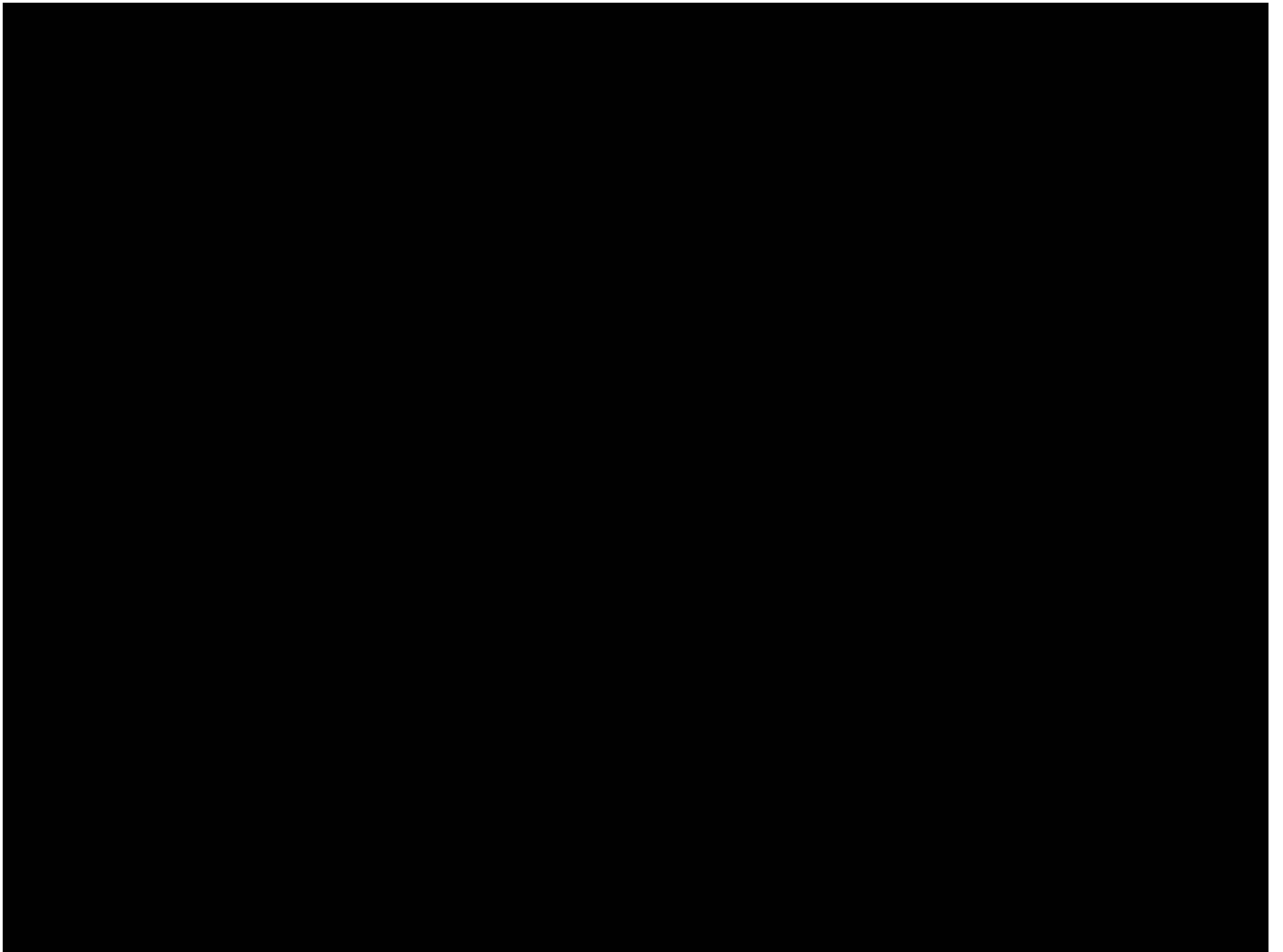
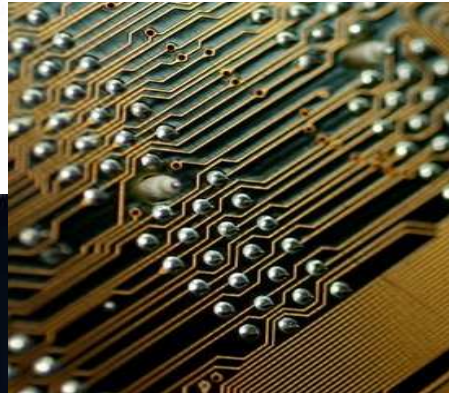






