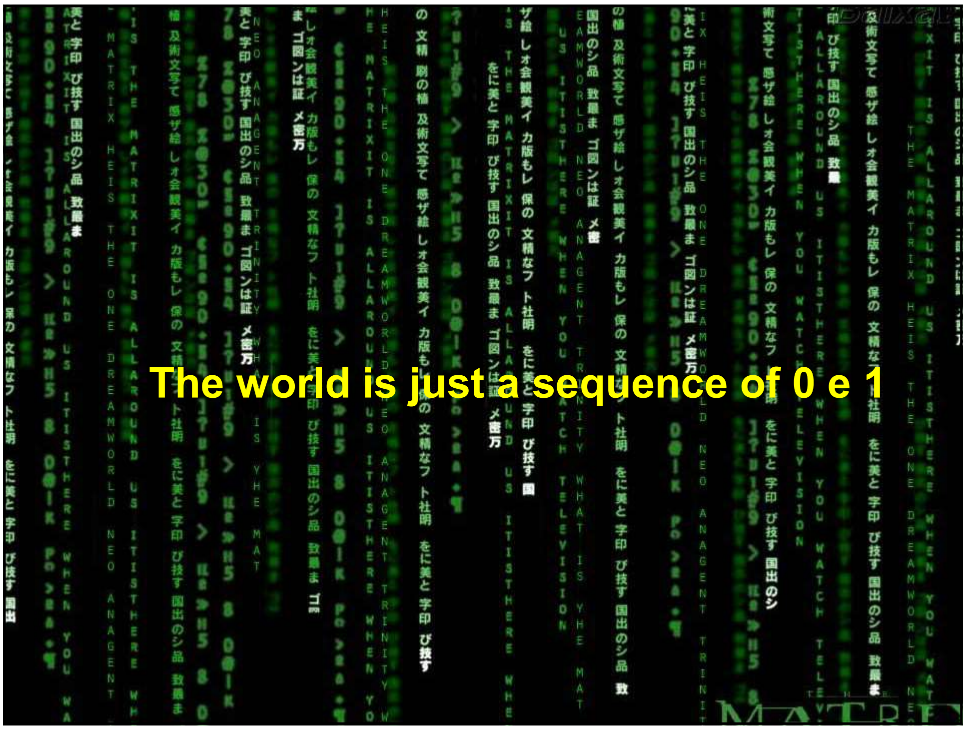
Hardware e Software das Tecnologias de Informação

V 1.3, V.Lobo, EN/ISEGI, 2017



Hardware & Software of IT

Prof. Victor Lobo

Licenciatura em Sistemas e Tecnologias de Informação

Objectives

- Understand the HARDWARE
  - What are computers made of ?
  - What is "computer architecture"?
  - What is a microprocessor ? (μp)
- Understand different types SOFTWARE
  - Machine Language
  - High level languages
  - Operating systems and "device drivers"

Syllabus (overview)

- **1. Introduction to *computing machines***
- **2. Data representation**
- **3. Boolean Algebra**
- **4. Digital Systems**
- **5. Memory systems**
- **6. Computer Architecture**
- **7. Microprocessors**
- **8. Peripherals**
- **9. Operating Systems, programming languages, and performance evaluation**

Why is it important ?

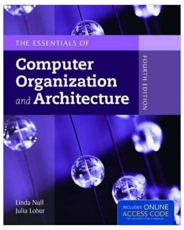
- Because **we want to understand** the world around us!
- Because only by understanding how computing machines work can we understand:
  - Their **limitations**
  - Their **capabilities**
  - **How to choose** them, to buy them, and to use them properly
- Because it is part of the STI *curriculum* …
  - You need to know this to get your degree…

Tough issues (or not…)

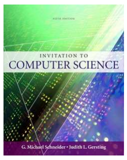
- **All information can be described in 0s and 1s**
  - The complete works of William Shakespeare; Michelangelo's Sistine Chapel; Handel's Messiah; your mother's face and voice…
- There is a **mathematical formulation** specifically for working with 0 a 1 (Boolean Algebra)
- You can build a **physical device** that can perform the *AND*, *OR*, and *NOT* logical operations
- **All information processing** can be made using **combinations of AND, OR, and NOT**

