



**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 4 – Programação Assembly no 8086/8088**

**Marabá
2021**

Microprocessadores e Microcontroladores
Experimento 4 – 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

1 - Introdução	4
2 - Atividades	4
2.1 - Desenvolva um programa que solicite a entrada de dois valores numéricos decimais positivos de um dígito, some os valores e apresente o resultado da operação como sendo um valor decimal. Este programa deve fazer uso da Instrução OR..	4
2.2 - Desenvolva um programa que exiba a contagem 9x de valores numéricos .	7
3 - Conclusão	9
4 - Referências	9

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, some os valores e apresente o resultado da operação como sendo um valor decimal. Este programa deve fazer uso da Instrução OR.
2. Desenvolva um programa que exiba a contagem 9x de valores numéricos .

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, some os valores e apresente o resultado da operação como sendo um valor decimal. Este programa deve fazer uso da Instrução OR..

Código da atividade 01:

org 100h

.DATA

msg1 DB 'Entre um valor numerico positivo (de 0 ate 9): ', 24h ; 24h caracter

;de fim de string \$

msg2 DB 0Dh, 0Ah, 'Entre um valor numerico positivo 2 (de 0 ate 9): ', 24h ;0Dh tecla <ENTER>

msg3 DB 0Dh, 0Ah, 'Valor soma: ', 24h ; 0Ah line feed

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

.CODE

JMP entrada1

JMP entrada2

JMP imprimir

entrada1:

LEA DX, msg1

CALL escreva ; chamando procedimento escreva para imprimir mensagem no terminal



MOV AH, 01h ; entrada do caracter pelo teclado
INT 21h ; controle da acao
CMP AL, 30h ; desvio de deteccao de erro, caso o valor < 30h(0d)
JL erro ; desvio para rotina erro
CMP AL, 39h ; desvio de deteccao de erro, caso o valor >=39h (9h)
JG erro ; desvia para rotina erro
SUB AL, 30h ; efetua a subtracao do valor 30h para deixar apenas no
OR BL, AL ; registrador AL o valor numerico

entrada2:

LEA DX, msg2
CALL escreva
MOV AH, 01h ; entrada do caracter pelo teclado
INT 21h ; controle da acao
CMP AL, 30h ; desvio de deteccao de erro, caso o valor < 30h(0d)
JL erro ; desvio para rotina erro
CMP AL, 39h ; desvio de deteccao de erro, caso o valor >=39h (9h)
JG erro ; desvia para rotina erro
SUB AL, 30h ; efetua a subtracao do valor 30h para deixar apenas no

ADD BL, AL ; somando AL + BL em BL

imprimir:

LEA DX, msg3 ; copiando endereco de msg para dx
CALL escreva ; chamando procedimento escreva para imprimir mensagem no terminal
MOV AH, 02h ; preparando interrupcao para imprimir caractere em dl
MOV DL, BL ; movendo soma de BL para DL
ADD DL, 30h ; asicionando 30h para mostrar o caractere em ASCII
CALL resultado ; chamando procedimento para imprimir resultado da soma
JMP saida ; chamando rotina de finalizacao

erro:

LEA DX, msg4 ; mensagem armazenada na variavel msg4
CALL escreva ; chamando procedimento escreva para imprimir mensagem no terminal
JMP saida ; chamando procedimento escreva para imprimir mensagem no terminal

saida:

INT 20h

escreva PROC NEAR

MOV AH, 09h ; preparando interrupcao para imprimir mensagem
INT 21h ; interrupcao que imprimi mensagem na tela
RET
escreva ENDP

resultado PROC NEAR

MOV AH, 02h ; preparando interrupcao para imprimir caractere
INT 21h ; interrupcao que imprimi caractere na tela

