Departamento de Engenharia Elétrica e de Computação - EESC-USP

SEL-0415 Introdução à Organização de Computadores

Aula 9 : Pilha e Interrupção

Profa. Luiza Maria Romeiro Codá

Tópicos:

1 - Modos de endereçamento do 8051

2 - Pilha e instruções de Pilha

3 - Instruções que usam pilha:
- instrução CALL
- instrução RET

4 - Interrupção

1 - Modos de Endereçamento do 8051

Os modos de endereçamento referem-se às diferentes formas (tipos de instruções) que o microprocessador oferece para definir e acessar dados.

- O microprocessador pode conter modos de endereçamento que facilitam o acesso à lista de dados .

- Dependendo do modo de endereçamento utilizado o programa pode ter maior ou menor número de bytes, sendo o processamento mais lento ou mais rápido, respectivamente

1 - Modos de Endereçamento

Os dados podem ser definidos nas instruções do mP

OU

Os dados podem estar armazenados nas seguintes áreas:

- Área de dados da EPROM
- Área de dados da RAM interna
- Área de dados da RAM externa
- Área de dados na Pilha

1 - Modos de Endereçamento do 8051
RAM interna

1.1 Imediato: o dado é definido na própria na própria instrução, sendo precedido por #.

Exemplos :

```
MOV    A, #dado8
MOV    A, #5FH ;   A = 5FH

MOV    DPTR, #dado16 ;
MOV    DPTR, #100BH ;   DPTR = 100BH  DPH = 10H  e DPL = 0Bh
```

1 - Modos de Endereçamento do 8051 RAM interna

1.1 Imediato

Após a execução da instrução como fica na Memória RAM interna:

Exemplos :

MOV A, #dado8

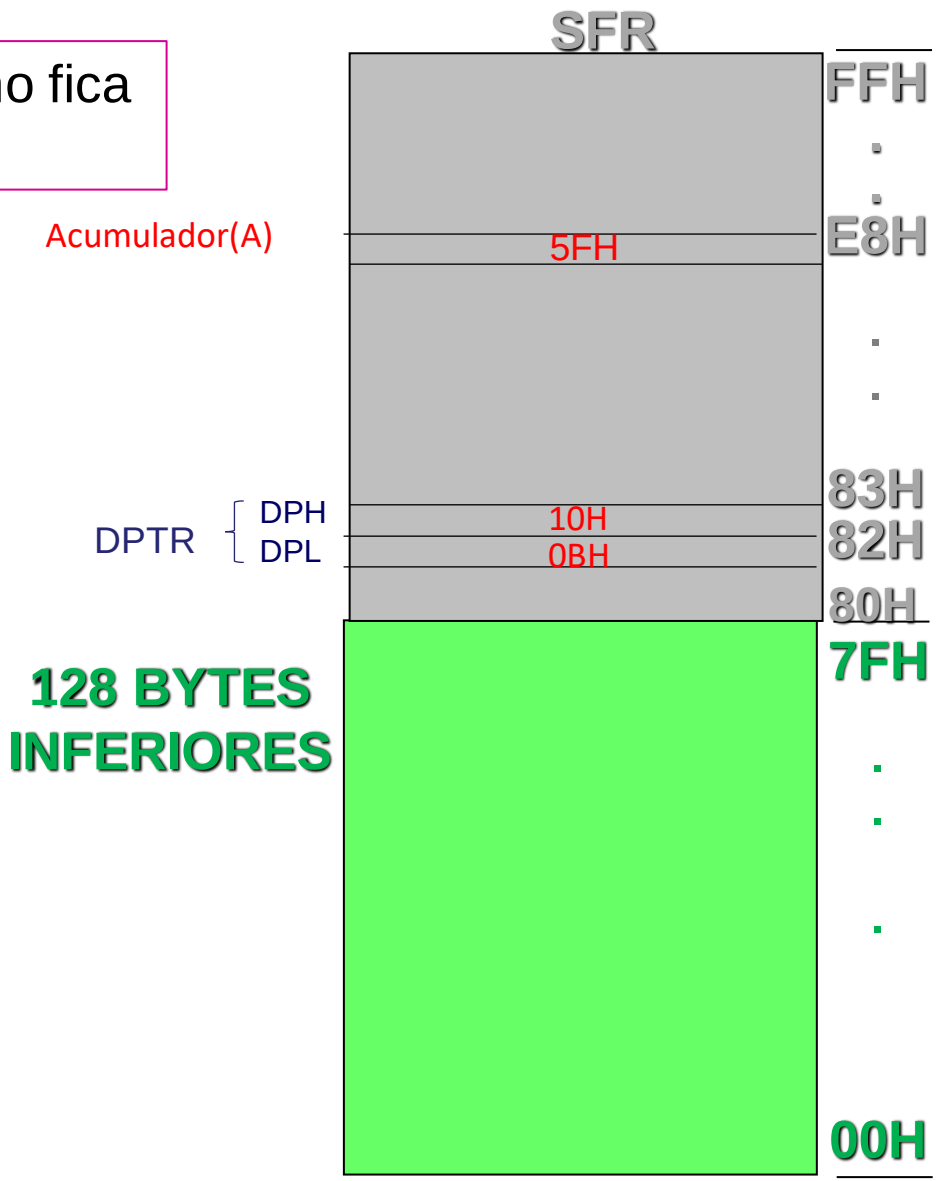
MOV A, #5FH ; A = 5FH

MOV DPTR, #dado16 ;

MOV DPTR, #100BH ;

DPTR = 100BH

DPH = 10 e DPL = 0Bh



1 - Modos de Endereçamento do 8051
RAM interna

1.2. Endereçamento por Registrador: A operação da instrução é realizada em um registrador específico.

Neste caso, não é necessário indicar um endereço.

Este tipo de instrução é codificada em 1 byte.

Exemplos:

MOV A, R3 ; uma cópia do conteúdo de R3 é movida para o A;

