Microcontroladores 8051

1. Microcontroladores

Microcontrolador é o nome dado ao componente que incorpora em um só "chip" todos os elementos necessários a um microcomputador . Deve ter :

CPU , Memória e Interfaces

As interfaces podem ser as mais diversas :

- Contador / temporizador
- conversor AD / DA
- I/O paralela
- Interface Serial, etc...

Além disso deve permitir a expansão externa de memória e periféricos .

2. Estudo da família MCS-51 de microcontroladores

Características do Núcleo (Core)

- CPU de 8 bits otimizado para aplicações de controle
- Capacidade de processamento booleano (lógica de um único bit)
- 64 Kbytes de espaço de memória de programa
- 64 Kbytes de espaço de memória de dados
- 4 Kbytes de espaço de memória de programa "on chip"
- 128 bytes de memória RAM de dados "on chip"
- 32 linhas de I/O bidirecionais endereçadas individualmente
- 2 Contadores / Temporizadores de 16 bits cada
- UART full duplex
- Estrutura de interrupção com níveis de prioridade
- oscilador "on chip "

2.1 Diagrama de Blocos do "Core "do 8051 básico

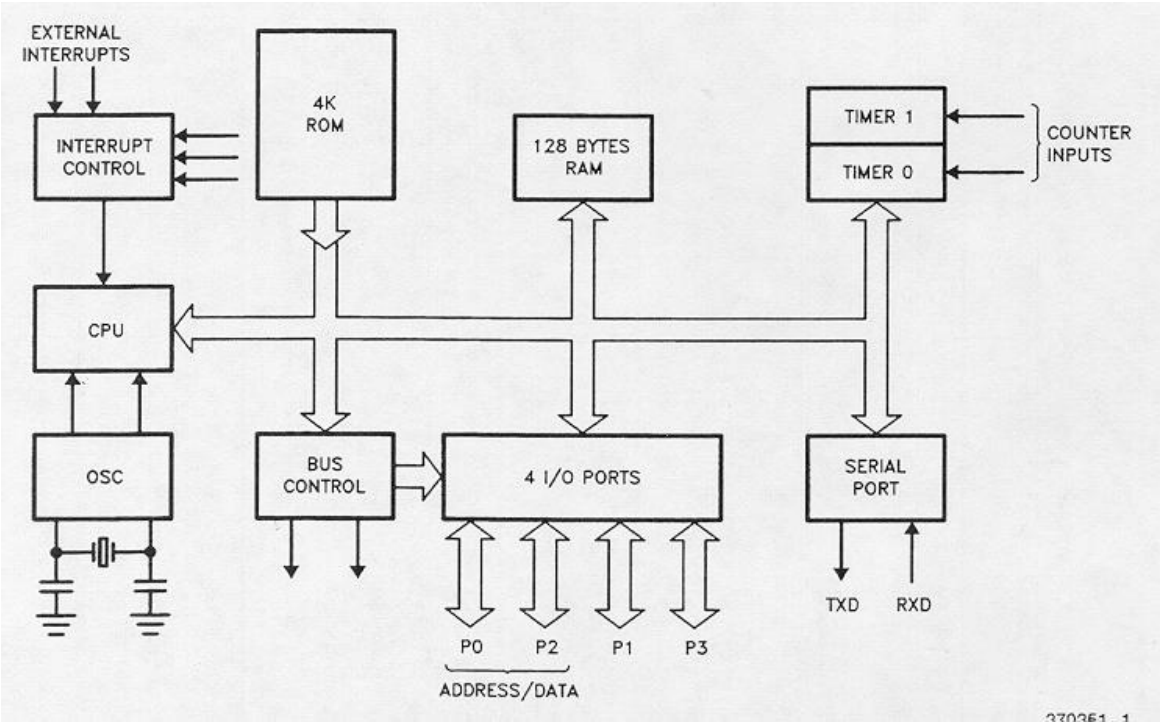
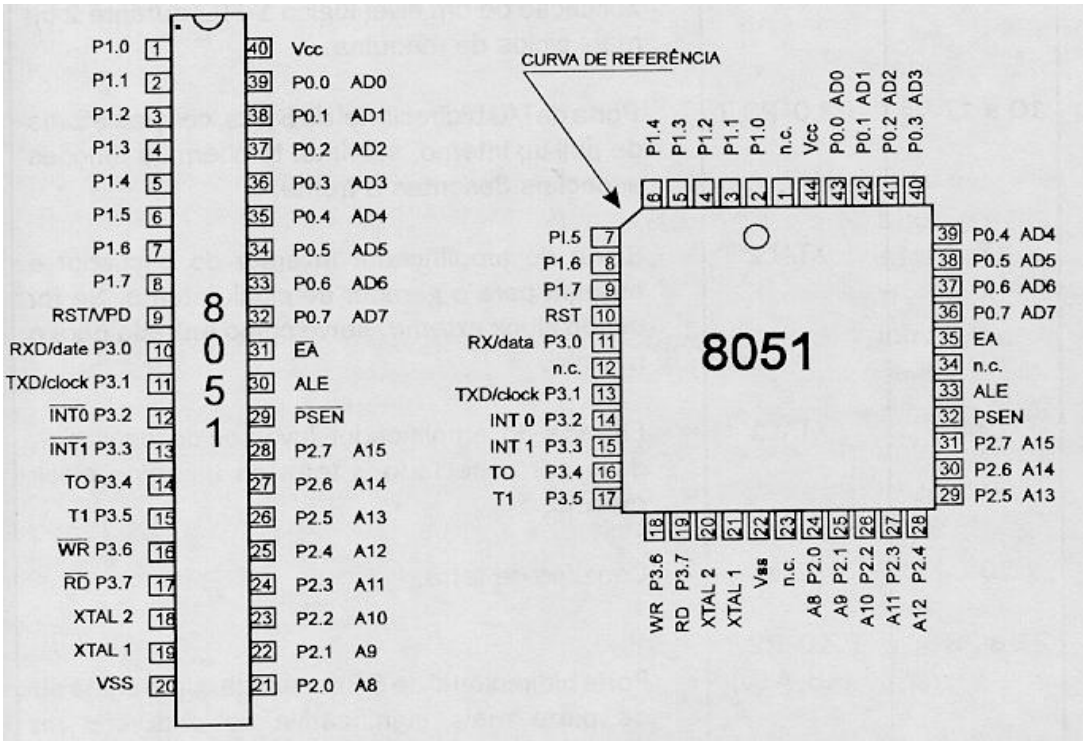


