CS252 – Graduate Computer Architecture

University of California
Dept. of Electrical Engineering and Computer Sciences
David E. Culler

HW 2. Due Feb 22 before class. Work in pairs. **Spring 2005**

Problem 1. Explain why register renaming is essential to exploiting the instruction level parallelism offered by loops.

Problem 2. Do problem 3.3 from the book.

Problem 3. Do problem 3.6a from the book.

Problem 4. If we start with a dynamic execution trace for an entire program, we can eliminate all artificial dependences. Each write to a register is renamed to a new register (assume an infinite number of them) and the reads of that register are renamed until the next write. Similarly, we can rename memory by assigning a unique name to each write. All of the branches are eliminated, since the program has completed execution. All we have is the data dependences. If we assume each operation takes unit time, we can construct an ideal parallelism profile, PP(t), such that each instance of an instruction occurs as soon as its operands are available. PP(t) is the number of instructions that can execute at step t.

We define T_1 to be the time to execute the program sequentially and T_∞ to be the length of the critical path, i.e., the time to execute the program on infinite resources. So $T_1 = \sum_{i=1 \text{ to } T\infty} PP(i)$. $P_{ave} = T_1 / T_\infty$. Let T_p be time to execute the program p instructions at time.

Show that speedup(P_{ave}) $\geq P_{ave} / 2$.

(Hint: try tiling the parallelism profile on time step at a time.)