



**Universidade Federal do Sul e Sudeste do Pará  
Faculdade de Computação e Engenharia Elétrica  
Curso de Engenharia da Computação  
Iago Costa das Flores**

**Microprocessadores e Microcontroladores  
Experimento 5 – Programação Assembly no 8086/8088**

**Marabá  
2021**

**Microprocessadores e Microcontroladores**  
**Experimento 5 – Programação Assembly no 8086/8088**

Relatório apresentado no curso de Engenharia da Computação, turma de 2018 como obtenção de nota parcial na disciplina de microprocessadores e microcontroladores, ministrada pelo Professor Dr. Elton Alves.



## **Sumário**

<b>1 - Introdução</b>	<b>4</b>
<b>2 - Atividades</b>	<b>4</b>
2.1 - Desenvolva um programa que solicite a entrada de dois valores numéricos decimais positivos de um dígito, subtraia os valores e apresente o resultado da operação como sendo um valor decimal. Para isso, desenvolva um programa .EXE.	4
<b>3 - Conclusão</b>	<b>8</b>
<b>4 - Referências</b>	<b>8</b>

# 1 - Introdução

O Trabalho visa apresentar os códigos fontes e resultados de execução das três atividades avaliativas a seguir:

1- Desenvolva um programa que solicite a entrada de dois valores numéricos decimais positivos de um dígito, subtraia os valores e apresente o resultado da operação como sendo um valor decimal. Para isso, desenvolva um programa .EXE.

## 2 - Atividades

As atividades demonstradas a seguir foram feitas com a ajuda do programa emu8086 para escrever e executar os códigos em assembly.

2.1 - Desenvolva um programa que solicite a entrada de dois valores numéricos decimais positivos de um dígito, subtraia os valores e apresente o resultado da operação como sendo um valor decimal. Para isso, desenvolva um programa .EXE.

### **Código da atividade 01:**

TITLE SUBTRACAO DE VALORES

#MAKE\_EXE# ; tipo de arquivo para ser gerado

DADOS SEGMENT 'DATA' ; define segmento de dados

m1 DB 'Valor 1: ', 24h

m2 DB 0Dh, 0Ah, 'Valor 2: ', 24h

m3 DB 0Dh, 0Ah, 'Result: ', 24h

msg4 DB 0Dh, 0Ah, 'Valor invalido', 24h

DADOS ENDS

PILHA SEGMENT STACK 'STACK' ; define segmento de pilha

DW 0100h DUP(?) ; define tamanho da pilha

PILHA ENDS

CODIGO SEGMENT 'CODE' ; define segmento de código

ASSUME CS:CODIGO, DS:DADOS, SS:PILHA ; declara cada segmento aos  
;registradores

fim MACRO ; define macro de finalizacao



MOV AH, 4ch ;encerramento do programa  
INT 21h ;Controle do SO  
ENDM ; finaliza macro fim

msg MACRO ; define macro para imprimir mensagem na tela  
MOV AH, 09h  
INT 21h ; apresentacao da mensagem  
ENDM ; finaliza macro msg

INICIO\_SCRIPT PROC FAR ; inicia procedimento INICIO do tipo FAR  
MOV AX, DADOS ; acesso do segmento de codigo ao segmento de dados  
MOV DS, AX ; movendo acesso para ds  
MOV ES, AX ;

MOV DX, OFFSET m1  
MSG ; chama macro que imprime caractere na tela  
CALL input ; chama rotina input para receber caractere  
MOV BH, AL

MOV DX, OFFSET m2  
MSG ; chama macro que imprime caractere na tela  
CALL input ; chama rotina input para receber caractere  
MOV BL, AL

MOV DX, OFFSET m3  
MSG ; chama macro que imprime caractere na tela

SUB BH, BL ; subtrai BH-BL para BH  
JGE valor\_positivo ; se BH>=BL pula para rotina valor\_positivo  
JL valor\_negativo ; se BH<BL pula para rotina valor\_negativo

valor\_positivo: ; rotina valor positivo  
CALL resultado ; chama rotina para resultado da subtracao

valor\_negativo:  
NEG BH ; transforma o valor negativo (complemento de dois) em seu equivalente positivo  
MOV AL, 2Dh ; movimenta o valor 2Dh para o AL (ASCII 2Dh = -)  
MOV AH, 0Eh ; chama comando da interrupcao  
INT 10h ; mostra na tela o caractere negativo  
CALL resultado ; chama rotina para resultado da subtracao

resultado:  
MOV AL, BH ; movimenta resultado de BH para AL  
MOV DL, AL ; movimenta resultado de AL para DL  
ADD AL, 30h ; adiciona o 0 decimal em ascii ao caractere  
MOV AH, 0Eh ; comando para interrupcao  
INT 10h ; mostra caractere na tela  
FIM ; finaliza

