**PRACTICAL – 1**

**AIM:**

|  |
| --- |
| **Introduction to 8086 Microprocessor & Assembly Language Programming.** |

**THEORY:**

**8086 Microprocessor**

* Intel 8086 microprocessor is the enhanced version of Intel 8085 microprocessor. It was designed by Intel in 1976.
* The 8086 microprocessor is a16-bit, N-channel, HMOS microprocessor. Where the HMOS is used for "**High-speed Metal Oxide Semiconductor**".
* Intel 8086 is built on a single semiconductor chip and packaged in a 40-pin IC package. The type of package is DIP (Dual Inline Package).
* Intel 8086 uses 20 address lines and 16 data- lines. It can directly address up to 220 = 1 Mbyte of memory.
* It consists of a powerful instruction set, which provides operation like division and multiplication very quickly.
* 8086 is designed to operate in two modes, i.e., Minimum and Maximum mode.

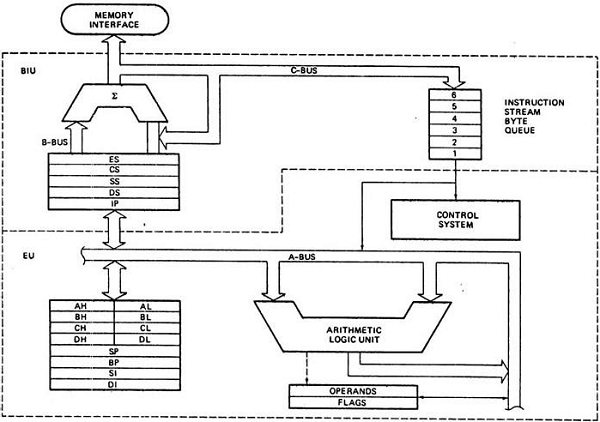
## Features of 8086

The most prominent features of a 8086 microprocessor are as follows −

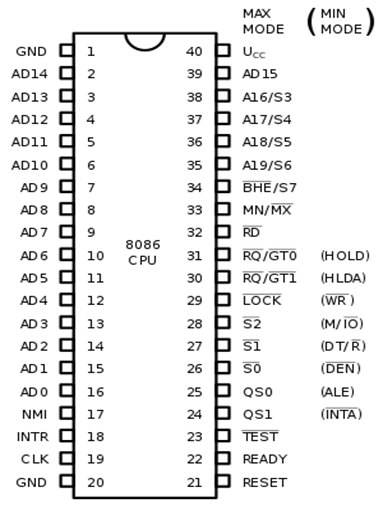
* It has an instruction queue, which is capable of storing six instruction bytes from the memory resulting in faster processing.
* It was the first 16-bit processor having 16-bit ALU, 16-bit registers, internal data bus, and 16-bit external data bus resulting in faster processing.
* It is available in 3 versions based on the frequency of operation −
  + 8086 → 5MHz
  + 8086-2 → 8MHz
  + (c)8086-1 → 10 MHz
* It uses two stages of pipelining, i.e. Fetch Stage and Execute Stage, which improves performance.
* Fetch stage can prefetch up to 6 bytes of instructions and stores them in the queue.
* Execute stage executes these instructions.
* It has 256 vectored interrupts.
* It consists of 29,000 transistors.

## Architecture of 8086

The following diagram depicts the architecture of a 8086 Microprocessor.



## 8086 pins configuration



**Assembly Language Programming**

* The assembly programming language is a low-level language which is developed by using mnemonics. The microcontroller or microprocessor can understand only the binary language like 0's or 1's therefore the assembler convert the assembly language to binary language and store it the memory to perform the tasks.

**CONCLUSION:**

**PRACTICAL – 2**

**AIM:**

|  |
| --- |
| **Store the data byte 32H into memory location 4000H.** |

**CODE:**

ORG 100H

MOV [4000H],32H

MOV AX,[4000H]

MOV [2000H],AX

RET

**OUTPUT:**

****

**CONCLUSION:**

**PRACTICAL – 3**

**AIM:**

|  |
| --- |
| **Exchange the contents of memory locations 2000H and 4000H** |

**CODE:**

ORG 100H

ORG 100H

MOV [4000H],32H

MOV [2000H],16H

MOV AX,[4000H]

MOV BX,[2000H]

