CSE 311: Foundations of Computing I
Assignment #1
September 28, 2010
Due: Wednesday, October 5

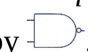
**Reading Assignment:** Read Sections 1.1-1.3 of 7th edition (1.1-1.2 of 5th or 6th edition) of the text. (make sure that you understand the examples).

**Problems:**

1. 7th edition:Section 1.1, exercise 14
   5th and 6th editions: Section 1.1, exercise 10.

2. 7th edition: Section 1.1 exercise 24, parts (a), (c), (f), (g)
   6th edition: Section 1.1, exercise 20, parts (a), (c), (f), (g)
   5th edition: Section 1.1, exercise 18, parts (a), (c), (f), (g).

3. The NAND connective takes two propositions and evaluates to false when both propositions are true and evaluates to true otherwise. NAND of $p$ and $q$ is written as $p|q$.

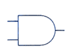
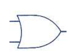
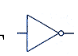
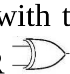
   (In circuit diagrams the gate for NAND is denoted by .)
   Show how to write propositional formulas *only using the NAND connective* that are equivalent to each of the following:

   (a) $\neg p$

   (b) $p \vee q$

   (c) $p \wedge q$

   (d) $p \rightarrow q$

4. Using only AND  gates, OR  gates. and inverters (NOT  gates), draw the diagram of a circuit with two inputs that computes the same function of its inputs that a single two-input XOR  gate does.

5. State in English the converse and contrapositive of each of the following implications:

   (a) If $a$ is pushed onto the stack before $b$, then $b$ is popped before $a$.

   (b) If the input is correct and the program terminates, then the output is correct. (Be sure to use De Morgan's Law to simplify the contrapositive so that the statement reads more naturally in English.)

6. The following two statements form the basis of the most important methods for automated theorem proving. Use truth tables to prove that they are tautologies (i.e., that they always evaluate to true).

(a) Resolution: $((p \vee q) \wedge (\neg q \vee r)) \rightarrow (p \vee r)$

(b) Modus ponens: $((p \wedge (p \rightarrow q)) \rightarrow q$

7. Show that $(p \rightarrow q) \vee (p \rightarrow r)$ and $p \rightarrow (q \vee r)$ are logically equivalent.

8. **Extra Credit:** You have two memory registers, each with the same number of bits. You have an operation, `XOR(R1,R2)`, which takes two registers, `R1` and `R2`, performs bitwise $\oplus$ between them, and stores the result in `R1`. Show how you can swap the contents of the two registers using a sequence of `XOR`s without temporary memory registers. Explain why this works.