MODULE 1: Computer Systems Foundations

Lecture 1.3
Levels of Abstraction

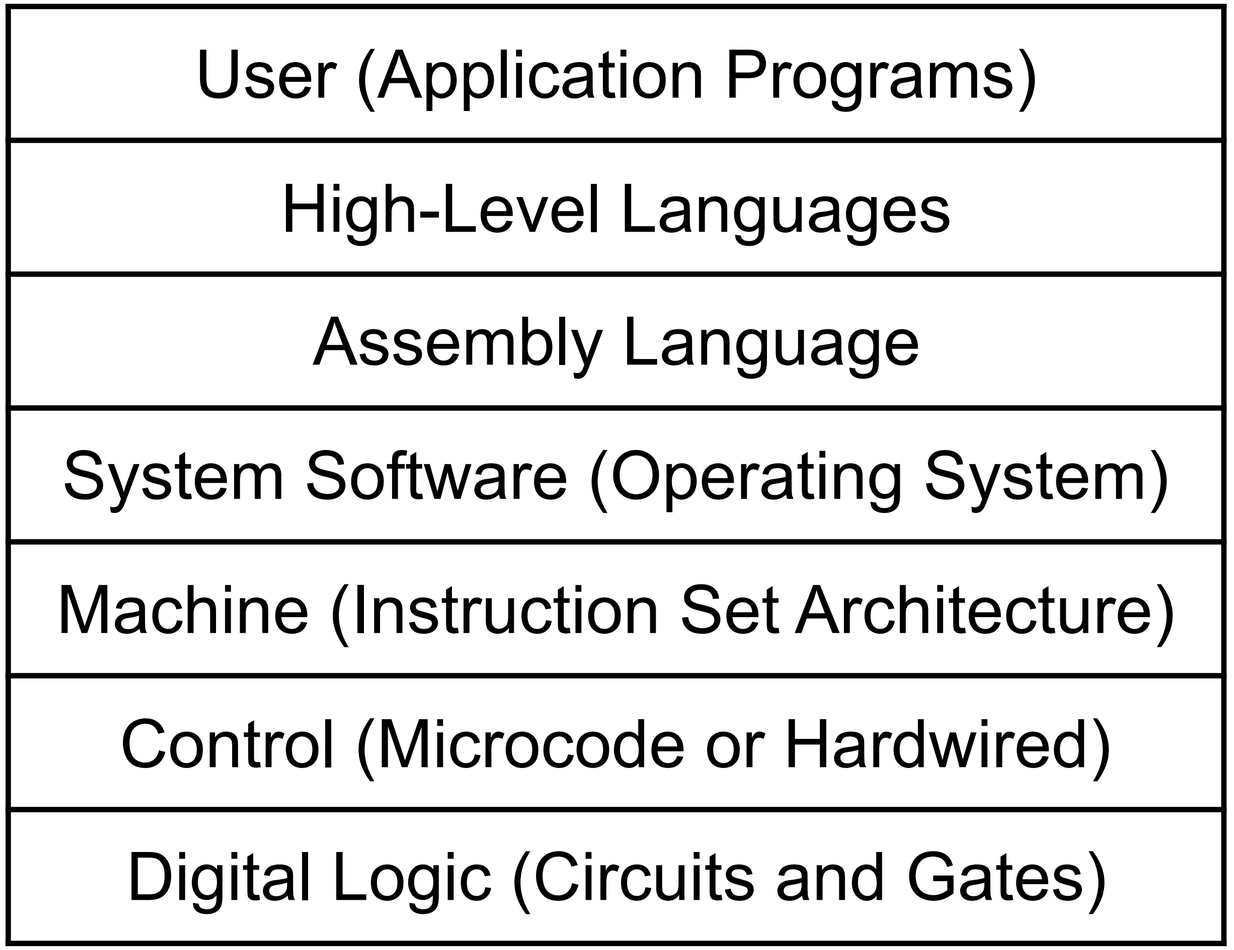
Prepared By:
- Scott F. Midkiff, PhD
- Luiz A. DaSilva, PhD
- Kendall E. Giles, PhD

