



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

**Marabá
2021**

Microprocessadores e Microcontroladores
Experimento 1 – 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 em Assembly que imprima seu nome.	4
2.2 - Escreva um programa para exibir um '?'. 2.3 - Escreva um programa que leia um caractere do teclado digitado pelo usuário e salvando-o em um registrador.	6
	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 em Assembly que imprima seu nome.
2. Escreva um programa para exibir um '?'.
3. Escreva um programa que ler um caractere do teclado digitado pelo usuário e salvando-o em um registrador:

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 em Assembly que imprima seu nome.

Na figura 1 temos a imagem do código fonte com todas as linhas comentadas com explicações de comandos usados. O procedimento padrão inicializa, executa uma interrupção, carrega uma mensagem no registrador de dados e executa a interrupção de exibição no terminal. A mensagem com o nome é carregada a partir do seu código ascII.

```
01 org 100h; inicializando o script na memoria
02 mov ah, 09h ; executando interrupcao de saida
03 lea dx,mensagem; salvando copia da referencia no registrador de dados
04 int 21h ; exibindo acumulador dx do tipo dados no terminal
05 int 20h ; finalizando o script
06 mensagem db 49h, 61h, 67h, 6fh, 24h ;escrevendo iago pela tabela ascII
07
08 ;mensagem db 'iago', 24h;
09
```

Figura 1: Código fonte da atividade 01.

Na figura 2 temos algumas informações sobre como estão sendo armazenados os dados nos registradores do microprocessador. Em Ax foi carregada a interrupção de saída e em Dx foi carregada a mensagem para exibição.

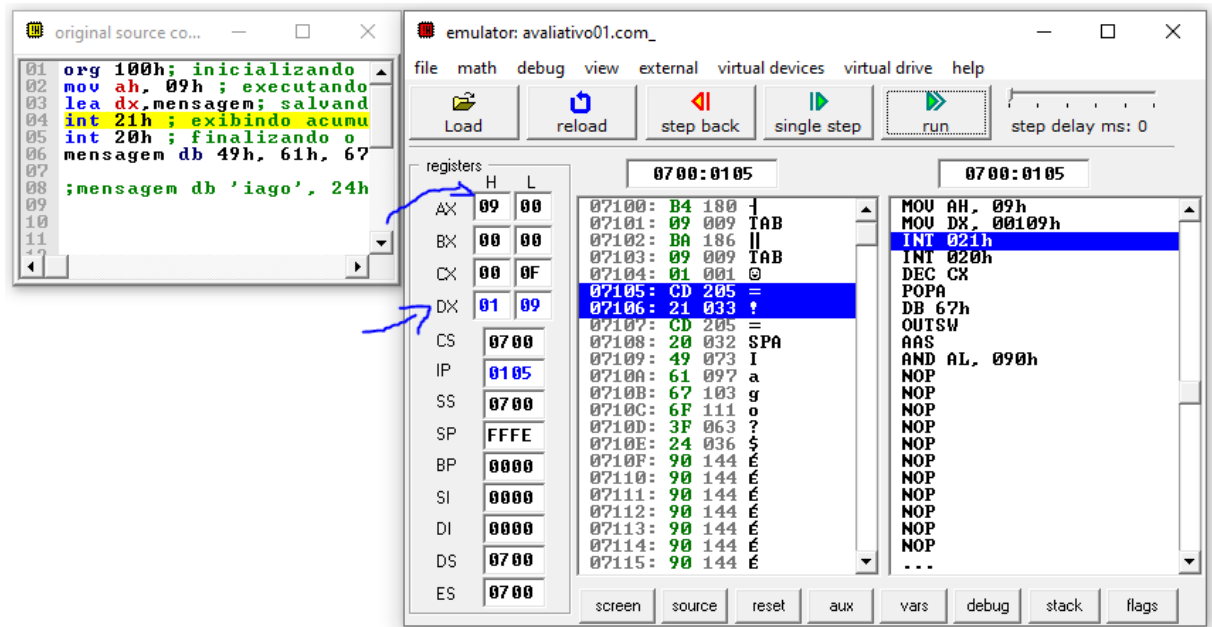


Figura 2: Carregando dados nos registradores.

Na figura 3 temos a demonstração da execução da interrupção de abertura do terminal e carregamento dos dados no registrador de dados Dx.

