ENGN 2910A Homework 04 (100 points) – Due Date: Oct 22nd

Professor: Sherief Reda

School of Engineering, Brown University

---

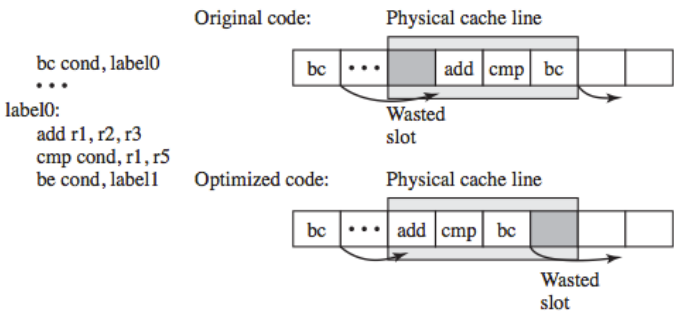
1. (From Shen and Lipasti – 20 points) In 1995, the IBM AS/4000 line of computers transitioned from a CISC instruction set to a RISC instruction set. Because if the simpler instruction set, the realizable clock frequency for a given technology generation and the CPI metric improved dramatically. However, for the same reason, the number of instructions per program also increased noticeably. Given the following parameters, compute the total performance improvement that occurred with this transition. Furthermore, compute the break-even clock frequency, break-even cycles per instruction, and break-even code expansion ratios for this transition, assuming the other two factors are held constant.

Performance factor	AS/400 CISC	AS/400 RISC	Actual ratio	break-even ratio
relative frequency	50 MHz	125 MHz	2.5	?
cycles per instruction	7	3	0.43	?
relative instructions per program (dynamic count)	1000	3300	3.3	?

2. (From Shen and Lipasti – 20 points) Superscalar pipelines require replication of pipeline resources across each parallel pipeline, naively including the replication of cache ports. In practice, however, a two-wide superscalar pipeline may have two data cache ports but only a single instruction cache port. Explain why this is possible, but also discuss why a single instruction cache port may perform worse than two (replicated) instruction cache ports.

3. (From Shen and Lipasti – 20 points) A compiler can generate object code where branch targets are aligned at the beginning of physical cache lines to increase the likelihood of fetching multiple instructions from the branch target in a single cycle. However, given a fixed number of instructions between taken branches, this approach may simply shift the unused fetch slots from before the branch target to after the branch that terminates sequential fetch at the target. For example, moving the code at label0 in the following figure so it aligns with a physical cache line will not improve fetch efficiency, since the wasted fetch slot shifts from the beginning of the physical line to the end.

Discuss the relationship between fetch block size and the dynamic distance between taken branches. Describe how one affects the other, describe how important is branch target alignment for small vs. large fetch blocks and short vs. long dynamic distance, and describe how well static compiler-based target alignment might work in all cases.



4. (From Shen and Lipasti – 20 points) The Pentium 4 processor operates its integer arithmetic units at double the nominal clock frequency of the rest of the processor. This is accomplished by pipelining the integer adder into two stages, computing the low-order 16 bits in the first cycle and the high-order 16 bits in the second cycle. Naively, this appears to increase ALU latency from one cycle to two cycles. However, assuming that two dependent instructions are both arithmetic instructions, it is possible to issue the second instruction in the cycle immediately following issue of the first instruction, since the low-order bits of the second instruction are dependent only on the low-order bits of the first instruction.

a. Sketch out a pipeline diagram of such an ALU along with the additional bypass paths needed to handle this optimized case.

b. Specify how many cycles a trailing dependent instruction of each of the following types must delay, following the issue of a leading arithmetic instruction: arithmetic, logical (and/or/xor), shift left, shift right.

5. (20 points) We examined in class the fetch alignment unit of the IBM RS/6000 system. Despite the clever design, a fetch misalignment can still occur when a fetch group crosses a cache line boundary. Thus, a the goal of attaining a four instructions fetched per cycle will not be attained on the average. Calculate the average fetch bandwidth of IBM RS/6000 system given the I-cache organization described in class. Hint: a probabilistic argument has to be made here.