Electrical and Computer Engineering
Virginia Tech

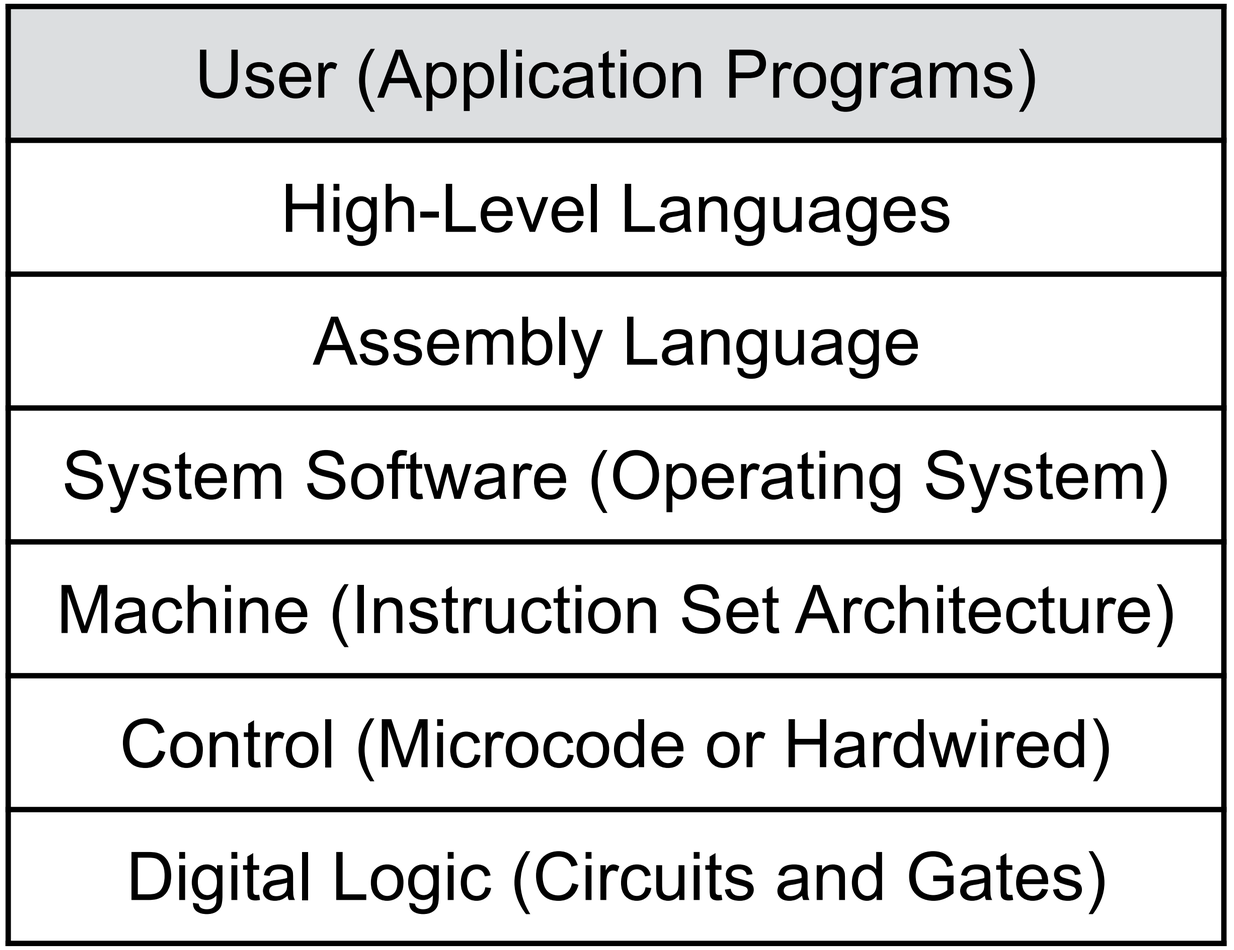
Lecture 1.3 Objectives

- Describe the different levels of abstractions in a computer system
- Explain how each level of a computer system relates to the level above it and the level below it
- Describe the role of computers and network equipment in a networked computer system

