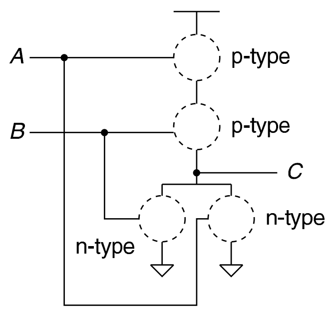
Intro. to Computer Architecture	**Homework 2 Solution**
CSE 240 Autumn 2004	**DUE: Fri. 24 September 2004**

Write your answers on these pages. Additional pages may be attached (with staple) if necessary. Please ensure that your answers are legible. Please show your work. Due at the ***beginning of class***. **Hints are provided for questions marked with on the last page of the assignment.** Total points: 48.

1. [4 Points] **Transistors.** In the circuit given below, the transistors are shown as circles (just like in the circuits in Figures 3.4 and 3.5 of the textbook). In this problem you must determine whether or not each of these transistors conduct or not based on the inputs (*A* and *B*).



For all values (*i.e.*, 0 or 1) of inputs *A* and *B* indicate in the table below whether each transistor conducts (*i.e.*, act like a piece of wire, denoted "1") or does not conduct (*i.e.*, acts like an open circuit, denoted "0"). Also indicate the value (*i.e.*, 0 or 1) of the output *C*. (In the table, the upper and lower p-type transistors are called p-up and p-down, respectively, while the left and right n-type transistors are called n-left and n-right, respectively.)

Answer:

A	B	C	P-up	P-down	N-left	N-right
0	0	1	1	1	0	0
0	1	0	1	0	1	0
1	0	0	0	1	0	1
1	1	0	0	0	1	1

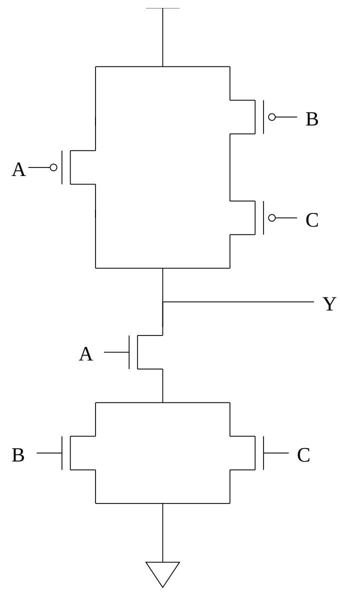
2. [8 Points] **Logic Gates.**

(a) The transistor-level circuit given below implements the following logic equation.

$$Y = \text{NOT}(A) \text{ OR } (\text{ NOT}(B) \text{ AND } \text{NOT}(C))$$

Label all the transistor "gate" inputs with the input (*i.e.*, *A*, *B*, or *C*) that should be applied to it. Also label the output (wherever it may be) with *Y*.

Answer:

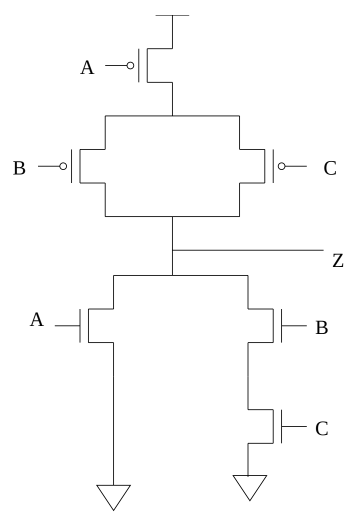


(b) Construct a transistor-level circuit (like the one above) for the following logic equation.

$$Z = \text{NOT}(A) \text{ AND } (\text{ NOT}(B) \text{ OR } \text{ NOT}(C))$$

Be sure to include labels for A, B, C, and Z.

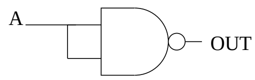
Answer:



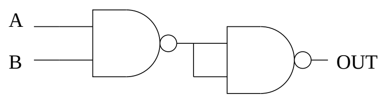
3. [12 Points] **Combinational Logic Circuits.** NAND is logically complete. Use only NAND gates to construct gate-level circuits that compute the following.

