

# VLSI Technology (4353206) - Winter 2024 Solution

Milav Dabgar

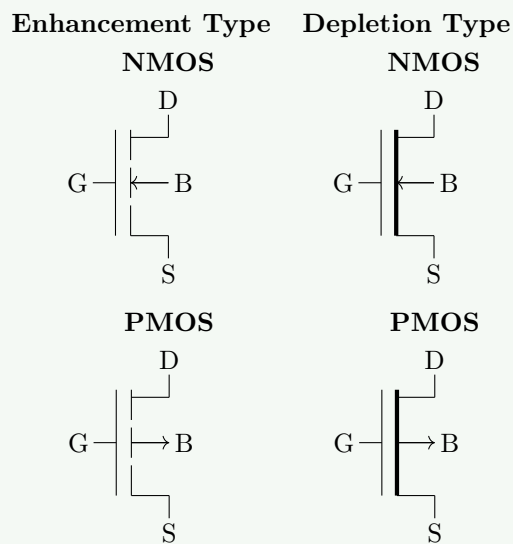
November 29, 2024

## Question 1(a) [3 marks]

Draw all symbols for enhancement and depletion type MOSFET.

### Solution

MOSFET Symbols:



### Key Differences:

- **Enhancement:** Broken line represents no physical channel at  $V_{GS} = 0$ .
- **Depletion:** Solid thick line represents existing physical channel at  $V_{GS} = 0$ .
- **Arrows:** Inward for NMOS (p-substrate), Outward for PMOS (n-substrate).

### Mnemonic

“Enhancement Needs voltage, Depletion has Default channel”

## Question 1(b) [4 marks]

Define: 1) Hierarchy 2) Regularity

### Solution

Definitions:

Term	Definition	Application
<b>Hierarchy</b>	Top-down design approach where complex systems are broken into smaller, manageable modules.	Used in VLSI design flow from system level to transistor level.
<b>Regularity</b>	Design technique using repeated identical structures to reduce complexity.	Memory arrays, processor datapaths use regular structures.

**Key Points:**

- **Hierarchy benefits:** Easier design verification, modular testing, team collaboration.
- **Regularity advantages:** Reduced design time, better yield, simplified layout.
- **Design flow:** System → Behavioral → RTL → Gate → Layout.

**Mnemonic**

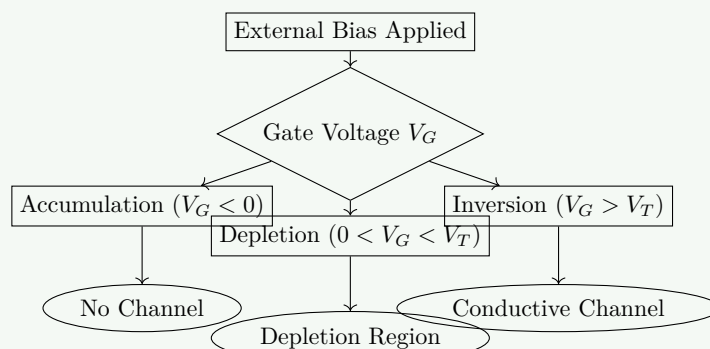
“Hierarchy Helps organize, Regularity Reduces complexity”

## Question 1(c) [7 marks]

Explain MOS under external bias.

**Solution****MOS Bias Conditions:**

Bias Condition	Gate Voltage	Channel Formation	Current Flow
<b>Accumulation</b>	$V_G < 0$ (NMOS)	Majority carriers accumulate	No channel
<b>Depletion</b>	$0 < V_G < V_T$	Depletion region forms	Minimal current
<b>Inversion</b>	$V_G > V_T$	Minority carriers form channel	Channel conducts

**Operation Flow:**

**Figure 1.** MOS Operating Modes

**Key Concepts:**

- **Band bending:** External voltage bends energy bands at oxide-silicon interface.
- **Threshold voltage ( $V_T$ ):** Minimum gate voltage needed for channel formation.
- **Inversion:** When surface potential  $\phi_s = 2\phi_F$ , strict inversion occurs.

