

Chapter 2.

[2.1] The nFET can pass any voltage in the range $[0, V_{max}]$ where $V_{max} = (V_G - V_{Th})$ with V_G the voltage on the gate. With the stated values, $V_{max} = 5 - 0.7 = 4.3$ V. If $V_{in} > V_{max}$ then V_{out} is restricted to V_{max} . However, the nFET passes any voltage $V_{in} < V_{max}$. This gives the following answers.

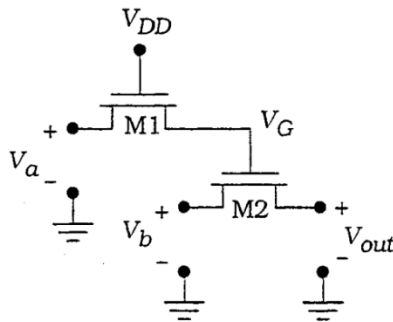
- (a) $V_{in} = 2$ V, $V_{out} = 2$ V;
- (b) $V_{in} = 4.5$ V, $V_{out} = 4.3$ V is limited;
- (c) $V_{in} = 3.5$ V, $V_{out} = 3.5$ V;
- (d) $V_{in} = 0.7$ V, $V_{out} = 0.7$ V.

The main idea is to show the effect of the threshold loss through an nFET.

[2.2] For $V_{in} < V_{max}$, then the input voltage is transmitted through the chain. If $V_{in} > V_{max}$, then a threshold drop occurs in the first transistor (only) and V_{max} makes it to the output. With the stated values, $V_{max} = 3.3 - 0.55 = 2.75$ V This gives the following answers.

- (a) $V_{in} = 2.9$ V, $V_{out} = 2.75$ V (limited);
- (b) $V_{in} = 3.0$ V, $V_{out} = 2.75$ V (limited);
- (c) $V_{in} = 1.4$ V, $V_{out} = 1.4$ V;
- (d) $V_{in} = 3.1$ V, $V_{out} = 2.75$ V (limited).

[2.3] The output of the upper FET M1 (with V_a applied) is used to control the gate voltage V_G of the lower transistor M2 (with V_a applied). Both are susceptible to threshold voltage



Problem [2.3]

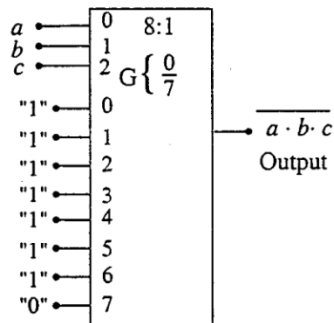
2

drops so that $\max(V_G) = (V_{DD} - V_{Th})$ and $\max(V_{out}) = (V_G - V_{Th})$. Using $\max(V_G) = (3.3 - 0.6) = 2.7$ V gives the following results.

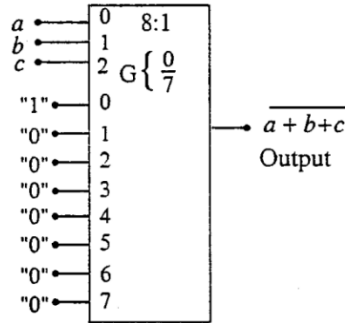
- (a) $V_a = 3.3$ V, $V_b = 3.3$ V: $V_G = 2.7$ V so $V_{out} = 2.7 - 0.6 = 2.1$ V.
- (b) $V_a = 0.5$ V, $V_b = 3$ V: $V_G = 0.5$ V so M2 is in cutoff. This makes V_{out} an unknown value since the transistor is an open circuit.
- (c) $V_a = 2$ V, $V_b = 2.5$ V: $V_G = 2$ V so $V_{out} = 2 - 0.6 = 1.4$ V.
- (d) $V_a = 3.3$ V, $V_b = 1.8$ V: $V_G = 2.7$ V so $V_{out} = 1.8$ V.

[2.4] NAND3 gate using an 8:1 MUX is shown in drawing below with Prob. [2.5] solution.

[2.5] NOR3 gate using an 8:1 MUX is shown in drawing below with Prob. [2.4] solution.



