

SEL-433 APLICAÇÕES DE MICROPROCESSADORES I

8051  
AULAS

Prof. Evandro L. L. Rodrigues

AMD vs. intel

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...



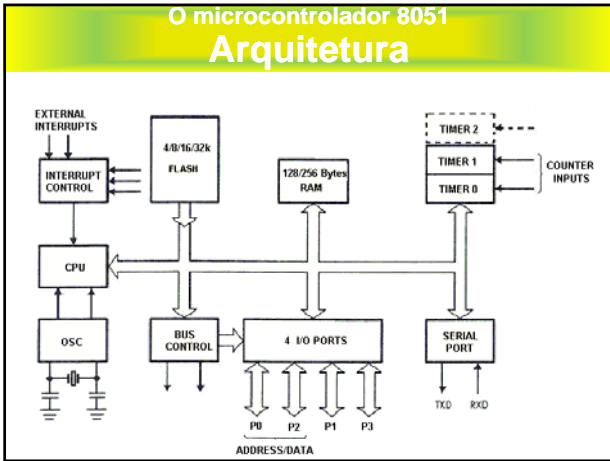
Ferramentas de auxílio para seleção das características



Conhecendo um Microprocessador/Microcontrolador

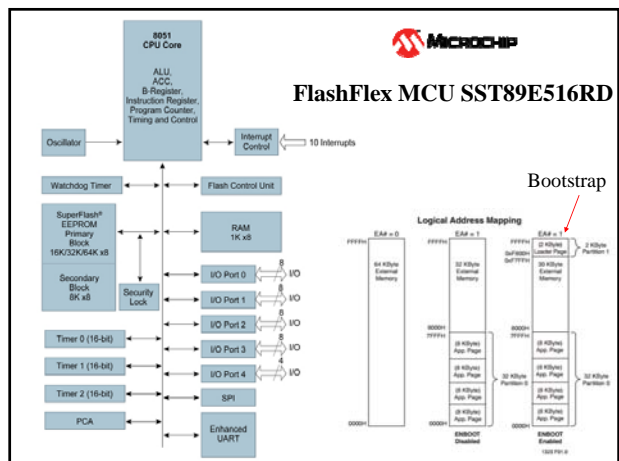
Cinco partes importantes

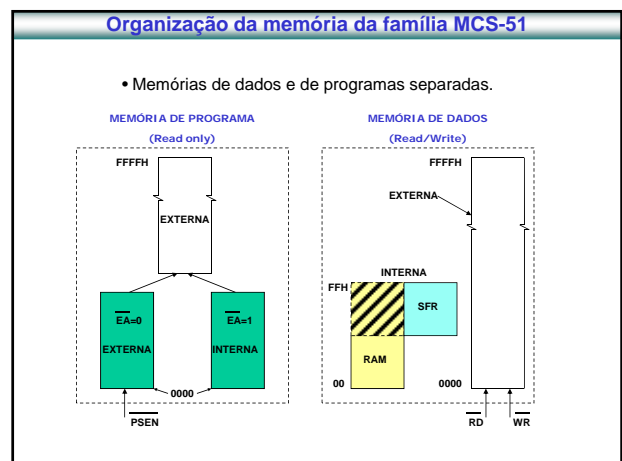
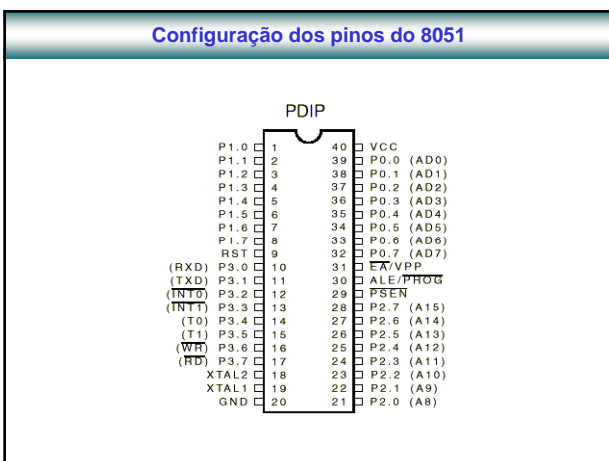
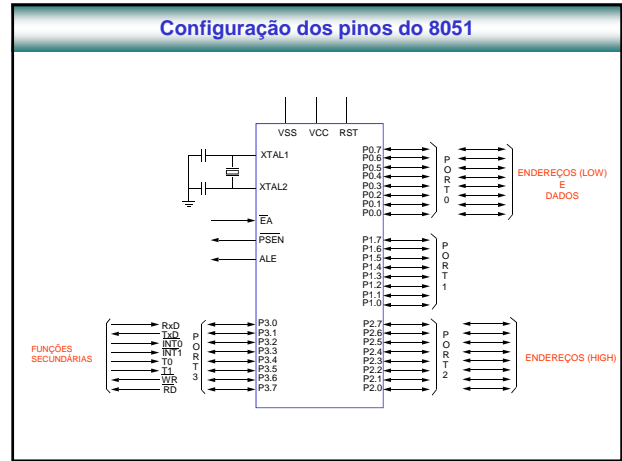
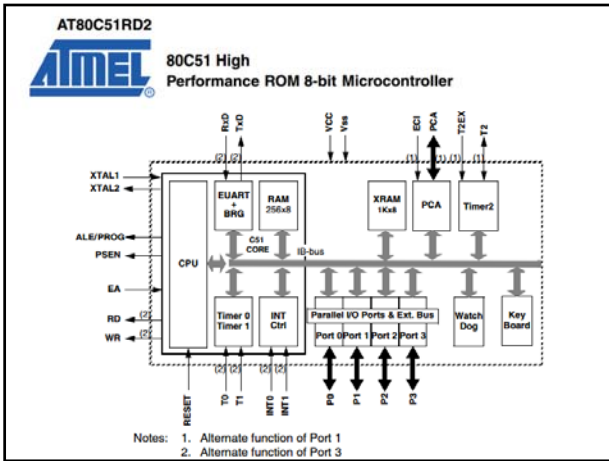
- Arquitetura interna
- Conjunto de Instruções
- Pinagem
- Características Elétricas
- Ambiente de Desenvolvimento Integrado - IDE



