

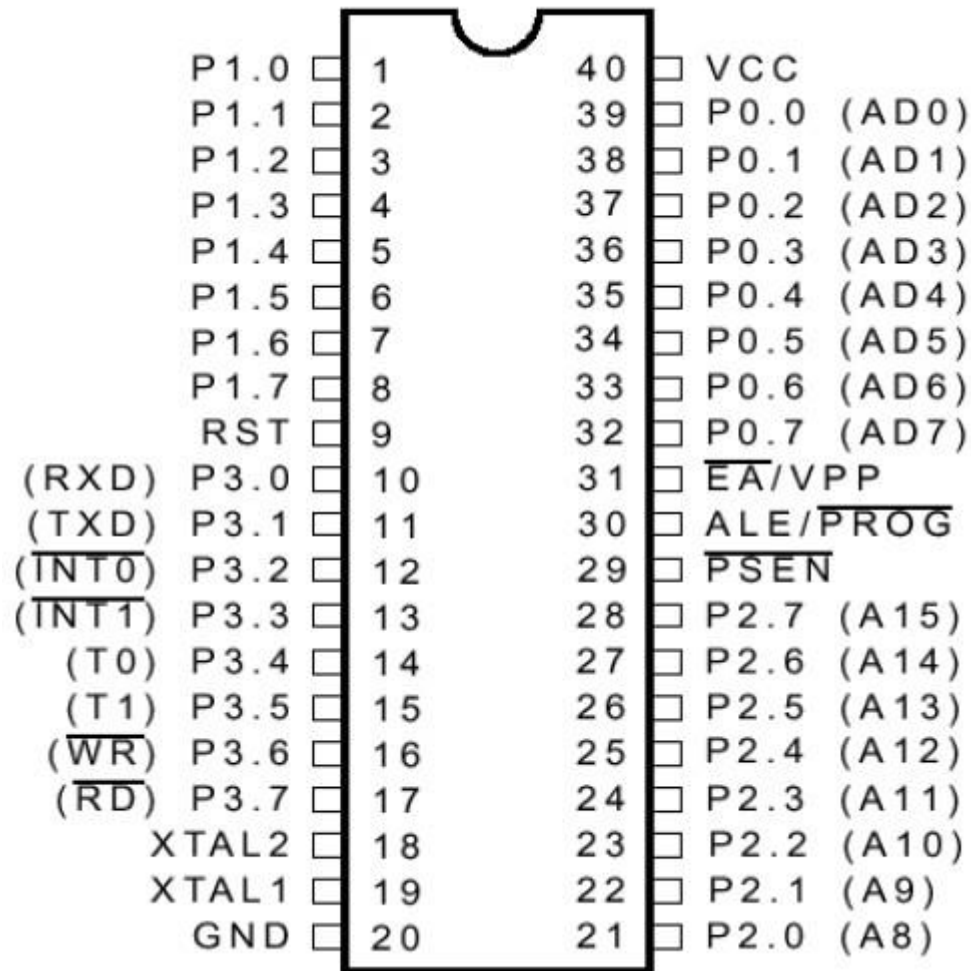
Departamento de Engenharia Elétrica e de Computação
EESC-USP

SEL-415 Introdução à Organização de Computadores

**Aula Exercícios assembly 8051
2ª. Prova –Parte2**

Profa. Luiza Maria Romeiro Codá

Microcontrolador 8051



Microcontrolador 8051

descrição dos pinos

| Pino | Nome | E/S | Função |
|---------|-------------|-----|---|
| 32 a 39 | P0.0 a P0.7 | E/S | O Port 0 é uma interface de E/S bidirecional com 8 bits individualmente endereçáveis em dreno aberto. Terminais que estejam no estado lógico 1 podem ser utilizados como entradas de alta impedância. O Port 0 também atua como barramento de dados e gera de maneira multiplexada a parte menos significativa dos endereços durante acessos às memórias externas de programa ou dados. |
| 1 a 8 | P1.0 a P1.7 | E/S | O Port 1 é uma interface de E/S bidirecional com 8 bits individualmente endereçáveis e resistores de <i>pull-up</i> internos. Terminais que estejam no estado lógico 1 podem ser utilizados como entradas. No 89S52, alguns pinos deste <i>port</i> possuem funções alternativas. |
| 21 a 28 | P2.0 a P2.7 | E/S | O Port 2 é uma interface de E/S bidirecional com 8 bits individualmente endereçáveis e resistores de <i>pull-up</i> internos. Terminais que estejam no estado lógico 1 podem ser utilizados como entradas. O Port 2 também gera a parte mais significativa dos endereços durante acessos às memórias externas de programa ou dados. |
| 10 a 17 | P3.0 a P3.7 | E/S | O Port 3 é uma interface de E/S bidirecional com 8 bits individualmente endereçáveis e resistores de <i>pull-up</i> internos. Terminais que estejam no estado lógico 1 podem ser utilizados como entradas. O Port 3 também possui funções alternativas em todos os pinos. |

Microcontrolador 8051

descrição dos pinos

| Pino | Nome | E/S | Função |
|------|--------------------------|-----|---|
| 40 | VCC | | Potencial de alimentação (+5v) |
| 20 | VSS | | Potencial de referência (terra). |
| 9 | RST | E | Quando aplicado nível lógico 1 a este terminal durante 2 ciclos de máquina (com o oscilador operando) ocorre o <i>reset</i> do microcontrolador. Um resistor interno conectado a VSS permite o <i>power-on-reset</i> com apenas um capacitor externo conectado a VCC. |
| 19 | XTAL1 | | Entrada do amplificador inversor do oscilador e entrada do gerador de <i>clock</i> interno. |
| 18 | XTAL2 | | Saída do amplificador inversor do oscilador. |
| 29 | $\overline{\text{PSEN}}$ | S | <i>PROGRAM STORE ENABLE</i> : Habilita o acesso à memória de programa externa durante a busca de instruções. Permanece em nível lógico 1 durante o acesso da memória de programa interna. |
| 30 | ALE | S | <i>ADDRESS LATCH ENABLE</i> : Fornece o sinal para armazenamento da parte menos significativa do endereço durante acessos às memórias externas de programa ou dados. |
| 31 | $\overline{\text{EA}}$ | E | <i>EXTERNAL ACCESS</i> : Quando em nível lógico 1, as instruções da memória de programa interna são executadas. Quando em nível lógico 0, todas as instruções são buscadas na memória de programa externa. No caso do 8031 este terminal deve sempre estar em nível lógico 0. |

