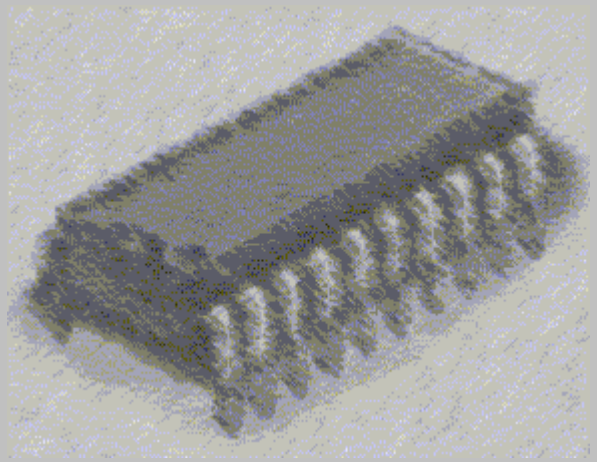
Lesson 1. Fundamentals of assembly language

Computer Structure and Organization

Graduate in Computer Sciences
Graduate in Computer Engineering



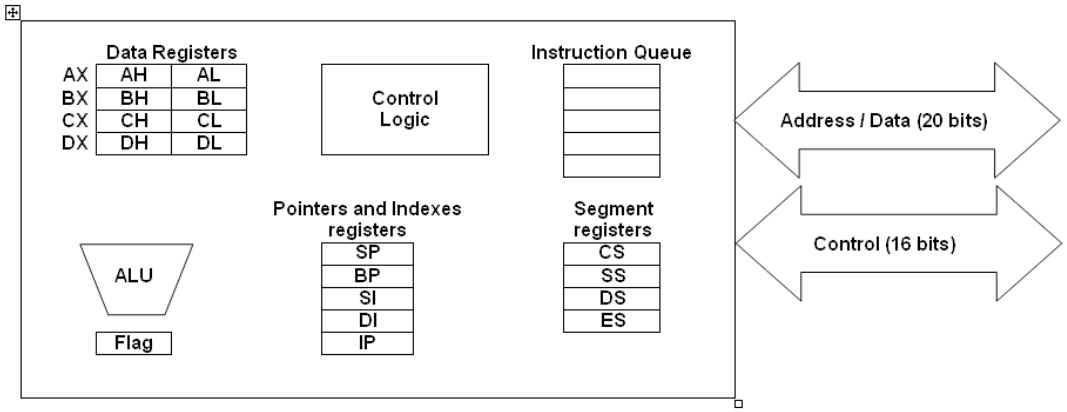
Automatic Deparment

Contents

- Intel 8086 structure
- Assembly program structure
- How to create an executable program
- Instructions types:
  - Transfer instructions
  - Arithmetic instructions
- My first assembly program: editing, assembling, linking and debugging your first assembly program

Intel i8086 structure (I)

- I8086 microprocessor has the following 14 registers:
  - Data registers
  - Segment registers
  - Stack Pointer Register
  - Index registers
  - Instruction Pointer Register
  - Flag register



Intel i8086 structure (and II)

- Data registers:
  - AX (AH, AL)
  - BX (BH, BL)
  - CX (CH, CL)
  - DX (DH, DL)

- Pointer registers:
  - SP - Stack Pointer
  - BP - Base Pointer
  - SI - Index Register
  - DI - Index Register
  - IP - Instruction Program

Assembly program structure

DOSSEG	←MS-DOS EXECUTION MODE PREPARATION
.MODEL SMALL	←COMPILE MODE DEFINITION
. STACK 100h	←STACK DEFINITION
.DATA	
DATA DEFINITIONS	←PLACE WHERE VARIABLES ARE DEFINED
.CODE	
MOV AX, @DATA	← THESE TWO INSTRUCTIONS SET THE DATA MEMORY ADDRESSES
MOV DS, AX	
PROGRAM INSTRUCTIONS	←THE INSTRUTIONS OF THE PROGRAM
MOV AH, 4Ch	← THESE TWO INSTRUCTTIONS ASK FOR END OF PROGRAM
INT 21h	← SERVICE TO MS-DOS
END	← END OF SOURCE FILE