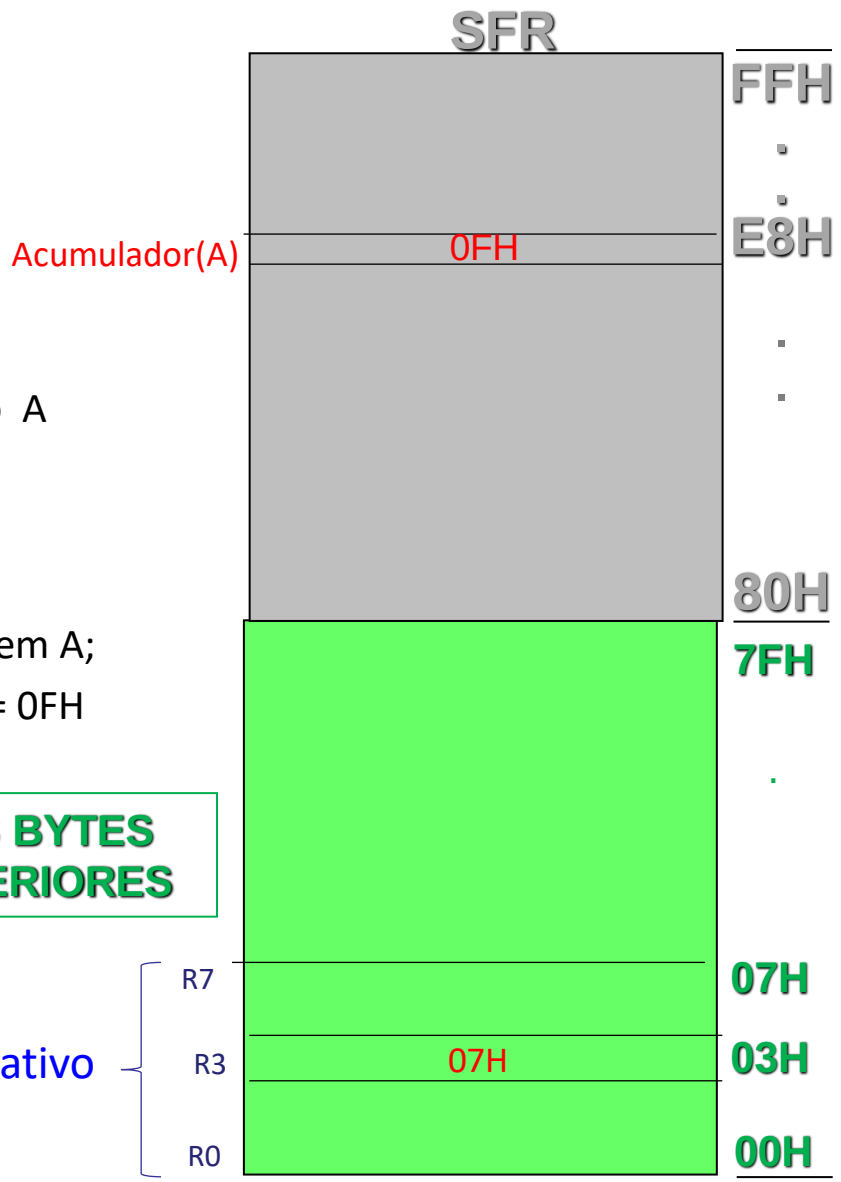
DEC R3 ; o conteúdo de R3 é decrementado R3 <- (R3)-1;

ADD A, R3 ; o conteúdo de R3 é somado com A, resultado em A;

1 - Modos de Endereçamento do 8051 RAM interna

1.2. Endereçamento por Registrador:

Supondo que antes da execução da instrução R3 = 16H e banco 0 ativo: como fica a Memória RAM interna após a execução das instruções:



Exemplos:

MOV A, R3 ; uma cópia do conteúdo de R3 é movida para o A
(supondo R3 = 08H , então A= 08H)

DEC R3 ; o conteúdo de R3 é decrementado
R3 <- (R3)-1; (então R3 = 07H)

ADD A, R3 ; o conteúdo de R3 é somado com A, resultado em A;
(então acumulador fica igual a (A)<- (A) + (R3) = 08h + 07H= 0FH

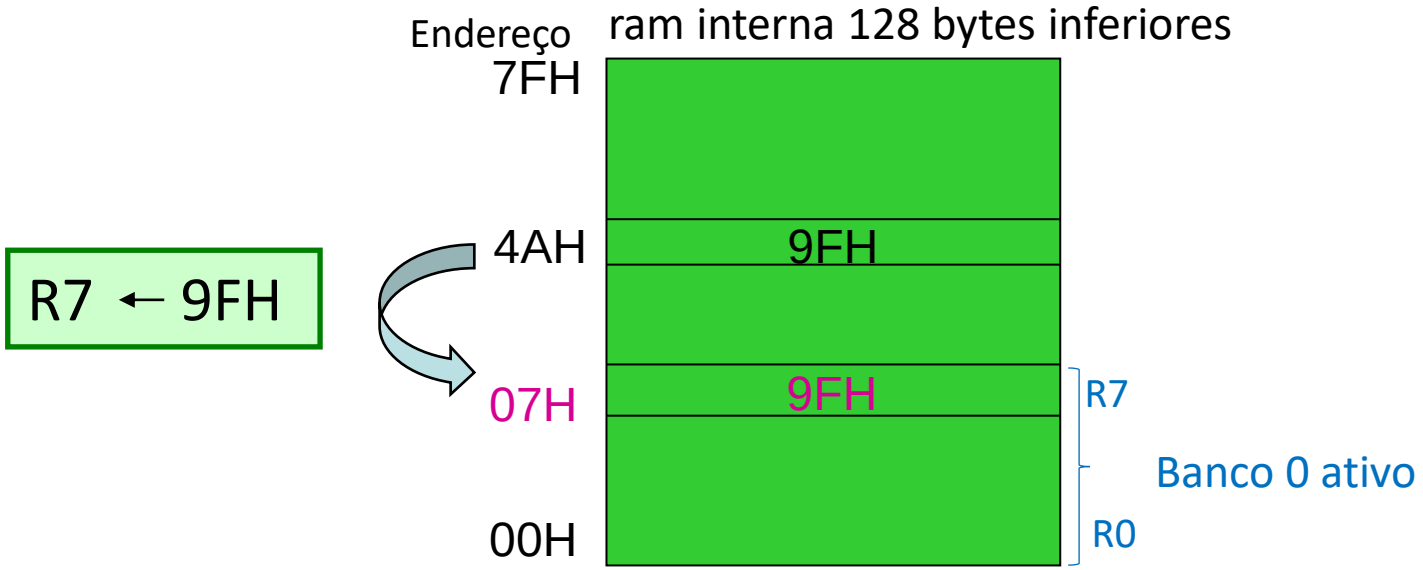
1 - Modos de Endereçamento do 8051

RAM interna

1.3. Endereçamento Direto: o dado é acessado através de seu endereço

Exemplo:

MOV R7, 4AH ; uma cópia do conteúdo do endereço 4AH da RAM interna, é armazenada em R7. E R7 está localizado na também na RAM interna e a posição depende de qual banco de registradores está ativo. Se for o banco 0, R7 se localiza na posição 07H.