RET ; retorna da chamada de rotina

FIM ; finaliza o script

INICIO\_SCRIPT ENDP ; finaliza procedimento inicio\_script

input PROC NEAR ; inicio procedimento input

MOV AH, 01h ; entrada do caracter pelo teclado

INT 21h

CMP AL, 30h ; compara o valor recebido ao o decimal em ascll

JL erro ; salta se caracteres recebido menor que zero --> 30h

CMP AL, 40h ; compara valor recebido ao decimal em ascll

JGE erro ; salta se caractere recebido maior ou igual a 40h

JMP fim\_validacao ; salta para rotina fim\_validacao

erro: ; rotina erro

MOV DX, OFFSET msg4 ; obtem endereco da variavel msg4

MSG ; chama macro para imprimir mensagem

FIM ; finaliza programa com macro

fim\_validacao: ; rotina fim\_validacao

SUB AL, 30h ; subtrai o valor zero decimal do caractere

RET ; retorna da rotina

input ENDP ; fim do procedimento input

CODIGO ENDS ; finaliza o segmento de codigo

END INICIO\_SCRIPT

Na figura 01 temos o código comentado do algoritmo de soma. Foram carregados todos os valores de mensagem e resultado final no terminal. O código foi escrito levando em consideração o padrão para gerar um arquivo executável.