Intro. to Embedded Systems

Ahmed Omran

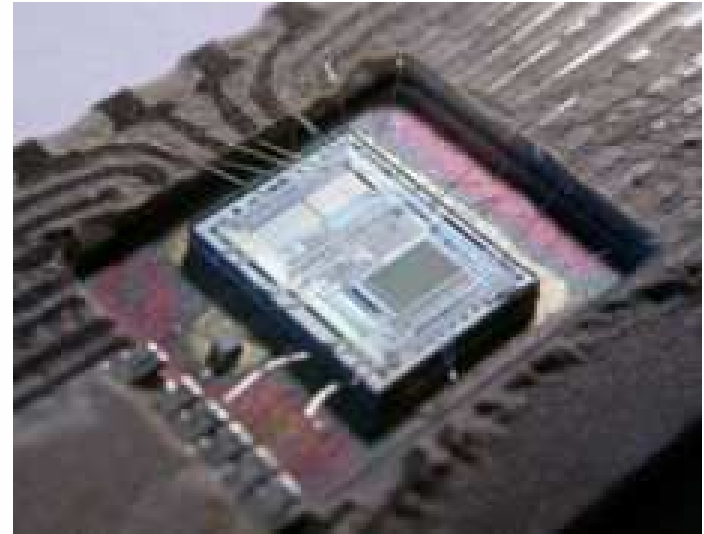


Agenda

- Embedded Systems Vs. General purpose Systems
- Examples of Embedded Systems
- What is a microcontroller?
- Difference between Microprocessor and Microcontroller
- Trade off between HW and SW
- Software characteristics
- Hardware characteristics
- System on chip and system on board
- HW/SW Partitioning
- Advanced Embedded System
- Difference between Emulator and Simulator
- Languages used in embedded systems programming

Embedded Systems Vs. General purpose Systems

- Embedded system is a special-purpose system designed to perform one or a few dedicated functions.
- It is usually **Embedded** as a part of a complete device including hardware and mechanical parts.
- A general-purpose system can do many different tasks depending on programming.
For ex.; microprocessors in PC's.





Example of Embedded Systems

- Embedded systems control many of the common devices in use today

- Digital watches
- Traffic lights
- Printers
- Medical equipments
- Robots
- DSP processing:
 - MP3 player
 - Digital Camera
- Communication:
 - Routers
 - Telephone switches
- Automotive:
 - Motor Engines
 - Climate Control



What is Microcontroller?

- A microcontroller is a single chip devices or single chip computers in a small size that its resources are far more limited than those of a desktop personal computer
- It is designed for standalone operation.
- It includes:
 - Processing Unit.
 - RAM and ROM.
 - I/O.
 - Buses.
 - Peripherals (SPI, PWM,...).





