



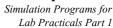
## 1) Program to add two 8-bit numbers, without user inputs or output.

SEGMENT DB OFFH Α DB OFFH SUM DB CARRY DB 00H DATA ENDS CODE SEGMENT ASSUME CS:CODE, DS:DATA START: MOV AX, DATA ; INITIALISE DS MOV DS, AX ; TAKE 1ST NUMBER INTO A MOV AL, A ; ADD 2ND NUMBER TO A ADD AL, B ; IF NO CARRY THEN SKIP ; ELSE INCREMENT CARRY ; STORE SUM JNC SKIP INC CARRY SKIP: MOV SUM, AL MOV AH, 4CH ; END THE PROGRAM... GO BACK TO DOS INT 21H CODE ENDS

END START

# Notes from Bharat Acharya

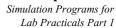
- 1) Copy and paste the above program in EMU8086 simulator as a new file
- 2) Click on "save", to save the file
- 3) Click on "emulate", to build the object code
- 4) Click on "run", to execute the program
- 5) Click on "vars" to observe the result in your variables
- 6) Run it several times and test different input numbers to verify the result
- 7) Erase the program and try to code it by yourself. Feel free to change variable names and even play with different registers. This is your first Assembly program... Own it!





### 2) Add two 8-bit numbers with user input and output

```
DATA SEGMENT
     MSG1 DB 0DH, 0AH, "PLEASE ENTER THE 1ST NUMBER: $"
     MSG2 DB ODH, OAH, "PLEASE ENTER THE 2ND NUMBER: $"
     MSG3 DB 0DH, 0AH, 0DH, 0AH, "SUM: $" MSG4 DB 0DH, 0AH, "CARRY: $"
     A DB ?
B DB ?
     SUM DB
               ?
     CARRY DB 00H
DATA ENDS
CODE SEGMENT
    ASSUME CS:CODE, DS:DATA
START:
     MOV AX, DATA ; INITIALISE DS
     MOV DS, AX
     MOV AH, 09H
                    ; DISPLAY HEADERS
     LEA DX, MSGH1
     INT 21H
     LEA DX, MSGH2
     INT 21H
     LEA DX, MSGH3
     INT 21H
     LEA DX, MSGH4
     INT 21H
     LEA DX, MSGH5
     INT 21H
                  ; ASK FOR 1ST NUMBER
     MOV AH, 09H
     LEA DX, MSG1
     INT 21H
     CALL BAGET8
     MOV A, AL
     MOV AH, 09H
                  ; ASK FOR 2ND NUMBER
     LEA DX, MSG2
     INT 21H
     CALL BAGET8
     MOV B, AL
```





```
MOV AL, A ; TAKE 1ST NUMBER INTO A ADD AL, B ; ADD 2ND NUMBER TO A JNC SKIP ; IF NO CARRY THEN SKIP INC CARRY ; ELSE INCREMENT CARRY SKIP: MOV SUM, AL ; STORE SUM
      MOV AH, 09H
                         ; DISPLAY SUM
      LEA DX, MSG3
       INT 21H
      LEA SI, SUM
      CALL BAPUT8
                       ; DISPLAY CARRY
      MOV AH, 09H
       LEA DX, MSG4
       INT 21H
       LEA SI, CARRY
      CALL BAPUT8
      MOV AH, 4CH ; END THE PROGRAM... GO BACK TO DOS
       INT 21H
PROC BAGET8
                         ; GETS AN 8 BIT NUMBER FROM THE SCREEN
      PUSH CX
      MOV AH, 01H
       INT 21H
       SUB AL, 30H
       CMP AL, 09H
       JLE G1
      SUB AL, 07H
G1: MOV CL, 04H
      ROL AL, CL
      MOV CH, AL
      MOV AH, 01H
       INT 21H
       SUB AL, 30H
       CMP AL, 09H
       JLE G2
      SUB AL, 07H
    ADD AL, CH ; RETURNS THE NUMBER IN AL
G2:
      POP CX
      RET
ENDP BAGET8
```



Simulation Programs for Lab Practicals Part 1

PROC BAPUT8 ; DISPLAYS 8 BIT NUMBER ON THE SCREEN PUSH CX MOV AL, [SI] AND AL, OFOH MOV CL, 04H ROL AL, CL ADD AL, 30H CMP AL, 39H JLE P1 ADD AL, 07H MOV AH, 02H P1: MOV DL, AL INT 21H MOV AL, [SI] AND AL, OFH ADD AL, 30H CMP AL, 39H JLE P2 ADD AL, 07H MOV AH, 02H P2: MOV DL, AL INT 21H POP CX RET ENDP BAPUT8 CODE ENDS END START

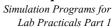
## Notes from Bharat Acharya

- 1) Copy and paste the above program in EMU8086 simulator as a new file
- 2) Click on "save", to save the file with the name "Add8UI"
- 3) Run the program
- 4) Test for various inputs
- 5) This is a complete program and can be used for your practicals
- 6) Modify variable names and message texts as it suits you.



