

TA: Scott Beamer

Problem 1: Iron Law

When we introduced the Iron Law earlier in the course, we used CPI (cycles per instruction). Rewrite it using IPC (instructions per cycle). What does this shift indicate? Is there an advantage to using CPI?

execution time = (instructions per program)(clock period)/(instructions per cycle)

This shift indicates the move to ILP and getting more done. CPI is easier to compute by hand since it is a simple weighted average. This isn't too big a deal since IPC is usually measured anyways.

Problem 2: Resource Scaling

Given a simple single issue in-order pipeline, you decide to add a second pipeline and make it dual issue (and fetch). By how much does just the hazard detection and forwarding logic increase?

It will actually quadruple rather than doubling, so for N way it is N^2 more.

Problem 3: Branch Predictor Necessity

In this problem we will continually modify and improve a processor, and see how important branch prediction is. Assume that branches occur once every 10 cycles. For all of these questions, respond with the CPI component for the branch penalty if there is no branch prediction, if it is 50% accurate, and if it is 90% accurate. You may assume that there are no non-branch hazards.

Note the following calculations aren't precise, but intended to convey the point that branch prediction is needed to keep wide and deep pipelines full.

a) Simple in-order single issue pipeline with a branch delay slot of 3 cycles

No BP: $0.1 * 3 = 0.3$

50% correct BP: $0.1 * 3 * 0.5 = 0.15$

90% correct BP: $0.1 * 3 * 0.1 = 0.03$

b) We decrease the clock rate by making the pipeline deeper so now the branch delay slot is 5 cycles

No BP: $0.1 * 5 = 0.5$

50% correct BP: $0.1 * 5 * 0.5 = 0.25$

90% correct BP: $0.1 * 5 * 0.1 = 0.05$

c) We make our system dual issue (& fetch)

No BP: $2 * 5 * .1 = 1$

50% correct BP: $2 * 5 * .1 * 0.5 = 0.5$

90% correct BP: $2 * 5 * .1 * 0.1 = 0.01$

d) We deepen our pipeline again so now our branch penalty is 12 cycles

No BP: $2 * 12 * .1 = 2.4$

50% correct BP: $2 * 12 * .1 * 0.5 = 1.2$

90% correct BP: $2 * 12 * .1 * 0.1 = 0.24$

e) We widen our CPU so now it is quad issue (& fetch)

No BP: $4 * 12 * .1 = 4.8$

50% correct BP: $4 * 12 * .1 * 0.5 = 2.4$

90% correct BP: $4 * 12 * .1 * 0.1 = 0.48$