Difference Between MC and MP

Microprocessor

Is a general purpose CPU

Microcontroller

Is a CPU with peripherals

Tradeoff between HW & SW

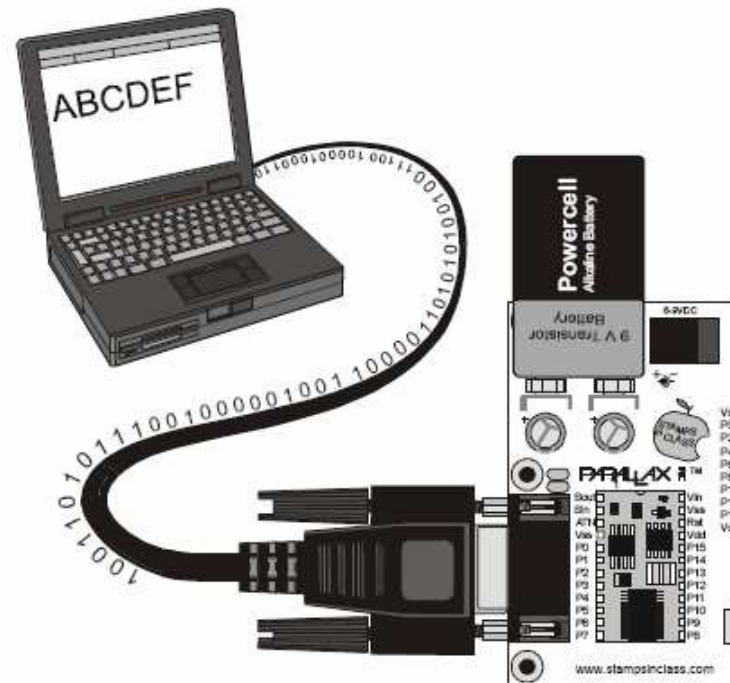
- For a certain application

Which functional blocks should be performed in Hardware??

Which functional blocks should be performed in software??

Software Characteristics

- Highly configurable
- Shorter development cycle
- Easier in versions updates
- Cheaper
- Constrained with processor speed which may satisfy real time application and may not

