

8051

AULA 2

Prof. Evandro L. L. Rodrigues

A escolha do microprocessador mais adequado depende basicamente da aplicação, e deve se levar em conta os seguintes critérios:

Técnicos:

Velocidade, capacidade de processamento e consumo;

Econômicos:

Custo do projeto, custo de reprodução

Políticos:

Confiança no fornecedor, experiência anterior da equipe, etc...

Estratégicos:

Disponibilidade de mais de um fornecedor, potencial de evolução do componente, etc...

Conhecendo um Microprocessador/Microcontrolador

O microcontrolador 8051

Quatro partes importantes

- Arquitetura interna;
- Pinagem e características elétricas;
- Linguagem de programação;
- Ambiente de Desenvolvimento (Ferramentas).

☞ O 8051 é membro da família MCS-51, e constitui o núcleo de todos os dispositivos MCS-51

☞ É um sistema de um *chip* único, que além do microprocessador de 8 bits, normalmente pode conter:

- . Memória de Programa e Memória de Dados
- . Portas de I/O
- . Comunicação Serial
- . Contadores/ "Timers"
- . Lógica para Controle de Interrupção
- . Conversores A/D e D/A
- . etc ...

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"

Características do Núcleo (Core)

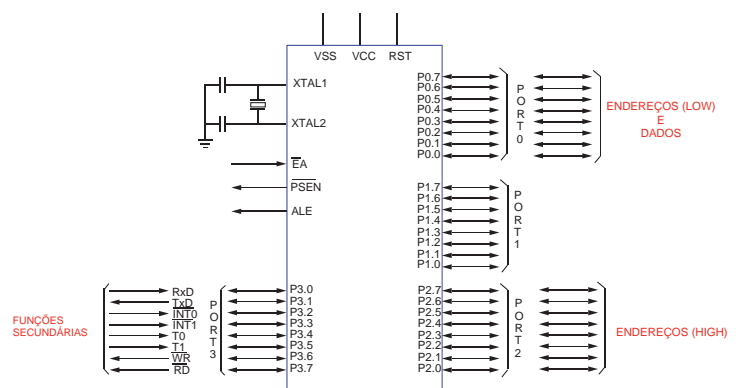
- 128 bytes de memória RAM de dados "on chip"
- 32 linhas de I/O bidirecionais endereçadas individualmente (portas)
- 2 Contadores / Temporizadores de 16 bits cada
- UART full duplex
- Estrutura de interrupção com níveis de prioridade
- Oscilador "on chip"

Características do 8051

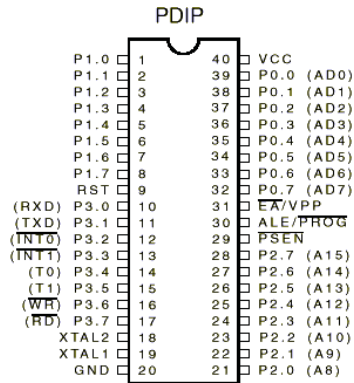
- Duto de dados e ULA de 8 bits
- Versões disponíveis de 12 a 100 MHz
 - instruções de um ciclo, de 1 µsec a 30 ns - DS89C450-ultra-high-speed;
 - C8051F123-GQ – 100MHz, 128K Flash, 8K RAM, ADC, DAC, I2C, SPI, UART, etc. (Silicon Labs)
- O conjunto de instruções inclui:
 - Multiplicação e Divisão
 - Bit set, reset, e test* (Instruções Booleanas).
- Diversos modos de endereçamento.