Problem [2.4] NAND 3

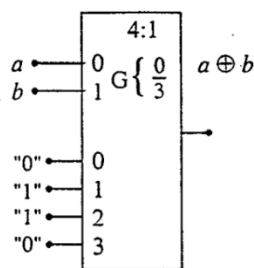


Problem [2.5] NOR 3

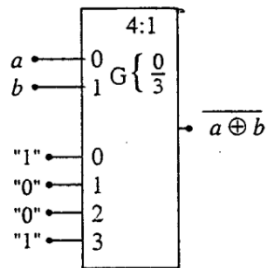
[2.6] The drawings below illustrate the XOR2 and XNOR2 MUX-based designs. To implement the full-adder sum expression, we use

$$s = (a \oplus b) \oplus c$$

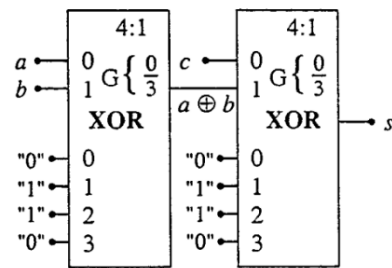
which shows that s can be calculated using 2 XOR gates. This is shown in Figure (c) below.



(a) XOR2



(b) XNOR2



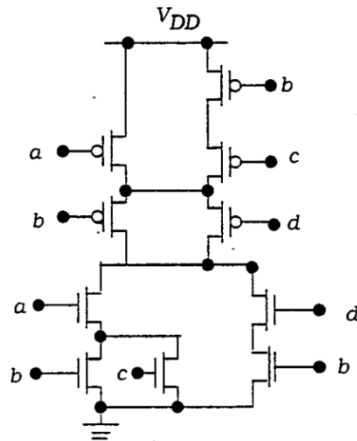
(c) sum

Problem [2.6]

[2.7] Rewrite the function to read

$$f = \overline{a \cdot (b+c) + b \cdot d}$$

to show that the function can be reduced to a simplified form. The gate is shown in the drawing.



Problem [2.7]

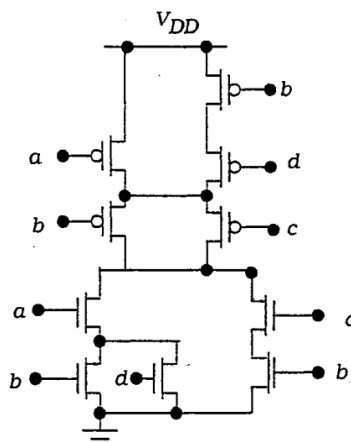
[2.8] The final design depends on the algebraic form selected. One approach is to first expand the terms as

$$\begin{aligned} (a+b) \cdot (a+c) &= a + a \cdot b + a \cdot c + b \cdot c \\ &= a \cdot (1+b) + a \cdot c + b \cdot c \\ &= a \cdot (1+c) + b \cdot c \\ &= a + b \cdot c \end{aligned}$$

Then

$$\begin{aligned} \bar{h} &= (a+b \cdot c) \cdot (b+d) \\ &= a \cdot b + b \cdot c + a \cdot d + b \cdot d \cdot c \\ &= a \cdot (b+d) + b \cdot c(1+d) \\ &= a \cdot (b+d) + b \cdot c \end{aligned}$$

This form of the logic function gives the AOI gate shown. Note that there are variations possible.

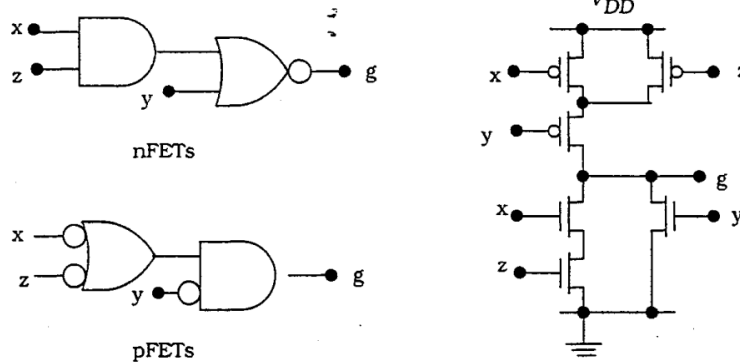


Problem [2.8]

[2.9] Write

$$\bar{g} = x \cdot (y + z) + y = x \cdot y + x \cdot z + y = y \cdot (1 + x) + x \cdot z = y + x \cdot z$$

This gives the simplified logic diagrams, which are then used to design the logic gate as shown.