Diagrama de pinos do 8051



Descrição dos pinos (DIP)

1 a 8	P1.0 a P1.7	Porta de I/O bidirecional
9	rst/vpd	Pino de Reset (ativado em nível lógico 1 por 2 ciclos de clock)
10 a 17	P3.0 a P3.7	Porta de I/O bidirecional
18 e 19	Xtal1e Xtal2	Cristal do oscilador interno
20	Vss	Terra
21 a 28	P2.0 a P2.7	Porta de I/O bidirecional
29	$\overline{PSEN}$	Habilitação de programa externo (faz o RD da EPROM externa)
30	ALE	Latch de endereços
31	$\overline{EA}$	Seleção da memória de programa 0=externa 1=interna
32 a 39	P0.0 a P0.7	Porta de I/O bidirecional
40	Vcc	5 V

Os pinos de P3 também são definidos com funções especiais :

P3.0 RxD Receptor de porta serial

P3.1 TxD transmissor de porta serial

P3.2 $\overline{Int0}$ Interrupção 0

P3.3 $\overline{Int1}$ Interrupção 1

P3.4 T0 Timer/ Counter 0

P3.5 T1 Timer/ Counter 1

P3.6 $\overline{WR}$ Escrita na memória externa

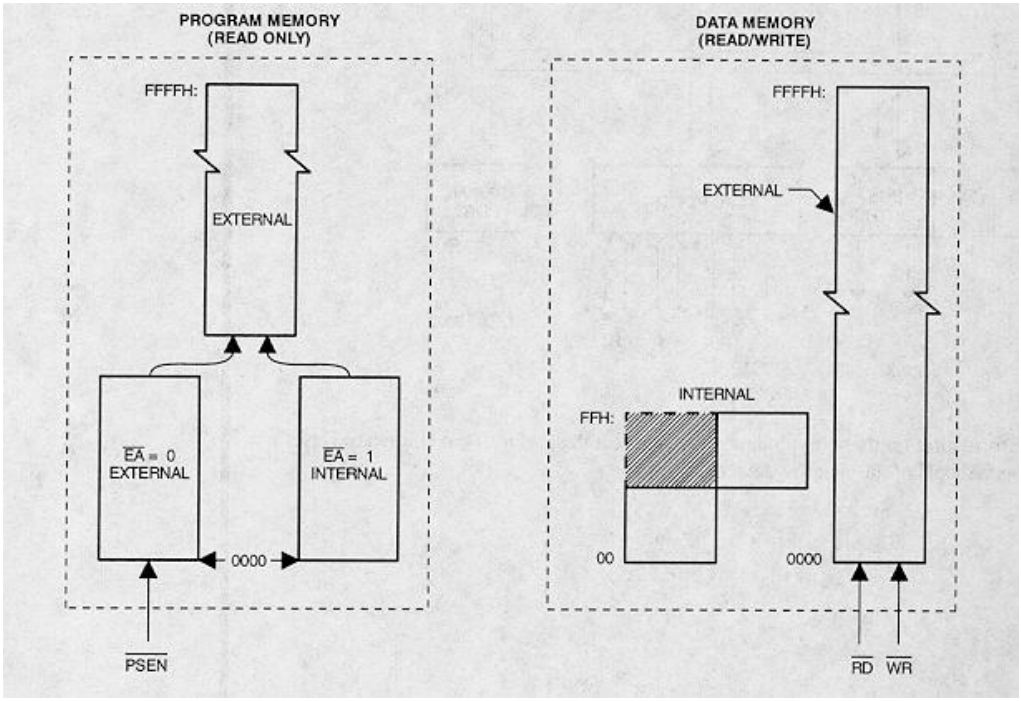
P3.7 $\overline{RD}$ Leitura na memória externa

Os pinos de P0.0 a P0.7 podem também ser usados como Barramento de Dados (D0 a D7) ou como os 8 Bits menos significativos do Barramento de Endereços (A0 a A7) multiplexados, ou seja, os pinos anotados como AD0 a AD7 ora podem trafegar dados e ora podem trafegar endereços. É preciso utilizar lógica externa para demultiplexar o Barramento de Dados do de Endereços.

Os pinos P2.0 a P2.7 podem ser usados como a parte mais significativa do Barramento de Endereços (A8 a A15) que juntamente com a parte menos significativa completam o número de linhas necessárias para endereçar 64Kbytes, ou seja, 16 linhas.

2.2 Organização da memória da família MCS-51

- **Memórias de dados e de programas separadas.**

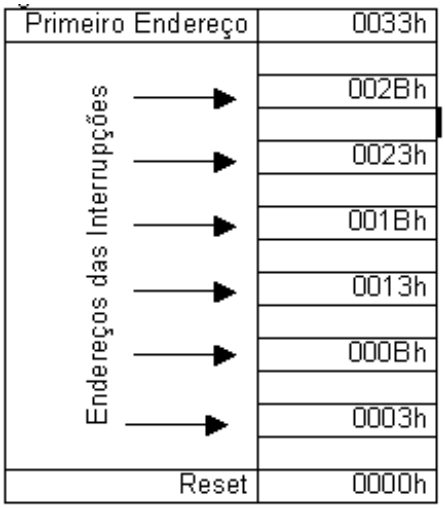


2.2.1 Memória de programa

Após um Reset a CPU começa a executar o programa do endereço 0000

- **Endereço das interrupções:**

  Cada interrupção causa um salto para um endereço fixo na memória de programa (ROM , EPROM ,...) a partir do endereço 0003 . Existem 8 bytes de intervalo entre dois endereços consecutivos, para que possa ser colocada uma rotina de atendimento de interrupção de um JUMP caso não haja espaço suficiente . Se as interrupções não forem usadas os endereços podem ser usados para o programa geral.



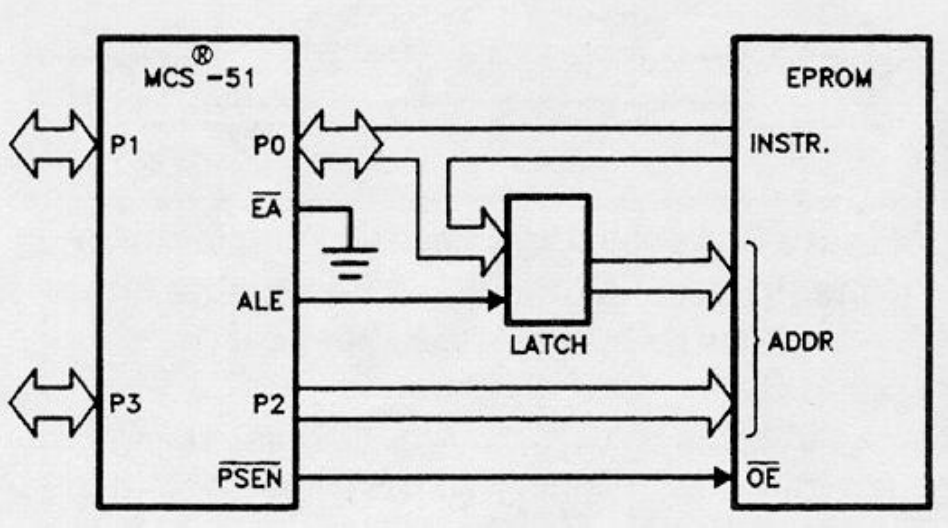
- **Endereços das memórias de programa interna e externa :**

Rom Interna EA = Vcc	Endereçamento Interno	Endereçamento Externo
4 K	0000h a 0FFFh	1000h a FFFFh
8 K	0000h a 1FFFh	2000h a FFFFh
16 K	0000h a 3FFFh	4000h a FFFFh
32 K	0000h a 7FFFh	8000h a FFFFh

Se EA = Gnd toda a memória de programa é externa : 0000h a FFFFh

- **Mapeamento de memória de programa externa :**

a. mapeamento completo (64 Kb externo)



b. ROM interna + ROM externa

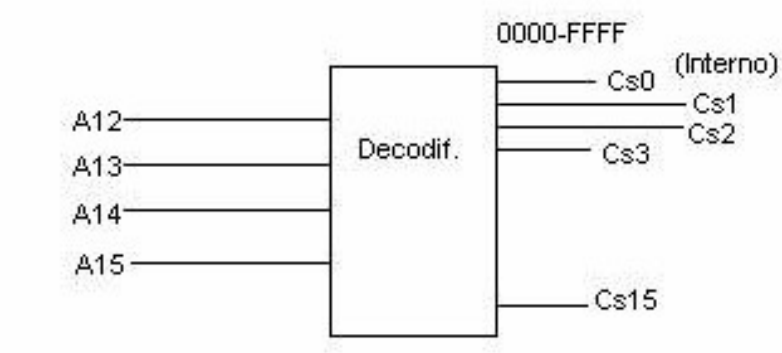
<u>Exercício</u> :

Com o microcontrolador 87C51 desenvolver o mapeamento externo de memória de programa tal que seja também usado o endereçamento interno .
(considerar bloco interno = bloco externo)

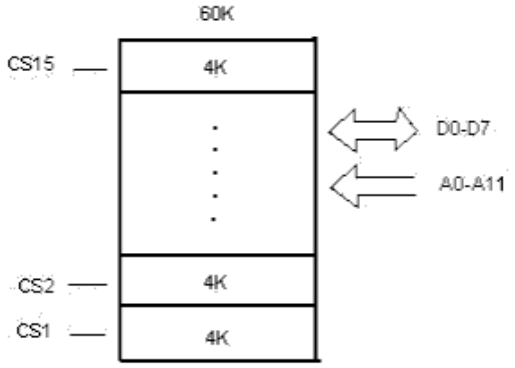
<u>Solução :</u>

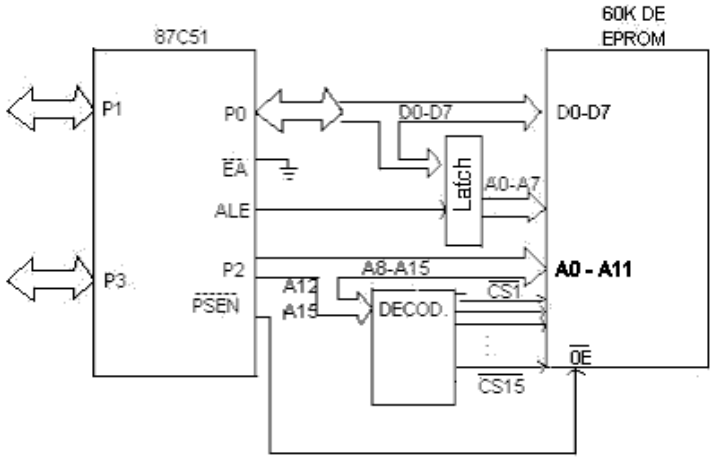
87C51 _____ 4 Kb de EPROM Interna ___ endereçamento externo = 1000h a FFFFh

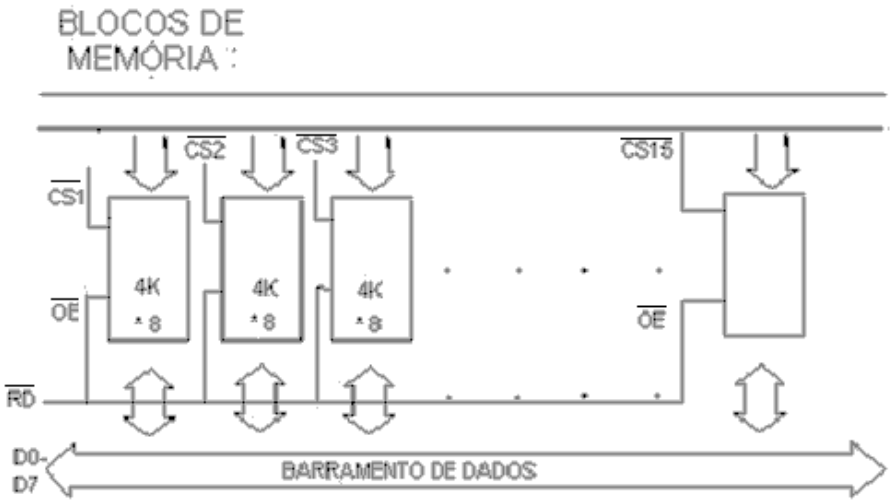
- **Decodificador de endereços**



- **Blocos de memória externa**







2.2.2 Memória de Dados

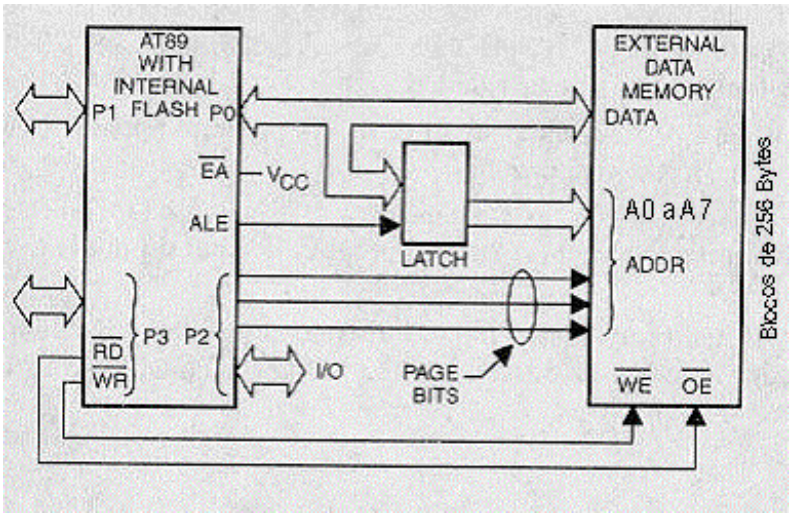
A memória de Dados da família MCS-51 é independente da memória de programa. A memória de Dados interna também é independente da externa

A memória de dados externa é endereçada apenas pela instrução **MOVX**

- **Memória de Dados externa :**

Pode ser endereçada usando um endereço de 8 ou 16 bits

- **Acesso através do modo 8 bits**



Instruções :

Movx a,@Ri
Movx @Ri,a

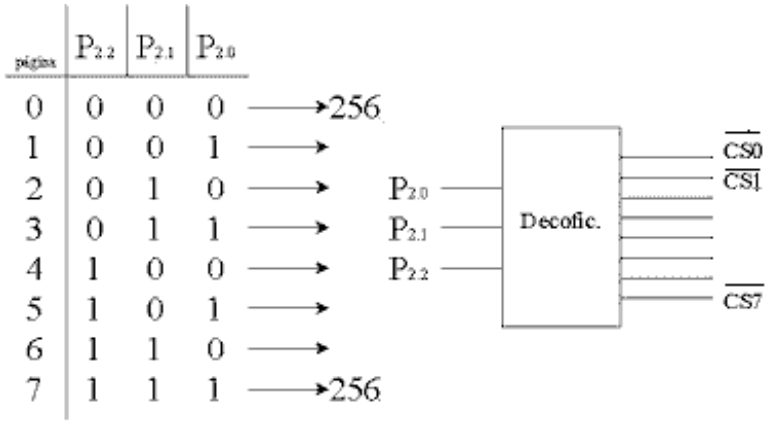
Ri = R0 ou R1

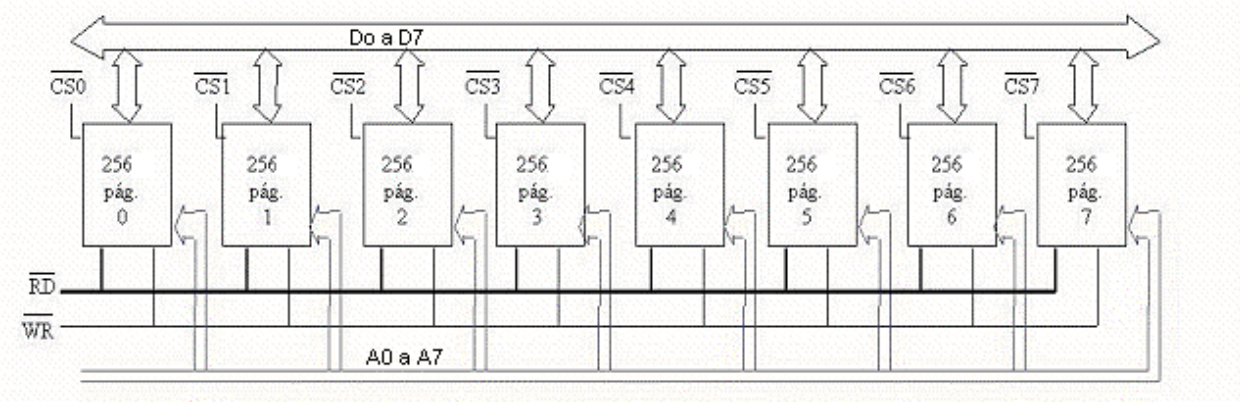
Caso seja utilizado o modo 8 bits, apenas 256 bytes podem ser endereçados (00h a FFh). Este endereço deve ser previamente armazenado em R0 ou R1 .