Bibliography

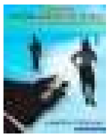
- Text book
  - **Computer Organization and Architecture**, Linda Null & Julia Lobur, Jones and Bartlett, 4th Edition, 2015

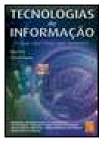


  - **Other books**
    - Introduction to Computer Science
      - An Invitation to Computer Science, 6th Ed, G.Michael Schneider, Judith Gersting, 2012



    - Brief introduction with (more or less) recent hardware
      - **Tecnologias de Informação**, Sérgio Sousa, FCA, 2009.
    - General introduction to Information Systems
      - **Introduction to Information Systems**, Rainer, Turban *et al.*, John Wiley & Sons, 2011





Bibliography (more detailed)

- **Digital Systems and Microprocessors**
  - Digital Fundamentals (10th Ed), Floyd, Prentice-Hall, 2010
  - Sistemas Digitais, Padilha, McGraw-Hill
- **Operating Systems**
  - Operating Systems (4th Ed), Tannenbaum, Prentice-Hall, 2014
  - Sistemas Operativos, Alves Marques *et al.*, FCA, 2009.

Assessment

- Final written exam
  - After the classes end (50 % of the grade) (up to 70%)
- Evaluation during the semester
  - Mini-tests (10+10%)(optional, mainly for feed-back)
    - Dates: 16th October, 4th December
    - If you fail in the mini-tests, the final exam will count more.
  - Assembly language program (20%)
    - Due date: 4th January.
  - Literature review and presentation (10%)
    - List of possible themes (suggestions are welcome)
    - **10 minute** presentation, and **2 page** report
  - Minimum grade in any item: 9/20

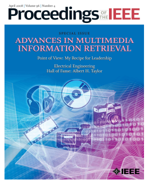
Objectives of the presentation

- **Search** for relevant information on Information Technologies
  - Learn about scientific on-line repositories
  - Get to know the most relevant publications
- Learn how to **assimilate** the relevant information
- Learn how to **present** it oraly, using visual aids
- Learn how **write** a short technical report

Themes for the presentations

- Choose a paper from an ACM or IEEE scientific journal
  - Go to the ACM and IEEE websites, and explore ! (using the B-ON access to full papers)
- Examples of “general purpose” journals:

Communications of the ACM



Proceedings df the IEEE

IEEE Spectrum

How to choose the theme

- Send na email to the teacher:
  - Use your novaims email
  - Start the SUBJECT with **HSTI**
  - Start the email with your NAME and STUDENT NUMBER
  - State the name of the paper/article, the complete reference of the Journal where you found it
  - The PDF of the paper in attachment
  - Send this information 1 WEEK before your presentation
  - You shall receive an email **accepting** your choice **or** suggesting you **choose another** theme
  - Paper/articles from other technical magazines/journals can be accepted if truly interesting
  - The exam will have questions regarding the presentations

