# MIPS

- RISC
- 32-bit Architecture
- Operands can be stored in registers, memory, or constants stored in the instruction itself (298)
- 32 registers (299 + 300)
- To clarify/add-to, MIPS uses 32-bit memory addresses & 32-bit words
- byte-addressable memory (each byte has a unique address)
-

> In the MIPS architecture, word addresses for `lw` and `sw` must be *word aligned*. That is, the address must be divisible by 4. Thus, the instruction `lw $s0, 7($0)` is an illegal instruction. Some architectures, such as x86, allow non-word-aligned data reads and writes, but MIPS requires strict alignment for simplicity. Of course, byte addresses for load byte and store

(304)

-