1 - Modos de Endereçamento do 8051

RAM interna

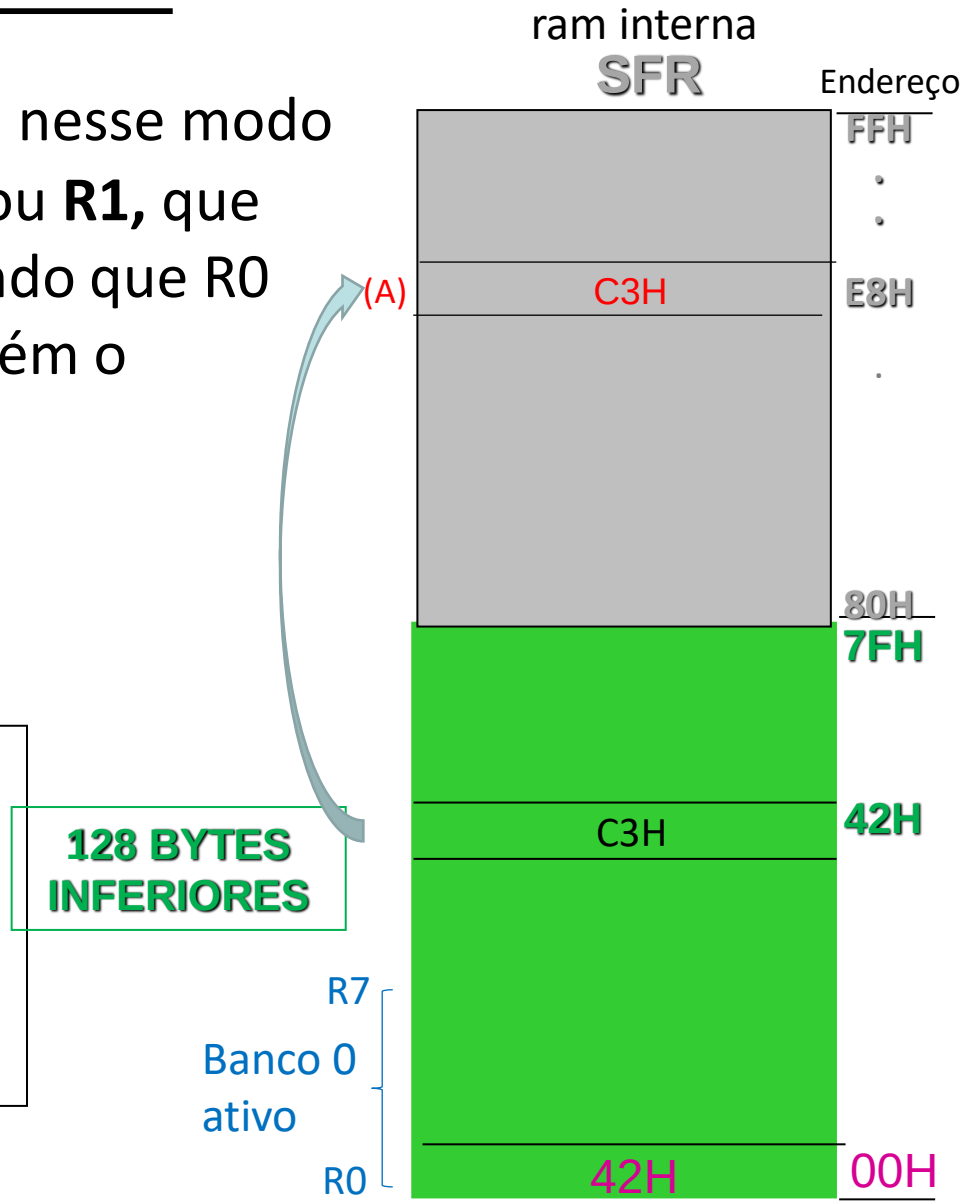
1.4. Endereçamento Indireto: nesse modo podem ser usados somente **R0** ou **R1,** que são precedidos por **@**, significando que R0 ou R1 são ponteiros, isto é, contém o endereço do dado.

Exemplo:

MOV A, @R0 ;

Supondo o banco 0 esteja ativo, se o conteúdo de (R0) = 42H, e nesse endereço está armazenado C3H, o acumulador recebe C3H

A ← C3H



1 - Modos de Endereçamento do 8051
Memória de Programa (EPROM)

1. 5. Endereçamento Indexado: o conteúdo do ponteiro DPTR é somado ao conteúdo do acumulador. O resultado é o valor do endereço na EEPROM que será acessado pela instrução

Instrução :

MOVC A,@A+DPTR

- Esta instrução acessa área de dados em EPROM
- DPTR é um ponteiro de 16 bits

Exemplo:

```
MOV    DPTR, #0F0BH
MOV    A, #02H
MOVC A,@A+DPTR
```

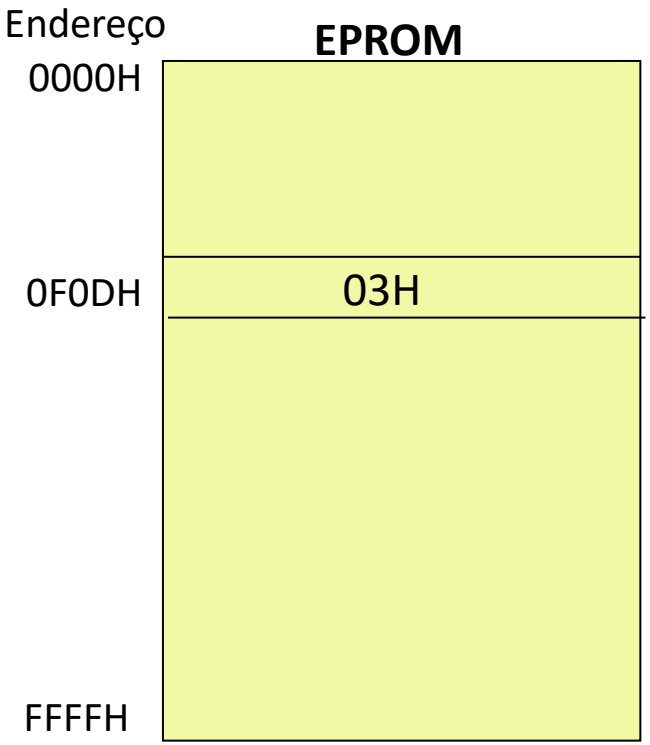
No exemplo o conteúdo do DPTR será somado ao conteúdo de A:
0F0BH + 02H = 0F0DH
O endereço resultante 0F0DH será acessado na EPROM, o seu conteúdo será lido e armazenado em A.

1 - Modos de Endereçamento do 8051
Memória de Programa (EPROM)

1. 5. Endereçamento Indexado:

Instrução :

MOVC A,@A+DPTR



Exemplo:

```
MOV     DPTR, #0F0BH
MOV     A, #02H
MOVC A,@A+DPTR
```

No exemplo o conteúdo do DPTR será somado ao conteúdo de A:
0F0BH + 02H = 0F0DH
O endereço resultante 0F0DH será acessado na EPROM, o seu conteúdo, supondo que seja 03H será lido e armazenado em A.
(A) = 03H

1 - Modos de Endereçamento do 8051
Memória RAM externa

1.6. Endereçamento Indireto com ponteiro de 16 bits (DPTR):

o conteúdo do ponteiro DPTR é o endereço que será acessado na RAM externa para leitura ou gravação.