Examples of themes for the presentation

Tecnologias de discos rígidos	Processadores ARM
Tecnologias de memórias flash	Processador Intel ATOM
Tecnologia de fabrico de circuitos impressos	Processadores Transmeta com code morphing
Tecnologia de fabrico de circuitos integrados	Microcontroladores PIC
Tecnologia de imressoras	Sistemas de visualização 3D
Tecnologia de impressoras (2D)	Sistema operativo Symbian
Tecnologia de impressoras (3D)	Sistema operativo Google Chrome
Tecnologia de Écrãs tácteis	Sistema operativo BeOS
Processador multi-core da PS3	Sistema operativo OpenVMS
Protocolo de comunicação Bluetooth	Sistema operativo OS/2
Protocolo de comunicação HDMI	Sistema operativo Minix
Protocolo de comunicação USB	Sistema operativo Anderoid
Protocolo de comunicação RS232	Sistema operativo iOS
Protocolo de comunicação CAN	Kits de microprocessadores Arduino
Protocolo de comunicação FireWire	Kits de microprocessadores TINI
Protocolo de comunicação SATA	Kits de microprocessadores Rabbit
Discos Blu-Ray	Kits com FPGA
Formatos de discos ópticos CDs/DVDs	Tablets
Super computadores	Computação Ubíqua
Vantagens e desvantagens de network-attached storage	Computação na "Cloud"
Computação quântica	Computação em automóveis
Computação optica	Computação para domótica
Quintas de servidores e gestão de energia	Vírus Informáticos
Blade PCs	Evolução dos sistemas de visualização: CRT a LCD e touch
Placas gráficas topo de gama	Interfaces homem/máquina com MS-Kinect
Processadores de topo de gama	Interfaces homem/máquina com feedback vibratório
Processadores gráficos	Interfaces homem/máquina com seguimento da retina
Processadores MIPS	Interfaces homem/máquina 3D

Dates for the presentation of your Literature review (dates were randomly seletected)

Date	Number	Name
26/set	2016656	Salvador Maria Silveira Paio
26/set	2015542	Ana Teresa de
26/set	2016665	Ricardo
27/set	2013478	
27/set	2015402	
3/out		
3/out		
3/out		
10/out		
31/out		
31/out		
31/out		
7/nov		
7/nov	2015579	
7/nov	2016653	
14/nov	2013524	João Carlos
14/nov	2016647	Pedro Nuno Ângelo
14/nov	2016640	Rita Nunes Pombo Marcelino

Date	Number	Name
15/nov	2016644	Sara Moiteiro Campos
		Miguel Farias Agostinho
		Silvestre
		Carmona
		Taylan
		Sofia Boto das Neves
18/dez	2014517	Ricardo Nuno Pascoal dos Santos

We will wait 1 week so that we have the complete list of enrolled students

Office hours and contacts

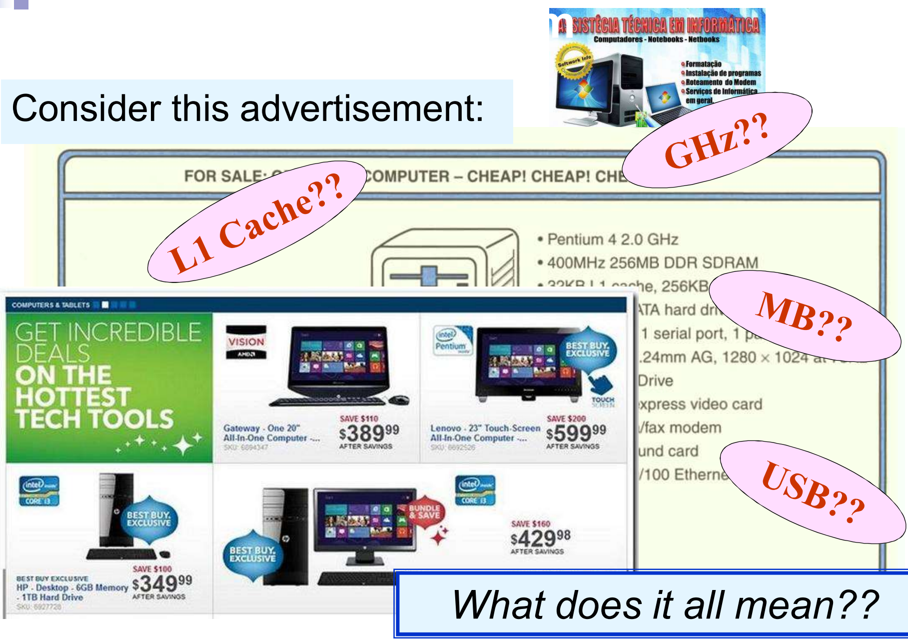
- Email: vlobo@novaims.unl.pt
  - Office hours
    - Tuesday at 18:30 and Wednesday at 20:00 (or whenever agreed)
    - By mail, anytime !
    - Whenever I am free at NovaIMS (!)
  - Supporting materials
    - www.novaims.unl.pt/docentes/vlobo, and Moodle platfrom
  - Changes in the class schedule
    - September 18th I will not be here (let's arrange another date now!)

