EEE201-HW2

Yukun.Zheng--2251625

➢ **Q1:**

In the following CMOS inverter logic gate, assume the transistor size of $(W/L)_n = 0.8\mu m/0.20\mu m$ and $(W/L)_p = 1.6\mu m/0.20\mu m$ and $V_{DD} = 3.0\,V$.

Sub-questions:

a) **Question:**
Determine the operation modes (e.g. cut-off, linear or saturation) of the NMOS and PMOS transistors respectively at V_{in} = 0.2 V, 0.9 V, 1.5 V, 2.1 V, and 2.8 V. Using a table to list the operation modes is recommended. (Note: You may need to do multiple calculations for the data point V_{in} = 1.5 V with V_{in} = 1.4 V or 1.6 V).

b) **Question:**
Using the tabulated results in (a) or otherwise, determine the current flowing through both the NMOS and PMOS transistors at the different input voltage V_{in} in (a).

c) **Question:**
Using *Matlab* or *Excel*, plot a graph of the current against the input voltage V_{in}.

d) **Question:**
Using the above results in (a) and (b) or otherwise, determine the output voltage V_{out} at the different input voltage V_{in} in (a).

e) **Question:**
Using *Matlab* or *Excel*, plot a graph of the **voltage transfer characteristics** (**VTC**) (i.e. output voltage V_{out} versus input voltage V_{in}) of the CMOS inverter with the five sets of data points obtained.

Answers:

- ***Methodology Overview***

Before presenting the answers to the questions, I will first demonstrate the methodology and thought process for solving Q1.

- ***Problem Statement***

Since $Vout$ is an unknown parameter linked to VDS and VSD, it is not possible to directly analyze typical cases to determine the operation modes and currents of the PMOS and NMOS transistors. Moreover, both transistors are active under a specific Vin value, indicating that the entire network operates in a transitional region between the two logic states (‘0’ and ‘1’).

- ***Analysis and Solution Strategy***

Although we cannot directly find the $Vout$ of one PMOS and one NMOS, what

about if we consider the MOSFET ID-Vout curve, since plotting ID-Vout curve requires VGS and other parameters and can give a plot of $Vout$ versus Id, we can try to plot the NMOS ID-Vout curve under various Vin values, then do the same procedure to get the PMOS ID-Vout curve, finally combine these two together by applying Kirchhoff's Current Law (KCL), which comes $IDp = -IDn$, then we will have the CMOS ID-Vout curve under 5 input voltages. After deriving the CMOS ID-Vout curve, we can derive Vout under different Vin, which means that we have to find the intersection point of NMOS ID-Vout curve and PMOS ID-Vout curve under the same Vin, and the x coordinate of this point is Vout under this specific Vin. Then, we are able to import data and write codes in ***MATLAB*** to determine the operation modes, and draw the relevant graphs for all these sub-questions. And My flowchart for solving this problem is shown in Figure 1, as plotted in ***Visio***.

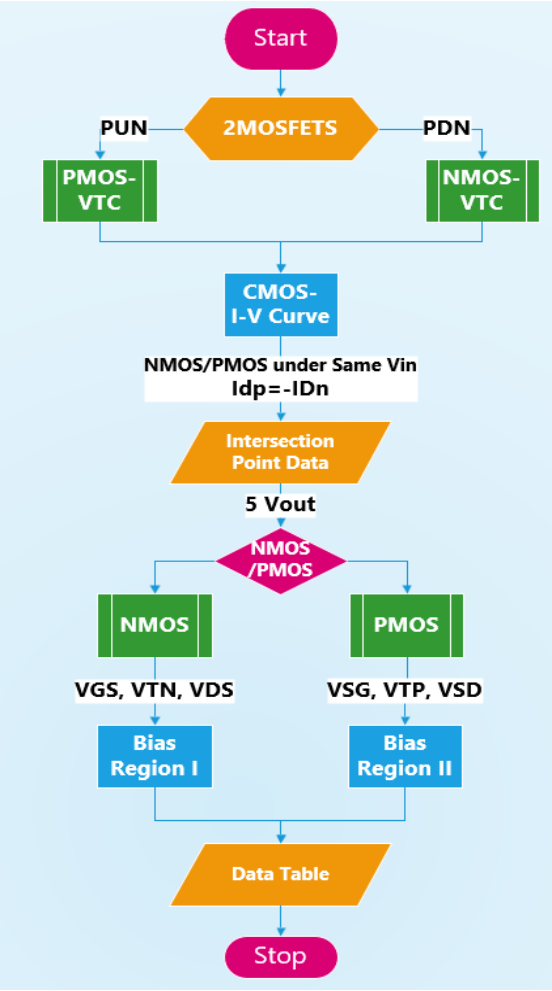


Figure 1. Q1 Flowchart

By following the steps, we have the final CMOS ID-Vout Curve plotted in ***MATLAB*** with intersection points clearly labeled in square shape shown in Figure 2.

