

VLSI (4361102) - Winter 2024 Solution - Gujarati

Milav Dabgar

November 21, 2024

પ્રશ્ન 1 [a ગુણ]

3 High K FINFET ના ફાયદા લખો.

જવાબ

કોષ્ટક 1. High K FINFET ના ફાયદા

ફાયદો	વર્ણન
ઓછો leakage current	સારું ગેટ કંટ્રોલ પાવર consumption ઘટાડે છે
સુધારેલી performance	વધુ ડ્રાઇવ કરંટ અને ઝડપી switching
વધુ સારી scalability	Moore's law scaling ચાલુ રાખવાની મંજૂરી આપે છે

- High K dielectric: ગેટ leakage નોંધપાત્ર રીતે ઘટાડે છે
- 3D structure: ચેનલ પર વધુ સારું electrostatic control
- ઓછી પાવર: static અને dynamic power બંને ઘટાડે છે

મેમરી ટ્રીક

High Performance, Low Power, Better Control

પ્રશ્ન 1 [b ગુણ]

4 વ્યાખ્યા કરો: (1) pinch off point (2) Threshold Voltage.

જવાબ

કોષ્ટક 2. Key MOSFET Parameters

શબ્દ	વ્યાખ્યા	મહત્વ
Pinch-off Point	ચેનલ સંપૂર્ણ deplete થતું સ્થાન	Saturation region માં પ્રવેશ દર્શાવે છે
Threshold Voltage	Conducting channel બનાવવા માટે લઘુત્તમ V_{GS}	ON/OFF switching point નિર્ધારિ છે

- Pinch-off point: $V_{DS} = V_{GS} - V_T$, ચેનલ શૂન્ય પહોળાઈ સુધી સંકુચિત થાય છે
- Threshold voltage: Enhancement MOSFET માટે સામાન્ય રીતે 0.7V
- મહત્વપૂર્ણ parameters: બંને MOSFET operating regions નિર્ધારિ છે

મેમરી ટ્રીક

Threshold Turns ON, Pinch-off Points to Saturation

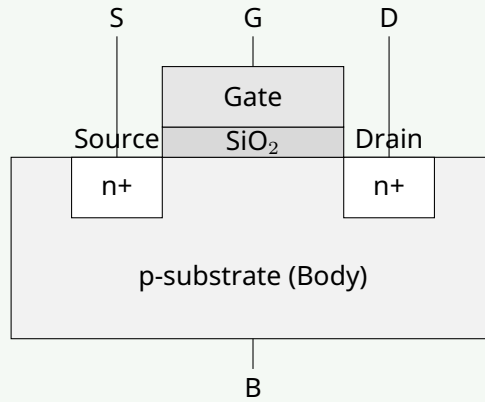
પ્રશ્ન 1 [c ગુણ]

7 MOSFET transistor નું બંધારણ દોરો અને સમજાવો.

જવાબ

કોષ્ટક 3. Structure Components

ઘટક	સામગ્રી	કાર્ય
Gate	Polysilicon/Metal	ચેનલ formation કંટ્રોલ કરે છે
Gate oxide	SiO ₂	Gate ને substrate થી અલગ કરે છે
Source/Drain	n+ doped silicon	Current ના પ્રવેશ/બહાર નીકળવાના સ્થળો
Substrate	p-type silicon	Body connection પૂરું પાડે છે



- ચેનલ formation: Oxide-semiconductor interface પર થાય છે
- Enhancement mode: $V_{GS} > V_T$ હોય ત્યારે ચેનલ બને છે
- ચાર-terminal device: Gate, Source, Drain, Body connections

મેમરી ટ્રીક

Gate Controls, Oxide Isolates, Source-Drain Conducts

પ્રશ્ન 1 [c ગુણ]

7 Full Voltage Scaling અને Constant Voltage Scaling ની સરખામણી કરો.

જવાબ

કોષ્ટક 4. Full Voltage Scaling vs Constant Voltage Scaling

Parameter	Full Voltage Scaling	Constant Voltage Scaling
Supply voltage	\propto વડે scale down	સ્થિર રહે છે
Gate oxide thickness	\propto વડે scale down	\propto વડે scale down
Channel length	\propto વડે scale down	\propto વડે scale down
Power density	સ્થિર રહે છે	\propto^2 વડે વધે છે
Performance	મધ્યમ સુધારો	વધુ સારી performance
Reliability	વધુ સારી	High fields ને કારણે નબળી