General background of students

- What optional areas did you study in high school ?
- Did you pass in Computation I and II ?

General overview of computing machinery

Consider this advertisement:

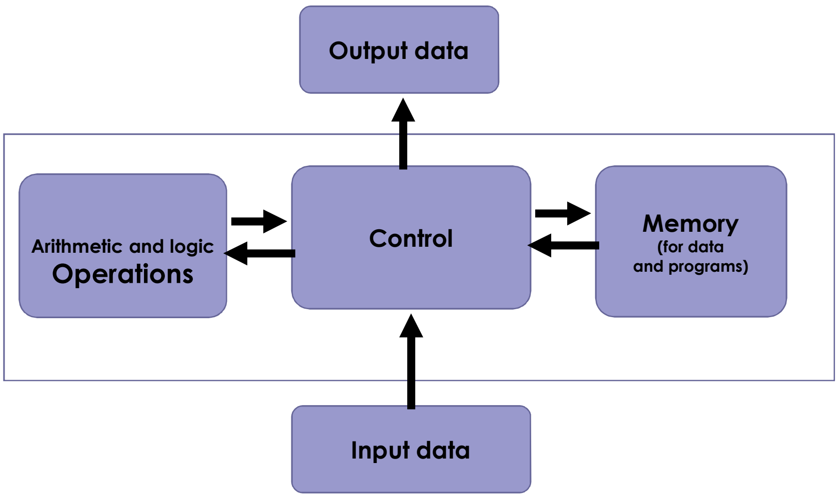


What does it all mean??

Basic components of a computer

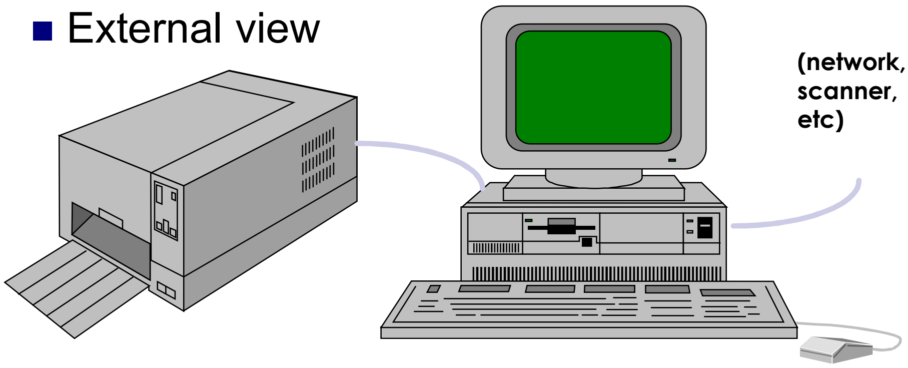
- **Processor** Unit
  - Process data, make computations
- **Memory** Unit
  - Store data, store instructions
- **Input/Output (I/O)** Units
  - Communicate with the outside world

