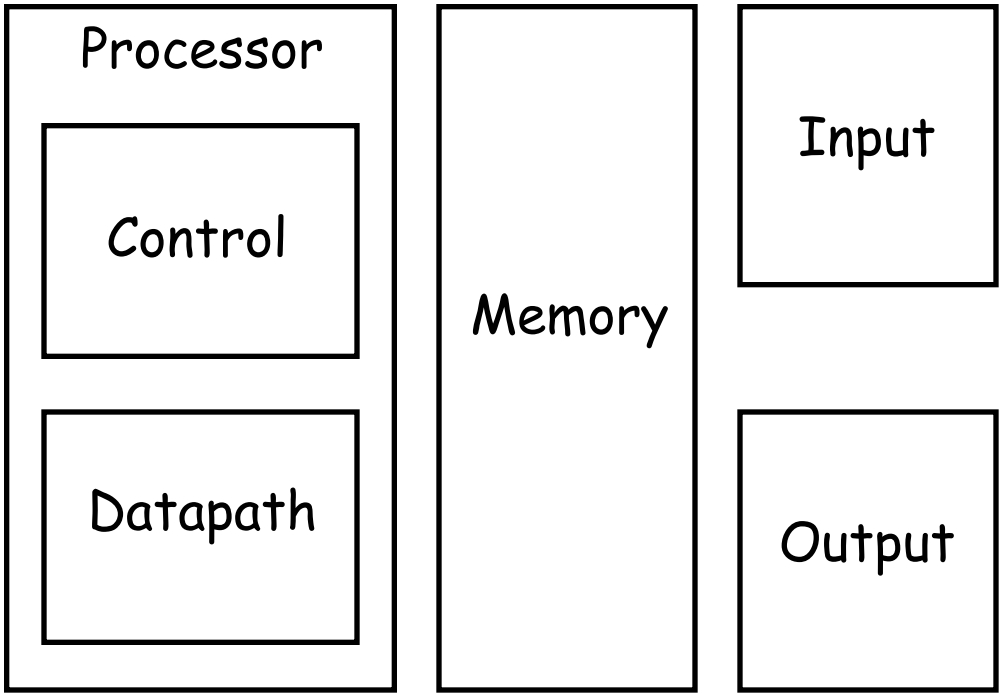
About Computer Architecture

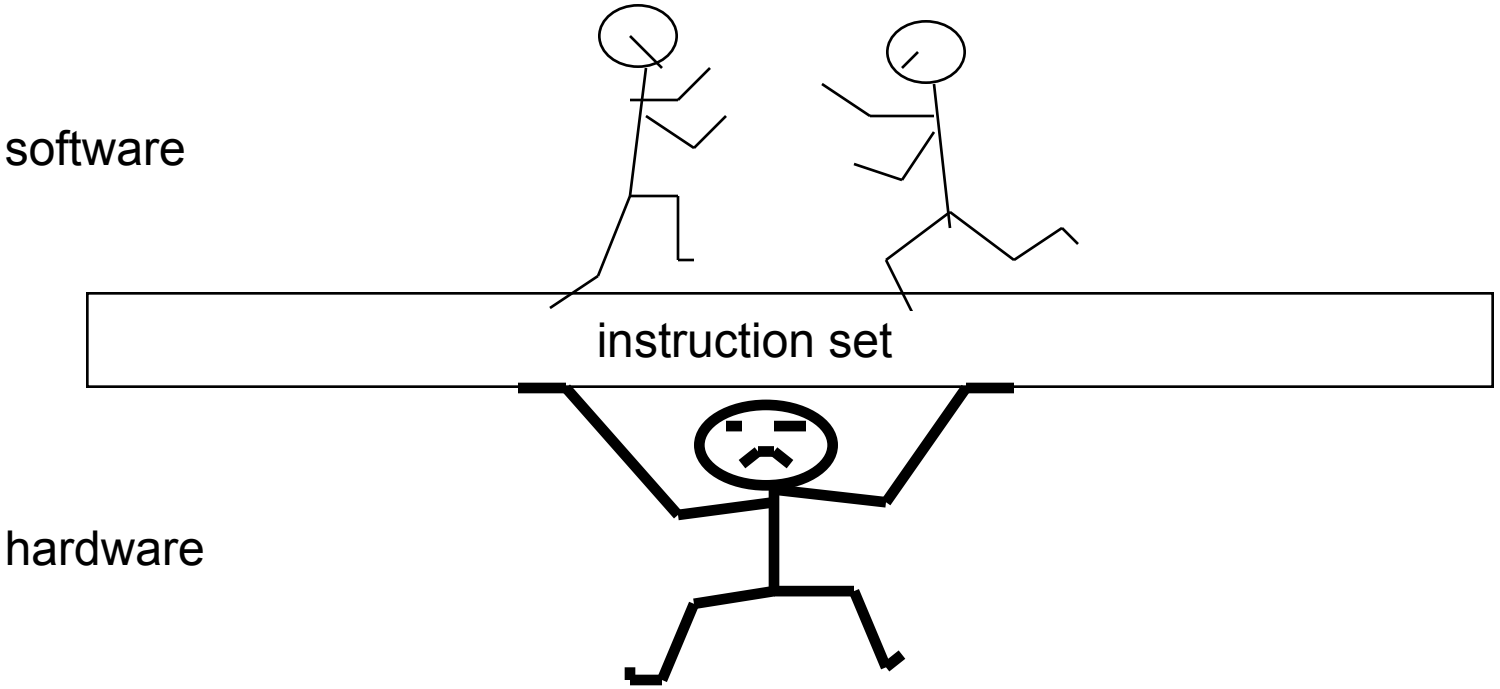
1. What ?
2. Why?
3. Where ?

