Lab # 5

# OBJECTive

Describe some basic instructions of Intel Architecture

# Theory

# intel:

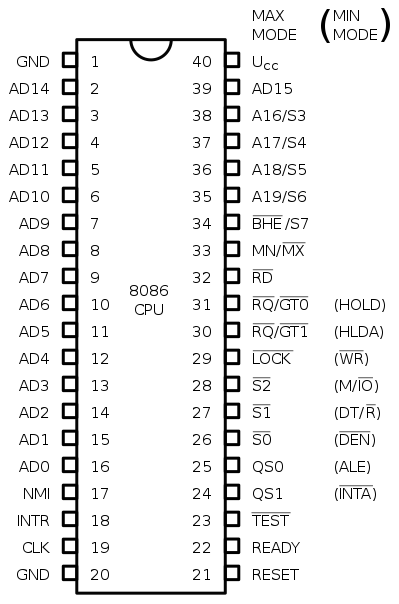
**IA-32** (short for "Intel Architecture, 32-bit", sometimes also called i386) is the [32-bit](https://en.wikipedia.org/wiki/32-bit) version of the [x86](https://en.wikipedia.org/wiki/X86) [instruction set architecture](https://en.wikipedia.org/wiki/Instruction_set_architecture), first implemented in the [Intel 80386](https://en.wikipedia.org/wiki/Intel_80386)[microprocessors](https://en.wikipedia.org/wiki/Microprocessor) in 1985. IA-32 is the first incarnation of x86 that supports 32-bit computing, as a result, the "IA-32" term may be used as a metonym to refer to all x86 versions that support 32-bit computing.

Intel" is the term for intelligence information also made the name appropriate. Intel was an early developer of [SRAM](https://en.wikipedia.org/wiki/Static_random-access_memory) and [DRAM](https://en.wikipedia.org/wiki/Dynamic_random-access_memory) memory chips, which represented the majority of its business until 1981. Although Intel created the world's first commercial microprocessor chip in 1971, it was not until the success of the [personal computer](https://en.wikipedia.org/wiki/Personal_computer) (PC) that this became its primary business.

During the 1990s, Intel invested heavily in new microprocessor designs fostering the rapid growth of the [computer industry](https://en.wikipedia.org/wiki/Computer_industry). During this period Intel became the [dominant](https://en.wikipedia.org/wiki/Market_dominance) supplier of microprocessors for PCs and was known for aggressive and anti-competitive tactics in defense of its market position,

The 8086 (also called iAPX 86) is a [16-bit](https://en.wikipedia.org/wiki/16-bit) [microprocessor](https://en.wikipedia.org/wiki/Microprocessor) chip designed by [Intel](https://en.wikipedia.org/wiki/Intel) between early 1976 and mid-1978, when it was released. The 8086 gave rise to the [x86 architecture](https://en.wikipedia.org/wiki/X86_architecture), which eventually became Intel's most successful line of processors.

**PIN DIAGRAM:**



**BASIC INSTRUCTION OF ASSEMBLY LANGUAGE:**

**MOV INSTRUCTION (MOV):**

MOV is the basic instruction that moves the constant data in the register or move that data from one register to another

**RESTRICTIONS:**

* Move between memory to memory is not allowed.
* A number directly inside a segment register is not allowed.
* Segment to segment registers, move is not allowed.

# Interrupt Instruction

Pentium processor has two memory architectures: real and protected. In real mode a Pentium works like fast 8086 processor. Real mode uses 16 bit addresses. The Real mode is also called as 16-bit mode, because all 20 bit physical address is constructed by 16 bit address. MS-DOS Operating system was the first operating system to implement Real-Address mode on IBM personal computer.

The INT instruction is the instruction which does the most work in any assembler program. INT instruction calls a DOS interrupt service routine (like a function) to perform a special task.

**INT Interrupt Number**

Where Interrupt Number ranges from 00H to 0FFH (i.e., from 0 to 255).

MS-DOS Operating system provides many common services through INT 21h. INT 21h MS-DOS services are procedures that provide input-output, file handling, and memory management. They are also called “MS-DOS function calls.”

The execution of an INT instruction causes an Interrupt Service Routine (ISR) associated with the Interrupt Number to be executed. Many of the ISRs have multiple sub-functions. To specify which sub-function is to be executed under a particular Interrupt Number, the AH register is assigned a sub-function number before the execution of the INT instruction.

