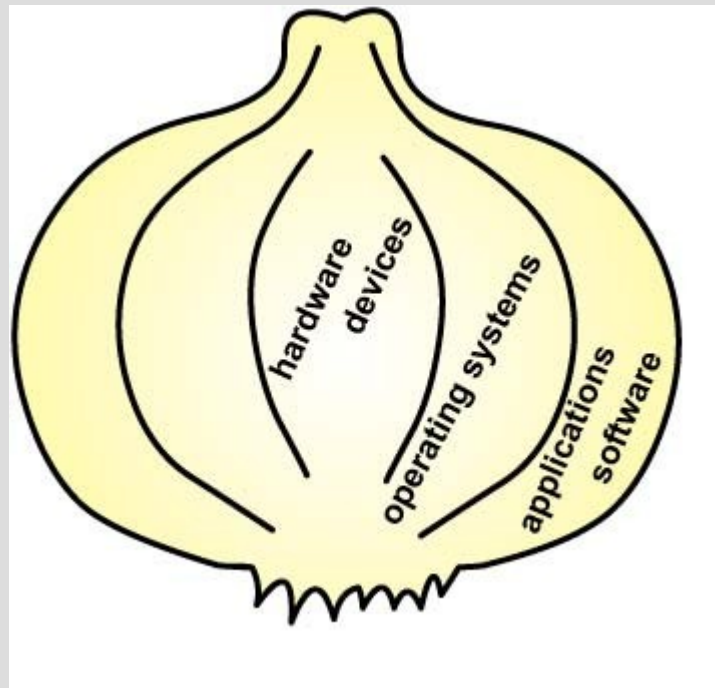


Computer Organization

Lecture 2: Computer as an Onion

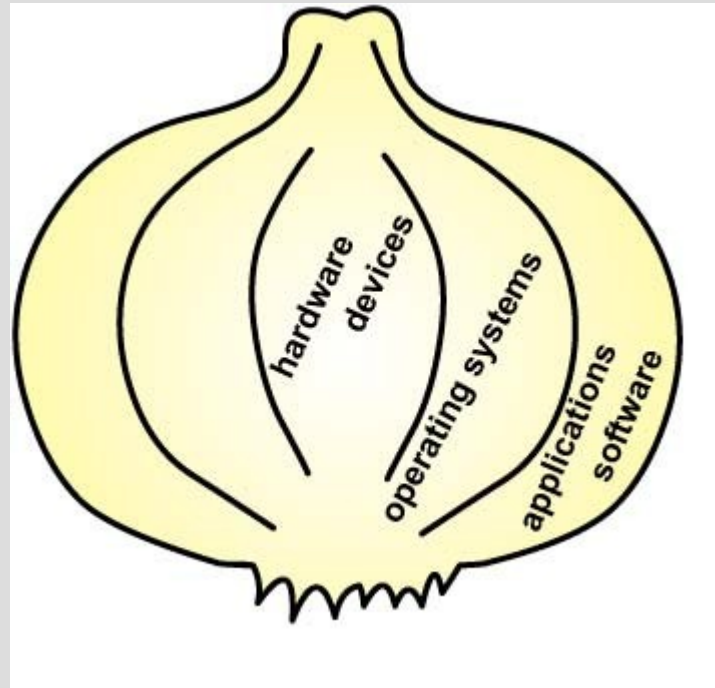


Computer Science and Technology Program
United International College

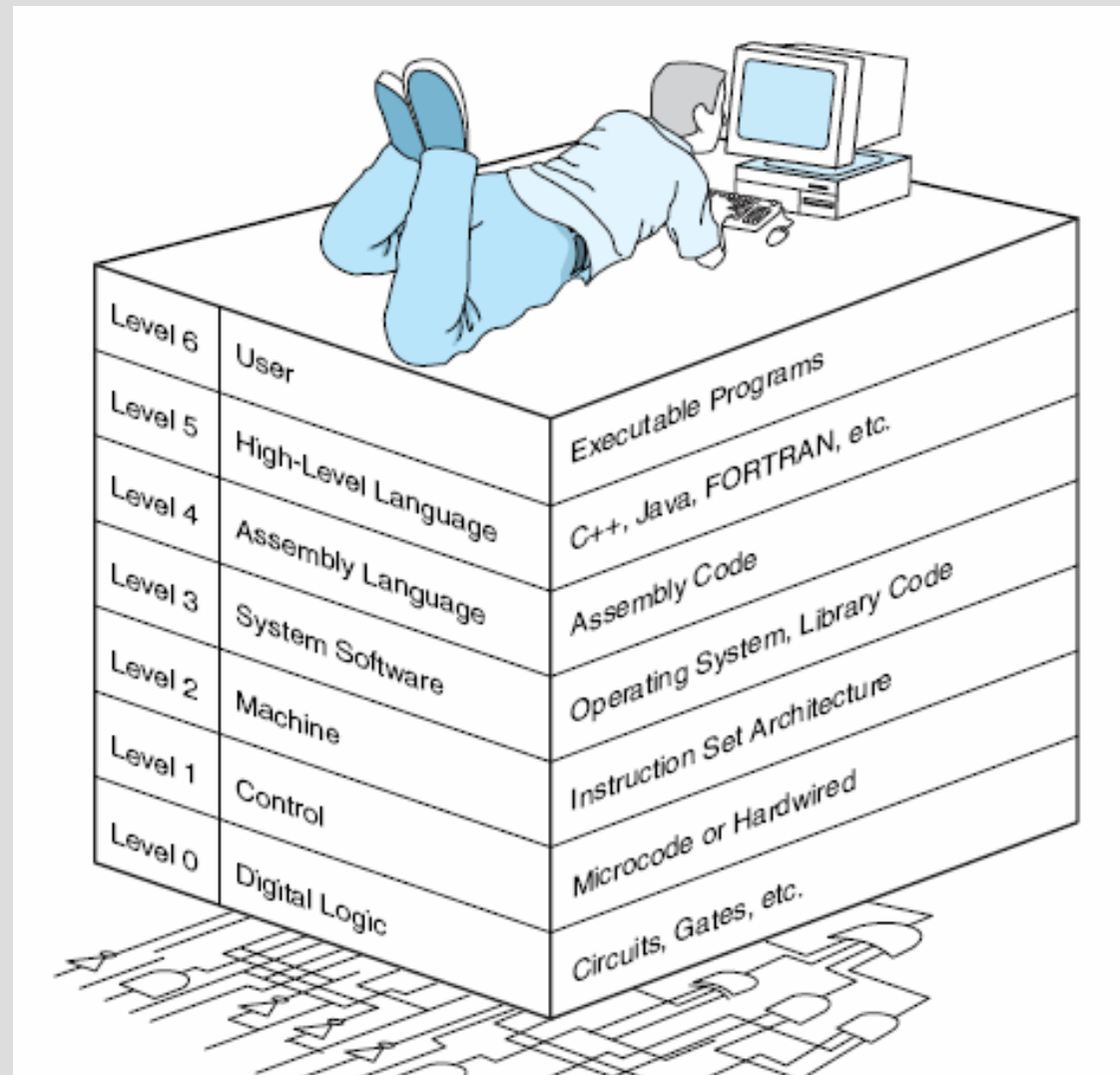
Abstraction

- Abstraction reduces details so that one can focus on a few concepts at a time
- For example, when we write a C program, we don't have to be concerned with the hardware carries out the instructions
- Un-abstraction is the ability to go from the abstraction back to the underlying details