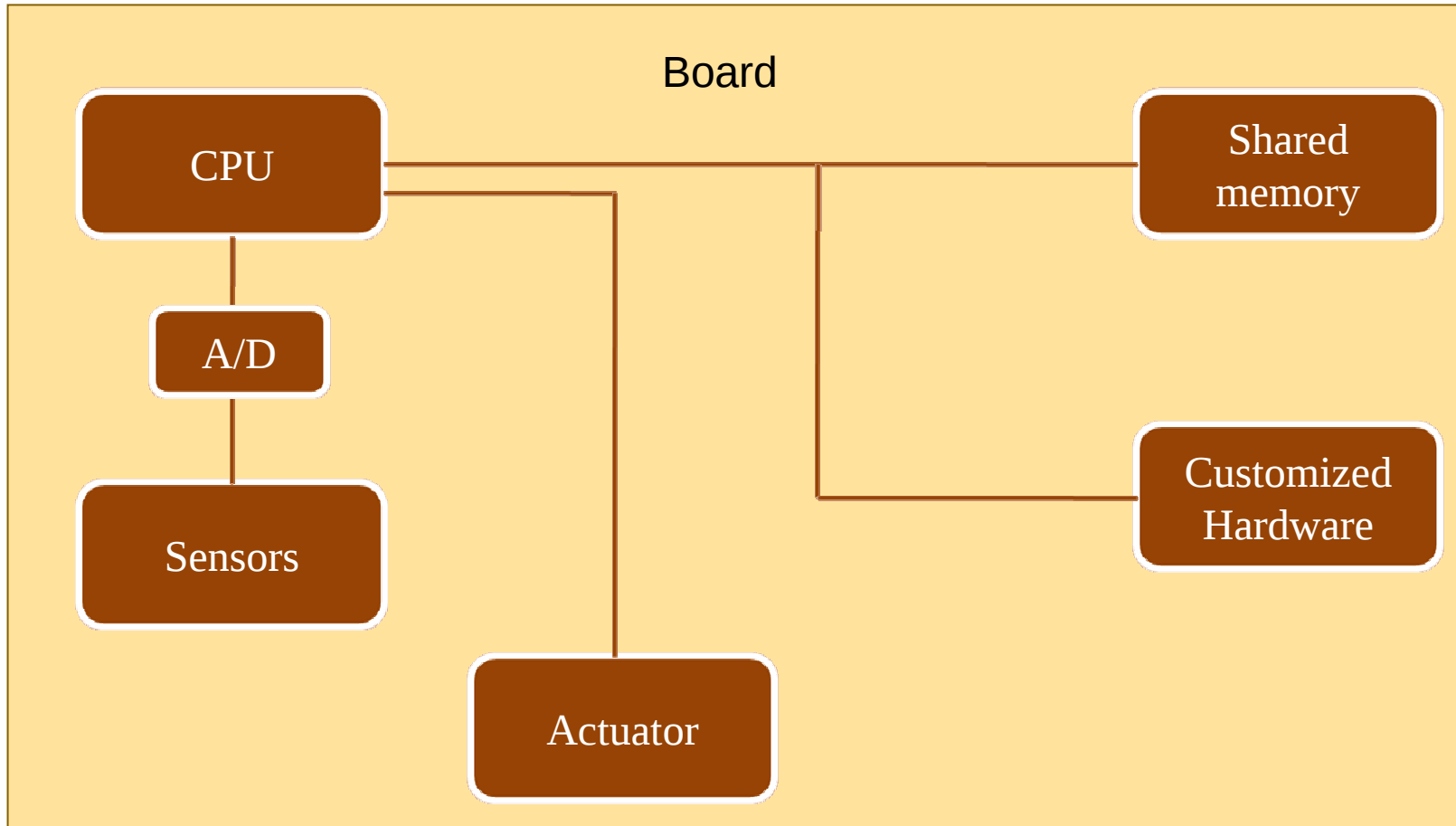


Hardware Characteristics

- Longer development cycle
- Customized for specific application
- Better performance in high speed real time application