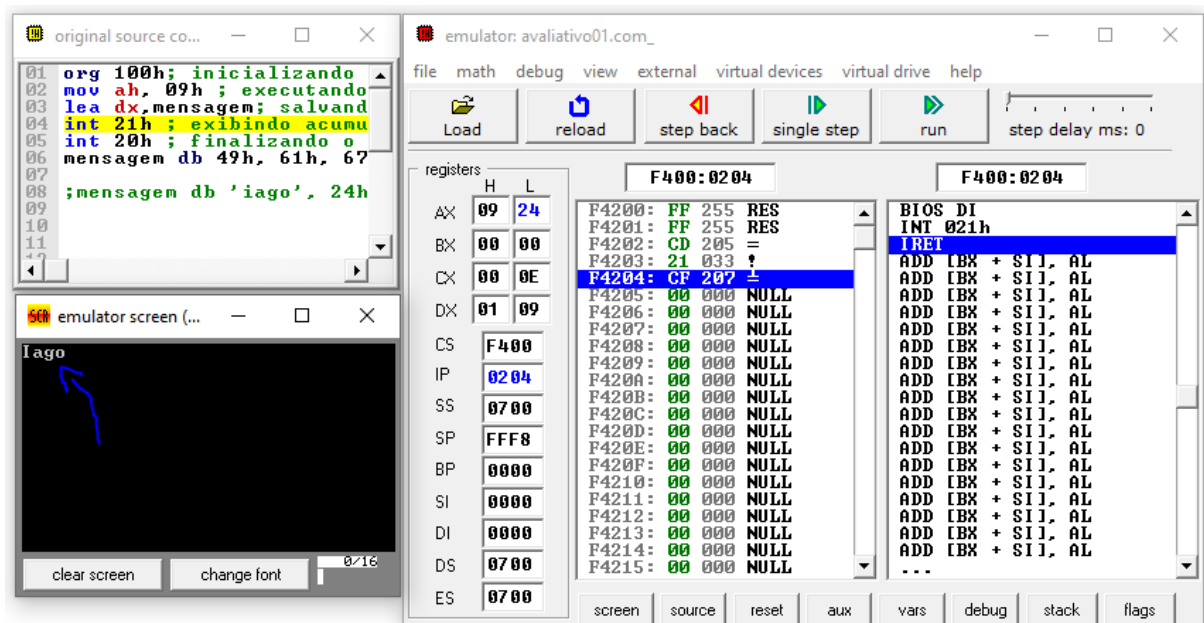


Figura 3: Exibindo nome no terminal.

2.2 - Escreva um programa para exibir um '?'.

Na figura 4 é mostrado a impressão do código fonte da atividade 2. Para realizar-se o comando da atividade é necessário uma interrupção e uma mensagem pré-armazenada que é passada para o registrador de dados, o qual é carregado no terminal de saída.

```

01 org 100h; inicializando o script na memoria
02 mov ah, 09h ; executando interrupcao de saída
03 lea dx, mensagem; salvando copia da referencia no registrador de dados
04 int 21h ; exibindo acumulador dx do tipo dados no terminal
05 int 20h ;finalizando o script
06 mensagem db 3fh, 24h ; escrevendo interrogacao (?) pela tabela ascii
07
08

```

Figura 4: Código fonte da atividade 02.

Na figura 5 comenta-se sobre o registro da interrupção em Ax e a cópia do endereço de memória da mensagem para DI com está indicado pelas setas na imagem.

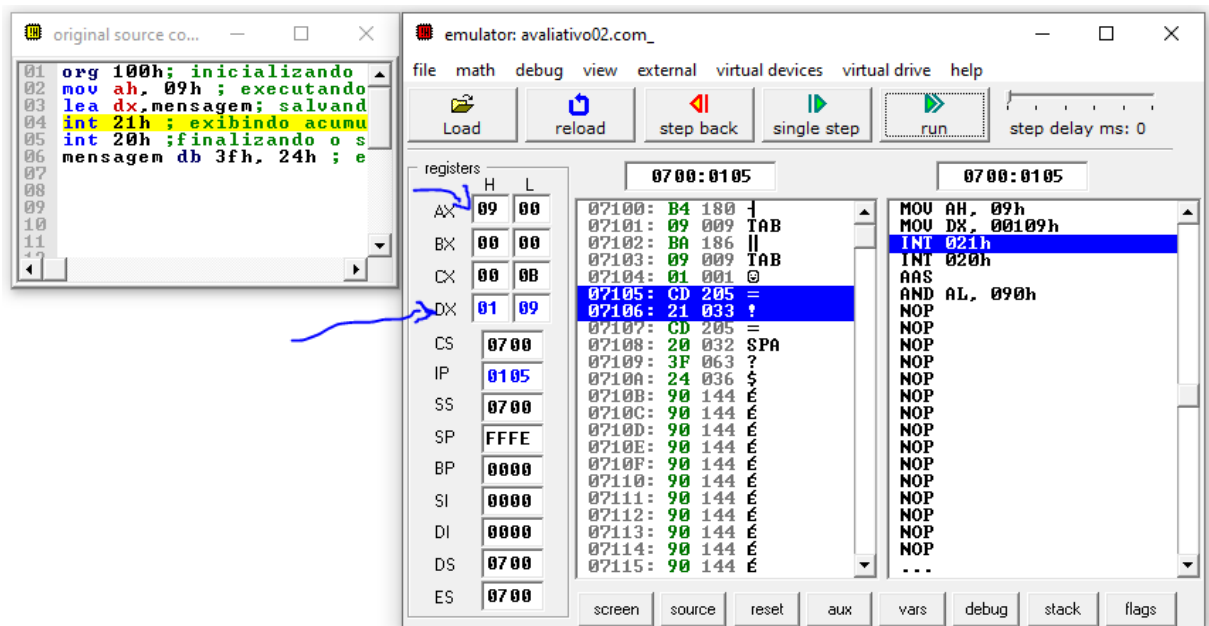


Figura 5: Carregando dados nos registradores.

Na figura 6 tem-se o resultado de exibição final do caractere armazenado. Para isso, como descreve a imagem utilizou-se a na linha 04 do código fonte a interrupção int 21h.

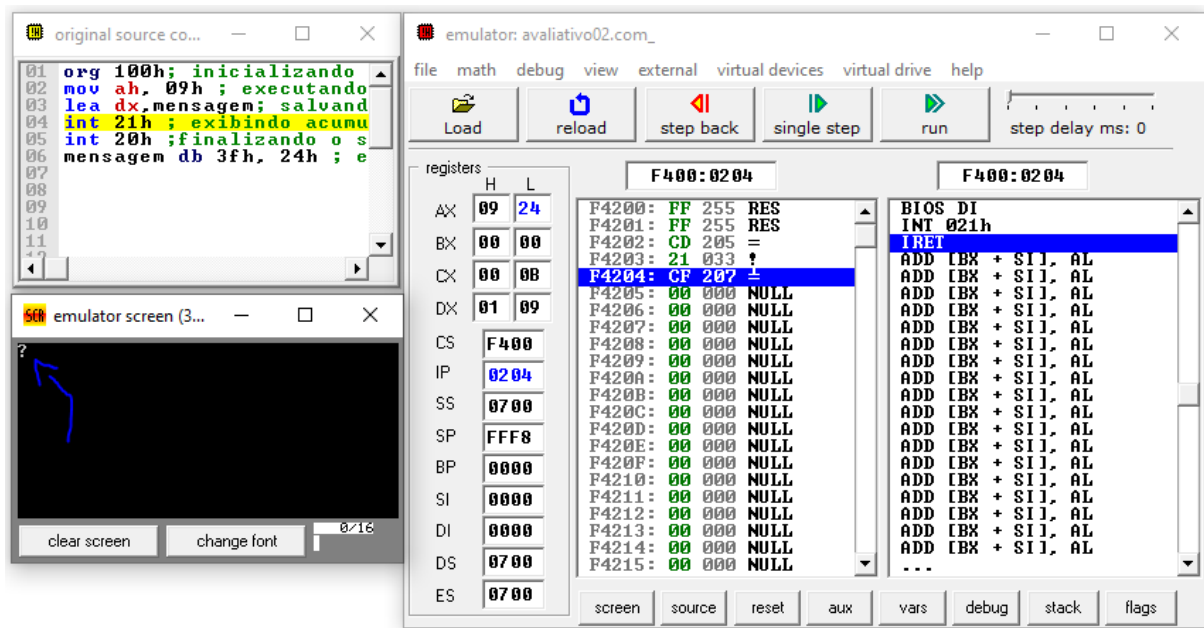


Figura 6: Mostrando caractere interrogação (?) no terminal.