**Example:**

**MOV AH, 08H**

**INT 21H**

Causes sub-function number 08H of Interrupt number 21H to be executed. In addition, some sub-functions require other values to be passed to the ISR in particular registers. Example: Sub-function 09H of Interrupt 21H displays a $-terminated string on the screen. The sub-function requires the offset of that string to be passed in the DX register:

**MOV DX, OFFSET STRING**

**MOV AH, 09H**

**INT 21H**

## Note: DPMI (DOS Protected Mode Interface) is an interface allowing a [DOS](http://en.wikipedia.org/wiki/DOS) program to run in [protected mode](http://en.wikipedia.org/wiki/Protected_mode) and to access [extended memory](http://en.wikipedia.org/wiki/Extended_memory) under a multitasking operating like [Microsoft Windows](http://en.wikipedia.org/wiki/Microsoft_Windows) 3.0 and later.

## DOS FUNCTION CALLS (INT 21H)

DOS function calls preserve the contents of all the registers except the AX register and any other register or registers in which they explicitly return data.

### TERMINATE PROGRAM AND RETURN TO DOS

Every time you want to terminate the program and return to DOS, you have to put the following codes:

| **Assembly Language** | **C Language** | **Meaning** |
| --- | --- | --- |
| mov AX , 4C00H  int 21h | exit(0) | Program terminates normally |
| mov AX, 4C01h  int 21h | exit(1) | Program terminates with error code 1. |

### CHARACTER INPUT WITH ECHO

To take single input character thru a keyboard, you have to put the following codes:

|  |  |
| --- | --- |
| **The Codes** | **The Result** |
| mov AH, 01h  int 21h | The program is waiting for the input. Once a user presses a key, the ASCII Code of the input character is returned in the **AL** register and the input character is displayed as well. |

NOTE: This service makes the program waits for the input. The user just needs to press the intended key WITHOUT pressing "enter" key.

### CHARACTER INPUT WITHOUT ECHO

MOV AH, 08H

INT 21H

The code of the input character is returned in the AL register.

**CHARACTER OUTPUT**

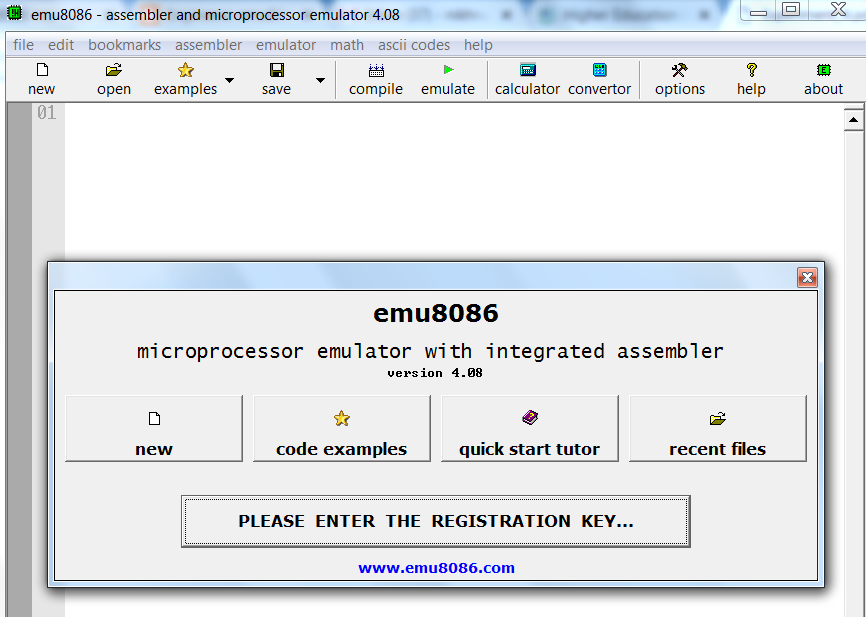
To display a character, you have to use the DOS function 02h.

|  |  |
| --- | --- |
| **The Initial requirement** | **The result** |
| AH = 02h  DL = Character or ASCII Code | The character stored in DL will be displayed. |