Levels of Abstraction in a Computer



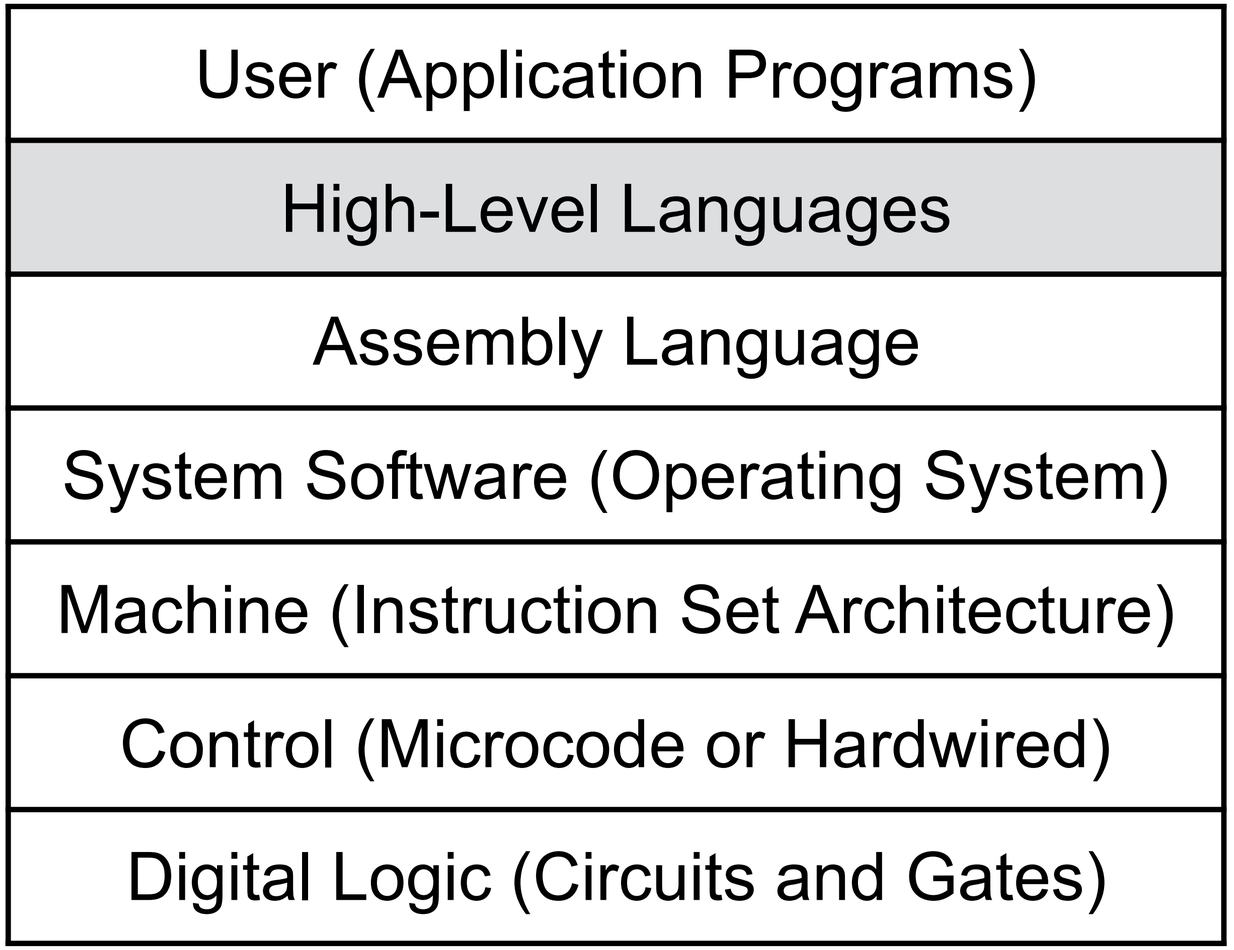
Applications or Executable Programs



Applications or Executable Programs

- Most people are users of application level programs. Thus, the computer is the application, like a web browser, a word processor, or a spreadsheet
- Many IT professionals develop application level programs or integrate application level programs to provide services to users
- The computer hardware and much of the system software is transparent to the user