Instruções :

MOVX A,@DPTR; leitura da RAM externa

MOVX @DPTR,A ; gravação na RAM externa

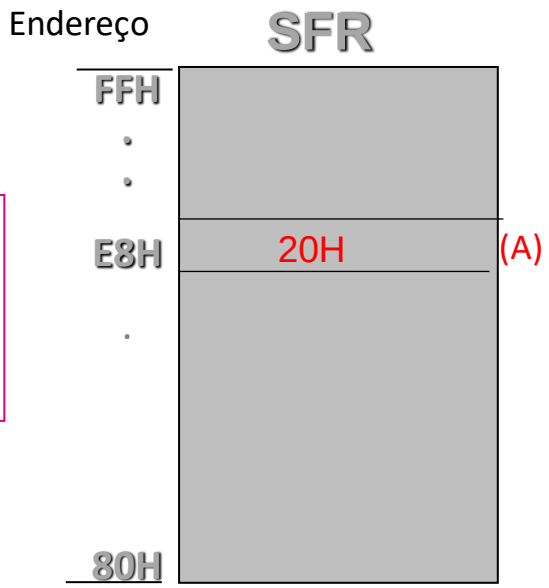
1 - Modos de Endereçamento do 8051
Memória RAM externa

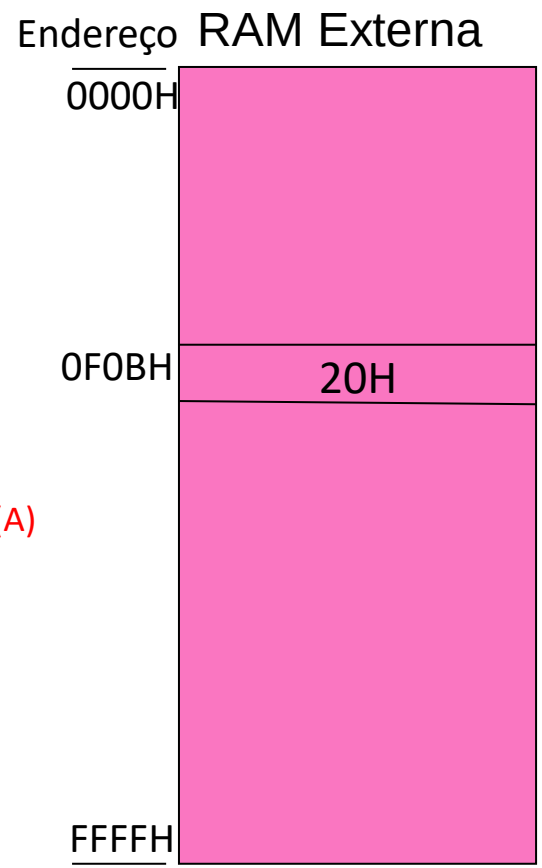
1.6. Endereçamento Indireto com ponteiro de 16 bits (DPTR): o conteúdo do ponteiro DPTR é o endereço que será acessado na RAM externa para leitura ou gravação.

Exemplo : gravação na RAM externa

```
MOV     DPTR, #0F0BH
MOV A, #20H; armazena o valor 20H em A
MOVX   @DPTR, A;  grava o conteúdo de A
                        ;na  posição 0F0BH da
                        ; RAM externa
```

No exemplo. Após a execução das instruções o valor do acumulador (20H) é gravado no endereço 0F0BH da RAM externa, endereço esse que está contido no DPTR





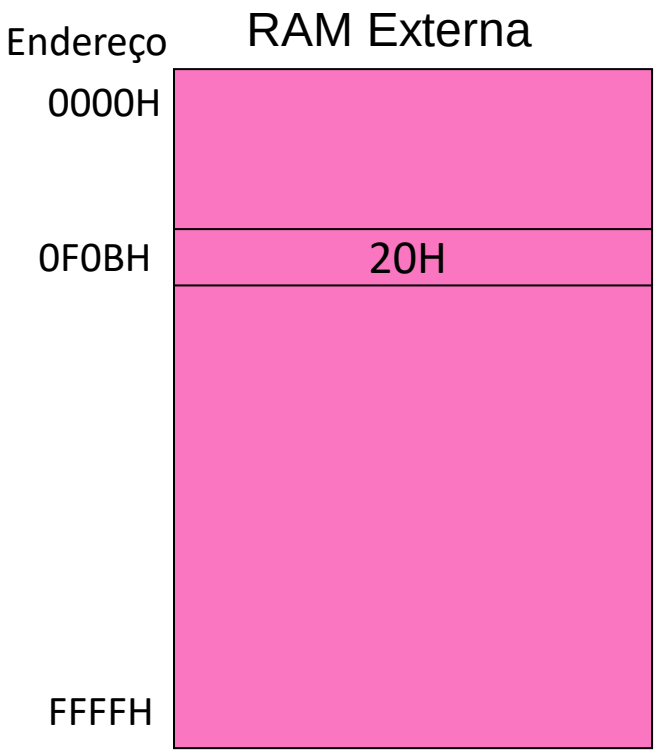
1 - Modos de Endereçamento do 8051
Memória RAM externa

1.6. Endereçamento Indireto com ponteiro de 16 bits (DPTR): o conteúdo do ponteiro DPTR é o endereço que será acessado na RAM externa para leitura ou gravação.

Exemplo: Leitura na RAM externa

```
MOV    DPTR, #0F0BH
MOVX   A,@DPTR;  lê o conteúdo da posição 0F0BH
                 ;da RAM externa e armazena no
                  ;acumulador.
```

No exemplo o conteúdo do endereço 0F0BH da RAM externa é 20H, então ele será copiado para o acumulador ((A)←20H)



2 – PILHA NO 8051

Pilha é uma área da memória RAM INTERNA, onde podem ser lidos ou gravados dados, sob o controle do **ponteiro SP**.

Característica da pilha no microprocessador 8051

É usada para :

- armazenamento de dados de 8 bits, com instruções de pilha (PUSH ou POP)
- guardar um endereço quando é executada uma instrução de chama da de subrotina: instrução CALL
- guardar um endereço quando uma interrupção é atendida

❖ O ponteiro **SP** é de 8 bits : é iniciado com o valor 07H da RAM interna ao se fazer "reset" no microcontrolador

❖ O ponteiro **SP** é incrementado antes de um dado ser armazenado na pilha, portanto a pilha inicia no endereço seguinte (08H)

2 – PILHA no 8051

Instruções de pilha:

```
PUSH    iram ;
POP     iram ;
```

OBS: o endereço iram(8), refere-se a um endereço (8bits) da RAM interna

2 – PILHA NO 8051

2.1 Instrução PUSH iram:

- Incrementa o SP e guarda o conteúdo do endereço iram na pilha

PUSH iram

1 . SP ⟵ SP +1

2 . (SP)⟵ (iram)