- Von Neumann Architecture



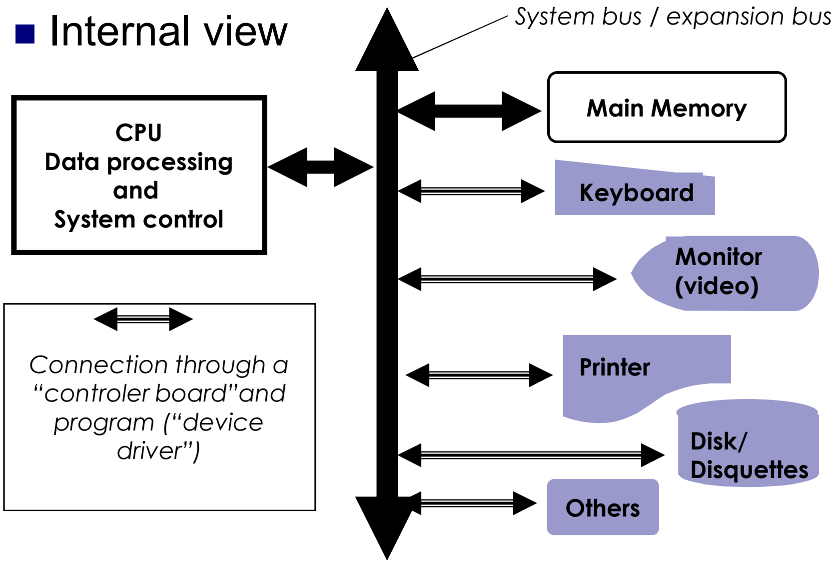
System components

- External view



System components

- Internal view

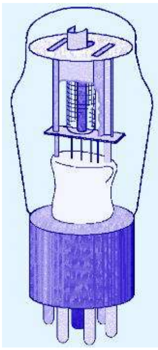


Computing history

History of *computing machines*

- Machines that process information
  - Make computations, store data, automate processes
- Before computers
  - Abacus
  - Pascal Machines and Leibniz Machines
    - Sums and subtractions with gear wheels
    - Artillery computers
  - Babbage Machines
    - Logarithm tables for navigation and "modern mechanical computer"
  - Hollerith Machines
    - Card readers and rudimentary information processing
  - Dedicated analog machines

First Generation computers: Vacuum Tube Computers (1945 - 1953)

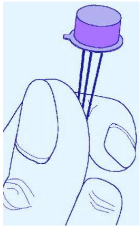


- Pre-war pioneers
  - Theorectical work
    - “On computable numbers” by Alan Turing
    - Theory for “Switching circuits”
  - Atanasoff-Berry Computer (1937 - 1938) solved systems of linear equations.

First Generation computers: Vacuum Tube Computers (1945 - 1953)

- 2nd World War efforts
  - Electronic Numerical Integrator and Computer (ENIAC)
    - John Mauchly and J. Presper Eckert
    - University of Pennsylvania, 1946
  - The ENIAC was the first *general-purpose* computer
  - Colossus
    - Bletchley Park, UK
- Post war efforts
  - First commercial applications by UNIVAC and IBM

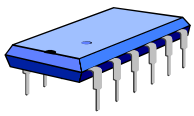
Second Generation: Transistorized Computers (1954 - 1965)



- IBM 7094 (scientific) and 1401 (business)
- Digital Equipment Corporation (DEC) PDP-1
- Univac 1100
- Control Data Corporation 1604.
- . . . and many others.

These systems had few architectural similarities.

Third Generation: Integrated Circuit Computers (1965-1980)



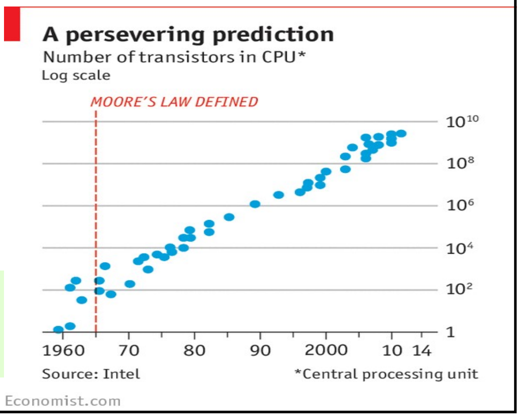
- Widespread industrial use
  - IBM 360
  - DEC PDP-8 and PDP-11
  - Cray-1 supercomputer
  - . . . and many others.
- By this time, IBM had gained overwhelming dominance in the industry.
  - Computer manufacturers of this era were characterized as IBM and the BUNCH (Burroughs, Unisys, NCR, Control Data, and Honeywell).

