Lecture of Week 4

Programming for Embedded Systems

By: **Dr Vahid Nabaei**

University of Glasgow
UESTC College

Programming Languages for Embedded Systems

Topics & Learning Outcomes

- Overview of Embedded Systems
- Overview of different Languages for Embedded Systems.
- Understanding the basic operation of microprocessor.
- Understanding the basic concept of microprocessor architecture.
- Basics of the machine language programs.
- Designing and writing programs in assembly language.

Programming Languages for Embedded Systems

References and Online Tutorials for Embedded System Programming

- http://www.efxkits.us/classification-of-embeddedsystems/
- http://www.slideshare.net/yayavaram/unit-1-embeddedsystems-
- and-applications
- https://sing.stanford.edu/site/publications/3
- https://en.wikipedia.org/wiki/Embedded_system

There are many free tutorials on programming for embedded systems available online. Some that I found useful are:

1- https://www.youtube.com/watch?v=3V9eqvkMzHA&t=8s

2-https://www.youtube.com/watch?v=gQOv8o5lS2k

3- https://www.youtube.com/watch?v=cZj284kfuE8

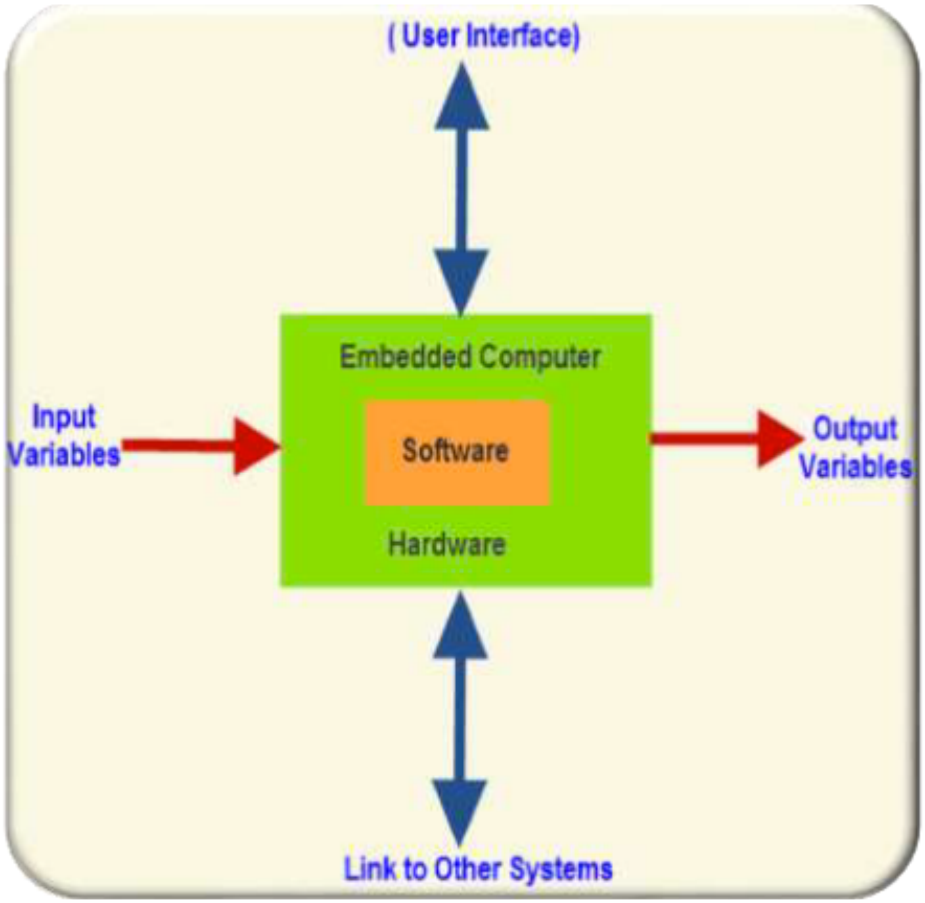
4- https://www.youtube.com/watch?v=o9WpXYBqdPU

5-https://www.youtube.com/watch?v=1Kjh0CAgnl4

What is the Embedded System?

❖ An Embedded system is a combination of computer hardware and software. As with any electronic system, this system requires a hardware platform and that is built with a microprocessor or microcontroller.

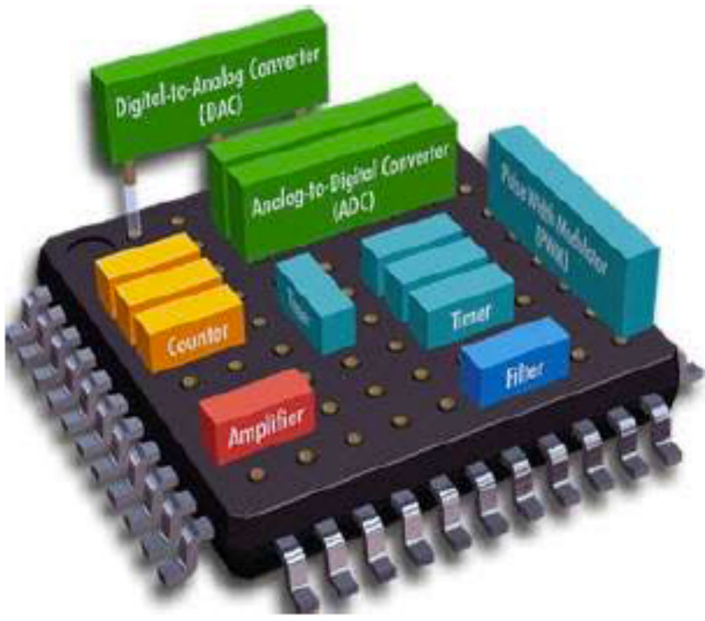
❖ The Embedded system hardware includes elements like user interface, Input/output interfaces, display and memory, etc.