- **Bits de página**

Um esquema que pode ser utilizado para acessar mais de 256 bytes externos é dividir a RAM externa em páginas de 256 bytes cada através , por exemplo , da porta P2

Exemplo : Usando os bits P2.0 , P2.1 e P2.2 (8 blocos de 256 bytes)





O endereçamento pode ser compreendido dessa forma :

end. página			**end. byte**							
P2.2	*P2.1*	*P2.0*	*A7*	*A6*	*A5*	*A4*	*A3*	*A2*	*A1*	*A0*

Para ler um byte nesta configuração :

mov P2, # end. página
movx a,@Ri

Para escrever um byte nesta configuração:

mov P2 ,# end. página
movx @Ri,a

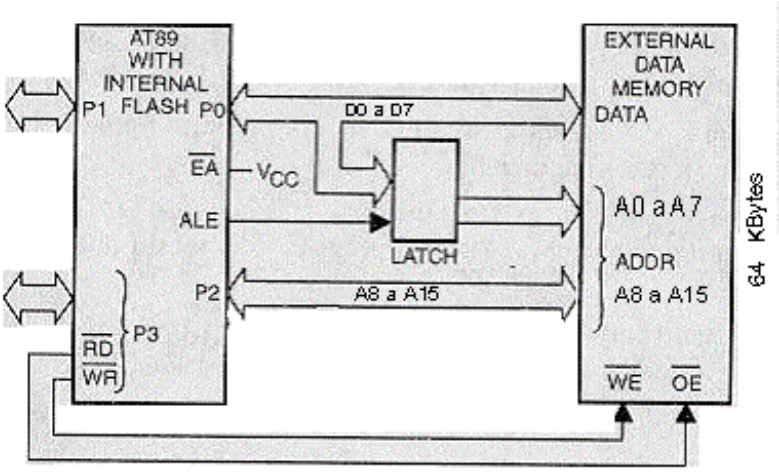
Exemplo: Escrever 3Fh no endereço 00h da página 0:

```
mov R0,#00h    ; endereço 00h em R0
mov a,#3Fh     ; dado 3Fh no acumulador
mov P2,#00h    ; página 0
movx @R0,a     ; escreve no endereço 00 da página 0 --> 3Fh
```

Ler o conteúdo do endereço 0FFh da página 5 :

```
mov R1,#0FFh   ; endereço 0FFh em R1
mov P2,#05h    ; página 5
movx a,@R1     ; armazena no acumulador o conteúdo do endereço 0FFh da página 5
```

- **Acesso através do modo 16 bits**



Instruções :

movx a,@DPTR
movx @DPTR,a

DPTR é um registrador de 16 bits , logo qualquer posição de 0000h a FFFFh externa pode ser acessada tanto na escrita como na leitura .

Exemplo :

a. Armazenar 3Fh na posição 34CBh da memória externa :

mov DPTR,#34CBh
mov a,#3Fh
movx @DPTR,a

b. Ler o conteúdo da posição 13F4h da memória externa :

mov DPTR,#13F4h
movx a,@DPTR

Exercicíos:

a. como acessar um endereço de 8 bits na memória externa mapeada para endereçamento de 16 bits ?

Solução :

Somente os 256 primeiros bytes são acessíveis :

movx @Ri,a (escreve) ou ***movx a,@Ri*** (lê)

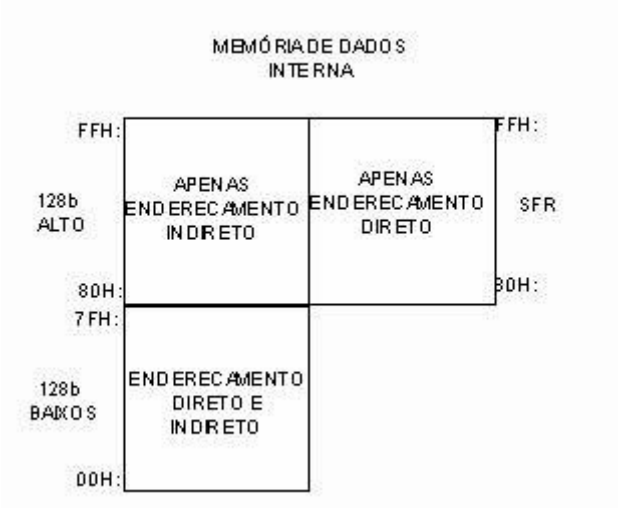
b. Como acessar com endereçamento de 16 bits uma memória externa mapeada para endereçamento de 8 bits ?