What?

- Since 1946 all computers have had 5 components

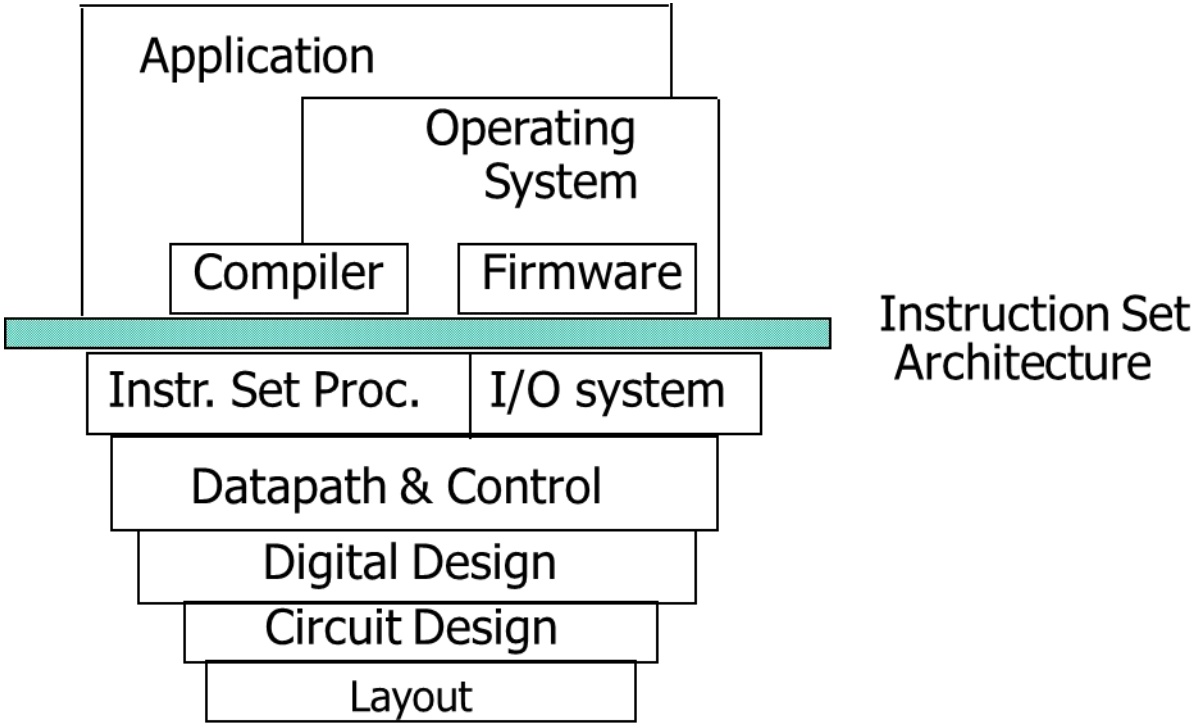


Instruction Set Architecture: Critical Interface

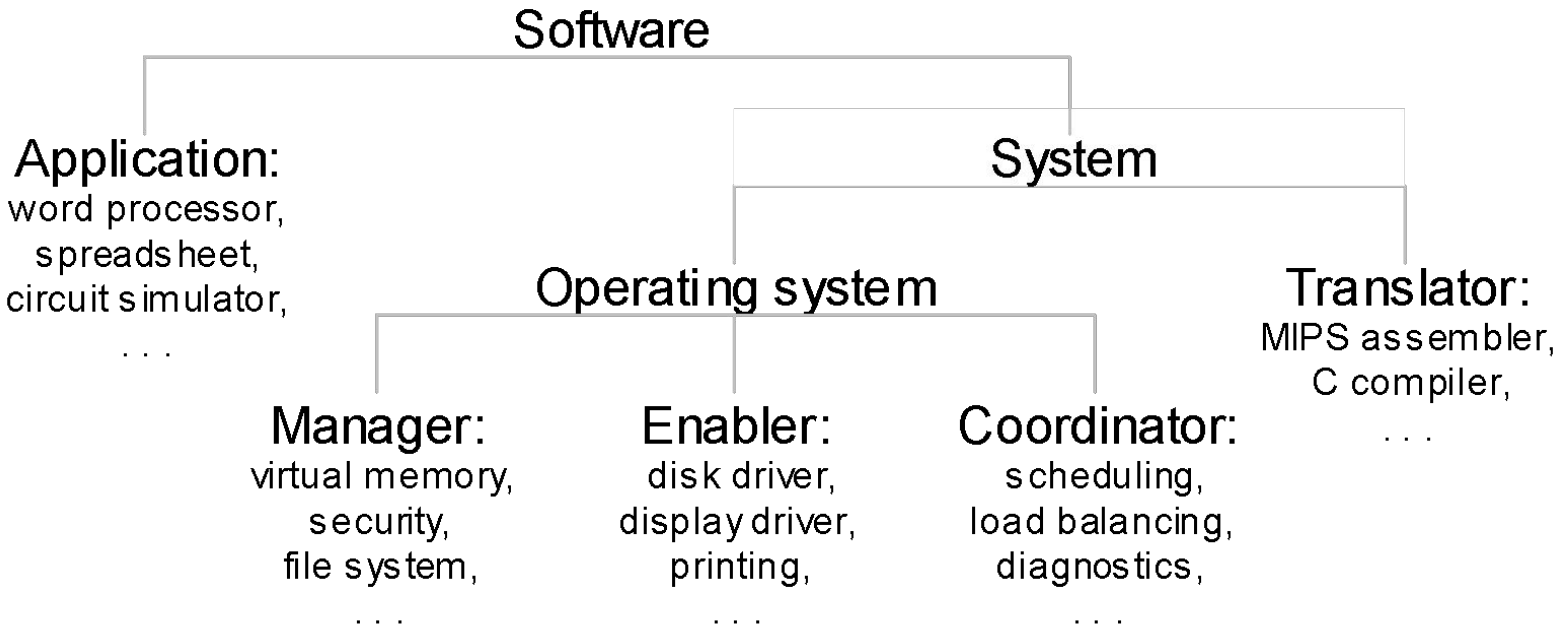


- Properties of a good abstraction
  - Lasts through many generations (portability)
  - Used in many different ways (generality)
  - Provides convenient functionality to higher levels
  - Permits an efficient implementation at lower levels