Embedded Programming Languages

In many ways, programming an embedded system is not too dissimilar to coding for a desktop computer, but there are some key differences:

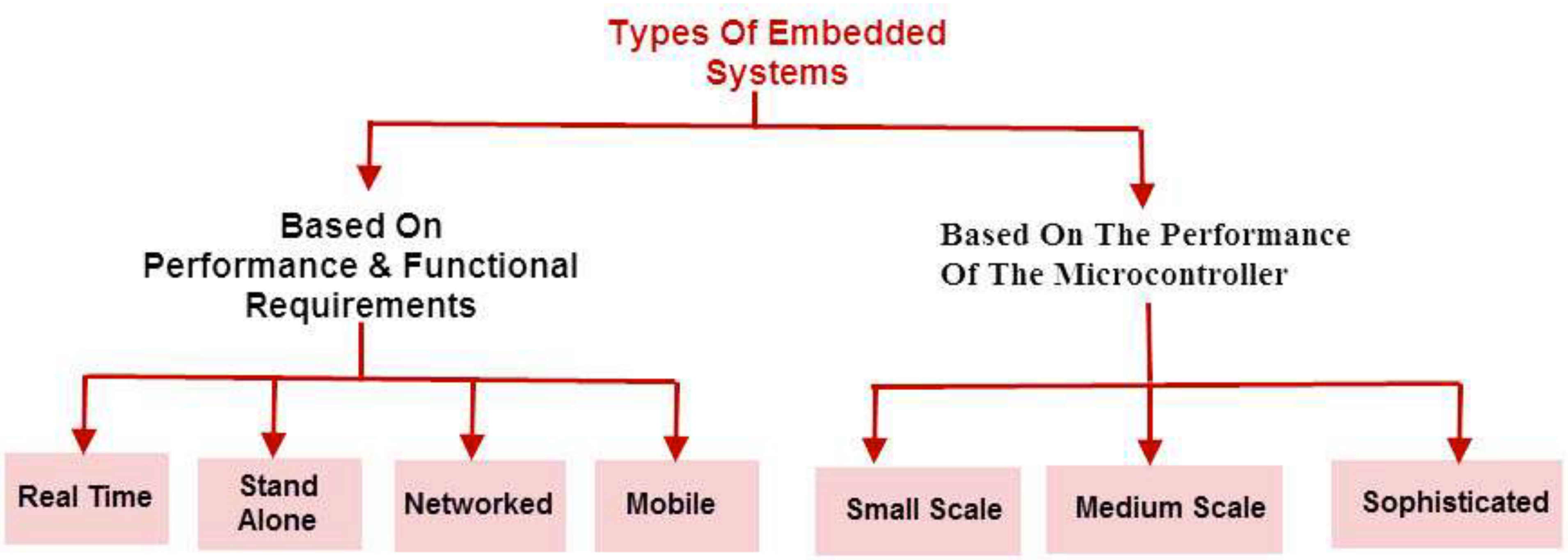
- On an embedded system, resources, memory and CPU power are limited. By comparison, with desktop systems, it is commonly assumed that they have no limits.
- Embedded systems are commonly real-time.
- The number of operating systems in use on desktop computers is quite small.
- To the first approximation, the hardware of all PCs is identical. By comparison, every embedded system is different, so programming close to the hardware is more common.


Digital-to-Analog Converter (DAC)
Analog-to-Digital Converter (ADC)
Pulse Width Modulator (PWM)
Counter
Timer
Filter
Amplifier

Embedded System

Types of Embedded Systems

❖ Embedded systems can be classified into different types based on performance, functional requirements and performance of the microcontroller.


Types Of Embedded Systems
Based On Performance & Functional Requirements
Based On The Performance Of The Microcontroller
Real Time
Stand Alone
Networked
Mobile
Small Scale
Medium Scale
Sophisticated

Real-time Embedded Systems

Real-time Embedded Systems

- This type of embedded system may be component of large system in which it is embedded, such a component called embedded system.
- The connection can be any wired or wireless.
- This type of embedded system is the fastest growing area in embedded system applications.
- The embedded web server is a type of system wherein all embedded devices are connected to a web server and accessed and controlled by a web browser.
- Example: Auto mobiles, vehicle model and controller design.

Real-time Programming Languages

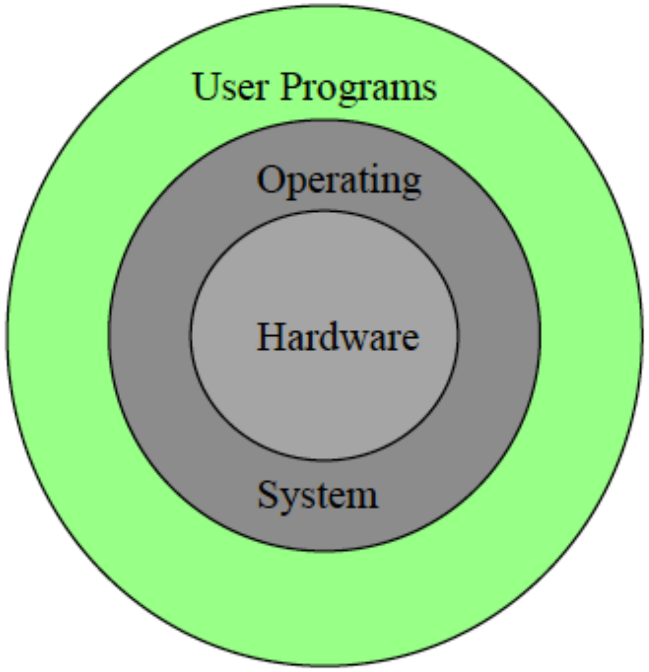
Real-time Programming Languages

- Components of large systems
- Assembly languages
- Sequential systems implementation languages — e.g. RTL/2, Coral 66, Jovial, C.
- Normally require operating system support.
- High-level concurrent languages. Impetus from the software crisis. e.g. Ada, Modula-2, Mesa, Java.

