VLSI Technology (4361102) - Summer 2024 Solution

Milav Dabgar

May 16, 2024

Question 1(a) [3 marks]

Draw the structure of FinFET and write its advantages.

Solution

Structure of FinFET showing Source, Drain, and Gate wrapping around the fin.

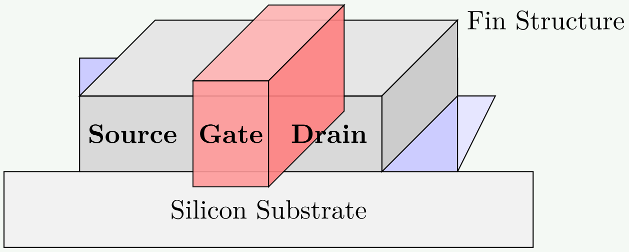

Figure 1. FinFET Structure

Advantages:

Advantage	Description
Better Control	Multiple gates provide superior channel control
Reduced Leakage	Lower off-state current due to 3D structure
Improved Performance	Higher drive current and faster switching

Table 1. FinFET Advantages

Mnemonic

"BCR - Better Control Reduces leakage"

Question 1(b) [4 marks]

Explain depletion and inversion of MOS structure under external bias

Solution

Table: MOS Bias Conditions

Bias Type	Gate Voltage	Channel State	Charge Carriers
Depletion	Slightly Positive	Depleted	Holes pushed away
Inversion	High Positive	Inverted	Electrons attracted

Table 2. MOS Bias Conditions

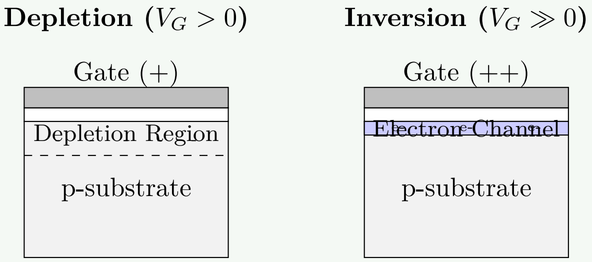

Figure 2. MOS Depletion and Inversion Layers

- **Depletion**: Positive gate voltage creates electric field pushing holes away, leaving immobile negative ions.
- **Inversion**: Higher voltage attracts electrons to surface, forming a conducting n-channel.

Mnemonic

"DI - Depletion Inverts to conducting channel"

Question 1(c) [7 marks]

Explain n-channel MOSFET with the help of its Current-Voltage characteristics.

Solution

Table: MOSFET Operating Regions

Region	Condition	Drain Current	Characteristics
Cut-off	$V_{GS} < V_{TH}$	$I_D \approx 0$	No conduction
Linear	$V_{DS} < V_{GS} - V_{TH}$	$I_D \propto V_{DS}$	Resistive behavior
Saturation	$V_{DS} \geq V_{GS} - V_{TH}$	$I_D \propto (V_{GS} - V_{TH})^2$	Current independent of V_{DS}

Table 3. MOSFET Operating Regions

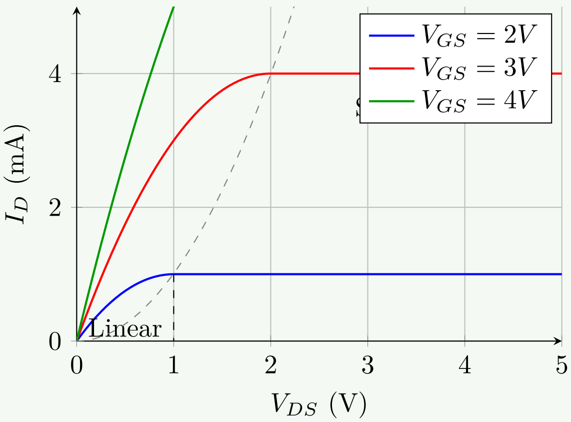

Figure 3. MOSFET IV Characteristics

Key Equations:

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

Mnemonic

"CLS - Cut-off, Linear, Saturation regions"

Question 1(c OR) [7 marks]

Define scaling. Compare full voltage scaling with constant voltage scaling. Write the disadvantages of scaling.

Solution

Definition: Scaling reduces device dimensions to increase density and performance.
Table: Scaling Comparison

Parameter	Full Voltage Scaling	Constant Voltage Scaling
Voltage	Reduced by α	Remains constant
Power Density	Constant	Increases by α
Electric Field	Constant	Increases by α
Performance	Better	Moderate improvement

Table 4. Scaling Comparison

Disadvantages:

- **Short Channel Effects**: Channel length modulation increases.
- **Hot Carrier Effects**: High electric fields damage devices.
- **Quantum Effects**: Tunneling currents increase significantly.