Microcontrolador 8051

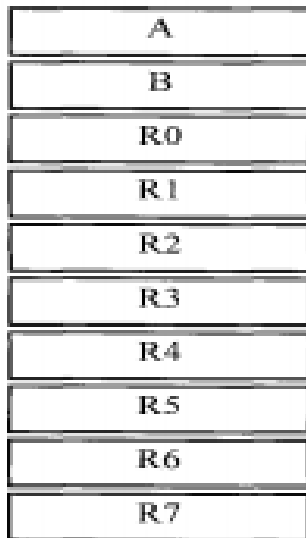
Função especial da porta P3

| Pino | Nome | Função |
|------|--------------------------|---|
| P3.0 | RXD/ data | Receptor da interface serial assíncrona ou entrada e saída de dados da interface serial síncrona. |
| P3.1 | TXD/ clock | Transmissor da interface serial assíncrona ou saída de clock da interface serial síncrona. |
| P3.2 | $\overline{\text{INT0}}$ | Entrada de interrupção externa 0 ou sinal de controle para o contador 0. |
| P3.3 | $\overline{\text{INT1}}$ | Entrada de interrupção externa 1 ou sinal de controle para o contador 1. |
| P3.4 | T0 | Entrada externa para o contador 0. |
| P3.5 | T1 | Entrada externa para o contador 1. |
| P3.6 | $\overline{\text{WR}}$ | Sinal de escrita na memória externa de dados. |
| P3.7 | $\overline{\text{RD}}$ | Sinal de leitura na memória externa de dados. |

Microcontrolador 8051(cont)

Registadores de uso do programador

Registadores de 8 bits

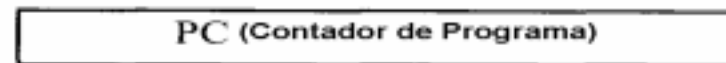


Registadores de 16 bits

DPTR

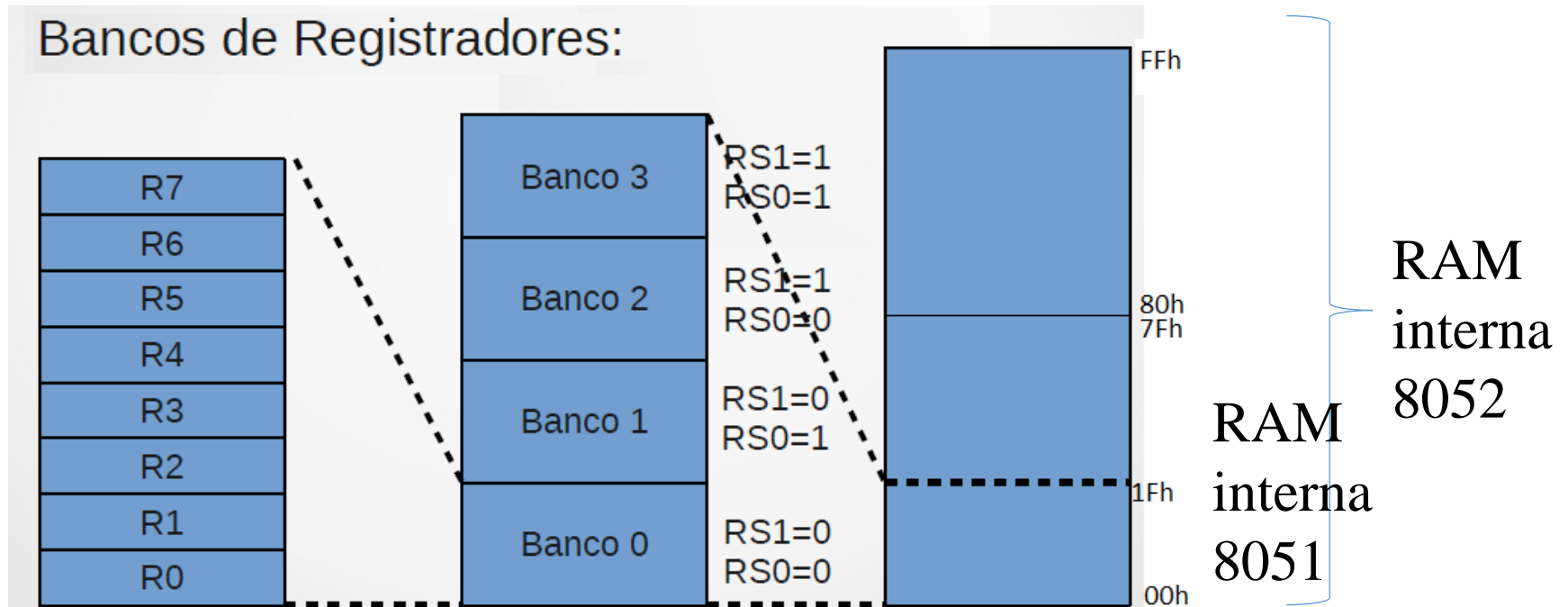


PC



Microcontrolador 8051(cont)