Abstract Model of the Computer



The Abstract Levels of Modern Computing Systems



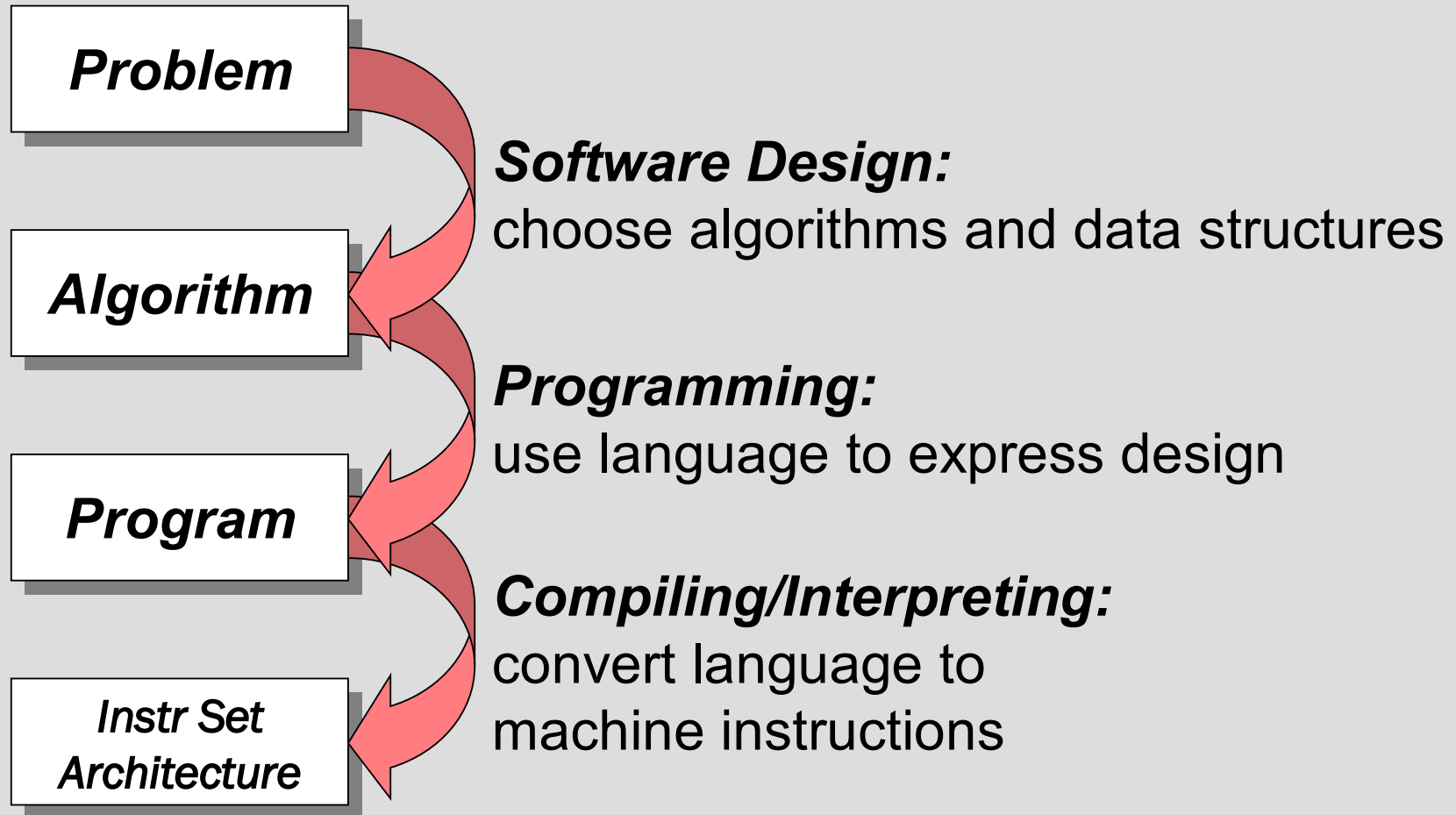
Levels of Abstraction

- User level: applications
- High level language: C, Java
- Assembly language
- Operating system
- Machine Language: ISA
- Control level: micro-code or hardwired
- Digital logic: circuits, gates

