

Lecture 6 Review

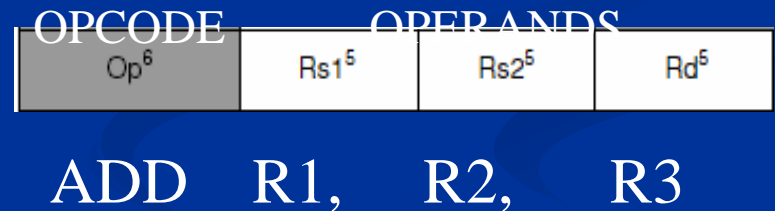
- Review of basic computer organization
 - Registers in the CPU, external memory
 - Program counter (PC) points to address of next instruction to fetch
 - Fetched instruction is loaded in the instruction register (IR)
- Load/Store architecture (only loads and stores can make data accesses to memory)

Lecture 6 Review

- Instruction Set Architecture (ISA)
 - Interface between hardware and software
 - Compilers translate high-level code into machine instructions. The set of all possible instructions and how they are used is called the ISA.
- Two philosophies – RISC vs. CISC
 - CISC includes lots of different instructions and ways of using them
 - RISC uses only a limited set
 - RISC allows for high-performance organizations (simple, regular instructions)
 - Today's CISC holdouts (Intel) internally translate to RISC instructions and execute those

Lecture 6 Review

- English versions of the instructions are called “assembly language”. Assembly instructions are one-to-one mappings to the machine instructions that directly control the operation of the hardware



Lecture 6 Review

- Data types
- Memory addressing
 - Alignment, byte ordering

Lecture 6 Review

- We saw there were different ways to classify ISAs based on their operands.
- Stack and accumulator ISAs use special-purpose registers to hold their operands.
- Instructions in Memory-Register ISAs take operands from memory or general-purpose registers
- Register-Register instructions take all their operands from registers
- Compilers work better if they are given general-purpose registers and are constrained by special-purpose registers.

Lecture 7 Review

- Addressing modes
 - Immediate, register-register, displacement
 - Read textbook for others (not used in MIPS)
- Control flow (branches)
 - Change PC based on a condition
- Compilers
 - Make the frequent case fast and the rare case correct

Lecture 8 Review

- MIPS-64 ISA
- Register transfer language (RTL)