Chapter 2
Instructions:
Language of the Computer

Learn a new processor

- Top-down approach (for example, pic16F917 data sheet, see http://www.microchip.com)
- Bottom-up approach (our textbook)

Compiling Two C Assignment Statements into MIPS (Section 2.2)

High-level programming language:

- a = b + c;
- d = a – e;

MIPS instructions:

- add a, b, c
- sub d, a, e

What have we learned?

- A MIPS instruction operates on two source operands and places the result in one destination operand.

Compiling a Complex C Assignment into MIPS

High-level language statement:

- f = (g + h) – (i + j)

MIPS Instructions

- add t0, g, h #temporary variable t0
- add t1, i, j #
- sub f, t0, t1 #

What have we learned?

- MIPS can't perform this task in one operation.

Why?

- MIPS only has one ALU (Arithmetic Logic Unit), which can only perform one operation (e.g., add or sub) at a time.

Any observations?

Q. What are those t0 and t1?

A. They are so-called registers.

Q. What are those f, g, h, i and j?

A. They are memory cells (locations).

How many registers should we have?

If you want a lot:

Pros:

- Plenty of storage spaces for holding values
- Reducing traffic to and from memory cells

Cons:

- More hardware
- Take more time to select/activate/specify the intended register.

So how many did the MIPS designer choose?

Possible answers: 2, 4, 8, 16, 32, 64, 128, 256

Answer:

MIPS has 32 registers. (Design Principle 2: Smaller is faster.)

Observation: Can we have 30 registers instead of 32?

A: