

LISTA DE EXERCÍCIOS 1

Entrega até 03/04/2017

\*\*\*\*\*Apresente o Fluxograma dos programas\*\*\*\*\*

**Objetivos:**

- Operações com memória (RAM interna, RAM externa e área de programa)
- Modos de endereçamento
- MCU8051 - Aprendizado e treinamento do ambiente de desenvolvimento

**Fornecer:**

- a) O programa em Assembly .asm;
- b) O código do programa em Hexadecimal no formato (Endereço --- Conteúdo)-ver Nota no final; No MCU8051, acesse Virtual MCU → Show CodeMemory;
- c) O conteúdo da RAM interna de Dados nas posições envolvidas;
- d) **Fluxograma** dos programas.
- e) **NÃO** enviar arquivos .doc ou similar. **Utilize arquivo PDF.**
- f) **No início do projeto no MCU8051, optar pela arquitetura 8052.**

Utilize <http://iris.sel.eesc.usp.br/sel433/instrucoes8051.pdf> para consultar sobre as instruções do 8051, caso necessário.

**1** - Um sistema utiliza um microcontrolador 8051 para realizar a leitura de três grandezas (resolução de 8 bits cada) pelas portas P0, P1 e P2, e as salva em três posições na área de RAM interna (endereços 02h, 22h e 42h). Utilize a diretiva EQU para definição das posições na área da RAM interna com os nomes (GRAN1, GRAN2 e GRAN3) e realize a soma destas grandezas, considerando o *overflow* (estouro) de 8 bits. Salve o resultado na posição 20h e o estouro na posição 21h, ambos na memória RAM interna. Inicie com P0=27h, P1=99h e P2=B6h.

Antes de rodar o programa insira os dados manualmente nas portas no ambiente do simulador do MCU8051:

**2** - Repita o programa anterior alterando a forma de armazenamento dos valores das grandezas para que seu próprio programa coloque os dados por endereçamento imediato para as posições e não mais pelas portas. Conteúdo das posições de memória:

(02h) = 4Fh

(22h) = D1h

(42h) = 11111010b

Analise brevemente o que ocorre quando a soma ultrapassa o valor máximo que 1 byte comporta (1111 1111b = FFh) para as duas somas.

**3** – Um sistema com o 8051 faz a leitura de três correntes, com dados de 16 bits cada, e os armazena nos endereços: COR1: (02h e 03h), COR2: (10h e 11h) e COR3: (80h e 81h). Utilizando o modo de endereçamento indireto para acesso aos dados, realize a soma desses valores e armazene o resultado de 16 bits no endereço RES: (20h e 21h), e o estouro no endereço 22h .

Conteúdo das posições de memória:

(02h) = 23h

(03h) = F5h

(20h) = 01101110b

(21h) = 10101010b

(80h) = C7

(81h) = 59

**4** - Um equipamento utiliza um microcontrolador 8051 para realizar a leitura de valores de resistores para facilitar a organização destes. Para testar o equipamento, três valores padrão de resistores (1 byte cada) são alocados na memória de programa do microcontrolador (colocar estes dados após a última instrução do programa, no endereço relocável DADOS). Crie um programa que salve o menor valor de

resistor na posição 20h e o maior na posição 30h da RAM externa. Execute o programa para os dois exemplos a seguir.

Exemplo1			Exemplo2		
org	0		org	0	
ini:			ini:		
-		;aqui o seu programa	-		;aqui o seu programa
-		;aqui o seu programa	-		;aqui o seu programa
-		;aqui o seu programa	-		;aqui o seu programa
aqui:	sjmp	aqui	aqui:	sjmp	aqui
DADOS:DB	150, 220, 33		DADOS:DB	220, 47, 180	
End			End		

Para os exemplos,

- Identificar a posição de endereços na memória de Programa onde os valores dos resistores estão armazenados.
- O conteúdo da RAM Externa de Dados nas posições envolvidas, antes e depois da execução do programa.

---

#### Nota

#### Sobre o Formato Intel HEX

Observando-se o arquivo “.hex” criado na compilação, o qual pode ser aberto por qualquer editor de texto, é possível obter o código de programa gerado pelo compilador.

O formato padrão é apresentado a seguir, onde se pode observar que todos os dados estão no formato HEXADECIMAL:

```

:10010000214601360121470136007EFE09D2190140
:100110002146017EB7C20001FF5F16002148011988
:10012000194E79234623965778239EDA3F01B2CAA7
:100130003F0156702B5E712B722B732146013421C7
:00000001FF
  
```

	Start code
	Byte count
	Address
	Record type
	Data
	Checksum

Mais informações em <http://www.keil.com/support/docs/1584/>.