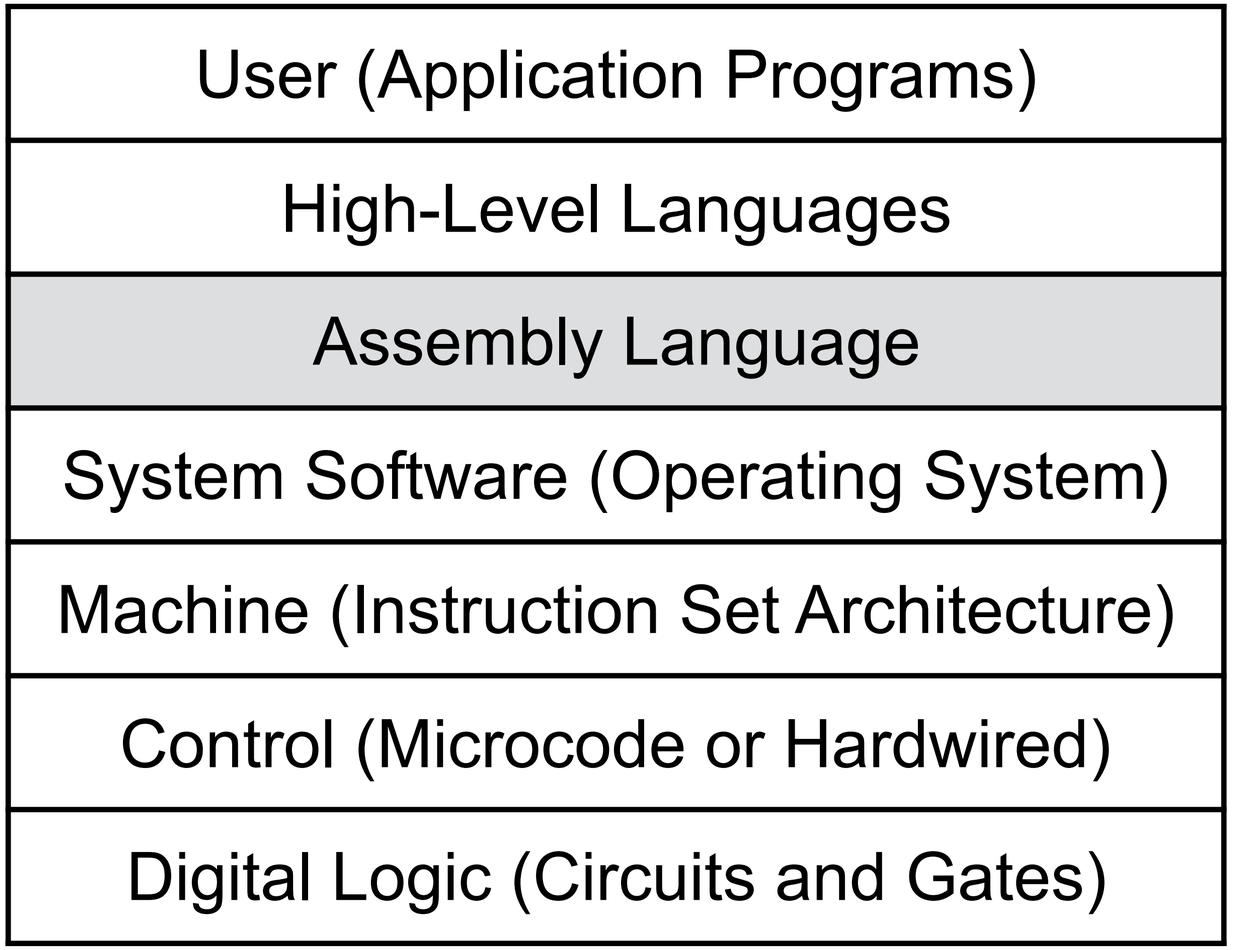
High-Level Languages



High-Level Languages

- The high-level language view of a computer system is the view seen by a programmer using a language like C, C++, Java, or FORTRAN
- Features of the programming language are visible, but not the particular computer being used
  - Some operating system services may be visible
- The same program may be re-compiled to run on different types of computers
  - The compiler maps from the high level language to the specific computer in use

Assembly Language and Machine Code



Assembly Language and Machine Code

- The assembly language or machine code level is the lowest level at which a computer can be programmed
  - Assembly language is a symbolic representation of the program
  - Machine code is the binary representation of the program
- Assembly language and machine code are unique to a particular computer
  - The instruction set architecture (ISA) is visible to the programmer at this level

Assembly Language and Machine Code

- Assembly language programs and machine code cannot be ported from instruction set architecture
- Binary compatibility can be maintained within a processor family, though
  - All features of an Intel 8086 machine code program will run on later Intel processors
  - Many features of a later Intel processor (such as a Xeon) will not execute on an earlier processor (such as the Intel 8086)
- Different hardware designs can implement the same instruction set architecture

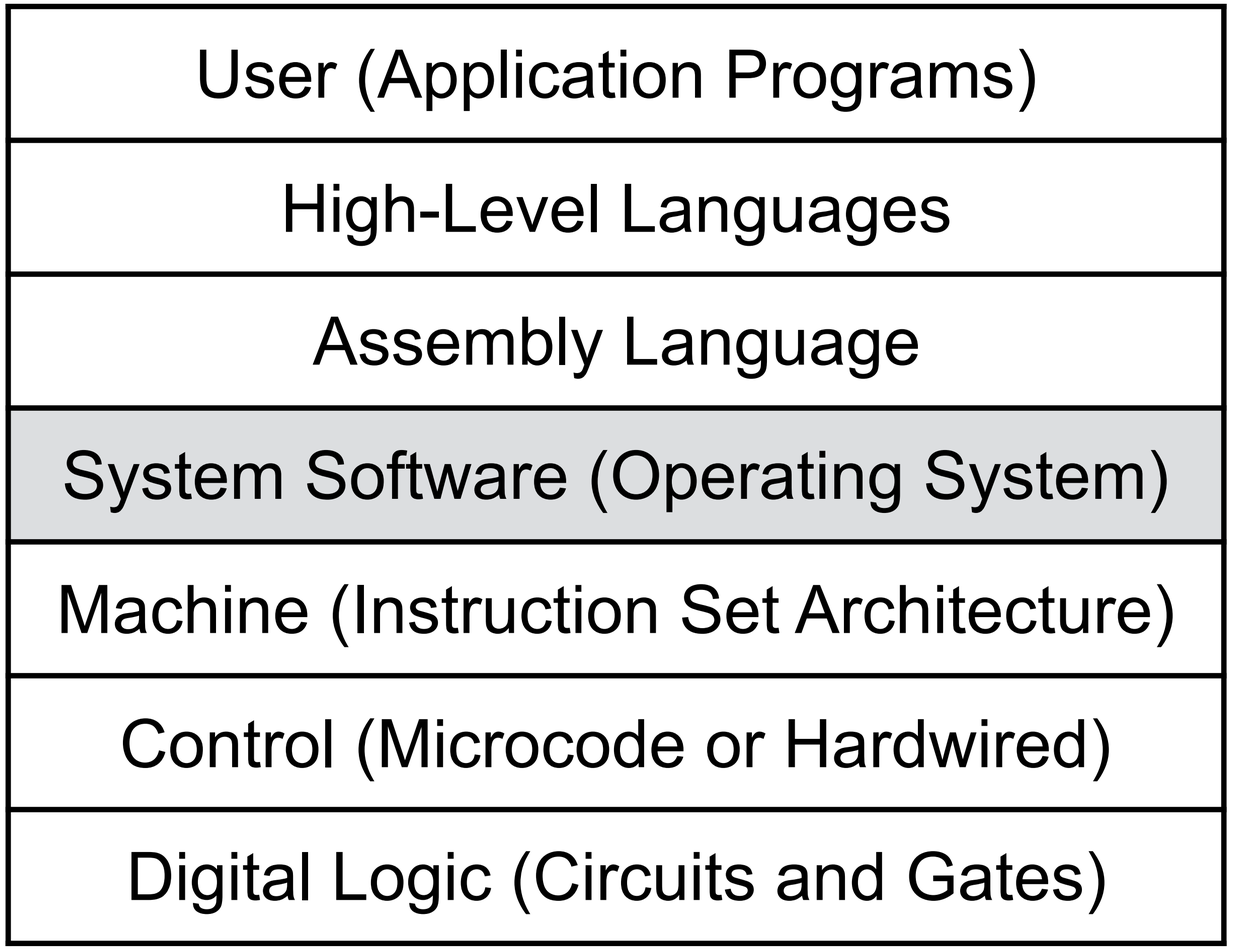
CHECK POINT

As a checkpoint of your understanding, please pause the video and make sure you can:

- Describe what we mean by levels of abstraction for a computer system
- Name and describe the three highest abstraction levels

If you have any difficulties, please review the lecture video before continuing.