How to create and executable file

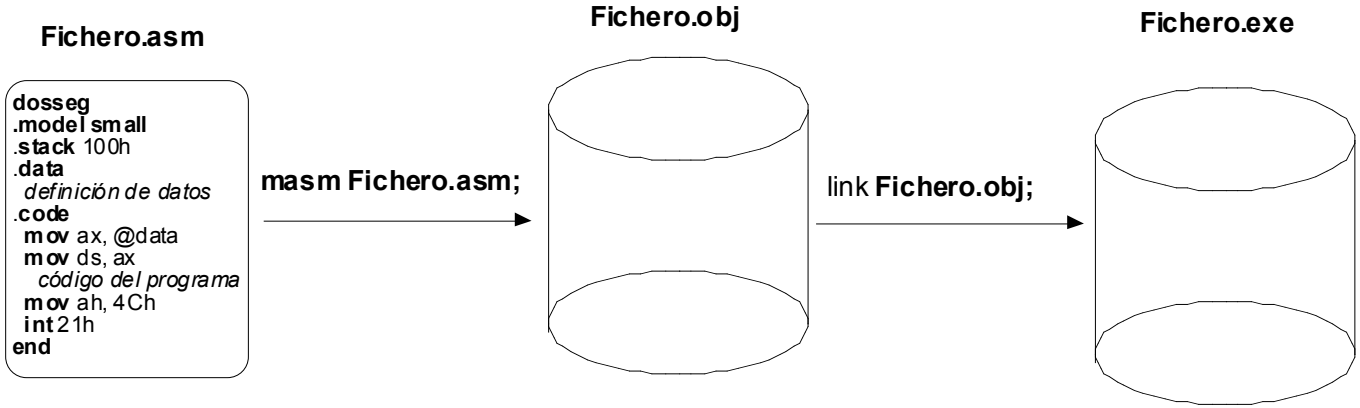
Programming:

- Proram must be writen in an ASCII plain text editor
- Source file extension must be ASM
- To assembly the source fiel type:

  MASM NombreFichero.ASM;

- You can link the executable file if no warning nor error messages appear by typeing:

  LINK NombreFichero.OBJ;



Instruction types

Instruction type	Description
Transfer instructions	Move information between registers, registers and memory positions. Usually flags are not changed
Arithmetic instructions	Perform arithmetic operations: add, subtract, etc.
Bits instructions	Logical, shift and rotary operations on the individual bits of registers or memory positions
Control transfer instructions	Control program instructions execution
Input-Output instructions	Move information between registers and I/O ports
String instructions	Perform different operations on bytes or word operands
Interrupt instructions	Accessing programmed services like reading a character, display data on the screen, etc.

Data transfer instructions (I)

- **Mnemonic:** MOV
- **Format:** MOV target, source
- **Description:**
- Transfer a byte or a word from the source operand to the target operand
- **Examples:**
  - MOV CX, 112h ; CX = 112h
  - MOV ES, AX ; ES = AX
  - MOV AL, 12h ; AL = 12h
  - MOV PAL_MEM, BX ; PAL_MEM = BX

Data transfer instructions (II)

- **Mnemonic:** PUSH
- **Format:** PUSH source
- **Description:**
  Decrement stack pointer (SP) in 2 and then transfers source word to the top of the stack
- **Example:**
  - PUSH BX ; Store BX on the top of the stack

Data transfer instructions (and III)

- **Mnemonic:** POP
- **Format:** POP target
- **Description:**
  Transfers a byte or word from the top of the stack to the target and the increment in two the stack pointer register
- **Example:**
  - POP BX ; Store in BX the top stack content

Arithmetic instructions (I)

- **Mnemonic:** ADD
- **Format:** ADD target, source
- **Description:**
  Adds two operands and the result is stored in target. Operand must have the same size
- **Examples:**
  - ADD CL, BL ; CL = CL + BL
  - ADD AL, 12h ; AL = AL + 12h
  - ADD CX, DX ; CX = CX + DX

Arithmetic instructions (II)

- **Mnemonic:** ADC
- **Format:** ADC target, source
- **Description:**
  Adds two operands and carry flag, the result is stored in target.
  Operand must have the same size
- **Examples:**
  - ADC CL, BL ; CL = CL + BL + CF
  - ADC AL, 12h ; AL = AL + 12h + CF
  - ADC CX, DX ; CX = CX + DX + CF

Arithmetic instructions (III)

- **Mnemonic:** SUB
- **Format:** SUB target, source
- **Description:**
  Subtracts two operands and the result is stored in target. Operand must have the same size
- **Examples:**
  - SUB CL, BL ; CL = CL - BL
  - SUB AL, 12h ; AL = AL - 12h
  - SUB CX, DX ; CX = CX - DX

Arithmetic instructions (IV)

- **Mnemonic:** SBB
- **Format:** SBB target, source
- **Description:**
  Subtracts two operands and carry flag, the result is stored in target. Operand must have the same size
- **Example:**
  - SBB CX, DX ; CX = CX - DX - CF

Arithmetic instructions (V)

- **Mnemonic:** MUL
- **Format:** MUL source
- **Description:**
  Multiplies two **unsigned** numbers. Operands are the AL or AX register and the source operand. The result is stored in AX register if source operand is one byte length. The result is stored in the concatenation of DX and AX if source operand is a word.
- **Example:**
  - ; AX = 1234h
  - ; BX = 1000h
  - MUL BX ; DX = 0123h, AX = 4000h

Arithmetic instructions (VI)

- **Mnemonic:** IMUL
- **Format:** IMUL source
- **Description:**
  Multiplies two **signed** numbers. Operands are the AL or AX register and the source operand. The result is stored in AX register if source operand is one byte length. The result is stored in the concatenation of DX and AX if source operand is a word.
- **Examples:**
  - ; AL = FEh = -2
  - ; BL = 12h = 18
  - IMUL BL ; AX = FFDCh = -36

Arithmetic instructions (VII)

- **Mnemonic:** DIV
- **Format:** DIV source
- **Description:**
  Divides **unsigned** numbers. AL or AX and their extended registers (AH or DX) by source operand. The quotient is stored in AL or AX, depending on the source size (byte or word).

  The remain is stored in AH or DX, depending on the source size (byte or word).
- **Examples:**
  - ; AX = 0013h = 19
  - ; BL = 02h = 2
  - DIV BL ; AH = 1, AL = 9

Arithmetic instructions (VIII)

- **Mnemonic:** IDIV
- **Format:** IDIV source
- **Description:**
  Divides **signed** numbers. AL or AX and their extended registers (AH or DX) by source operand. The quotient is stored in AL or AX, depending on the source size (byte or word).

  The remain is stored in AH or DX, depending on the source size (byte or word).
- **Examples:**
  - ; AX = FFEDh = -19
  - ; BL = 02h = 2
  - IDIV BL ; AH = 1, AL = F7h = -9

Arithmetic instructions (IX)

- **Mnemonic:** INC
- **Format:** INC target
- **Description:**
  Adds one to target operand. Operand size could be byte or word.
- **Examples:**
  - ; AX = 1234h
  - INC AX ; AX = 1235h
  - INC AH ; AH = 13h

Arithmetic instructions (X)

- **Mnemonic:** DEC
- **Format:** DEC target
- **Description:**
  Subtracts one to target operand. Operand size could be byte or word.
- **Examples:**
  - ; AX = 1234h
  - DEC AX ; AX = 1233h
  - DEC AH ; AH = 11h

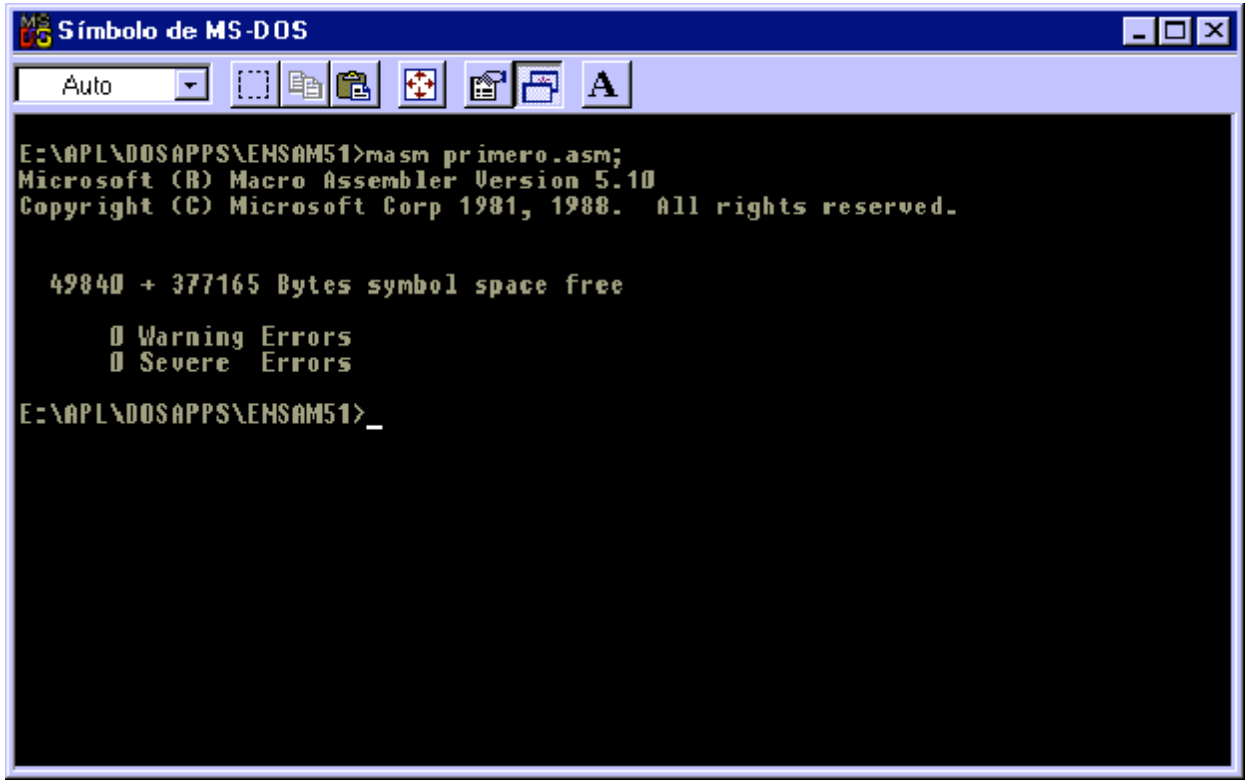
Arithmetic instructions (and XI)

