

Operações Lógicas

1) Analise o programa em Assembly da família MCS51 dado a seguir e forneça a equação lógica do resultado final armazenado no acumulador em função dos registros utilizados:

```
LOGICA: MOV A, R0
CPL A
ANL R1
MOV B, A
MOV A, R2
ANL A, R3
CPL A
XRL R4
ORL A, B
LJMP LOGICA
```

2) Implemente um programa em Assembly da família MCS51 que realize a seguinte equação lógica: $ACC = [(R0 + R1) \cdot R3] \text{ or } R4$.

3) Apresente 2 formas de se zerar o conteúdo do acumulador utilizando instruções lógicas do Assembly da família MCS51.

4) Apresente 2 formas de se complementar o conteúdo do acumulador utilizando instruções lógicas do Assembly da família MCS51.

5) Implemente um programa em Assembly da família MCS51 que seja capaz de setar apenas o bit 7 e de zerar apenas o bit 0 do registro de função especial B, sem alterar o estado dos seus demais bits.

6) Qual é o significado aritmético de uma rotação à esquerda do acumulador?

7) Qual é o significado aritmético de uma rotação à direita do acumulador?

Operações Aritméticas

1) Execute os programas dados a seguir no simulador no Proteus e tire conclusões sobre o efeito de cada instrução executada. Para uma melhor noção do que ocorre, sugere-se que os programas sejam executados passo a passo, visualizando-se as janelas "CPU Registers" e "CPU Internal Memory".

a) SOMA1: MOV PSW, #00H ;adicao de 16 bits:

```
MOV R0, #23H ; R0R1
MOV R1, #90H ;+R2R3
MOV R2, #14H ;-----
MOV R3, #59H ; R4R5
MOV A, R1
ADD A, R3 ;adiciona R1 e R3
MOV R5, A ;LSB em R5
MOV A, R0
ADDC A, R2 ;adiciona R0 e R2
MOV R4, A ;MSB em R4
LJMP SOMA1
```

b) SOMA2: MOV PSW, #00H ;adicao de 16 bits em BCD:

```
MOV R0, #23H ; R0R1
MOV R1, #90H ;+R2R3
MOV R2, #14H ;-----
MOV R3, #59H ; R4R5
MOV A, R1
```

```

ADD A, R3 ;adiciona R1 e R3
DA A ;ajusta resultado
MOV R5, A ;LSB em R5
MOV A, R0
ADDC A, R2 ;adiciona R0 e R2
DA A ;ajusta resultado
MOV R4, A ;MSB em R4
LJMP SOMA2

```

```

c) SUBT: MOV PSW, #00H ;subtracao de 16 bits:
MOV R0, #23H ; R0R1
MOV R1, #90H ; -R2R3
MOV R2, #14H ; ----
MOV R3, #59H ; R4R5
MOV A, R1
SUBB A, R3 ;subtrai R3 de R1
MOV R5, A ;LSB em R5
MOV A, R0
SUBB A, R2 ;subtrai R2 de R0
MOV R4, A ;MSB em R4
LJMP SUBT

```

```

d) MULT: MOV PSW, #00H ;multiplicacao de 16 bits:
MOV R0, #23H ; R0R1
MOV R1, #90H ; xR2R3
MOV R2, #14H ; -----
MOV R3, #59H ; R4R5R6R7
MOV A, R1
MOV B, R3
MUL AB ;multiplica R1 e R3
MOV R7, A ;LSB em R7
MOV R6, B ;MSB em R6R7
MOV A, R1
MOV B, R2
MUL AB ;multiplica R1 e R2
ADD A, R6
MOV R6, A ;adiciona LSB em R6
MOV A, B
ADDC A, #00H
MOV R5, A ;MSB em R5
MOV A, R0
MOV B, R3
MUL AB ;multiplica R0 e R3
ADD A, R6
MOV R6, A ;adiciona LSB em R6
MOV A, B
ADDC A, R5 ;adiciona MSB em R5
MOV R5, A
MOV A, R0
MOV B, R2
MUL AB ;multiplica R0 e R2
ADD A, R5
MOV R5, A ;adiciona LSB em R5
MOV A, B
ADDC A, #00H
MOV R4, A ;MSB em R4
LJMP MULT

```

Manipulação de Variáveis Booleanas (Bits)

- 1) Apresente 2 formas de setar o estado do flag de carry utilizando instruções de manipulação de variáveis booleanas e operações lógicas do Assembly da família MCS51.
- 2) De que maneira você acredita que o microcontrolador identifica se os operandos das instruções `ANL`, `ORL` e `MOV` são registros ou bits? (Sugestão: analise a tabela de instruções completa existente nos anexos.)
- 3) Supondo que as variáveis booleanas X e Y foram atribuídas aos bits de endereço 10H e 20H, implemente um programa em Assembly da família MCS51 que realize a seguinte equação lógica: $C=X(or)Y$.

Interrupção

- 1) Determinado sistema de controle baseado no 80C31 necessita de duas fontes de interrupção: a interrupção da interface serial e a interrupção externa 0. Elabore um trecho de programa em Assembly que habilite essas interrupções, determinando a interrupção da interface serial como mais prioritária. Programe também a interrupção externa 0 para sensibilidade a borda de descida.
- 2) Dado o seguinte trecho de programa em Assembly para um sistema microcontrolado baseado no 80C32, responda as questões abaixo:

```
ORG 0003H
EXT0: PUSH PSW
SETB RS0
MOV A, #20H
MOV R0, #40H
POP PSW
RETI
ORG 0013H
EXT1: PUSH ACC
MOV A, #20H
MOV R0, #10H
POP ACC
RETI
ORG 0100H
PROG: CLR RS0 ;o programa começa aqui!
MOV TCON, #00000101B
MOV IP, #00000100B
MOV IE, #10000001B
MOV R0, #50H ;instrução X
MOV A, #05H ;instrução Y
ADD A, R0 ;instrução Z
FINAL: JMP FINA
```

- a) Supondo que ocorre uma borda de descida no sinal aplicado ao terminal P3.2 (INT0) durante a execução da instrução X, qual será o valor contido no acumulador ao final do programa?
- b) Supondo que ocorre uma borda de descida no sinal aplicado ao terminal P3.2 (INT0) durante a execução da instrução Y, qual será o valor contido no acumulador ao final do programa?
- c) Supondo que ocorrem bordas de descida nos sinais aplicados aos terminais P3.2 (INT0) e P3.3 (INT1) simultaneamente durante a execução da instrução Z, qual será o valor contido no acumulador ao final do programa?

TEMPORIZADORES / CONTADORES DE EVENTOS

1) Implemente um programa em Assembly da família MCS51 que simule um freqüencímetro. Isto deverá ser feito através da contagem de pulsos que o sinal digital de entrada possui em um intervalo de um segundo. Utilize os temporizadores internos do 8031 para temporização e contagem e utilize um dos bits do port P1 como entrada do sinal digital cuja freqüência deseja-se medir.

2) A subrotina dada a seguir gera um tempo de atraso fazendo uso do T/C0 de um 8031. Supondo que a freqüência de clock do microcontrolador é de 12MHz, calcule o tempo de atraso gerado.

```
;*****  
; ATRASO - gera um tempo de atraso por hardware  
; ENTRADA: nada  
; SAIDA: tempo de atraso  
;*****  
ATRASO: MOV TCON, #00000000B  
MOV TMOD, #00000001B ;T/C0 no modo 1  
MOV TH0, #HIGH(1000H);MSB da constante de tempo  
MOV TL0, #LOW (1000H);LSB da constante de tempo  
SETB TR0 ;dispara contagem do T/C0  
ESPERA: JNB TF0, ESPERA ;aguarda estouro  
CLR TF0 ;para contagem do T/C0  
CLR TR0 ;reseta estouro  
RET
```

3) Suponha que determinada aplicação empregando o microcontrolador 89C51 exija interrupções periódicas para realizar a varredura de displays de 7 segmentos. Escreva um trecho de programa em Assembly que apenas programe o temporizador 0 no modo 2, de maneira que sejam geradas interrupções a cada 1ms. Esse trecho de programa deve prever também a habilitação da interrupção correspondente e o disparo da contagem do temporizador. A freqüência de clock do microcontrolador é de 6MHz.

Interface Serial

Taxa de Transmissão e Recepção Serial

Modo 0: a taxa é igual a 1/12 da freqüência de clock (fixa).

Modo 2: a taxa depende apenas do valor do bit SMOD no registro PCON.

Para SMOD=0 a taxa é igual a 1/32 da freqüência de clock e para SMOD=1 a taxa é igual a 1/64 da freqüência de clock.

Modos 1 e 3: a taxa é fornecida pelo T/C1, de maneira que os dados são transmitidos a cada estouro de contagem. Nesse caso o modo de funcionamento mais comum do T/C1 é o Modo 2 (recarga automática). A interrupção do T/C1 deve ser desabilitada.

O cálculo da taxa de transmissão e recepção da interface serial com o T/C1 no Modo 2 pode ser feito pela seguinte fórmula:

$$Taxa = \frac{2^{SMOD}}{32} \cdot \frac{f_{clock}}{12.(256 - TH1)}$$

A seguinte tabela com os valores mais usuais de taxa de transmissão e recepção indica os valores de recarga do T/C1 operando no Modo 2, com SMOD=0 e $f_{clock}=11,0592\text{MHz}$:

Taxa	300 bps	600 bps	1200 bps	2400 bps	4800 bps	9600 bps
TH1	160	208	232	244	250	253

1) Com base nas informações acima, programe a interface serial de um microcontrolador 80C31 com clock de 7,3728MHz para operar com comunicação serial assíncrona a 4800 bps. Escreva um programa em Assembly que inicialize a interface serial e o temporizador responsável pela taxa de transmissão e recepção de modo a satisfazer esses requisitos. Uma vez programados os registros especiais necessários, envie o valor B2H pela interface serial.