RET

resultado ENDP

Na figura 01 temos o código comentado do algoritmo de soma. Estão separados o .DATA e o .CODE. Foram carregados todos os valores de mensagem e resultado final no terminal, além disso na linha 26 é possível visualizar o uso do comando OR para transferir o valor de AL para BL para receber o 2 valor e somar em BL no final.

```

001  brg 100h
002
003  .DATA
004  msg1 DB 'Entre um valor numerico positivo (de 0 ate 9): ', 24h ; 24h caracter
005  ;de fim de string $
006  msg2 DB 00h, 0Ah, 'Entre um valor numerico positivo 2 (de 0 ate 9): ', 24h ;00h tecla <ENTER>
007  msg3 DB 00h, 0Ah, 'Valor soma: ', 24h ; 0Ah line feed
008  msg4 DB 00h, 0Ah, 'Caractere invalido', 24h
009
010  .CODE
011
012  JMP entrada1
013  JMP entrada2
014  JMP imprimir
015
016  entrada1:
017  LEA DX, msg2
018  CALL escreva ; chamando procedimento escreva para imprimir mensagem no terminal
019  MOV AH, 01h ; entrada do caracter pelo teclado
020  INT 21h ; controle da acao
021  CMP AL, 30h ; desvio de deteccao de erro, caso o valor < 30h(0d)
022  JL erro ; desvio para rotina erro
023  CMP AL, 39h ; desvio de deteccao de erro, caso o valor >=39h (9h)
024  JG erro ; desvia para rotina erro
025  SUB AL, 30h ; efetua a subtracao do valor 30h para deixar apenas no
026  OR BL, AL ; registrador AL o valor numerico
027
028  entrada2:
029  LEA DX, msg2
030  CALL escreva
031  MOV AH, 01h ; entrada do caracter pelo teclado
032  INT 21h ; controle da acao
033  CMP AL, 30h ; desvio de deteccao de erro, caso o valor < 30h(0d)
034  JL erro ; desvio para rotina erro
035  CMP AL, 39h ; desvio de deteccao de erro, caso o valor >=39h (9h)
036  JG erro ; desvia para rotina erro
037  SUB AL, 30h ; efetua a subtracao do valor 30h para deixar apenas no
038  ADD BL, AL ; somando AL + BL em BL
039
040  imprimir:
041  LEA DX, msg3 ; copiando endereco de msg para dx
042  CALL escreva ; chamando procedimento escreva para imprimir mensagem no terminal
043  MOV AH, 02h ; preparando interrupcao para imprimir caractere em dl
044  MOV DL, BL ; movendo soma de BL para DL
045  ADD DL, 30h ; adicionando 30h para mostrar o caractere em ASCII
046  CALL resultado ; chamando procedimento para imprimir resultado da soma
047  JMP saida ; chamando rotina de finalizacao
048
049  erro:
050  LEA DX, msg4 ; mensagem armazenada na variavel msg4
051  CALL escreva ; chamando procedimento escreva para imprimir mensagem no terminal
052  JMP saida ; chamando procedimento escreva para imprimir mensagem no terminal
053
054  saida:
055  INT 20h
056
057  escreva PROC NEAR
058  MOV AH, 09h ; preparando interrupcao para imprimir mensagem
059  INT 21h ; interrupcao que imprime mensagem na tela
060  RET
061  escreva ENDP
062
063  resultado PROC NEAR
064  MOV AH, 02h ; preparando interrupcao para imprimir caractere
065  INT 21h ; interrupcao que imprime caractere na tela
066  RET
067  resultado ENDP

```

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 soma em decimal tudo no terminal.

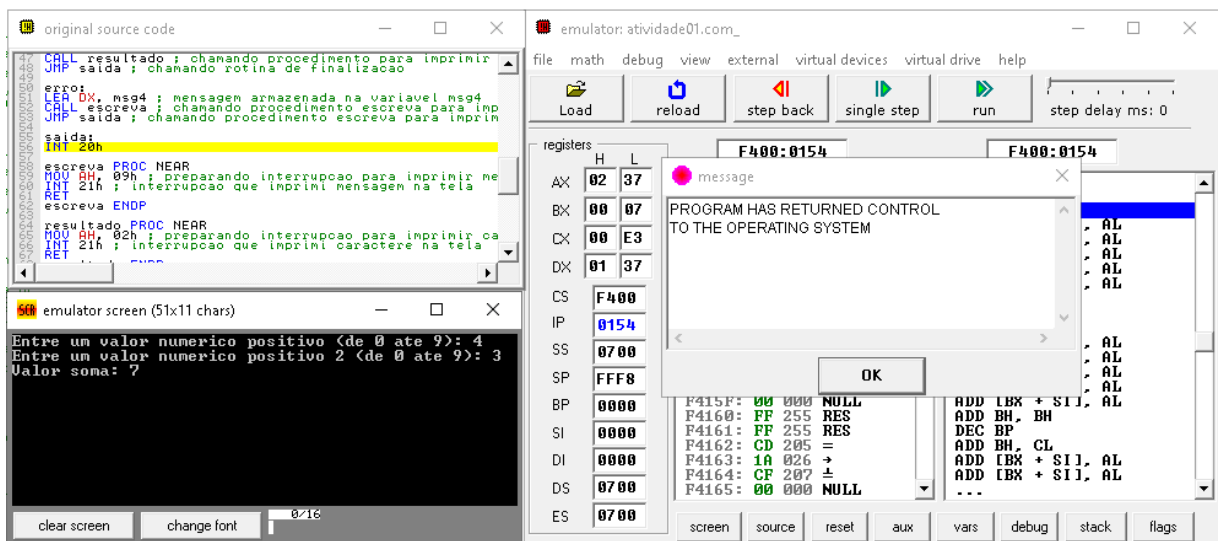


Figura 02: Execução da atividade 01

2.2 - Desenvolva um programa que exiba a contagem 9x de valores numéricos .