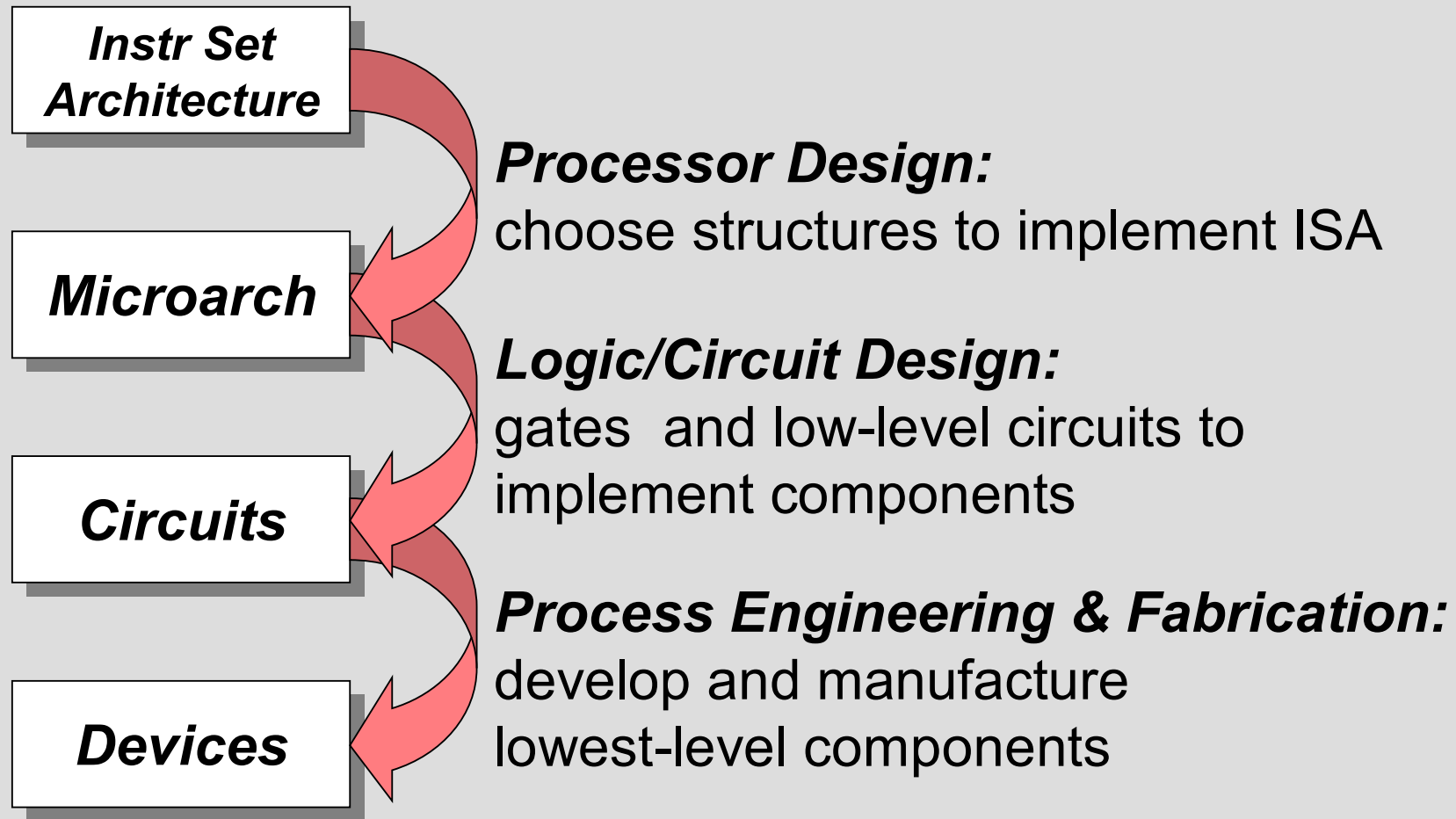
Gap between the User and the Gates

- How can you get the electrons to solve your problem such as $1 + 2 + \dots + 100 = ?$
- There is a gap between what the user wants and what the electrons can do
- Answer: you must go through levels of transformations

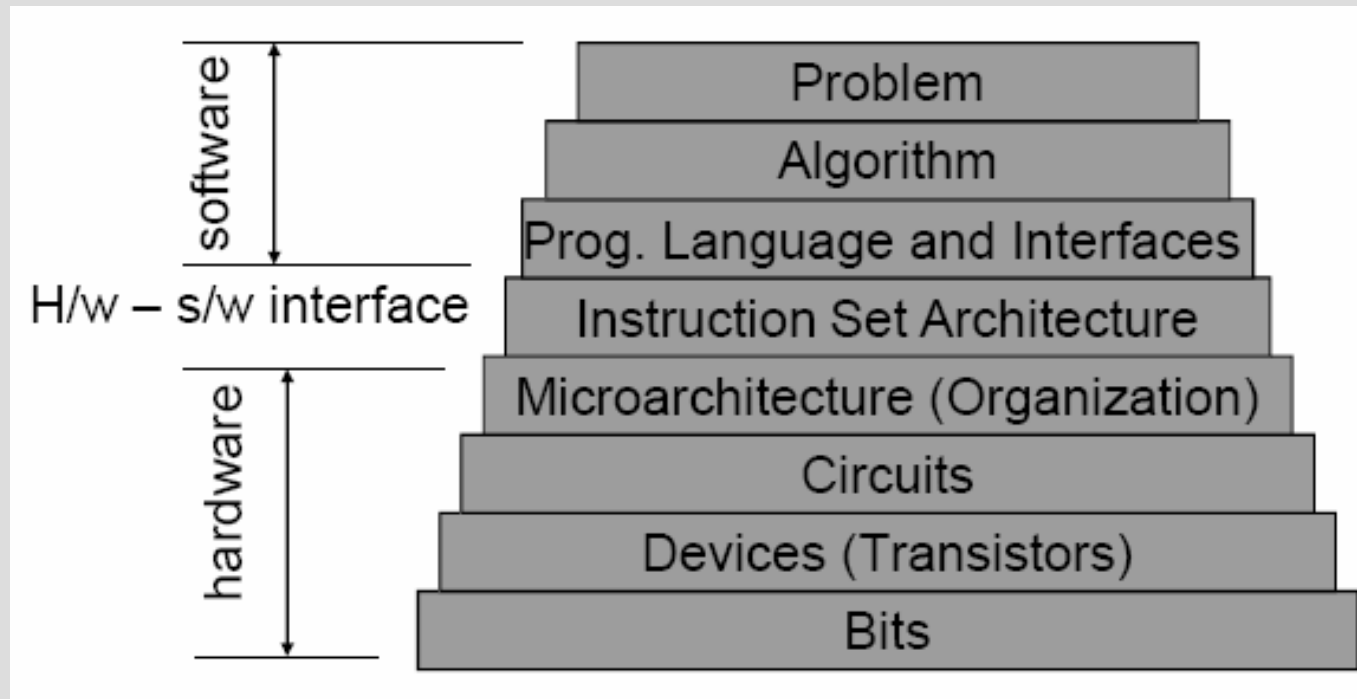
Levels of Transformations



Lower Layers



Layers of Transformations



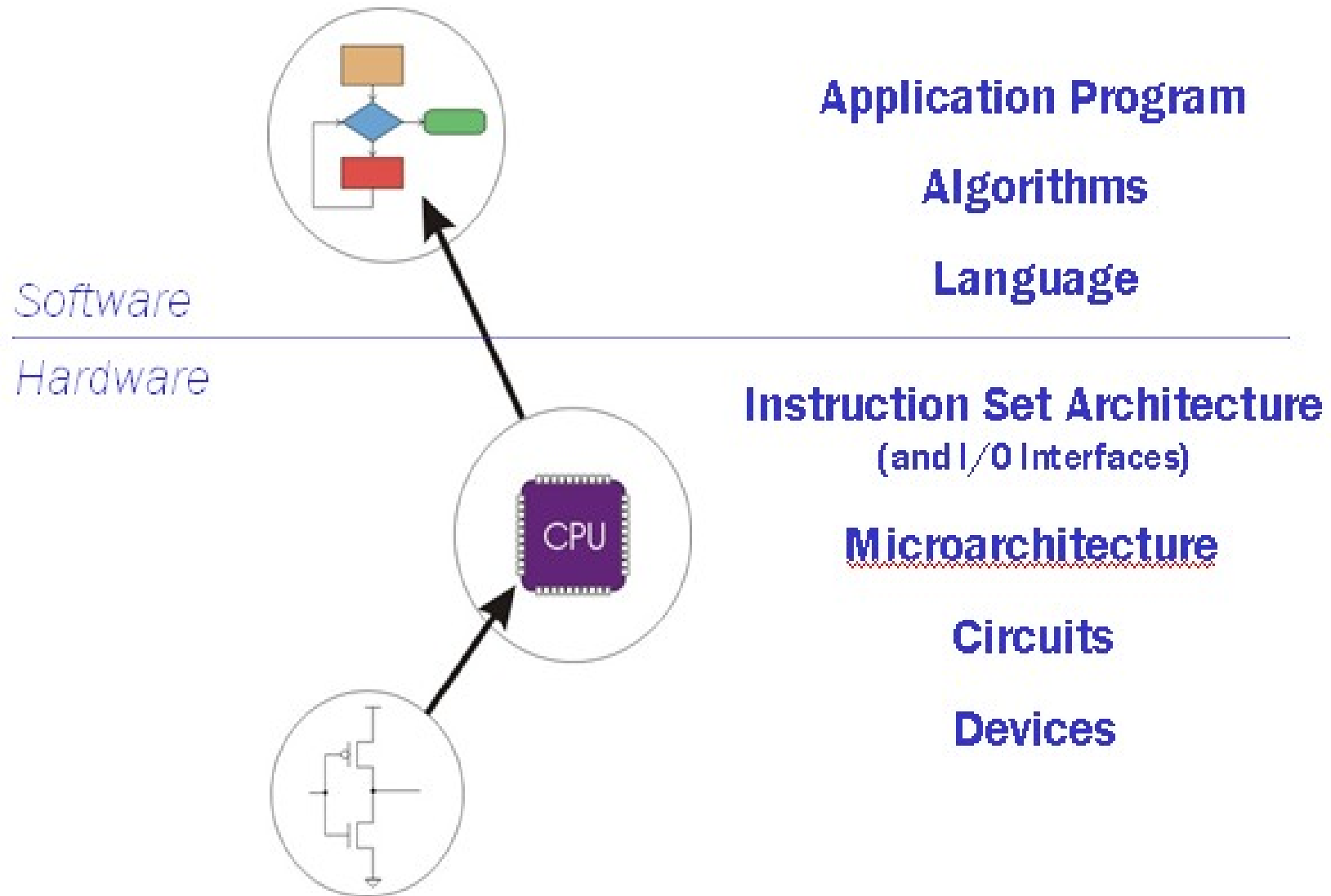
The Program Level

- Problem: What you want to do
- Algorithm: step-by-step procedure to solve problem
- Programming Languages and interfaces: We get a program in this layer

The Machine Level

- Instruction Set Architecture (ISA): instructions that a CPU can do
- Microarchitecture: implementation of ISA
- Circuits: Details of electrical circuits
- Devices (transistors): Circuits are built by interconnecting transistors
- Bits: Transistors operate on bits (“0” or “1”)