**Mnemonic**

“Accumulation Attracts, Depletion Depletes, Inversion Inverts carriers”

## Question 1(c) OR [7 marks]

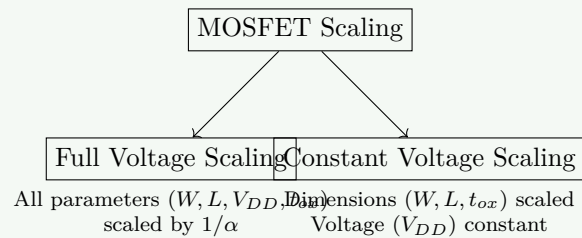
What is the need for scaling? Explain types of scaling with its effect.

### Solution

Need for Scaling:

Parameter	Benefit	Impact
Area reduction	More transistors per chip	Higher integration density
Speed increase	Reduced delays	Better performance
Power reduction	Lower power consumption	Portable devices
Cost reduction	Cheaper per function	Market competitiveness

Types of Scaling:



Scaling Effects:

- **Full voltage scaling:** Maintains constant electric field. Power density remains constant.
- **Constant voltage scaling:** Electric field increases. Power density increases significantly.

### Mnemonic

“Scaling Saves Space, Speed, and Spending”

## Question 2(a) [3 marks]

Write short note on FPGA.

### Solution

FPGA Characteristics:

Feature	Description	Advantage
Field Programmable	Configurable after manufacturing	Flexibility in design
Gate Array	Array of logic blocks	Parallel processing
Reconfigurable	Can be reprogrammed	Prototype development

Details:

- **Applications:** Digital signal processing, embedded systems, prototyping.
- **Architecture:** Matrix of CLBs (Configurable Logic Blocks) connected by programmable routing.
- **Programming:** Typically SRAM-based (volatile) or Flash/Antifuse based.

### Mnemonic

“FPGA: Flexible Programming for Gate Arrays”

## Question 2(b) [4 marks]

Compare semi-custom and full custom design methodologies.

### Solution

Comparison:

Parameter	Semi-Custom	Full Custom
Design Time	Shorter (weeks)	Longer (months)
Cost	Lower development cost	Higher development cost
Performance	Moderate performance	Highest performance
Area Efficiency	Less efficient	Most efficient
Applications	ASICs, moderate volume	Microprocessors, high volume
Design Effort	Standard cells used	Every transistor designed manually

### Mnemonic

“Semi-custom is Standard, Full custom is Finest”

## Question 2(c) [7 marks]

Explain MOSFET operation for 1)  $0 < V_{DS} < V_{DSAT}$  2)  $V_{DS} = V_{DSAT}$  3)  $V_{DS} > V_{DSAT}$

### Solution

Operating Regions:

Region	Condition	Channel State	Current ( $I_D$ )
Linear	$V_{DS} < V_{DSAT}$	Uniform continuous channel	$\propto V_{DS}$
Saturation Onset	$V_{DS} = V_{DSAT}$	Pinch-off begins at drain	Max linear current
Saturation	$V_{DS} > V_{DSAT}$	Pinched off channel	Constant

Diagrammatic Representation:

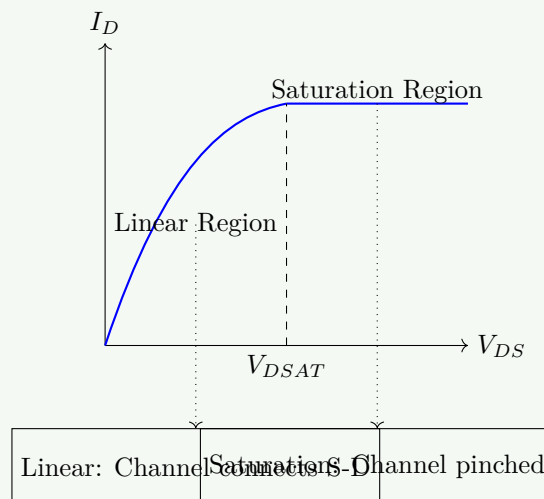


Figure 2. MOSFET I-V Characteristic

Analysis:

- **Linear Region:** Channel behaves as a voltage-controlled resistor.  $I_D$  increases linearly with  $V_{DS}$ .
- **Saturation Region:** Channel is pinched off at drain end. Current flow is due to electric field drift across depletion region.  $I_D$  becomes independent of  $V_{DS}$  (ignoring channel length modulation).
- $V_{DSAT}$ : Saturation voltage, typically  $V_{GS} - V_T$ .

#### Mnemonic

“Linear Likes VDS, Saturation Says no more”

## Question 2(a) OR [3 marks]

Explain standard cell-based design.