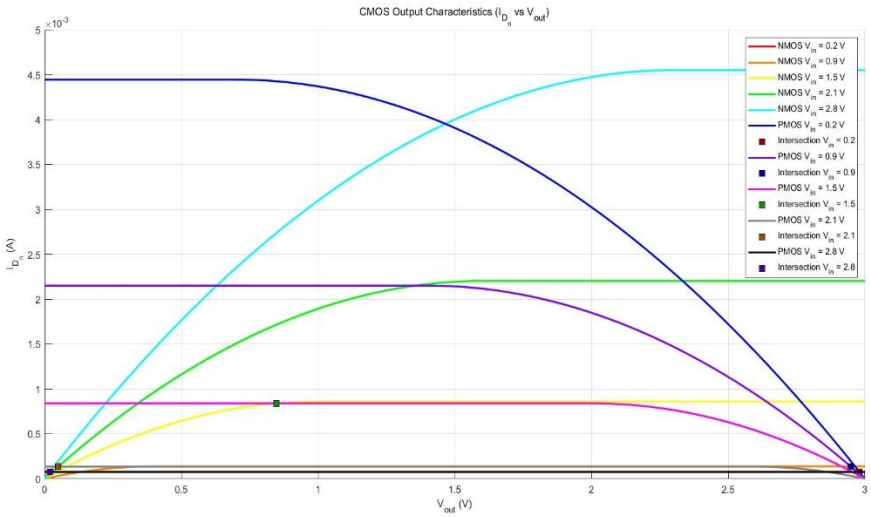


Figure 2: CMOS ID-Vout Curve

This CMOS ID-Vout Curve diagram allows us to solve all sub-questions systematically and effectively (*See codes in appendix*).

- ***Solutions and Explanations***

a) Answer:

We have to separate two cases for PMOS and NMOS to determine their operation modes respectively.

Note that since the CMOS ID-Vout Curve has already been derived, there is no need to perform multiple calculations around Vin=1.4 V,1.5 V, 1.6V (even though the question suggests doing so).

✧ Case 1: PMOS Transistors

For PMOS Transistor, we can determine the operation mode as listed in the table below.

$Vin\ (V)$	0.2	0.9	1.5	2.1	2.8
$Vout\ (V)$	2.9798	2.9479	0.8476	0.0497	0.0192
$VSG\ (V)$	2.8	2.1	1.5	0.9	0.2
$\lvert VSG + VTP\rvert\ (V)$	|2.3|	|1.6|	|1|	|0.4|	|-0.3|
$\lvert VSD\rvert\ (V)$	0.0202	0.0521	2.1524	2.9503	2.9808
$VTP\ (V)$	-0.5	-0.5	-0.5	-0.5	-0.5
Operation Mode	Linear	Linear	Saturation	Saturation	Cut-off

Table 1: PMOS Characteristics

✧ Case 2: NMOS Transistors

For NMOS Transistor, we can determine the operation mode as listed in the table below.

$Vin\ (V)$	0.2	0.9	1.5	2.1	2.8
$Vout\ (V)$	2.9798	2.9479	0.8476	0.0497	0.0192
$VGS\ (V)$	0.2	0.9	1.5	2.1	2.8
$\lvert VGS - VTN\rvert\ (V)$	0.3	0.4	1.0	1.6	2.3
$\lvert VDS\rvert\ (V)$	2.9798	2.9479	0.8476	0.0497	0.0192

$VTN\ (V)$	+0.5	+0.5	+0.5	+0.5	+0.5
Operation Mode	Cut-Off	Saturation	Linear	Linear	Linear

Table 2: NMOS Characteristics

b) Answer:

From (a), we have already found the relevant voltage values under different inputs and corresponding operation mode. The next step is to write code to calculate the various currents under different inputs (*See codes in appendix*). The results are displayed in the following tables.

✧ Case 1: PMOS Currents

For PMOS Transistor, we can determine the operation mode as listed in the table below.

$Vin\ (V)$	0.2	0.9	1.5	2.1	2.8
$Vout\ (V)$	2.9798	2.9479	0.8476	0.0497	0.0192
$IDp\ (mA)$	-0.0777	-0.1378	-0.8400	-0.1344	0.0000

Table 3: PMOS Currents under Different Inputs

✧ Case 2: NMOS Currents

For NMOS Transistor, we can determine the operation mode as listed in the table below.

