Notes for 9/28/09

CPI is clock cycles per instruction. Inverse of CPI is IPC: instructions per cycle.

RISC vs. CISC:

$$\frac{seconds}{program} = \frac{instructions}{program} * \frac{clock\,cycles}{instruction} * \frac{seconds}{clock\,cycle}$$

With RISC, instructions per program hurts you. But, with CISC, clock cycles per instruction and seconds per clock cycle hurt you, so RISC is generally considered better.

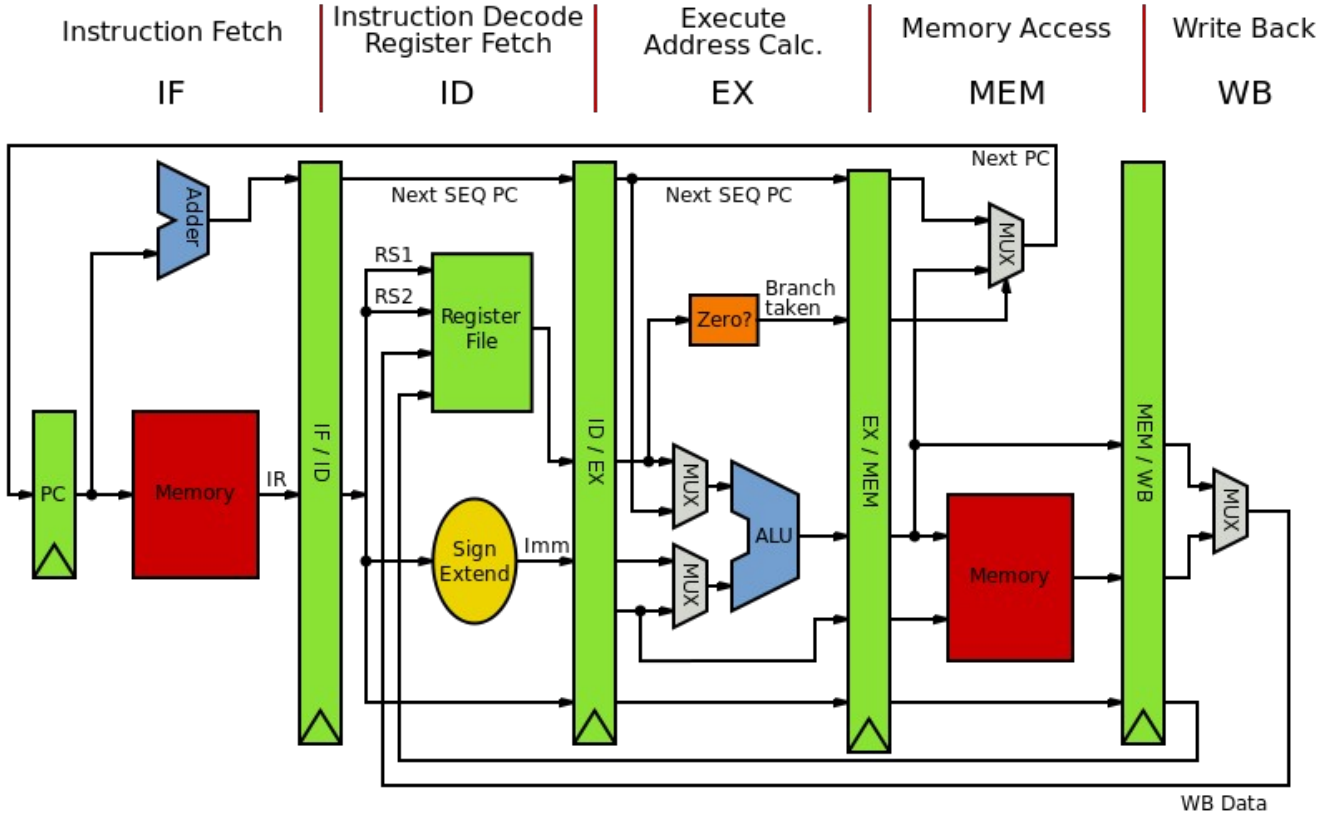
With scalar processors, IPC ≤ 1.
With superscalar processors, IPC can be > 1.

MIPS (not the ISA this time)—millions of instructions per second.

MIPS can be misleading:
Case A: 10 billion, clock rate 4GHz, CPI 1.0, same program
Case B: 8 billion, clock rate 4 GHz, CPI 1.1, same program
Case A takes 2.5 seconds, MIPS 4000
Case B takes 2.2 seconds, MIPS 3636

Even though B finishes earlier, A wins vis-a-vis MIPS. Why? Because B requires fewer instructions.

MIPS pipeline:



1. Instruction fetch: read instruction from I-Cache/DRAM and increment PC by 4.
2. Instruction decode/register fetch: get encoded registers and/or immediate values from instruction, sign extend them, and then pass them to ALU
3. Execute/address calculation: do arithmetic and branch evaluation
4. Memory access: load values from D-Cache/DRAM
5. Write back: write back into the register file