

Licenciatura em Engenharia Electrónica Industrial

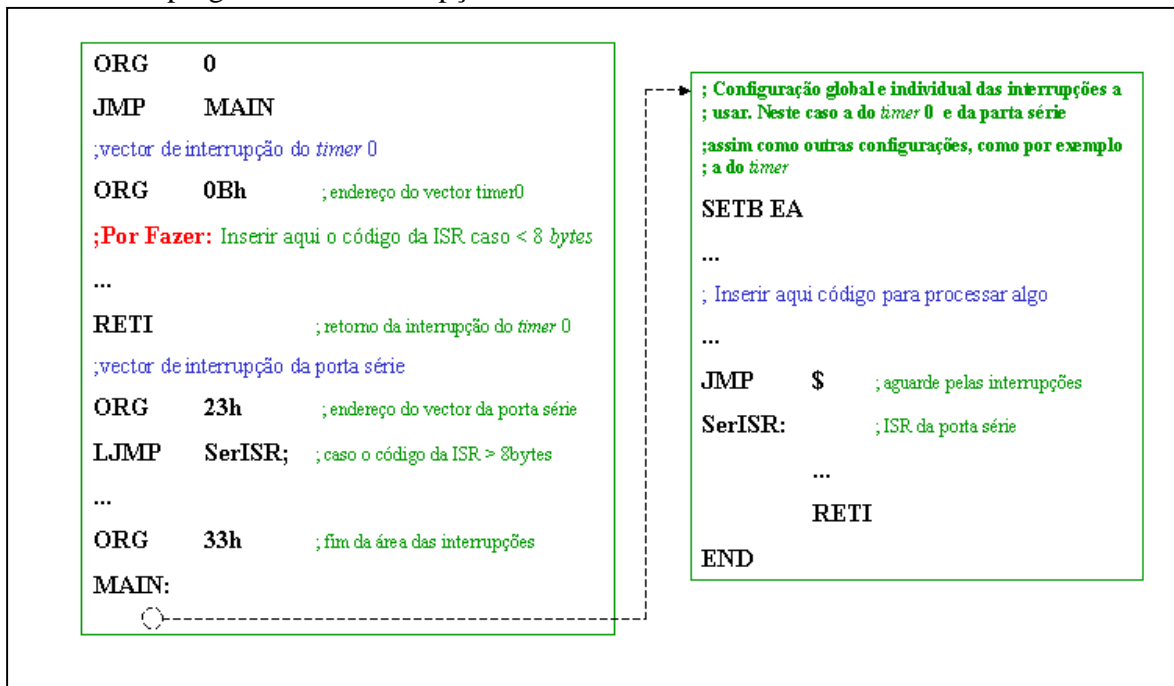
MICROPROCESSADORES

16/04/2003

Nome: Número: Turma:

- 1) Descreva a estrutura de um programa com interrupção no 8051 assim como a sequência de acções levadas a cabo até a conclusão da ISR associada

Estrutura do programa com interrupção no 8051



Sequência de acções

- i. A instrução actualmente em execução será executada até ao fim
- ii. O PC é salvaguardado na pilha
- iii. O estado actual da interrupção é guardado internamente
- iv. Interrupções são bloqueadas ao nível da interrupção
- v. O PC é carregado com o endereço do vector da ISR
- vi. ISR é executada

A ISR é concluída com a instrução RETI que:

- a. Restaura o PC a partir da pilha
- b. Restaura o estado das interrupções
- c. A função *main()* retoma a execução
- d. a partir do ponto onde foi interrompido

- 2) Escrever um programa para calcular a média de uma série de números e acender um led ligado a P3.4 caso o média for menor do que zero. A série de números deve ser definida e inicializada com 4 números entre -10 a 20 na memória externa a partir do endereço 1000H.

ADDR	EQU	1000H	;endereço da série	
LEN	EQU	4	;dimensão da série	
	ORG	0000H		
			;ex: inicializar a série com os números -10,-8,-6, -4	
	MOV	A,#10		
	MOV	B,#LEN		
	MOV	DPTR,#SERIE		
INIT:	MOVX	@DPTR,A		
	INC	DPTR		
	ADD	A,#2		
	DJNZ	B,INIT		
	MOV	R0,#LEN		
	MOV	B,#0		
	MOV	DPTR,#SERIE		
MEAN:	MOVX	A,@DPTR	nº de iterações para somatório	
	ADD	A,B	inicializar o somatório	
	MOV	B,A	apontar para a série de números	
	INC	DPTR	carregar o número apontado	
	DJNZ	R0,MEAN	somatório+=somatório	
	MOV	A,B	apontar para próximo nº	
	MOV	B,#LEN	todos os nº foram somados?	
	JB	ACC.7,NEG	em caso afirmativo,carrega o dividendo	
	DIV	AB	je divisor para o cálculo da média	
	JMP	FIM	somatório<0?	
NEG:	CALL	COMP2	senão obtém a média positiva	
	CLR	P3.4	je acabou	
	DIV	AB	obtem o somatório	
	CALL	COMP2	acende o led (ex: no nível 0)	
FIM:	JMP	\$	obtem média	
			obtem a média negativa	

COMP2: CPL
INC
RET
A ;obtem o compl-2
A ;do (A)

SERIE: XSEG AT ADDR
DS LEN
END

Entrada:

(1000hRAM) = F6h = -10

(1001hRAM) = F8h = -8

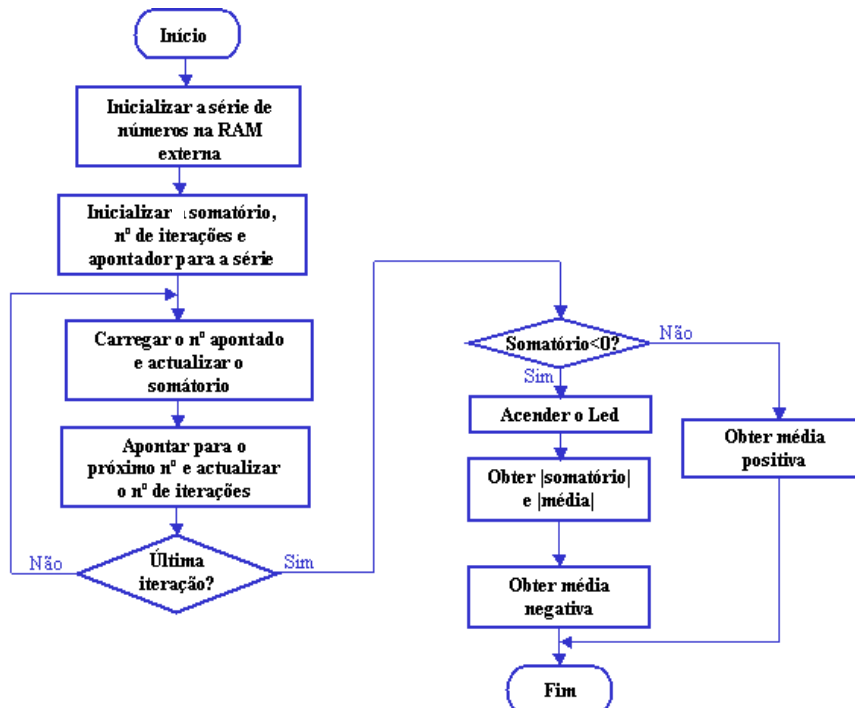
(1002hRAM) = FAh = -6

(1003hRAM) = FCh = -4

Saída:

MÉDIA = F9h = -7

P3.4 = 0 (ACENDER O LED)



Análise do Problema:

1. $-10 \leq \text{somatório} \leq 80$

Todos os cálculos podem ser efectuados ao nível do byte

2. $\text{somatório} \leq 0$

Deve-se efectuar a divisão usando o |somatório| e posteriormente multiplicar o quociente por -1

Licenciatura em Engenharia Electrónica Industrial

MICROPROCESSADORES

16/04/2003

- 3) Analise o programa em baixo, apontando todas as possíveis anomalias assim como possíveis soluções. Pretende-se que o programa seja capaz de responder aos três eventos externos para os quais foi programado.

```
1      ORG 0000H
2      LJMP MAIN
3      DW 12H,14H,1234H,"ABGG"
4      ORG 0033H
5  MAIN:
6      MOV TMOD,#02H
7      MOV TH0,#10
8      MOV IE,#77H
9      SUBB A,#15
10     JMP $

11     ORG 000BH
12     PUSH 01H
13     MOVX A,@DPTR
14     INC DPTR
15     INC DPTR
16     XCH A,R1
17     ADD A,R1
18     ADDC A,#15
19     POP 01H

20     ORG 0003H
21     MOV A,#10
22     MOV R1,#12
23     CALL SOM
24     CALL SUB
25     RETI

26     ORG 0013H
27     MOV A,P1
28     RETI

29 SUB: CLR C
30     SUBB A,R1
31     RET
32 SOM: ADD A,R1
33     RET
34     END
```

A tabela na linha 3 ocupa espaço reservado à interrupção externa 0 – 0003h. Resolve-se, por exemplo, usando ORG para posicionar a tabela

Repare que o bit EA não está activo: SETB EA (ou 88h)
As ISRs que alteram o carry devem preservar o PSW

Repare que esta ISR ocupa mais do que 8 bytes. Deve-se usar um LJMP no endereço 000Bh e colocar o código noutra posição, evitando a sobreposição com o código da ISR da interrupção externa 1

Repare que esta interrupção nunca é gerada dado que o timer não foi “arrancado” no MAIN após o a instrução da linha 7: SETB TR0

Deve-se preceder ADD por CLR C, caso não estiver associado a uma subtracção multi-bytes.

Restaurar o PSW devido ao carry; Antes da linha 12 deve salvar o PSW devido ao carry
Falta um RETI após última instrução

Repare que esta ISR ocupa mais do que 8 bytes. Deve-se usar um LJMP no endereço 0003h e colocar o código noutra posição, evitando a sobreposição com o código da ISR da interrupção do timer 0

No MAIN, antes de JMP \$ configurar P1 para entrada, usando a instrução: MOV P1, #0FFH