#### Solution

##### Overview:

Component	Description	Benefit
Standard Cells	Pre-designed logic gates (AND, OR, FF)	Faster design cycle
Cell Library	Collection of characterized cells with physical layouts	Predictable performance
Place & Route	Automated layout generation using tools	Reduced manual effort

##### Design Flow:

- Logic Synthesis → Placement → Routing → Verification.
- EDA tools handle the complex physical implementation.
- Provides a balance between performance, area, and power.

#### Mnemonic

“Standard Cells Speed up Synthesis”

## Question 2(b) OR [4 marks]

Draw and explain Y-chart.

#### Solution

Gajski-Kuhn Y-Chart:

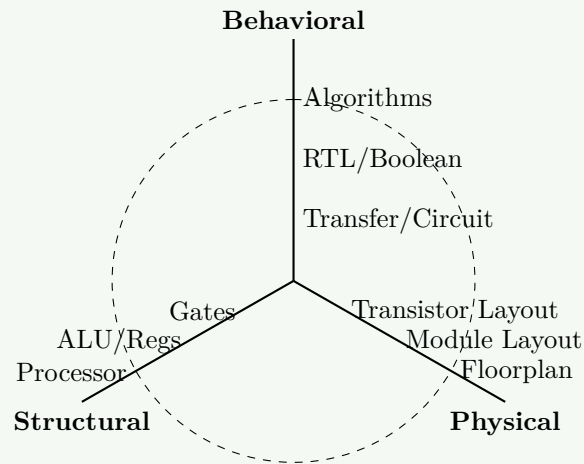


Figure 3. Y-Chart Representation

**Domains:**

- **Behavioral:** Describes *what* the system does (Functionality).
- **Structural:** Describes *how* components are interconnected.
- **Physical:** Describes the *geometry* and layout of the implementation.

**Mnemonic**

“Y-chart: behaVior, Structure, PHYsical”

**Question 2(c) OR [7 marks]**

Explain gradual channel approximation for MOSFET current-voltage characteristics.

**Solution****Gradual Channel Approximation (GCA):****Assumptions:**

Assumption	Description	Justification
<b>Gradual Channel</b>	Variation of field along channel ( $y$ ) $\ll$ variation perpendicular ( $x$ ).	Valid for long channel devices ( $L \gg t_{ox}$ ).
<b>1D Analysis</b>	Current flows mainly in $y$ -direction (source to drain).	Simplifies potential analysis.
<b>Drift Current</b>	Diffusion current is neglected.	Dominant mechanism in strong inversion.

**Derivation Summary:**

- Induced charge density:  $Q_n(y) = -C_{ox}[V_{GS} - V(y) - V_T]$ .
- Drain Current:  $I_D = -W\mu_n Q_n(y) \frac{dV}{dy}$ .
- Integrating from  $y = 0$  to  $L$  and  $V = 0$  to  $V_{DS}$ :

$$I_D = \mu_n C_{ox} \frac{W}{L} \left[ (V_{GS} - V_T)V_{DS} - \frac{V_{DS}^2}{2} \right]$$

**Limitations:**

- **Short Channel Effects:** GCA fails when  $L$  is comparable to depletion widths.
- **Velocity Saturation:** Carrier velocity doesn't increase linearly with high fields.

**Mnemonic**

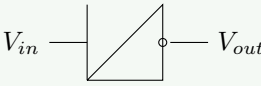
“Gradual change Gives simple Gain equations”

**Question 3(a) [3 marks]**

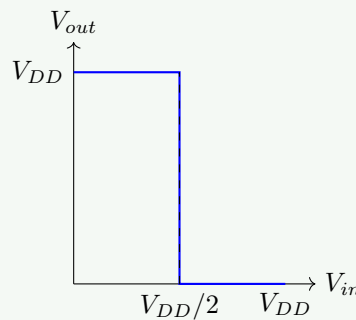
Draw symbol and write truth table of ideal inverter. Draw and explain VTC of ideal inverter.

