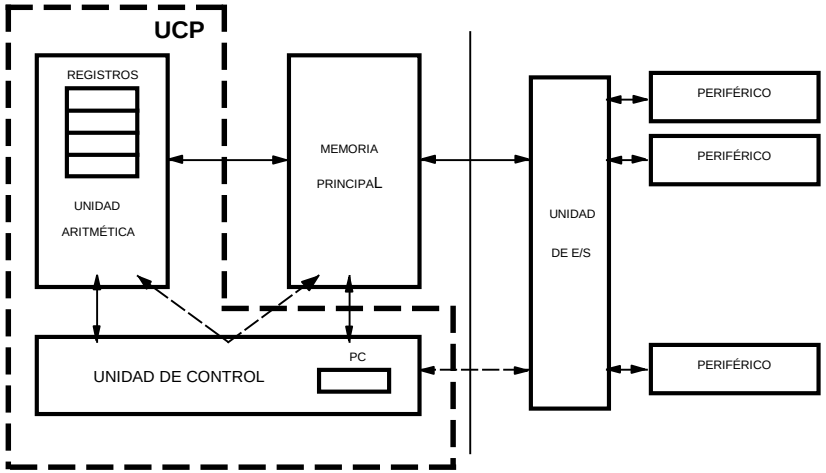
Practice 0

- MSX88 SIMULATOR

Von Neumann Architecture

- Developed 1945 by John von Neumann
- Executes machine instructions from a program stored in the memory

- Blocks:
  - Memory
  - Arithmetic-Logic Unit and registers
  - Control Unit
  - Input/Output unit



- **Buses** connect different parts:

data bus, address bus, control bus

Área de Arquitectura y Tecnología de Computadores
Departamento de Automática
Universidad de Alcalá

Instruction execution cycles

1. **Fetch**
   CU generates signals to read a instruction from memory in the direction pointed by the program counter PC
2. **Decode:**
   CU receives instruction at IR and decodes it
3. **Fetch operands:**
   CU reads, if neccessary, operands from memory or registers
4. **Executions and store results**
   CU generates signals to execute instruction, and stores result in memory or registers
5. **Program Counter update.** CU updates the program counter to point to the next instruction to be executed.
   - sequential working
   - sequence modification ↓ PC modification ↓ bifurcation or jump

Área de Arquitectura y Tecnología de Computadores
Departamento de Automática
Universidad de Alcalá

Programming languages (I)

High level language:
-Set of instructions and syntaxis (PASCAL, C)
-portable (same code compiles in different machines)

Low level language

- **Machine language**:
  Instructions are written in binary
  - Difficult and a lot of mistakes
    ⬇
    Solution: high level language and compile
- **Assembly language**:
  Instructions are represented with symbolic names or mnemonics
  - Each instruction correspond with a machine instruction

Área de Arquitectura y Tecnología de Computadores
Departamento de Automática
Universidad de Alcalá

Programming languages (II)

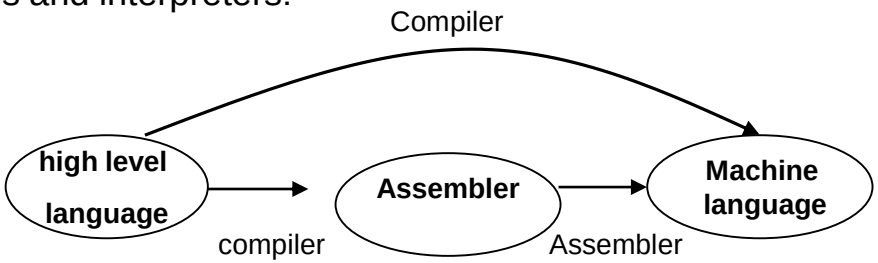
- High level language (example: PASCAL)

```
BEGIN
   Resta:= Minuendo - Sustraendo
END.
```

- Machine language and assembler (Example: i80x86)

```
A10000          MOV AX, Minuendo
2B060200        SUB AX, Sustrayendo
A30400          MOV Resta, AX
```

- Traduction from high level program to machine language are carried on by compilers and interpreters:

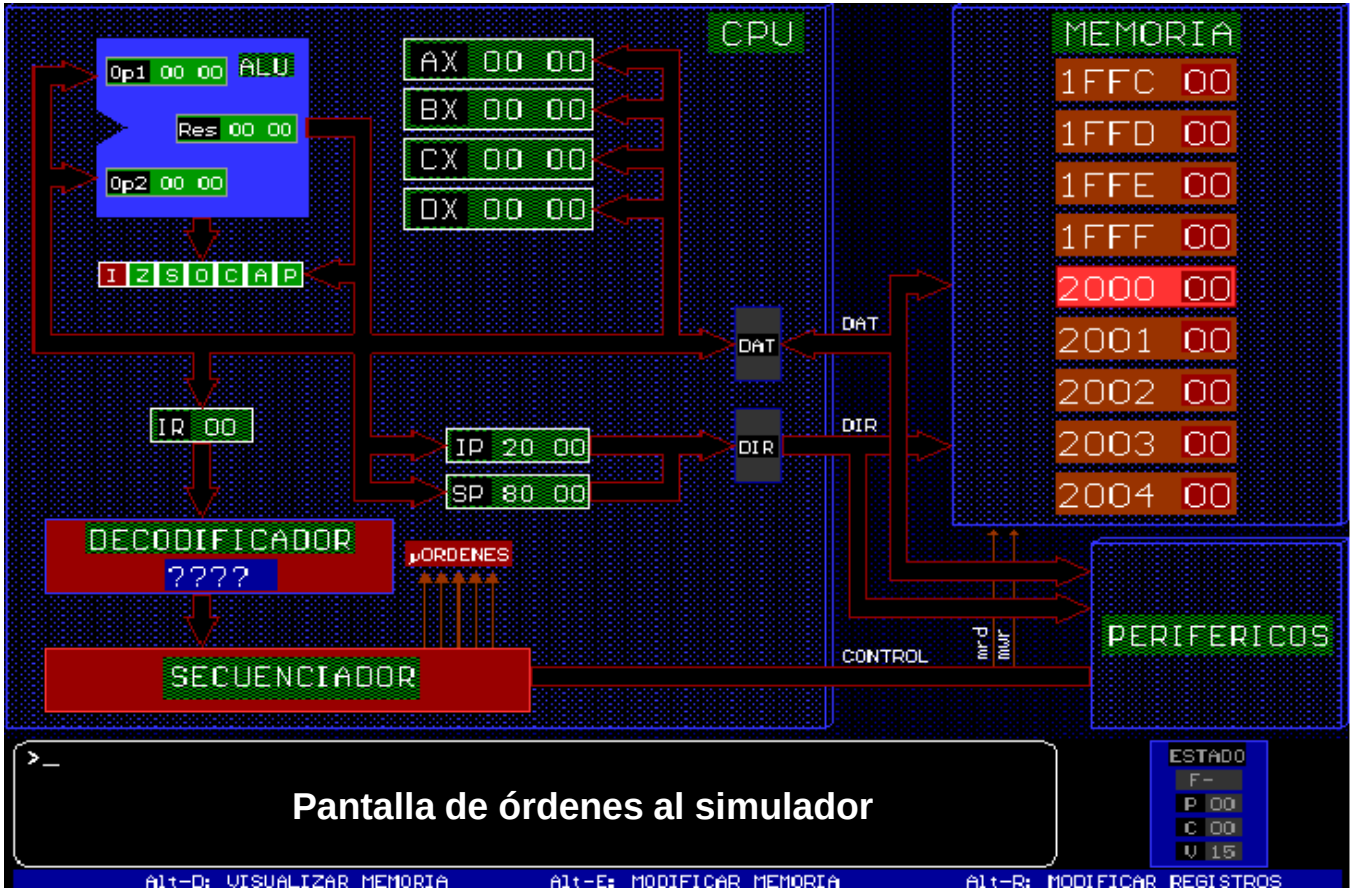


Área de Arquitectura y Tecnología de Computadores
Departamento de Automática
Universidad de Alcalá

Simulator MSX88 (I)

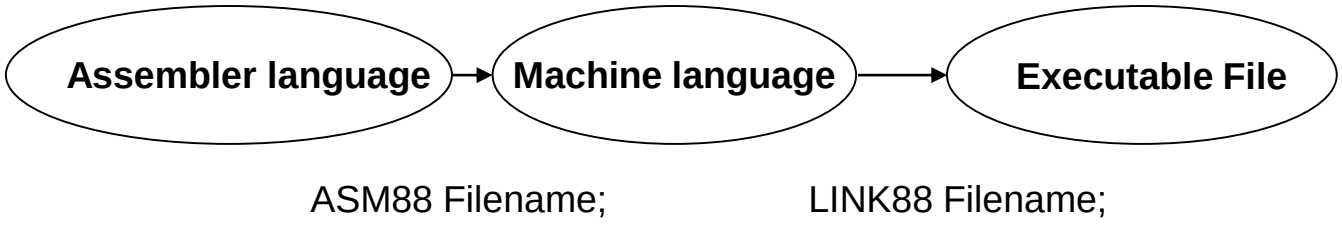
- This simulator is a simplified version of the i8088
- It has a 64Kb RAM
- Allows connection of peripherals
- It has a program to monitor the MSX88
- The instructions provided by this CPU are a real part of those offered by i8086 / 8088 microprocessors
- It has the tools:
  ASM88 (assembler)
  LINK88 (assembler program)

Simulator MSX88 (II)



Simulator MSX88 (III)

- To change an assembly file to be executable these steps must be followed:
  - Open an MS-DOS session
  - Write the command: cd C: \ MSX88
  - FileName ASM88;
  - FileName LINK88;



Simulator MSX88 (IV)

- To use MSX88 simulator you should write down:
  - MSX88 (on MS-DOS screen)

- When the MSX88 screen is open:
  - L FileName (on the order screen of simulator)
  - To execute instruction: F6 (instruction by instruction) or F7 (cycle machine)
  - R IP 2000h (on the order screen of simulator)
  - Q To leave the simulator