RISC Architectures
RISC I

- RISC: Reduced Instruction Set Computer
- Alternative to Complex Instruction Set Computer (CISC)
- Simple instructions and addressing modes
  - High effective throughput (low CPI)
    - Effective pipeline: Most instructions execute in one cycle
  - Short cycle time
  - Short design cycle
- But larger programs
  - 2x larger than VAX 11/780
RISC Instruction Format

- Most instructions execute in one cycle
- Fixed size (32-bits)
- Only loads and stores access memory
  - Two cycle instructions (compare current latencies)
  - Rest of instructions operate between registers
- Support high level languages (HLL)
- Paper: Instruction format and addressing modes
RISC I Design Approach

- New architectures should be designed for HLL
- Does not matter which part of the system is in hardware and which is in software
- Architecture tradeoffs to build a cost-effective system:
  - Which language constructs are used frequently?
  - What is the distribution of various instructions?
  - Dedicate available area for the most frequent constructs and operations (Paper: Tables)
  - Remember Amdahl’s law
Amdahl’s Law

\[
\text{Speedup} = \frac{1}{1 - P + P/S}
\]

P = proportion of computation improved
S = improvement speedup

Example: Parallel Execution
P: Parallel portion, S: Serial portion = 1-P
N: Number of Cores

\[
\text{Speedup} = \frac{1}{S + P/N} = \frac{1}{1 - P + P/N}
\]
RISC I Performance Features

- Large number of registers addressable by instructions
  - 32 general purpose registers (GPRs)
  - R0 is always zero (to support addressing modes)
- Register windows for fast call and return operations (Paper)
- Delayed branch
Delayed Branch

Static Program: I1
                Jump Target
                I2

Target: I3

Branch execution sequence: I1, I3
Delayed branch execution sequence: I1, I2, I3

Question: Which modern machine still supports delayed branches?

Question: What is the downside for delayed branches?
Reading Assignment

- Review due before class on Monday