- **Mnemonic:** NEG
- **Format:** NEG target
- **Description:**
  Changes the sign of the target operand. Representation system is complement 2. Operand size could be byte or word.
- **Example:**
  - NEG AL ; If AL = F2h before instruction, after instruction AL = 0Eh

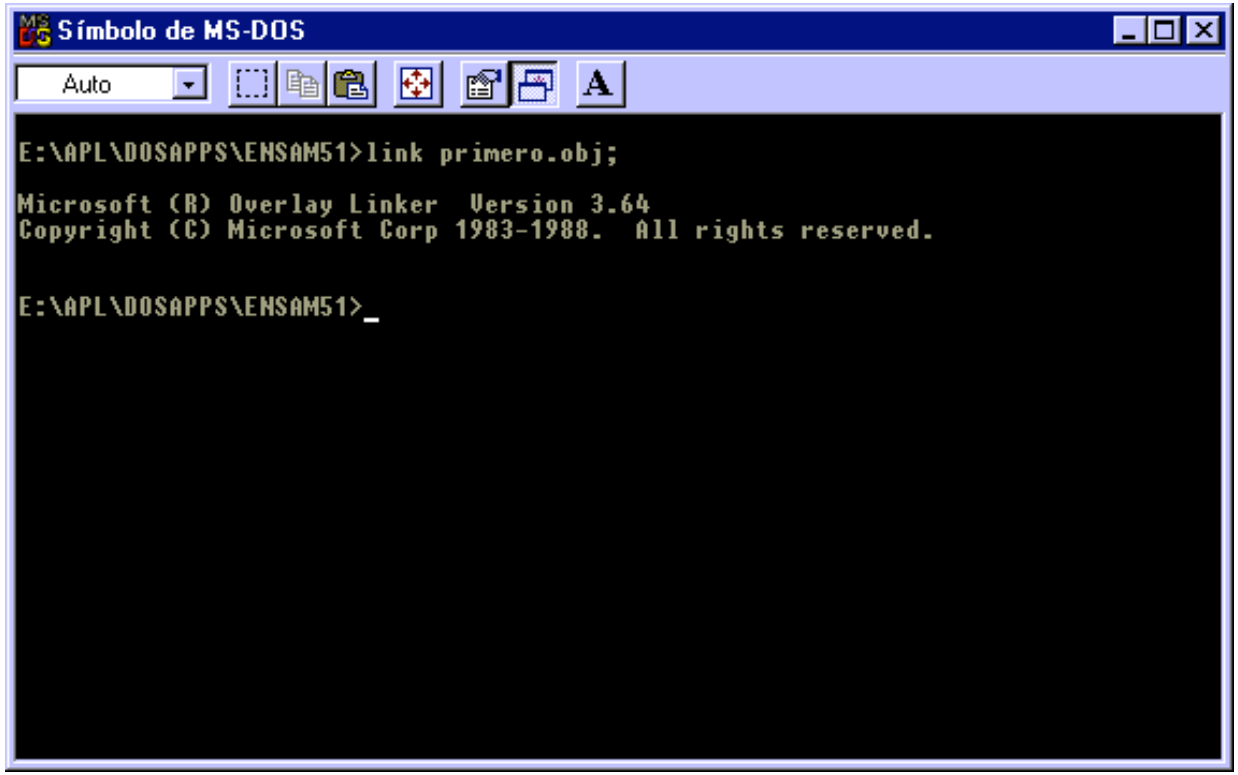
My first assembly program (I)
Program primero.asm file editing process

```
dosseg
.model small
.stack 100h
.data
    num1        db 12h
    num2        db 10h
.code
    mov ax, @data
    mov ds, ax
    mov al, num1
    mov bl, num2
    mul bl
    mov ah, 4Ch
    int 21h
end
```

