Structure of the course

- In the lectures, we go through the …
  - Embedded systems concepts,
  - Programming language (Assembly and C) concepts and
  - Sofware engineering concepts
- … that are needed in order to do the labs
- In the labs, students need to…
  - Understand the concepts
  - Analyse and Use the concepts
  - Reflect and Discuss the concepts
- …in order to fulfill the ILOs of the course.

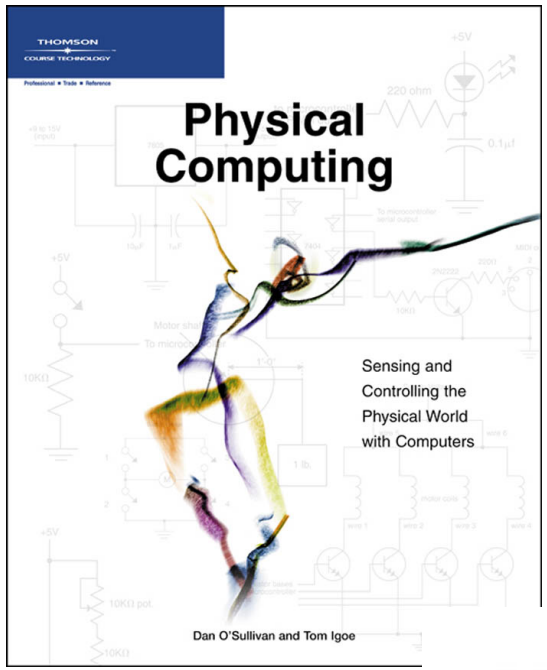
Kursstruktur DA215A, DA346A

- Föreläsningarna samläses (föreläsningsbilder är ibland något olika, för att spegla det som behövs till labbarna…)
- Labbarna samlokaliseras, och har samma syfte, men ser olika ut…
- DA215A DA346A

	X	Lab0	Bitmanipulation i Java + utvecklingsmiljön
X	X	Lab1 (ASM)	Keyboard + på labplattan (Arduino)
X	X	Lab2 (ASM)	Timing+LCD
X	X	Lab3 (ASM)	Spel ”Tärning” (data i minnet)
X	X	Lab4 (C)	Spel ”gissa talet” (strängar och pekare)
X	X	Lab5 (C)	Tillståndsmaskin, ADC
X	(X)	Lab6 (C)	PWM - motorstyrning

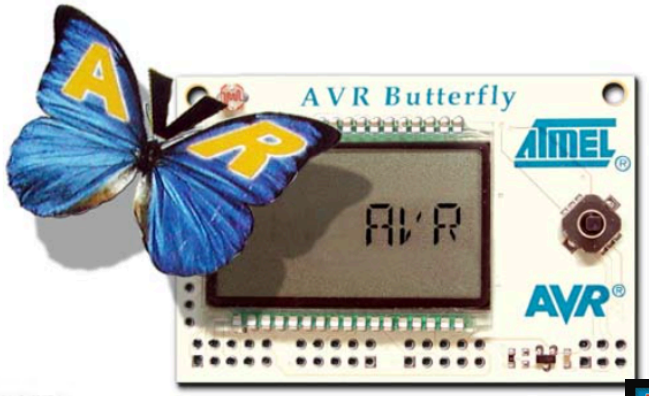
Course litterature…

- Lecture slides (and lectures!) contains almost all material that is needed in the course, but:
  - References to, e.g. Data Sheets are included. It may be necessary to read the referenced material.
  - Also other material may be referenced. In most cases this is for deeper knowledge, but may be necessary in order to understand certain concepts
- Books and other material may be useful. The next slide shows some examples, that have been reviewed when preparing the lecture slides.



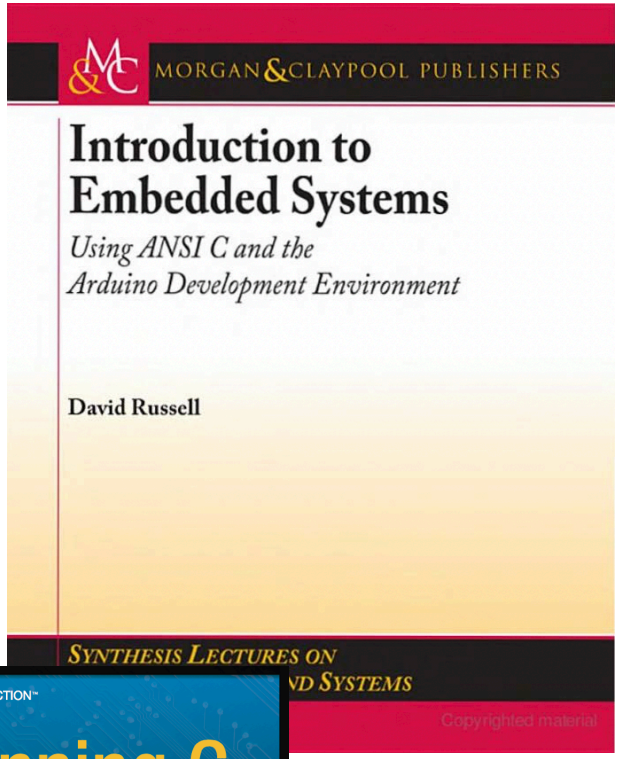
C Programming for Microcontrollers

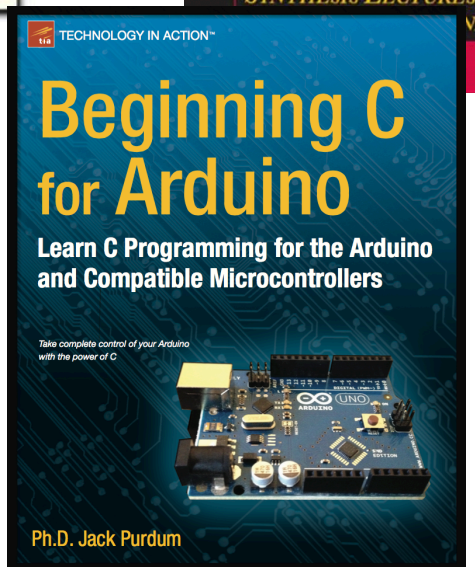
Featuring ATMEL's AVR Butterfly and the Free WinAVR Compiler