Mnemonic

"SHQ - Short channel, Hot carriers, Quantum effects"

Question 2(a) [3 marks]

Draw two input NAND gate using CMOS.

Solution

CMOS NAND gate uses two parallel pMOS transistors and two series nMOS transistors.

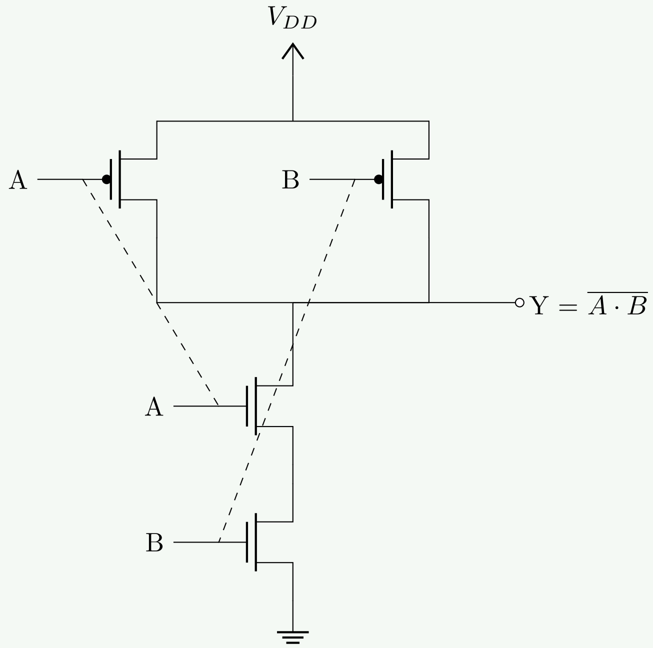

Figure 4. 2-Input CMOS NAND Gate

Table: NAND Truth Table

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

Table 5. NAND Truth Table

Mnemonic

"PP-SS: Parallel PMOS, Series NMOS"

Question 2(b) [4 marks]

Explain noise immunity and noise margin for nMOS inverter.

Solution

Table: Noise Parameters

Parameter	Definition	Formula
NMH	High noise margin	$V_{OH} - V_{IH}$
NML	Low noise margin	$V_{IL} - V_{OL}$
Noise Immunity	Ability to reject noise	Min(NMH, NML)

Table 6. Noise Parameters

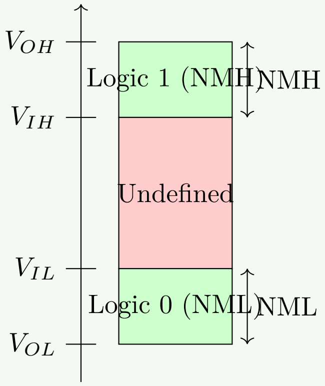

Figure 5. Noise Margins

- V_{IL}: Maximum low input voltage recognized as logic 0.
- V_{IH}: Minimum high input voltage recognized as logic 1.
- **Good noise immunity**: Large noise margins prevent false switching due to noise.

Mnemonic

"HILOL - High/Low Input/Output Levels"

Question 2(c) [7 marks]

Explain Voltage Transfer Characteristics (VTC) of CMOS inverter.

Solution

Table: VTC Regions

Region	Input Range	Output	Transistor States
A	0 to V_{TN}	V_{DD}	pMOS ON, nMOS OFF
B	V_{TN} to $V_{DD}/2$	Transition	Both ON (Sat/Lin)
C	$V_{DD}/2$ (Threshold)	Rapid Drop	Both Saturation
D	$V_{DD}/2$ to $V_{DD} - \vert V_{TP}\vert$	Transition	Both ON (Lin/Sat)
E	$V_{DD} - \vert V_{TP}\vert$ to V_{DD}	$0V$	pMOS OFF, nMOS ON