   Answer:

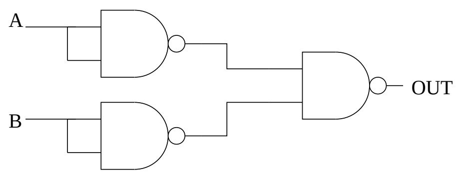
   NOT



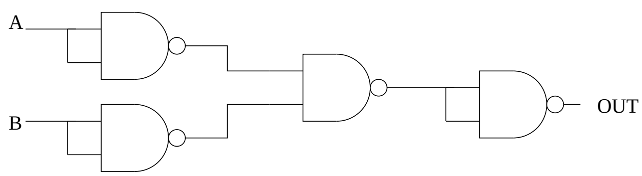
   AND



   OR



   NOR



4. [12 Points] **Combinational Logic Circuits.**

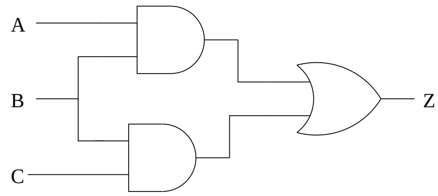
   (a) i. Complete the truth table for $Z = (A \text{ AND } B) \text{ OR } (B \text{ AND } C)$.

   Answer:

A	B	C	$Z = (A \text{ AND } B) \text{ OR } (B \text{ AND } C)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

ii. Construct the gate-level logic circuit for Z (above) using AND and OR .

Answer:



iii. How many transistors does your circuit require if AND and OR gates are implemented as given in Figures 3.6 and 3.7 of the textbook (pages 56 and 57)?

Answer: 18 transistors. (6 transistors each for the three gates used).

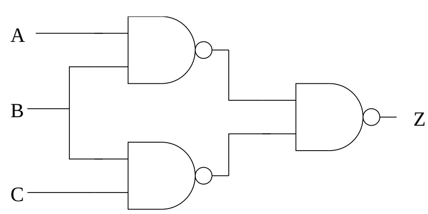
(b) i. Complete the truth table for $Y =$ NOT(NOT(A AND B) AND NOT(B AND C)). You should find that Y is equivalent to Z (from previous question). Indeed, Y can be reduced to Z by application of DeMorgan's Law (and the fact that NOT(NOT(X)) $= X$).

Answer:

A	B	C	$Y =$ NOT(NOT(A AND B) AND NOT(B AND C))
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

ii. Construct the gate-level logic circuit for Y (above) using only NAND gates.

Answer:



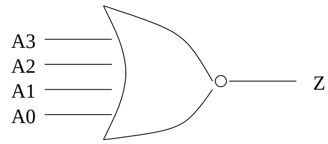
iii. How many transistors does your circuit require if NAND gates are implemented as described in Figure 3.7 of the textbook (page 57)? Does the NAND -based implementation require a greater or lesser number of transistors than the AND - OR implementation?

Answer: 12 transistors. (4 transistors for each NAND gate).

5. [12 Points] **Combinational Logic Circuits.** In this problem we will construct gate-level logic circuits to determine whether a number is zero, negative, or positive. For each part, below, assume the input (A_3, A_2, A_1, A_0) represents a 4-bit 2's complement integer (A_0 representing the least significant digit). You may use any logic gates discussed in the textbook (including gates with more than two inputs, *e.g.*, 4-input AND gates).

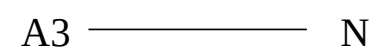
(a) Output Z is 1 if and only if the input represents zero. Construct the gate-level logic circuit to produce output Z from the input. Be sure to label your inputs ($A_3 - A_0$) and output (Z).

Answer:



(b) Output N is 1 if and only if the input represents a negative number. Construct the gate-level logic circuit to produce output N from the input. Be sure to label your inputs and output.

Answer:



(c) Output P is 1 if and only if the input represents a positive number. Construct the gate-level logic circuit to produce output P from the input. Be sure to label your inputs and output.

Answer: There are (at least) two ways to do this:

i. A 2's complement number is positive if its highest order bit (A_3 in this case) is zero AND at least one of the remaining bits (A_2, A_1, A_0 in this case) is one. This is implemented in the figure on the left.

ii. A 2's complement number is positive if it is neither zero nor negative. This is implemented in the figure on the right.