RAM Interna

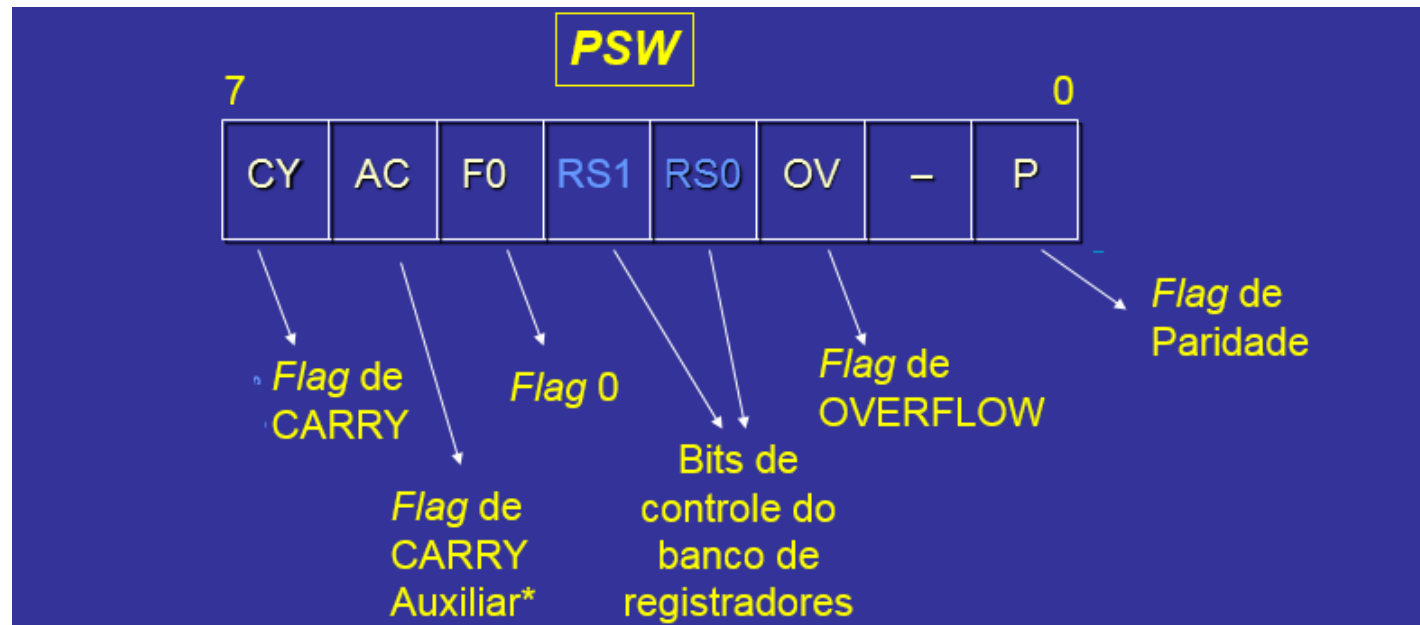


Microcontrolador 8051: RAM interna

Banco de Registradores: selecionado pelos bits RS0 e RS1 do PSW

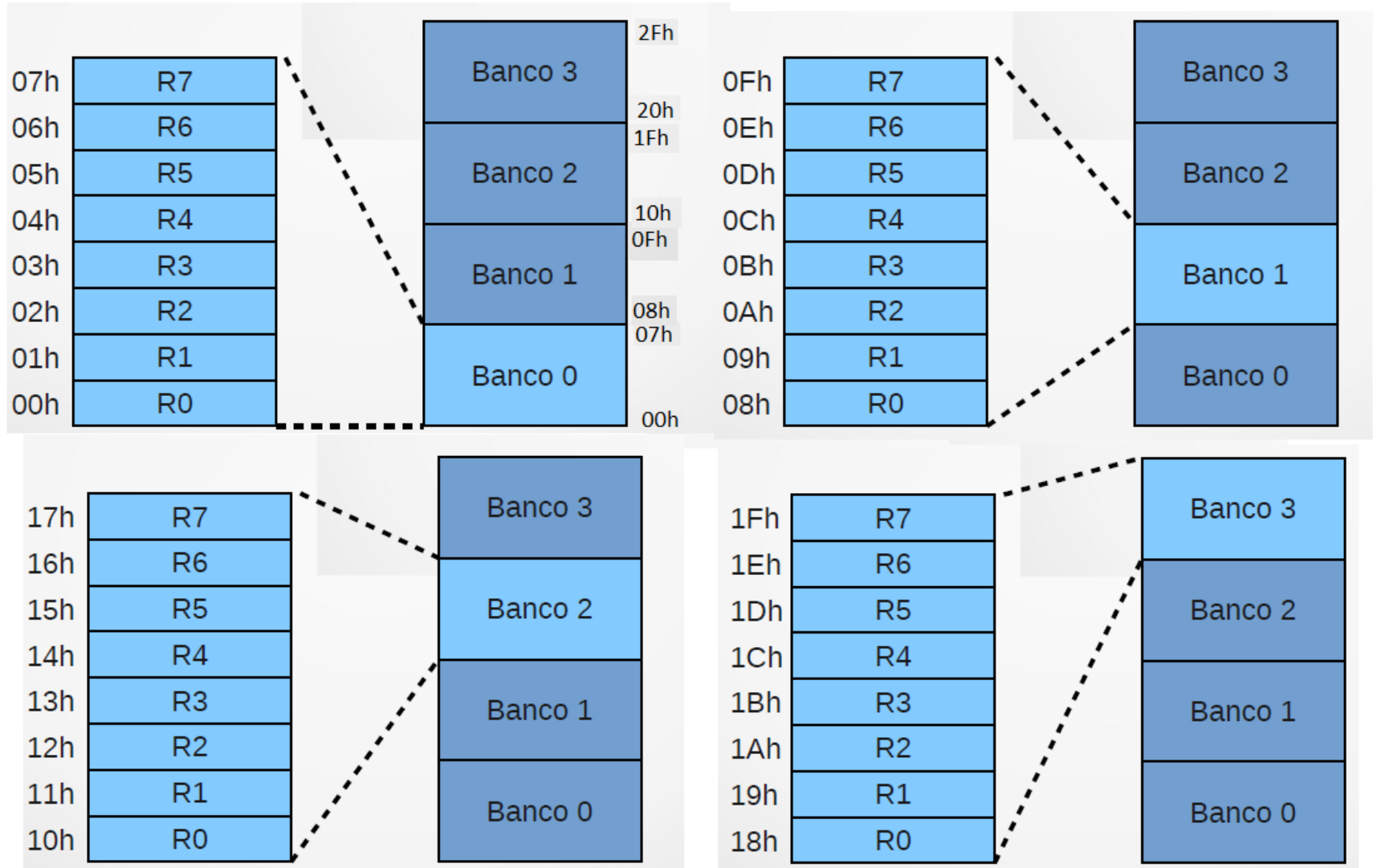
| Nº do banco | RS0 | RS1 |
|-------------|-----|-----|
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 2 | 1 | 0 |
| 3 | 1 | 1 |

