ટ
ફિઝિકલ	લેઆઉટ, routing	GDSII ફાઇલ

- ડિઝાઇન વેરિફિકેશન: દરેક લેવલે માન્યતા જરૂરી
- પુનરાવર્તન: ઓપ્ટિમાઇઝેશન માટે ફીડબેક લૂપ્સ
- CAD ટૂલ્સ: જટિલ ડિઝાઇન્સ માટે ઓટોમેશન આવશ્યક
- ટાઇમ-ટુ-માર્કેટ: કાર્યક્ષમ ફલો ડિઝાઇન સાચકલ ઘટાડે છે

મેમરી ટ્રીક

"System Architects Love Circuit Physical Fabrication"

OR

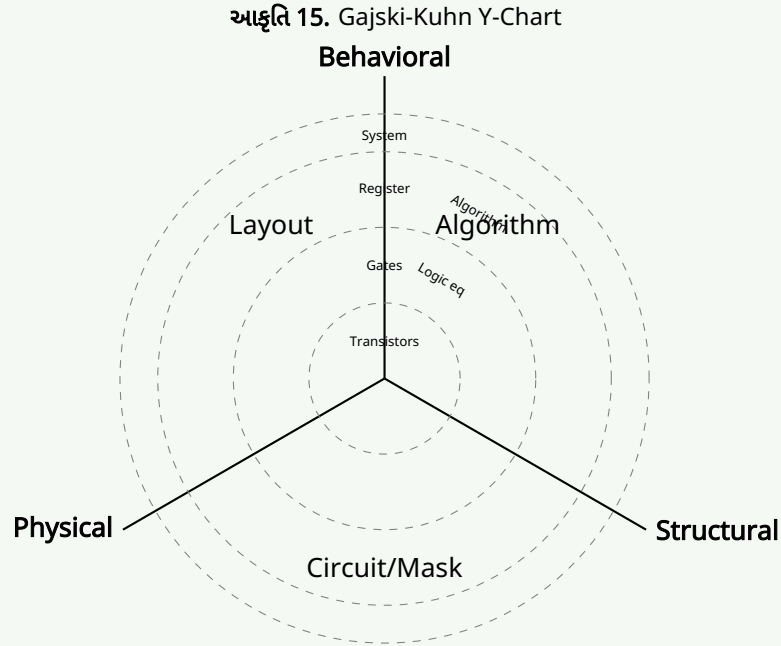
પ્રશ્ન 4(a) [3 ગુણ]

Draw and Explain Y-chart

જવાબ

Y-ચાર્ટ VLSI ડિઝાઇનમાં ત્રણ ડિઝાઇન ડોમેન અને તેમના abstraction levels દર્શાવે છે.

Diagram:



- ત્રણ ડોમેન: Behavioral (ફંક્શન), Structural (components), Physical (geometry)
- એબ્સ્ટ્રેક્શન લેવલ: System → Algorithm → Gate → Circuit → Layout
- ડિઝાઇન પદ્ધતિ: સમાન abstraction level પર ડોમેન વચ્ચે ફરવું

મેમરી ટ્રીક

“Behavior, Structure, Physics at All Levels”

પ્રશ્ન 4(b) [4 ગુણ]

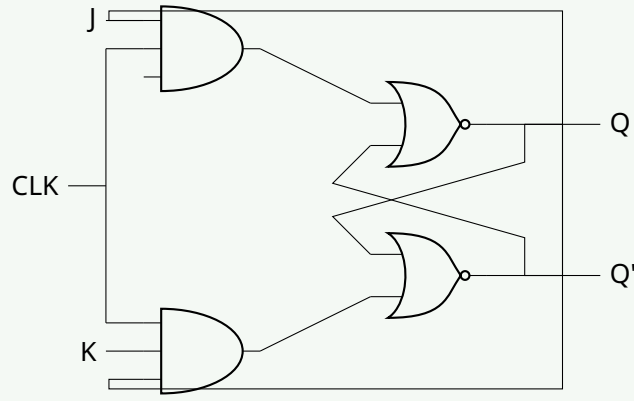
Implement clocked JK latch (NOR gate) using CMOS inverter

જવાબ

JK latch toggle ક્ષમતા સાથે SR latch ની પ્રતિબંધિત સ્થિતિ દૂર કરે છે.

Diagram:

આકૃતિ 16. Clocked JK Latch



કોષ્ટક 15. JK Latch Truth Table

J	K	Q(next)	ઓપરેશન
0	0	Q	Hold
0	1	0	Reset
1	0	1	Set
1	1	Q'	Toggle

- ટોગલ મોડ: $J=K=1$ આઉટપુટ સ્થિતિ ફ્લિપ કરે છે
- ક્લોક એનેબલ: માત્ર $CLK=1$ હોય ત્યારે જ સક્રિય
- ફીડબેક: ઇનપુટ્સ સક્ષમ કરવા માટે વર્તમાન આઉટપુટ વાપરે છે

મેમરી ટ્રીક

"JK Toggles, No Forbidden State"

પ્રશ્ન 4(c) [7 ગુણ]

Explain: 1) Lithography 2) Etching 3) Deposition 4) Oxidation 5) Ion Implantation 6) Diffusion

જવાબ

Integrated circuits બનાવવા માટે આવશ્યક semiconductor fabrication processes.