- Full scaling: બધા dimensions અને voltages પ્રમાણસર scale કરાય છે
- Constant voltage: ફક્ત physical dimensions scale કરાય છે, voltage અપરિવર્તિત

- Trade-off: Performance vs power vs reliability

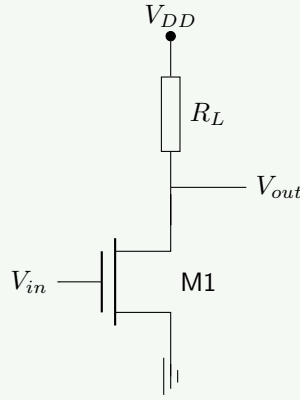
મેમરી ટ્રીક

Full scales All, Constant keeps Voltage

પ્રશ્ન 2 [a ગુણ]

3 રેસિસ્ટિવ લોડ ઇનવર્ટર દોરો. જુદા જુદા ઓપરેશન રીજન માટે ઇનપુટ વોલ્ટેજની રેન્જ લખો.

જવાબ



કોષ્ટક 5. Operating Regions

રીજન	ઇનપુટ વોલ્ટેજ રેન્જ	આઉટપુટ સ્થિતિ
Cut-off	$V_{in} < V_T$	$V_{out} = V_{DD}$
Triode	$V_T < V_{in} < V_{DD} - V_T$	ટ્રાન્ઝિશન
Saturation	$V_{in} > V_{DD} - V_T$	$V_{out} \approx 0V$

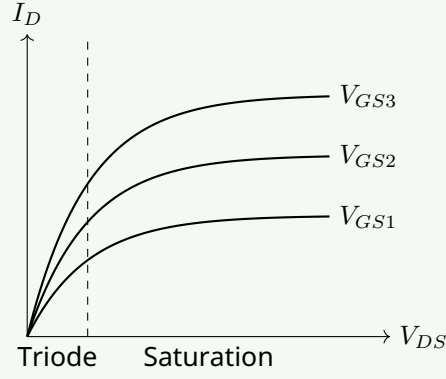
મેમરી ટ્રીક

Cut-off High, Triode Transition, Saturation Low

પ્રશ્ન 2 [b ગુણ]

4 N channel MOSFET ની VDS-ID અને VGS-ID લાક્ષણિકતાઓ દોરો અને સમજાવો.

જવાબ



કોષ્ટક 6. NMOS Characteristics

લાક્ષણિકતા	રીજન	વર્તન
$V_{DS} - I_D$	Triode	V_{DS} સાથે Linear વૃદ્ધિ
$V_{DS} - I_D$	Saturation	સ્થિર I_D (square law)
$V_{GS} - I_D$	Sub-threshold	Exponential વૃદ્ધિ
$V_{GS} - I_D$	V_T ઉપર	Square law relationship

- Triode region: I_D વડે V_{DS} સાથે linearly વધે છે
- Saturation: I_D V_{DS} થી સ્વતંત્ર, V_{GS} પર આધારિત
- Square law: Saturation માં $I_D \propto (V_{GS} - V_T)^2$

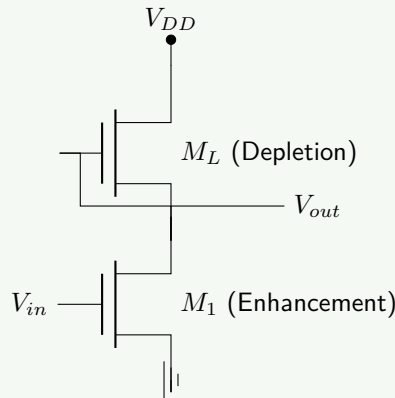
મેમરી ટ્રીક

Linear in Triode, Square in Saturation

પ્રશ્ન 2 [c ગુણ]

7 ડિપ્લેશન લોડ NMOS ઇનવર્ટર સર્કિટ દોરો અને તેની કાર્યપદ્ધતિ સમજાવો.