Table 7. VTC Regions

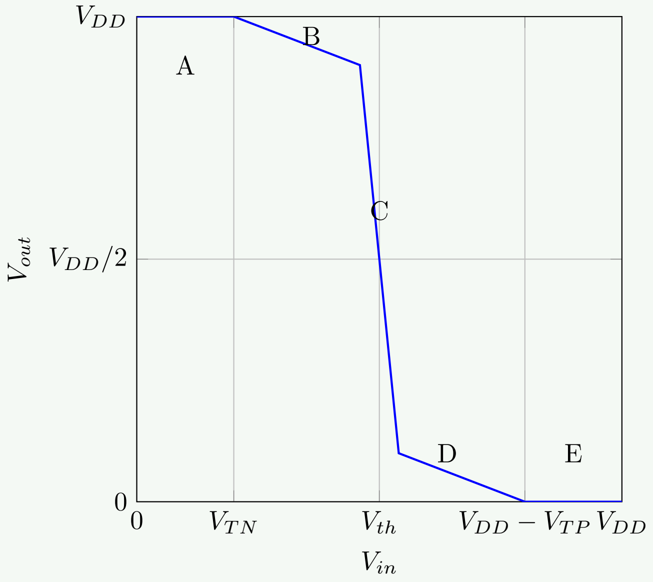

Figure 6. CMOS Inverter VTC

Key Features:

- **Sharp transition**: Ideal switching behavior at V_{th}.
- **High gain**: Large slope in transition region (Region C).
- **Rail-to-rail**: Output swings full supply range (0 to V_{DD}).

Mnemonic

"ASH - A-region, Sharp transition, High gain"

Question 2(a OR) [3 marks]

Implement NOR2 gate using depletion load nMOS.

Solution

Depletion load NOR2 uses a depletion nMOS as load and two parallel nMOS as drivers.

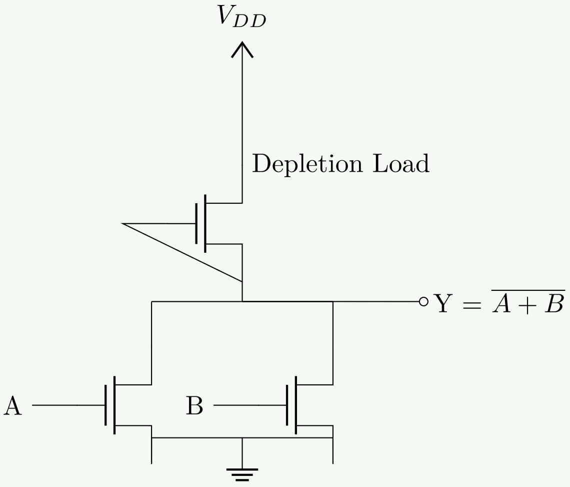

Figure 7. Depletion Load NOR2

Table: NOR2 Truth Table

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

Table 8. NOR2 Truth Table

Mnemonic

"DPN - Depletion load, Parallel NMOS"

Question 2(b OR) [4 marks]

Differentiate between enhancement load inverter and Depletion load inverter.

Solution

Table: Load Inverter Comparison

Parameter	**Enhancement Load**	**Depletion Load**
Threshold Voltage	$V_T > 0$	$V_T < 0$
Gate Connection	$V_{GS} = V_{DS}$	$V_{GS} = 0$
Logic High (V_{OH})	$V_{DD} - V_T$	V_{DD}
Power Consumption	Higher	Lower
Switching Speed	Slower	Faster

Table 9. Inverter Load Types Comparison

Mnemonic

"EPDLH - Enhancement Positive, Depletion Lower power, Higher speed"

Question 2(c OR) [7 marks]

Explain Depletion load nMOS inverter with its VTC.

Solution

Circuit Operation:

- **Load transistor**: Always conducting ($V_{GS} = 0, V_T < 0$). Acts as a constant current source.
- **Driver transistor**: Controlled by input voltage (V_{in}).
- **Output**: Determined by voltage divider action between load and driver resistances.

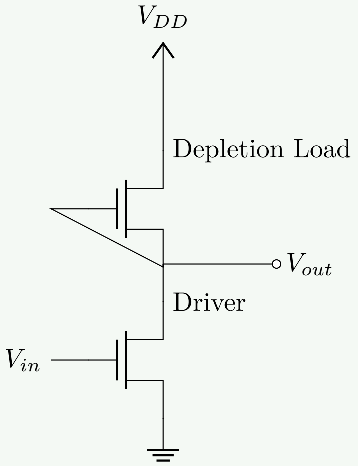

Figure 8. Depletion Load Inverter

Table: Operating Points

Input State	Driver	Load	Output
$\mathbf{V_{in} = 0}$	OFF	ON (Linear)	V_{DD}
$\mathbf{V_{in} = V_{DD}}$	ON (Linear)	ON (Sat)	$\approx 0V$

Table 10. Inverter Operating Points

VTC Characteristics:

- V_{OH}: Reaches full V_{DD} (unlike enhancement load which drops V_T).
- V_{OL}: Low voltage close to 0V.
- **Transition**: Sharp switching due to better load characteristics.