કોષ્ટક 16. Fabrication Processes

પ્રક્રિયા	હેતુ	પદ્ધતિ
લિથોગ્રાફી	પેટર્ન ટ્રાન્સફર	માસ્ક દ્વારા UV exposure
એચિંગ	મેટેરિયલ રિમૂવલ	ભીના/સૂકા રાસાયણિક પ્રક્રિયાઓ
જમાવટ	લેયર ઉમેરો	CVD, PVD, sputtering
ઓક્સિડેશન	ઇન્સ્યુલેટર વૃદ્ધિ	થર્મલ/પ્લાઝ્મા ઓક્સિડેશન
આયન પ્રત્યારોપણ	ડોપિંગ પરિચય	ઉચ્ચ-ઊર્જા આયન bombardment
પ્રસરણ	ડોપન્ટ વિતરણ	ઉચ્ચ તાપમાને ફેલાવો

- પેટર્ન વ્યાખ્યા: લિથોગ્રાફી ડિવાઇસ લક્ષણો બનાવે છે
- પસંદગીયુક્ત રિમૂવલ: એચિંગ અનચાહેલ મેટેરિયલ દૂર કરે છે
- લેયર બિલ્ડિંગ: જમાવટ જરૂરી મેટેરિયલ ઉમેરે છે
- ડોપિંગ કંટ્રોલ: પ્રત્યારોપણ અને પ્રસરણ junctions બનાવે છે
- ગુણવત્તા નિયંત્રણ: દરેક પગલું અંતિમ ડિવાઇસ પરફોર્મન્સને અસર કરે છે

મેમરી ટ્રીક

“Light Etches Deposited Oxides, Ions Diffuse”

પ્રશ્ન 5(a) [3 ગુણ]

Implement 2 input XNOR gate using Verilog

જવાબ

XNOR ગેટ ઇનપુટ્સ સમાન હોય ત્યારે ઊંચો (High) આઉટપુટ આપે છે.

Listing 1. Verilog Code for XNOR Gate

```
1 module xnor_gate(
2     input a, b,
3     output y
4 );
5     assign y = ~(a ^ b);
6 endmodule
```

- **લોજિક ફંક્શન:** $Y = (A \oplus B)' = A'B' + AB$
- **હાર્ડ આઉટપુટ:** જ્યારે બંને ઇનપુટ્સ 0 હોય કે બંને 1 હોય
- **ઉપયોગ:** સમાનતા તુલનાકાર (Equality comparator), પેરિટી ચેકર

મેમરી ટ્રીક

“XNOR Equals Equal Inputs”

પ્રશ્ન 5(b) [4 ગુણ]

Implement Encoder (8:3) using CASE statement in Verilog

જવાબ

પ્રાયોરિટી એનકોડર 8-બિટ ઇનપુટને 3-બિટ બાઇનરી આઉટપુટમાં રૂપાંતરિત કરે છે, જે માટે behavioral modeling માં case statement વપરાય છે.

Listing 2. 8:3 Priority Encoder using CASE

```
1 module encoder_8to3(
2     input [7:0] in,
3     output reg [2:0] out
4 );
5     always @(*) begin
6         case(in)
7             8'b00000001: out = 3'b000;
8             8'b00000010: out = 3'b001;
9             8'b00000100: out = 3'b010;
10            8'b00001000: out = 3'b011;
11            8'b00010000: out = 3'b100;
12            8'b00100000: out = 3'b101;
13            8'b01000000: out = 3'b110;
14            8'b10000000: out = 3'b111;
15            default: out = 3'b000;
16        endcase
17    end
```

18 | `endmodule`

- બિહેવિયરલ મોડેલિંગ: CASE સ્ટેટમેન્ટ ફંક્શનને વર્ણવે છે
- કોમ્બિનેશનલ લોજિક: `always @(*)` બ્લોક વપરાય છે
- ડિફોલ્ટ કેસ: અમાન્ય સ્થિતિઓ (દા.ત. બધા શૂન્ય) સંભાળે છે

મેમરી ટ્રીક

"One Hot Input, Binary Output"

પ્રશ્ન 5(c) [7 ગુણ]

Explain case statement in Verilog with suitable examples

જવાબ

CASE સ્ટેટમેન્ટ એક્સપ્રેશન કિંમત પર આધારિત મલ્ટિ-વે બ્રાન્ચિંગ પૂરું પાડે છે, જે સામાન્ય રીતે કોમ્બિનેશનલ લોજિક (ડિકોડર, મલ્ટિપ્લેક્સર) અને સ્ટેટ મશીન્સમાં વપરાય છે.

Syntax:

```
1 case (expression)
2   value1: statement1;
3   value2: statement2;
4   default: default_statement;
5 endcase
```

Example 1: 4:1 Multiplexer

```
1 module mux_4to1(
2   input [1:0] sel,
3   input [3:0] in,
4   output reg out
5 );
6 always @(*) begin
7   case(sel)
8     2'b00: out = in[0];
9     2'b01: out = in[1];
10    2'b10: out = in[2];
11    2'b11: out = in[3];
12  endcase
13 end
14 endmodule
```

Advantages:

- વાંચનક્ષમતા: અનેક if-else સ્ટેટમેન્ટ કરતા વધુ સ્પષ્ટ
- સિન્થેસિસ: મલ્ટિપ્લેક્સર કે ROM માં કાર્યક્ષમ રીતે મેપ થાય છે
- પેરેલલ ઇવેલ્યુએશન: હાર્ડવેર બધી શરતો એકસાથે તપાસે છે

કોષ્ટક 17. CASE Variants

વેરિઅન્ટ	સિન્ટેક્સ	ઉપયોગ
case	<code>case(expr)</code>	ચોક્કસ બીટ મેચિંગ (0, 1, x, z)
casex	<code>casex(expr)</code>	'x' અને 'z' ને don't care ગણે છે
casez	<code>casez(expr)</code>	'z' ને don't care ગણે છે

મેમરી ટ્રીક

“CASE Chooses Actions Systematically Everywhere”