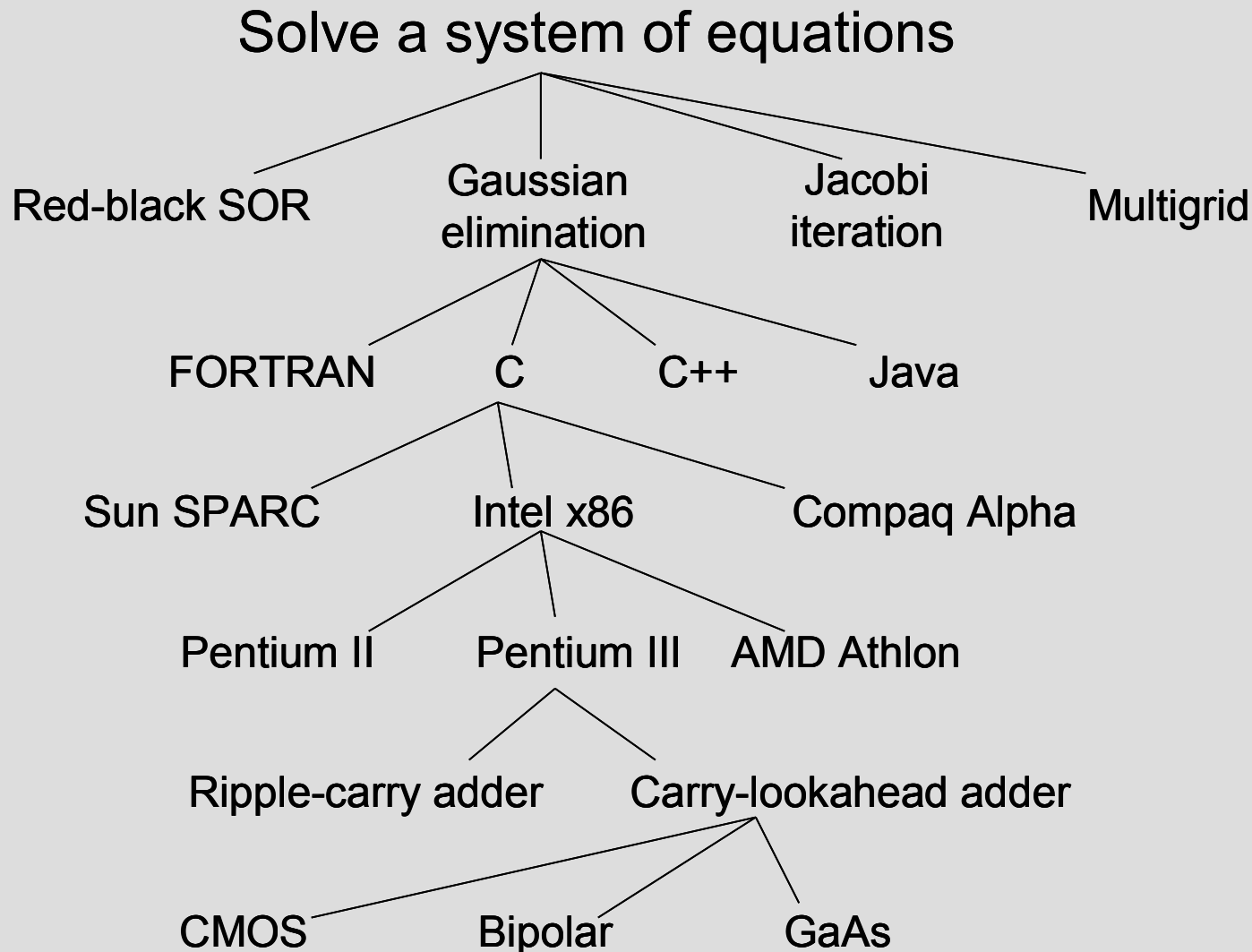
Hardware vs Software



Principle of Equivalence of Hardware and Software

- Anything that can be done with software can also be done with hardware
- And anything that can be done with hardware can also be done with software
- Modern computers are implementations of algorithms that execute other algorithms
- Special-purpose hardware implementations are always faster
- But software implementations are more flexible

Many Choices at Each Level



Tradeoff:
cost,
performance,
flexibility
(etc.)

What's Next in this Course

We start with the bottom level: bits

