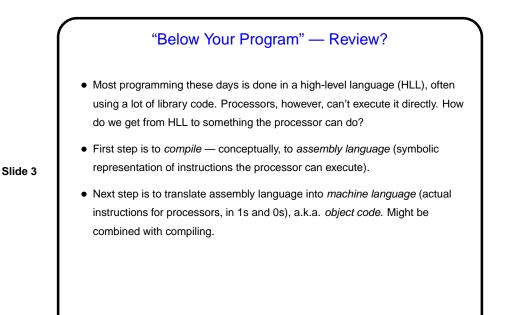
Administrivia

• First homework to be on the Web soon, due a week from Tuesday. (I will send mail.)

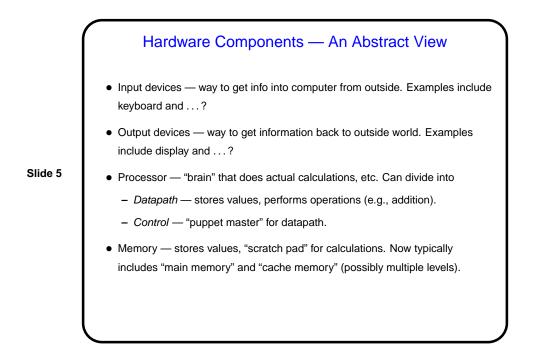
Slide 1

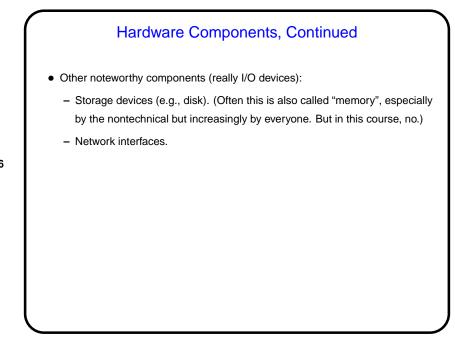
Introduction "Computers are everywhere" — you know about desktops and servers, which are more and more central to our lives, but also consider "embedded processors", largely invisible but even more prevalent. It seems to be a truism that however fast computers can process information, they can't keep up with humans' ability to imagine things for them to do. So performance matters. Factors that affect performance include both the ones you learn about in programming courses (order of magnitude of algorithms, e.g.) and lower-level ones (how well the compiler can map HLL onto hardware in various respects, how fast the hardware can execute instructions).



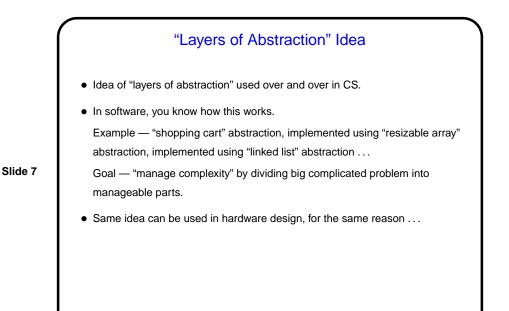


- Final step is to combine object code for your program with library object code. Can be done as part of compiling process to create an *executable file* or at runtime, or some combination of the two.
- Actual execution of program typically involves operating system (something manages physical resources / provides abstraction for applications).
 Contents/format of executable files depends on operating system as well as hardware.
- Worth noting that some languages/implementations don't exactly follow this scheme some languages (e.g., shell scripts) are translated/interpreted at runtime, and others (e.g., Scala and Java) are compiled to machine language for a virtual processor (the JVM), which may then be translated into "native code" at runtime.
- Slide 4