જવાબ



કોષ્ટક 7. Depletion Load Inverter Operation

ઇનપુટ	M1 સ્થિતિ	ML સ્થિતિ	આઉટપુટ
Low (0V)	Cut-off	Active load	High (V_{DD})
High (V_{DD})	Saturated	Linear	Low

- Depletion load: હંમેશા conducting, current source તરીકે કાર્ય કરે છે
- વધુ સારી performance: Resistive load કરતાં higher output voltage swing
- Gate connection: Depletion operation માટે ML નું gate source સાથે જોડાયેલું
- સુધારેલું noise margin: Enhancement load કરતાં વધુ સારું V_{OH}

મેમરી ટ્રીક

Depletion Always ON, Enhancement Controls Flow

પ્રશ્ન 2 [a ગુણ]

3 CMOS ઇનવર્ટર ના ફાયદા વર્ણવો.

જવાબ

કોષ્ટક 8. Advantages of CMOS Inverter

ફાયદો	લાભ
શૂન્ય static power	Steady state માં કોઈ current નહીં
સંપૂર્ણ voltage swing	આઉટપુટ 0V થી V_{DD} સુધી swing કરે છે
ઉચ્ચ noise margins	વધુ સારી noise immunity

- Complementary operation: એક transistor હંમેશા OFF
- ઉચ્ચ input impedance: Gate isolation ઉચ્ચ impedance પૂરું પાડે છે
- ઝડપી switching: ઓછા parasitic capacitances

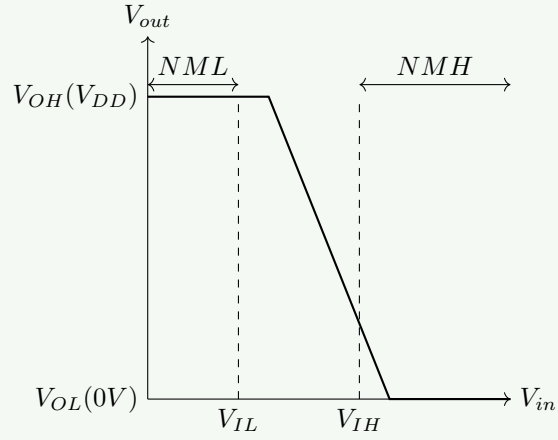
મેમરી ટ્રીક

Zero Power, Full Swing, High Immunity

પ્રશ્ન 2 [b ગુણ]

4 નોઇસ માર્જિન વિગતવાર દોરો અને સમજાવો.

જવાબ



કોષ્ટક 9. Noise Margin Parameters

Parameter	Formula	સામાન્ય મૂલ્ય
N_{MH}	$V_{OH} - V_{IH}$	V_{DD} ની 40%
N_{ML}	$V_{IL} - V_{OL}$	V_{DD} ની 40%

- High noise margin: Positive noise સામે immunity
- Low noise margin: Negative noise સામે immunity
- વધુ સારા CMOS: અન્ય logic families કરતાં higher noise margins

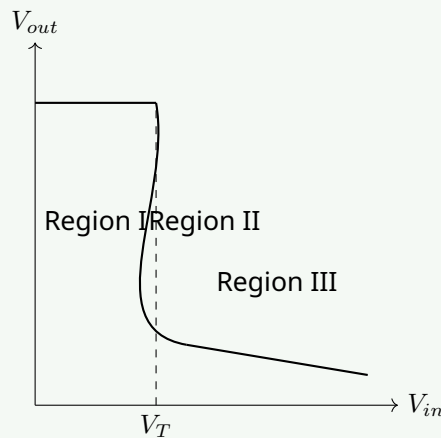
મેમરી ટ્રીક

High goes Higher, Low goes Lower

પ્રશ્ન 2 [c ગુણ]

7 N MOS ઇનવર્ટર ની VTC દોરો અને સમજાવો.

જવાબ



કોષ્ટક 10. NMOS Inverter Operating Regions

રીજન	V_{in} રેન્જ	M1 સ્થિતિ	V_{out}
I	0 થી V_T	Cut-off	V_{DD}
II	V_T થી $V_T + V_{TL}$	Saturation	ઘટતું
III	$V_T + V_{TL}$ થી V_{DD}	Triode	નીચું

- Region I: M1 OFF, કોઈ current flow નહીં, $V_{out} = V_{DD}$
- Region II: M1 saturation માં, તીવ્ર transition
- Region III: M1 triode માં, ધીમેથી ઘટાડો
- Load line: Operating point intersection નિર્ધારિ છે

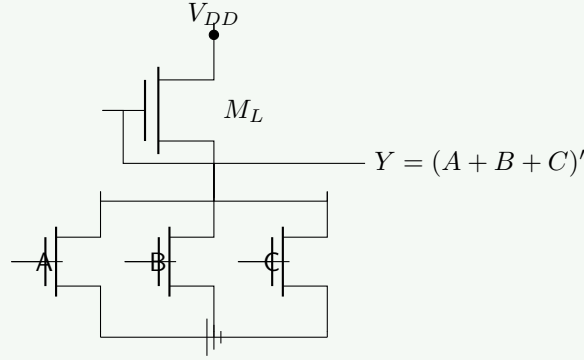
મેમરી ટ્રીક

Cut-off High, Saturation Sharp, Triode Low

પ્રશ્ન 3 [a ગુણ]

3 Generalized મલ્ટીપલ ઇનપુટ NOR gate નું બાંધકામ ડિપ્લેશન NMOS લોડ સાથે દોરો અને સમજાવો.

જવાબ



કોષ્ટક 11. Truth Table

ઇનપુટ્સ	કોઈ ઇનપુટ High?	આઉટપુટ Y
બધા Low	ના	High (1)
કોઈ High	હા	Low (0)

- Parallel NMOS: કોઈપણ input HIGH હોય તો output LOW થાય છે
- NOR operation: $Y = (A + B + C)'$
- Depletion load: Pull-up current પૂરું પાડે છે

મેમરી ટ્રીક

Parallel Pulls Down, Depletion Pulls Up

પ્રશ્ન 3 [b ગુણ]

4 AOI અને OAI ના તફાવત લખો.

જવાબ

કોષ્ટક 12. AOI vs OAI Logic

Parameter	AOI (AND-OR-Invert)	OAI (OR-AND-Invert)
Logic function	$Y = (AB + CD)'$	$Y = ((A + B)(C + D))'$
NMOS structure	Series-parallel	Parallel-series
PMOS structure	Parallel-series	Series-parallel
જટિલતા	મધ્યમ	મધ્યમ

- AOI: AND terms ORed પછી inverted
- OAI: OR terms ANDed પછી inverted
- CMOS implementation: Dual network structure
- Applications: Single stage માં complex logic functions

મેમરી ટ્રીક

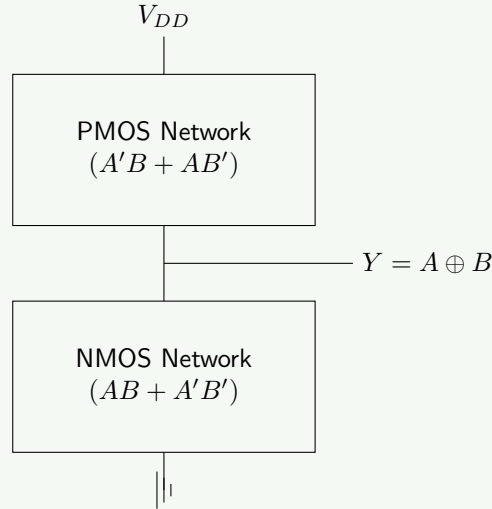
AOI: AND-OR-Invert, OAI: OR-AND-Invert

પ્રશ્ન 3 [c ગુણ]

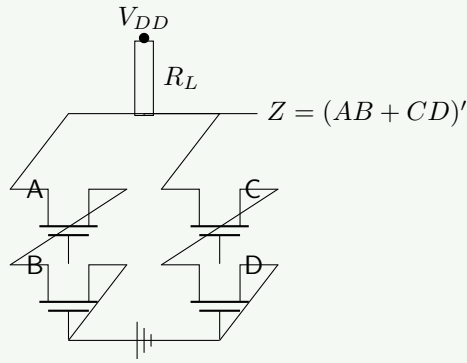
7 EX-OR gate CMOS ની મદદથી અને લોજિક ફંક્શન $Z = (AB + CD)'$ NMOS લોડની મદદથી અમલમાં મૂકો.