Solução :

O DPTR deve conter os endereços 0000h a 00FFh e faz-se a seleção de página normalmente :

mov P2,# end. página
movx a,@DPRT (ler) ou ***movx @DPRT,a*** (escrever)

- **Memória de dados interna :**



- O endereçamento é feito no modo 8 bits
- Chips com 128 bytes de RAM não possuem a área (Apenas Endereçamento Indireto)
- ***Área A:*** 128 bytes inferiores (00h a 7Fh) , acessíveis por endereçamento direto e indireto (existe em toda a família MCS-51)
- ***Área D :*** SFR (Special Function Register) acessível por endereçamento direto (80h a FFh) também existe em todos os membros da família MCS-51
- ***Área I :*** 128 bytes superiores (80h a FFh acessível somente por endereçamento indireto, só existe nos chips de 256 bytes de RAM interna (80C32,80C52,...)).

Exemplos:

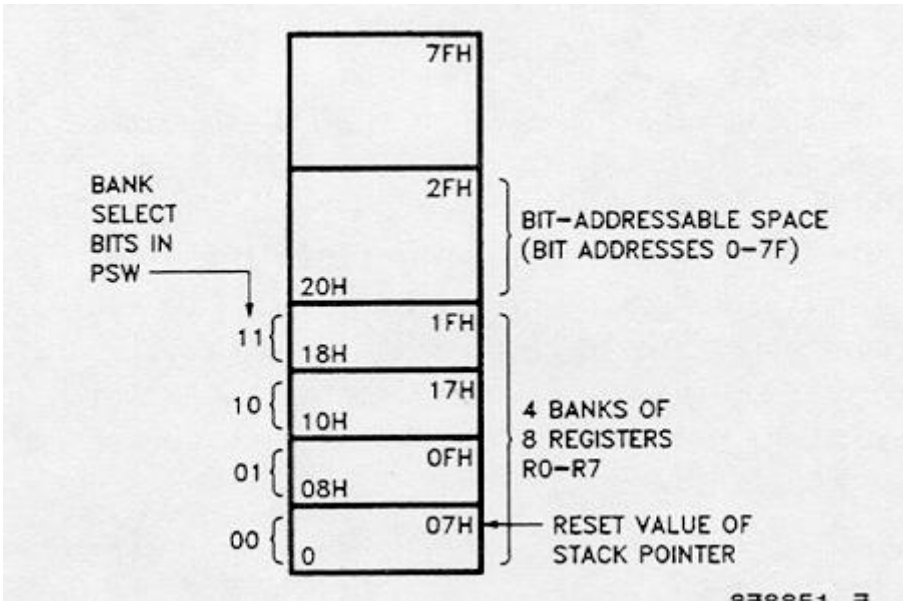
a. Escrever 0AAh na Porta 0 (P0 = 80h ----- área D)

mov 80h,#0AAh

b. Escrever 0AAh no endereço 80h da RAM (área I de um microcontrolador com 256 bytes de RAM interna)

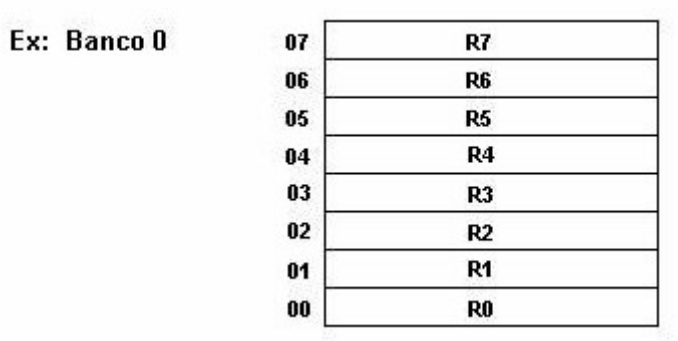
mov R0,#80h
mov @R0,#0AAh

- **Mapeamento da área A da RAM interna**



- **Bancos de registradores**

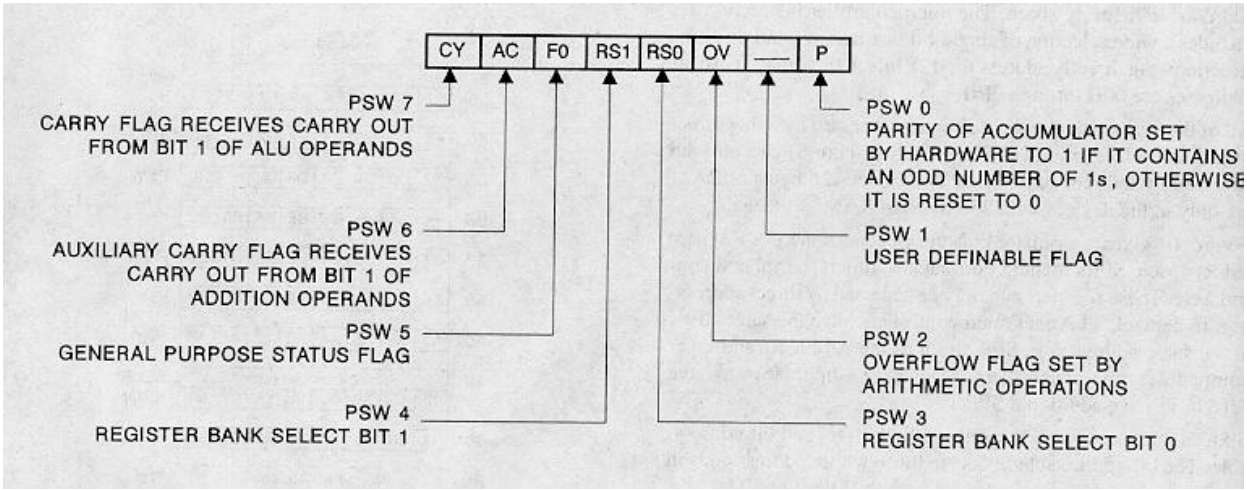
São 4 bancos de 8 registradores cada :



Obs : Cada banco é formado pelos registradores R0 a R7. A seleção entre os Bancos de registradores é feita pelos bits 3 e 4 do byte PSW (program status word)

- **Program Status Word (PSW)**

Esse byte traz informações sobre o estado da CPU e fica localizado no SFR.