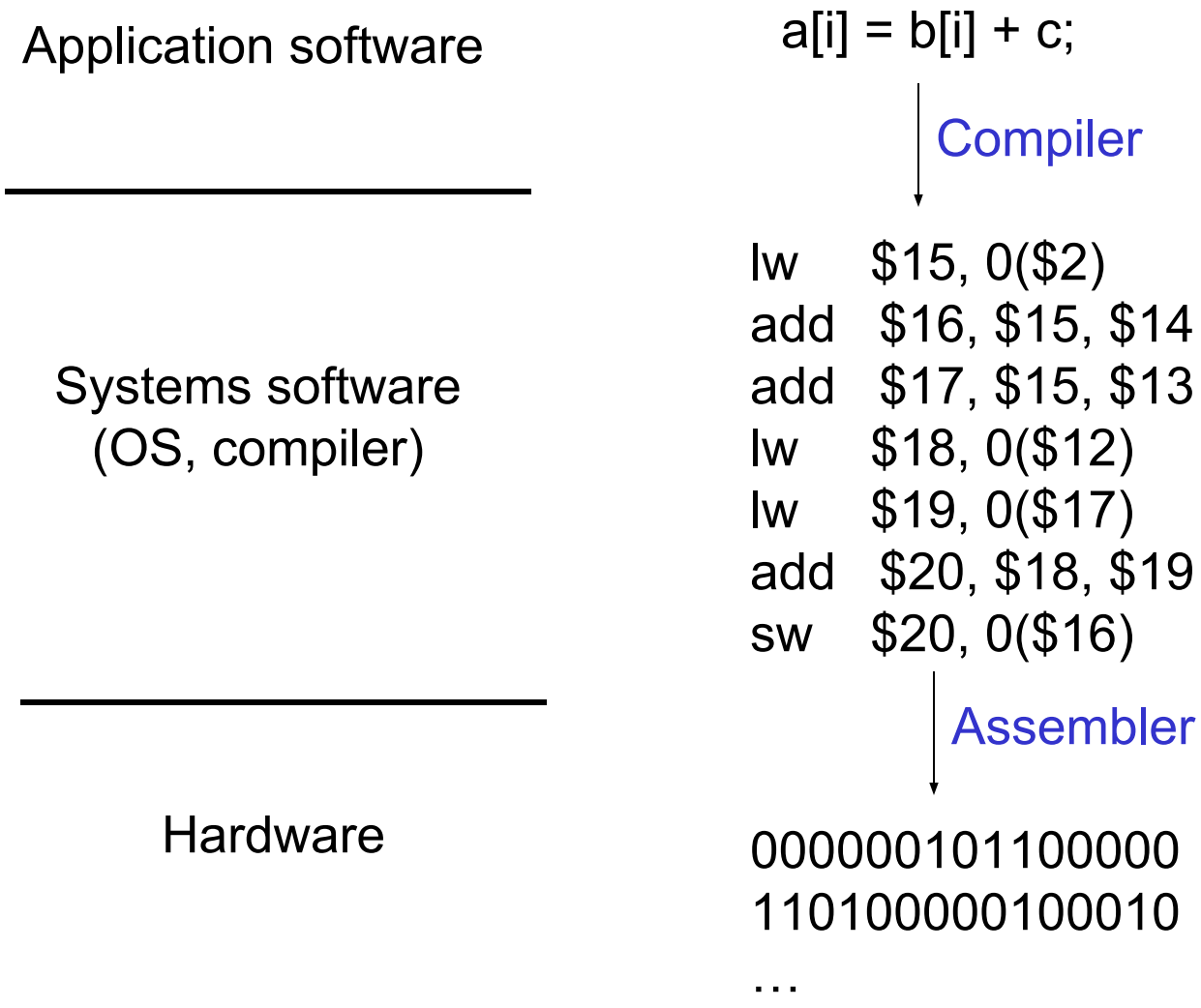
Computer Architecture

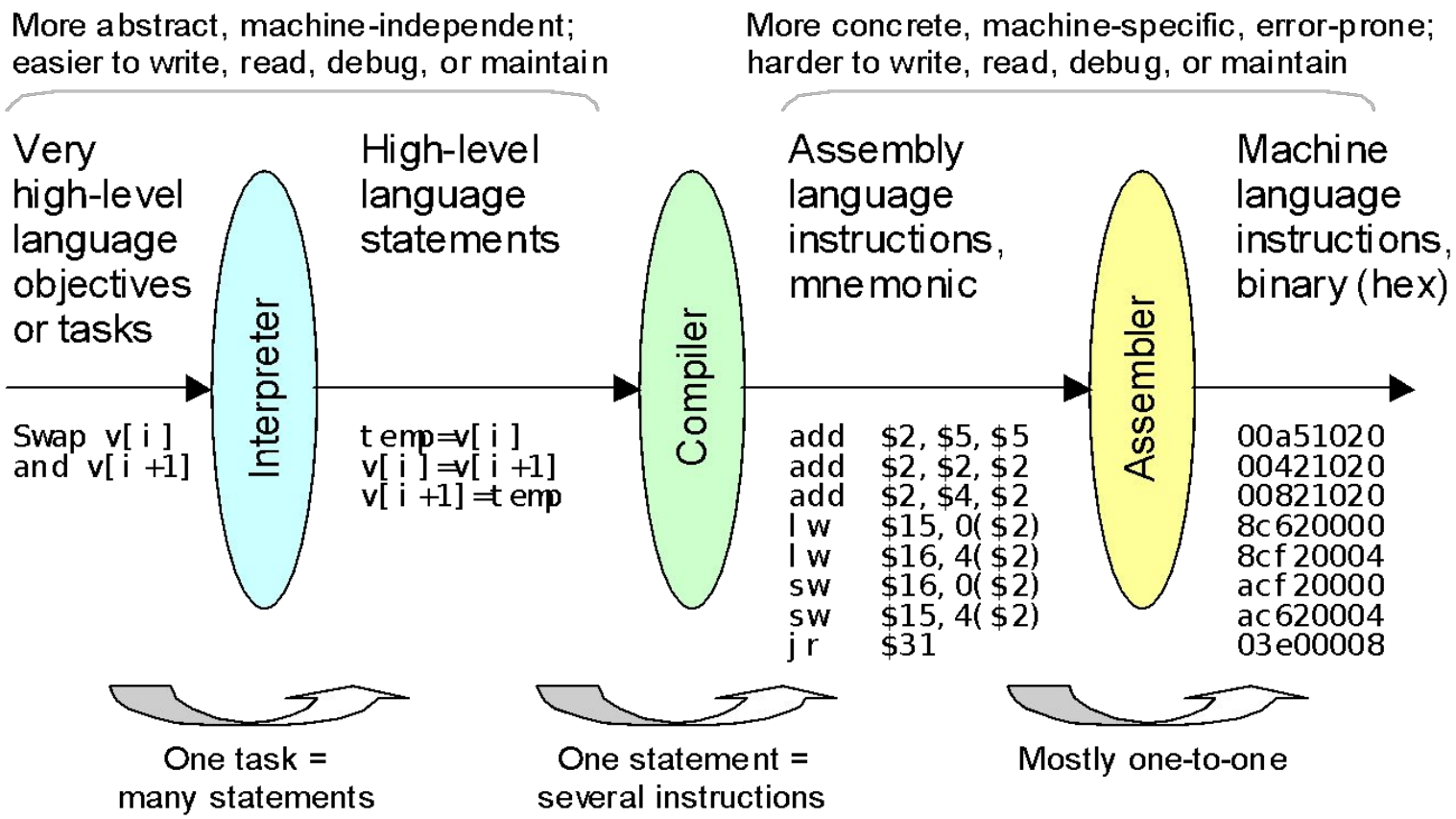


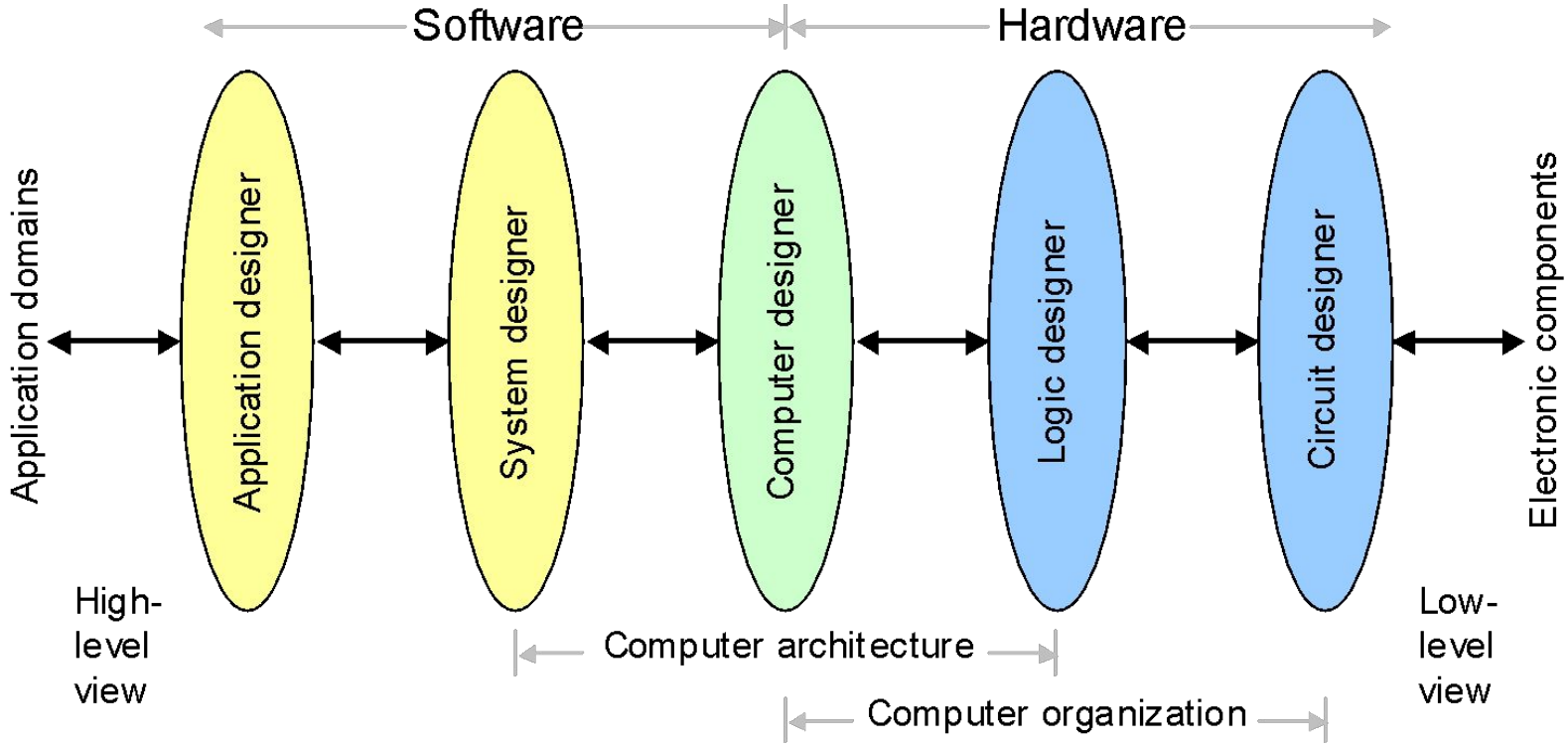
- Coordination of many levels of abstraction
- Under a rapidly changing set of forces
- Design, Measurement, and Evaluation


Software
Application:
word processor,
spreadsheet,
circuit simulator,
. . .
System
Operating system
Translator:
MIPS assembler,
C compiler,
. . .
Manager:
virtual memory,
security,
file system,
. . .
Enabler:
disk driver,
display driver,
printing,
. . .
Coordinator:
scheduling,
load balancing,
diagnostics,
. . .