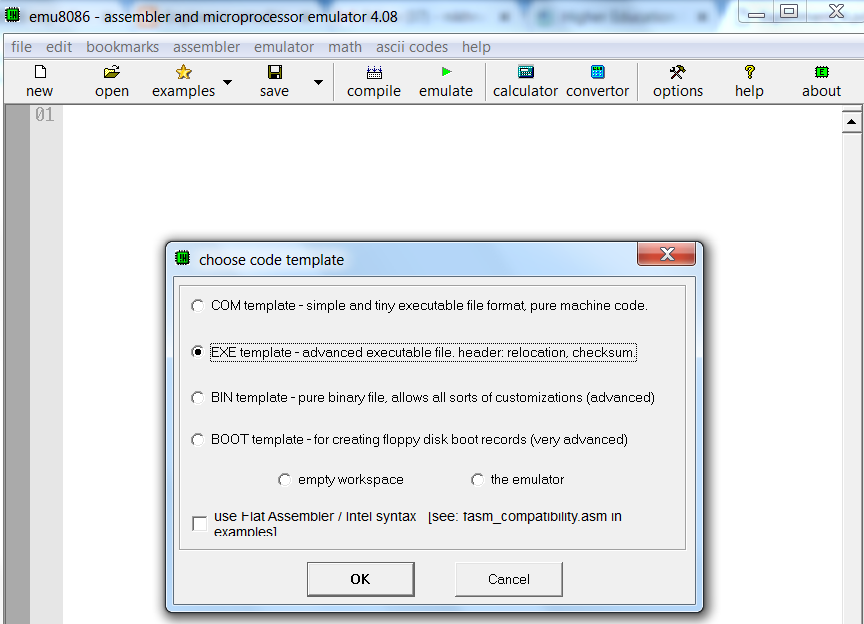
**EMU 8086**

EMULATOR is a free emulator for multiple platforms. It provides its user with the ability to emulate old 8086 processors, which were used in Macintosh and Windows computers from the 1980s and early 1990s.

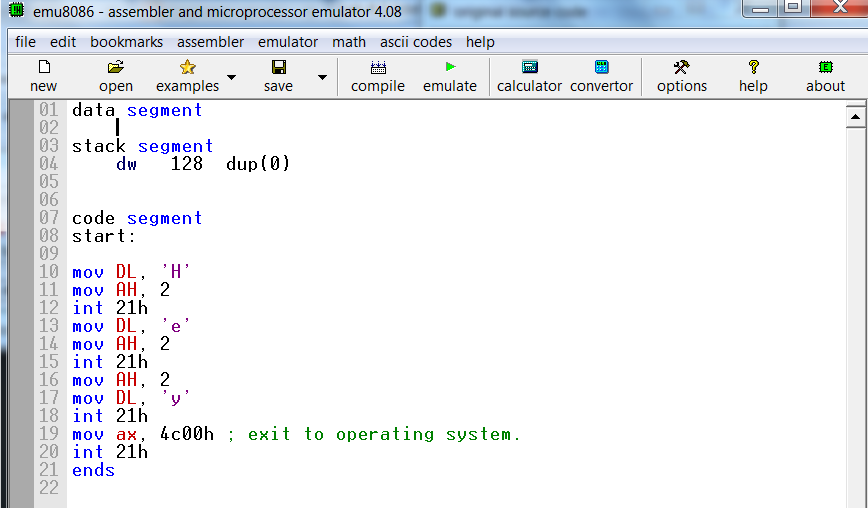
**STEP#1**

****

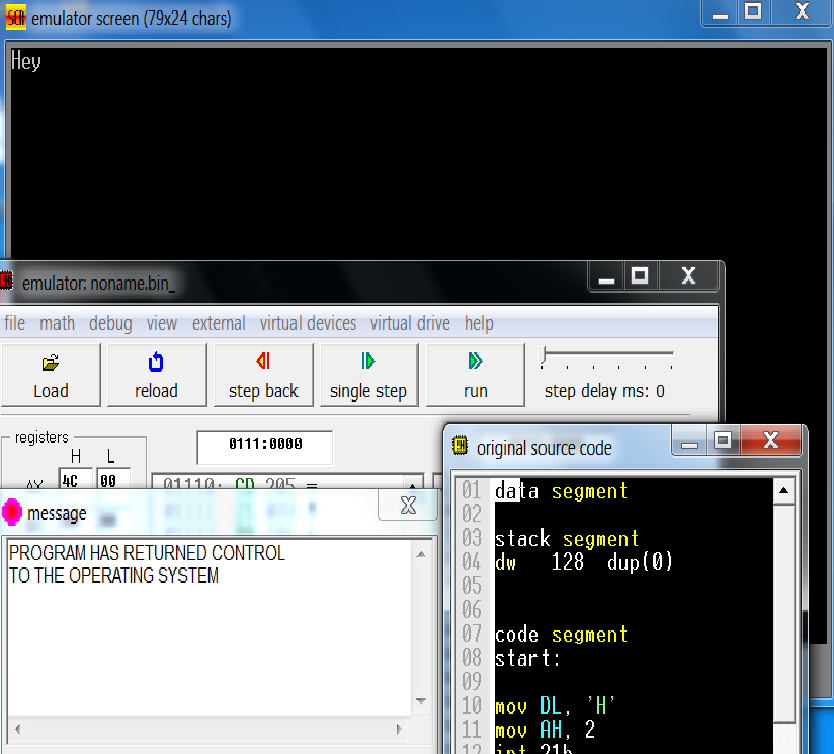
**STEP#2**

****

**STEP#3**

****

**STEP#4 (FINAL OUTPUT):**

****

**EXERCISE:**

Task#1: Write a program to display your name using ‘Mov instruction’.