

# Microcontroladores

## Arquitetura

Prof. Guilherme Peron  
Prof. Ronnier Rohrich

# Introdução a Microcontroladores

# Apresentação

- Site da disciplina:

Moodle → [moodle.utfpr.edu.br](http://moodle.utfpr.edu.br)

Cursos → Curitiba → Departamentos Acadêmicos → DAELN → Engenharia → CSW40

Senha: verificar com cada professor

- E-mail dos professor:

[rohrich@utfpr.edu.br](mailto:rohrich@utfpr.edu.br)

# Microprocessador

O que é?

Para que serve?

Eu já vi algum?

Qual o tamanho?

É caro?

É de comer?



# Microprocessador

Vamos fazer um bolo de 1 kg...



# Microprocessador

O que precisamos?

- 1) Livro de receitas -> passo a passo de como fazer (rotina);
- 2) Definir o sabor;
- 3) Acessar a receita do livro
- 4) Descobrir os ingredientes
- 5) Vamos até a página correspondente a receita  
-> Software;
- 6) Em qual página está? “XYZ”

# Microprocessador

Qual é a minha atitude neste instante?

R: Somente ler a receita!!! (não se afobe, se não seu bolo vai ficar uma m...)

Analisando o caso, qual seria a nossa memória?

# Microprocessador

Memória ROM (Read only Memory);

Receita:

- 1) 3 xícaras de farinha;
- 2) 2 xícaras de açúcar;

Cobertura: (vide pág. “ABC”)

**Por que a cobertura não está escrita aqui também?**



# Microprocessador

R: Várias receitas utilizam!!! (isso seria uma sub-rotina)

O que devemos fazer???

# Microprocessador

R: Vamos fazer a cobertura e depois voltamos e damos continuidade a nossa receita.

Sequência:

**Executando XYZ->Executo ABC->retorno XYZ**

Pergunta: E se eu quiser fazer um bolo de 2 kg?

# Microprocessador

R: Podemos utilizar uma folha de rascunho. - RAM  
(Random Access Memory)

Resumindo:

- 1) Cozinheiro - microprocessador (Hardware);
- 2) Ingredientes - eletrônica (Hardware)
- 3) Livro de receitas - ROM (Software);
- 4) Rascunho - RAM (Software).

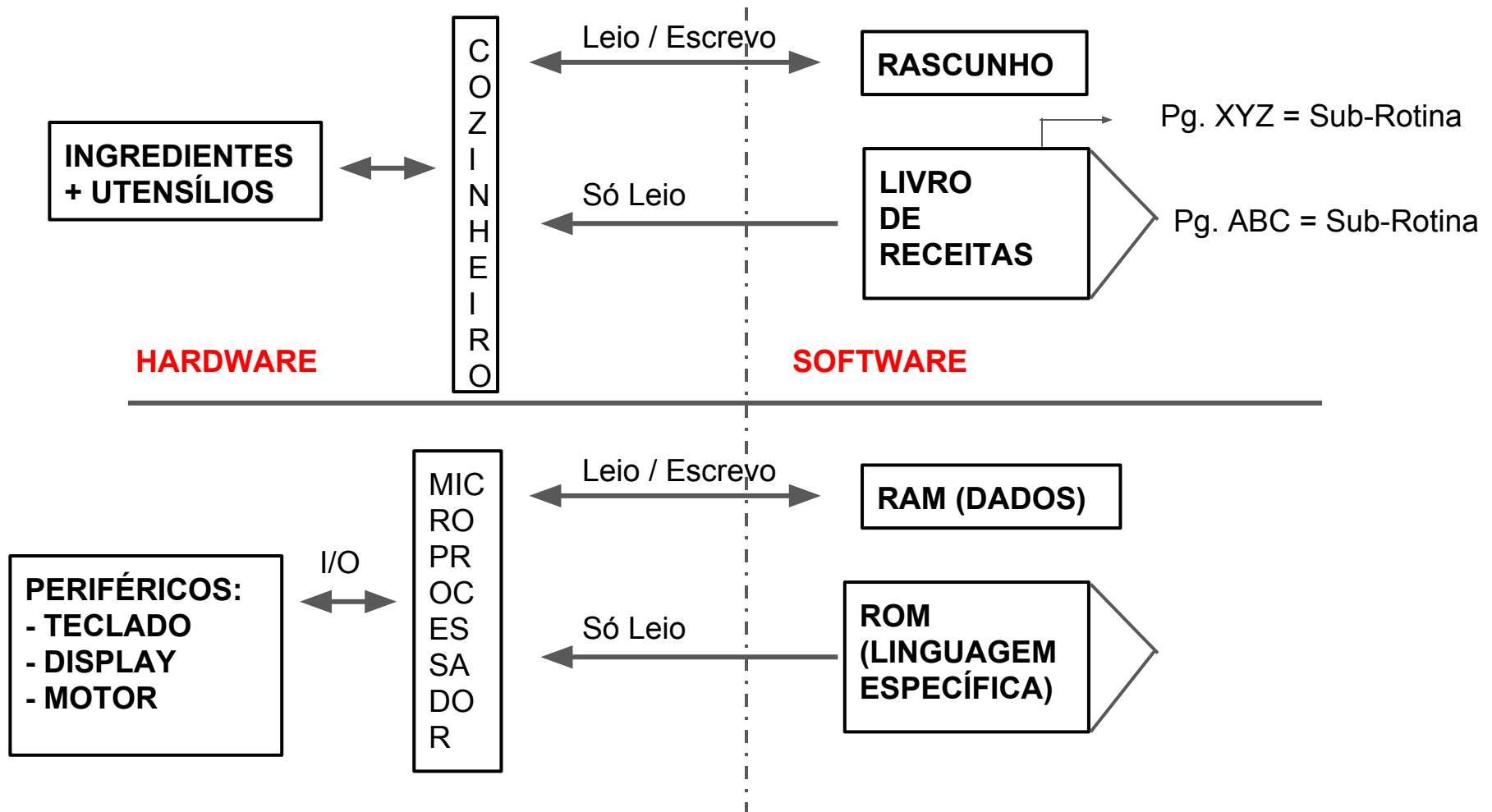
Qual a diferença entre o microprocessador e o cozinheiro?