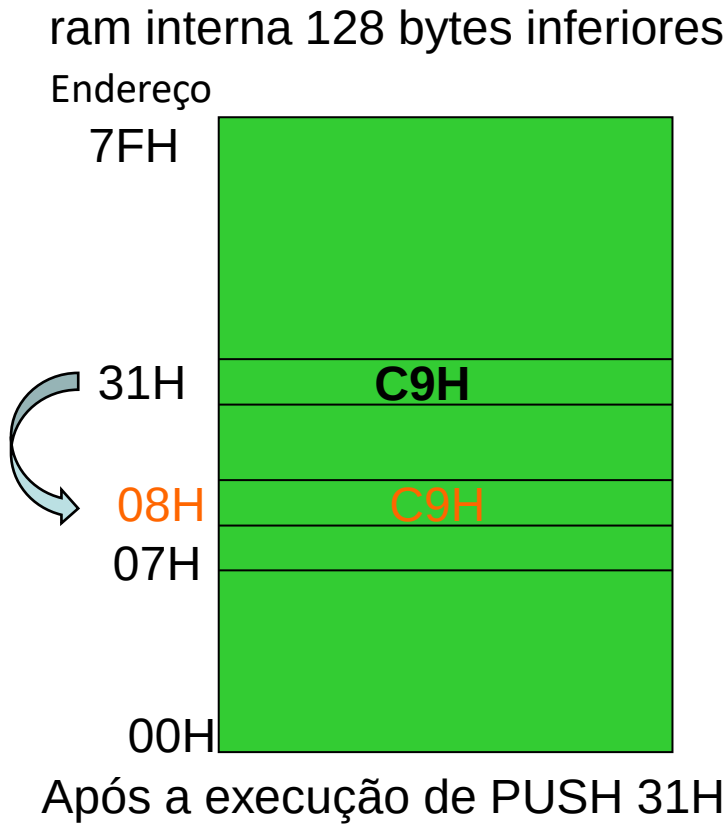
Exemplo: PUSH 31H

Valor inicial de SP = 07H

Na execução: SP = 07 + 01= 08H

(08H)⟵ (31H);

o valor do conteúdo da posição 31H é armazenado na posição cujo SP aponta (pilha), portanto conteúdo da pilha é C9H. Com isso, a posição 31H fica livre para ser utilizada para outra operação



2 – PILHA NO 8051

2.2 Instrução POP iram:

- Armazena o dado apontado por SP, no endereço **iram**
- em seguida decrementa o ponteiro SP

POP iram
(iram) ← (SP)
SP ← SP - 1

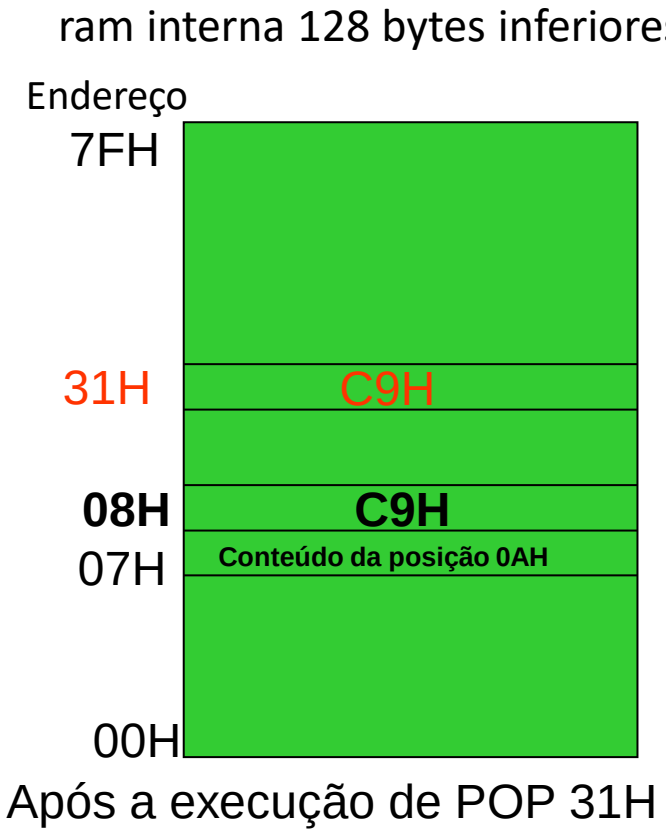
Exemplo: POP 31H

Valor inicial de SP = 08H

Na execução: (31H) ← (SP)

SP = 08 - 01= 07H

o conteúda original da posição 31H é recuperado



3 – Instruções que usam Pilha

3.1 Instrução CALL

Na execução da instrução CALL:

- o endereço da instrução seguinte (contido no ponteiro PC) é armazenado na Pilha
- o ponteiro de programa PC, é carregado com o endereço especificado na instrução CALL

CALL endereço da subrotina

((SP) + 1) ← (PCL)

((SP) +2) ← (PCH)

(SP) ← (SP) + 2

(PC) ← endereço da subrotina

3 – Instruções que usam Pilha

3.1 Instrução CALL

Exemplo: valor inicial de SP = 1FH

Endereço	instrução
07FEH	**LCALL 100BH**
	MOV B,A
	.....

A instrução LCALL é de 3 bytes e ocupa 3 endereços: 07FEH, 07FFH E 0800H

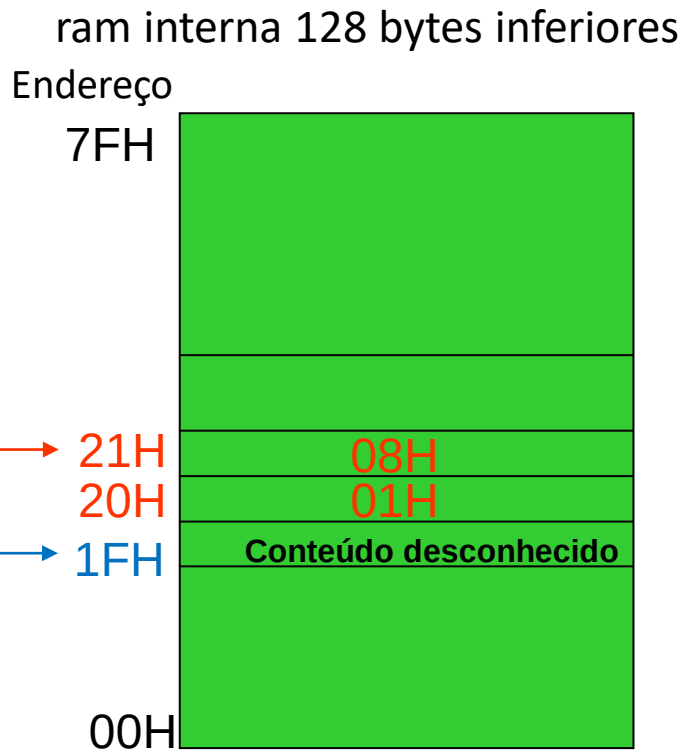
PORTANTO o endereço da instrução MOV B, A é **0801H.**

Ações executadas:

1 . O endereço 0801H é armazenado na pilha

2 . O PONTEIRO DE PROGRAMA PC é carregado com o endereço 100Bh

3. O valor final de SP = 21H



3 – Instruções que usam Pilha

3.1 Instrução RET

Na execução da instrução RET:

- o endereço armazenado na Pilha, que é o endereço da instrução seguinte à instrução CALL, é recuperado para ponteiro o PC;
- o ponteiro SP é decrementado 2 vezes, voltando a apontar para a mesma posição anterior à execução da instrução CALL;
- O PC contendo o endereço da intrução segunte à instrução CALL, o programa principal continua a ser executado desta a partir desta posição.

