Date: 23/02/2021

Student's name	Anuj Shah
Roll Number	18104B0024
Name of Professor	Bharat Acharya

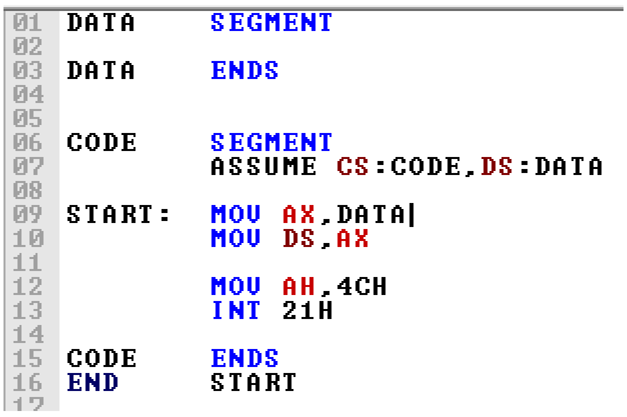
Experiment number	1
Experiment title	My first x86 assembly program
Hardware requirement	–
Software requirement	emu8086

Aim	To write a program to add two 16-bit numbers.

Theory

Boilerplate in 8086 programming: In programming, boilerplate code are sections of code that are repeated in multiple places with little to no variation.
Source: https://en.wikipedia.org/wiki/Boilerplate_code

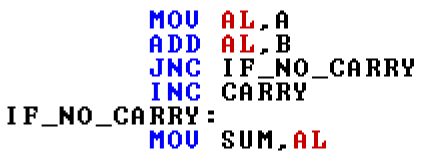
In x86 assembly programming, there is a lot of boilerplate:



```
DATA    SEGMENT

DATA    ENDS


CODE    SEGMENT
        ASSUME CS:CODE,DS:DATA

START:  MOV  AX,DATA
        MOV  DS,AX

        MOV  AH,4CH
        INT  21H

CODE    ENDS
END     START
```

Algorithm/ Flowchart

While the whole program is big (due to boilerplate), the heart of the program (which is actually responsible for performing addition) is quite small:

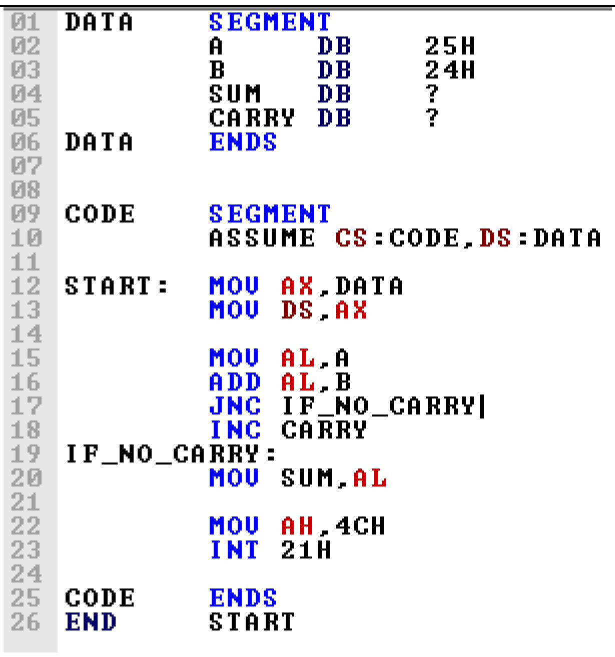


```
            MOV  AL,A
            ADD  AL,B
            JNC  IF_NO_CARRY
            INC  CARRY
IF_NO_CARRY:
            MOV  SUM,AL
```

1. "MOV AL,A" moves the hex number 0x25(decimal = 37) into register AL.
2. "ADD AL,B" adds the hex number 0x24 (decimal = 36) to 0x25 which is already stored in AL; and stores the result (hex = 0x49, decimal = 73) in register AL.
3. Because the sum of 0x24 and 0x49 doesn't generate any carry, thus the "JNC IF_NO_CARRY" instruction causes our program to jump to

	the IF_NO_CARRY subroutine. If the numbers A and B were different, such that the carry flag was indeed activated, then the JNC (Jump if No Carry) flag would not jump to the IF_NO_CARRY segment. Rather, the program would continue in its regular sequence, thus reaching the "INC CARRY" instruction, changing the CARRY variable from 0 to 1. **IF_NO_CARRY subroutine:** 1. Moves the hex number 0x49 into the variable named SUM.

Program



```
DATA      SEGMENT
          A       DB      25H
          B       DB      24H
          SUM     DB      ?
          CARRY   DB      ?
DATA      ENDS


CODE      SEGMENT
          ASSUME CS:CODE,DS:DATA

START:    MOV  AX,DATA
          MOV  DS,AX

          MOV  AL,A
          ADD  AL,B
          JNC  IF_NO_CARRY|
          INC  CARRY
IF_NO_CARRY:
          MOV  SUM,AL

          MOV  AH,4CH
          INT  21H

CODE      ENDS
END       START
```

Results/ Output	<u>variables</u> A 25h B 24h SUM 49h CARRY 00h
Conclusion	In the world of Assembly programming, a program which adds two bytes is similar to "Hello World" in higher-level languages; it may seem very trivial, but it allows beginners to get comfortable with the fundamentals.

Faculty Sign   Grade received