PSW (“*program status word*”) : contém *Flags* que são bits indicadores de estado são colocados em “1” ou “0” dependendo do resultado das operações da CPU

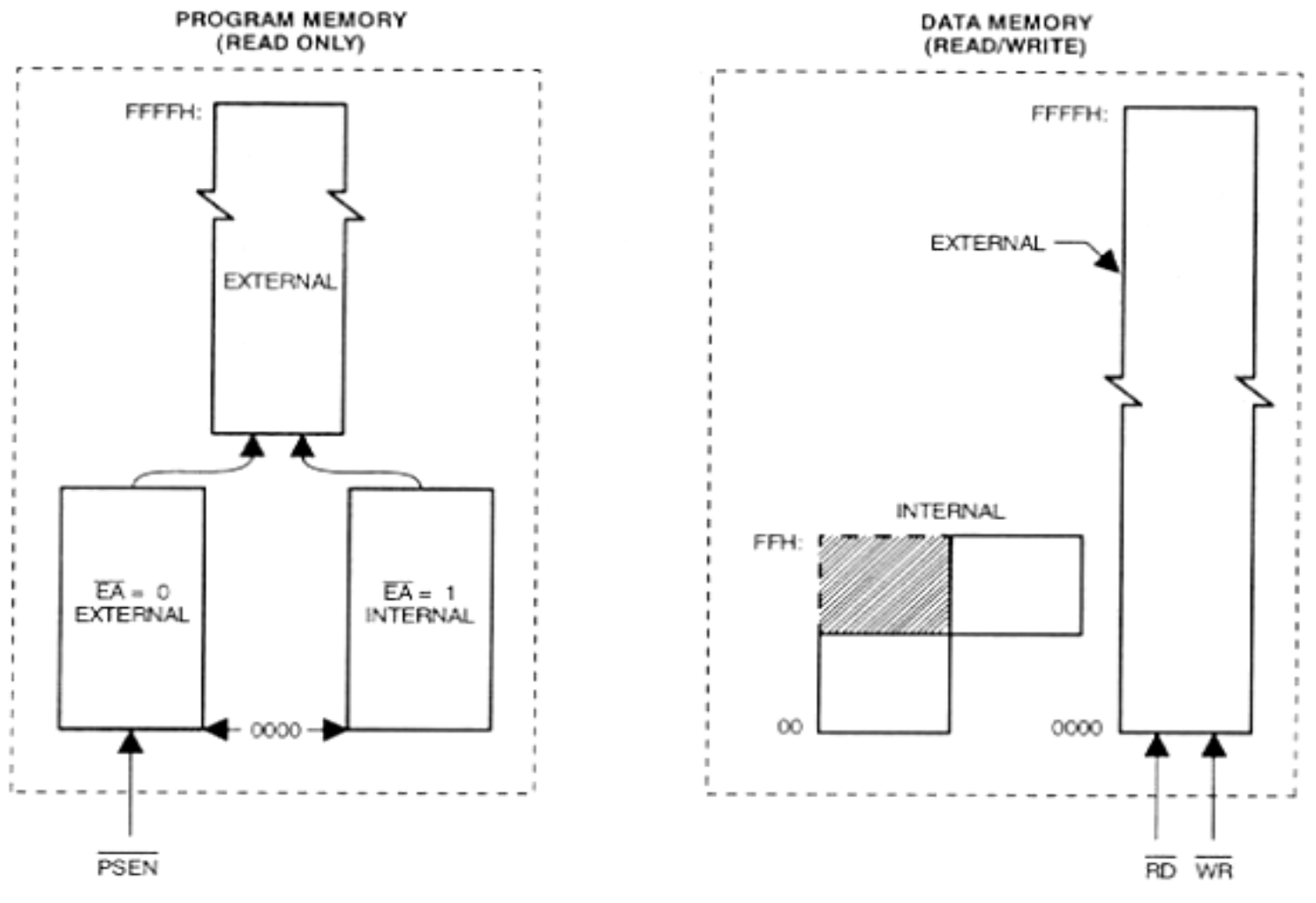


Microcontrolador 8051: RAM interna

Endereço do Banco de Registradores

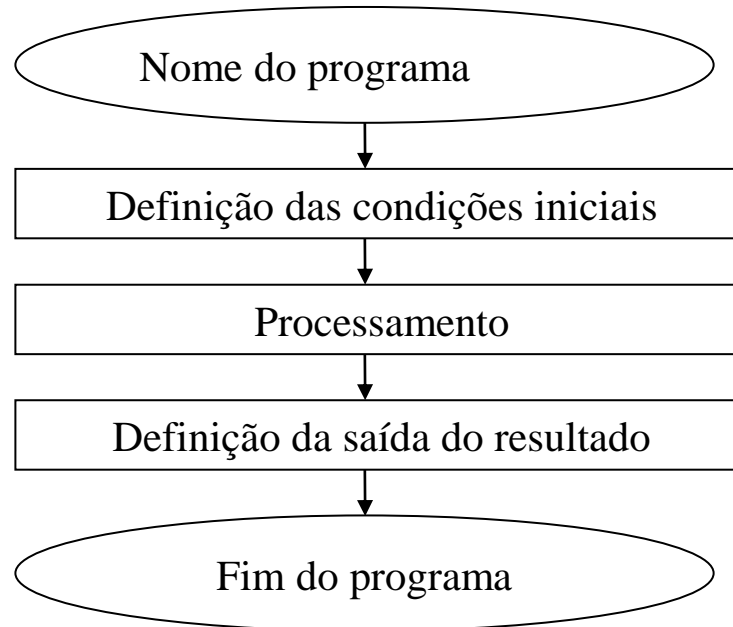


Organização das memórias do Microcontrolador 8051

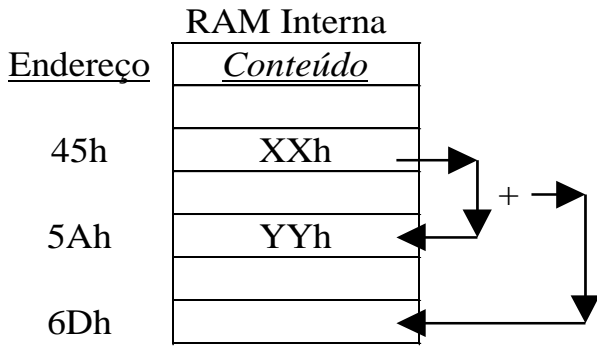


FLUXOGRAMA

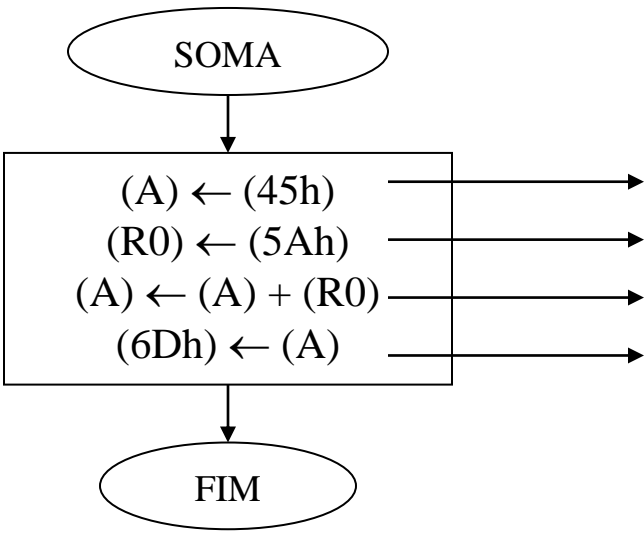
Representação típica de um fluxograma de uma rotina de um programa simples.



Programa de adição de 2 números armazenados na RAM interna utilizando a instrução ADD A,Rn



O programa soma o conteúdo da posição 45h com o conteúdo da posição 5Ah RAM interna e armazena o resultado na posição 6Dh da RAM interna