- The Fourth Generation: VLSI Computers (1980 - ????)
  - Very large scale integrated circuits (VLSI) have more than 10,000 components per chip.
  - Enabled the creation of microprocessors.
  - The first was the 4-bit Intel 4004.
  - Later versions, such as the 8080, 8086, and 8088 spawned the idea of "personal computing."
- Next ?
  - Massively parallel computers, Quantum computing ?

- Moore's Law (1965)
  - Gordon Moore, Intel founder
  - "The density of transistors in an integrated circuit will double every year."
- Contemporary version:

  "The density of silicon chips doubles every 18 months."

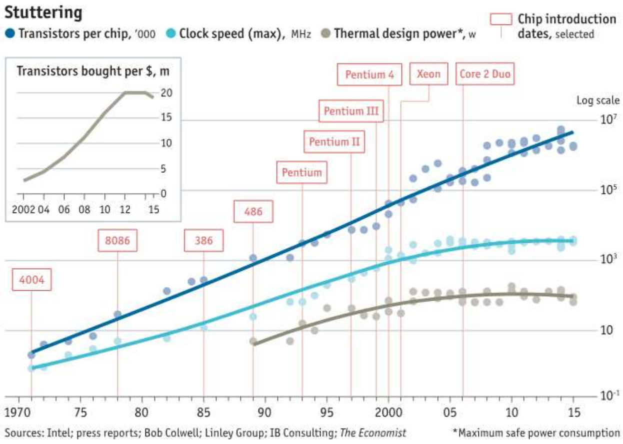
But this "law" cannot hold forever ...



- Rock's Law
  - Arthur Rock, Intel financier
  - "The cost of capital equipment to build semiconductors will double every four years."
  - In 1968, a new chip plant cost about $12,000.
  - In 2005, a chip plant under construction cost over $2.5 billion.
  - For Moore's Law to hold, Rock's Law must fall, or vice versa. But no one can say which will give out first.

But it's breaking down !

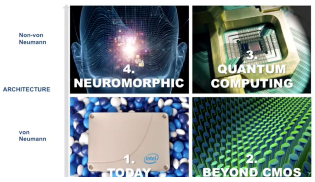
- Clock frequency
  - Wavelenght and synchronization
  - Heat emission
  - Radio emission
- Physical size
  - Going to 10 atoms across
- Multi-core has filled the gap…



Moor’s Law Breakdown could be good for science !

What is being done

- IEEE International Conference on **Rebooting Computing**, San Diego, 2016

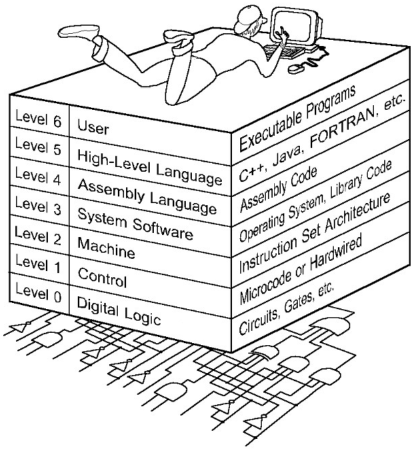


- Technologies “in the aisle”
  - Quantum Computing
  - Neural Computing
    - Neural Hardware, Deep learning, Spiking Networks
  - Approximate Stochastic Computing
    - Bayesian neural computing
  - Analog, dedicated hardware and asynchronous computing
  - Superconducting Computers
  - Advances in massively parallel computing
  - Advances in software engineering with alternate architectures

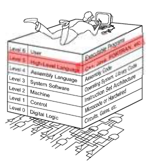
Layered struture of organization in computers

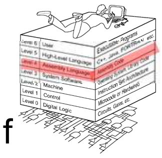
- Computers consist of many things **besides chips**.
- Before a computer can do anything worthwhile, it must also use **software**.
- Writing complex programs requires a "**divide and conquer**" approach, where each program module solves a smaller problem.
- Complex computer systems employ a similar technique through a series of **virtual machine layers**.