Joe Pardue

SmileyMicros.com





Embedded systems…

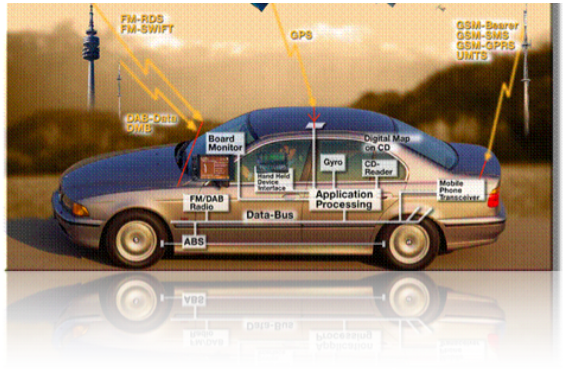
- This lecture provides a brief overview of what distinguishes embedded systems programming from "ordinary programming." It then touches upon facilities that become prominent or problems when working "close to the hardware" such as bit manipulation, and coding standards.
- Remember: not all computers are little grey boxes hiding under desks in offices.

Embedded systems programming

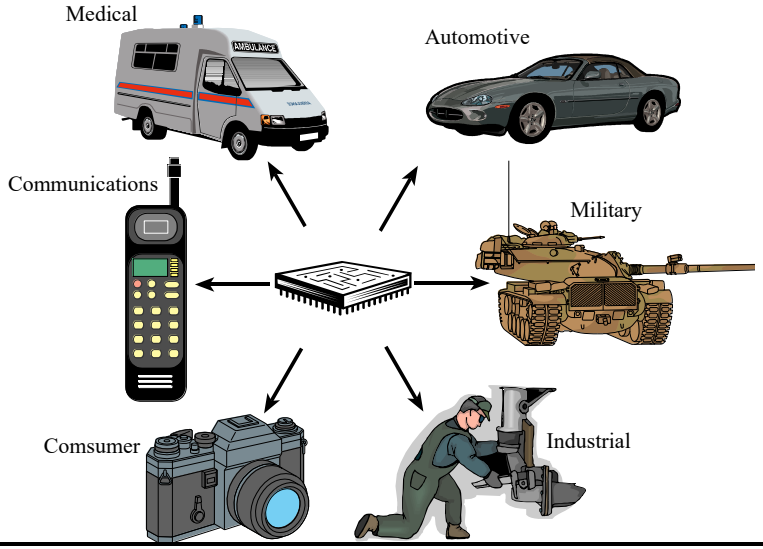
- You (usually) have to be much more aware of the resources consumed in embedded systems programming than you have to in “ordinary” programs
  - Time
  - Space
  - Communication channels
  - RAM (Data Memory)
  - Flash memory (Code)
  - …
- You must take the time to learn about the way your language features are implemented for a particular platform
  - Hardware
  - Operating system
  - Libraries

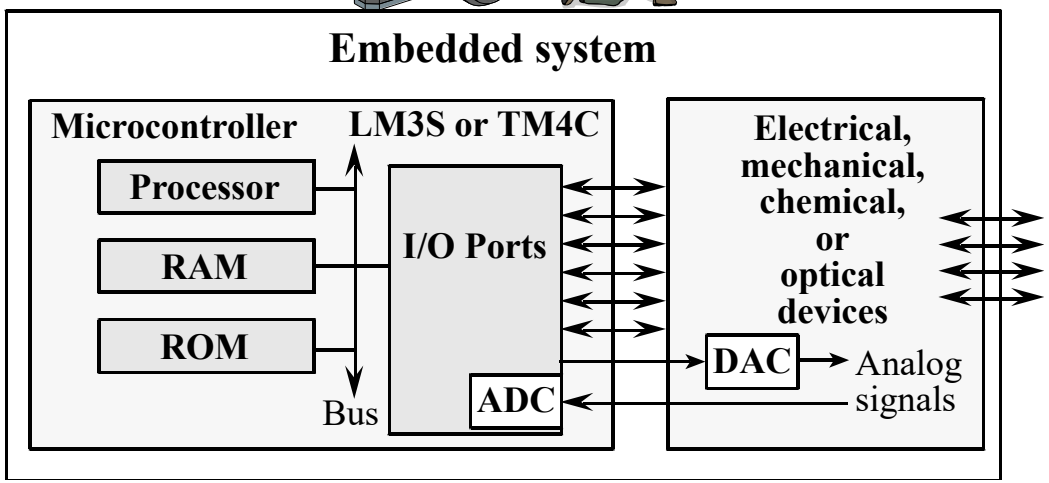
What is an Embedded System

- An embedded system contains a computer as part of a larger system and does not exist primarily to provide standard computing services to a user.