**Solution**

**Ideal Inverter:**

Symbol	Truth Table						
	<table> <tr> <th><math>V_{in}</math></th><th><math>V_{out}</math></th></tr> <tr> <td>0</td><td><math>1(V_{DD})</math></td></tr> <tr> <td><math>1(V_{DD})</math></td><td>0</td></tr> </table>	$V_{in}$	$V_{out}$	0	$1(V_{DD})$	$1(V_{DD})$	0
$V_{in}$	$V_{out}$						
0	$1(V_{DD})$						
$1(V_{DD})$	0						

**Voltage Transfer Characteristic (VTC):**



**Figure 4.** Ideal VTC

**Characteristics:**

- Infinite gain at switching threshold ( $V_{DD}/2$ ).
- Noise margins  $NM_H = NM_L = V_{DD}/2$ .
- Zero power consumption in steady states.

**Mnemonic**

“Ideal Inverter: Infinite gain, Instant switching”

**Question 3(b) [4 marks]**

Explain generalized inverter circuit with its VTC.

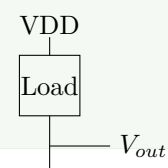
**Solution**

**Generalized Inverter Structure:**

**Components:**

1. **Driver:** Pull-down NMOS transistor.
2. **Load:** Pull-up device (Resistor/Transistor).
3. **Operation:** Input separates driver ON/OFF states.

**Circuit:**



$V_{in}$  Driver

**VTC Regions:**

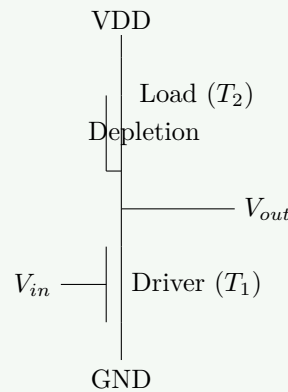
- **Region 1 (High Output):**  $V_{in} < V_T$ . Driver OFF, Load pulls up to  $V_{OH} \approx V_{DD}$ .
- **Region 2 (Transition):** Both devices conducting. Voltage drops sharply.
- **Region 3 (Low Output):**  $V_{in}$  High. Driver ON (Linear).  $V_{out} = V_{OL}$ .

**Mnemonic**

“Generalized design: Driver pulls Down, Load lifts Up”

**Question 3(c) [7 marks]**

Describe depletion load nMOS inverter with its circuit, operating region and VTC.

**Solution****Depletion Load NMOS Inverter:**

**Figure 5.** Depletion Load Inverter Circuit

**Operating Regions:**

Input State	Driver ( $T_1$ )	Load ( $T_2$ )	Output
$V_{in} < V_{TN}$ (Low)	OFF (Cutoff)	ON (Linear)	$V_{OH} = V_{DD}$
$V_{in}$ Trans.	Saturation	Saturation	Falling
$V_{in} > V_{IH}$ (High)	ON (Linear)	ON (Saturation)	$V_{OL}$ (Small)

**VTC Characteristics:**

- **High Output:** Full  $V_{DD}$  because depletion load pulls up fully.
- **Transition:** Sharp, providing good noise margins.
- **Low Output:** Non-zero  $V_{OL} \approx 0$  (Ratioed logic).

**Mnemonic**

“Depletion Device Delivers Decent drive”

**Question 3(a) OR [3 marks]**

Explain noise margin.



**Solution**

**Definition:** Noise Margin is the maximum noise voltage added to an input signal that does not cause a change in the logic state of the output. It measures noise immunity.

**Parameters:**

Parameter	Formula	Description
NMH	$V_{OH} - V_{IH}$	High Noise Margin
NML	$V_{IL} - V_{OL}$	Low Noise Margin
$V_{OH}$	-	Min Output High Voltage
$V_{OL}$	-	Max Output Low Voltage
$V_{IH}$	-	Min Input High Voltage
$V_{IL}$	-	Max Input Low Voltage

**Mnemonic**

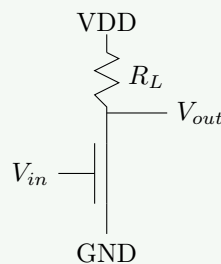
“Noise Margins Maintain signal integrity”

**Question 3(b) OR [4 marks]**

Explain resistive load inverter.

**Solution**

**Circuit and Analysis:**



**Figure 6.** Resistive Load Inverter

**Operation:**

- **Input Low:** NMOS OFF. Output pulled to  $V_{DD}$  through  $R_L$ .
- **Input High:** NMOS ON. Current flows through  $R_L$  and NMOS. Output  $V_{OL} = V_{DD} \frac{R_{MN}}{R_{MN} + R_L}$ .

**Drawbacks:**

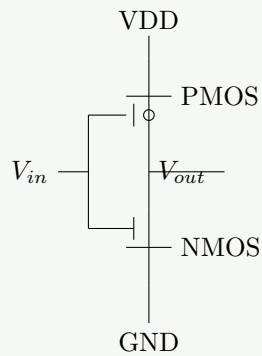
- Large area required for resistor on-chip.
- Static power consumption when output is low ( $V_{DD}^2/R_L$ ).

**Mnemonic**

“Resistor Restricts current, Reduces performance”

**Question 3(c) OR [7 marks]**

Explain CMOS inverter with its VTC.

**Solution****CMOS Inverter:****Figure 7.** CMOS Inverter Circuit**VTC Regions & Operation:**

Region	Input Range	PMOS	NMOS	Output
1	$V_{in} < V_{TN}$	ON (Linear)	OFF	$V_{DD}$
2	$V_{TN} < V_{in} < V_{DD}/2$	ON (Lin)	ON (Sat)	High Drop
3	$V_{in} \approx V_{DD}/2$	Saturation	Saturation	Switch
4	$V_{DD}/2 < V_{in} < V_{DD} + V_{TP}$	ON (Sat)	ON (Lin)	Low Drop
5	$V_{in} > V_{DD} + V_{TP}$	OFF	ON (Linear)	0

**Advantages:** - Zero static power. - Full rail-to-rail logic swing. - High noise margins.

**Mnemonic**

“CMOS: Complementary for Complete performance”

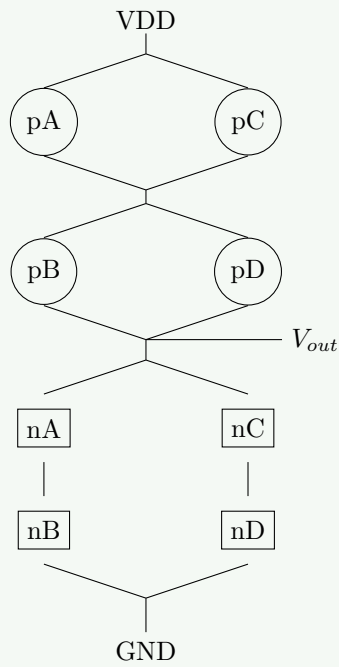
**Question 4(a) [3 marks]**

Draw AOI with CMOS implementation.

**Solution**

**AOI Logic:**  $Y = \overline{AB + CD}$

**CMOS Implementation:**



### Mnemonic

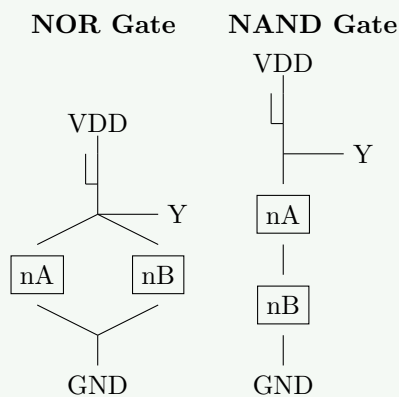
“AOI: AND-OR then Invert”

Question 4(b) [4 marks]

Implement two input NOR and NAND gate using depletion load nMOS.

## Solution

### Depletion Load Gates:



**Truth Tables:**

A	B	NOR	NAND
0	0	1	1
0	1	0	1
1	0	0	1
1	1	0	0

**Mnemonic**

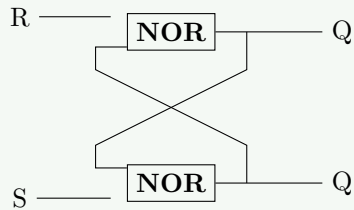
“NOR needs None high, NAND Needs All high to be low”

**Question 4(c) [7 marks]**

Implement CMOS SR latch using NOR2 and NAND2 gates.

**Solution**

SR Latch using NOR Gates:



**Figure 9.** SR Latch Logic Symbol

**CMOS Implementation (NOR Latch):** Two CMOS NOR2 gates with cross-coupled inputs.

- **Top NOR:** Inputs R and Q'. Output Q.
- **Bottom NOR:** Inputs S and Q. Output Q'.