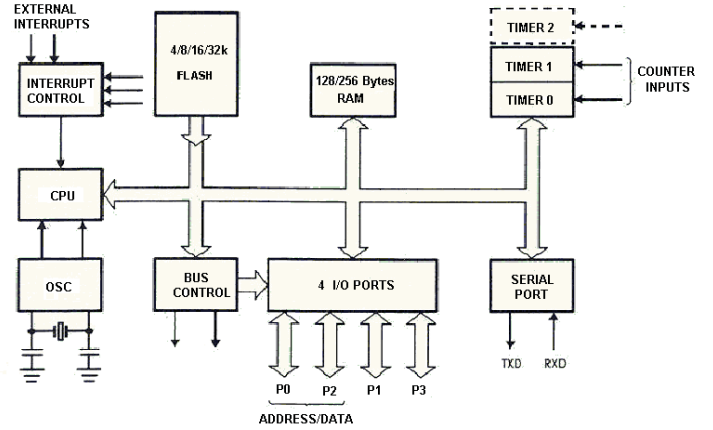
Configuração dos pinos do 8051



Configuração dos pinos do 8051

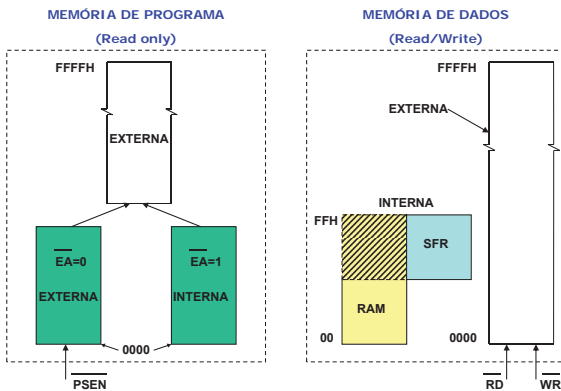


Arquitetura do 8051



Organização da memória da família MCS-51

- Memórias de dados e de programas separadas.



Espaço para Endereçamento

- espaço para endereçamento de até 64Kx8 - Memória de Programa
- espaço para endereçamento de até 64Kx8 de RAM - Memória externa de dados.
- RAM de 128 x 8 - Memória interna de dados.
- SFRs de 128 x 8 : Special Function Registers .
- Endereçamento a bit em 16 posições da RAM e 16 SFRs.

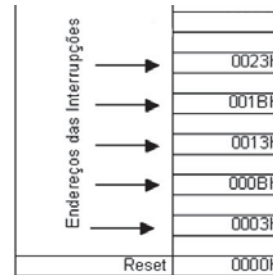
Memória de Programa

- Contador de Programa (PC) de 16 bits
- Ponteiro de Dados (DPTR) de 16 bits.
- endereçamento relativo para acessar *look-up tables*:
 - PC + ACC (Move).
 - DPTR + ACC (Move and jump).
- pino EA em "0" torna inativa a FLASH interna e habilita a memória de programa externa.

Memória de Programa

• Endereços das interrupções:

Cada interrupção causa um salto para um endereço fixo na memória de programa (ROM , EPROM , FLASH...) a partir do endereço 0003



Memória de Programa

• Endereços das memórias de programa interna e externa :

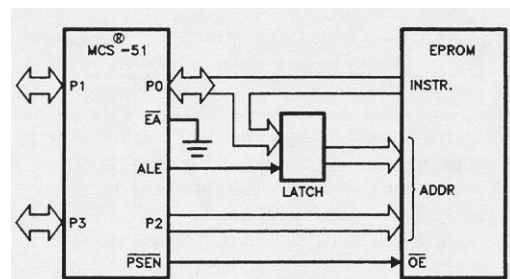
Rom Interna EA = Vcc	Endereçamento Interno	Endereçamento Externo
4 K	0000h a 0FFFh	1000h a FFFFh
8 K	0000h a 1FFFh	2000h a FFFFh
16 K	0000h a 3FFFh	4000h a FFFFh
32 K	0000h a 7FFFh	8000h a FFFFh

Se EA = "0" toda a memória de programa é externa : 0000h a FFFFh

Memória de Programa

• Mapeamento de memória de programa externa

• Mapeamento completo (64 Kb externo)



Memória de Programa

- Mapeamento com EPROM interna + EPROM externa

Exercício :

Com o microcontrolador 87C51 desenvolver o mapeamento externo de memória de programa tal que seja também usado o endereçamento interno .

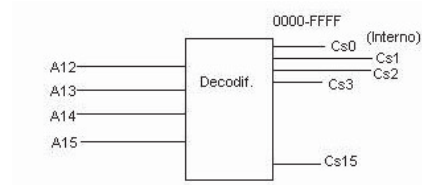
(considerar bloco interno = bloco externo)

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

Memória de Programa

Solução:

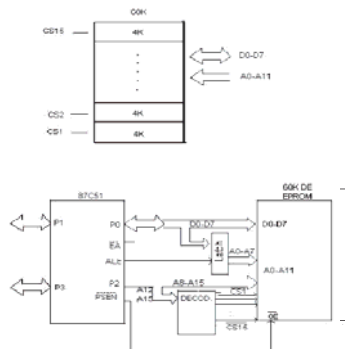
Decodificador de endereços



Memória de Programa

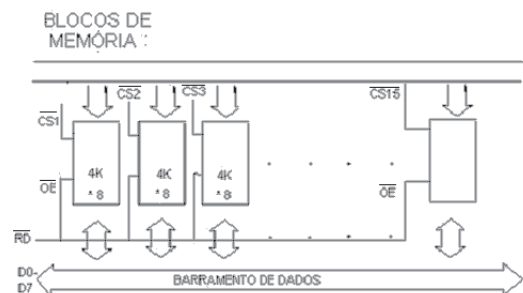
Blocos de memória externa

Solução:



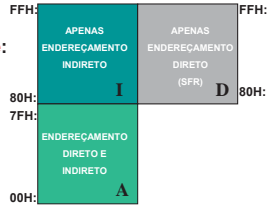
Memória de Programa

Solução:



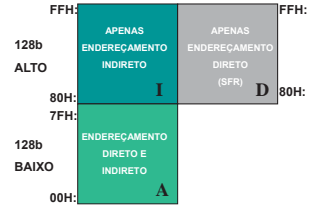
Memória de Dados Interna

- faixa de endereço endereçável **indiretamente**: 00 a FF hexadecimal.
- faixa de endereço endereçável **direta e indiretamente**: 00 a 7F hexadecimal.



Memória de Dados Interna

- O endereçamento é feito com 8 bits
- Chips com 128 bytes de RAM não possuem a área I (Apenas Endereçamento Indireto)



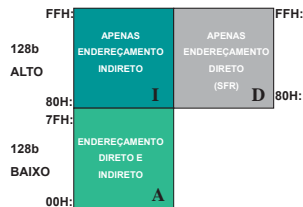
- Área A:** 128 bytes inferiores (00h a 7Fh), acessíveis por endereçamento **direto e indireto** (existe em toda a família MCS-51)
- Área D:** SFR (special function register) acessível por endereçamento **direto** (80h a FFh) também existe em todos os membros da família MCS-51
- Área I:** 128 bytes superiores (80h a FFh) acessível somente por endereçamento **indireto**, só existe nos chips de 256 bytes de RAM interna (8032,8052,...).

Memória de Dados Interna

Exemplos:

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

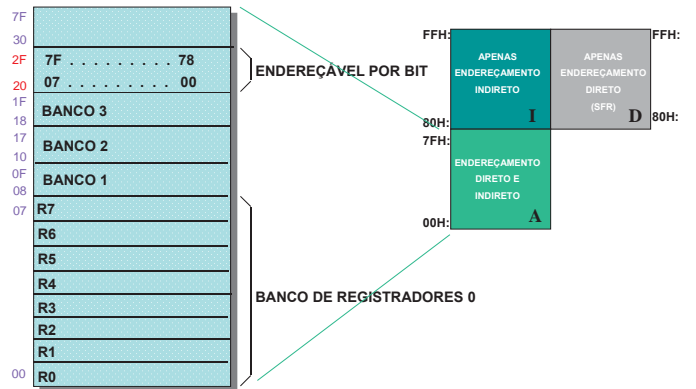
```
mov 80h,#0AAh
```



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
```

Memória de Dados Interna



Memória de Dados Interna

Banco de Registradores

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)

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

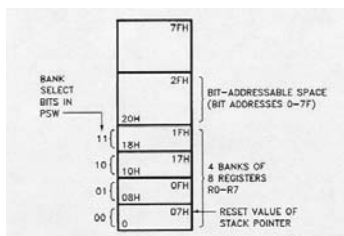
PSW : Program Status Word

CY	AC	F0	RS1	RS0	OV	----	P
----	----	----	-----	-----	----	------	---

Parei aqui 02/03/2014

Memória de Dados Interna

Endereço inicial da Pilha



Ao se resetar a CPU, RS1 e RS0 são 0, portanto o banco de registradores 'default' é o **Banco 0**.

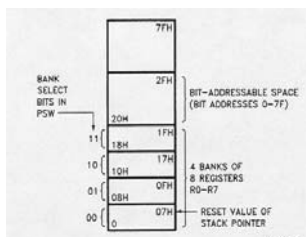
O reset inicializa o Stack Pointer (SP) na posição 07h, e é incrementando a cada vez que é usado.

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).

Memória de Dados Interna

Memória endereçável a Bit

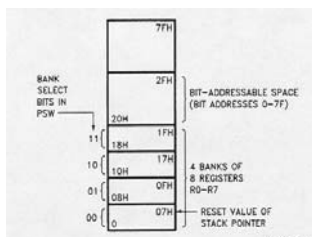
As instruções de Bit são :



CLR bit zera o bit diretamente
SETB bit seta o bit diretamente
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

Memória de Dados Interna

Memória endereçável a Bit



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

Exemplo:

a) Endereçamento Direto
`mov 20h,#0AAh`

b) Endereçamento Indireto
`mov R0,#2Fh`
`mov @R0,#0AAh`

Memória de Dados Interna

Memória endereçável a Bit

Exemplo : Setar o bit 2 da posição 21h

setb 0Ah

OU

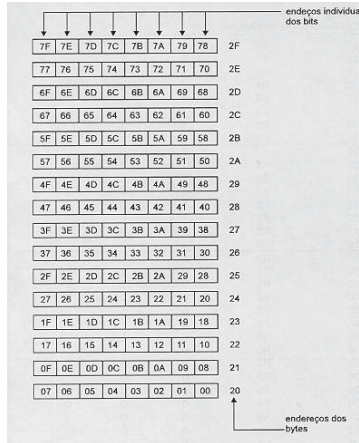
setb 21h.2

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

Exemplo:

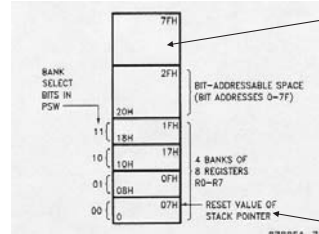
```
mov 20h,#0AAh
mov R0,#2Fh
mov @R0,#0AAh
```

Endereços individuais dos Bits:



Memória de Dados Interna

Área de dados (Scratch Pad)



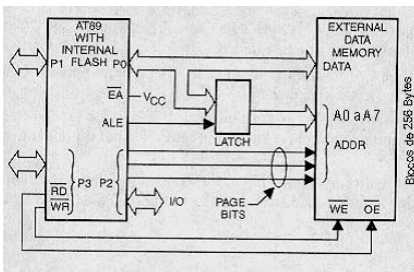
As posições de 30h a 7Fh da RAM interna, são disponíveis para leitura e escrita, através de endereçamento direto e indireto.

Se o SP for inicializado no início desta área, deve-se reservar um espaço para a pilha .

Memória de Dados Externa

Pode ser endereçada usando um endereço de 8 ou 16 bits

Acesso através de endereço de 8 bits



Endereçamento indireto através de R0 ou R1, em segmentos de 256 bytes

Instruções :

```
Movx a,@Ri
Movx @Ri,a
```

Ri = R0 ou R1

Memória de Dados Externa

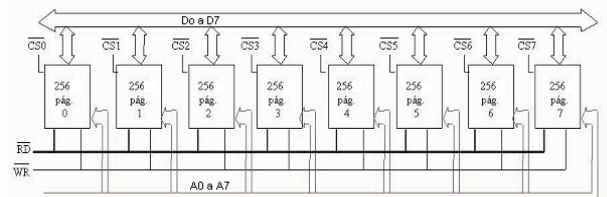
Bits de página

página	P _{2,2}	P _{2,1}	P _{2,0}	→
0	0	0	0	→256
1	0	0	1	→
2	0	1	0	→ P _{2,3}
3	0	1	1	→ P _{2,2}
4	1	0	0	→
5	1	0	1	→
6	1	1	0	→
7	1	1	1	→256

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

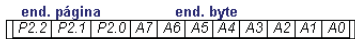
end. página end. byte

P _{2,2}	P _{2,1}	P _{2,0}	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀
------------------	------------------	------------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------



Memória de Dados Externa

Bits de página



Para ler um byte nesta configuração :

```
mov R0,#end.byte
mov P2,#end.página
movx a,@R0
```

Para escrever um byte nesta configuração:

```
mov R0,#end.byte
mov P2,#end.página
movx @R0,a
```

Memória de Dados Externa

Bits de página

Exemplo:

Escrever 3Fh no endereço 00h da página 0:

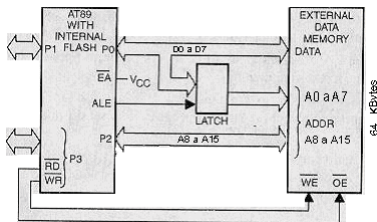
```
mov R0,#00h ; endereço 00h em R0
mov a,#3Fh ; dado 3Fh no acumulador
mov P2,#00h ; página 0
movx @R0,a ; escreve no endereço 00 da página 0 --> 3Fh
```

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
```

Memória de Dados Externa

Acesso através de endereço de 16 bits



- espaço de endereço de 64K bytes
- espaço todo é indiretamente endereçável pelo ponteiro de dados DPTR.