Embedded System





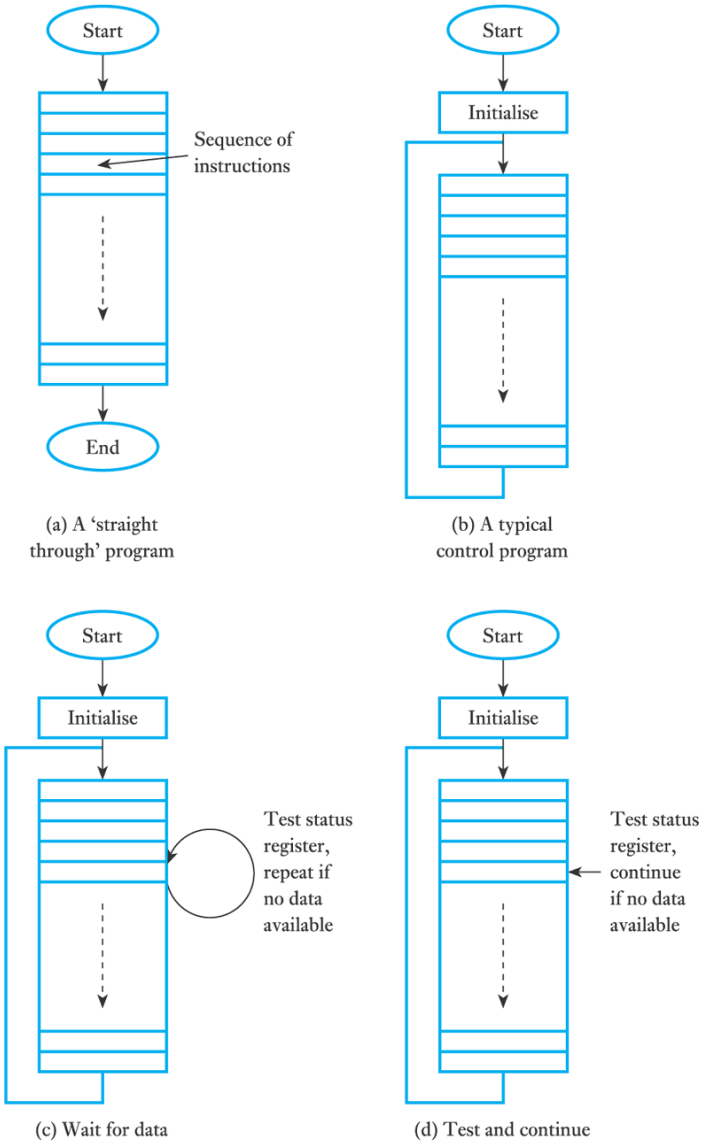
- Embedded Systems are everywhere
  - Ubiquitous, invisible
  - Hidden (computer inside)
  - Dedicated purpose
- MicroProcessor
  - Only processor, no memory or I/O…
- MicroController
  - Processor+Memory+ I/O Ports (Interfaces)

Some important concepts… we will cover all these at some point in the course…

- Polling/Interrupt
- Bit/Byte
- Port
- Finite State Machines
- Real-time
- Data communication
- Processor/Memory organisation (von Neuman/Harward)
- Sensors and Actuators
- LEDs, Switches and Buttons
- Registers (Port registers, direction registers)
- The Stack
- Dynamic and Static allocation
- ADC/DAC
- Status register
- IO-mapping (memory mapped/IO-mapped

Programmed controlled input/output

- polling of I/O devices
- This picture is also used in later lectures…



Concepts in embedded systems

- Single-functioned – often do one thing, no generic computer
- Tightly constrained – not much memory, keyboard, display, etc
- **Reactive and Real time** − Many embedded systems must continually **react to changes** in the system's environment and must compute certain results **in real time** without any delay.
- Wikipedia: **Real-time computing** (**RTC**), or **reactive computing** is the computer science term for hardware and software systems subject to a "real-time constraint", for example from event to system response.Real-time programs must guarantee response within specified time constraints, often referred to as "deadlines"
- Connected – sometimes include data communication.
- Sensors and Actuators
- ADC/DAC – Analog-to-Digital and Digital-to-Analog Conversion

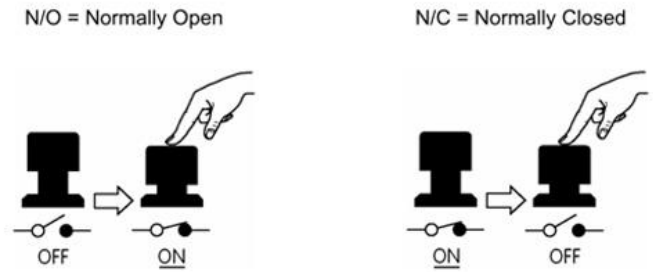
- We will come back to all these in later lectures…

Concepts in embedded systems

- PORT register — An interface between the computer and the outside world. Ports contain BITs.
- DATA DIRECTION register — A place where the direction (in or out) of each bit in the port is configured
- LEDs — A way to display the status of a single bit. The LED (Light Emitting Diode) can be on/off
- SWITCHes — A way to give an input. A switch can be on/off (stays in that state even if it is not touched)
- BUTTONs — A way to give an input. A button can be on or off.

Often On when the button is pressed:
”NO – Normally Open”

The opposite can also apply:
”NC – Normally Closed”

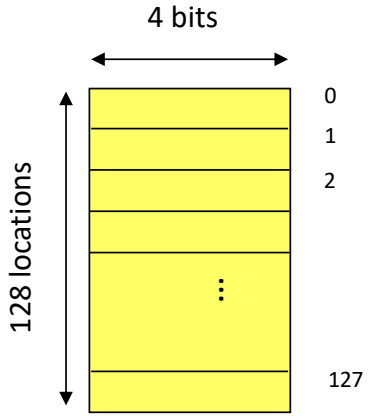


The Computer - Internal organization

- The different parts of a computer
  - I/O
  - Memory
  - CPU
- Connecting the different parts
  - Connecting memory to CPU
  - Connecting I/Os to CPU
- How computers work