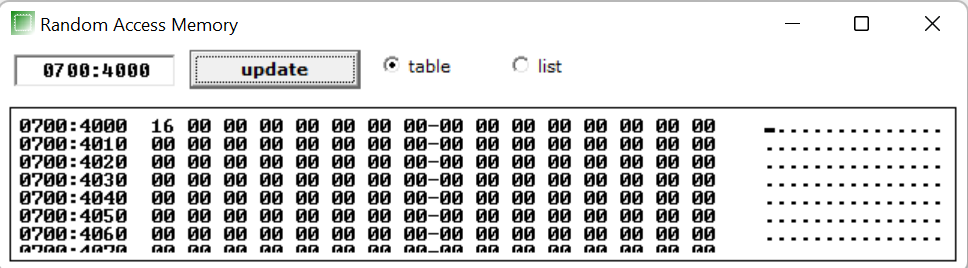
MOV [4000H],BX

MOV [2000H],AX

RET

**OUTPUT:**

****

****

**CONCLUSION:**

**PRACTICAL – 4**

**AIM:**

|  |
| --- |
| **Convert the below given C Program into Assembly Language.**  **main()**  **{**  **Int l, n, o, p;**  **l = m + n + o + p;**  **}** |

**CODE:**

ORG 100H

MOV BL,05H

MOV CL,04H

MOV DL,02H

MOV AH,05H

ADD BL,CL

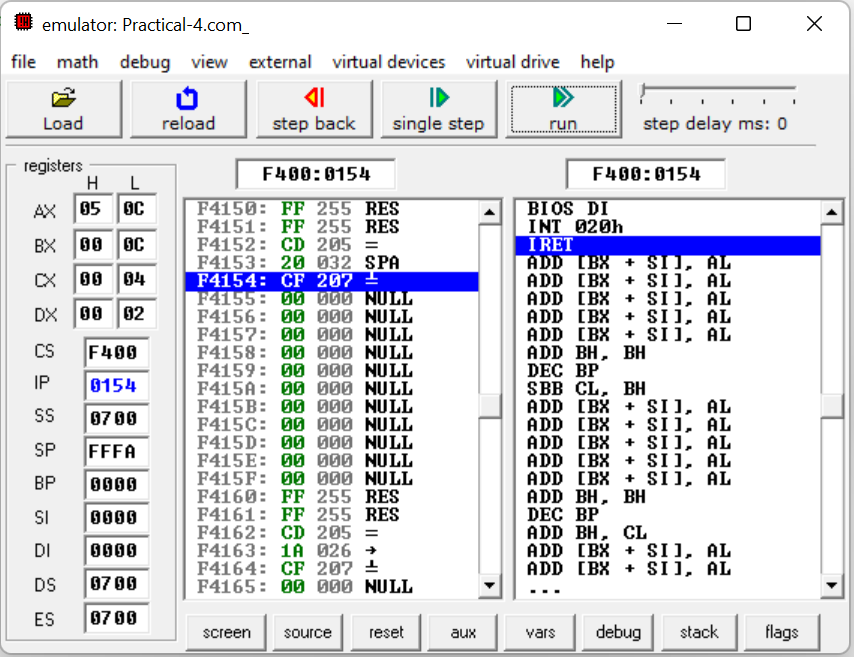
SUB BL,DL

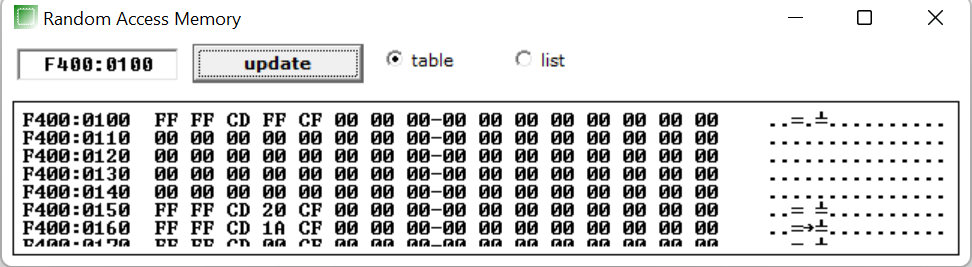
ADD BL,AH

MOV AL,BL

RET

**OUTPUT:**



****

**CONCLUSION:**

**PRACTICAL – 5**

**AIM:**

**Subtract the contents of memory location 4001H from the memory location 2000H and place the result in memory location 4002H.**

**CODE:**

ORG 100H

MOV [4000H],0505H

MOV [2000H],0202H

MOV AX,[4000H]