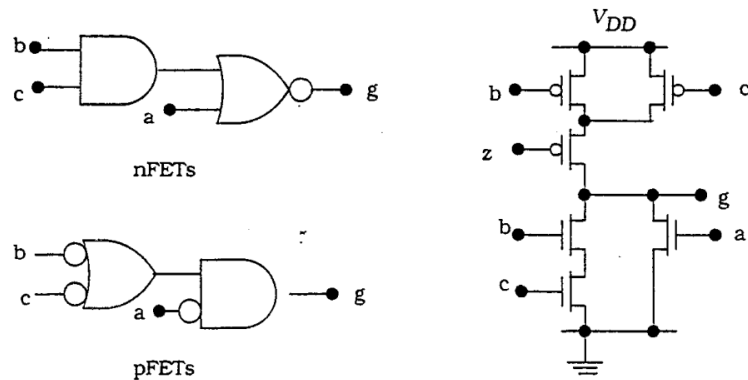


Problem [2.9]

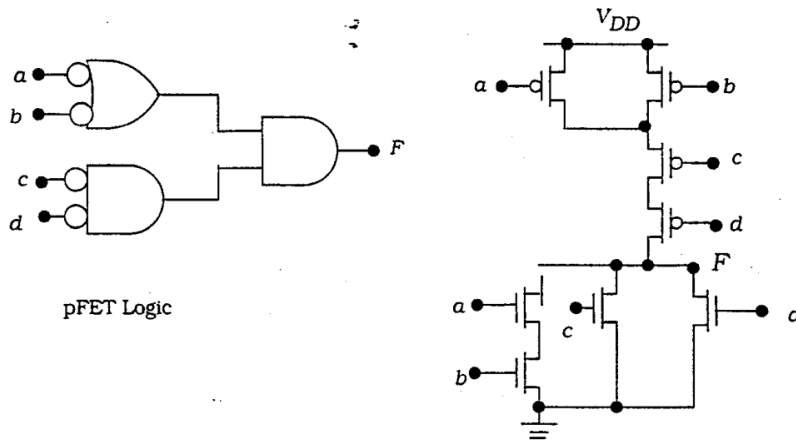
[2.10] Write

$$\bar{F} = a + b \cdot c + a \cdot b \cdot c = a \cdot (1 + b \cdot c) + b \cdot c = a + bc$$

After reducing we see that this is the same circuit as for Problem [2.4] with the inputs relabeled. This is shown for completeness

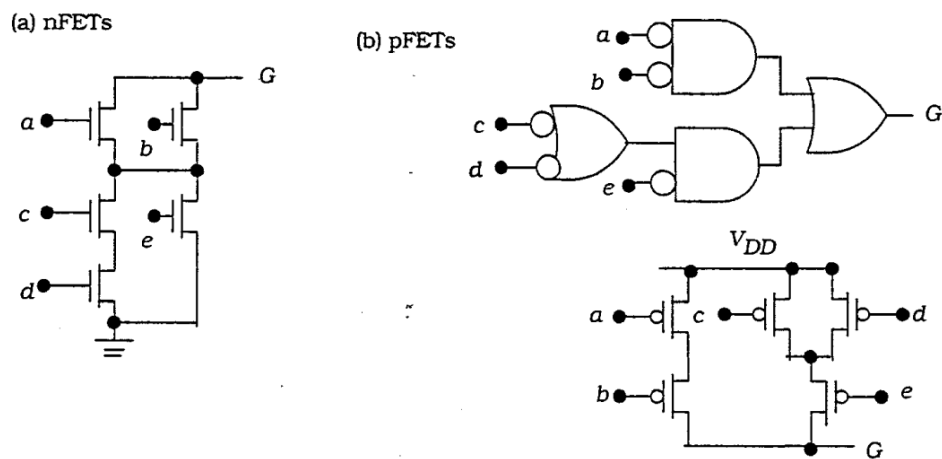


[2.11] The pFET logic diagram and resulting gate are shown.