- **Each virtual machine layer is an abstraction of the level below it.**
- **The machines at each level execute their own particular instructions, calling upon machines at lower levels to perform tasks as required.**
- **Computer circuits ultimately carry out the work.**



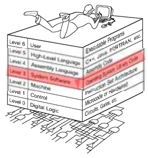
- Level 6: The User Level
  - Program execution and user interface level.
  - The level with which **we are most familiar**.
- Level 5: High-Level Language Level
  - The level with which we interact when we write programs in languages such as **C, Pascal, Lisp, and Java**.

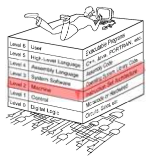




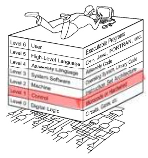
- Level 4: Assembly Language Level
  - Acts upon **assembly** language produced f Level 5, as well as instructions programmed directly at this level.

- Level 3: System Software Level
  - Controls executing **processes on the system**.
  - Protects system resources.
  - Assembly language instructions often pass through Level 3 without modification.

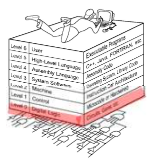




- Level 2: Machine Level
  - Also known as the Instruction Set Architecture (ISA) Level.
  - Consists of instructions that are **particular to the architecture** of the machine.
  - Programs written in machine language need no compilers, interpreters, or assemblers.



- Level 1: Control Level
  - A *control unit* decodes and executes instructions and moves data through the system.
  - Control units can be ***microprogrammed* or *hardwired***.
  - A microprogram is a program written in a low-level language that is implemented by the hardware.
  - Hardwired control units consist of hardware that directly executes machine instructions.



- Level 0: Digital Logic Level
  - This level is where we find **digital circuits** (the chips).
  - Digital circuits consist of gates and wires.
  - These components implement the mathematical logic of all other levels.

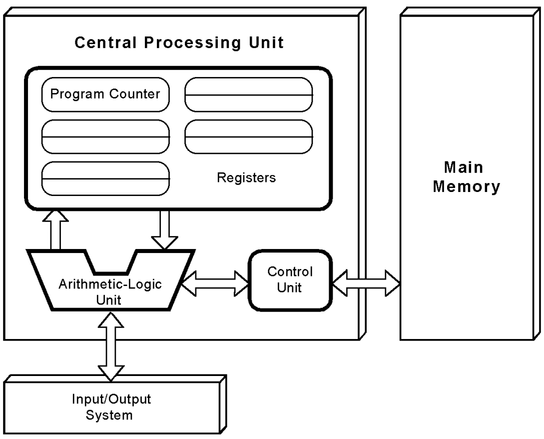
- On the ENIAC, all programming was done at the digital logic level.
- Programming the computer involved moving plugs and wires.
- A different hardware configuration was needed to solve every unique problem type.

Configuring the ENIAC to solve a "simple" problem required many days labor by skilled technicians.

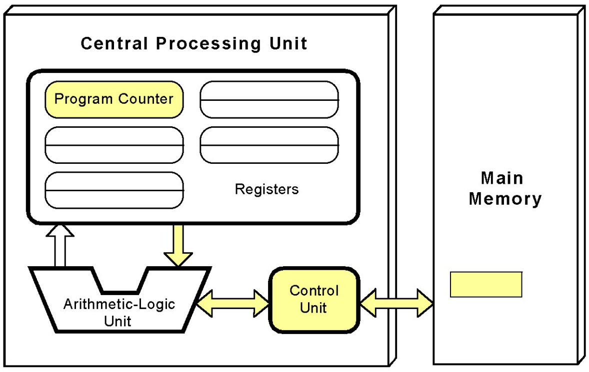
- Inventors of the ENIAC, John Mauchley and J. Presper Eckert, conceived of a computer that could **store instructions in memory**.
- The invention of this idea has since been ascribed to a mathematician, John von Neumann, who was a contemporary of Mauchley and Eckert.
- Stored-program computers have become known as **von Neumann Architecture systems**.