$Vin\ (V)$	0.2	0.9	1.5	2.1	2.8
$Vout\ (V)$	2.9798	2.9479	0.8476	0.0497	0.0192
$IDn\ (mA)$	0.0000	0.1376	0.8400	0.1347	0.0756

✧ Table 4: NMOS Currents under Different Inputs

c) Answer:

Utilizing the data in (b), we can write ***MATLAB*** code to generate a graph of current versus input voltages, as shown in Figure 3.

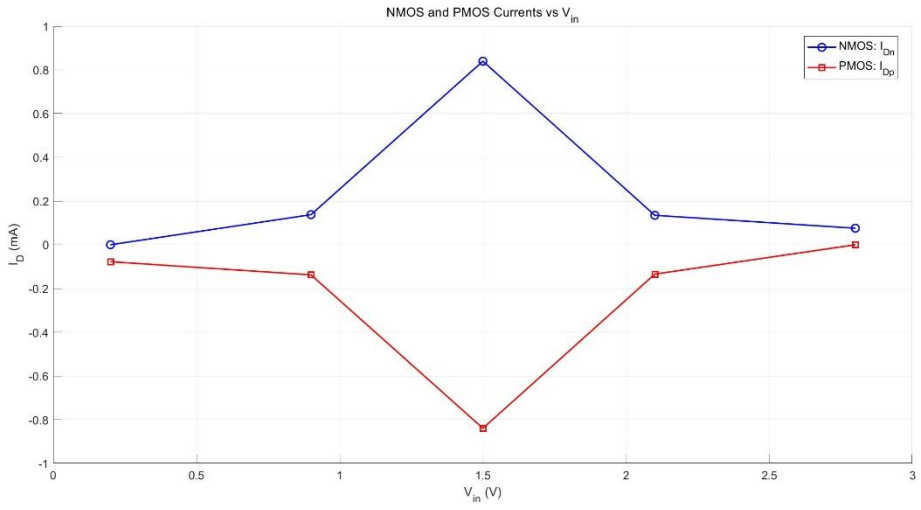


Figure 3: Diagram of Current ID versus Vin

d) Answer:

Utilizing the data in both (a) and (b), we can generate a table of Vin versus Voyt as shown in table 5.

$Vin\ (V)$	0.2	0.9	1.5	2.1	2.8
$Vout\ (V)$	2.9798	2.9479	0.8476	0.0497	0.0192

Table 5: Table of Vout versus Vin

e) Answer:

Utilizing the data in both (a) and (b), we can write ***MATLAB*** code to generate a graph of current versus input voltages, as shown in Figure 4.

Note that I choose Vin as the horizontal axis, whose graph may be different from that choose Vout as x-axis.

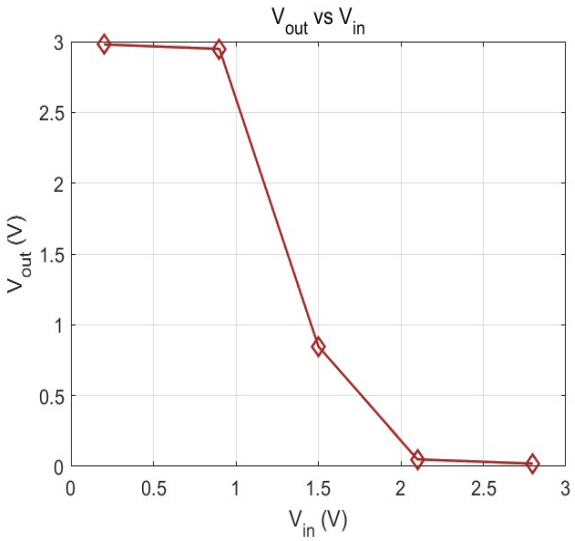
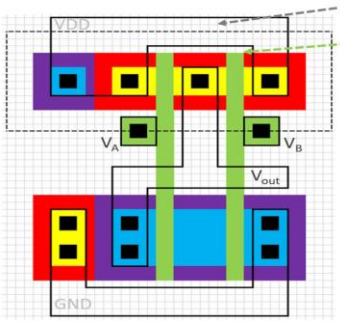


Figure 4: Diagram of CMOS VTC

➢ **Q2:**

The drawn mask layout of a CMOS logic gate is shown below. Assume a p-type silicon substrate.