RS1	RS0	Register Bank	Address
0	0	0	00H-07H
0	1	1	08H-0FH
1	0	2	10H-17H
1	1	3	18H-1FH

Obs : Ao se resetar a CPU, RS1 e RS0 são 0 , portanto o banco de registradores 'default' é o Banco 0. Além disso o reset inicializa o Stack Pointer (SP) na posição 07h, incrementando a cada vez que é usado. Para que se possa usar mais que um banco de registradores, o SP deve ser inicializado no programa em uma outra posição da RAM (por exemplo 30h).

Exemplo:

mov SP,#30h

- **Área endereçável a bit**

16 Bytes da Área A da RAM podem ser direcionadas diretamente a Bit .São as posições de 20h a 2Fh . Cada Bit nesta área possui um endereço individual podendo ser associado diretamente por uma instrução de Bit .

As instruções de Bit sao :

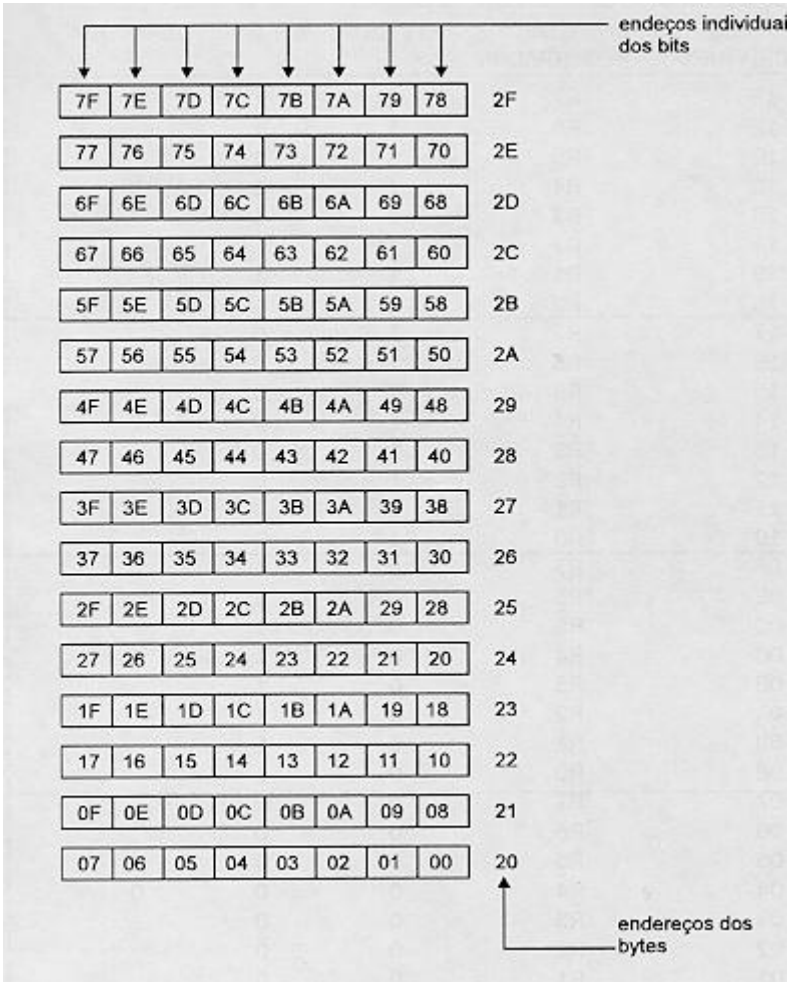
CLR bit_____zera o bit diretamente
SETB bit_____seta o bit diiretamente
CPL bit_____complementa o bit diretamente
ANL C,bit_____AND entre o bit e o carry
ANL C,/bit_____AND entre o complemento do bit e o carry
ORL C,bit_____OR entre o bit e o carry
ORL C,/bit_____OR entre o complemento do bit e o carry
MOV C,bit_____move o bit para o carry
MOV bit,C_____move o carry para o bit
JB bit,rel_____jmp para rel se bit = 1
JNB bit,rel_____jmp para rel se bit = 0
JBC bit,rel_____jmp para rel se bit = 1 e zere o bit

Cada uma dessas posições de memória pode também ser acessada direta ou indiretamente por byte .

Exemplo:

```
mov   20h,#0AAh
mov   R0,#2Fh
mov   @R0,0AAh
```

- **Endereços individuais dos bits (20h a 2Fh)**



Exemplo : Setar o bit 2 da posição 21h

setb 0Ah

ou

setb 21h.2

- **Área de dados (Scratch Pad)**

As posições de 30h a 7Fh da RAM interna, disponíveis para leitura e escrita, através de endereçamento direto e indireto. Se o SP for inicializado no início desta área, deve-se reservar um espaço para a pilha .

- **Mapeamento da área D da RAM interna** (128 Bytes, de 80h a 0FFh)

Esta área é o local dos SFR. O único endereçamento permitido é o direto .