RET

(PCH) ← (SP)

(PCL) ← ((SP) - 1)

(SP) ← ((SP) - 2)

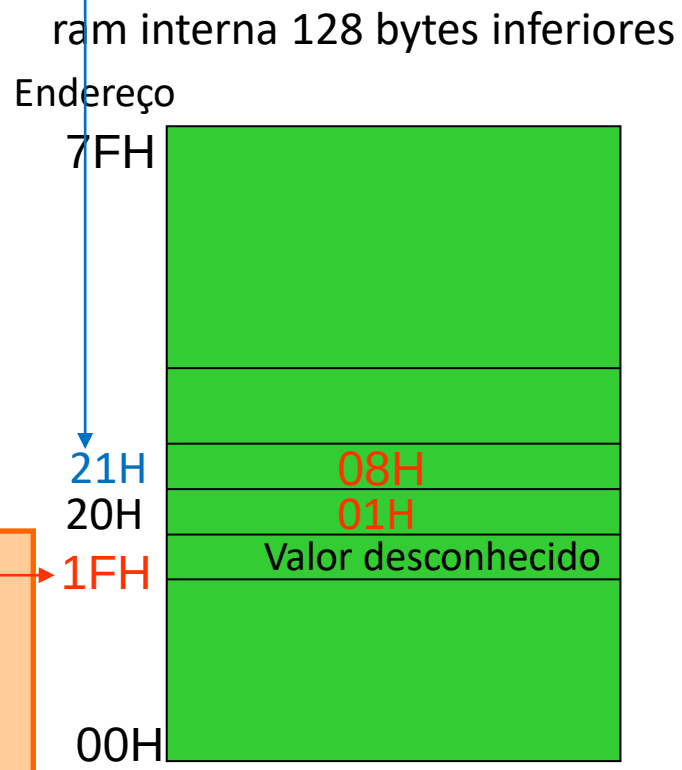
(PC) ← endereço da instrução seguinte à instrução CALL

3 – Instruções que usam Pilha

3.2 Instrução RET

EXEMPLO: valor inicial do SP =21H

```
Endereço   instrução
O7FEH      LCALL  conv
           MOV    B,A
              ...

           ORG   100BH ;  conv= 100BH subrotina
conv:      SUBB  30H ;
           RET
```



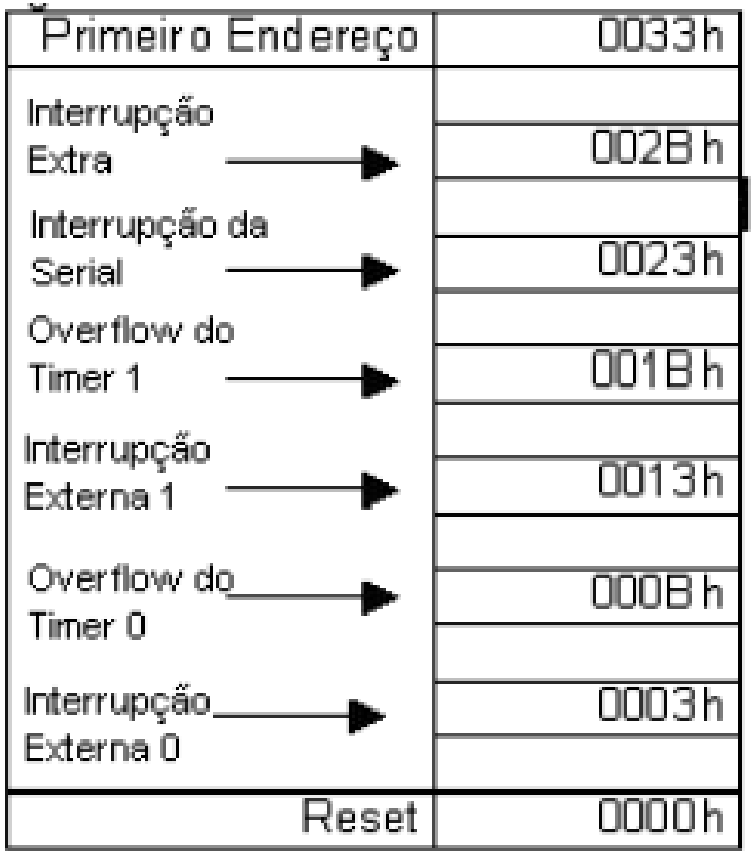
Ações executadas:

1 . O ponteiro de programa (PC) recebe da pilha o valor 0801H e volta a executar a partir deste endereço.

2. SP é decrementado de duas unidades, SP = 1FH

4 – Interrupção

Na condição de reset o ponteiro PC é iniciado é carregado com o endereço zero da memória de programa. Isso ocorre devido à localização dos vetores de interrupção. Os vetores de interrupção ocupam posições fixas na memória de programa, e o reset é considerado uma “interrupção” estabelecida no endereço zero.



4 – Interrupção

Alguns periféricos podem gerar interrupções e para isso alguns endereços são reservados:

- IE0: 0003H, Interrupção externa 0;
- TF0: 000BH, Interrupção por overflow do timer 0;
- IE1: 0013H, Interrupção externa 1;
- TF1: 001B, Interrupção por overflow do timer 1;
- RI e TI: 0023, Interrupção do canal serial.

Se o programador não alterar a Prioridade entre as interrupções a sequencia de atendimento segue :

primeiro IE0, em seguida TF0, IE1, TF1, RI e por ultimo, TI.

4 – Interrupção

Se a interrupção não é utilizada, seu endereço é considerado de propósito geral. Além disso, os vetores são espaçados em 8 bytes. Portanto, se a rotina de interrupção é curta, essa região de 8 bytes pode ser utilizada, já as rotinas longas podem utilizar instruções de JUMP, e alocar a subrotina de interrupção em outra região da memória de programa.