Problem [2.11]

[2.12] Solution is shown

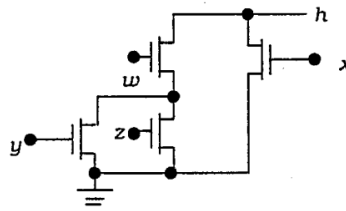


Problem [2.12]

[2.13] The logic expression is

$$\begin{aligned}
 h &= \bar{x} \cdot (\bar{y} \cdot \bar{z} + \bar{w}) \\
 &\approx \bar{x} \cdot \overline{(y + z + w)} \\
 &= \bar{x} \cdot \overline{(y + z)} \cdot \bar{w} \\
 &= \overline{x + (y + z)} \cdot \bar{w}
 \end{aligned}$$

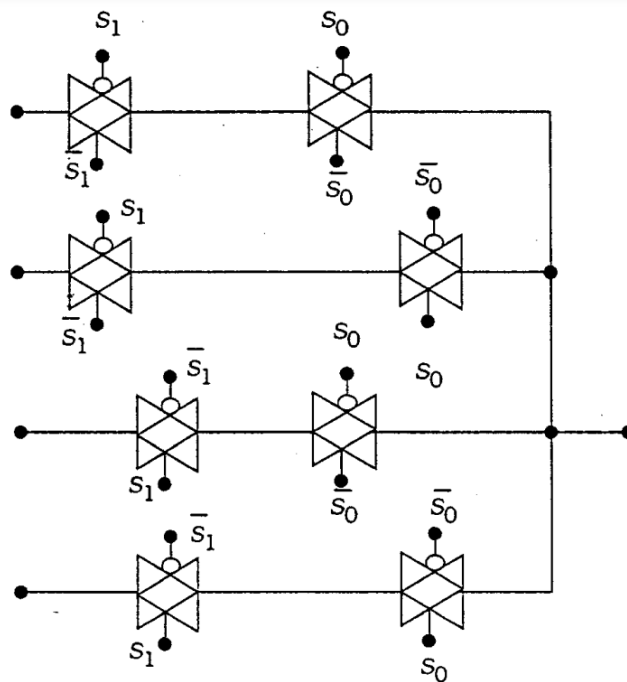
This leads to the nFET array shown. The circuit can be checked using series-parallel structuring.



Problem [2.13]

This example shows how the pFET logic equations can be used to describe a pFET network. The relationship to nFET equations is through the DeMorgan relations.

[2.14] The 4:1 circuit directly illustrates how TGs are used in switching.



Problem [2.14]

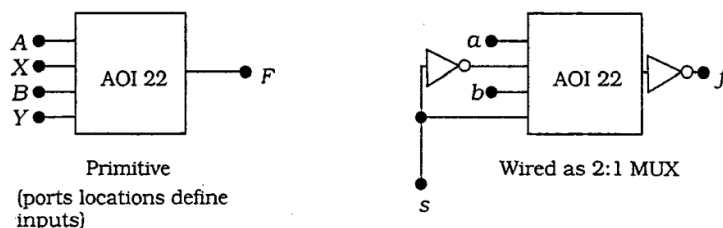
[2.15] The logic equation can be written as

$$f = a \cdot \bar{s} + b \cdot s$$

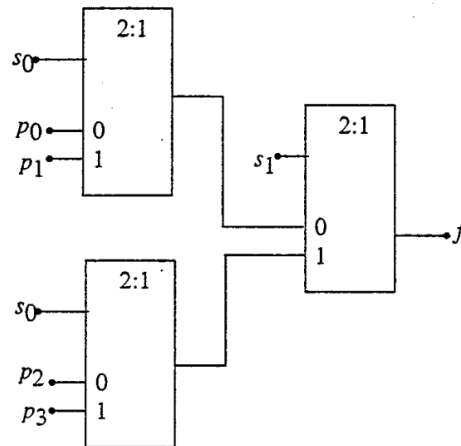
where a and b are the inputs, and s , \bar{s} are the controls. An AOI22 gate would produce an output of

$$F = \overline{A \cdot X + B \cdot Y}$$

so we assign $a = A$, $b = B$, $X = \bar{s}$, and $Y = s$ and add an inverter. It is important to remember that the location of the ports define how the output function is formed.



[2.16] This can be designed by using two input 2:1 MUXes that are controlled by s_0 , and the MUXing the outputs by a 2:1 that is controlled by s_1 .



Problem [2.16]

[2.17] The period is

$$T = \frac{1}{f} = \frac{1}{2.1 \times 10^9} = 476 \text{ ps}$$

Why such an easy problem? To illustrate the time frame that logic gates in a high-speed digital system must operate. As we will see in Part 2, fast circuits (short logic delays) can be difficult to design.

[2.18] The smallest clock frequency is

$$f_{min} = \frac{1}{2t_{hold}} = \frac{1}{2(0.120)} = 4 \text{ Hz}$$

While this is very slow, it does show that the clock cannot be idled.