Instruções :

```
movx a,@DPTR
movx @DPTR,a
```

Memória de Dados Externa

Acesso através de endereço de 16 bits

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
```

Memória de Dados Externa

Acesso através de endereço de 16 bits

Exercícios:

Como acessar, usando um endereço de 8 bits, a memória externa mapeada para endereçamento de 16 bits ?

Solução :

Somente os 256 primeiros bytes serão acessíveis :

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

Memória de Dados Externa

Acesso através de endereço de 16 bits

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

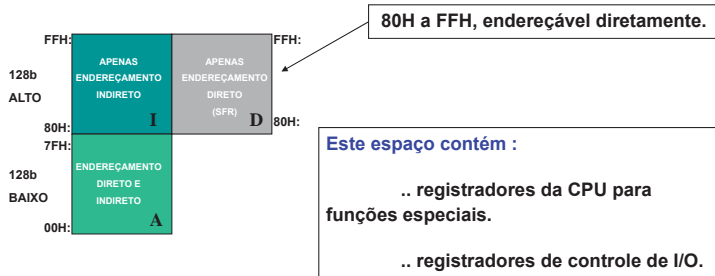
Solução :

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

`mov P2,# end. página`
`movx a,@DPTR` (ler) ou `movx @DPTR,a` (escrever)

Registadores de Funções Especiais (SFR)

Espaço de Endereço dos Registradores de Funções Especiais (SFR)



16 posições são endereçáveis por bit (endereços terminando em 0 ou 8)

Mapa dos Registradores de Funções Especiais (SFR)

endereçável por Bit

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	DPH	DPL					PCON	87

Registadores de Funções Especiais (SFR)

registadores da CPU:

- ACC : Accumulador.
- B : Registrador B.
- PSW : Program Status Word.
- SP : Stack Pointer.
- DPTR : Data Pointer (DPH, DPL).

controle de interrupção:

- IE : Interrupt Enable.
- IP : Interrupt Priority.

portas de I/O:

- P0 : Port 0.
- P1 : Port 1.
- P2 : Port 2.
- P3 : Port 3.

Registadores de Funções Especiais (SFR)

timers:

- TMOD : modo do Timer
- TCON : controle do Timer
- TH0 : byte + sign. do Timer 0
- TL0 : byte -sign do Timer 0
- TH1 : byte + sign. do Timer 1
- TL1 : byte - sign. do Timer1

comunicação serial :

- SCON : Serial port control.
- SBUF : Serial data registers.

Outro:

- PCON : Power control & misc.

Registadores de Funções Especiais (SFR)

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`

Registadores de Funções Especiais (SFR)

F8	Y
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

SFRs endereçáveis a Bit

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

Modos de acesso ao Bit

a) por endereço do Bit dentro do Byte:

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)

P0	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
80h	80.7	80.6	80.5	80.4	80.3	80.2	80.1	80.0

Registadores de Funções Especiais (SFR)

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

SFRs endereçáveis a Bit

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)

P0	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
80h	80.7	80.6	80.5	80.4	80.3	80.2	80.1	80.0

Registadores de Funções Especiais (SFR)

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

SFRs endereçáveis a Bit

c) pelo endereço absoluto 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)

P0	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
80h	80.7	80.6	80.5	80.4	80.3	80.2	80.1	80.0
	87	86	85	84	83	82	81	80

Registadores de Funções Especiais (SFR)

SFRs endereçáveis a Bit

FF	FE	FD	FC	FB	FA	F9	F8	
F7	F6	F5	F4	F3	F2	F1	F0	B
							E8	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

Registadores de Funções Especiais (SFR)

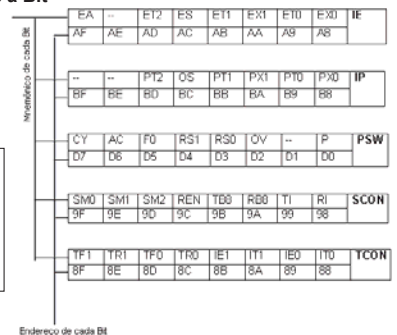
SFRs endereçáveis a Bit

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



Atenção! :

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

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