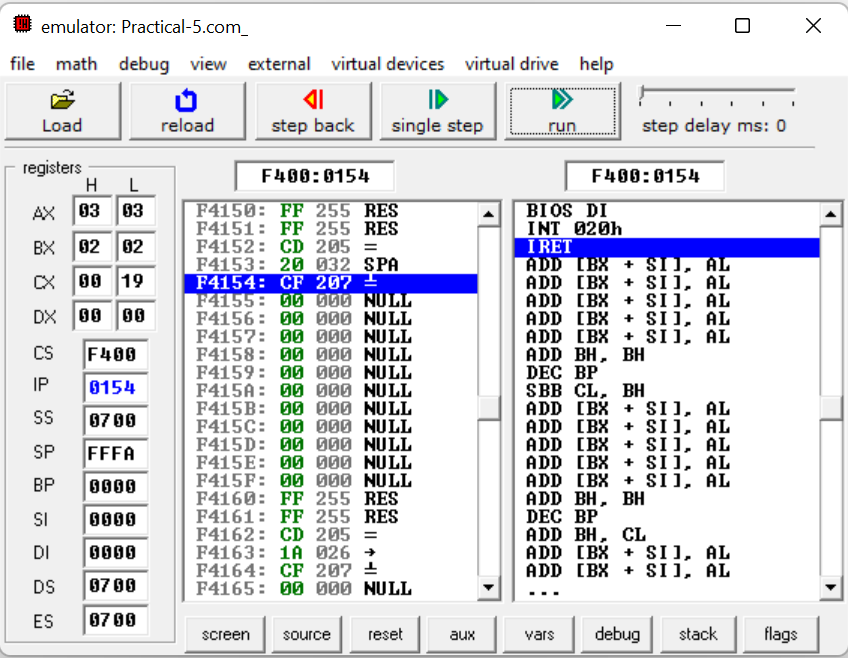
MOV BX,[2000H]

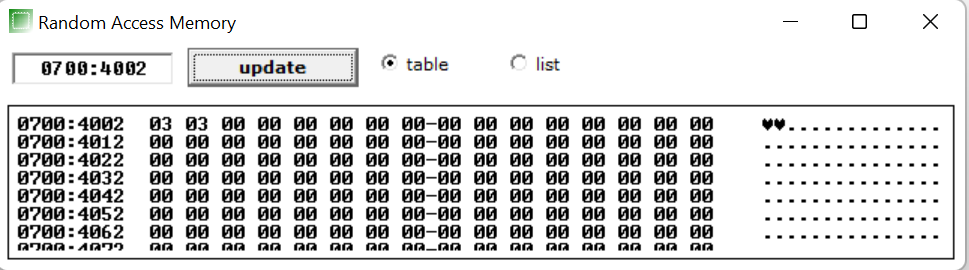
SUB AX,BX

MOV [4002H],AX

RET

**OUTPUT:**



****

**CONCLUSION:**

**PRACTICAL – 6**

**AIM:**

**Add the 16-bit number in memory locations 4000H and 4001H to the 16-bit number in memory locations 4002H and 4003H. The most significant eight bits of the two numbers to be added are in memory locations 4001H and 4003H. Store the result in memory locations 4004H and 4005H with the most significant byte in memory location 4005H.**

**CODE:**

ORG 100H

MOV [4000H],5050H

MOV [4002H],2020H

MOV AX,[4000H]

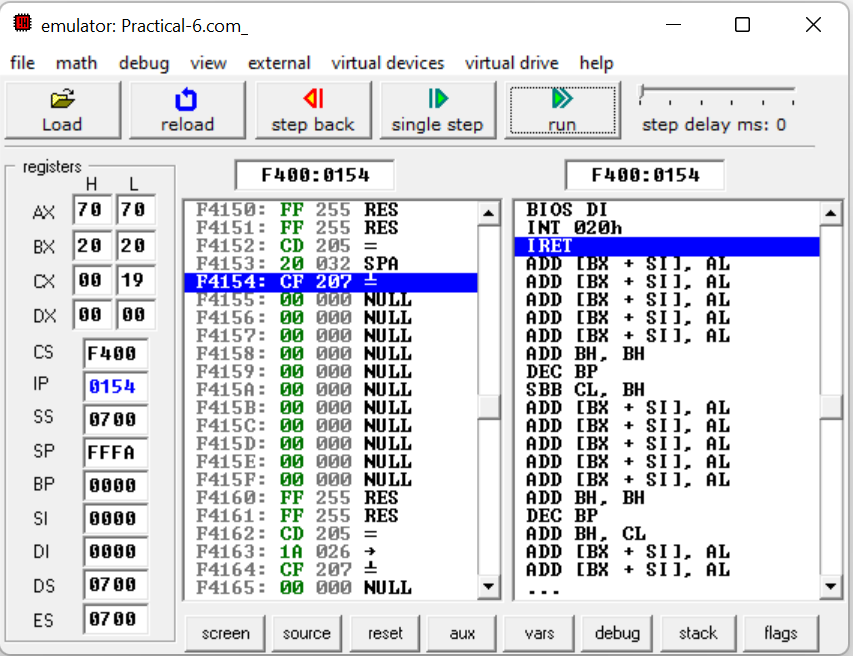
MOV BX,[4002H]