Sub-questions:

a) **Question:**

Draw the schematic diagram of the CMOS transistor circuit using standard circuit symbols. Make sure that all the terminals of the transistors and the circuits are clearly labeled.

b) **Question:**

Using the scalable CMOS layout design rules or also called λ layout design rules, determine the transistor size (i.e. W/L) of each transistor of the logic circuit.

Answers:

a) **Answer:**

Since two transistors in the PUN shares the same terminal to serve an output, this indicates that PUN contains two PMOS transistors in parallel. Following the CMOS complimentary rule, so the PDN contains two NMOS transistors in series, and we can draw this circuit (CMOS NAND Gate Circuit) with clear terminal labels as shown in Figure 5.

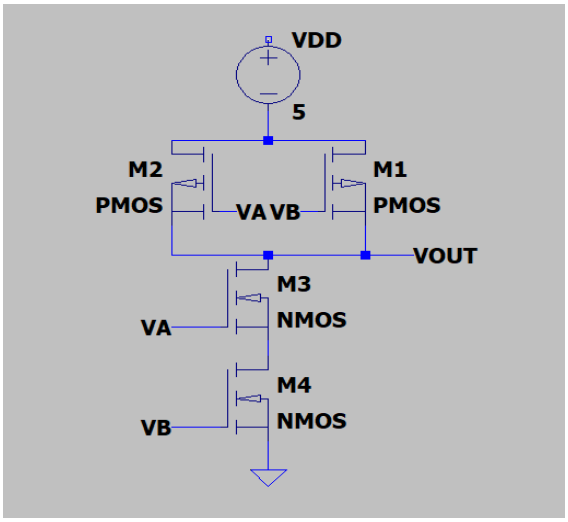


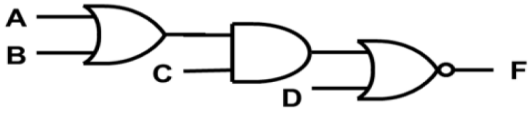
Figure 5: CMOS NAND Gate Circuit Schematic

b) **Answer:**

Based on the layout rule, the gate length is always 2λ, which corresponds to 2 grid units in the given layout diagram (with each grid unit representing 1λ). The shallow blue area in the PDN represents its active region, which has a width of 8λ. Similarly, the yellow area represents the active region of the PUN, which has a width of 4λ. Therefore, we can conclude that the width ratio of the PMOS to the NMOS is $Wp: Wn = 1: 2$, given that both have the same gate length.

➢ **Q3:**

The logic gate schematic of a digital circuit is shown below.



Sub-questions:

a) **Question:**

Draw the schematic diagram of the logic circuit of CMOS implementation (i.e. using PMOS and NMOS transistors in the complementary way to minimize the static power dissipation) with the smallest possible number of transistors. You must use standard circuit symbols

Answers:

a) Answer:

First, according to the digital logic circuit, we can derive its function representation as:

$$F = \overline{(A + B) \cdot C + D} \tag{1}$$

Which corresponds to a NAND Gate formula.

Note that this expression is the most efficient, as it minimizes transistor usage, requiring only 8 transistors in total. However, if we try to utilize Boolean Algebra to further simplify the expressions to:

$$F = \overline{(A + B + D) \cdot (C + D)} \tag{2}$$

The, it will take 10 transistors in total, which fails to meet the goal of the least number of transistors. Then we take:

$$F = \overline{(A + B) \cdot C + D} \tag{3}$$

draw the circuit by combinational logic in CMOS implantation as shown in Figure 6.