Mnemonic

"DLB - Depletion Load gives Better high output"

Question 3(a) [3 marks]

Implement EX-OR using Depletion load nMOS.

Solution

To implement XOR ($Y = A \oplus B = A\bar{B} + \bar{A}B$) using depletion load nMOS logic (which implements $\overline{\text{Pull-Down}}$), we organize the pull-down network to realize the complement of XOR, which is XNOR ($AB + \bar{A}\bar{B}$). A standard implementation uses 4 transistors in the pull-down network to realize $Y = \overline{(A + \bar{B})(\bar{A} + B)} = A \oplus B$.

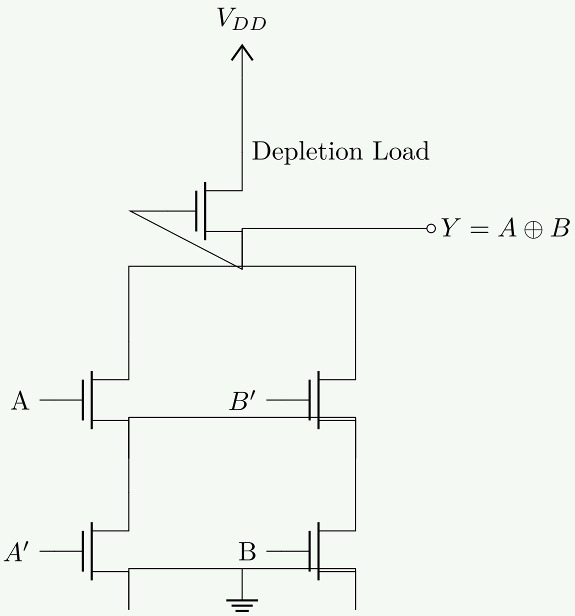

Figure 9. Depletion Load NMOS EX-OR Gate

Logic Equation: $Y = \overline{(A + \bar{B}) \cdot (\bar{A} + B)} = \overline{A\bar{A} + AB + \bar{B}\bar{A} + \bar{B}B} = \overline{0 + AB + \bar{A}\bar{B} + 0} = \overline{A \odot B} = A \oplus B$

Mnemonic

"XOR - eXclusive OR, different inputs give 1"

Question 3(b) [4 marks]

Explain design hierarchy with example.

Solution

Table: Hierarchy Levels

Level	Component	Example
System	Complete chip	Microprocessor
Module	Functional blocks	ALU, Memory
Gate	Logic gates	NAND, NOR
Transistor	Individual devices	MOSFET

Table 11. Design Hierarchy Levels

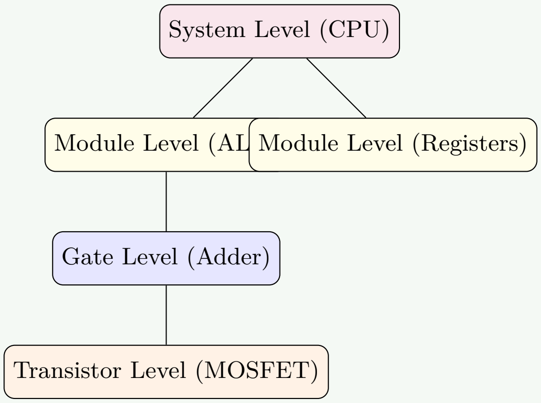

Figure 10. Design Hierarchy Tree

Benefits:

- **Modularity**: Independent design and testing of blocks.
- **Reusability**: Common blocks (like adders) used multiple times.
- **Maintainability**: Easy debugging and modification.

Mnemonic

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

Question 3(c) [7 marks]

Draw and explain Y chart design flow.

Solution

The Y-chart represents the three domains of VLSI design: Behavioral, Structural, and Physical.