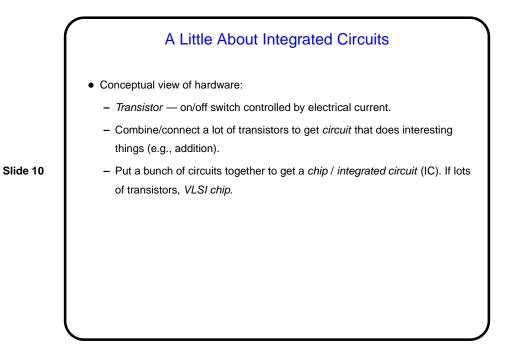


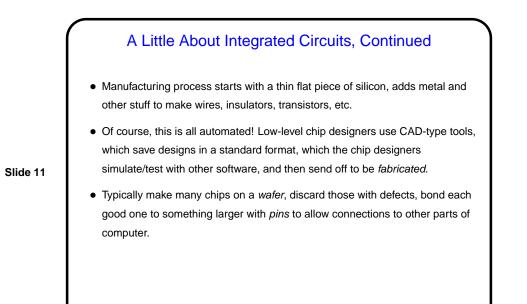
Slide 8



"Layers of Abstraction" Idea in Hardware
Instruction set architecture (ISA or architecture) — a definition/specification of how the hardware behaves, detailed enough for programming at assembly-language level.
E.g., "x86 architecture", "MIPS architecture", "IBM 360 architecture".
Implementations of an architecture — actual hardware that behaves as defined. Can have many implementations of an architecture, allowing the same program executable to run on (somewhat) different hardware systems.
E.g., Intel chips, IBM 360 family of processors.







 Defining Performance

 • What does it mean to say that computer A "has better performance than" computer B?

 • Really — "it depends". Some answers:

 • Computer A has better response time / smaller execution time.

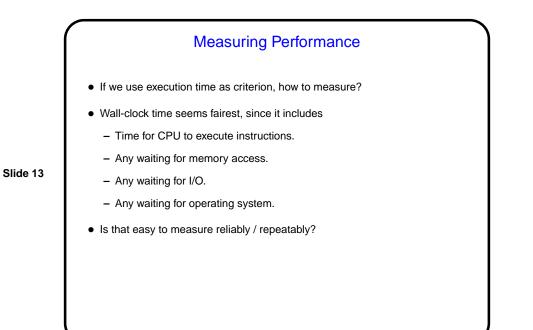
 • Computer A has higher throughput.

 • We'll use execution time, and say

 $\frac{Performance_A}{Performance_B} = n$

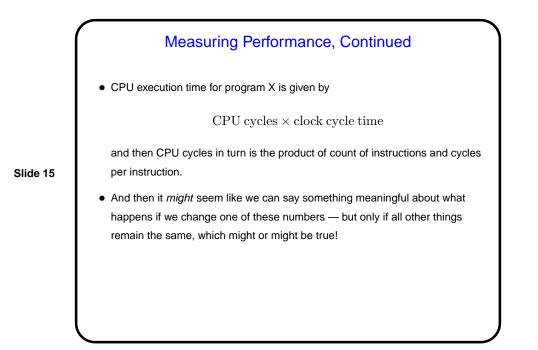
 exactly when

 $\frac{Execution time_B}{Execution time_A} = n$



Measuring Performance, Continued

- No to get repeatable measure of wall clock time, need an otherwise unused system.
- So instead we could use "CPU performance" amount of time CPU needs to run program. Easier to measure, more consistent, and at least says something about the processor.
- Even that, though, is not as simple as it might seem.



Evaluating / Comparing Performance
Trickier than it sounds to come up with one number that means something.
Approaches include

Use the actual workload, on the actual hardware platform(s), and compare times.

Slide 16

Put together a representative simulated workload — "benchmark"; run and compare times.
Compare code size.
Compare number of instructions per second ("MIPS" or "MFLOPS").

Alas, all of these are flawed in some way. (In particular, paraphrasing someone whose name I don't remember, "peak MIPS is just the number you can't go any faster than.") Slide 17

Minute Essay

- Suppose you are trying to decide which of two computers, call them Foo and Bar, will give you the best performance. You run two test programs on Foo and observe execution times of 10 seconds for one and 20 seconds for the other. If the first program takes 5 seconds on Bar, how long does the second program take? (Hint: This might be something of a trick question.)
- Other questions?

Minute Essay Answer

 It might seem like that second program would take 10 seconds on Bar, but in truth you probably can't be sure without doing the experiment, since the two machines, or the two test programs, could differ in ways that would make this obvious answer wrong.