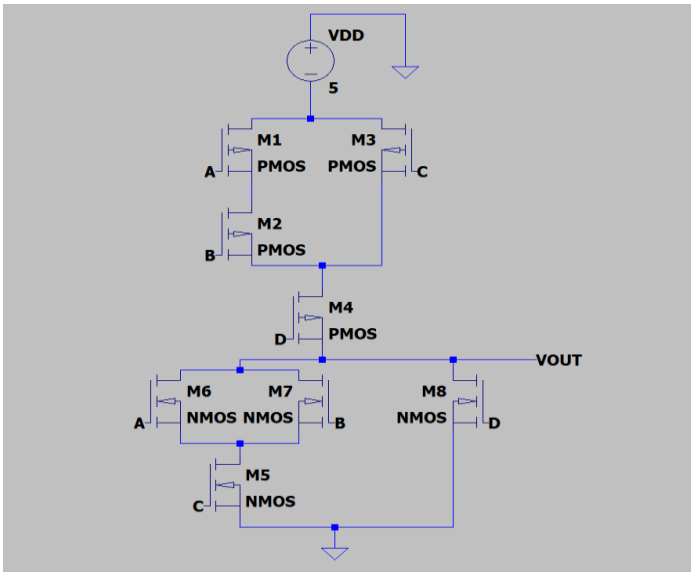
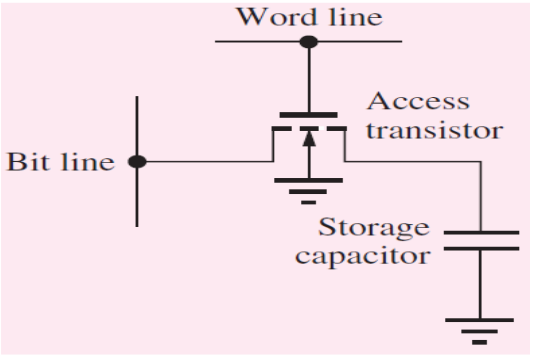
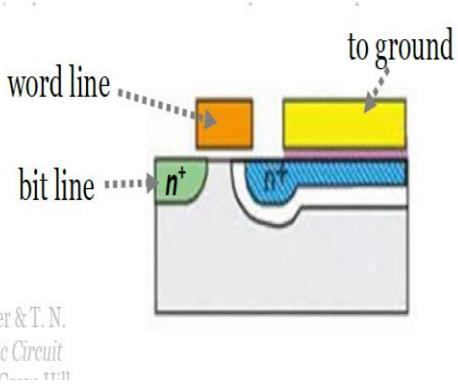


Figure 6. Combination CMOS Logic Circuit Schematic

➢ **Q4:**

The schematic circuit diagram (on the left) and cross-sectional structure (on the right) of a **dynamic random-access memory** (**DRAM**) unit cell is shown below.





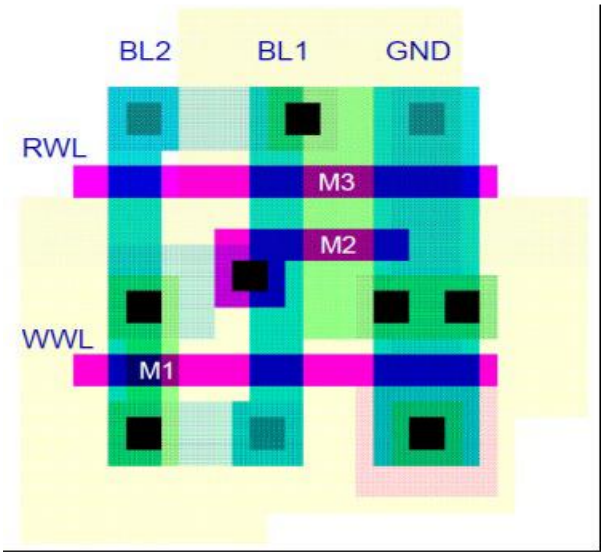
Sub-questions:

a) **Question:**

If the **storage capacitor** is implemented using a **MOS capacitor** (as shown in the cross-sectional structure above) of the smallest possible size according to the λ-layout design rules for an n-channel MOS transistor, sketch the layout of the **DRAM cell**.

b) **Question:**

Instead of using only one transistor for the implementation of a DRAM cell, three transistors can be used for an improved implementation. The IC layout of a three-transistor (3T) DRAM cell is shown below. Sketch the schematic circuit diagram of such a 3T DRAM cell. Proper labels must be included. Hint: The 3T DRAM cell is similar to the 1T implementation but with additional transistors and signal connections.



Answers:

a) Answer:

If we assume that the storage capacitor of a DRAM cell is implemented as a MOS capacitor, we can use the NMOS design rules to determine the layout of a single DRAM cell. Based on these rules, the layout of one DRAM cell is illustrated in Figure 7, with the design parameters clearly defined in terms of λ.