ADD AX,BX

MOV [4004H],AX

RET

**OUTPT:**



****

**CONCLUSION:**

**PRACTICAL – 7**

**AIM:**

**Subtract the 16-bit number in memory locations 4002H and 4003H from the 16-bit number in memory locations 4000H and 4001H. The most significant eight bits of the two numbers are in memory locations 4001H and 4003H. Store the result in memory locations 4004H and 4005H with the most significant byte in Memory location 4005H.**

**CODE:**

**CONCLUSION:**

**PRACTICAL – 8**

**AIM:**

Add Two 32-bit numbers stored in consecutive memory locations and store the result in memory locations starting from 7000H

**CODE:**

ORG 100H

MOV AX, 5050H

MOV BX, 2020H

MOV CX, 4000H

MOV DX, 4002H

SUB AX,CX

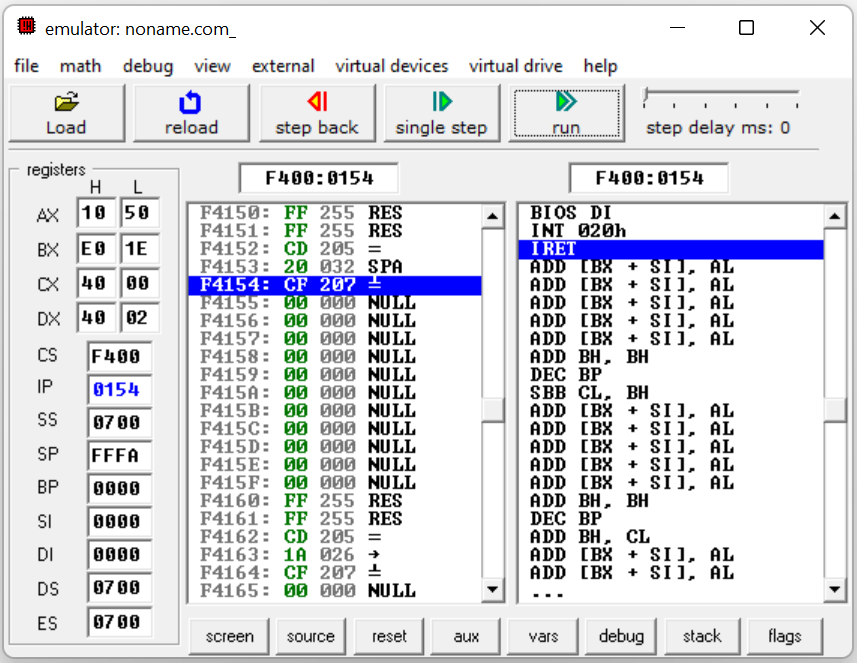
SUB BX,DX

MOV [7000H],AX

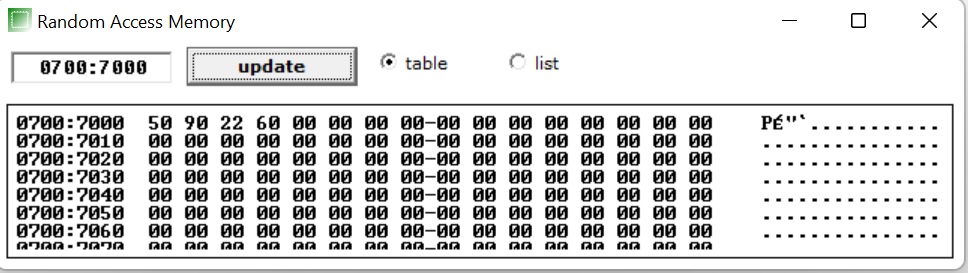
MOV [7002H],BX

RET

**OUTPUT:**

****

****

****

**CONCLUSION:**

**PRACTICAL – 9**

**AIM:**

Subtract Two 32-bit numbers stored in consecutive memory locations and store the result in memory locations starting from 7000H