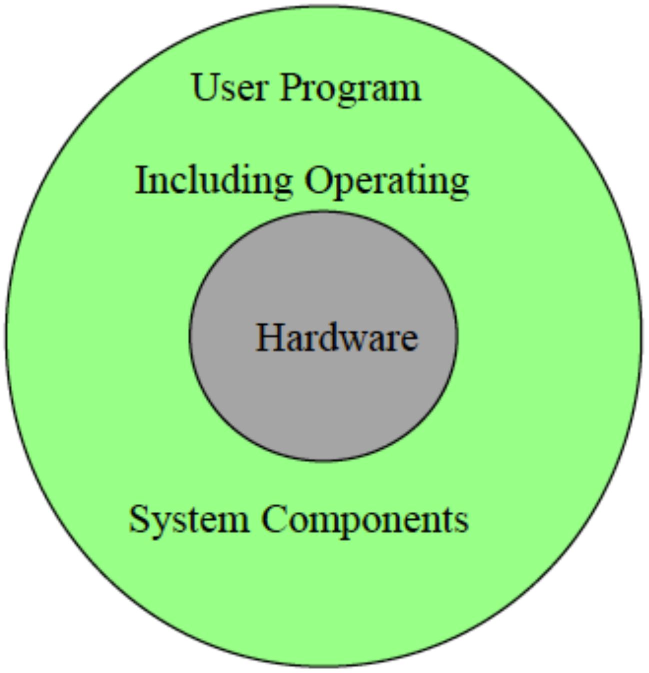
Most Common languages:

- Java/Real-Time Java
- C and Real-Time POSIX
- Ada 95
- Modula-1 for device driving

Real-Time Languages and OS's



Typical OS Configuration



Typical Embedded Configuration

Stand Alone Embedded Systems

- Stand alone embedded systems do not require a host system like a computer, it works by itself.
- It takes the input from the input ports (either analog or digital) and processes, calculates and converts the data and gives the resulting data through the connected device-Which either controls, drives and displays the connected devices.
- Examples for the stand alone embedded systems are mp3 players, digital cameras, video game consoles, microwave ovens and temperature measurement systems.
- **Languages used in:**
- **Java:**

It provides a technique known as “native methods", where c/c++ or assembly code can be called directly from java to manipulate the hardware registers and memory directly using pointers.

This can sometimes be useful in embedded systems.

Network Embedded Systems

- These types of embedded systems are related to a network to access the resources.

- The connected network can be LAN, WAN or the internet.

- The connection can be any wired or wireless.

- This type of embedded system is the fastest growing area in embedded system applications.
- Example for the LAN networked embedded system is a home security system wherein all sensors are connected and run on the protocol TCP/IP.

- **Languages used in:**
- **nesC**
- Networked embedded system is a sensor network, which consists of (potentially) thousands of tiny, low-power 'motes,' each of which execute concurrent, reactive programs that must operate with severe memory and power constraints.

Mobile Embedded Systems

- Mobile embedded systems are used in portable embedded devices like cell phones, mobiles, digital cameras, mp3 players and personal digital assistants.
- The basic limitation of these devices is the other resources and limitation of memory.
- Smart-phones contain several embedded systems, like the modem core and the single-chip WiFi+BT+GPS solutions.
- Smart phones usually have custom-purpose hardware and software for a specific task (i.e. making phone calls and accessing the cellular data network) which technically makes them embedded systems.

Small Scale Embedded System

- These types of embedded systems are designed with a single 8 or 16 bit microcontroller, that may even be activated by a battery.

- For developing embedded software for small scale embedded systems, the main programming tools are an editor, assembler, cross assembler and integrated development environment (IDE).

Language Used In Small- Scale Embedded Systems:

- µITRON
- ITRON is a Japanese open standard for a real-time operating system (RTOS).

Median Scale Embedded System

- These types of embedded systems design with a single or 16 or 32 bit microcontroller, RISCs or DSPs.
- These types of embedded systems have both hardware and software complexities.
- For developing embedded software for medium scale embedded systems, the main programming tools are C, C++, JAVA, Visual C++, RTOS, debugger, source code engineering tool, simulator and IDE

Examples:

- Computer networking systems for example : routers, front end processor in a server.
- Entertainment systems such as video game and a music systems.
- Signal tracking systems such as automatic signal tracking.
- Communication systems such as mobile communication SIM card, cellular phone, a cable TV terminal.
- Image filtering, image processing, pattern recognizer, speech processing and video processing.

Sophisticated Embedded System

- These types of embedded systems have enormous hardware and software complexities, that may need ASIPs, IPs, PLAs, scalable or configurable processors.
- They are used for cutting-edge applications that need hardware and software Co-design and components which have to assemble in the final system.

Examples:

- Multimedia processing systems.
- Wireless LAN and convergent technology devices.
- Interface and networking systems using high speed and ultra high speed and large band width.
- System for space life boats (under development by NASA).

Programing Language for Embedded System

- Embedded systems are application-specific computers that interact with the physical world.

- Each has a diverse set of tasks to perform, and although a very flexible language might be able to handle all of them, instead a variety of problem-domain-specific languages have evolved that are easier to write, analyze, and compile.

Programing Language for Embedded System

Microprocessors & Microcontroller Assembly Language

Programing Language for Embedded System

Overview

- The majority of people think that computers are some kind of complicated device that is impossible to learn and infinitely intelligent, able to think better than a person.
  - The truth is much less glamorous.
- A computer can only do what the programmer has told it to do, in the form of a program.
- A program is just a sequence of very simple commands that lead the computer to solve some problem.
- Once the program is written and debugged, the computer can execute the instructions very fast, and always do it the same, every time, without a mistake.

Overview

- Even though the program consists of very simple instructions, the overall result can be very impressive, due mostly to the speed at which the computer can process the instructions.

- Even though each step in the program is very simple, the sequence of instructions, executing at millions of steps per second, can appear to be very complicated, when taken as a whole.

- The trick is not to think of it as a whole, but as a series of very simple steps, or commands.

Overview

- The microprocessor itself is usually a single integrated circuit (IC).
- Most microprocessors (MPU), or very small computers, have much the same commands or instructions that they can perform.
  - They vary mostly in the names used to describe each command.
- In a typical MPU, there are commands to move data around, do simple math (add, subtract, multiply, and divide), bring data into the micro from the outside world, and send data out of the micro to the outside world.
  - Sounds too simple....right? .

Microprocessors

- The microprocessor is a programmable integrated device that has computing and decision-making capability similar to that of the central processing unit (CPU) of a computer.

- The fact that the microprocessor is programmable means it can be instructed to perform given tasks within its capability.

- The microprocessor is a clock-driven semiconductor device consisting of electronic logic circuits manufactured by using either a large-scale integration (LSI) or very-large-scale integration (VLSI) technique.

Microprocessors

- A typical MPU has three basic parts inside. They are:
  - the Program Counter (PC)
  - Memory, and
  - Input / Output (I/O).
- The Program Counter keeps track of which command is to be executed.
- The Memory contains the commands to be executed.
- The Input / Output handles the transfer of data to and from the outside world (outside the MPU physical package).
- There are many other actual parts inside the MPU, however, we will learn about every single one, one step at a time.