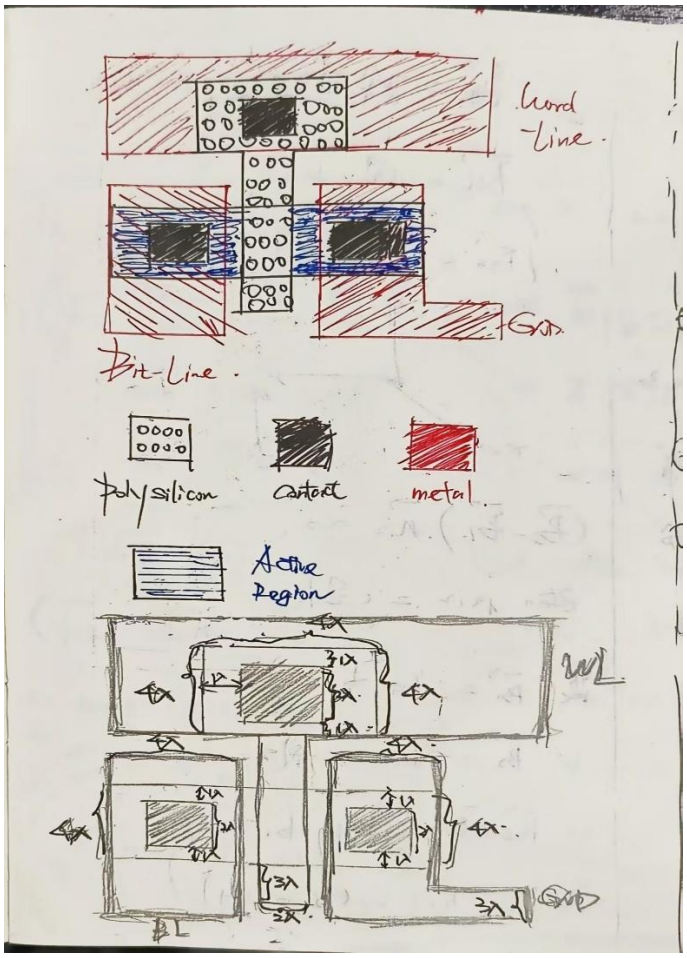


Figure 7. DRAM Cell Layout

b) Answer:

By examining the IC layout, we observe that there are two bitlines and two wordlines connected to three transistors. For transistor M1, its gate terminal is connected to the WWL (Write Wordline), while for M3, the gate terminal is connected to the RWL (Read Wordline).

Next, we need to determine the position and connections of M2. Since M3 cannot have either one source or one drain, the absence of one indicates that it may not be directly connected to a voltage source. This observation suggests that the missing source or drain of M3 must be directly connected to either the source or drain of M2. Based on this reasoning, the contact inside the gate region of M2 is connected to a capacitor, as the capacitor is responsible for storing the charge

transferred via this contact. The remaining terminal of M2 (either source or drain) is grounded to complete the circuit. This deduction aligns with the functional requirements of the design and the observed layout structure with my speculated circuit topology presented in figure 8.