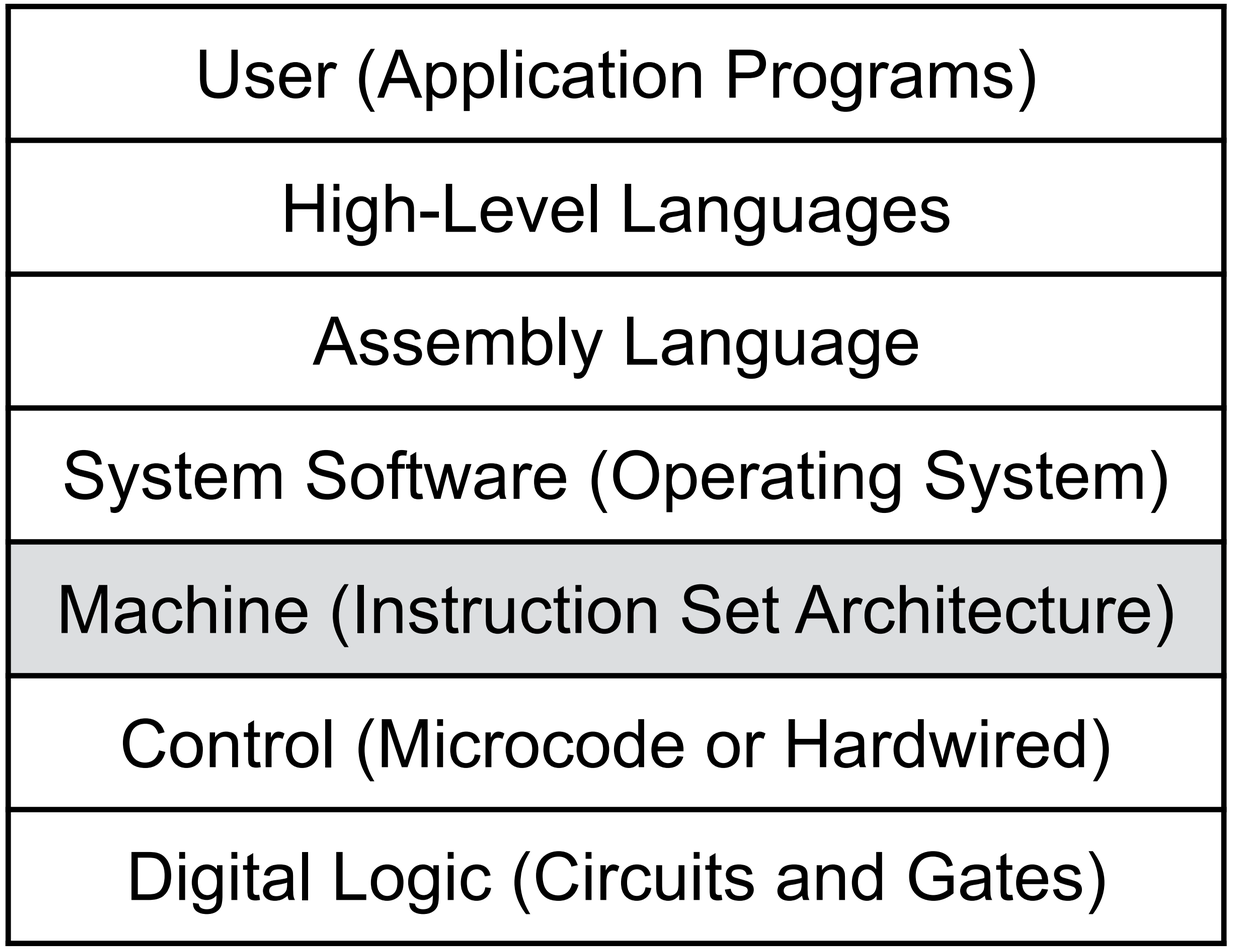
System Software



System Software

- System software, especially the operating system, is responsible for overall system coordination
  - Multiprogramming–allowing multiple programs to execute simultaneously (although usually only one is active at a time)
  - Memory protection–separating different users and protecting system software from user programs
  - Synchronizing processing–ensuring that the proper execution order is maintained, e.g. for input/output
- Machine instructions themselves are largely passed untouched to the instruction set architecture