**State Table:**

S	R	Q(n+1)	State
0	0	Q(n)	Hold
0	1	0	Reset
1	0	1	Set
1	1	0	Invalid

**Mnemonic**

“SR latch: Set-Reset with cross-coupled gates”

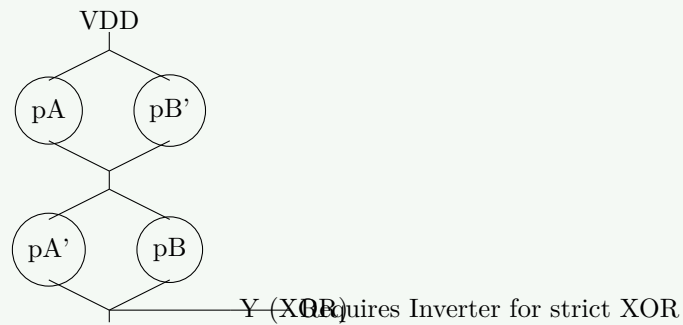
**Question 4(a) OR [3 marks]**

Implement XOR function using CMOS.

**Solution**

**XOR Function:**  $Y = A \oplus B = A\bar{B} + \bar{A}B$ . **Inverted logic:**  $\bar{Y} = \overline{A\bar{B} + \bar{A}B} = (A + B)(\bar{A} + \bar{B}) = XNOR$ . Usually XOR is implemented using 12 transistors (with inverters) or transmission gates (6-8 transistors).

**Static CMOS (12T):** Implement XNOR in PDN and invert, or XOR directly using complex gate. Direct XOR PDN: Parallel (A series B') and (A' series B). PUN: Series (A parallel B') and (A' parallel B).



**Figure 10.** CMOS Structure (Logic Analysis)

*Note: Standard static CMOS XOR requires identifying  $\bar{Y} = XNOR = AB + \bar{A}\bar{B}$ . The PDN implements  $AB + \bar{A}\bar{B}$ . The PUN is dual.*

#### Mnemonic

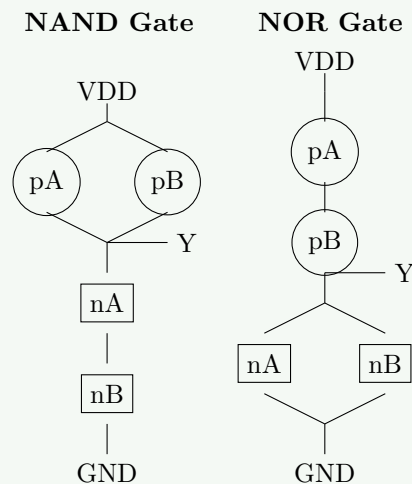
“XOR: eXclusive OR, different inputs give 1”

### Question 4(b) OR [4 marks]

Implement two input NOR and NAND gate using CMOS.

#### Solution

CMOS Gates:



#### Mnemonic

“NAND: Parallel PMOS, Series NMOS. NOR: Series PMOS, Parallel NMOS.”

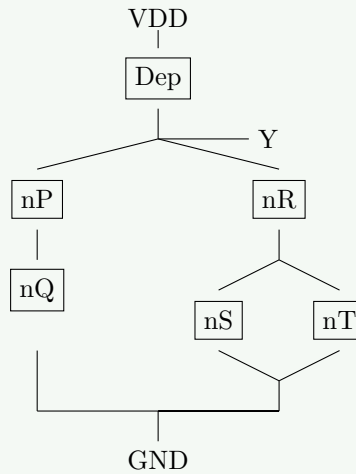
### Question 4(c) OR [7 marks]

Implement  $Y = [PQ + R(S + T)]'$  Boolean equation using depletion load nMOS and CMOS.

## Solution

**Function:**  $Y = \overline{PQ + R(S + T)}$

**1. Depletion Load nMOS:** PDN implements the function without inversion. Structure:  $S$  parallel  $T$ , in series with  $R$ . This block parallel with  $P$  series  $Q$ .



**Figure 11.** Depletion Load Implementation

**2. CMOS Implementation:** PDN is same as above. PUN is dual. **Dual of PDN:** Dual of  $(S + T)$  is  $S \cdot T$ . Dual of  $R \cdot (S + T)$  is  $R + (S \cdot T)$ . Dual of  $P \cdot Q$  is  $P + Q$ . Dual of  $(P \cdot Q) + (R \cdot (S + T))$  is  $(P + Q) \cdot (R + ST)$ . Structure:  $(P||Q)$  in series with  $(R||(S \text{ series } T))$ .