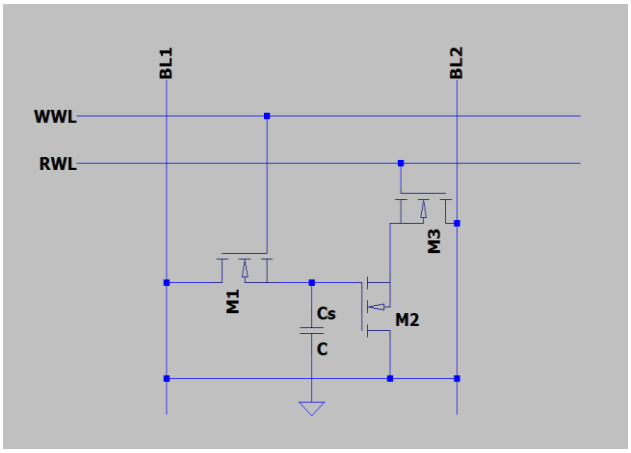


Figure 8. Deducted 3T-DRAM Cell Circuit

➢ Appendix

✧ Q1-Overall: CMOS ID-Vout Curve

```
% NMOS Parameters
mu_n_Cox = 430e-6;  % Mobility * oxide capacitance
Wn_Ln = 4;  % Width-to-length ratio (W/L)
V_T0n = 0.50;  % Threshold voltage (V)

% PMOS Parameters
mu_p_Cox = 210e-6;  % Mobility * oxide capacitance
Wp_Lp = 8;  % Width-to-length ratio (W/L)
V_T0p = -0.50;  % Threshold voltage (V)

% Define VDS range for both NMOS and PMOS
VDS_n = linspace(0, 3, 100);  % VDS range for NMOS (positive)
VDS_p = linspace(-3, 0, 100);  % VDS range for PMOS (negative)

% Define different Vin values for both NMOS and PMOS
Vin_values = [0.2, 0.9, 1.5, 2.1, 2.8];

% Manually define 10 distinct colors
colors = [
    [1, 0, 0];  % Red
    [1, 0.5, 0];  % Orange
    [1, 1, 0];  % Yellow
    [0, 1, 0];  % Green
    [0, 1, 1];  % Cyan
    [0, 0, 1];  % Blue
    [0.5, 0, 1];  % Purple
    [1, 0, 1];  % Magenta
    [0.5, 0.5, 0.5];  % Gray
    [0, 0, 0]  % Black
];

% Define deep colors for intersection points
deep_colors = [
    [0.6, 0, 0];    % Dark red
    [0, 0, 0.6];    % Dark blue
    [0, 0.6, 0];    % Dark green
    [0.6, 0.3, 0];  % Dark orange
    [0.3, 0, 0.6];  % Dark purple
];

% Start plotting
figure;
```

```
hold on;

% Initialize storage for NMOS data
IDS_n_all = cell(1, length(Vin_values));
VDS_n_all = cell(1, length(Vin_values));

% Plot NMOS curves
for idx = 1:length(Vin_values)
    VGS = Vin_values(idx);
    IDS_n = zeros(size(VDS_n));  % Initialize IDS array for NMOS
    for i = 1:length(VDS_n)
        if VDS_n(i) < VGS - V_T0n
            % Linear region equation for NMOS
            IDS_n(i) = mu_n_Cox * Wn_Ln * ((VGS - V_T0n) * VDS_n(i) -
0.5 * VDS_n(i)^2);
        else
            % Saturation region equation for NMOS
            IDS_n(i) = 0.5 * mu_n_Cox * Wn_Ln * (VGS - V_T0n)^2;
        end
    end
    % Plot NMOS curve with Vout = VDS for NMOS
    plot(VDS_n, IDS_n, 'LineWidth', 2, 'Color', colors(idx, :),
'DisplayName', ['NMOS V_{in} = ' num2str(VGS) ' V']);

    % Store NMOS data for intersection calculation
    IDS_n_all{idx} = IDS_n;
    VDS_n_all{idx} = VDS_n;
end

% Plot PMOS curves and calculate intersections
for idx = 1:length(Vin_values)
    VGS_p = Vin_values(idx);
    IDS_p = zeros(size(VDS_p));  % Initialize IDS array for PMOS
    for i = 1:length(VDS_p)
        VSG_p = VGS_p - 3;  % VSG for PMOS
        if VDS_p(i) > VSG_p - V_T0p
            % Linear region equation for PMOS
            IDS_p(i) = -mu_p_Cox * Wp_Lp * ((VSG_p - V_T0p) * VDS_p(i) -
0.5 * VDS_p(i)^2);
        else
            % Saturation region equation for PMOS
            IDS_p(i) = -0.5 * mu_p_Cox * Wp_Lp * (VSG_p - V_T0p)^2;
        end
    end
```

```
    % Calculate Vout = VDS + 3 for PMOS
    Vout_p = VDS_p + 3;

    % Plot PMOS curve with Vout on x-axis and -IDS on y-axis
    plot(Vout_p, -IDS_p, 'LineWidth', 2, 'Color', colors(idx + 5, :),
'DisplayName', ['PMOS V_{in} = ' num2str(VGS_p) ' V']);

    % Calculate intersections using interpolation
    [common_Vout, common_ID] = polyxpoly(VDS_n_all{idx},
IDS_n_all{idx}, Vout_p, -IDS_p);
    if ~isempty(common_Vout)
        % Choose a deep color for the current intersection
        current_color = deep_colors(mod(idx - 1, size(deep_colors, 1)) +
1, :);

        % Plot intersection point with a specific color
        plot(common_Vout, common_ID, 's', 'MarkerSize', 8,
'MarkerFaceColor', current_color, ...
            'MarkerEdgeColor', 'k', 'DisplayName', ['Intersection V_{in}
= ' num2str(VGS_p)]);
    end
end

% Set plot attributes
xlabel('V_{out} (V)');  % X-axis label
ylabel('I_{D_n} (A)');  % Y-axis label
title('CMOS Output Characteristics (I_{D_n} vs V_{out})');  % Title
legend('show');  % Show legend
grid on;  % Enable grid
hold off;
```

✧ Q1(b): Data of ID versus Vin

```
% Given parameters
mu_n_Cox = 430e-6;  % μ_n * C_ox in A/V^2
Wn_Ln = 4;          % W/L ratio
Vin = [0.2, 0.9, 1.5, 2.1, 2.8];  % Input voltages (V)
Vout = [2.9798, 2.9479, 0.8476, 0.0497, 0.0192];  % Output voltages (V)
VGS = [0.2, 0.9, 1.5, 2.1, 2.8];  % Gate-source voltages (V)
VTN = 0.5;  % Threshold voltage (V)
modes = ["Cut-Off", "Saturation", "Linear", "Linear", "Linear"];  %
Modes

% Preallocate array for ID
ID = zeros(size(Vin));
```

```
% Calculate ID based on modes
for i = 1:length(Vin)
    VDS = Vout(i);
    VGS_minus_VTN = VGS(i) - VTN;

    switch modes(i)
        case "Cut-Off"
            % In Cut-Off, ID = 0
            ID(i) = 0;

        case "Saturation"
            % In Saturation region
            ID(i) = 0.5 * mu_n_Cox * Wn_Ln * VGS_minus_VTN^2;

        case "Linear"
            % In Linear region
            ID(i) = mu_n_Cox * Wn_Ln * (VGS_minus_VTN * VDS - 0.5 *
VDS^2);
    end
end

% Display results with formatted output
for i = 1:length(Vin)
    fprintf('When Vin = %.1f V, IDn is %.4f mA\n', Vin(i), ID(i) *
1e3);
end



% PMOS Parameters
mu_p_Cox = 210e-6;  % μ_p * C_ox in A/V^2
Wp_Lp = 8;          % W/L ratio
V_T0p = -0.50;      % Threshold voltage (V)

% Given data
Vin = [0.2, 0.9, 1.5, 2.1, 2.8];  % Input voltages (V)
Vout = [2.9798, 2.9479, 0.8476, 0.0497, 0.0192];  % Output voltages (V)
VSG = [2.8, 2.1, 1.5, 0.9, 0.2];  % Source-gate voltages (V)
VTP = -0.5;  % PMOS threshold voltage (V)
VSD = [0.0202, 0.0521, 2.1524, 2.9503, 2.9808];  % Source-drain
voltages (V)
modes = ["Linear", "Linear", "Saturation", "Saturation", "Cut-off"];  %
Modes
```

```
% Preallocate array for IDp
IDp = zeros(size(Vin));

% Calculate IDp based on modes
for i = 1:length(Vin)
    VSG_minus_VTP = abs(VSG(i) + VTP);  % |VSG + VTP|

    switch modes(i)
        case "Cut-off"
            % In Cut-off, IDp = 0
            IDp(i) = 0;

        case "Saturation"
            % In Saturation region
            IDp(i) = -0.5 * mu_p_Cox * Wp_Lp * VSG_minus_VTP^2;

        case "Linear"
            % In Linear region
            IDp(i) = -mu_p_Cox * Wp_Lp * (VSG_minus_VTP * VSD(i) - 0.5 *
VSD(i)^2);
    end
end

% Display results with formatted output
for i = 1:length(Vin)
    fprintf('When Vin = %.1f V, IDp is %.4f mA\n', Vin(i), IDp(i) *
1e3);
end
```

✧ Q1(c): Graph of ID versus Vin

```
% Data
Vin = [0.2, 0.9, 1.5, 2.1, 2.8]; % Input voltage (V)
IDn = [0.0000, 0.1376, 0.8400, 0.1347, 0.0756]; % NMOS current (mA)
IDp = [-0.0777, -0.1378, -0.8400, -0.1344, 0.0000]; % PMOS current (mA)

% Plotting
figure;
hold on; % Enable overlay of plots
plot(Vin, IDn, '-o', 'LineWidth', 1.5, 'MarkerSize', 8, 'Color', 'b',
'DisplayName', 'NMOS: I_{Dn}'); % NMOS curve
plot(Vin, IDp, '-s', 'LineWidth', 1.5, 'MarkerSize', 8, 'Color', 'r',
'DisplayName', 'PMOS: I_{Dp}'); % PMOS curve
hold off; % Disable overlay of plots
```

```
% Add grid and labels
grid on;
xlabel('V_{in} (V)', 'FontSize', 12); % X-axis label
ylabel('I_D (mA)', 'FontSize', 12); % Y-axis label
title('NMOS and PMOS Currents vs V_{in}', 'FontSize', 14); % Chart
title
legend('Location', 'best', 'FontSize', 12); % Add legend
set(gca, 'FontSize', 12); % Set font size for axes
```

✧ Q1(e): Graph of Vout versus Vin

```
% Data
Vin = [0.2, 0.9, 1.5, 2.1, 2.8]; % Input voltage (V)
Vout = [2.9798, 2.9479, 0.8476, 0.0497, 0.0192]; % Output voltage (V)

% Plotting
figure;
plot(Vin, Vout, '-d', 'LineWidth', 1.5, 'MarkerSize', 8, 'Color',
[0.65, 0.16, 0.16]); % Brown curve
grid on; % Add grid
xlabel('V_{in} (V)', 'FontSize', 12); % X-axis label
ylabel('V_{out} (V)', 'FontSize', 12); % Y-axis label
title('V_{out} vs V_{in}', 'FontSize', 14); % Chart title
set(gca, 'FontSize', 12); % Set font size for axes
```