2.3 - Escreva um programa que leia um caractere do teclado digitado pelo usuário e salvando-o em um registrador.

Na figura 7 temos o código fonte da atividade 3. Inicialmente o algoritmo executa uma interrupção de entrada de dados, move a entrada dos dados de um registrador para outro e executa a interrupção final para exibir no terminal de saída.

```

01 org 100h ; inicializa o script na memoria
02
03
04 int 16h ; pega valor do teclado
05 mov bl, al ; movendo do registrador al para o registrador bl
06 mov dl, bl ; passando o codigo ascii do acumulador al para o dados dl
07 mov ah, 2h ; chamando interrupcao para exibir caractere no terminal
08
09 int 21h ; exibindo o terminal
10 int 20h ; finalizando o script
11
12

```

Figura 7: Código fonte da atividade 3.

Na figura 8 demonstramos o momento em que o programa interrompe e espera uma entrada do teclado para continuar a execução, o comando usado é o da linha 4: int 16h.

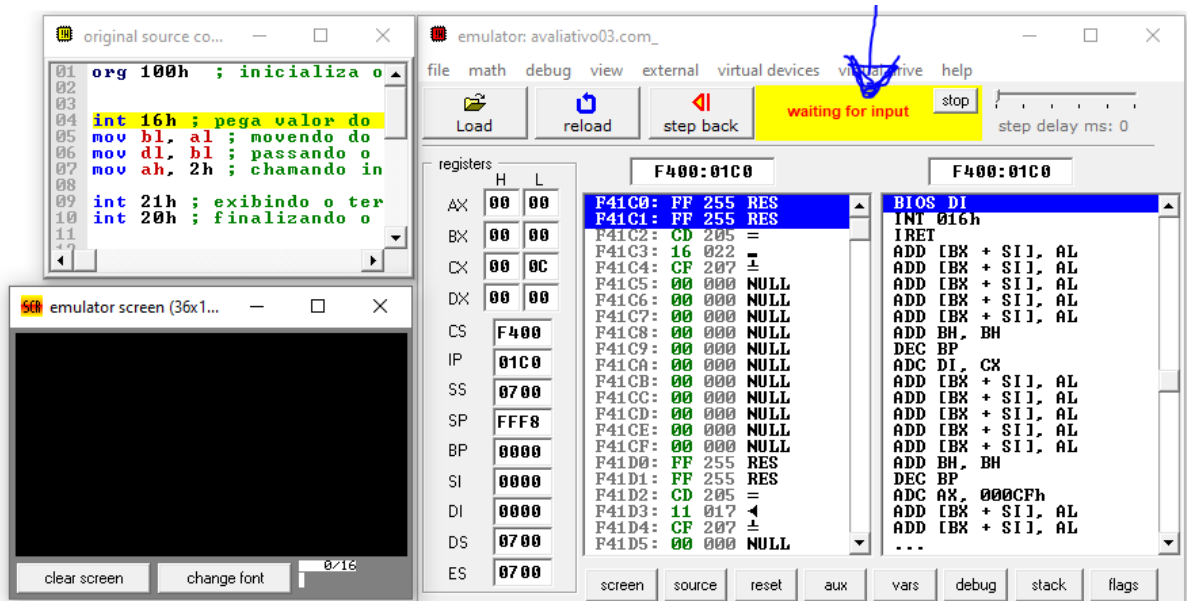


Figura 8: Recebendo input de caractere do teclado.

Na figura 9 temos a realização da transferência de dados de um registrador para outro movendo o caractere em ASCII de Al para Bl.