## Question 5(a) [3 marks]

Explain design styles used in Verilog.

## Solution

Verilog Design Styles:

Style	Description	Example
Gate Level	Structural modeling using primitive gates (and, or, not).	<code>and g1(y, a, b);</code>
Data Flow	Describes signal flow using continuous assignment.	<code>assign y = a &amp; b;</code>
Behavioral	Describes functionality using procedural blocks.	<code>always @(*) y = a &amp; b;</code>

**Mnemonic:** Gate-Data-Behavior: Three ways to Model

## Question 5(b) [4 marks]

Write Verilog program for full adder using behavioral modeling.

## Solution

```

1 module full_adder_behavioral (
2     input wire a, b, cin,
3     output reg sum, cout
4 );
5

```

```

6  always @(*) begin
7      case ({a, b, cin})
8          3'b000: {cout, sum} = 2'b00;
9          3'b001: {cout, sum} = 2'b01;
10         3'b010: {cout, sum} = 2'b01;
11         3'b011: {cout, sum} = 2'b10;
12         3'b100: {cout, sum} = 2'b01;
13         3'b101: {cout, sum} = 2'b10;
14         3'b110: {cout, sum} = 2'b10;
15         3'b111: {cout, sum} = 2'b11;
16         default: {cout, sum} = 2'b00;
17     endcase
18 end
19 endmodule

```

### Question 5(c) [7 marks]

Describe the function of CASE statement. Write Verilog code of 3x8 decoder using CASE statement.

#### Solution

**CASE Statement:** Multi-way branching construct. **Features:**

- Compares expression with case items.
- Execute first matching item.
- default item covers unmatched cases.

**3x8 Decoder:**

```

1  module decoder_3x8 (
2      input wire [2:0] sel,
3      input wire en,
4      output reg [7:0] y
5  );
6  always @(*) begin
7      if (en) begin
8          case (sel)
9              3'b000: y = 8'b00000001;
10             3'b001: y = 8'b00000010;
11             3'b010: y = 8'b00000100;
12             3'b011: y = 8'b00001000;
13             3'b100: y = 8'b00010000;
14             3'b101: y = 8'b00100000;
15             3'b110: y = 8'b01000000;
16             3'b111: y = 8'b10000000;
17             default: y = 8'b00000000;
18         endcase
19     end else y = 0;
20 end
21 endmodule

```

### Question 5(a) OR [3 marks]

Write Verilog code to implement 2:1 multiplexer.

## Solution

```

1 // Behavioral
2 module mux21 (input a, b, s, output reg y);
3     always @(*) begin
4         if(s) y = b;
5         else y = a;
6     end
7 endmodule
8
9 // Data Flow
10 module mux21_df (input a, b, s, output y);
11     assign y = s ? b : a;
12 endmodule

```

## Question 5(b) OR [4 marks]

Write Verilog program for D flip-flop using behavioral modeling.

## Solution

```

1 module d_ff (
2     input clk, rst, d,
3     output reg q, qbar
4 );
5 always @(posedge clk or posedge rst) begin
6     if (rst) begin
7         q <= 0;
8         qbar <= 1;
9     end else begin
10        q <= d;
11        qbar <= ~d;
12    end
13 end
14 endmodule

```

## Question 5(c) OR [7 marks]

Explain testbench in brief. Write Verilog code to implement 4-bit down counter.

## Solution

**Testbench:** A module used to verify design functionality by applying stimuli (inputs) and monitoring responses (outputs). It is non-synthesizable.

**4-bit Down Counter:**

```

1 module down_counter (
2     input clk, rst, en,
3     output reg [3:0] count
4 );
5 always @(posedge clk or posedge rst) begin
6     if (rst) count <= 4'b1111;
7     else if (en) count <= count - 1;
8 end

```



```
9  endmodule
```

**Testbench Code:**

```
1  module tb_counter;
2      reg clk, rst, en;
3      wire [3:0] count;
4
5      down_counter dut (clk, rst, en, count);
6
7      always #5 clk = ~clk;
8
9      initial begin
10         clk=0; rst=1; en=0;
11         #10 rst=0; en=1;
12         #200 $finish;
13     end
14
15     initial $monitor("T=%t C=%b", $time, count);
16 endmodule
```