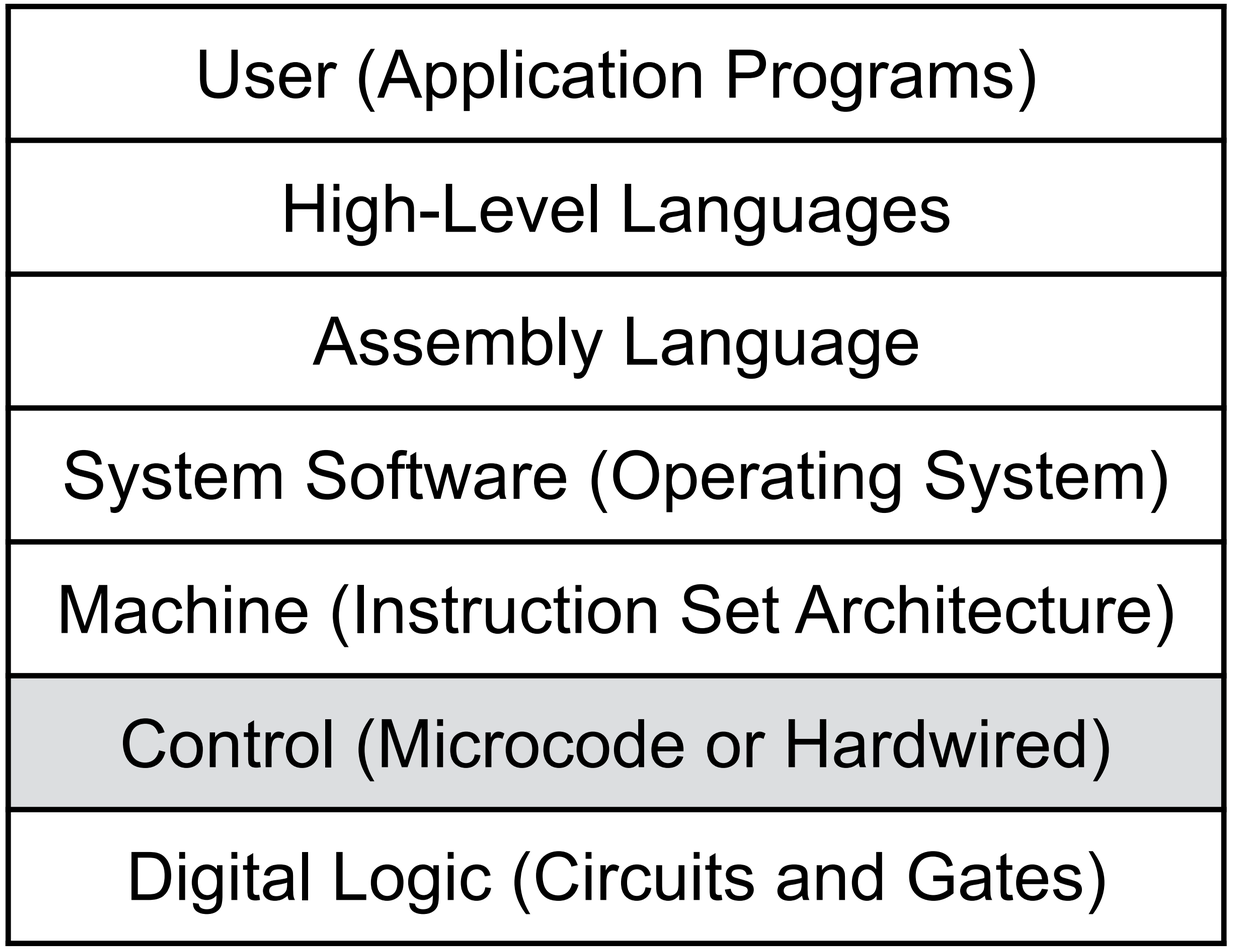
Machine Level



Machine Level

- The machine level realizes the instruction set architecture (ISA)
- The instruction set architecture is the view of the processor as seen by the lowest level programmer, i.e., the assembly language programmer