## 3) Add two 16-bit numbers with user input and output

```
DATA SEGMENT
     MSGH1 DB ODH, OAH, "BHARAT ACHARYA EDUCATION $"
MSGH2 DB ODH, OAH, "8086 PRACTICALS $"
MSGH3 DB ODH, OAH, "------ $"
                ODH, OAH, "ADD TWO 16-BIT NUMBERS $"
ODH, OAH, " $"
      MSGH4 DB
      MSGH5 DB
      MSG1 DB ODH, OAH, "PLEASE ENTER THE 1ST NUMBER: $"
      MSG2 DB 0DH, 0AH, "PLEASE ENTER THE 2ND NUMBER: $"
      MSG3 DB 0DH, 0AH, 0DH, 0AH, "SUM: $"
      MSG4 DB ODH, OAH, "CARRY: $"
      A DW
B DW
                  ?
                 ?
      SUM DW ?
      CARRY DB 00H
DATA ENDS
CODE SEGMENT
     ASSUME CS:CODE, DS:DATA
START:
      MOV AX, DATA ; INITIALISE DS
      MOV DS, AX
      MOV AH, 09H ; DISPLAY HEADERS
      LEA DX, MSGH1
      INT 21H
      LEA DX, MSGH2
      INT 21H
      LEA DX, MSGH3
      INT 21H
      LEA DX, MSGH4
      INT 21H
      LEA DX, MSGH5
      INT 21H
      MOV AH, 09H
                     ; ASK FOR 1ST NUMBER
      LEA DX, MSG1
      INT 21H
      LEA SI, A
      CALL BAGET8
      MOV [SI+1], AL
      CALL BAGET8
      MOV [SI], AL
```





```
; ASK FOR 2ND NUMBER
      MOV AH, 09H
      LEA DX, MSG2
      INT 21H
      LEA SI, B
      CALL BAGET8
      MOV [SI+1], AL
      CALL BAGET8
      MOV [SI], AL
                       ; TAKE 1ST NUMBER INTO A
      ADD AX, B
JNC SKIP
      MOV AX, A
                    ; ADD 2ND NUMBER TO A
; IF NO CARRY THEN SKIP
; ELSE INCREMENT CARRY
; STORE SUM
      INC CARRY
SKIP: MOV SUM, AX
      MOV AH, 09H
                       ; DISPLAY SUM
      LEA DX, MSG3
      INT 21H
      LEA SI, SUM
      INC SI
      CALL BAPUT8
      DEC SI
      CALL BAPUT8
                     ; DISPLAY CARRY
      MOV AH, 09H
      LEA DX, MSG4
      INT 21H
      LEA SI, CARRY
      CALL BAPUT8
      MOV AH, 4CH ; END THE PROGRAM... GO BACK TO DOS
      INT 21H
PROC BAGET8
                       ; GETS AN 8 BIT NUMBER FROM THE SCREEN
      PUSH CX
      MOV AH, 01H
      INT 21H
      SUB AL, 30H
      CMP AL, 09H
      JLE G1
      SUB AL, 07H
G1: MOV CL, 04H
      ROL AL, CL
      MOV CH, AL
      MOV AH, 01H
      INT 21H
      SUB AL, 30H
      CMP AL, 09H
      JLE G2
      SUB AL, 07H
     ADD AL, CH
                      ; RETURNS THE NUMBER IN AL
      POP CX
      RET
ENDP BAGET8
```

Simulation Programs for Lab Practicals Part 1



PROC BAPUT8 ; DISPLAYS 8 BIT NUMBER ON THE SCREEN PUSH CX MOV AL, [SI] AND AL, OFOH MOV CL, 04H ROL AL, CL ADD AL, 30H CMP AL, 39H JLE P1 ADD AL, 07H MOV AH, 02H P1: MOV DL, AL INT 21H MOV AL, [SI] AND AL, OFH ADD AL, 30H CMP AL, 39H JLE P2 ADD AL, 07H P2: MOV AH, 02H MOV DL, AL INT 21H POP CX RET ENDP BAPUT8 CODE ENDS END START

## Notes from Bharat Acharya

- 1) Copy and paste the above program in EMU8086 simulator as a new file
- 2) Click on "save", to save the file with the name "Add16UI"
- 3) Run the program
- 4) Test for various inputs
- 5) This is a complete program and can be used for your practicals
- 6) Modify variable names and message texts as it suits you.