Figure 5. SFR Memory Map

	8 Bytes								
F8									FF
F0	B								F7
E8									EF
E0	ACC								E7
D8									DF
D0	PSW*								D7
C8	T2CON*+	T2MOD+	RCAP2L+	RCAP2H+	TL2+	TH2+			CF
C0									C7
B8	IP*								BF
B0	P3								B7
A8	IE*								AF
A0	P2								A7
98	SCON*	SBUF							9F
90	P1								97
88	TCON*	TMOD*	TL0	TL1	TH0	TH1			8F
80	P0	SP	DPL	DPH				PCON*	87

↑ Bit Addressable

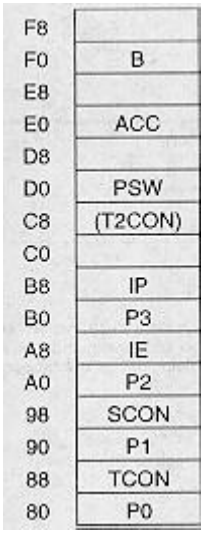
Acc - Acumulador
B - Registrador B
PSW - Palavra do status do programa
SP - Ponteiro de pilha
DPL - LSB do DTPR (Ponteiro de Dados)
DPH MSB do DPTR (Ponteiro de Dados)
P0 - porta 0
P1 - porta 1
P2 - porta 2
P3 - porta 3
IP - Controle de prioridade de Interrupção
IE - Controle de habilitação de interrupção
TMOD - Controle de modo do Temporizador/Contador
TCON - Controle do Temporizador/Contador
T2COM - Controle 2 do Temporizador/Contador
TH0 - MSB do Temporizador/Contador 0
TL0 - LSB do Temporizador/Contador 0
TH1 - MSB do Temporizador/Contador 1
TL1 - LSB do Temporizador/Contador 1
TH2 - MSB do Temporizador/Contador 2
TL2 - LSB do Temporizador/Contador 2
RCAP2H - Registro de captura do T/C 2 - MSB
RCAP2L - Registro de captura do T/C 2 - LSB
SCON - Controle da Serial
SBUF - Buffer de dados da Serial
PCON - Controle de Potência

Qualquer dos SFRs podem ser endereçados a byte diretamente através do endereço de cada um ou do nome.

Exemplo:
mov P0,#3Fh ou ***mov 80h,#3fh***
mov DPL,DPH ou ***mov 82h,83h***

- **SFRs endereçáveis a Bit**

Os SFRs cujos endereços terminam em 0 ou 8h podem também ser endereçados a bit .



F8	
F0	B
E8	
E0	ACC
D8	
D0	PSW
C8	(T2CON)
C0	
B8	IP
B0	P3
A8	IE
A0	P2
98	SCON
90	P1
88	TCON
80	P0

- **Modos de acesso ao Bit**

Os SFRs endereçáveis a bit podem ser utilizados da seguinte maneira:

a) por endereço :

1. ***setb 80h.1*** ; seta o bit 1 do endereço 80h (port 0)

2. ***clr 80h.2*** ; zera o bit 2 do endereço 80h (port 0)

b) por nome :

1. ***setb P0.1*** ; seta o bit 1 do endereço 80h (port 0)

2. ***clr P0.2*** ; zera o bit 2 do endereço 80h (port 0)

c) pelo endereço do bit :

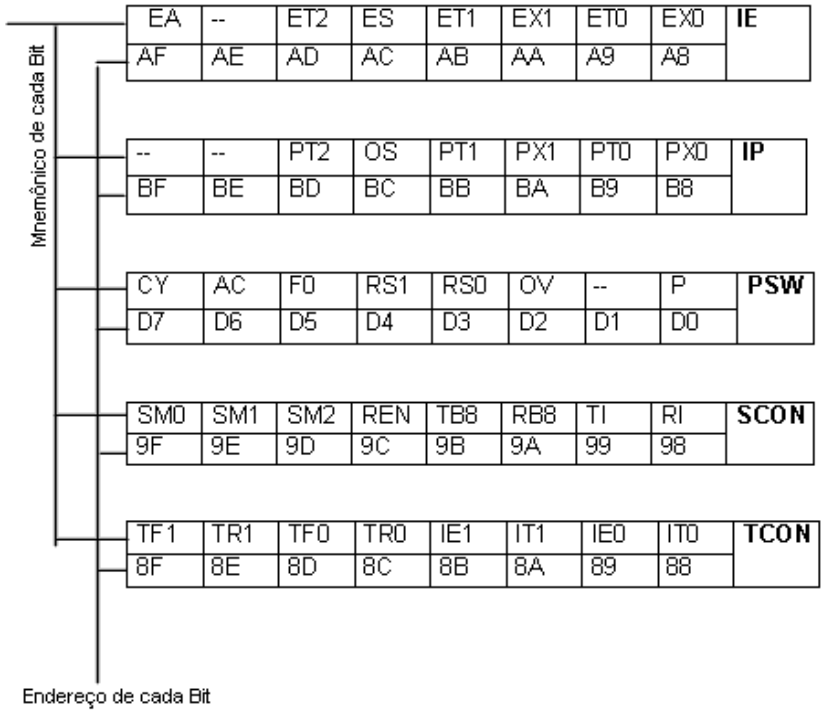
1. ***setb 81h*** ; seta o bit 1 do endereço 80h (port 0)

2. ***clr 82h*** ; zera o bit 2 do endereço 80h (port 0)

FF	FE	FD	FC	FB	FA	F9	F8	
F7	F6	F5	F4	F3	F2	F1	F0	**B**
							E8	
							E0	**Acc**
							D8	
							D0	**PSW**
							C8	**T2CON**
							C0	
							B8	**IP**
B7	B6	B5	B4	B3	B2	B1	B0	**P3**
							A8	**IE**
							A0	**P2**
							98	**SCON**
97	96	95	94	93	92	91	90	**P1**
8F	8E	8D	8C	8B	8A	89	88	**TCON**
87	86	85	84	83	82	81	80	**P0**

Endereço de cada Bit

Além disso, os SFRs endereçáveis a bit que determinam funções, podem ser endereçados através do Mnemônico de cada bit:



Exemplo:

setb EA ; faz o bit 7 de IE=1

setb 0AFh ; idem

Obs :

clr AC ; zera o bit 6 do PSW (Carry auxiliar)

clr 0ACh ; zera o bit de endereço 0ACh , ou seja , o bit 4 do IE