- Today's stored-program computers have the following characteristics:
  - **Three hardware systems**:
    - A central processing unit (CPU)
    - A main memory system
    - An I/O system
  - The capacity to carry out sequential instruction processing.
  - A single data path between the CPU and main memory.
    - This single path is known as the *von Neumann bottleneck*.

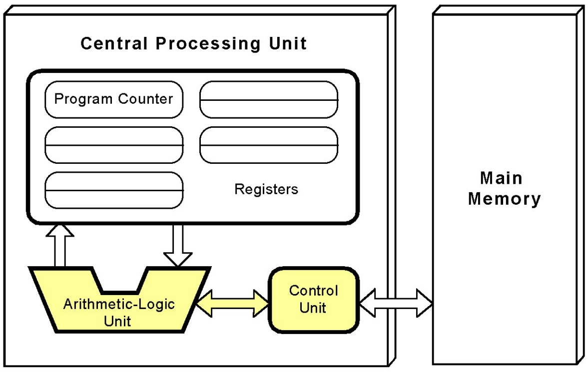
- **This is a general depiction of a von Neumann system:**
- **These computers employ a fetch-decode-execute cycle to run programs as follows . . .**



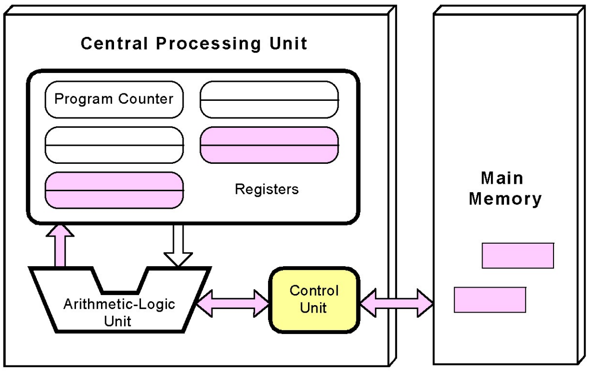
- The control unit **fetches** the next instruction from memory using the program counter to determine where the instruction is located.



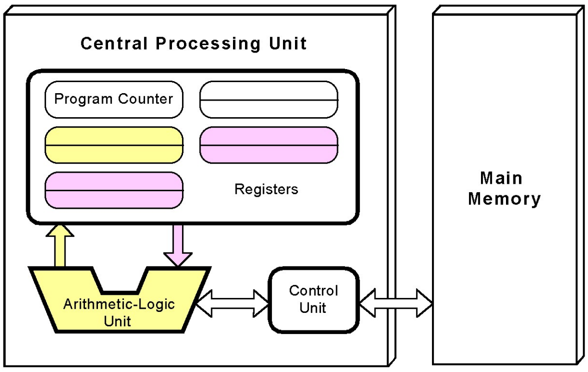
- The instruction is **decoded** into a language that the ALU can understand.



- Any data operands required to execute the instruction are fetched from memory and placed into registers within the CPU.



- The ALU executes the instruction and places results in registers or memory.



- Conventional stored-program computers have undergone many incremental improvements over the years.
- These improvements include adding **specialized buses**, **floating-point units**, and **cache memories**, to name only a few.
- But enormous improvements in computational power require departure from the classic von Neumann architecture.
- Adding processors is one approach.

- In the late 1960s, high-performance computer systems were equipped with dual processors to increase computational throughput.
- In the 1970s supercomputer systems were introduced with 32 processors.
- Supercomputers with 1,000 processors were built in the 1980s.
- In 1999, IBM announced its Blue Gene system containing over 1 million processors.

- **Parallel processing** is only one method of providing increased computational power.
- More radical systems have reinvented the fundamental concepts of computation.
  - These advanced systems include **neural hardware computers**, **genetic computers**, **quantum computers**, dataflow systems,
  - At this point, it is unclear whether any of these systems will provide the basis for the next generation of computers.

- This chapter has given you an overview of the subject of computer architecture.
- You should now be sufficiently familiar with general system structure to guide your studies throughout the remainder of this course.
- Subsequent chapters will explore many of these topics in great detail.