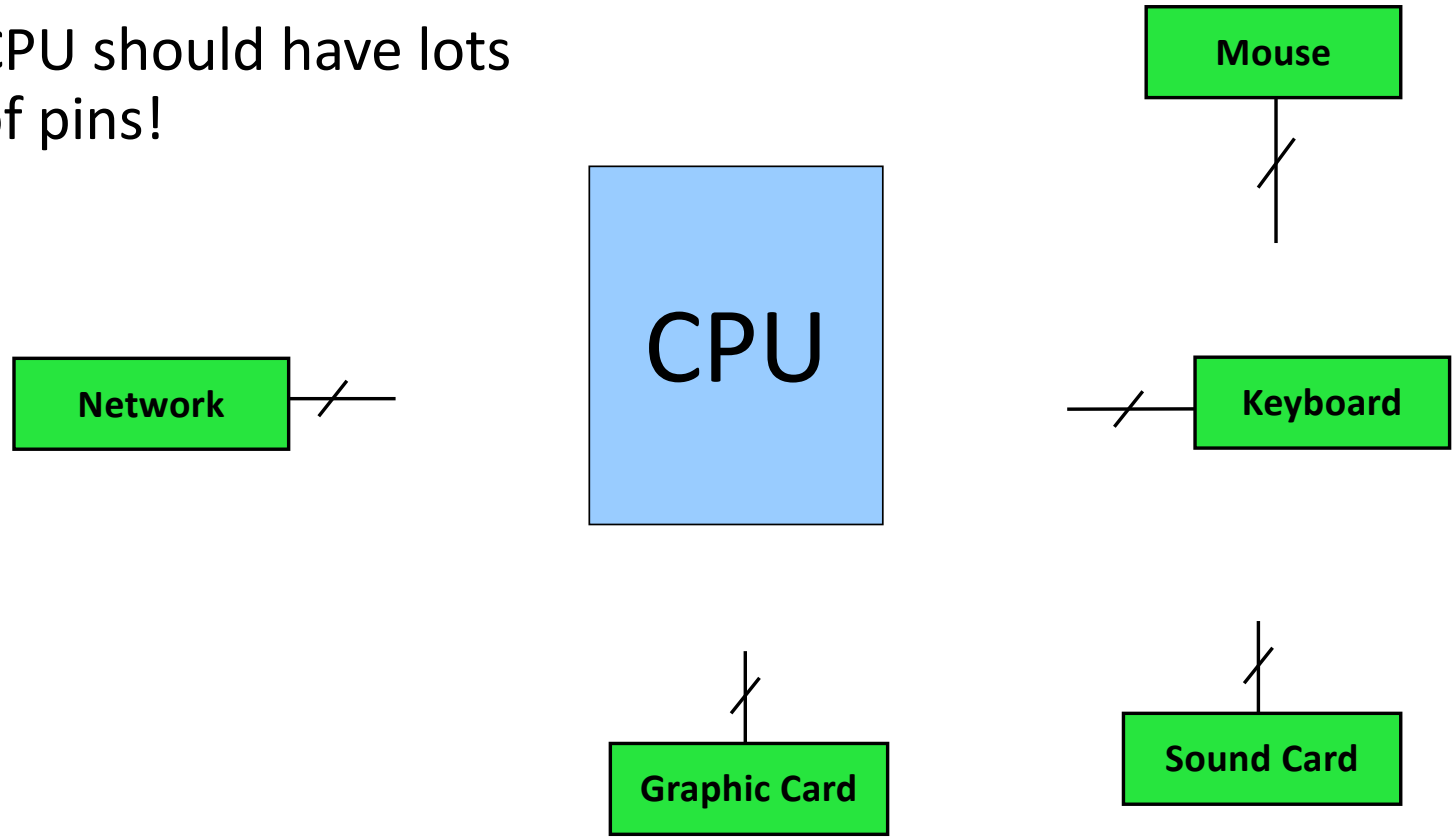
Memory characteristics

- Capacity
  - The number of bits that a memory can store.
    - E.g. 128 Kbits, 256 Mbits
- Organization
  - How the locations are organized
    - E.g. a 128 x 4 memory has 128 locations, 4 bits each
- Access time
  - How long it takes to get data from memory

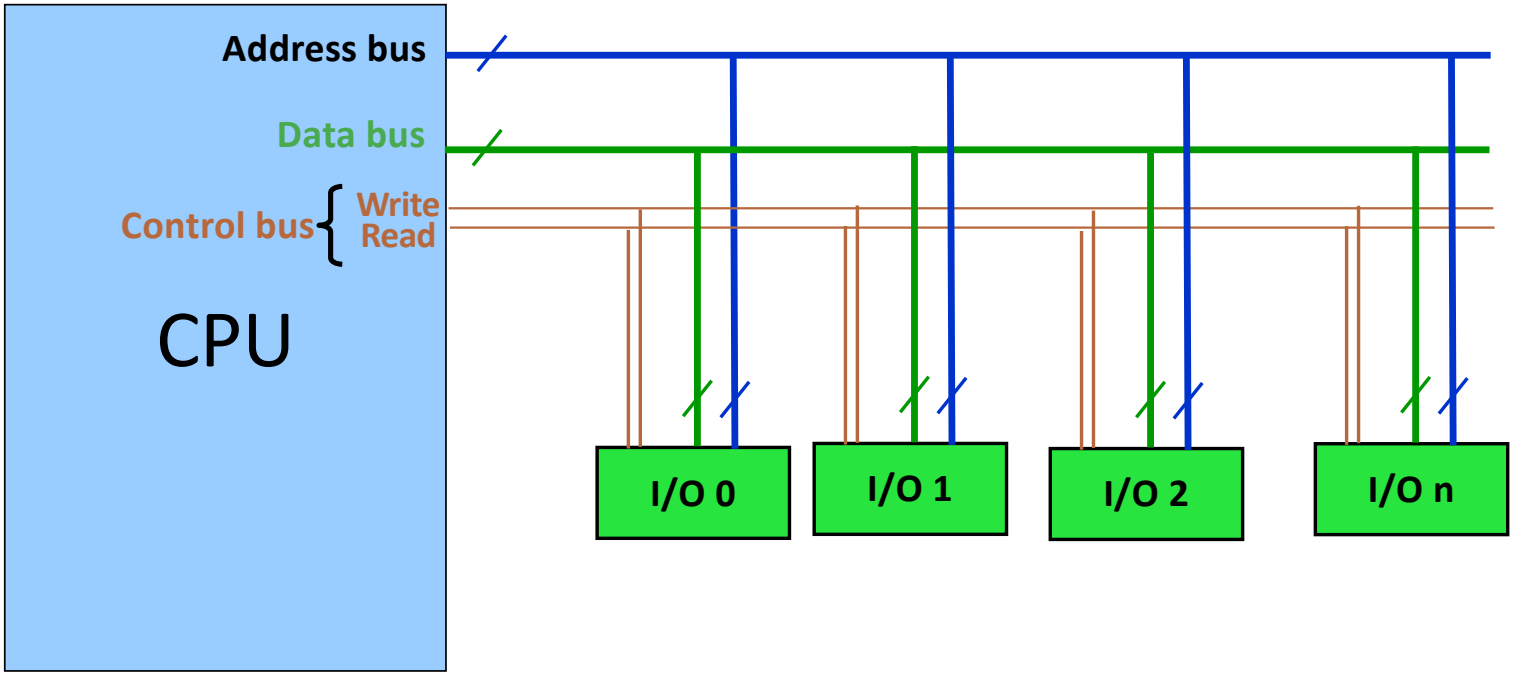


Connecting I/Os to CPU

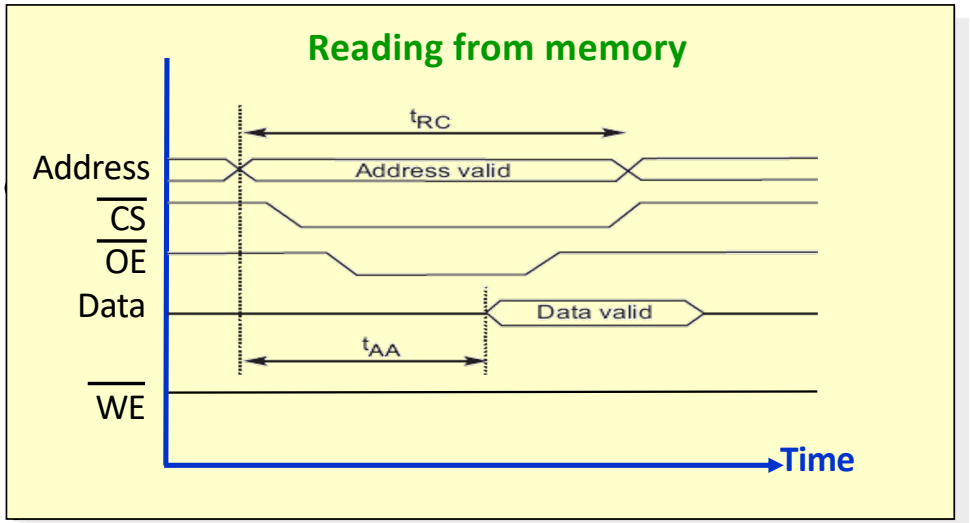
- CPU should have lots of pins!

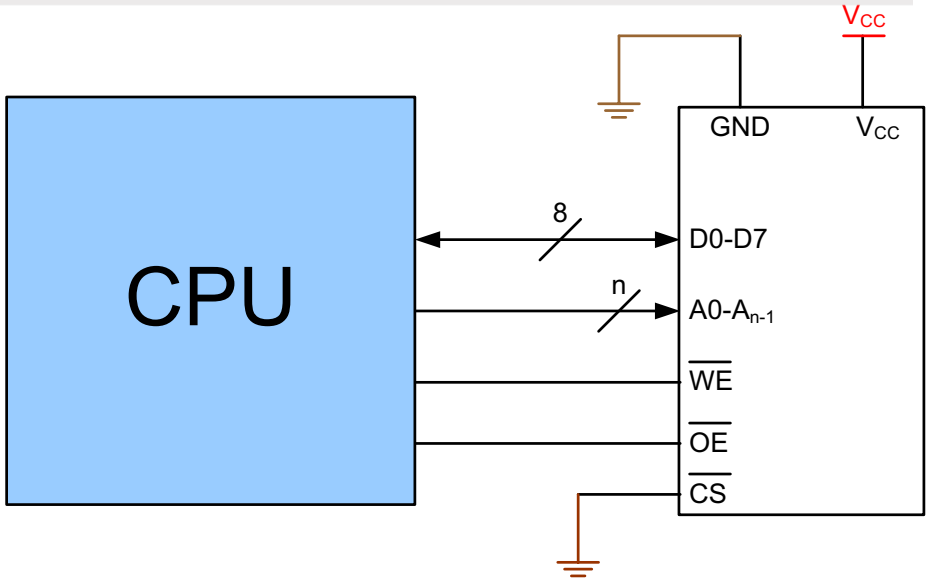


Connecting I/Os to CPU using the “bus” concept

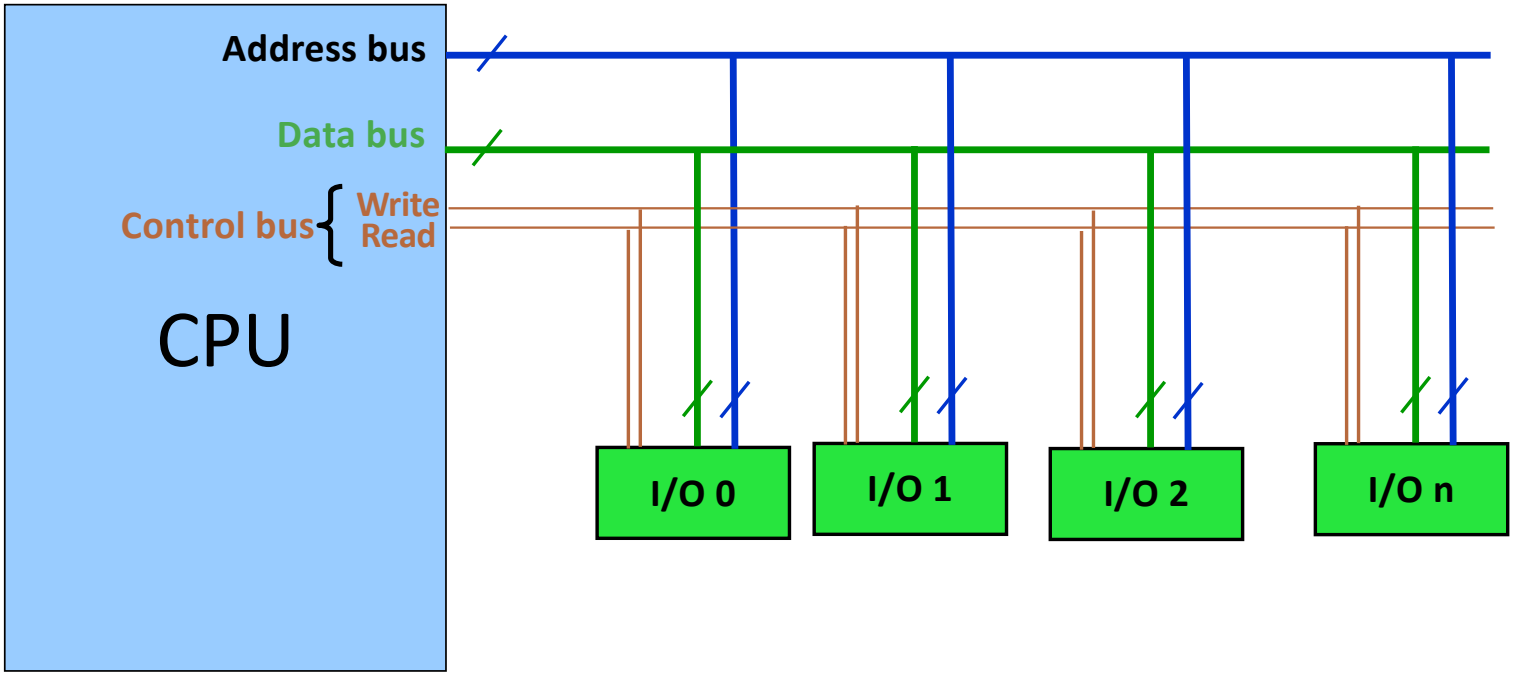


Connectin

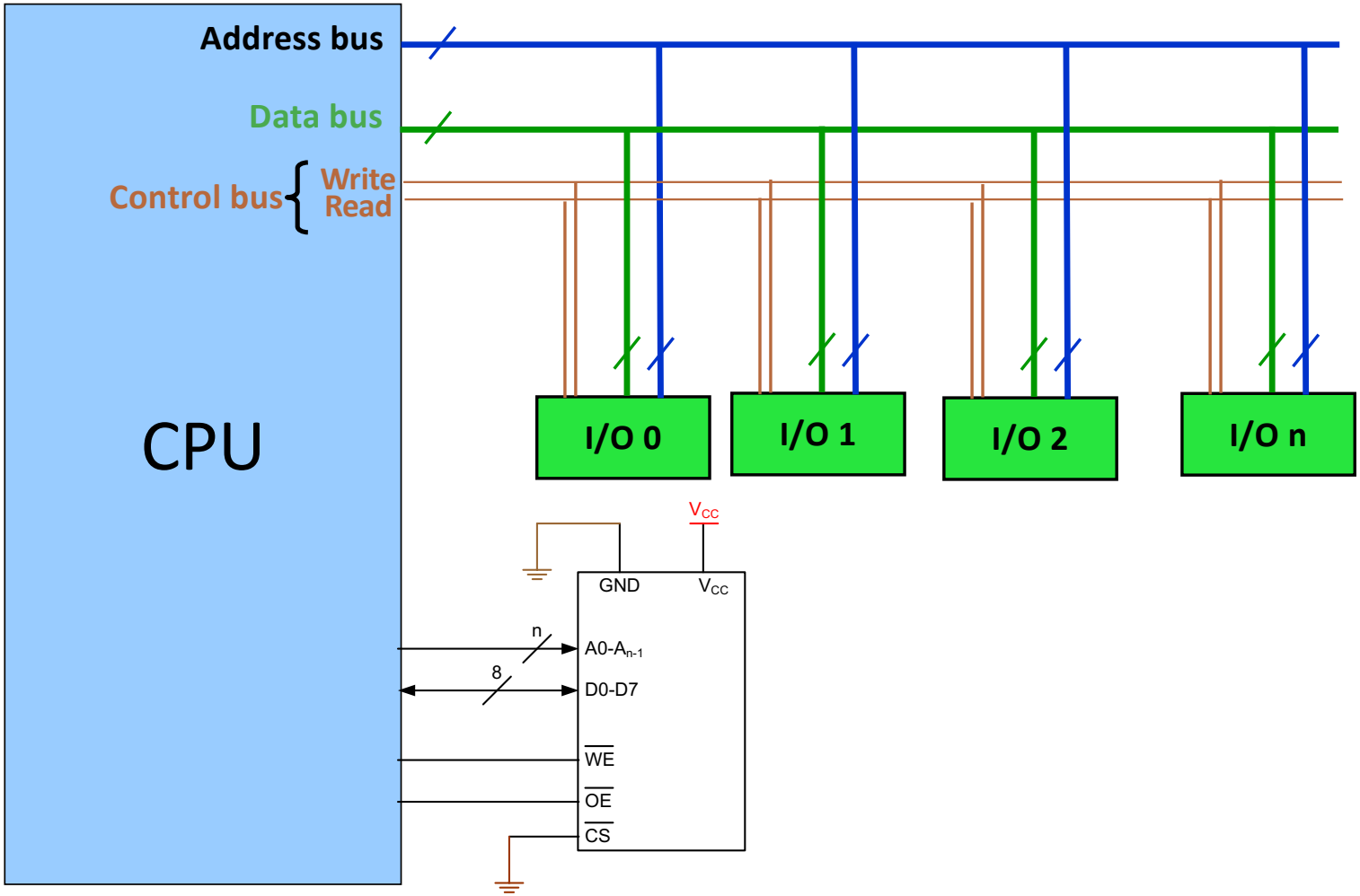

Reading from memory
t_{RC}
Address
Address valid
CS
OE
Data
Data valid
t_{AA}
WE
Time