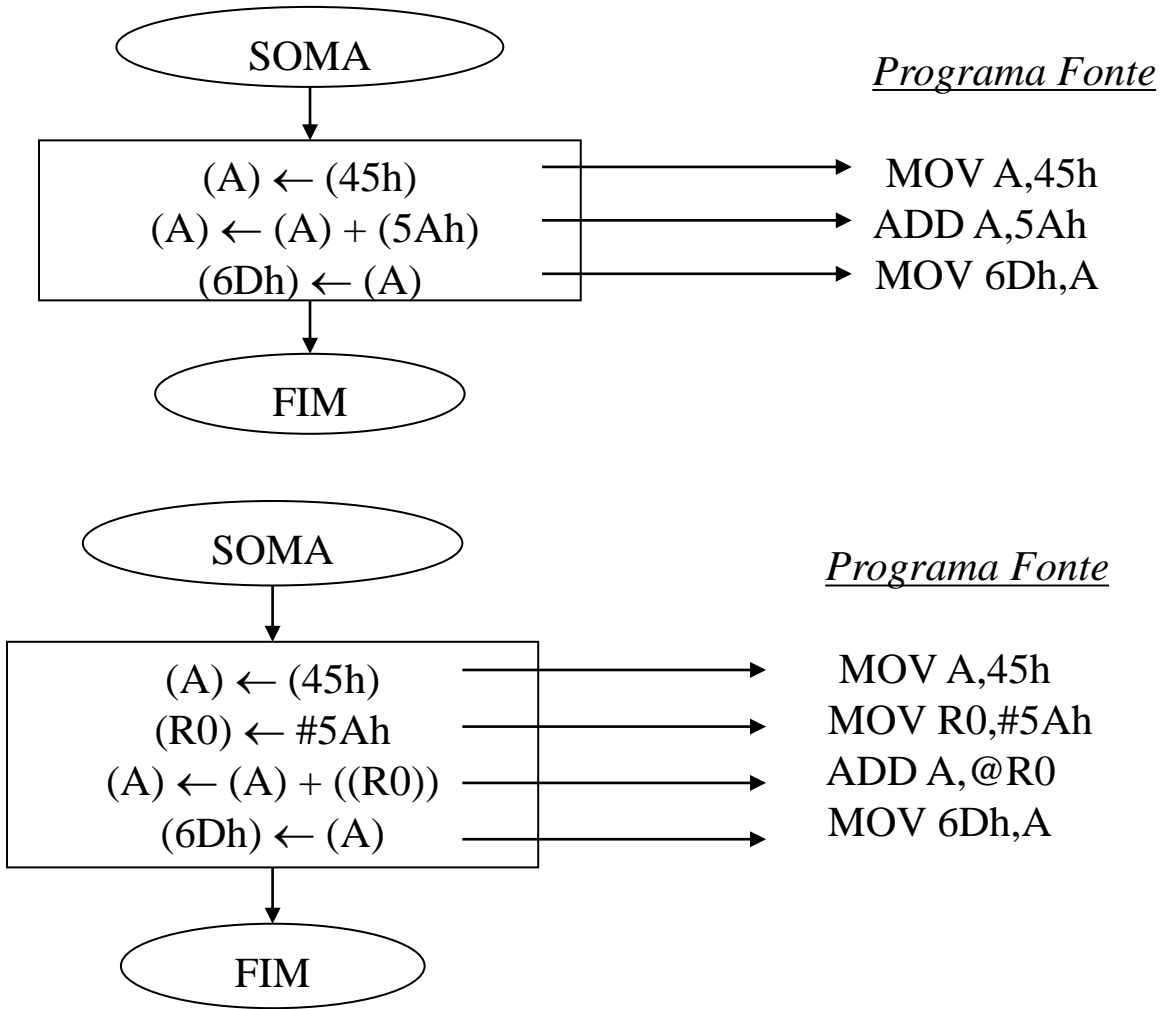


Programa Fonte

```
ORG 0
MOV A,45h
MOV R0,5Ah
ADD A,R0
MOV 6Dh,A
SJMP $
END
```

| Endereço da EEPROM | Conteúdo da EEPROM |
|--------------------|--------------------|
| 0000h | 74h |
| 0001h | 45h |