Código fonte atividade 02:

```
org 100h
```

```
.DATA
```

```
msg DB 13d, 12o, 24h ; 13d==0Dh , 12o==0Ah e 24h
;(controle da String)
```

```
.CODE
```

```
MOV BL, 30h ; adiciona 0 em ascii para BL
```

```
MOV CX, 9d ; estabelece em CX o valor decimal (contagem do laço de repeticao)
```

```
laço:
```

```
    CALL contador ; chama procedimento contador
```

```
    CALL escreva ; chama procedimento escreva
```

```
    LOOP laço ; CX sera decrementado em 1
```

```
INT 20h ; finaliza o script
```

```
escreva PROC NEAR
```

```
    LEA DX, msg ; msg para quebra de linha no terminal
```

```
    MOV AH, 09h ; preparando interrupcao para imprimir mensagem
```

```
    INT 21h ; interrupcao que imprime mensagem na tela
```

```
    RET
```

```
escreva ENDP
```

contador PROC NEAR
MOV AH, 02h ; prepara interrupcao para imprimir caractere
INC BL ; incrementa mais um em BL
LEA DL, BL ; copia enredo de BL para DL
INT 21h ; exibicao do caractere no terminal
RET
contador ENDP

Na figura 03 temos o código comentado do algoritmo do contador 9x de números decimais.

```
01 org 100h
02
03 .DATA
04 msg DB 13d, 12o, 24h ; 13d==0Dh , 12o==0Ah e 24h
05 ;<controle da String>
06
07 .CODE
08 MOV BL, 30h ; adiciona 0 em ascii para BL
09 MOV CX, 9d ; estabelece em CX o valor decimal <contagem do laco de repeticao>
10
11 laco:
12     CALL contador ; chama procedimento contador
13     CALL escreva ; chama procedimento escreva
14     LOOP laco ; CX sera decrementado em 1
15 INT 20h ; finaliza o script|
16
17
18 escreva PROC NEAR
19 LEA DX, msg ; msg para quebra de linha no terminal
20 MOV AH, 09h ; preparando interrupcao para imprimir mensagem
21 INT 21h ; interrupcao que imprime mensagem na tela
22 RET
23 escreva ENDP
24
25 contador PROC NEAR
26 MOV AH, 02h ; prepara interrupcao para imprimir caractere
27 INC BL ; incrementa mais um em BL
28 LEA DL, BL ; copia enredo de BL para DL
29 INT 21h ; exibicao do caractere no terminal
30 RET
31 contador ENDP
```

Figura 03: Código fonte da atividade 02



Na figura 04, o resultado do loop com o contador. Foi contado em BL com a ajuda do loop de 1 até 9 e impresso no terminal em DL para visualização.

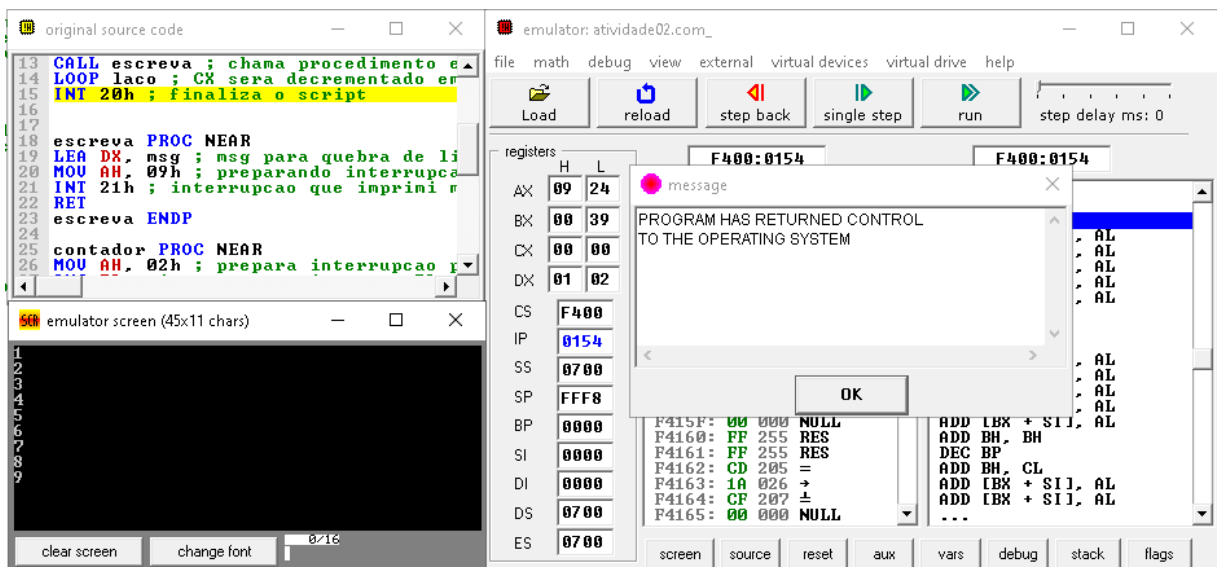


Figura 04: Execução da atividade 02

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 as 2 atividades foram feitas conforme o solicitado no comando do trabalho.

4 - Referências

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