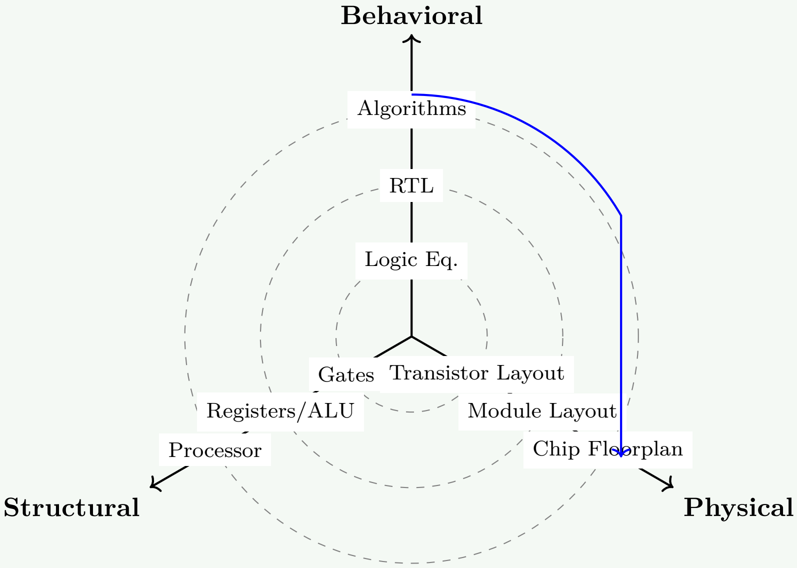

Figure 11. Gajski-Kuhn Y-Chart

Table: Y-Chart Domains

Domain	Description	Examples
Behavioral	What system does	Algorithms, RTL
Structural	How it's organized	Architecture, Gates
Physical	Where components placed	Floorplan, Layout

Table 12. Y-Chart Domains

Design Flow:

- **Top-down**: Behavioral → Structural → Physical.
- **Bottom-up**: Physical constraints influence upper levels.

Mnemonic

"BSP - Behavioral, Structural, Physical domains"

Question 3(a OR) [3 marks]

Implement NAND2 - SR latch using CMOS

Solution

NAND SR Latch consists of two cross-coupled NAND gates.

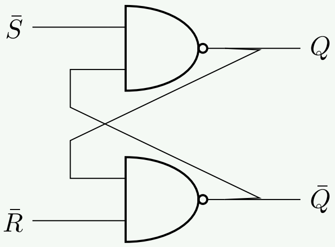

Figure 12. CMOS NAND SR Latch

Table: SR Latch Operation

S	R	Q	Q'	State
0	0	1	1	Invalid
0	1	1	0	Set
1	0	0	1	Reset
1	1	Q	Q'	Hold

Table 13. NAND SR Latch Truth Table (Active Low Inputs)

Mnemonic

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

Question 3(b OR) [4 marks]

Which method is used to transfer pattern or mask on the silicon wafer? Explain it with

neat diagrams

Solution

Method: **Photolithography** is used to transfer geometric shapes on a mask to the surface of a silicon wafer.

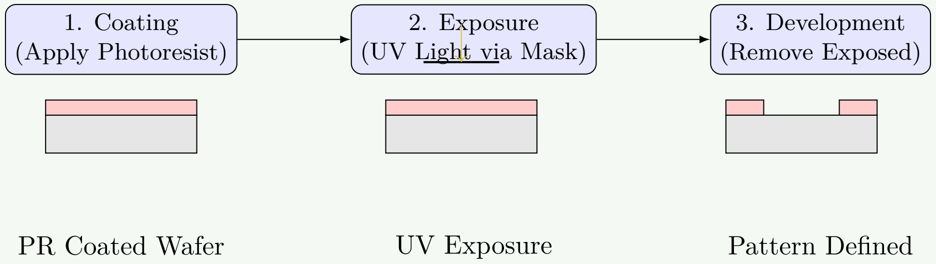

Figure 13. Photolithography Process

Process Steps:

- **Coating**: Applying a thin layer of photoresist.
- **Exposure**: Exposing resist to UV light through a mask.
- **Development**: Dissolving the exposed (positive) or unexposed (negative) resist.

Mnemonic

"CED - Coating, Exposure, Development"

Question 3(c OR) [7 marks]

Which are the methods used to deposit metal in MOSFET fabrication? Explain deposition in detail with proper diagram.

Solution

Table: Metal Deposition Methods

Method	Technique	Application
PVD	Sputtering, Evaporation	Aluminum, Copper
CVD	CVD, PECVD	Tungsten, Titanium
Electroplating	Electrochemical	Copper interconnects

Table 14. Metal Deposition Methods

Sputtering Process: In sputtering, ions from a plasma are accelerated towards a target material, ejecting atoms which settle on the wafer.

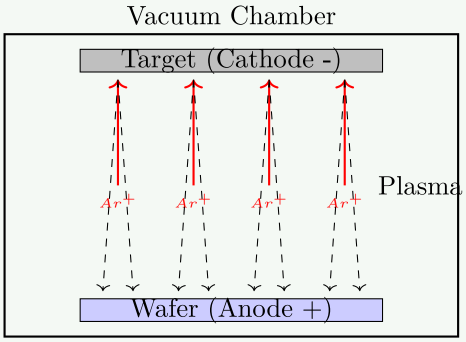

Figure 14. Sputtering System

Advantages:

- **Uniform thickness**: Excellent step coverage.
- **Low temperature**: Preserves device integrity.
- **Variety**: Can deposit alloys and compounds.

Mnemonic

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

Question 4(a) [3 marks]

Implement Z= ((A+B+C) · (D+E+F). G)' with depletion nMOS load.

Solution

The logic function $Z = \overline{(A + B + C) \cdot (D + E + F) \cdot G}$ is implemented using a depletion load and a pull-down network. The pull-down network consists of three series blocks corresponding to the AND operations: 1. Parallel inputs A, B, C (OR) 2. Parallel inputs D, E, F (OR) 3. Single input G Since these are ANDed, the blocks are in series.

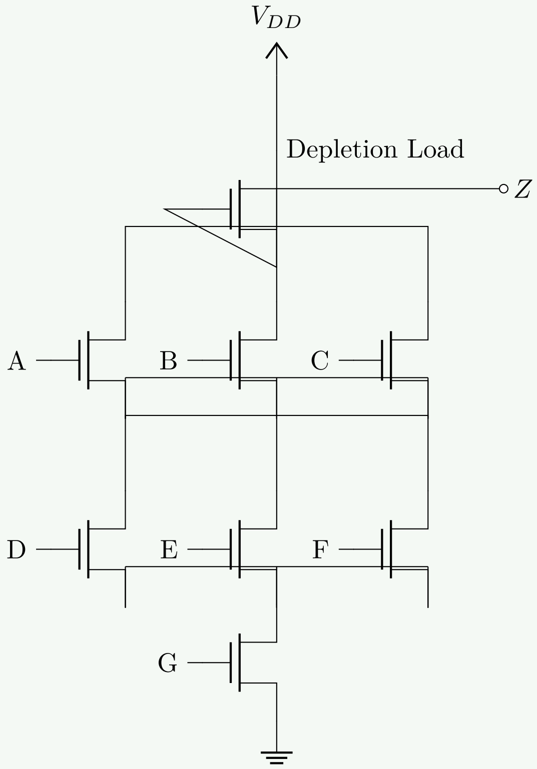

Figure 15. Logic Implementation of Z

Mnemonic

"POI - Parallel OR, Inversion at output"

Question 4(b) [4 marks]

List and explain the design styles used in VERILOG.

Solution

Table: Verilog Design Styles

Style	**Description**	**Use Case**	**Example**
Behavioral	Algorithm description	High-level modeling	`always` blocks
Dataflow	Boolean expressions	Combinational logic	`assign` statements
Structural	Component instantiation	Hierarchical design	module connections
Gate-level	Primitive gates	Low-level design	`and`, `or`, `not` gates

Table 15. Verilog Design Styles

Characteristics:

- **Behavioral**: Describes what circuit does (functionality).
- **Structural**: Shows how components connect (netlist).
- **Mixed**: Combines multiple styles for complex ideas.

Mnemonic

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

Question 4(c) [7 marks]

Implement NAND2 SR latch using CMOS and also implement NOR2 SR latch using CMOS.

Solution

NAND2 SR Latch (Verilog):

```verilog
module nand_sr_latch(
    input S, R,
    output Q, Q_bar
);
    nand(Q, S, Q_bar);
    nand(Q_bar, R, Q);
endmodule
```

NOR2 SR Latch (Verilog):

```verilog
module nor_sr_latch(
    input S, R,
    output Q, Q_bar
);
    nor(Q_bar, R, Q);
    nor(Q, S, Q_bar);
endmodule
```

Comparison:

- **NAND**: Set/Reset with low inputs (Active Low).
- **NOR**: Set/Reset with high inputs (Active High).

Mnemonic

"NAND-Low, NOR-High active"

Question 4(a OR) [3 marks]

Implement Y= (ABC + DE + F)' with depletion nMOS load.