| 0002h | 78h |
| 0003h | 5Ah |
| 0004h | 28h |
| 0005h | F5h |
| 0006h | 6Dh |
| 0007h | 80h |
| 0008h | FEh |

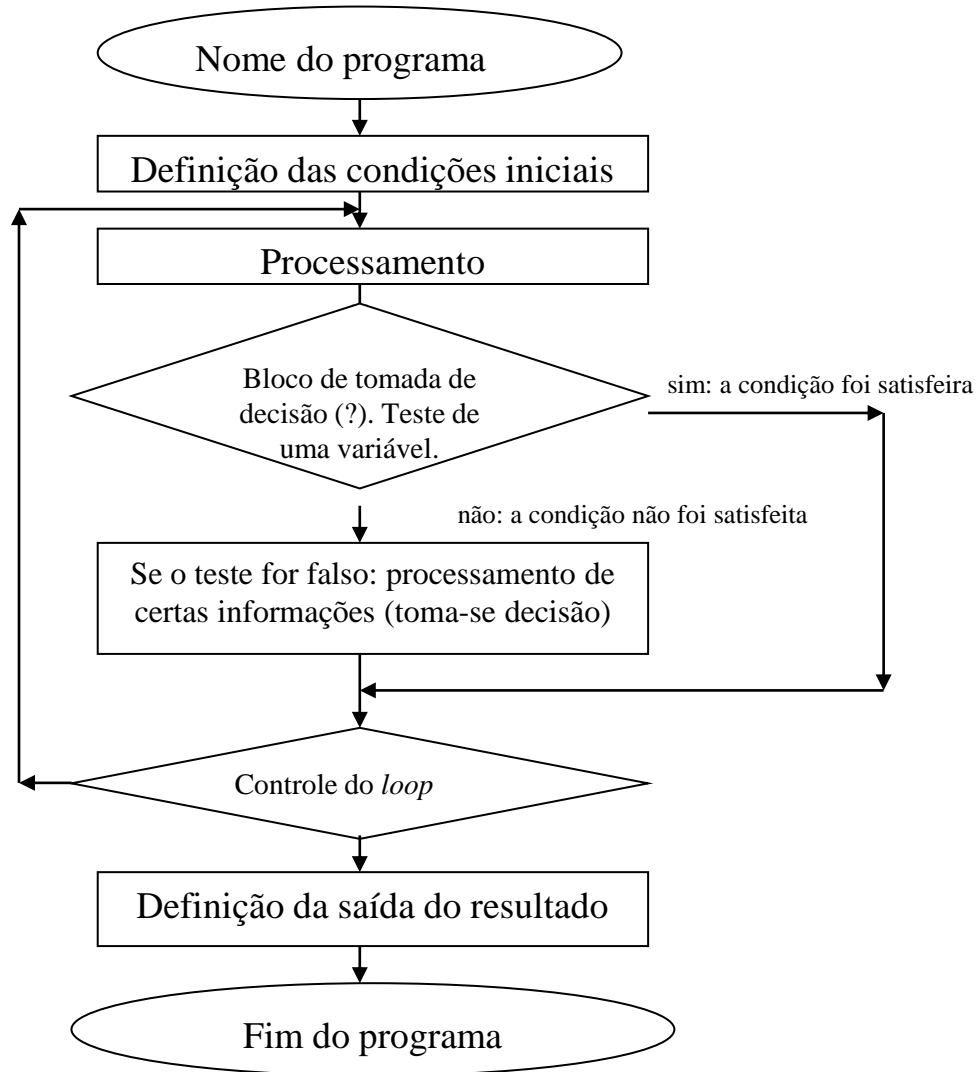
Progama de adição de 2 números armazenados na RAM interna utilizando a instrução ADD A,Rn e ADD A, @Ro