V_{CC}
GND
V_{CC}
CPU
8
D0-D7
n
A0-A_{n-1}
WE
OE
CS

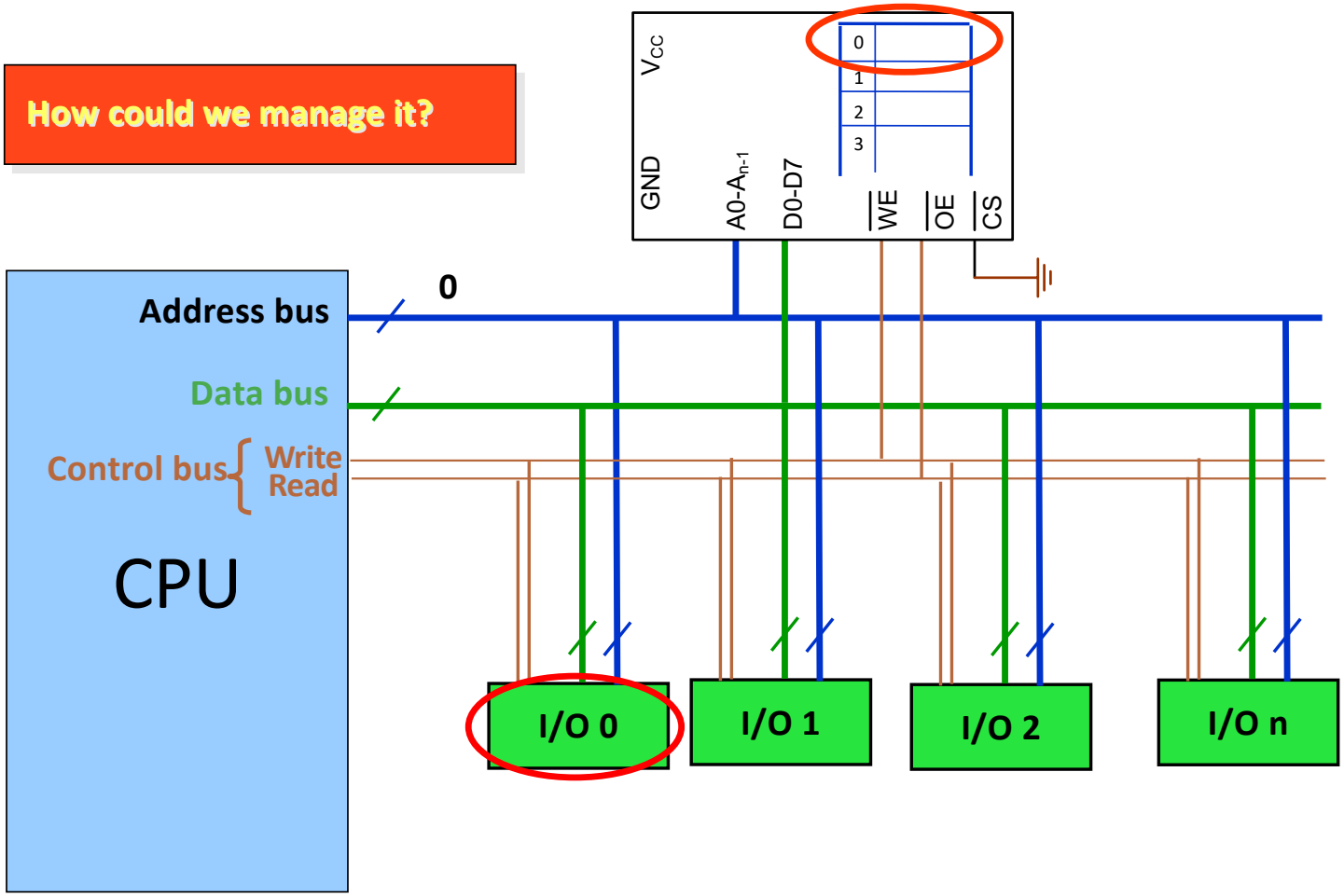
Connecting I/Os to CPU using bus



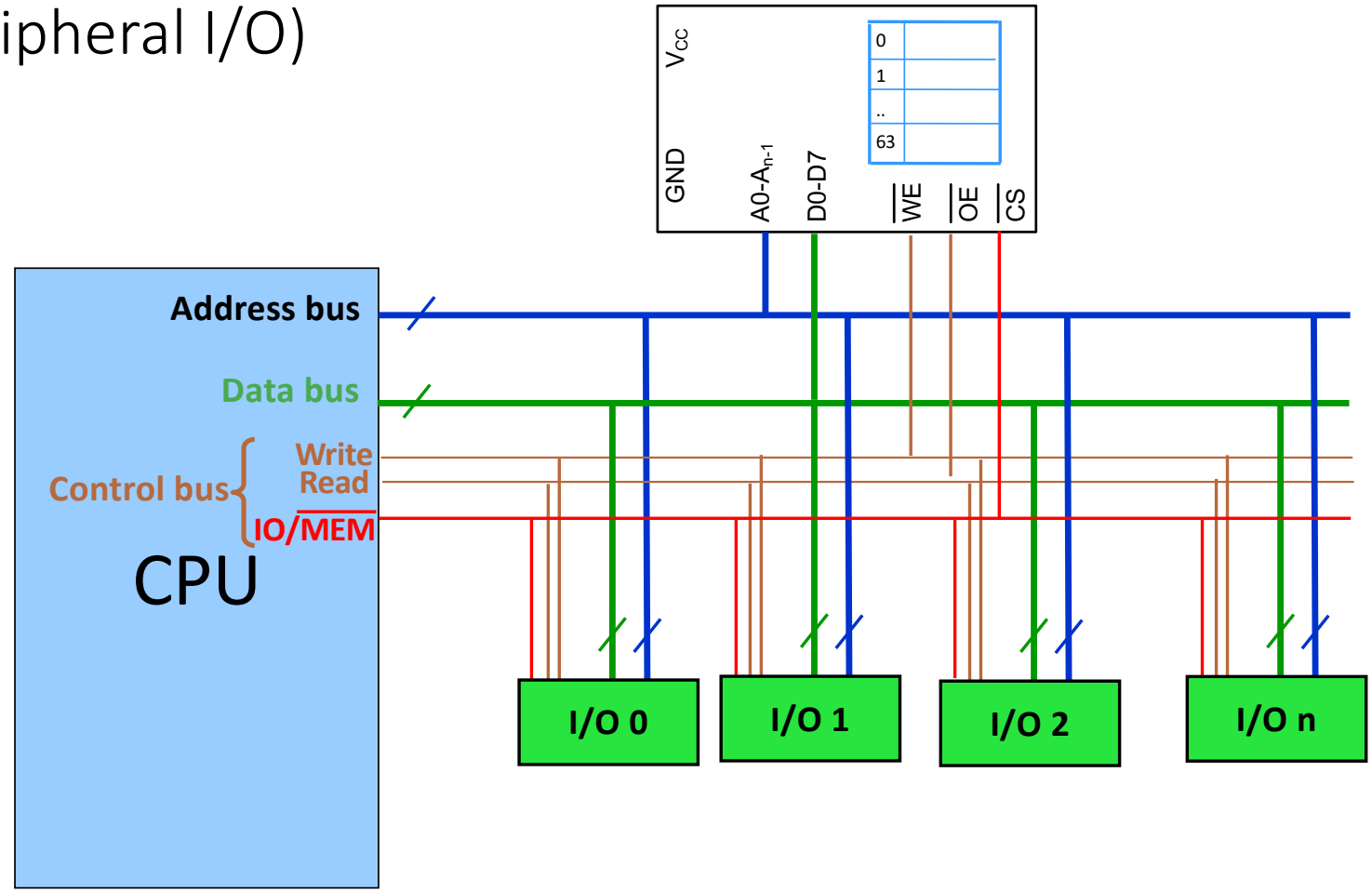
Connecting I/Os and Memory to CPU



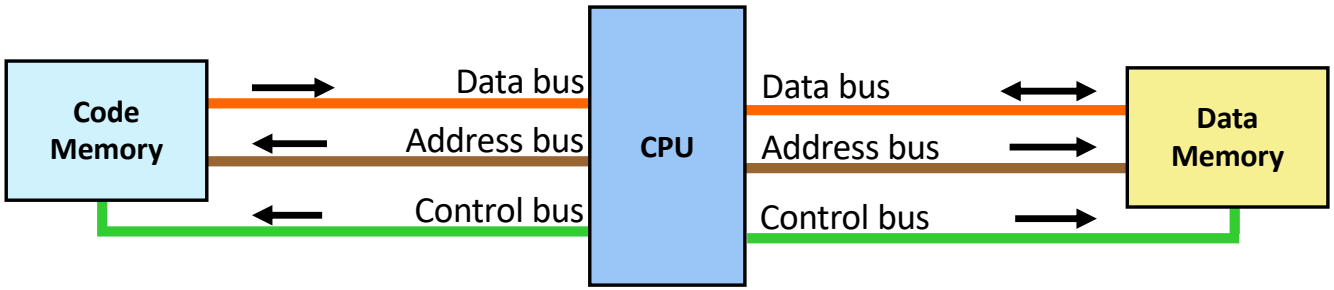
Connecting I/Os and memory to CPU using bus



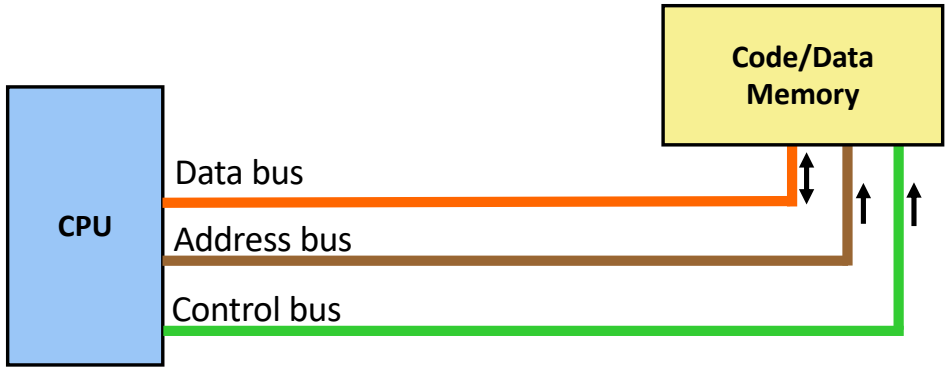
