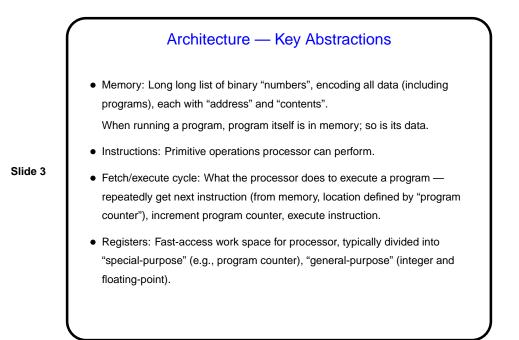


"Architecture" as Interface Definition

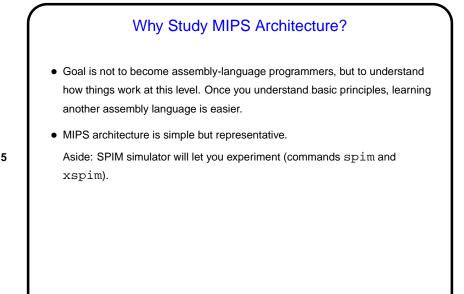
- From software perspective, "architecture" defines lowest-level building blocks

 what operations are possible, what kinds of operands, binary data formats, etc.
- From hardware perspective, "architecture" is a specification designers must build something that behaves the way the specification says.

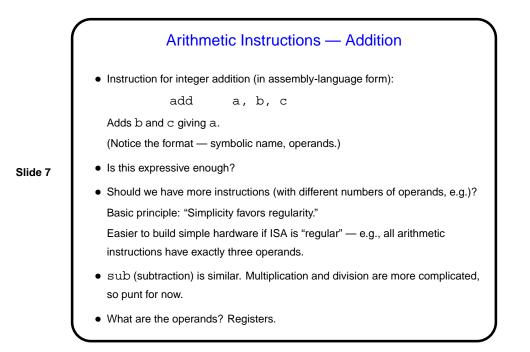


Design Goals for Instruction Set

- From software perspective expressivity.
- From hardware perspective good performance, low cost.



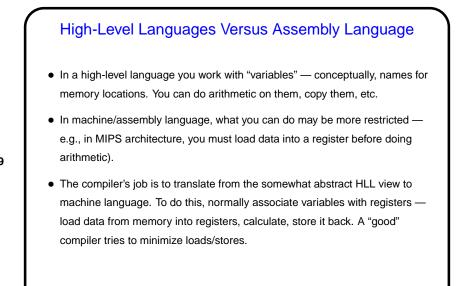
A Bit About Assembly Language Syntax Syntax for high-level languages can be complex. Allows for good expressivity, but translation into processor instructions is complicated. Syntax for assembly language, in contrast, is very simple. Less expressivity but much easier to translate into (binary form of) instructions.



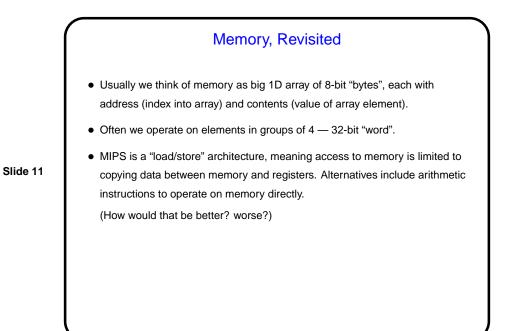
Registers
Access to main memory is slow compared to processor speed, so it's useful to have a within-the-chip memory — "registers".
MIPS architecture defines 32 "general-purpose" registers, each 32 bits.
Would more be better?

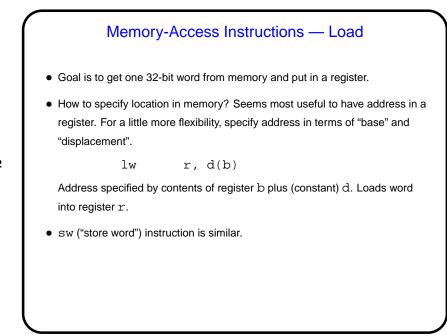
Basic principle: "Smaller is faster."

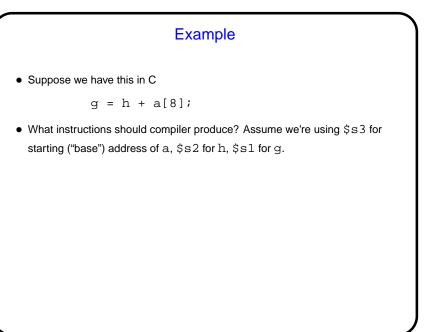
In machine language, reference by number.
In assembly language, useful to adopt conventions for which registers to use for what, use symbolic names indicating usage.
E.g., use registers 8 through 15 for "temporary" values (short-term), refer to as \$t0 through \$t7.

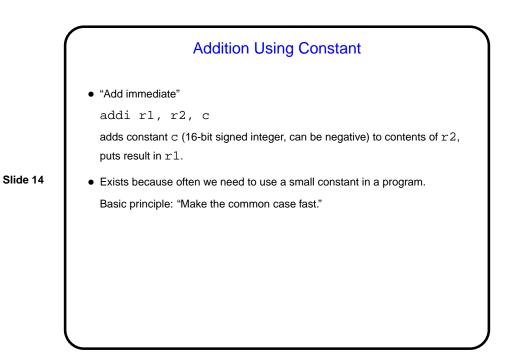


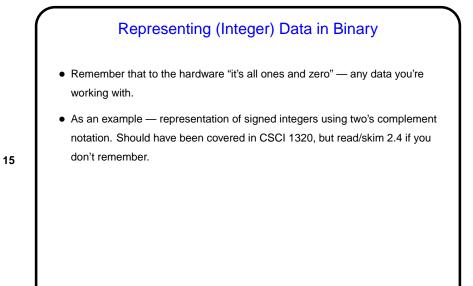
Example
Suppose we have this in C
f = (g + h) - (i + j)
 What instructions should compiler produce? (Assume we're using \$\$0 for f, \$\$1 for g, \$\$2 for h, \$\$3 for i, \$\$4 for j.)











A Little About the Simulator Your code goes in a file with extension . s. (Sample starter code on "Sample programs" page. Contains many things we haven't talked about yet but could still be useful for trying things out.) Start the simulator with command xspim (spim for command-line version). (Short demo.)

Minute Essay• Write MIPS assembly code for the following C program fragment:a = b + c + d + eAssume we have b, c, d, e in \$s1 through \$s4 and want to have a in \$s0Optional: Can you think of more than one way to do it? If you can, does one seem better than the other, and why?OR• Write MIPS assembler code to exchange the values of a [0] and a [1].Assume register \$s0 contains the address of a (start of the array), and a is an array of integers.

(Minute Essay Answer	
	• One way:	
	add \$s0, \$s1, \$s2	
	add \$s0, \$s0, \$s3	
	add \$s0, \$s0, \$s4	
	Another way (not as good since uses more registers?):	
Slide 18	add \$t0, \$s1, \$s2	
	add \$t1, \$s3, \$s4	
	add \$s0, \$t0, \$t1	
	One way:	
	lw \$t0, 0(\$s0)	
	lw \$t1, 4(\$s0)	
	sw \$t0, 4(\$s0)	
	sw \$t1, 0(\$s0)	