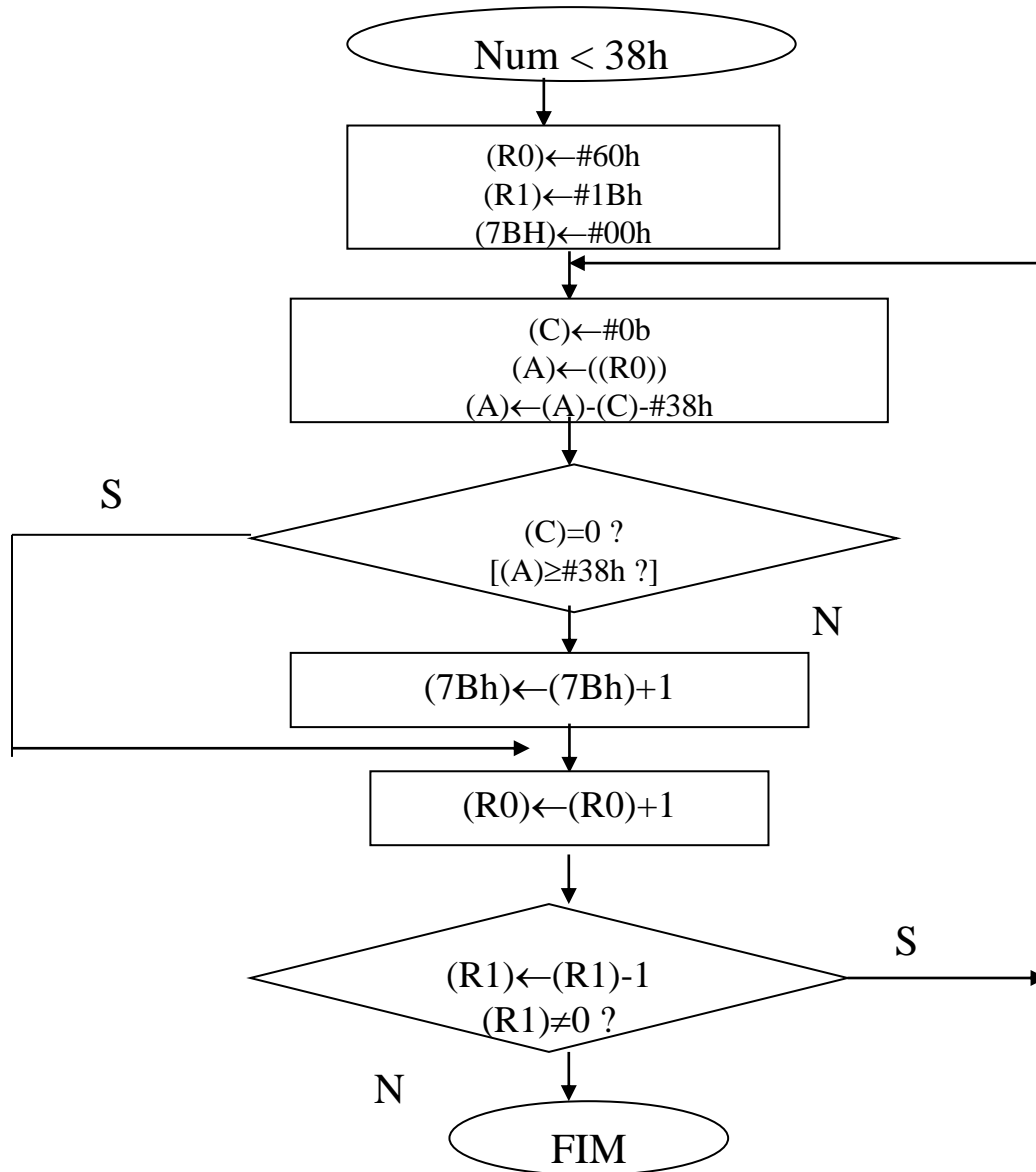
Fonte: Microcontolador 8051- Prentice Hall

FLUXOGRAMA

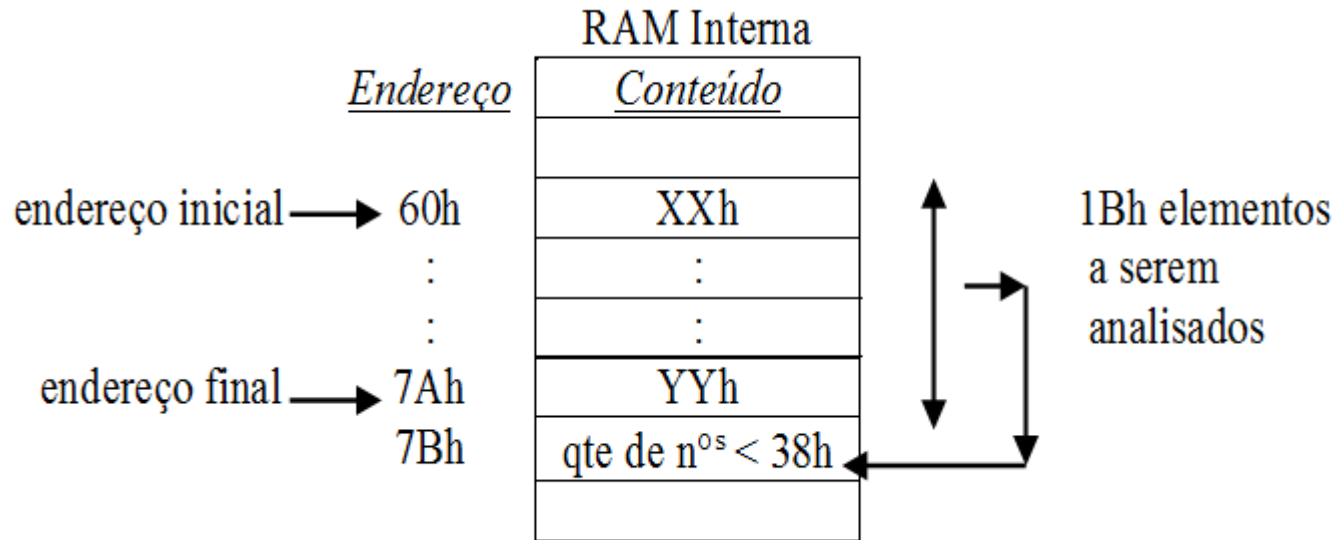
Representação típica de um fluxograma de uma rotina de um programa com *loop*.



Programa que calcula a quantidade de números menores que 38h de um *buffer* de memória.



Programa que calcula a quantidade de números menores que 38h de um *buffer* de memória.



Programa Fonte

Programa que calcula a quantidade de números menores que 38h de um *buffer* de memória.

```
; Início do programa que calcula a qte de elementos < #38h
    ORG 100h
INICIO: MOV    R0,#60h      ;Endereço inicial do buffer de memória
        MOV    R1,#1Bh     ;Quantidades de elementos do buffer de memória
        MOV    7Bh,#00     ;Zera o contador que armazena a qte de num < #38h
ADR2:   CLR     C           ;Limpa o conteúdo do carry-bit
        MOV    A,@R0       ;Armazena no (A) um elemento do buffer
        SUBB   A,#38h      ;Subtrai o (A) da constante #38h e define o flag (C)
        JNC    ADR1        ;Se (C)=0 (A>=#38h) => (PC)=ADR1 [não soma um no (7Ah)]
        INC    7Bh         ;Se (C)=1 (A<#38h), soma um no (7Bh)
ADR1:   INC     R0          ;Aponta para a próxima posição de memória a ser analisada
        DJNZ   R1,ADR2     ;Se a qte de elementos a serem analisadas é ≠ 0, então salta para ADR2
        END              ;Termina o programa se (R1)=0.
```