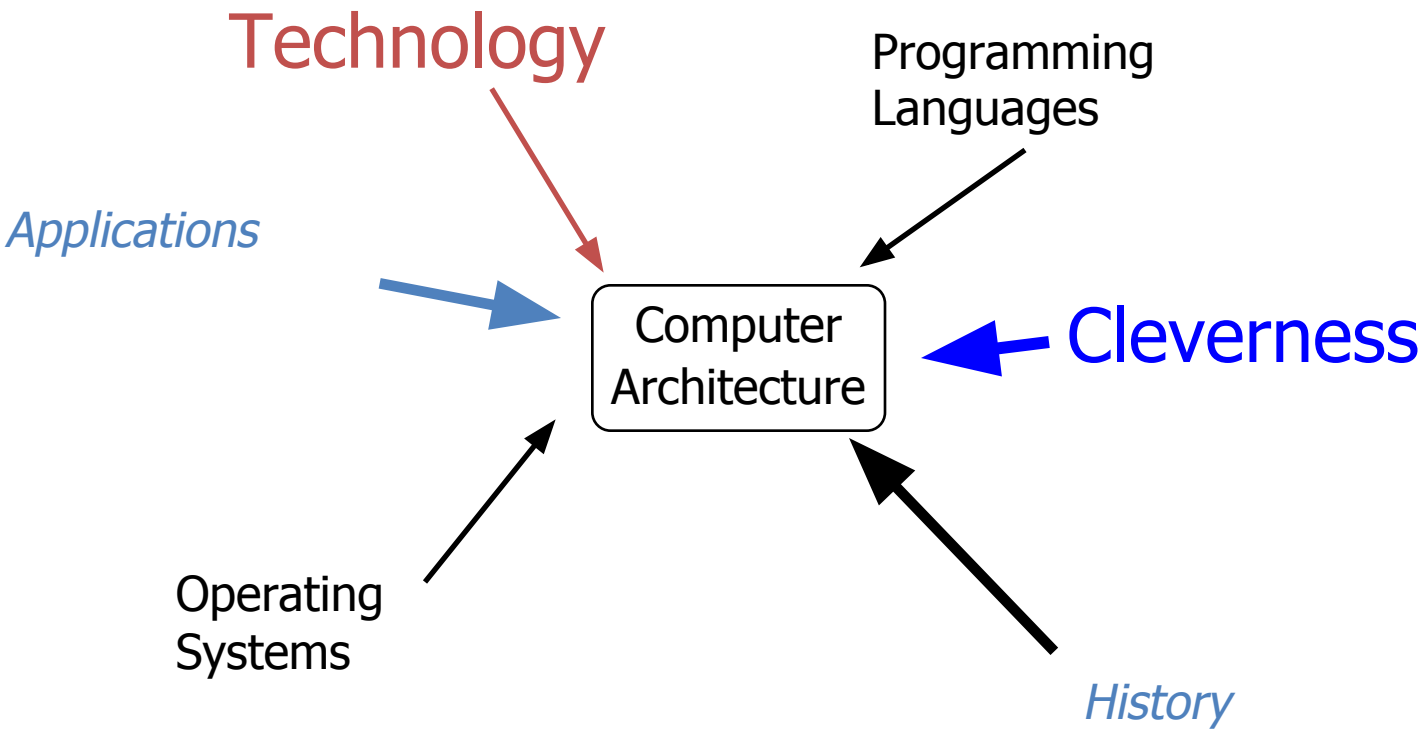
The HW/SW Interface




More abstract, machine-independent;
easier to write, read, debug, or maintain
More concrete, machine-specific, error-prone;
harder to write, read, debug, or maintain
Very high-level language objectives or tasks
High-level language statements
Assembly language instructions, mnemonic
Machine language instructions, binary (hex)
Interpreter
Compiler
Assembler
Swap v[i] and v[i+1]
temp=v[i]
v[i]=v[i+1]
v[i+1]=temp
add $2, $5, $5
add $2, $2, $2
add $2, $4, $2
lw $15, 0($2)
lw $16, 4($2)
sw $16, 0($2)
sw $15, 4($2)
jr $31
00a51020
00421020
00821020
8c620000
8cf20004
acf20000
ac620004
03e00008
One task = many statements
One statement = several instructions
Mostly one-to-one