Solution

The logic function $Y = \overline{(ABC) + (DE) + F}$ implements Sum of Products (ORing of AND terms). The pull-down network consists of three parallel branches: 1. A, B, C in Series (AND) 2. D, E in Series (AND) 3. F Single

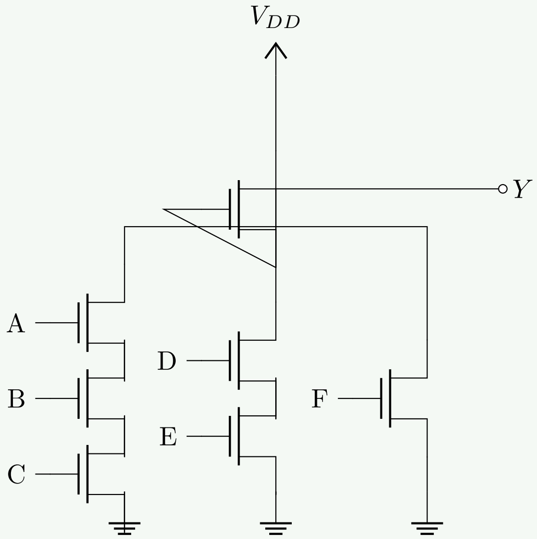

Figure 16. Logic Implementation of Y

Mnemonic

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

Question 4(b OR) [4 marks]

Write Verilog Code to implement full adder.

Solution

```
module full_adder(
    input a, b, cin,
    output sum, cout
);
    assign sum = a ^ b ^ cin;
    assign cout = (a & b) | (cin & (a ^ b));
endmodule
```

Logic Functions:

- **Sum**: Triple XOR operation ($A \oplus B \oplus C_{in}$)
- **Carry**: Majority function of inputs ($AB + BC_{in} + AC_{in}$)

Mnemonic

"XOR-Sum, Majority-Carry"

Question 4(c OR) [7 marks]

Implement Y =(S1'S0'I0 + S1'S0 I1 + S1 S0' I2 + S1 S2 I3) using depletion load

Solution

4:1 Multiplexer Verilog Code (Assuming S2 corresponds to S0 in context of 4-input):

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

Table: Multiplexer Selection

S1	S0	Selected Input	Output
0	0	I0	$Y = I0$
0	1	I1	$Y = I1$
1	0	I2	$Y = I2$
1	1	I3	$Y = I3$

Table 16. Multiplexer Selection

Mnemonic

"DAO - Decoder, AND gates, OR combination"

Question 5(a) [3 marks]

Implement the logic function G = (PQR +U(S+T))' using CMOS

Solution

The function $G = \overline{(PQR) + U(S+T)}$ requires a CMOS implementation. **Pull-Down Network (PDN)**: Implements $(PQR) + U(S+T)$ via nMOS.

- P, Q, R in Series (AND).
- S, T in Parallel (OR) is correct for $(S+T)$.
- U in Series with $(S||T)$.
- The block $(P - Q - R)$ is in Parallel with $(U - (S||T))$.

Pull-Up Network (PUN): Implements dual via pMOS.

- P, Q, R in Parallel.
- S, T in Series.
- U in Parallel with $(S - T)$.
- The block $(P||Q||R)$ is in Series with $(U||(S - T))$.

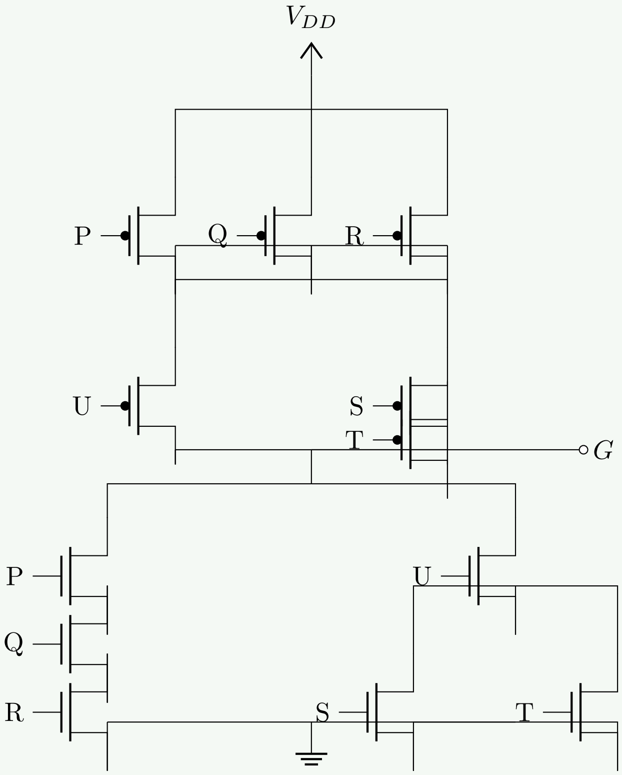

Figure 17. CMOS Implementation of Logic G

Mnemonic

"PSSP - Parallel Series Series Parallel"

Question 5(b) [4 marks]

Implement 8x1 multiplexer using Verilog

Solution

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

Table: 8:1 MUX Selection

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

Table 17. 8:1 MUX Truth Table (Partial)

Mnemonic

"Case-Always: Use case statement in always block"

Question 5(c) [7 marks]

Implement 4 bit full adder using structural modeling style in Verilog.