જવાબ

EX-OR CMOS Implementation:



$Z = (AB + CD)'$ NMOS Implementation:



કોષ્ટક 13. Logic Implementation Comparison

સર્કિટ	ફંક્શન	Implementation
EX-OR	$A \oplus B$	Complementary CMOS
AOI	$(AB + CD)'$	Series-parallel NMOS

- EX-OR: Efficient implementation માટે transmission gates જરૂરી
- AOI function: Natural NMOS implementation
- Power consideration: CMOS માં zero static power

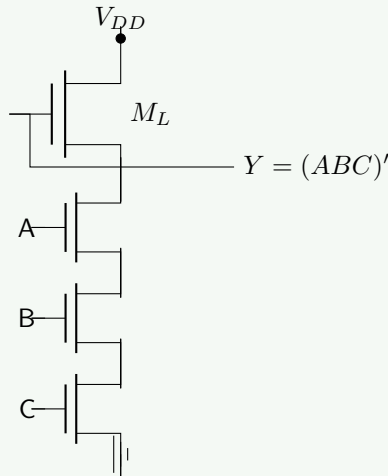
મેમરી ટ્રીક

EX-OR needs Transmission, AOI uses Series-Parallel

પ્રશ્ન 3 [a ગુણ]

3 Generalized મલ્ટીપલ ઇનપુટ NAND gate નું બાંધકામ ડિઝાઇન NMOS લોડ સાથે દોરો અને સમજાવો.

જવાબ



કોષ્ટક 14. NAND Gate Operation

સ્થિતિ	Ground તરફ પાથ	આઉટપુટ Y
બધા inputs HIGH	સંપૂર્ણ પાથ	Low (0)
કોઈ input LOW	તૂટેલો પાથ	High (1)

- Series NMOS: બધા inputs HIGH હોવા જરૂરી output LOW કરવા માટે
- NAND operation: $Y = (ABC)'$
- Depletion load: હંમેશા pull-up current પૂરું પાડે છે

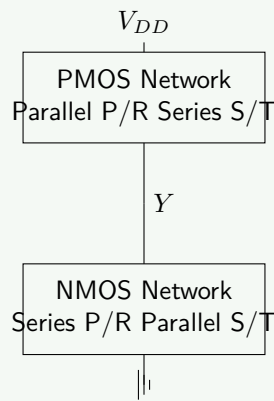
મેમરી ટ્રીક

Series Needs All, NAND Says Not-AND

પ્રશ્ન ૩ [b ગુણ]

4 $((P+R)(S+T))'$ લોજિક ફંક્શન CMOS લોજિકની મદદથી અમલીકરણ કરો.

જવાબ



કોષ્ટક 15. Truth Table Implementation

PMOS Network	NMOS Network	ઓપરેશન
$(P + R)' + (S + T)'$	$(P + R)(S + T)$	Complementary
$P' R' + S' T'$	$PS + PT + RS + RT$	De Morgan's law

- PMOS: Groups વિધિન parallel, groups વચ્ચે series
- NMOS: Groups વિધિન series, groups વચ્ચે parallel
- Dual network: Complementary operation સુનિશ્ચિત કરે છે

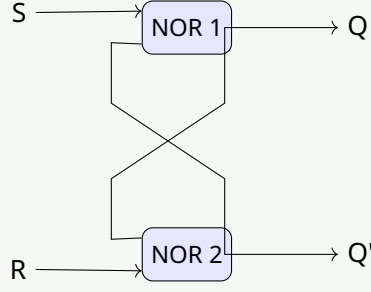
મેમરી ટ્રીક

PMOS does Opposite of NMOS

પ્રશ્ન ૩ [c ગુણ]

7 SR latch circuit ની કાર્યપદ્ધતિ વર્ણવો.

જવાબ



કોષ્ટક 16. SR Latch Truth Table

S	R	Q(n+1)	Q'(n+1)	સ્થિતિ
0	0	Q(n)	Q'(n)	Hold
0	1	0	1	Reset
1	0	1	0	Set
1	1	0	0	અમાન્ય

- Set operation: $S=1, R=0$ થી $Q=1$ થાય છે
- Reset operation: $S=0, R=1$ થી $Q=0$ થાય છે
- Hold state: $S=0, R=0$ પહેલાની state જાળવે છે
- અમાન્ય state: $S=1, R=1$ ટાળવી જોઈએ
- Cross-coupled: એક gate નું output બીજાના input માં જાય છે

મેમરી ટ્રીક

Set Sets, Reset Resets, Both Bad

પ્રશ્ન 4 [a ગુણ]

3 ચિપ ફેબ્રિકેશન માં Etching methods ની સરખામણી કરો.