Microprocessors

- Nowadays, the microprocessor is being used in a wide range of products called microprocessor-based products or systems.
- The microprocessor can he embedded in a larger system, can be a stand alone unit controlling processes, or it can function as the CPU of a computer called a microcomputer.

Microprocessors

- The microprocessor communicates and operates in the binary numbers 0 and 1, called bits.
- Each microprocessor has a fixed set of instructions in the form of binary patterns called a machine language.
- It is difficult for humans to communicate in the language of 0 s and 1 s.
- Therefore, the binary instructions are given abbreviated names, called mnenomics, which form the assembly language for a given microprocessor.

Assembly Language for Embedded System

Microprocessors

- A typical programmable machine can be represented with four components: microprocessor, memory, input, and output.
- These four components work together or interact with each other to perform a given task.
- The physical components of this system are called hardware.
- A set of instructions written for the microprocessor to perform a task is called a program, and a group of programs is called software.

Microprocessors

- The microprocessor applications are classified primarily in two categories:
  - reprogrammable systems
  - embedded systems.

Microprocessors

- In reprogrammable systems, such as microcomputers, the microprocessor is used for computing and data processing. These systems include:
  - general-purpose microprocessors capable of handling large data, mass storage devices (such as disks and CD-ROMs), and peripherals such as printers;
  - a personal computer (PC) is a typical illustration.

Microprocessors

- In embedded systems, the microprocessor is a part of a final product and is not available for reprogramming to the end user. Example:
  - copying machine
  - washing machine.
  - Air-conditioner

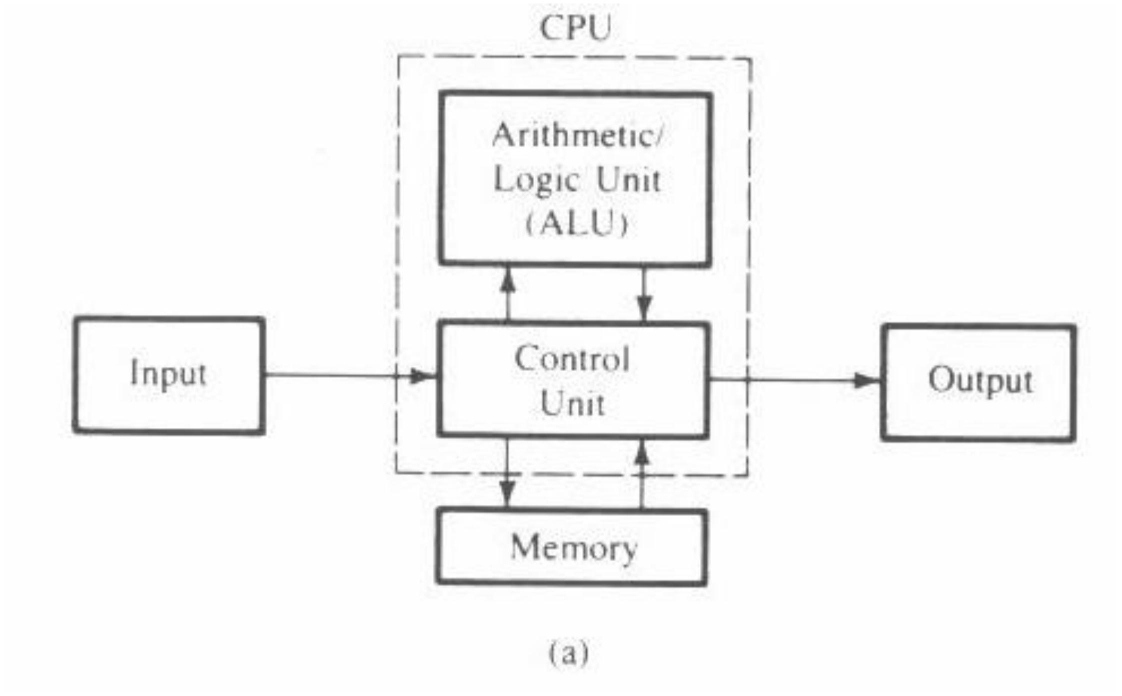
Microprocessor, CPU & Microcontroller

- Microprocessor (MPU) - a semiconductor device (integrated circuit) manufactured by using the LSI technique.
  - It includes the ALU, register arrays, and control circuits on a single chip.
- CPU - the central processing unit.
  - The group of circuits that processes data and provides control signals and timing. It includes the arithmetic/logic unit, registers, instruction decoder, and the control unit.
- Microcontroller - a device that includes microprocessor, memory, and I/O signal lines on a single chip, fabricated using VLSI technology.

Microprocessor, CPU & Microcontroller

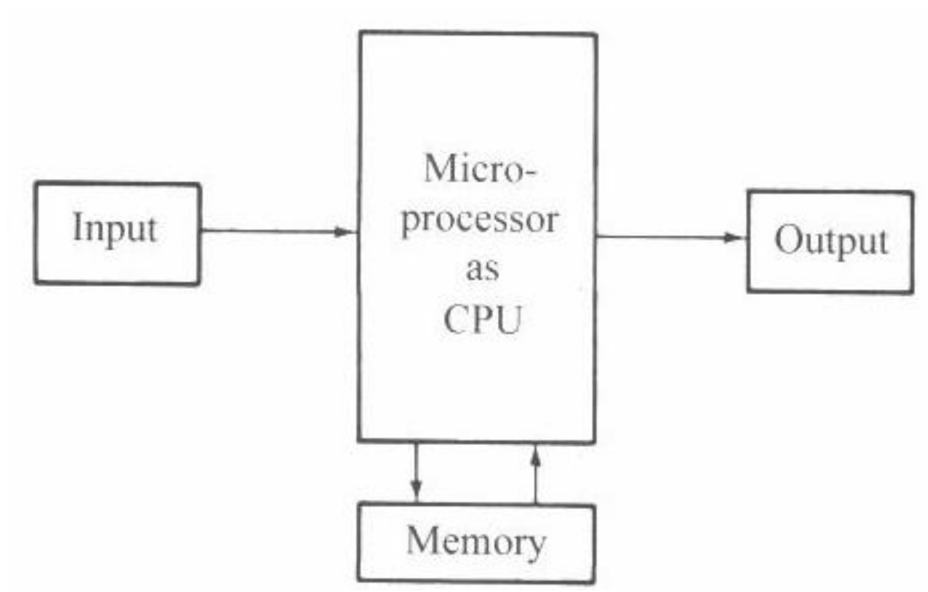
- In large computers, a CPU implemented on one or more circuit boards performs these computing functions.
- The microprocessor is in many ways similar to the CPU, but includes all the logic circuitry, including the control unit, on one chip.

Block diagram of a computer



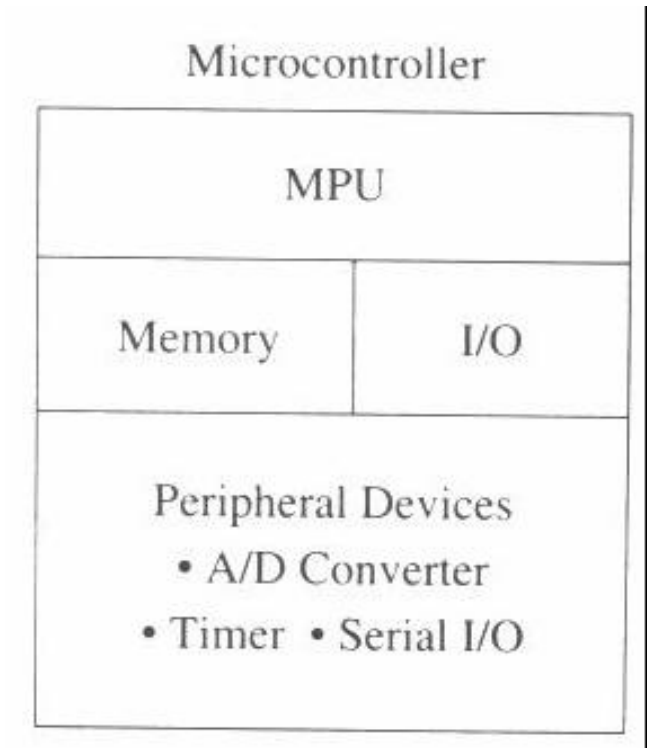
Traditional block diagram of a computer

Block diagram of a computer



Block diagram of a computer with the microprocessor as a CPU

Block diagram of a Microcontroller



Block diagram of a microcontroller

Program

- In a MPU, the problems are different but the logical steps to solve the problem are similar, that is, a series of very simple steps, leading to the solution of a larger problem.
- Also notice that since the steps are numbered, 1 through 11, that is the order in which they're executed.
  - The Program Counter (PC), in this case, starting with 1 and ending with 11, doing what each one says.
  - The PC automatically advances to the next step, after doing what the current step says, unless a branch, or jump, is encountered.
  - A branch is an instruction that directs the PC to go to a specific step, other than the next in the sequence.

Program

- The point of this lesson is to show how a simple set of instructions can solve a bigger problem.
  - Taken as a whole, the solution could appear to be more complicated than any of the separate steps it took to solve it.
- The most difficult problem to be solved in programming a MPU is to define the problem you are trying to solve.
  - This is the Logical Thought Process.
  - It is having a good understanding of the problem you're trying to solve.

Decimal, Binary & Hex

- The microprocessor operates in binary digits, 0 and 1, also known as bits.
  - Bit is an abbreviation for the term binary digit.
  - These digits are represented in terms of electrical voltages in the machine: Generally, 0 represents low voltage level, and 1 represents high voltage level.
- Each MPU recognizes and processes a group of bits called the word.
  - A word is a group of bits the computer recognizes and processes at a time.
- MPUs are classified according to their word length.
  - For example, a processor with an 8-bit word is known as an 8-bit microprocessor, and a processor with a 32-bit word is known as a 32-bit microprocessor.

Decimal, Binary & Hex

- All numbering systems follow the same rules.
- Decimal is Base 10, Binary is Base 2, and Hex(adecimal) is Base 16.
- The base of a system refers to how many possible numbers can be in each digit position.
  - In decimal, a single digit number is 0 through 9.
  - In binary a single digit number is 0 or 1.
  - In hex a single digit number is 0 through 9, A,B,C,D,E, and F.

Decimal, Binary & Hex

- General format to represent number:

$$N = A_nB^n + A_{n-1}B^{n-1} + \ldots\ldots\ldots + A_1B^1 + A_0B^0$$

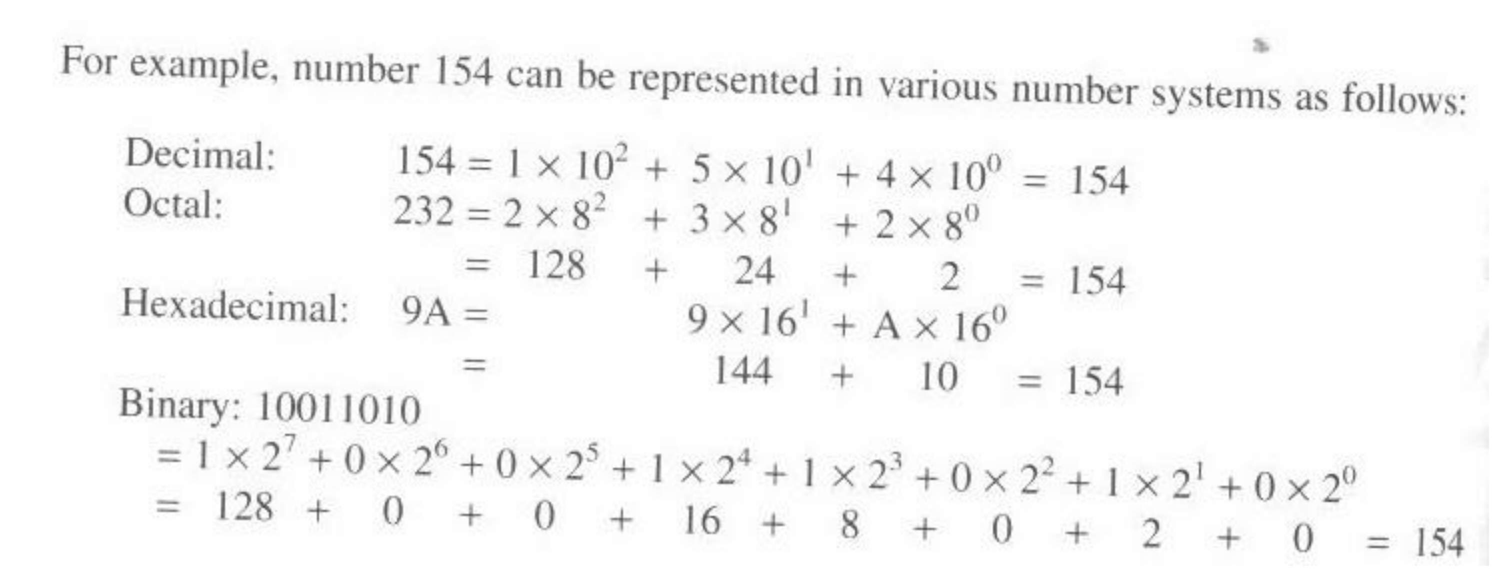
- Where,
- N is number
- B is base
- A is any digit in that base.

A binary 10 (one zero) is decimal 2

A decimal 10 is ten

A hex 10 is decimal 16.

Number Conversion (revision)



For example, number 154 can be represented in various number systems as follows:

Decimal: $154 = 1 \times 10^2 + 5 \times 10^1 + 4 \times 10^0 = 154$

Octal: $232 = 2 \times 8^2 + 3 \times 8^1 + 2 \times 8^0$

$= 128 + 24 + 2 = 154$

Hexadecimal: $9A = 9 \times 16^1 + A \times 16^0$

$= 144 + 10 = 154$

Binary: 10011010

$= 1 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$

$= 128 + 0 + 0 + 16 + 8 + 0 + 2 + 0 = 154$

Number Conversion

- Convert the binary number 1001 1011 into its hex:
  - Arrange the binary digits in groups of four:
  - 1001 1011

  - Convert each group into its equivalent Hex number.
  - $\underbrace{1001}_{9}\ \underbrace{1011}_{B}$

Advances in Semiconductor Technology

- After the invention of the transistor, integrated circuits (ICs) appeared on the scene at the end of the 1950s.
  - an entire circuit consisting of several transistors, diodes, and resistors could be designed on a single chip.
- In the early 1960s, logic gates 7400 series were commonly available as ICs, and the technology of integrating the circuits of a logic gate on a single chip became known as small-scale integration (SSI).

Advances in Semiconductor Technology

- As semiconductor technology advanced, more than 100 gates were fabricated on one chip:
  - medium-scale integration (MSI).
  - Example: a decade counter (7490).
- Within a few years, it was possible to fabricate more than 1000 gates on a single chip
  - large-scale integration (LSI).
- Now we are in the era of very-large- scale integration (VLSI) and super-large-scale integration (SLSI).
- The lines of demarcation between these different scales of integration are rather ill defined and arbitrary.

Historical Perspective

- The microprocessor revolution began with a bold and innovative approach in logic design pioneered by Intel engineer Ted Hoff.
- In 1969, Intel was primarily in the business of designing semiconductor memory.
    - it introduced a 64-bit bipolar RAM chip that year.

Organization of a Microprocessor-Based System

- It includes three components:
    - Microprocessor;
    - I/O (input/output) and
    - memory (read/write memory and read-only memory).
- These components are organized around a common communication path called a bus.
- The entire group of components is also referred to as a system or a microcomputer system.

Assembly Language for Embedded System

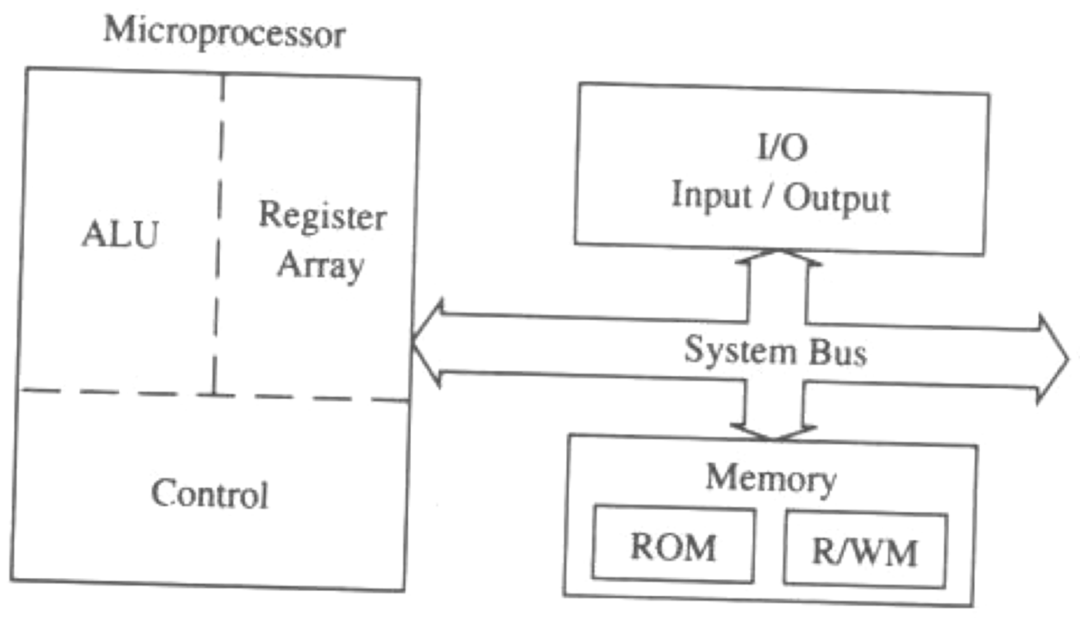


FIGURE 1.3
Microprocessor-Based System with Bus Architecture

Organization of a Microprocessor-Based System

- The functions of various components:
  - The microprocessor
    - reads instructions from memory.
    - communicates with all peripherals (memory and I/Os) using the system bus.
    - controls the timing of information flow.
    - performs the computing tasks specified in a program.

Organization of a Microprocessor-Based System

- The memory
  - stores binary information, called instructions and data.
  - provides the instructions and data to the microprocessor on request.
  - stores results and data for the microprocessor.
- The input device
  - enters data and instructions under the control of a program such as program.
- The output device
  - accepts data from the microprocessor as specified in a program.
- The bus
  - carries bits between the microprocessor and memory and I/Os.

Microprocessor Instruction Set and Computer Languages

- Microprocessors recognize and operate in binary numbers.
- Each microprocessor has its own binary words, meanings, and language.
- The words are formed by combining a number of bits for a given machine.
  - The word (or word length) is defined as the number of bits the microprocessor recognizes and processes at a time.
  - The word length ranges from 4-bit to 64-bit.
- Another term commonly used to express word length is byte.
  - A byte is defined as a group of eight bits.
  - For example, a 16-bit microprocessor has a word length to two bytes.
- The term nibble stands for a group of four bits.
  - A byte has two nibbles.

Microprocessor Instruction Set and Computer Languages

- Each machine has its own set of instructions based on the design of its CPU or of its microprocessor.
- To communicate with the computer, one must give instructions in binary language (machine language).
    - Difficult for most people to write programs in sets of 0s and 1s, computer manufacturers have devised English-like words to represent the binary instructions of a machine - assembly language.
    - An assembly language is machine-specific.

Microprocessor Instruction Set and Computer Languages

- The 8085 is a microprocessor with 8-bit word length:
  - its instruction set (or language) is designed by using various combinations of these eight bits.
  - 8085 has 74 different instructions - instruction set.

Microprocessor Instruction Set and Computer Languages

- For convenience, the 8085 instructions can be written in hexadecimal code and entered in a single-board microcomputer by using Hex keys.
  - E.g., the binary instruction $(0011\ 1100)_2 \equiv (3C)_h$.
  - This instruction can be entered in a single-board microcomputer system with a Hex keyboard by pressing two keys: 3 and C.
  - The monitor program of the system translates these keys into their equivalent binary pattern.

Assembly Language for Embedded System

8085 Assembly Language

- Even though the instructions can be written in hexadecimal code, it is still difficult to understand a program written in hexadecimal numbers.
  - Therefore, each manufacturer of a MPU has devised a symbolic code for each instruction, called a mnemonic.
  - The mnemonic for a particular instruction consists of letters that suggest the operation to be performed by that instruction.
  - For example, $(0011\ 1100)_2$ is represented by the mnemonic INR A.

8085 Assembly Language

- The complete set of 8085 mnemonics is called the 8085 assembly language.
- A program written in these mnemonics is called an assembly language program.
- Machine language and assembly language are microprocessor-specific and are both considered low-level languages.
- The machine language is in binary, and the assembly language is in English-like words; however, the microprocessor understands only the binary.

8085 Assembly Language

- The mnemonics can be written by hand on paper and translated manually in hexadecimal code, called hand assembly.
- Or the mnemonics can be written on a computer using a program called an Editor in the ASCII code and translated into binary code by using the program called an assembler.

- ASCII—American Standard Code for Information Interchange. This is a 7-bit alphanumeric code with 128 combinations. Each combination is assigned to either a letter, decimal digit, a symbol, or a machine command.

Hand Assembly

- To manually write and execute an assembly language program on a single-board computer, with a Hex keyboard for input and LEDs for output, the following steps are necessary:
  - Write the instructions in mnemonics obtained from the instruction set supplied by the manufacturer.
  - Find the hexadecimal machine code for each instruction by searching through the set of instructions.
  - Enter (load) the program in the user memory in a sequential order by using the Hex keyboard as the input device.
  - Execute the program by pressing the Execute key. The answer will be displayed by the LEDs.

Assembler

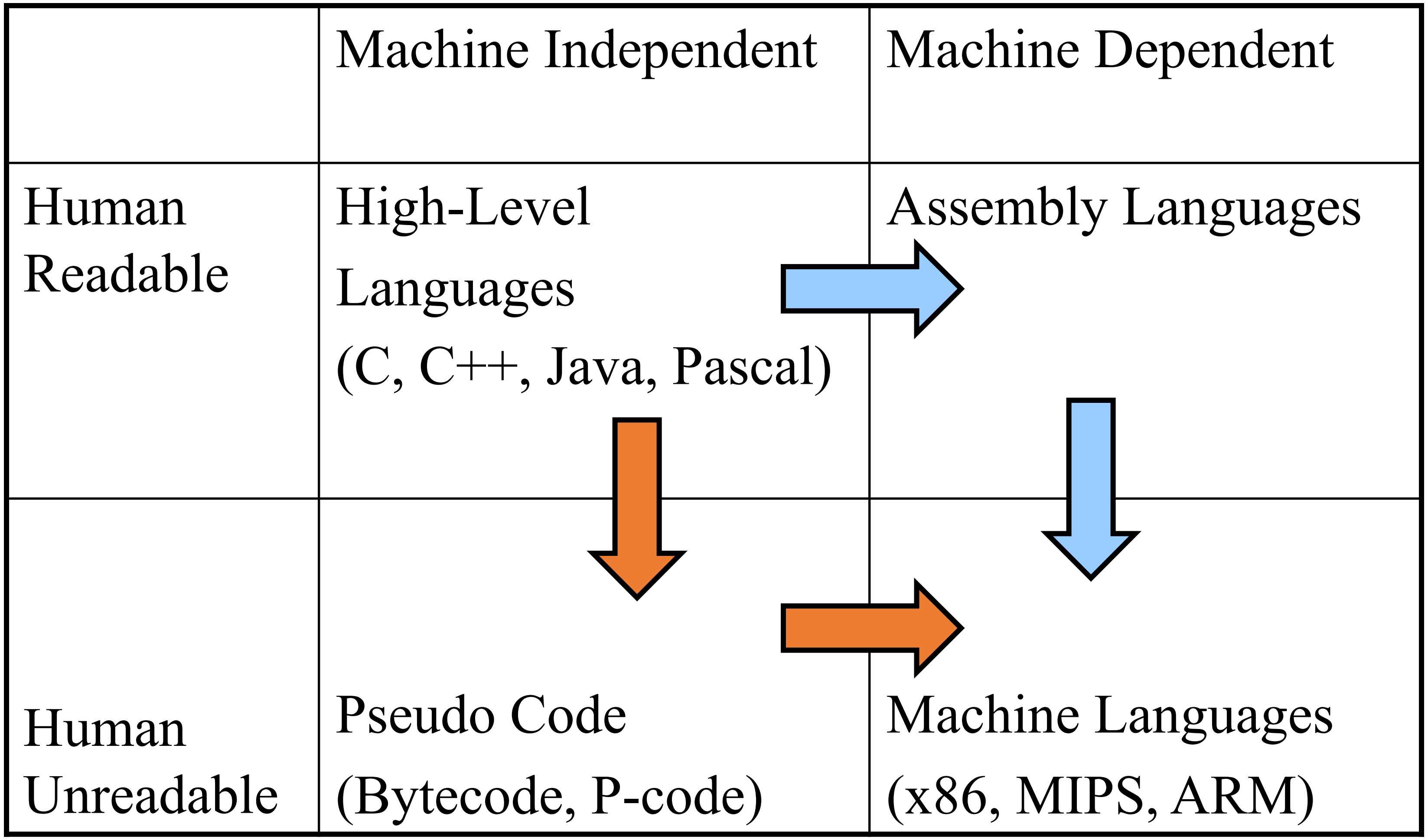
- The hand assembly:
  - tedious and subject to errors;
  - suited for small programs.
- Alternative, use assembler:
  - The assembler is a program that translates the mnemonics entered by the ASCII keyboard into the corresponding binary machine codes of the microprocessor.
  - Each microprocessor has its own assembler because the mnemonics and machine codes are specific to the microprocessor being used, and each assembler has rules that must be followed by the programmer.