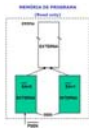
- ### Características do Núcleo (Core)
- CPU de 8 bits otimizada
  - Capacidade de processamento booleano (lógica de um único bit)
  - 64 Kbytes de espaço de memória de programa
  - 64 Kbytes de espaço de memória de dados
  - Espaço de memória de programa "on chip"
  - RAM interna (128/256 bytes)
  - 4 Portas (comandos / comunicação)
  - Temporizadores (2 ou 3)

- ### Características do 8051
- 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)
  - 512 byte/1024 byte RAM
  - 0-40MHz @ 5V
  - Dual-block SuperFlash –16/32/64 Kbyte + 8KByte
  - Programmable Counter Array
  - UART & SPI interface
  - Brown-out detection, etc.



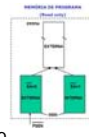


### Memória de Programa



- 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

Primeiro Endereço	0033h
Endereços das Interrupções	002Bh
	0023h
	001Bh
	0013h
	000Bh
	0003h
Reset	0000h

### 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 Interna de Dados

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

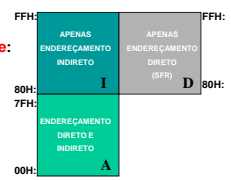


Diagrama da Memória Interna de Dados:

- 00h a 7Fh: ENDEREÇAMENTO DIRETO E INDIRETO (A)
- 80h a FFh: APENAS ENDEREÇAMENTO DIRETO (D)
- 80h a FFh: APENAS ENDEREÇAMENTO INDIRETO (I)

### Memória Interna de Dados

•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 Interna de Dados

**Exemplos:**

a. Escrever 0AAh na Porta 0 (PO = 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 Interna de Dados

**ENDEREÇAVEL POR BIT**

**BANCO DE REGISTRADORES 0**

### Memória Interna de Dados

#### Banco de Registradores

Cada banco é formado pelos registradores R0 a R7.  
 A seleção dos 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

No Reset da CPU, RS1 e RS0 são 0 , portanto o banco de registradores 'default' é o **Banco 0**.