Exemplo:

```
        ORG 0003H
        SJUMP  SUBINT; instrução de salto para o endereço da subrotina de interrupção
        ORG 0050H
SUBINT: CLR A;
          .
          .
          .
        RETI
```

4 – Interrupção

As interrupções são habilitadas e configuradas por meio de 2 registradores endereçáveis bit a bit: IE e IP os quais se localizam, respectivamente, nas posições A8H e B8H da RAM interna dos SFRs

O registrador IE configura as Interrupções:

IE.7	IE.6	IE.5	IE.4	IE.3	IE.2	IE.1	IE.0
EA	—	ET2	ES	ET1	EX1	ET0	EX0

O registrador IP define o nível de prioridade das Interrupções

IP.7	IP.6	IP.5	IP.4	IP.3	IP.2	IP.1	IP.0
—	—	PT2	PS	PT1	PX1	PT0	PX0

Configuração da Interrupção

O registrador IE habilita as Interrupções

IE.7	IE.6	IE.5	IE.4	IE.3	IE.2	IE.1	IE.0
EA	—	ET2	ES	ET1	EX1	ET0	EX0

- EA bit IE.7 : Se EA=0 todas as interrupções são desabilitadas e nenhuma interrupção será reconhecida;
  Se EA=1 cada interrupção é individualmente habilitada ou desabilitada setando ou limpando seu bit de habilitação;
- — bit IE.6: Nenhum uso;
- ET2 bit IE.5: habilita ou desabilita *overflow* do Timer 2 (ou interrupção para o 8052);
- ES bit IE.4: habilita (em 1) ou desabilita a interrupção da porta serial;
- ET1 bit IE.3: habilita(em 1) ou desabilita interrupção de *overflow* do Timer 1;
- EX1 bit IE.2: habilita (em 1) ou desabilita Interrupção externa 1;
- ET0 bit IE.1: habilita (em 1) ou desabilita interrupção de *overflow* do Timer 0;
- EX0 bit IE.0: habilita (em 1) ou desabilita Interrupção externa 0;

Configuração da Interrupção

O registrador IP define o nível de prioridade das Interrupções

IP.7	IP.6	IP.5	IP.4	IP.3	IP.2	IP.1	IP.0
—	—	PT2	PS	PT1	PX1	PT0	PX0

- — bit IE.7 e IE6: Nenhum uso;
- PT2 bit IE.5: define o nível de prioridade da interrupção do Timer 2 (somente para o 8052);
- PS bit IE.4: define o nível de prioridade da interrupção da porta serial;
- PT1 bit IE.3: define o nível de prioridade da interrupção do Timer 1;
- PX1 bit IE.2: define o nível de prioridade da Interrupção externa 1;
- PT0 bit IE.1: define o nível de prioridade da interrupção do Timer 0;
- PX0 bit IE.0: define o nível de prioridade da Interrupção externa 0;

4 – Interrupção

Para utilizar uma interrupção deve-se:

1. Escrever a rotina de interrupção com início no endereço do vetor correspondente. Isso é feito a partir da diretiva:

   ORG <endereço da interrupção>;

2. O bit correspondente a interrupção desejada deve ser habilitado no registrador IE;
3. O bit EA (Enable All) deve ser ativado no registrador IE;
4. O registrador IP deve ser configurado estabelecendo prioridades nas interrupções habilitadas ou escolhendo baixa prioridade;
5. Caso a interrupção seja disparada com sinal de descida ao invés de ser a nível, o registrador TCON deve ser configurado

Configuração da Interrupção

O registrador TCON define se o sinal que dispara a interrupção deve ser a nível baixo ou borda de descida

TCON.7	TCON.6	TCON.5	TCON.4	TCON.3	TCON.2	TCON.1	TCON.0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

- – IE0 e IE1: indicam a ocorrência de um evento de interrupção externa nos pino P3.2 e P3.3 respectivamente. São resetados por hw quando a interrupção é atendida;;
- IT0 e IT1: definem o tipo de sinal que gera a interrupção externa:
  - ✓ [0] gera interrupção quando um sinal de nível baixo está presente no pino;
  - ✓ [1] gera interrupção quando ocorre uma borda de descida no pino.

4 – Interrupção

O que acontece quando ocorre mais uma solicitação de interrupção simultaneamente?

no caso de interrupções com a mesma prioridade a seguinte ordem de avaliação é estabelecida:

primeiro IE0, em seguida TF0, IE1, TF1, RI e por ultimo, TI.

O que acontece quando uma interrupção é disparada?

O microcontrolador finaliza a execução da instrução do programa principal na qual ocorreu a solicitação de interrupção, armazena na pilha o endereço da instrução seguinte, e carrega o ponteiro PC com o endereço da subrotina de atendimento da interrupção.

4 – Interrupção

A solicitação de interrupção ocorre em qualquer momento da execução do programa, quando qualquer instrução esteja sendo executada, a partir da habilitação da interrupção. Não tem como prever quando vai ocorrer. Então, supondo que ocorreu uma solicitação de interrupção quando estava sendo executada a instrução MOV DPTR , #20FCH, que é uma instrução de 3 bytes.

Dados:

SP = 07H

endereço da subrotina de interrupção: 0003H

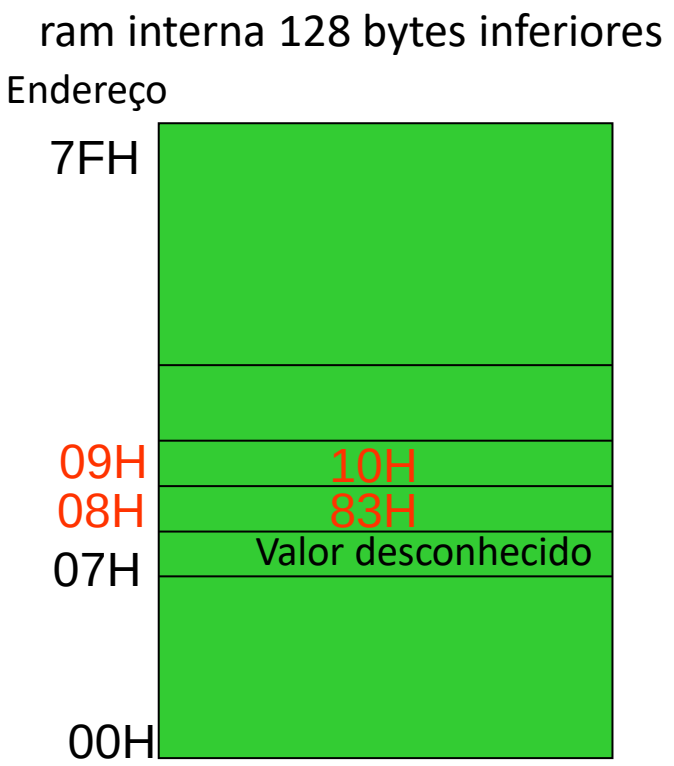
Endereço	instrução (programa principal)
1080H	MOV DPTR , #20FCH
	MOVC A,@A + DPTR

Ações executadas pelo microprocessador ao atender a interrupção:

1- conclui a execução da instrução corrente: MOV DPTR , #20FCH que ocupa 3 bytes (1080H, 1081H e 1082H)

2 - salva na pilha o endereço da instrução seguinte (1083H): MOVC A,@A + DPTR

3 – carrega o PC com o endereço 0003H, da interrupção



4 – Interrupção

A subrotina de interrupção deve ser finalizada com a instrução RETI, que tem o mesmo efeito da instrução RET já explicada anteriormente.