જવાબ

કોષ્ટક 17. Etching Methods Comparison

પદ્ધતિ	પ્રકાર	ફાયદા	નુકસાન
Wet Etching	રાસાયણિક	ઉચ્ચ selectivity, સરળ	Isotropic, undercut
Dry Etching	ભૌતિક/રાસાયણિક	Anisotropic, ચોક્કસ	જટિલ સાધનો
Plasma Etching	Ion bombardment	Directional control	સપાટીને નુકસાન

- Wet etching: પ્રવાહી રસાયણો વાપરે છે, બધી દિશાઓમાં હુમલો
- Dry etching: ગેસ/plasma વાપરે છે, વધુ સારું directional control
- Selectivity: એક સામગ્રીને બીજા કરતાં etch કરવાની ક્ષમતા

મેમરી ટ્રીક

Wet is Wide, Dry is Directional

પ્રશ્ન 4 [b ગુણ]

4 ટૂંક નોંધ લખો : Lithography

જવાબ

કોષ્ટક 18. Lithography Process Steps

સ્ટેપ	પ્રક્રિયા	હેતુ
Resist coating	Photoresist નું spin-on	પ્રકાશ-સંવેદનશીલ layer
Exposure	Mask દ્વારા UV light	Pattern transfer
Development	Exposed resist દૂર કરવું	Pattern પ્રગટ કરવું
Etching	અસુરક્ષિત material દૂર કરવું	Features બનાવવા

- Pattern transfer: Mask થી silicon wafer પર
- Resolution: Minimum feature size નિર્ધારિ છે
- Alignment: Multiple layer processing માટે મહત્વપૂર્ણ
- UV wavelength: ટૂંકી wavelength વધુ સારું resolution આપે છે

મેમરી ટ્રીક

Coat, Expose, Develop, Etch

પ્રશ્ન 4 [c ગુણ]

7 Regularity, Modularity and Locality સમજાવો.

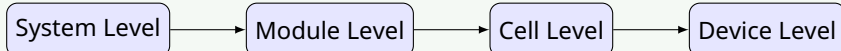
જવાબ

કોષ્ટક 19. Design Principles

સિદ્ધાંત	વ્યાખ્યા	ફાયદા	ઉદાહરણ
Regularity	સમાન structures નું પુનરાવર્તન	સરળ design, testing	Memory arrays
Modularity	Hierarchical design blocks	Reusability, maintainability	Standard cells
Locality	સંબંધિત functions નું જૂથ	ઓછું interconnect	Functional blocks

Implementation વિગતો:

- Regularity: સમાન cell બારંબાર વાપરવાથી design complexity ઘટે છે
- Modularity: Well-defined interfaces સાથે top-down design
- Locality: Wire delays અને routing congestion ઘટાડે છે
- Design benefits: ઝડપી design cycle, વધુ સારી testability
- Manufacturing: Regular patterns દ્વારા સુધારેલી yield



મેમરી ટ્રીક

Regular Modules with Local Connections

પ્રશ્ન 4 [a ગુણ]

3 Design Hierarchy વ્યાખ્યાયિત કરો.

જવાબ

કોષ્ટક 20. Design Hierarchy Levels

સ્તર	વિવરણ	ઘટકો
System	સંપૂર્ણ chip functionality	Processors, memories
Module	મુખ્ય functional blocks	ALU, cache, I/O
Cell	મૂળભૂત logic elements	Gates, flip-flops

- Top-down approach: System નાના modules માં વિભાજિત
- Abstraction levels: દરેક level નીચેની details છુપાવે છે
- Interface definition: Levels વચ્ચે સ્પષ્ટ boundaries

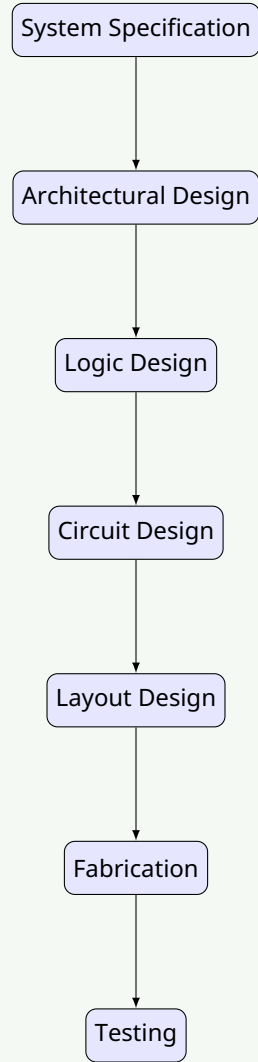
મેમરી ટ્રીક

System to Module to Cell

પ્રશ્ન 4 [b ગુણ]