**PSW : Program Status Word**

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

### Memória Interna de Dados

Endereço inicial da Pilha

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

7F	
30	
2F	7F ..... 78
20	07 ..... 00
1F	
18	BANCO 3
17	
10	BANCO 2
0F	
08	BANCO 1
07	R7
06	R6
05	R5
04	R4
03	R3
02	R2
01	R1
00	R0

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 Interna de Dados

Memória endereçável a Bit

As instruções de Bit são :

7F	
30	
2F	7F ..... 78
20	07 ..... 00
1F	
18	BANCO 3
17	
10	BANCO 2
0F	
08	BANCO 1
07	R7
06	R6
05	R5
04	R4
03	R3
02	R2
01	R1
00	R0

**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 zero o bit

### Memória Interna de Dados

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`

7F	
30	
2F	7F ..... 78
20	07 ..... 00
1F	
18	BANCO 3
17	
10	BANCO 2
0F	
08	BANCO 1
07	R7
06	R6
05	R5
04	R4
03	R3
02	R2
01	R1
00	R0

### Memória Interna de Dados

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																
7F	7E	7D	7C	7B	7A	79	78	77	76	75	74	73	72	71	70	6F
2F	2E	2D	2C	2B	2A	29	28	27	26	25	24	23	22	21	20	1F
EF	EE	ED	EC	EB	EA	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0	DF
6F	6E	6D	6C	6B	6A	69	68	67	66	65	64	63	62	61	60	5F
5F	5E	5D	5C	5B	5A	59	58	57	56	55	54	53	52	51	50	4F
3F	3E	3D	3C	3B	3A	39	38	37	36	35	34	33	32	31	30	2F
4F	4E	4D	4C	4B	4A	49	48	47	46	45	44	43	42	41	40	3F
6F	6E	6D	6C	6B	6A	69	68	67	66	65	64	63	62	61	60	5F
2F	2E	2D	2C	2B	2A	29	28	27	26	25	24	23	22	21	20	1F
3F	3E	3D	3C	3B	3A	39	38	37	36	35	34	33	32	31	30	2F
2F	2E	2D	2C	2B	2A	29	28	27	26	25	24	23	22	21	20	1F
1F	1E	1D	1C	1B	1A	19	18	17	16	15	14	13	12	11	10	0F
0F	0E	0D	0C	0B	0A	09	08	07	06	05	04	03	02	01	00	FF

Endereços dos bytes

### Memória Interna de Dados

**Área de dados (Scratch Pad)**

7F	
30	
2F	7F ..... 78
20	07 ..... 00
1F	
18	BANCO 3
17	
10	BANCO 2
0F	
08	BANCO 1
07	
R7	
R6	
R5	
R4	
R3	
R2	
R1	
R0	

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**

**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 :

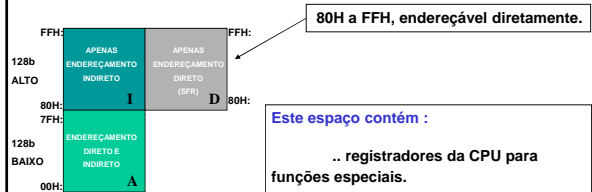
```
mov DPTR,#34CBh
mov a,#3Fh
movx @DPTR,a
```

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

```
mov DPTR,#13F4h
movx a,@DPTR
```

### 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)

Revisão: agosto 07/03/2016

### 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											C7
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)

registradores 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	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**

**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)

**SFRs endereçáveis a Bit**

c) pelo endereço absoluto do bit :

- setb 81h** ; seta o bit 1 do endereço 80h (Port 0)
- clr 82h** ; zera o bit 2 do endereço 80h (Port 0)

P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
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**

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

### Registadores de Funções Especiais (SFR)

**PSW : Program Status Word**

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

Carry  
Auxiliary Carry  
Propósito Geral

Seleção de Banco de Registradores

Overflow  
OV = 1 indica uma condição de erro, o resultado não pode ser representado como um número sinalizado (ex. soma de números negativos resultando em positivo)

Paridade  
quantidade de n°s 1 no ACC é par ou não.

### Explorando um pouco a Arquitetura 8051

- Registradores
- Modos de Endereçamento (Aula2)
- Diretivas (Aula2\_MCU8051)
- Flags
- Condições de desvio
- Ambiente de Simulação
- Arquivos gerados

### Sobre o Formato Intel HEX

Arquivo ".hex" → pode ser aberto por qualquer editor de texto;

O formato padrão:

```
:10010000214601360121470136007EFE09D2190140
:100110002146017EB7C20001FF5F16002148011988
:10012000194E79234623965778239EDA3F0182CAA7
:100130003F0156702B5E712B722B732146013421C7
:00000001FF
```

Start code  
Byte count  
Address  
Record type  
Data  
Checksum

Mais informações em <http://www.keil.com/support/docs/1584/>.