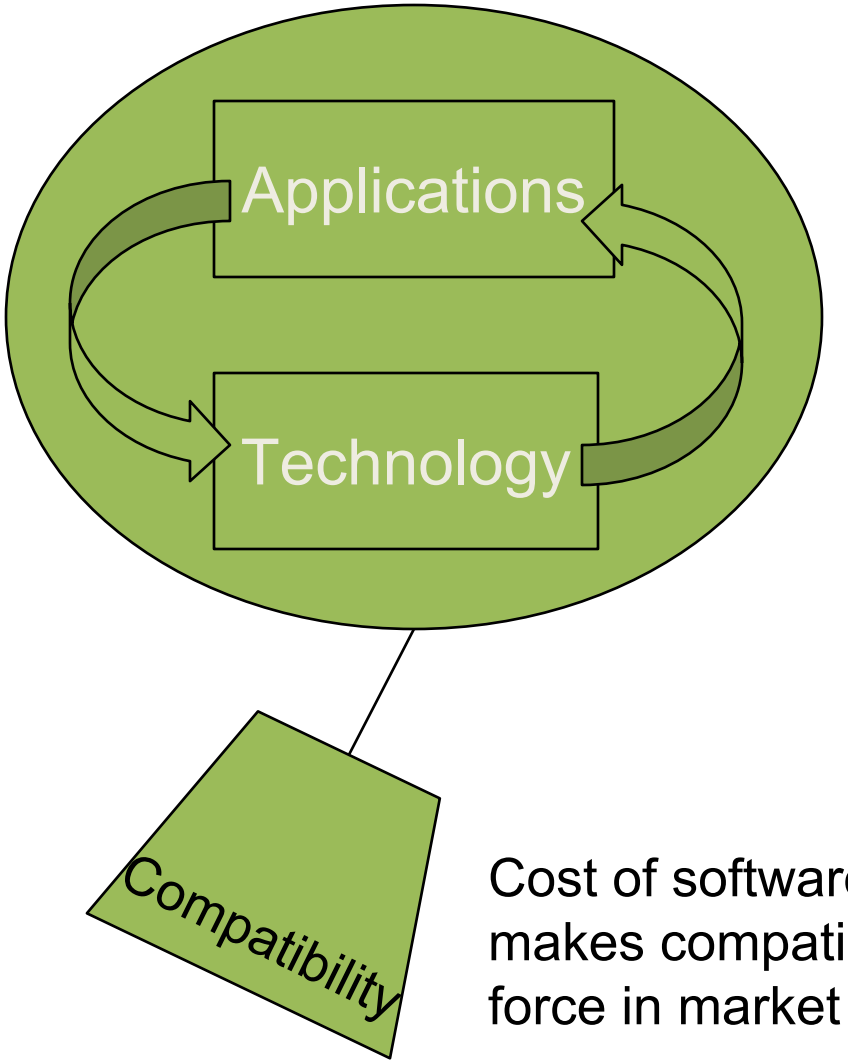

Software
Hardware
Application domains
Application designer
System designer
Computer designer
Logic designer
Circuit designer
Electronic components
High-level view
Low-level view
Computer architecture
Computer organization

Why?


Technology
Programming Languages
Applications
Computer Architecture
Cleverness
Operating Systems
History

Computer Architecture

Applications suggest how to improve technology, provide revenue to fund development



Improved technologies make new applications possible

Cost of software development makes compatibility a major force in market

Why Computer Organization

- Embarrassing if you are a BS in CS/CE and can't make sense of the following terms: DRAM, pipelining, cache hierarchies, I/O, virtual memory, …
- Embarrassing if you are a BS in CS/CE and can't decide which processor to buy: (helps us reason about performance/power), …
- Obvious first step for chip designers, compiler/OS writers
- Will knowledge of the hardware help you write better and more secure programs?

What Does This Mean to a Programmer?