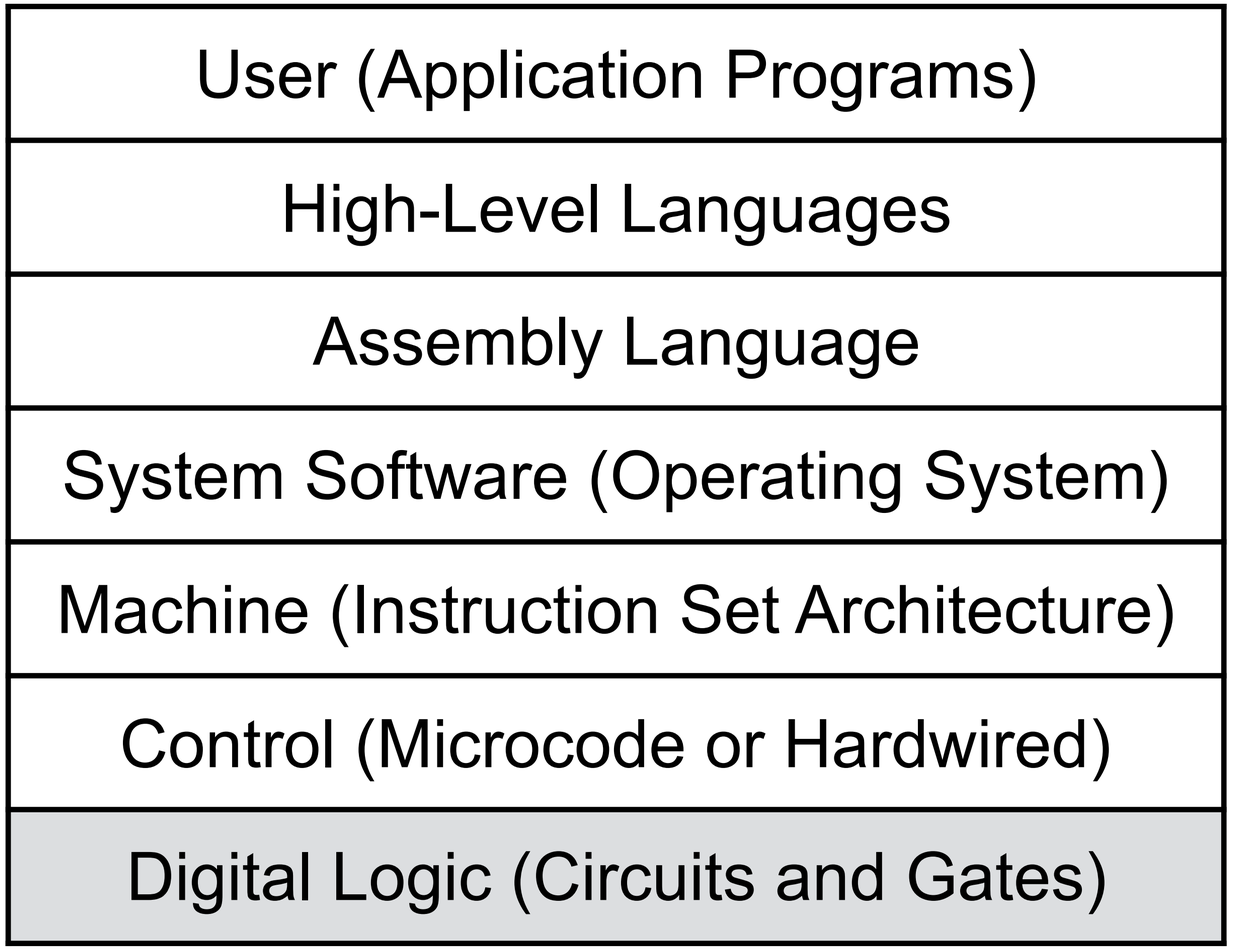
Control Level



Control Level

- The hardware resources are controlled to implement the instruction set architecture
- Functions of control
  - Sequence control signals to enable hardware to fetch and execute instructions
- Forms of control
  - Microprogrammed
  - Hardwired

Digital Logic



Digital Logic

- Digital logic is used to realize the computer's hardware
- Hardware is needed to execute instructions by moving, processing, and storing data and instructions
- Operations are performed by functional units, such as:
  - Arithmetic logic unit (ALU)
  - Multipliers
  - Registers
  - Bus controllers
  - Memory
- Functional units are realized using logic gates and flip-flops

Digital Logic

- Logic gates perform logical operations such as AND, OR, and NOT
- Flip-flops can store one bit (binary digit) of information
  - "1" or "0"
  - TRUE or FALSE
- Multiple logic gates and flip-flops can be used to realize functional units and control logic
- Millions of gates make up contemporary processors

Digital Logic

- Even logic gates, flip-flops, and memory devices are abstractions
- Ultimately, gates, flip-flops, memory must be implemented from transistors (devices) and connected by wires (interconnects)
- Transistors and wires are fabricated from materials such as silicon and aluminum

