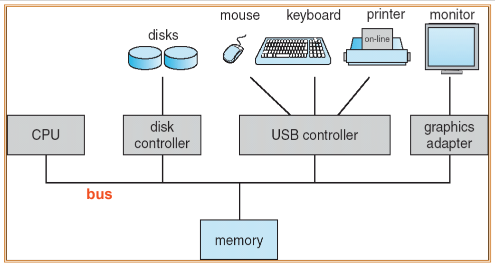
CSE 675.02: Introduction to Computer Architecture

Instruction Set Architecture of MIPS Processor

Presentation B

Basic (Single CPU) Computer Structure



- CPU and device controllers connect through common bus providing access to shared memory
- Multiprocessor has multiple CPUs

MIPS Processor

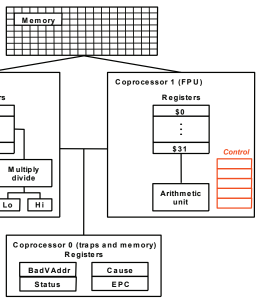


Figure A.10.1

Reading Assignment: 1, 2.1-2.9; Handout “MIPS Instructions:..”

MIPS arithmetic

- All instructions have 3 operands
- Operand order is fixed (destination first)

Example:

C code: `a = b + c`

MIPS ‘code’: `add a, b, c`

“The natural number of operands for an operation like addition is three…requiring every instruction to have exactly three operands, no more and no less, conforms to the philosophy of keeping the hardware simple”

MIPS arithmetic

- Design Principle: simplicity favors regularity.
- Of course this complicates some things...

C code: `a = b + c + d;`

MIPS code:

```
add a, b, c
add a, a, d
```

- Operands must be registers, only 32 registers provided
- Each register contains 32 bits
- Design Principle: smaller is faster. Why?

Registers vs. Memory

- Arithmetic instructions operands must be registers,
  — only 32 registers provided
- Compiler associates variables with registers
- What about programs with lots of variables?

MIPS Registers

- Main Processor (integer manipulations):
  - 32 32-bit general purpose registers – GPRs (r0 – r31);

    r0 has fixed value of zero. Attempt to write into r0 is not illegal, but its value will not change;
  - two 32-bit registers – Hi & Lo, hold results of integer multiply and divide
  - 32-bit program counter – PC;
- Floating Point Processor – FPU (Coprocessor 1 – CP1; real number manipulations):
  - 32 32-bit floating point registers – FPRs (f0 – f31);
  - five control registers;

MIPS Registers (continued)

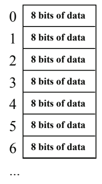
- Coprocessor 0 – CP0 is incorporated on the MIPS CPU chip and it provides functions necessary to support operating system: exception handling, memory management scheduling and control of critical resources.
- Coprocessor 0 (CP0) registers (partial list):
  - Status register (CP0reg12) – processor status and control;
  - Cause register (CP0reg13) – cause of the most recent interrupt;
  - EPC register (CP0reg14) – program counter at the last interrupt;
  - BadVAddr register (CP0reg08) – the address for the most recent address related exception;

MIPS Data Types

- MIPS operates on:
  - 32-bit (unsigned or 2's complement) integers,
  - 32-bit (single precision floating point) real numbers,
  - 64-bit (double precision floating point) real numbers;
- 32-bit words, bytes and half words can be loaded into GPRs
- After loading into GPRs, bytes and half words are either zero or sign bit expanded to fill the 32 bits;
- Only 32-bit units can be loaded into FPRs and 32-bit real numbers are stored in even numbered FPRs.
- 64-bit real numbers are stored in two consecutive FPRs, starting with even-numbered register.

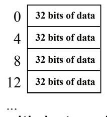
Memory Organization

- Viewed as a large, single-dimension array, with an address.
- A memory address is an index into the array
- "Byte addressing" means that the index points to a byte of memory.



Memory Organization

- Bytes are nice, but most data items use larger "words"
- For MIPS, a word is 32 bits or 4 bytes.



Registers hold 32 bits of data

- 2^{32} bytes with byte addresses from 0 to 2^{32}-1
- 2^{30} words with byte addresses 0, 4, 8, ... 2^{32}-4
- Words are aligned
  i.e., what are the least 2 significant bits of a word address?

MIPS Addressing Modes

- register addressing;
- immediate addressing;
- register indexed is the only memory data addressing; (in MIPS terminology called base addressing):
  - *memory address* = register content plus offset
- since r0 always contains value 0:
  - r0 + offset → absolute addressing;
- offset = 0 → register indirect;
- branch instructions use PC relative addressing:
  - branch address = [PC] + 4 + 4×offset
- jump instructions use:
  - pseudo-direct addressing *with* 28-bit addresses (jumps inside 256MB regions),
  - direct (absolute) addressing *with* 32-bit addresses.

MIPS Alignment

- MIPS supports byte addressability:
  - it means that a byte is the smallest unit with its address;
- MIPS restricts memory accesses to be aligned as follows:
  - 32-bit word has to start at byte address that is multiple of 4;

    Thus, 32-bit word at address 4n includes four bytes with addresses: 4n, 4n+1, 4n+2, and 4n+3.
  - 16-bit half word has to start at byte address that is multiple of 2; Thus, 16-bit word at address 2n includes two bytes with addresses: 2n and 2n+1.
- MIPS supports 32-bit addresses:
  - it means that an address is given as 32-bit unsigned integer;

MIPS Instruction Classes

- *Instructions that move data:*
  - *load to register from memory,*
  - *store from register to memory,*
  - *move between registers in same and different coprocessors*
- *ALU integer instructions:* register – register and register-immediate computational instructions*,*
- *Floating point instructions:* register – register computational instructions and real number to/from integer conversion instructions*,*
- *Control-related instructions,*
- *Special control-related instructions.*