RET I

(PCH) ⟵ (SP)
(PCL) ⟵ ((SP) - 1)
(SP) ⟵ ((SP) - 2)
(PC) ⟵ endereço da instrução seguinte à instrução onde ocorreu a interrupção

4 – Interrupção

Execução da Instrução RETI

Exemplo:

programa principal

```
Endereço    instrução
            ORG 107FH
            CLR A
1080H       MOV DPTR , #20FCH
            MOVC   A,@A + DPTR
                   :
```

Instrução sendo executada quando ocorreu a interrupção

Próxima instrução (que se encontra na posição 1083H) a ser executada no programa principal, após o atendimento da subrotina de interrupção, ou seja, após a execução da instrução RET I da sub rotina.

Subrotina de Interrupção

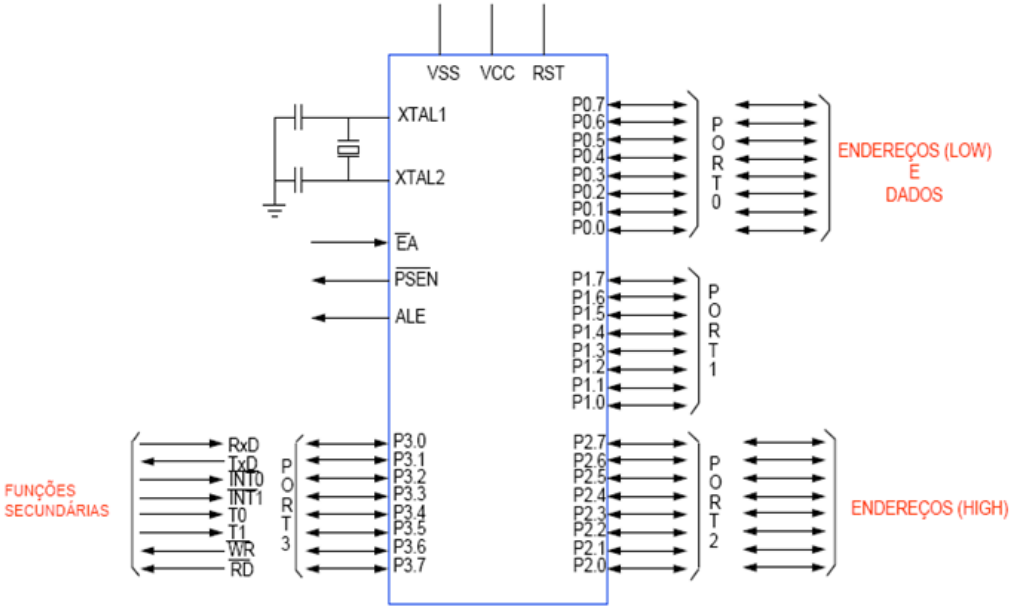
```
Endereço   instrução
           ORG 0003H
sub:       ADD   A, #20H ; Subrotina de interrupção
           RET I
```

Programa utilizando Interrupção

O programa utiliza a interrupção externa INTO com baixa prioridade e sensível à borda de descida. A interrupção é disparada quando um botão, ligado ao pino P3.2 (INT0), é acionado aplicando uma borda de descida ao mesmo.
O atendimento da interrupção faz com que uma subrotina seja executada a qual realiza o complemento do valor da porta P0.

No programa principal, a porta P0 deve ser configurada como saída (e, para isso, deve ser inicialmente enviado à ela, o valor 00h).

Na porta P0 estão ligados 8 LEDs na configuração anodo comum, portanto acendem com nível baixo, para possibilitar a visualização do valor enviado à porta P0.

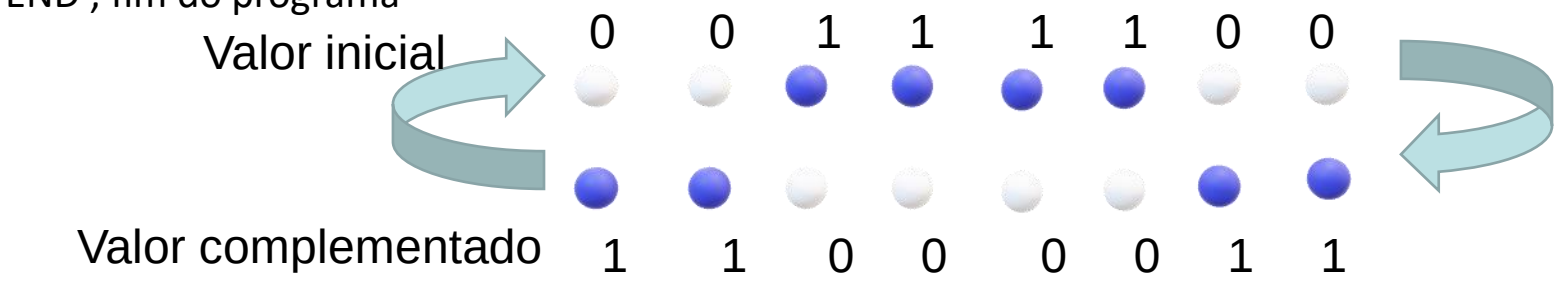


Programa utilizando Interrupção

```
        ORG 0000H ; endereço da interrupção reset
        ajump  inicio  ; o PC inicializa com o valor 0000H então vem na posição 0000h
                       ;buscar a  1ª. Instrução
                        ; essa instrução ajump carrega no PC o endereço inicio, onde está o
                        ;programa principal

        ORG 0003H  ; endereço da interrupção INT0
        cpl a              ;complementa o valor do acumulador
        mov P0,a      ; move conteúdo do acumulador para a porta P0
        RETI              ;recupera o endereço da instrução do programa principal para o PC
                          ; inicio do programa principal
inicio: mov a,#00H ; armazena a constante 00h no acumulador
        mov P0,a      ; configura a porta P0 como saída enviando o valor 00h para ela
        mov a,#FFH ; armazena a constante FFh no acumulador
        mov P0,a    ; inicializa a porta P0 deixando apagados os LEDS
        mov a, 10000001b  ; move a constante 10000001b para o acumulador para configurar
                                ;o registrador   IE
```

Programa utilizando Interrupção (cont.)

```
mov ie,a   ; move o valor 10000001b para ie habilitando as interrupções globais (EA=1) e a
           ;interrupção INT0 que é habilitada colocando o bit IE.0 em 1

mov a,00000000b  ; move a constante 00000000b para o acumulador para configurar
           ;o registrador  IP

mov ip,a       ; move o valor 00000000b para ip para configurar a interrupção INTO com
           ;baixa prioridade

mov a,00000001b  ; move a constante 00000001b para o acumulador para configurar o
           ;registrador  TCON

mov tcon,a ; configura INT0 como sensível á bora de descida colocando o bit TCON.0 em 1

valor: mov a,#3CH ; 3CH (0011 1100)b é o  valor que a porta P0 será inicializada antes da subrotina
           ; ser executada
SJUMP $        ; loop infinito nesse endereço até ocorrer uma interrupção( quando uma
           ;chave for acionada na entrada P3.2),
SJUMP  valor  ; instrução executada após retorno da subrotina
END ; fim do programa
```



FIM