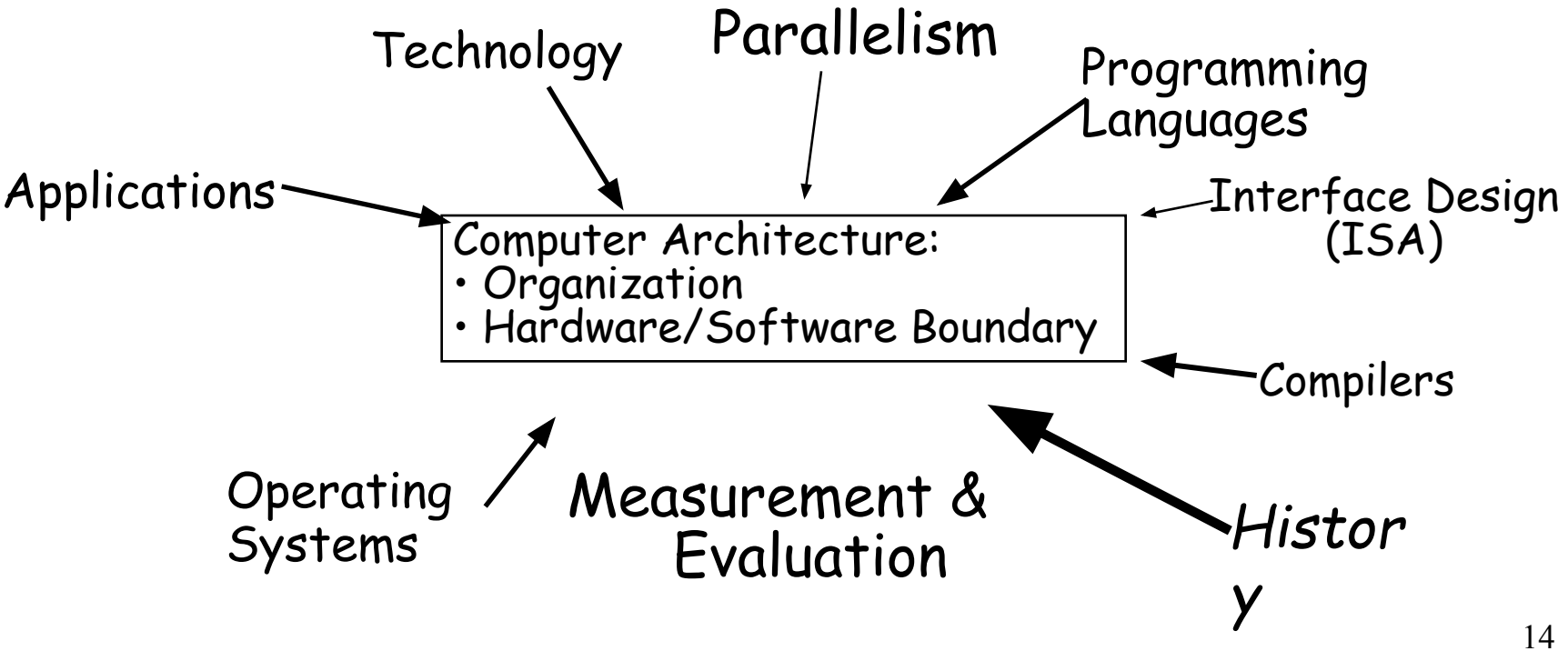
- Today, one can expect only a 20% annual improvement; the improvement is even lower if the program is not multi-threaded
  - A program needs many threads
  - The threads need efficient synchronization and communication
  - Data placement in the memory hierarchy is important
  - Accelerators should be used when possible

Challenges for Hardware Designers

- Find efficient ways to
  - improve single-thread performance and energy
  - improve data sharing
  - boost programmer productivity
  - manage the memory system
  - build accelerators for important kernels
  - provide security

Course Focus

Understanding the design techniques, machine structures, technology factors, evaluation methods that will determine the form of computers in 21st Century



An **operating system** (**OS**) is system software that manages **computer** hardware, software resources, and provides common services for **computer** programs.

In electronic systems and computing, **firmware** is a tangible electronic component with embedded software instructions, such as a BIOS. Typically, those software instructions are used to tell an electronic device how to operate. ... The **firmware** contained in these devices provides the control program for the device

In computing, a compiler is a computer program that translates computer code written in one programming language (the source language) into another language (the target language). The name "compiler" is primarily used for programs that translate source code from a high-level programming language to a lower level language (e.g., assembly language, object code, or machine code) to create an executable program.

A Kernel is a computer program that is the heart and core of an Operating System. Since the Operating System has control over the system so, the Kernel also has control over everything in the system. It is the most important part of an Operating System. Whenever a system starts, the Kernel is the first program that is loaded after the bootloader because the Kernel has to handle the rest of the thing of the system for the Operating System. The Kernel remains in the memory until the Operating System is shut-down.

Why BIOS is used ?

BIOS is known as basic input & output system.

Basic Definition: It is a program used by the processor of the computer to get the system started once it is turned on.

Why BIOS is used ?

The main function of the BIOS is to manage the data flow between computer's operating system and attached devices like hard disk,keyboard,mouse and etc. It is a type of firmware used by the processor during the booting process.

2 FIRMWARE: Firmware is a permanent software or program that is written in the ROM(Read Only memory) during the time of manufacturing.