Problem 1 - Charge Sharing and Leakage

A dynamic logic gate is shown above. During the pre-charge phase ($\phi = 0$) output $F$ is charged to $V_{DD}$ and it is given that $A = B = C = D = 0V$. During the evaluate phase ($\phi = 0$), any of these inputs may or may not change.

a) What is the worst-case combination of input transitions in terms of charge sharing? That is, what input combination will corrupt $F$ the most when the output is supposed to remain high?

b) For the worst-case identified in part (a), what is the final voltage at node $F$? Use $V_{DD} = 1.5V$, NMOS model $V_{TH0} = 0.5V$, $\gamma = 0.4V$, $2\phi_F = 0.6V$, assume that $C_{db} = C_{sb} = 1fF$ for all transistors and ignore gate capacitance.

c) In addition to the charge sharing calculated in part (b), now assume that there is also a constant leakage current of 2nA discharging node $F$ to ground. What is that minimum clock speed that this gate can operate at if the next gate has $V_{IH} = 1.0V$? Assume that clock $\phi$ is high and low for equal periods of time.

d) Repeat part (b) in the case where $C_L = 0fF$. 

Problem 2 - Pass Transistor Logic

The function \( F = A \text{ XOR } B \) is to be implemented in pass transistor logic according to the diagram below.

![Pass Transistor Network](image)

a) How would you implement the pass transistor logic with NMOS-only switches? Assume both true and complimentary input signals are available.

b) What is the minimum voltage at which this circuit will operate correctly (and why)? For the NMOS switches use \( V_{TH0} = 0.5V \), \( \gamma = 0.4V \), \( 2\phi_F = 0.6V \). Assume that the inverter has an ideal VTC that switches when its input is at \( V_{DD}/2 \).

c) A level restoring PMOS transistor is now added as shown in the figure below. What are the benefits and drawbacks of this modification?

![Pass Transistor Network with level restoring PMOS](image)

d) If the level restoring PMOS transistor has \( V_{TH0} = -0.5V \) and \( W/L = 4 \) and \( \mu_N = 3\mu_P \), what is the minimum \( W/L \) for the NMOS switches in order for the circuit to function properly?
Problem 3 – Adders

A neat four-bit adder circuit is shown below. The switch arrows are multiplexers and the arrows are drawn in the position they take when their controls are at logical 0. They switch to the left when the control is at logical 1.

Using a pass-transistor implementation, the circuit schematic for each adder cell can be:

a) Fill in the following truth tables for \( S^0 \), \( S^1 \), \( C^0 \), and \( C^1 \) for each cell. In the last row indicate the logical function (the boolean expression) for each of these outputs.

<table>
<thead>
<tr>
<th>( A )</th>
<th>( B )</th>
<th>( S^0 )</th>
<th>( S^1 )</th>
<th>( C^0 )</th>
<th>( C^1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Logical Function:

b) Explain how this adder works. First, what are the logical interpretations of \( S^0 \) and \( S^1 \), and \( C^0 \) and \( C^1 \)? Then describe the purpose of the multiplexers and the general operation of this adder.
c) Assuming that inputs to each cell propagate to the outputs $S^0$, $S^1$, $C^0$, and $C^1$ in fixed time $t_{cell}$ and that signals propagate through the multiplexers (switches) in fixed time $t_{mux}$ after the switch is set, what is the overall propagation delay to $C_{out}$ and $S_3$. What would the delay be for a 16-bit adder based on this architecture?

d) On the job as a digital designer you think about this circuit for a while and discover a clever way to speed up the circuit, shown below. Assuming the same delays as in part (b), what is the worst case delay to $C_{out}$ and $S^3$ for this circuit?

e) Assume that a carry-in signal is added to the first stage (which increases the delay of the circuit by $t_{mux}$, it is possible to chain four of these circuits together to form a 16-bit adder. What would the worst case delay be for this 16-bit adder? Is there a faster way to form a 16-bit adder using these blocks? No need to draw it, just describe the idea.