4 VLSI design flow chart દોરો અને સમજાવો.

જવાબ



કોષ્ટક 21. Design Flow

તબક્કો	ઇનપુટ	આઉટપુટ	સાધનો
Architecture	Specifications	Block diagram	High-level modeling
Logic	Architecture	Gate netlist	HDL synthesis
Circuit	Netlist	Transistor sizing	SPICE simulation
Layout	Circuit	Mask data	Place & route

મેમરી ટ્રીક

Specify, Architect, Logic, Circuit, Layout, Fabricate, Test

પ્રશ્ન 4 [c ગુણ]

7 ટૂંક નોંધ લખો : 'VLSI Fabrication Process'

જવાબ

કોષ્ટક 22. Major Fabrication Steps

પ્રક્રિયા	હેતુ	પરિણામ
Oxidation	SiO ₂ layer વૃદ્ધિ	Gate oxide formation
Lithography	Pattern transfer	Device areas વ્યાખ્યા
Etching	અનાવશ્યક material દૂર કરવું	Device structures બનાવવા
Ion Implantation	Dopants ઉમેરવા	P/N regions બનાવવા
Deposition	Material layers ઉમેરવા	Metal interconnects
Diffusion	Dopants ફેલાવવા	Junction formation

Process Flow:

- Wafer preparation: સ્વચ્છ silicon substrate
- Device formation: બિનેક steps દ્વારા transistors બનાવવા
- Interconnect: Connections માટે metal layers ઉમેરવા
- Passivation: પૂર્ણ થયેલા circuit ની સુરક્ષા
- Testing: Packaging પહેલાં functionality verify કરવી

Clean Room જરૂરિયાતો:

- Class 1-10: અત્યંત સ્વચ્છ વાતાવરણ જરૂરી
- Temperature control: ચોક્કસ process control
- Chemical purity: ઉચ્ચ-ગ્રેડ materials જરૂરી

મેમરી ટ્રીક

Oxidize, Pattern, Etch, Implant, Deposit, Diffuse

પ્રશ્ન 5 [a ગુણ]

3 વેરીલોગ પ્રોગ્રામિંગની જુદી જુદી પદ્ધતિ સરખાવો.

જવાબ

કોષ્ટક 23. Verilog Modeling Styles

શૈલી	વિવરણ	ઉપયોગ
Behavioral	Algorithm description	High-level modeling
Dataflow	Boolean expressions	Combinational logic
Structural	Gate-level description	Hardware representation

- Behavioral: Always blocks, if-else, case statements વાપરે છે
- Dataflow: Boolean operators સાથે assign statements વાપરે છે
- Structural: Gates અને modules explicitly instantiate કરે છે

મેમરી ટ્રીક

Behavior Describes, Dataflow Assigns, Structure Connects

પ્રશ્ન 5 [b ગુણ]

4 બિહેવિયરલ પદ્ધતિ થી NAND gate નો વેરીલોગ પ્રોગ્રામ લખો.

જવાબ

```

1 module nand_gate_behavioral(
2     input wire a, b,
3     output reg y
4 );
5
6 always @(a or b) begin
7     if (a == 1'b1 && b == 1'b1)
8         y = 1'b0;
9     else
10        y = 1'b1;
11 end
12
13 endmodule

```

કોડ સમજૂતી:

- Always block: Inputs બદલાય ત્યારે execute થાય છે
- Sensitivity list: બધા input signals સમાવે છે
- Conditional statement: NAND logic implement કરે છે
- Reg declaration: Procedural assignment માટે જરૂરી

મેમરી ટ્રીક

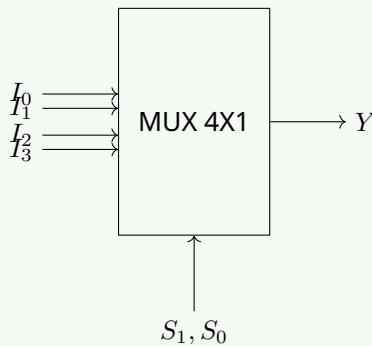
Always watch, IF both high THEN low ELSE high

પ્રશ્ન 5 [c ગુણ]

7 4X1 multiplexer ની સર્કિટ દોરો. Case સ્ટેટમેન્ટ થી આ સર્કિટ નો વેરીલોગ પ્રોગ્રામ બનાવો.