High-Level Languages

- How are words in English converted into the binary languages of different microprocessors?
  - Through another program called either a compiler or an interpreter.
  - These programs accept English-like statements as their input, called the source code.
  - The compiler or interpreter then translates the source code into the machine language compatible (object code) with the microprocessor being used in the system.
  - Each microprocessor needs its own compiler or an interpreter for each high-level language.

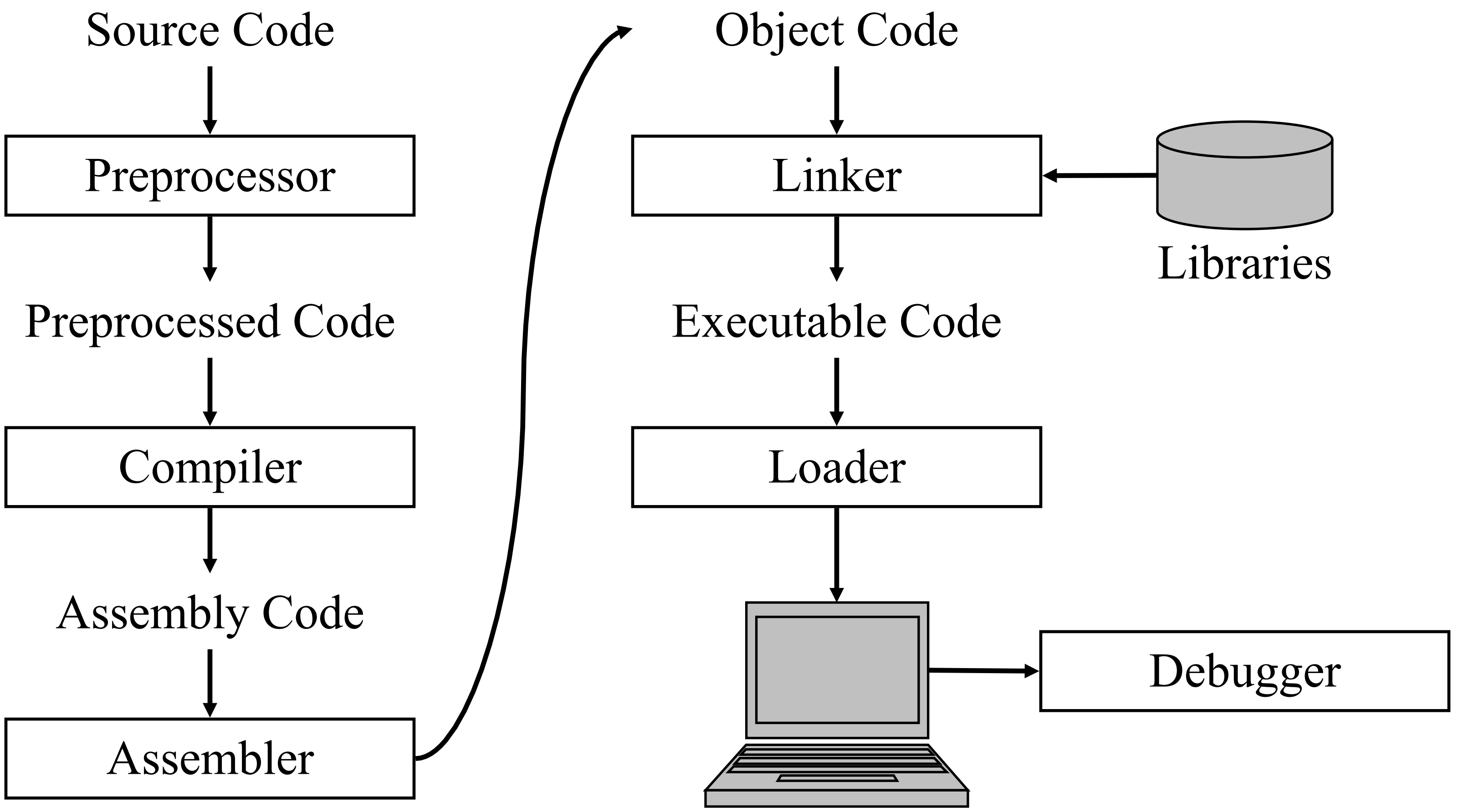
Programing Language for Embedded System

Approaches



	Machine Independent	Machine Dependent
Human Readable	High-Level Languages (C, C++, Java, Pascal)	Assembly Languages
Human Unreadable	Pseudo Code (Bytecode, P-code)	Machine Languages (x86, MIPS, ARM)

From Source to Executable



High-Level Languages

- Compiler - a program that translates English-like words of a high-level language into the machine language of a computer.
  - A compiler reads a given program, called a source code, in its entirety and then translates the program into the machine language, which is called an object code.
- Interpreter - a program that translates the English-like statements of a high-level language into the machine language of a computer.
  - An interpreter translates one statement at a time from a source code to an object code.
- Assembler - a computer program that translates an assembly language program from mnemonics to the binary machine code of a computer.

Question?