Connecting I/Os and Memory to CPU using bus (Peripheral I/O)



Von Neumann vs. Harvard architecture



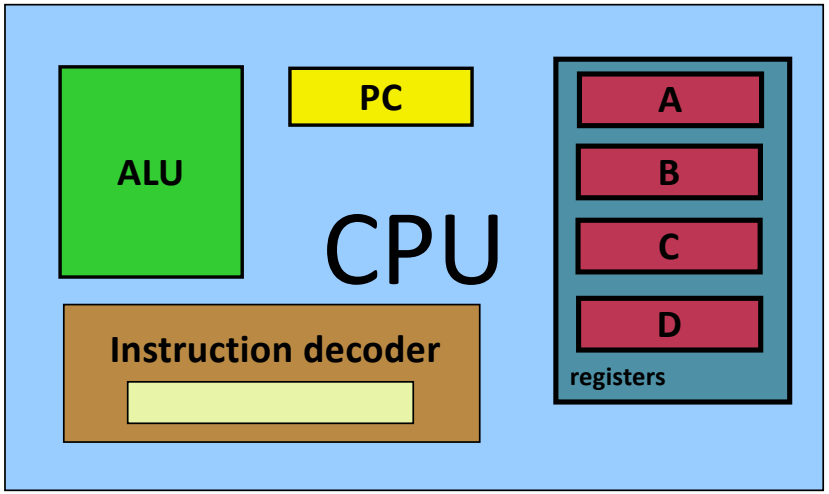
- Harvard architecture



- Von Neumann architecture

Inside the CPU

- PC (Program Counter)
- Instruction decoder
- ALU (Arithmetic Logic Unit)
- Registers



Animation of how computer executes short program…

- First part of: “IngSys_F1-5_animation”