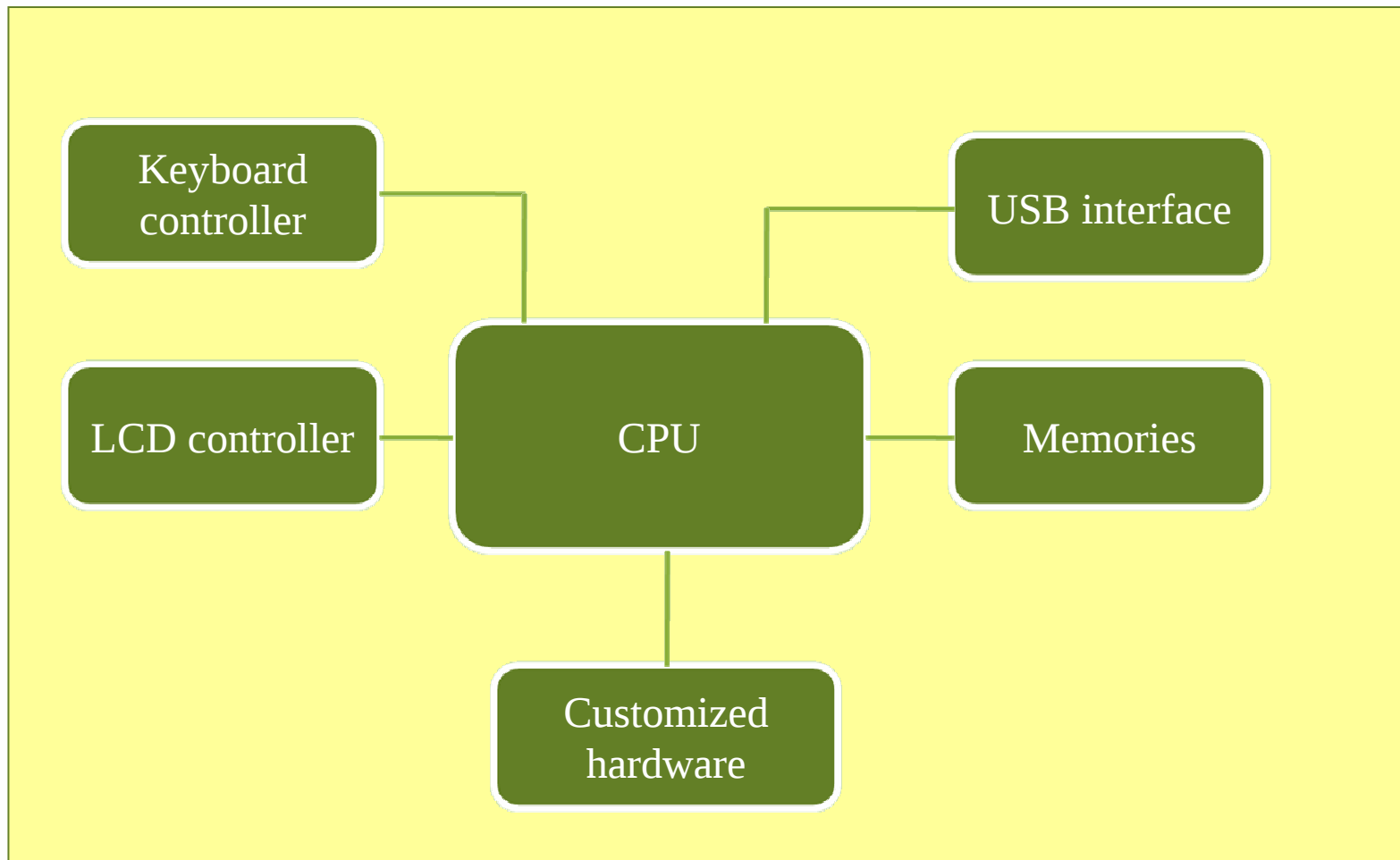
System on Board



System on Chip

- **System-on-a-chip (SOC):**
 - Integrating all components of a computer or other electronic system into a single integrated circuit (chip).
 - It may contain digital, analog, mixed-signal, and often radio-frequency functions – all on one chip.
 - typical application is in the area of embedded systems.
- An **application-specific integrated circuit (ASIC)** is an integrated circuit (IC) customized for a particular use. For example, a chip designed solely to run a cell phone is an ASIC.

System on Chip



HW/SW Partitioning

- In complicated systems functional blocks could be:
 - Level I: external discrete hardware component on board
 - Level II: hardware integrated with CPU on chip (SoC)
 - Level III: done by software running on CPU





Advanced Embedded Systems

- Multi-core system on chip
Like mobile handset which has one chip contains:
 - DSP processor
 - Embedded processors like ARM
 - Custom hardware for GSM
 - Custom peripherals for board interface (keyboard, touch screen, memory card interface)





Advanced Embedded systems

- Network of embedded microcontrollers on board :
 - Many microcontrollers on one / many board(s) communicated together through specific bus protocol like LIN bus, CAN bus in automotive application.





Emulators & Simulators

- An **emulator** duplicates the functions of one system using a different system, so that the second system behaves like (and appears to be) the first system.
- This focus on exact reproduction of external behavior is in contrast to some other forms of computer simulation, which can concern an abstract model of the system being simulated.
- **Simulation** is the imitation of some real thing, state of affairs, or process.
- The act of simulating something generally entails representing certain key characteristics or behaviors of a selected physical or abstract system.

Languages used in ES

- Assembly Language:
 - The term **Assembly Language** refers to the **lowest-level** human-readable method for programming a particular computer.
 - Assembly Languages are **platform specific**, and therefore there is a different Assembly Language necessary for programming every different type of computer.
 - Assembly Language generally has a one-to-one correspondence with the underlying machine language, which is not human readable. (**no need for compilation**)
 - Some compilers, such as GCC, will convert high-level languages (such as C) into platform-dependent assembly language before assembling into machine language.



Languages used in ES

- Assembly Disadvantages:
 - Platform Specific.
 - Operate by acting on specific instructions.
 - Can perform no other actions besides those specifically listed.
 - Instructions are highly atomic: each instruction performs a single, small instruction.
 - Too difficult to program large applications.



Languages used in ES

- C Language:
 - Assembly language can be directly converted into binary with no compilation, C is instead '**linked**' together, meaning the individual source files are all put into one file and converted to binary.
 - C also gives the programmer the ability to program in assembly right inside the C-code, giving programmers the option to optimize a very important or heavily used piece of code in Assembly.
 - Easy to write and portable.
 - Knowing C will enable you to understand and appreciate an entire family of programming languages built upon the traditions of C. Knowledge of C enables freedom.



Q & A



Thanks You



info@escommittee.com

education@escommittee.com