**CODE:**

ORG 100H

MOV AX, 5050H

MOV BX, 2020H

MOV CX, 4000H

MOV DX, 4002H

SUB AX,CX

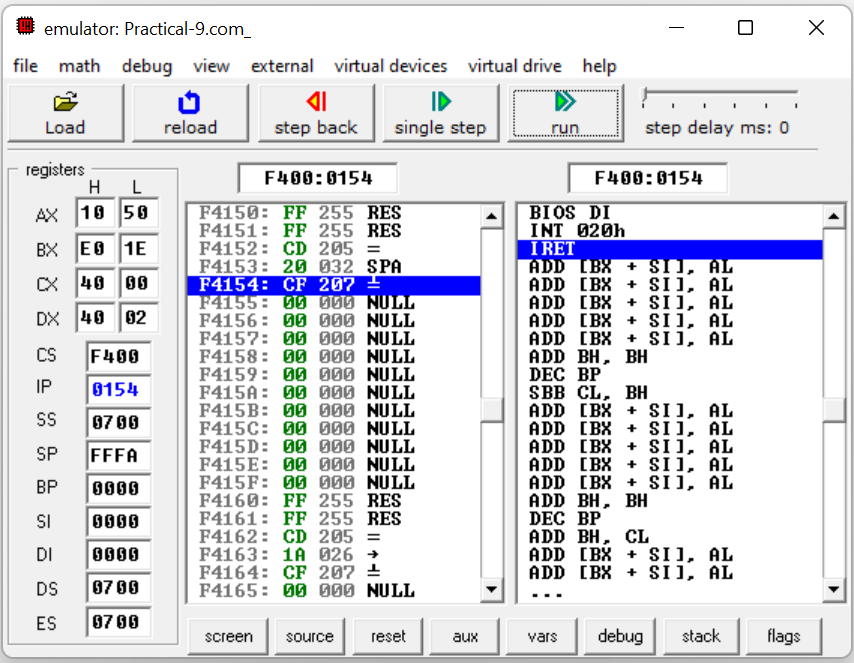
SUB BX,DX

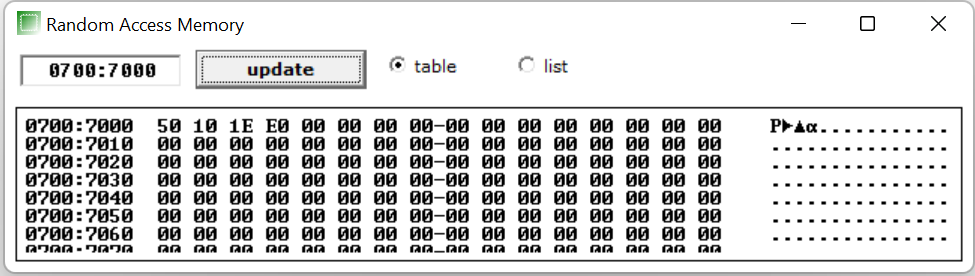
MOV [7000H],AX

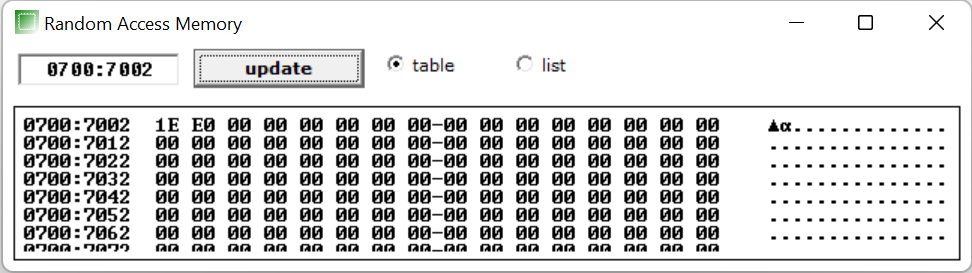
MOV [7002H],BX

RET

**OUTPUT:**

****

****

****

**CONCLUSION:**

**PRACTICAL – 10**

**AIM:**

Write an assembly language program to convert temperature in F to C. C=(F-32) \* 5/9

**CODE:**

**OUTPUT:**

**CONCLUSION:**

**PRACTICAL – 1**

**AIM:**

**CODE:**

**OUTPUT:**

**CONCLUSION:**