CS 234 — Fall 2008	**Problem Set #3**
Computer-Aided Verification	**Due November 5, 2008**

There are two kinds of problems: problems marked "T" are theoretical problems, and problems marked "E" involve coding and experimentation.

Problem T3.1. Given a circuit C over inputs $x_1, \ldots, x_n$, show how you can construct a CNF formula ϕ such that ϕ is satisfiable iff there exists some input values $v_1, \ldots, v_n$ to $x_1, \ldots, x_n$ such that $C(v_1, \ldots, v_n) = 1$. (The formula ϕ can have more variables than $x_1, \ldots, x_n$.)

Problem T3.2. Let X be the set $\{x_0, x_1, y_0, y_1, \mathit{out}_0, \mathit{out}_1, \mathit{carry}\}$. Choose an appropriate variable ordering, and construct the BDD for the requirement that the output $\mathit{out}_1 \mathit{out}_0$ together with the carry bit carry is the sum of the inputs $x_1 x_0$ and $y_1 y_0$. Is your ordering optimal?

Problem T3.3. Given predicates p and q over the set X of variables, the predicate r over X is said to be a *p-simplification of q* if $p \rightarrow (q \leftrightarrow r)$ is valid. Thus, a p-simplification of q must include states that satisfy both p and q, must exclude states that satisfy p but not q, and can treat the remaining states as "don't care" states. Notice that a p-simplification is not unique, and the size of a BDD representation for a p-simplifcation of q can be much smaller than a BDD representation of q. Give an algorithm that computes, given BDD representations of p and q, the BDD representation of some p-simplification of q. The objective should be to reduce the size of the output BDD by exploiting the freedom afforded by the "don't care" when p is false.

Problem T3.4. Prove that Pudlak's algorithm to construct an interpolant for (A, B) (shown in class) is correct.