Solution

Verilog Code (Structural):

```verilog
module full_adder(
    input a, b, cin,
    output sum, cout
);
    assign sum = a ^ b ^ cin;
    assign cout = (a & b) | (cin & (a ^ b));
endmodule

module full_adder_4bit(
    input [3:0] a, b,
    input cin,
    output [3:0] sum,
    output cout
);
    wire c1, c2, c3;

    // Instantiating 4 Full Adders
    full_adder fa0(.a(a[0]), .b(b[0]), .cin(cin),
                   .sum(sum[0]), .cout(c1));
    full_adder fa1(.a(a[1]), .b(b[1]), .cin(c1),
                   .sum(sum[1]), .cout(c2));
    full_adder fa2(.a(a[2]), .b(b[2]), .cin(c2),
                   .sum(sum[2]), .cout(c3));
    full_adder fa3(.a(a[3]), .b(b[3]), .cin(c3),
                   .sum(sum[3]), .cout(cout));
endmodule
```

Table: Ripple Carry Addition

Stage	Inputs	Carry In	Sum	Carry Out
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

Table 18. Ripple Carry Structure

Mnemonic

"RCC - Ripple Carry Chain connection"

Question 5(a OR) [3 marks]

Implement logic function Y = ((AF(D + E))+ (B+ C))' using CMOS.

Solution

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

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

PUN (Implementation of Dual):

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

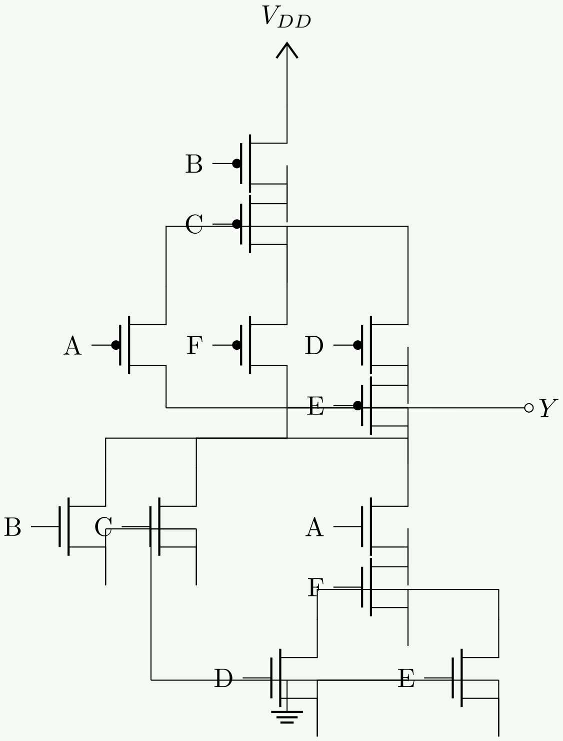

Figure 18. CMOS Implementation of Y

Mnemonic

"PNAI - PMOS Network Applies Inversion"

Question 5(b OR) [4 marks]

Implement 4 bit up counter using Verilog

Solution

```verilog
module counter_4bit_up(
    input clk, reset,
    output reg [3:0] count
);
    always @(posedge clk or posedge reset) begin
        if (reset)
            count <= 4'b0000;
        else
            count <= count + 1;
    end
endmodule
```

Table: Counter Sequence

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

Table 19. Counter Sequence

Mnemonic

"SRA - Synchronous Reset with Auto rollover"

Question 5(c OR) [7 marks]

Implement 3:8 decoder using behavioral modeling style in Verilog.

Solution

```verilog
module decoder_3to8(
    input [2:0] select,
    input enable,
    output reg [7:0] out
);
    always @(*) begin
        if (enable) begin
            case(select)
                3'b000: out = 8'b00000001;
                3'b001: out = 8'b00000010;
                3'b010: out = 8'b00000100;
                3'b011: out = 8'b00001000;
                3'b100: out = 8'b00010000;
                3'b101: out = 8'b00100000;
                3'b110: out = 8'b01000000;
                3'b111: out = 8'b10000000;
                default: out = 8'b00000000;
            endcase
        end else begin
            out = 8'b00000000;
        end
    end
endmodule
```

Applications:

- **Memory addressing**: Chip select generation.
- **Data routing**: Channel selection.
- **Control logic**: State machine outputs.

Mnemonic

"BEOH - Behavioral Enable One-Hot decoder"