



LINGUAGEM DE MONTAGEM (8086) – PARTE III

Objetivos

- Aprimorar/desenvolver programas básicos em linguagem “Assembly / 8086”;
- Realizar simulações de sistemas de controle através de plantas didáticas virtuais;
- Estudar novas instruções do 8086.

Roteiro

1. O programa mostrado na listagem I faz o uso de um aplicativo externo (“thermometer.exe”) que simula virtualmente um conjunto queimador-termômetro. A interface entre o programa em assembler e este aplicativo é feita através dos endereços de I/O 125 (leitura da temperatura) e 127 (saída para acionamento do queimador). 1.Implemente, execute, entenda e comente este programa.
2. Modifique o programa fornecido na listagem I para que o mesmo, em tempo real, exiba na tela de saída o valor da temperatura em graus Celsius e Fahrenheit. Exemplo de formato: “25°F (77°F)”. ■
3. A partir do programa desenvolvido no item 2, inclua um código adicional para receber um valor de temperatura mínima e outro de temperatura máxima, os quais devem ser utilizados para definir os limites do controle de temperatura. ■
4. Modifique o programa da listagem II para configurar “corretamente” os estados (não pode, por exemplo, ocorrer “verdes e/ou amarelos conflitantes”). Em seguida, ajuste “adequadamente” os tempos de amarelo e verde. (por exemplo, 4 segundos para amarelo e 10 segundos para verde). ■
5. O programa mostrado na listagem III exibe o conteúdo dos 10 primeiros bytes da memória de programa no formato hexadecimal. Adicione um código para listar 16 bytes, em 4 linhas, conforme o formato indicado a seguir: ■

```
0100: 41 42 43 44 ==> "ABCD"
0104: 30 31 32 33 ==> "0123"
0108: 33 33 33 33 ==> "3333"
010C:41 42 43 44 ==> "ABCD"
```

Listagens de programas

Listagem I	Listagem II	Listagem III
<pre>#start=thermometer.exe# org 100h start: in AL,125 cmp AL,22 jl low cmp AL,55 jle ok jg high low: mov AL,1 out 127,AL jmp ok high: mov AL,0 out 127,AL ok: jmp start</pre>	<pre>#start=Traffic_Lights.exe# org 100h jmp start semaforo DW 0000_0011_0000_1100b s1 DW 0000_0110_1001_1010b s2 DW 0000_1000_0110_0001b s3 DW 0000_1000_0110_0001b s4 DW 0000_0100_1101_0010b final DW ? start: loop0: MOV SI , OFFSET semaforo loop1: MOV AX, [SI] OUT 04, AX MOV CX, 4Ch MOV DX, 4B40h MOV AH, 86h INT 15h ADD SI, 2 CMP SI, OFFSET final JB loop1 JMP loop0 .EXIT</pre>	<pre>org 100h ini: mov cx,10 lea si,ini cmp al, 9 jg fl mov dl,[si] call hexb inc si loop l1 fim: .EXIT hexd proc near add al, ('A'-10) mov ah, 2 mov dl, al int 21h pop dx ret hexb proc near add al, ('A'-10) mov ah, 2 mov dl, al int 21h pop dx ret</pre>

Questões adicionais

- 1) Estude o programa exemplo do EMU8006 “0_sample_add_huge_numbers.asm”. (Para abrir este código, clique no ícone “examples” na barra superior da tela principal do EMU8086).
- 2) Implemente um semáforo de 4 tempos.