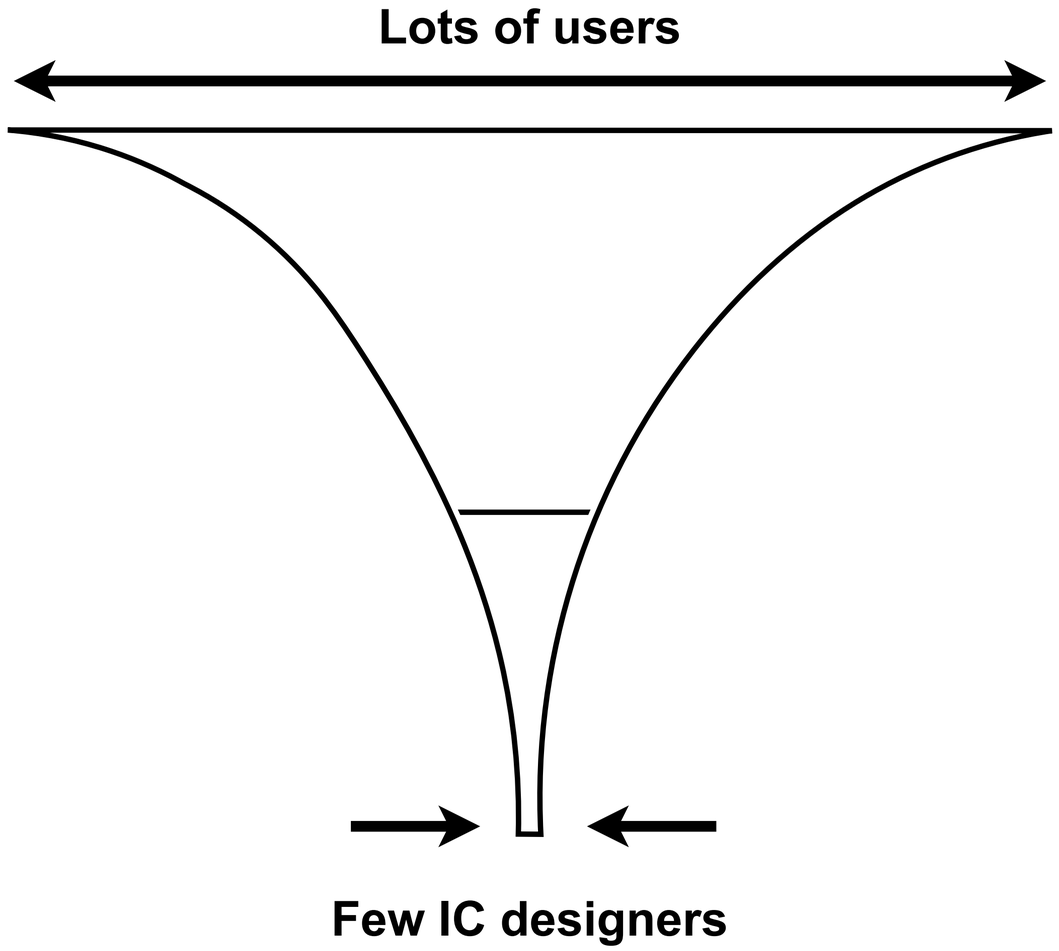
CHECK POINT

As a checkpoint of your understanding, please pause the video and make sure you can:

- Name and describe the bottom four abstraction levels
- Explain how each level of a computer system relates to the level above it and to the level below it

If you have any difficulties, please review the lecture video before continuing.

Human Resources

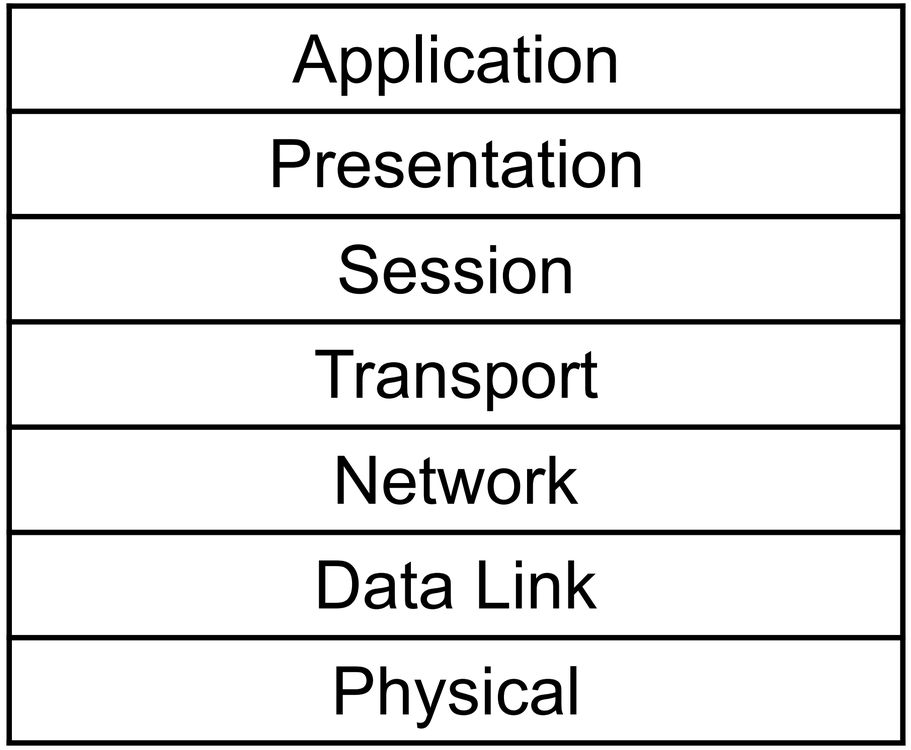


User
High-Level Languages
Assembly Language
System Software
Machine
Control
Digital Logic

Networked Computer Systems

- Today, most general-purpose computers are connected to a network
  - Distributed or client-server applications
  - Internet and intranet access
- The network is becoming an integral part of the computer system
  - System busses need to support high speed input/output for high data rate network interface cards
  - Operating systems need to provide efficient network services
  - Applications are network aware

Network Abstractions



OSI Reference Model

- Network protocols can also be viewed as a layered model
- Different layers perform different functions
- Provide services to layers above
- Use the services of layers below

CHECK POINT

As a checkpoint of your understanding, please pause the video and make sure you can:

- Describe the relative numbers of people who interface with a computer system at the various levels of abstraction
- Describe the importance of networks to today's computer systems and state an example abstraction model used to help deal with the complexity of network systems

If you have any difficulties, please review the lecture video before continuing.

Summary

- A computer system can be viewed at different levels of abstraction
- Application level programs down to digital logic (or even electronic devices and materials)
- Abstractions are essential for designing, building, and using large computers
- Computer systems are often networked
- There are also abstractions of networked computer systems

MODULE 1: Computer Systems Foundations

Lecture 1.3
Levels of Abstraction

Prepared By:
- Scott F. Midkiff, PhD
- Luiz A. DaSilva, PhD
- Kendall E. Giles, PhD

Electrical and Computer Engineering
Virginia Tech