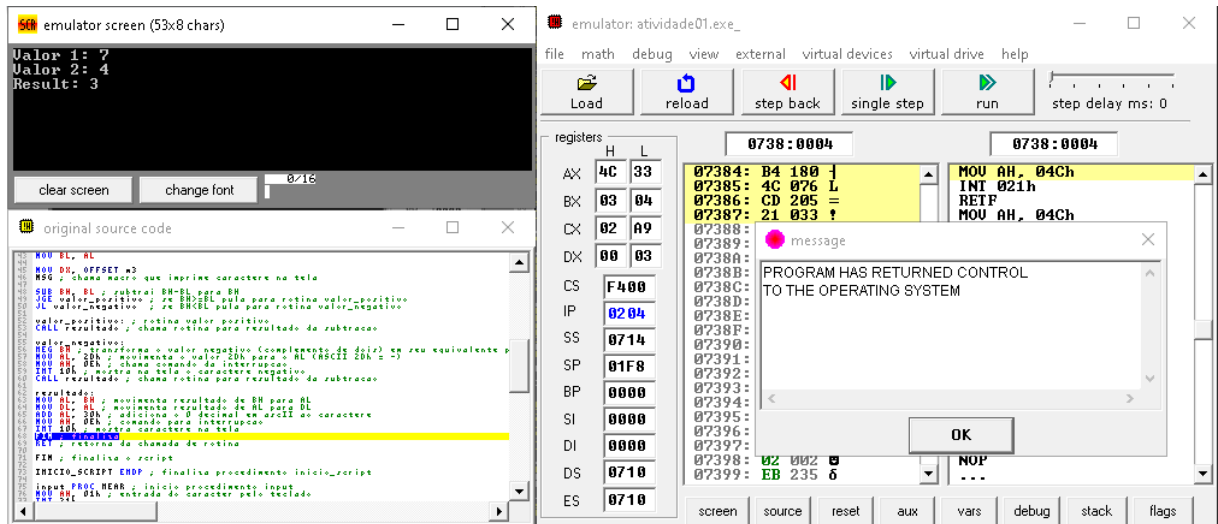
```

01 TITLE SUBTRACAO DE VALORES
02
03 MAKE_EXEM ; tipo de arquivo para ser gerado
04
05 DADOS SEGMENT 'DATA' ; define segmento de dados
06 w1 DB 'Valor 1:', 24h
07 w2 DB 00h, 0Ah, 'Valor 2:', 24h
08 w3 DB 00h, 0Ah, 'Result: ', 24h
09 w4 DB 00h, 0Ah, 'Valor invalido', 24h
10 DADOS ENDS
11
12 PILHA SEGMENT STACK 'STACK' ; define segmento de pilha
13 DW 0100h DUP(?) ; define tamanho da pilha
14 PILHA ENDS
15
16 CODIGO SEGMENT 'CODE' ; define segmento de codigo
17 ASSUME CS:CODIGO, DS:DADOS, SS:PILHA ; declara cada segmento aos
18 ; registradores
19
20 fim MACRO ; define macro de finalizacao
21 MOV AH, 4ch ; encerramento do programa
22 INT 21h ; Controle do SO
23 ENDM ; finaliza macro fim
24
25 msg MACRO ; define macro para imprimir mensagem na tela
26 MOV AH, 09h
27 INT 21h ; apresentacao da mensagem
28 ENDM ; finaliza macro msg
29
30 INICIO_SCRIPT PROC FAR ; inicia procedimento INICIO do tipo FAR
31 MOV AX, DADOS ; acesso do segmento de dados ao segmento de dados
32 MOV DS, AX ; movendo acesso para ds
33 MOV ES, AX ;
34
35 MOV DX, OFFSET w1
36 MSG ; chama macro que imprime caractere na tela
37 CALL input ; chama rotina input para receber caractere
38 MOV BH, AL
39
40 MOV DX, OFFSET w2
41 MSG ; chama macro que imprime caractere na tela
42 CALL input ; chama rotina input para receber caractere
43 MOV BL, AL
44
45 MOV DX, OFFSET w3
46 MSG ; chama macro que imprime caractere na tela
47
48 SUB BH, BL ; subtrai BH-BL para BH
49 JGE valor_positivo ; se BH>=BL pula para rotina valor_positivo
50 JL valor_negativo ; se BH<BL pula para rotina valor_negativo
51
52 valor_positivo: ; rotina valor positivo
53 CALL resultado ; chama rotina para resultado da subtracao
54
55 valor_negativo:
56 NEG BH ; transforma o valor negativo (complemento de dois) em seu equivalente positivo
57 MOV AL, 20h ; movimenta o valor 20h para o AL (ASCII 20h = -)
58 MOV AH, 0Eh ; chama comando de interrupcao
59 INT 10h ; mostra na tela o caractere negativo
60 CALL resultado ; chama rotina para resultado da subtracao
61
62 resultado:
63 MOV AL, BH ; movimenta resultado de BH para AL
64 MOV DL, AL ; movimenta resultado de AL para DL
65 ADD AL, 30h ; adiciona o 0 decimal em ascii ao caractere
66 MOV AH, 0Eh ; comando para interrupcao
67 INT 10h ; mostra caractere na tela
68 FIN ; finaliza
69 RET ; retorna da chamada de rotina
70
71 FIN ; finaliza o script
72
73 INICIO_SCRIPT ENDP ; finaliza procedimento inicio_script
74
75 input PROC NEAR ; inicio procedimento input
76 MOV BH, 01h ; entrada do caracter pelo teclado
77 INT 21h
78 CMP AL, 30h ; compara o valor recebido ao 0 decimal em ascii
79 JL erro ; salta se caracteres recebido menor que zero --> 30h
80 CMP AL, 40h ; compara valor recebido ao decimal em ascii
81 JGE erro ; salta se caractere recebido maior ou igual a 40h
82 JMP fim_validacao ; salta para rotina fim_validacao
83
84 erro: ; rotina erro
85 MOV DX, OFFSET msg4 ; obtem endereco da variavel msg4
86 MSG ; chama macro para imprimir mensagem
87 FIN ; finaliza programa com macro
88
89 fim_validacao: ; rotina fim_validacao
90 SUB AL, 30h ; subtrai o valor zero decimal do caractere
91 RET ; retorna da rotina
92 input ENDP ; fim do procedimento input
93
94 CODIGO ENDS ; finaliza o segmento de codigo
95
96 END INICIO_SCRIPT

```

**Figura 01:** Código fonte da atividade 01

Na figura 02, temos a execução da atividade 01, onde mostramos o recebimento dos valores decimais positivos, e logo após o resultado da subtração em decimal tudo no terminal.



**Figura 02:** Execução da atividade 01

### 3 - Conclusão

Foram demonstradas as impressões do código e execução dos mesmos através das imagens apresentadas com suas devidas explicações. Os códigos fontes foram todos comentados e a atividade foi feita conforme o solicitado no comando do trabalho.

### 4 - Referências

MANZANO, J. A. **Programação assembly: padrão IBM - PC 8086/8088**. 6ªed. Ed.Erica, 2012.