જવાબ

4X1 Multiplexer સર્કિટ:



Verilog કોડ:

```

1 module mux_4x1_case(
2     input wire [1:0] sel,
3     input wire i0, i1, i2, i3,
4     output reg y
5 );
6
7 always @(*) begin
8     case (sel)
9         2'b00: y = i0;
10        2'b01: y = i1;

```



```

11     2'b10: y = i2;
12     2'b11: y = i3;
13     default: y = 1'bx;
14 endcase
15 end
16
17 endmodule

```

કોષ્ટક 24. Truth Table

S1	S0	આઉટપુટ Y
0	0	I0
0	1	I1
1	0	I2
1	1	I3

મેમરી ટ્રીક

Case Selects, Default Protects

પ્રશ્ન 5 [a ગુણ]

3 ઉદાહરણ સાથે Testbench વ્યાખ્યાયિત કરો.

જવાબ

Testbench વ્યાખ્યા: Testbench એ Verilog module છે જે design under test (DUT) ને stimulus પૂરાં પાડે છે અને તેના response ને monitor કરે છે.

ઉદાહરણ Testbench:

```

1 module test_and_gate;
2     reg a, b;
3     wire y;
4
5     and_gate dut(.a(a), .b(b), .y(y));
6
7     initial begin
8         a = 0; b = 0; #10;
9         a = 0; b = 1; #10;
10        a = 1; b = 0; #10;
11        a = 1; b = 1; #10;
12    end
13 endmodule

```

- DUT instantiation: Design under test નું instance બનાવે છે
- Stimulus generation: Input test vectors પૂરાં પાડે છે
- કોઈ ports નહીં: Testbench top-level module છે

મેમરી ટ્રીક

Test Provides Stimulus, Monitors Response

પ્રશ્ન 5 [b ગુણ]

4 ડેટા ફ્લો પદ્ધતિ થી Half Adder નો વેરીલોગ પ્રોગ્રામ લખો.

જવાબ

```
1 module half_adder_dataflow(
2   input wire a, b,
3   output wire sum, carry
4 );
5
6 assign sum = a ^ b; // XOR for sum
7 assign carry = a & b; // AND for carry
8
9 endmodule
```

લોજિક Implementation:

- Sum: Inputs વચ્ચે XOR operation
- Carry: Inputs વચ્ચે AND operation
- Assign statement: Dataflow માટે continuous assignment
- Boolean operators: ^ (XOR), & (AND)

કોષ્ટક 25. Truth Table

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

મેમરી ટ્રીક

XOR Sums, AND Carries

પ્રશ્ન 5 [c ગુણ]

7 Encoder નું કાર્ય લખો. if..else વડે 8X3 Encoder નો વેરીલોગ કોડ બનાવો.

જવાબ

Encoder કાર્ય: Encoder 2^n input lines ને n output lines માં convert કરે છે. 8X3 encoder 8 inputs ને 3-bit binary output માં convert કરે છે.

કોષ્ટક 26. Priority Table

ઇનપુટ	Binary આઉટપુટ
I7	111
I6	110
I5	101
I4	100
I3	011
I2	010
I1	001
I0	000

Verilog કોડ:

```

1 module encoder_8x3(
2     input wire [7:0] i,
3     output reg [2:0] y
4 );
5
6 always @(*) begin
7     if (i[7])
8         y = 3'b111;
9     else if (i[6])
10        y = 3'b110;
11    else if (i[5])
12        y = 3'b101;
13    else if (i[4])
14        y = 3'b100;
15    else if (i[3])
16        y = 3'b011;
17    else if (i[2])
18        y = 3'b010;
19    else if (i[1])
20        y = 3'b001;
21    else if (i[0])
22        y = 3'b000;
23    else
24        y = 3'bxxx;
25 end
26
27 endmodule

```

- Priority encoding: ઉચ્ચ index inputs ને priority
- If-else chain: Priority logic implement કરે છે
- Binary encoding: Active input ને binary representation માં convert કરે છે

મેમરી ટ્રીક

Priority from High to Low, Binary Output Flows