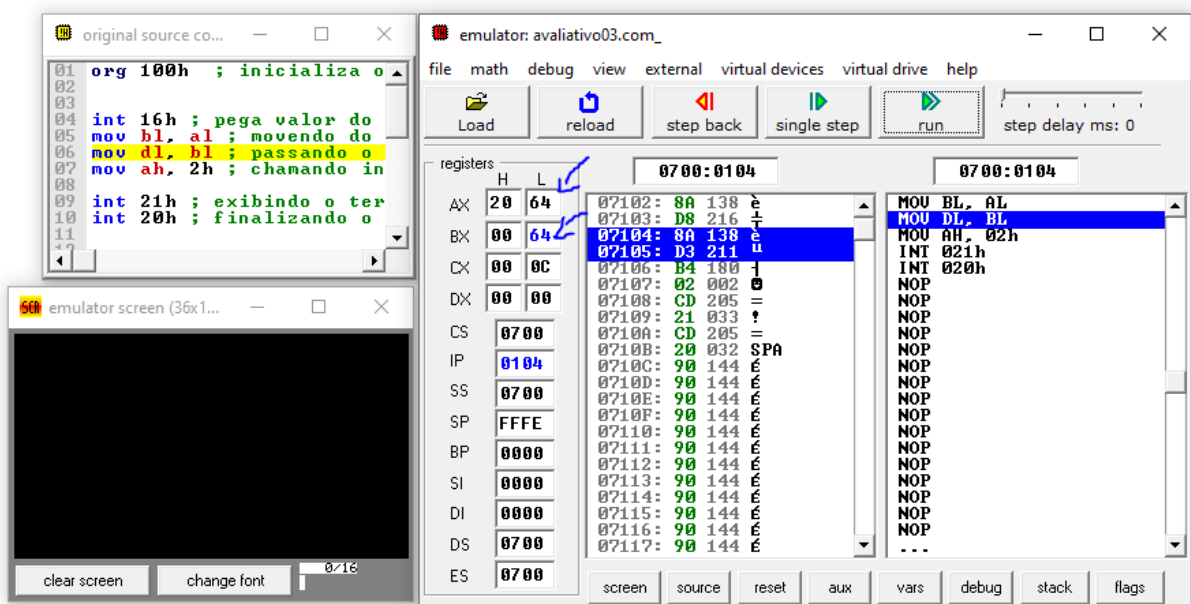


Figura 9: Movendo dados de um registrador para outro.

Na figura 10 temos a saída dos dados para o terminal. O terminal mostra o que está armazenado em Dx, que foi o caractere pego pela interrupção 16h. A interrupção 16h salva primeiramente em Ax mais especificamente em Al e repassa

para BI e posteriormente de BI para DI. De onde foi usado para exibir o caractere no terminal.

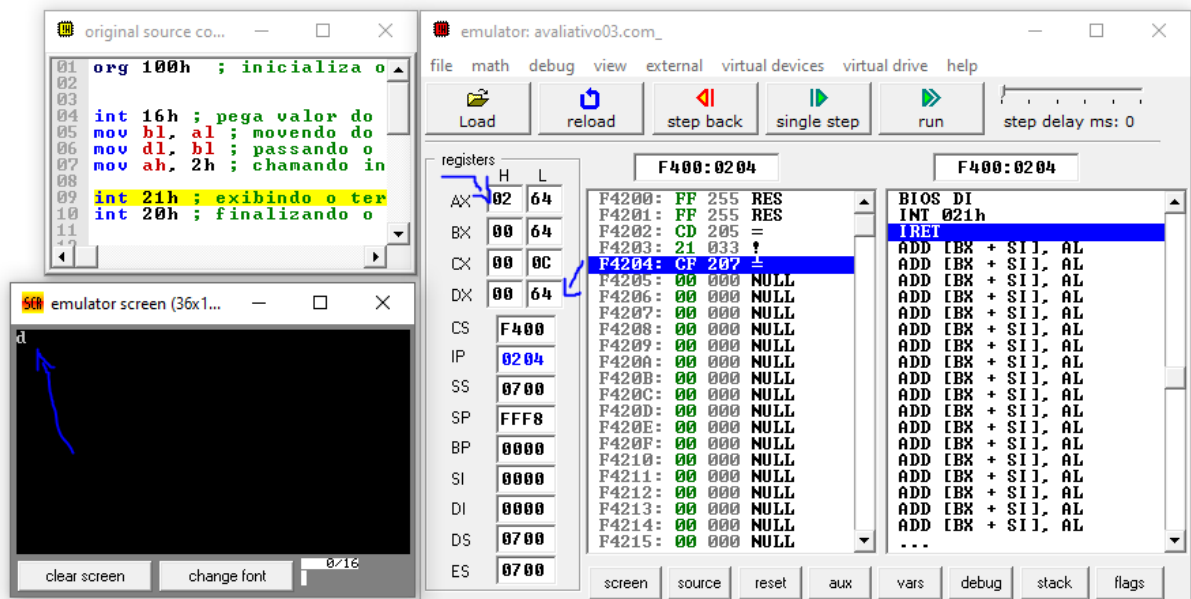


Figura 10: Caractere digitado retornado no terminal.

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 3 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.