Programa que calcula a quantidade de números menores que 38h de um *buffer* de memória.

```

                ORG 0100h
INICIO:        MOV     R0,#60h
                MOV     R1,#1Bh
                MOV     7Bh,#00
ADR2:          CLR     C
                MOV     A,@R0
                SUBB    A,#38h
                JNC     ADR1
                INC     7Bh
ADR1:          INC     R0
                DJNZ    R1,ADR2
                END
    
```

Informações sobre as instruções

| Instrução | OPCODE | Nºbytes |
|--------------------|--------|---------|
| MOV R0, #data | 78h | 2 |
| MOV R1, #data | 79h | 2 |
| MOV data addr,data | 75h | 3 |
| CLR C | C3h | 1 |
| MOC A,@R0 | E6h | 1 |
| SUBB A, #data | 94h | 2 |
| JNC addr rel | 50h | 2 |
| INC data addr | 05h | 2 |
| INC R0 | 08h | 1 |
| DJNZ R1,end rel | D9h | 2 |

Conteúdo da memória de programa

| | | | | Endereço da EEPROM | Conteúdo da EEPROM |
|---------|-----------|---------|---|--------------------|--------------------|
| | ORG 0100h | | | 0100h | 78h |
| INICIO: | MOV | R0,#60h | → | 0101h | 60h |
| | MOV | R1,#1Bh | → | 0102h | 79h |
| | | | | 0103h | 1Bh |
| | | | | 0104h | 75h |
| | MOV | 7Bh,#00 | → | 0105h | 7Bh |
| | | | | 0106h | 00h |
| ADR2: | CLR | C | → | 0107h | C3h |
| | MOV | A,@R0 | → | 0108h | E6h |
| | SUBB | A,#38h | → | 0109h | 94h |
| | | | | 010Ah | 38h |
| | JNC | ADR1 | → | 010Bh | 50h |
| | | | | 010Ch | 03h |
| | INC | 7Bh | → | 010Dh | 05h |
| | | | | 010Eh | 7Bh |
| ADR1: | INC | R0 | → | 010Fh | 08h |
| | DJNZ | R1,ADR2 | → | 0110h | D9h |
| | END | | | 0111h | F5h |

Conteúdo da memória RAM interna

MOV R0,#60h : RAM interna posição 00h(pois bloco 0 de registradores que está ativo) é armazenado 60h

MOV R1,#1Bh : RAM interna posição 01h(pois bloco 0 de registradores que está ativo) é armazenado 1Bh

MOV 7Bh,#00 : RAM interna posição 7Bh é armazenado 00h quantidade de dados memores que 38h

ADR2: CLR C : O carry C do PSW é zerado

MOV A,@R0: o conteúdo da posição 60h da RAM interna é copiado para o registrador acumulador (primeiro dado a ser verificado)

SUBB A,#38h : é extraído do valor do acumulador o valor 38h

JNC ADR1: o programa é desviado para o endereço que contém a instrução

INC R0 no endereço 010Fh (PC = 010Fh) se o bit Carry do PSW for diferente de zero, caso em que A é maior que 38h

INC 7Bh :incrementa o conteúdo do endereço7Bh (se 00h será 01h)

ADR1: INC R0 : R0 é incrementado (de 60 p 61h) para apontar p o prox dado

DJNZ R1,ADR2: decrementa o valor de R1(que inicialmente é 1Bh passando para 1Ah e salta para a posição ADR2 no endereço 0107h (PC=0107h) se R1 não for zero

END

Modo de endereçamento das instruções do 8051

Endereçamento Direto: O endereço do operando é especificado por um campo na instrução. Somente Os Bytes menos significativos da RAM de dados interna e região de SFR podem ser diretamente acessadas; (Ex; MOV 30h,A)

Endereçamento Indireto: a instrução especifica um registrador que contém o endereço do operando. RAM externa e RAM interna podem ser indiretamente endereçáveis.
Se o endereço for de 8 bits, seu valor é armazenado em R0 ou R1;
Se o endereço for de 16 bits, seu valor é armazenado no registrador DPTR;
(Exs: MOVX A,@DPTR e MOV @R0,A)

Endereçamento por Registrador: Instruções que acessam registradores de R0 a R7 do banco de registradores ou algumas instruções específicas a certos registradores.
Tais instruções são eficientes por eliminarem um byte de endereço. O próprio OP-CODE já indica qual a tarefa a ser realizada. (Ex: MOV R0,A)

Endereçamento Imediato: quando uma constante é apresentada na instrução após o símbolo #;
(Ex : ADD R0,#30h)

Endereçamento Indexado: Sómente a memória de Programas, cuja única operação é leitura, pode ser acessada por esse modo. É usado para fazer leituras em tabelas na Memória de Programa. Um registrador de 16 bits(DPTR) aponta para o início da tabela enquanto o Acumulador é ajustado para a e-nesima posição da mesma. O endereço de um valor da tabela é formado pelo soma do valor do acumulador com o valor do DPTR. (Ex: **MOVC A,@A+DPTR**)

Modo de endereçamento das instruções e Memória acessada

| Instrução | Modo de endereçamento | Memória acessada |
|---------------------|-----------------------|-----------------------|
| MOV R0,#60h | imediato | RAM interna LSB |
| MOV R1,#1Bh | imediato | RAM interna LSB |
| MOV 7Bh,#00 | imediato | RAM interna LSB |
| CLR C | Por registrador | SFR |
| MOV A,@R0 | Indireto | RAM interna LSB e MSB |
| SUBB A,#38h | imediato | SFR |
| JNC ADRI | direto | RAM interna LSB |
| INC 7Bh | direto | RAM interna LSB |
| INC R0 | Por registrador | RAM interna LSB |
| DJNZ R1,ADR2 | direto | RAM interna LSB |

FIM