My first assembly program (II)
Program primero.asm file assembling process



My first assembly program (III)
Program primero.obj file linking process

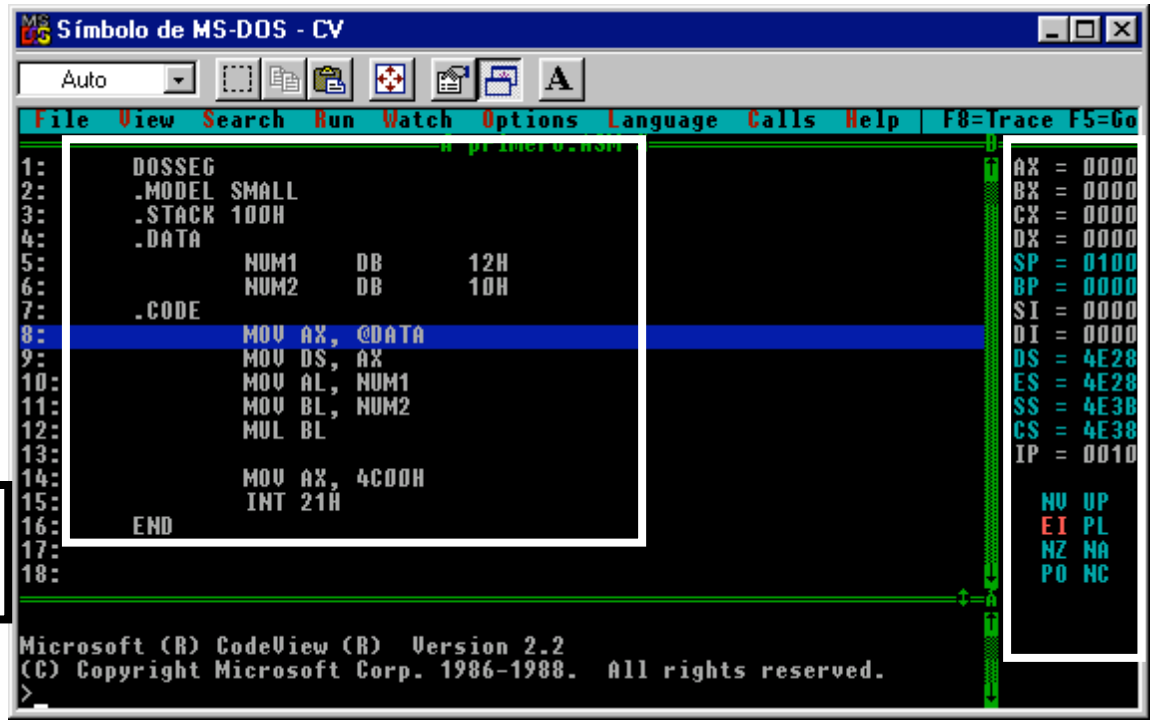


My first assembly program (IV)
Program step by step execution: Code View (I)

Debugging:

- Step by step execution of programs is possible in order of correcting programming errors.
- You can do that with CODEVIEW program
- Syntax: CV NombreFichero.EXE
- Depress F2 for showing / hiding registers window
- Depress F8 for step by step execution (jump into procedures option)
- Depress F10 for step by step execution (procedures are excluded)

My first assembly program (and V)
Program step by step execution: Code View (&II)



Code Area

I8086 Registers

Commands area

Bibliography

- 8088-8086/8087 programación ensamblador en entorno MS-DOS
  Miguel Angel Roselló.
  Ed. Anaya Multimedia.

- Arquitectura, programación y diseño de sistemas basados en microprocesadores (8086/80186/80286)
  Yu-Cheng Lu, Glen A. Gibson.
  Ed. Anaya Multimedia.

- Lenguajes ensambladores
  R. Martínez Tomás.
  Ed. Paraninfo