# Microprocessador

R: Vocabulário.

Resultado: precisamos entender o vocabulário do microprocessador.

# Microprocessador



# Microprocessador

1) Componente Eletrônico;

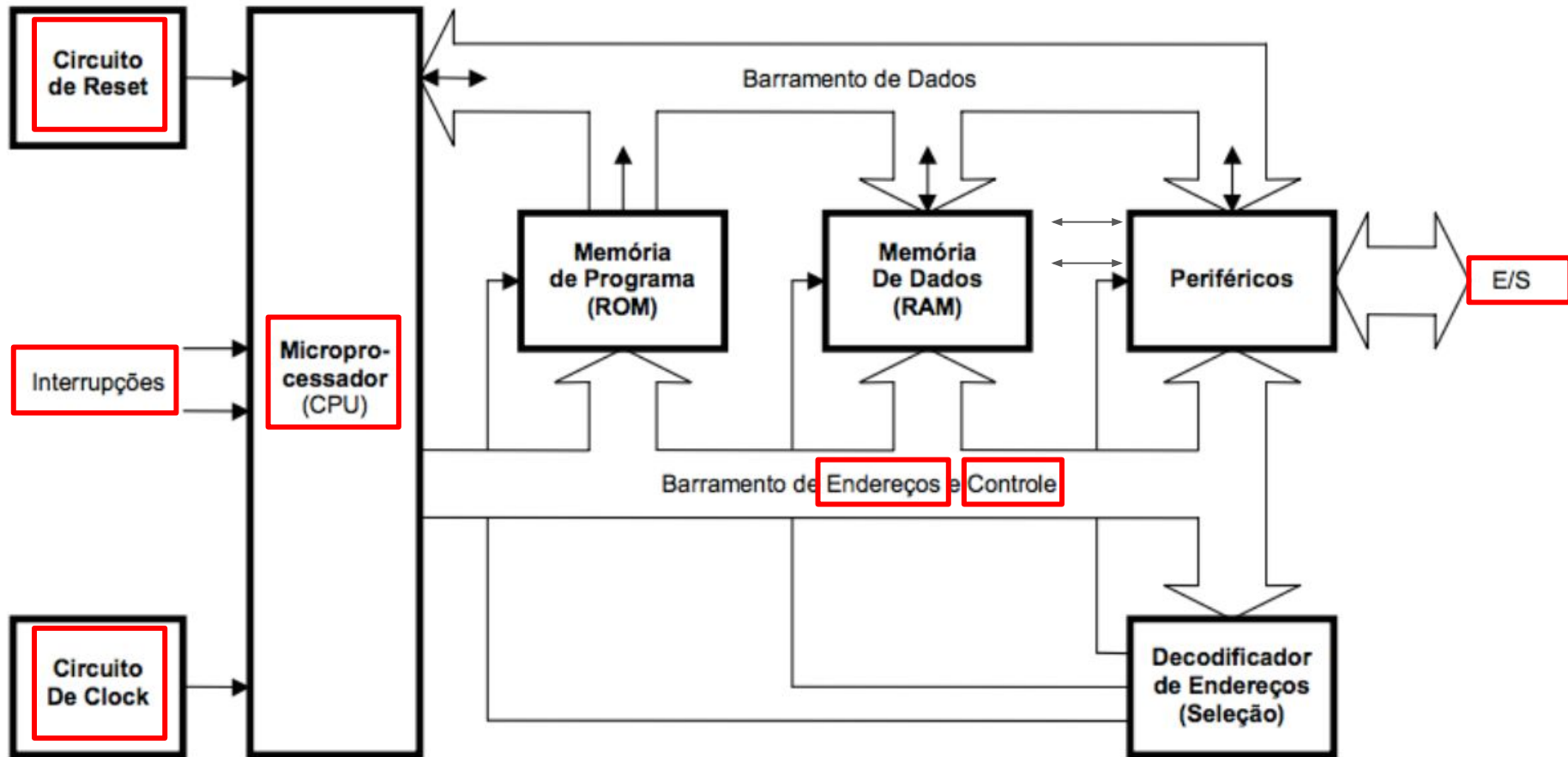
2) Composição Básica:

- a) ROM: Ler INSTRUÇÕES que devem ser executadas;
- b) RAM: Armazenar temporariamente informações de uso das INSTRUÇÕES;
- c) Barramentos de Dados e Endereços.

3) Atribuições:

- a) Executar tarefa da ROM;
- b) Se comunicar com o mundo (teclado, impressora, LCD...)

# Diagrama Genérico



# Funções

- 1) Barramento de Endereços: Selecionar com qual posição de memória ou periférico deseja se comunicar;
- 2) Controle: Permitem o microprocessador acionar a RAM e a ROM em um certo tempo específico e vice-versa (ligar/desligar);
- 3) Barramento de E/S (I/O): Comunicação com o mundo externo;
- 4) CPU (Engenheiro formado e casado): Se comunicar e acionar todos os barramentos, obedecendo a ROM.



# Funções da CPU

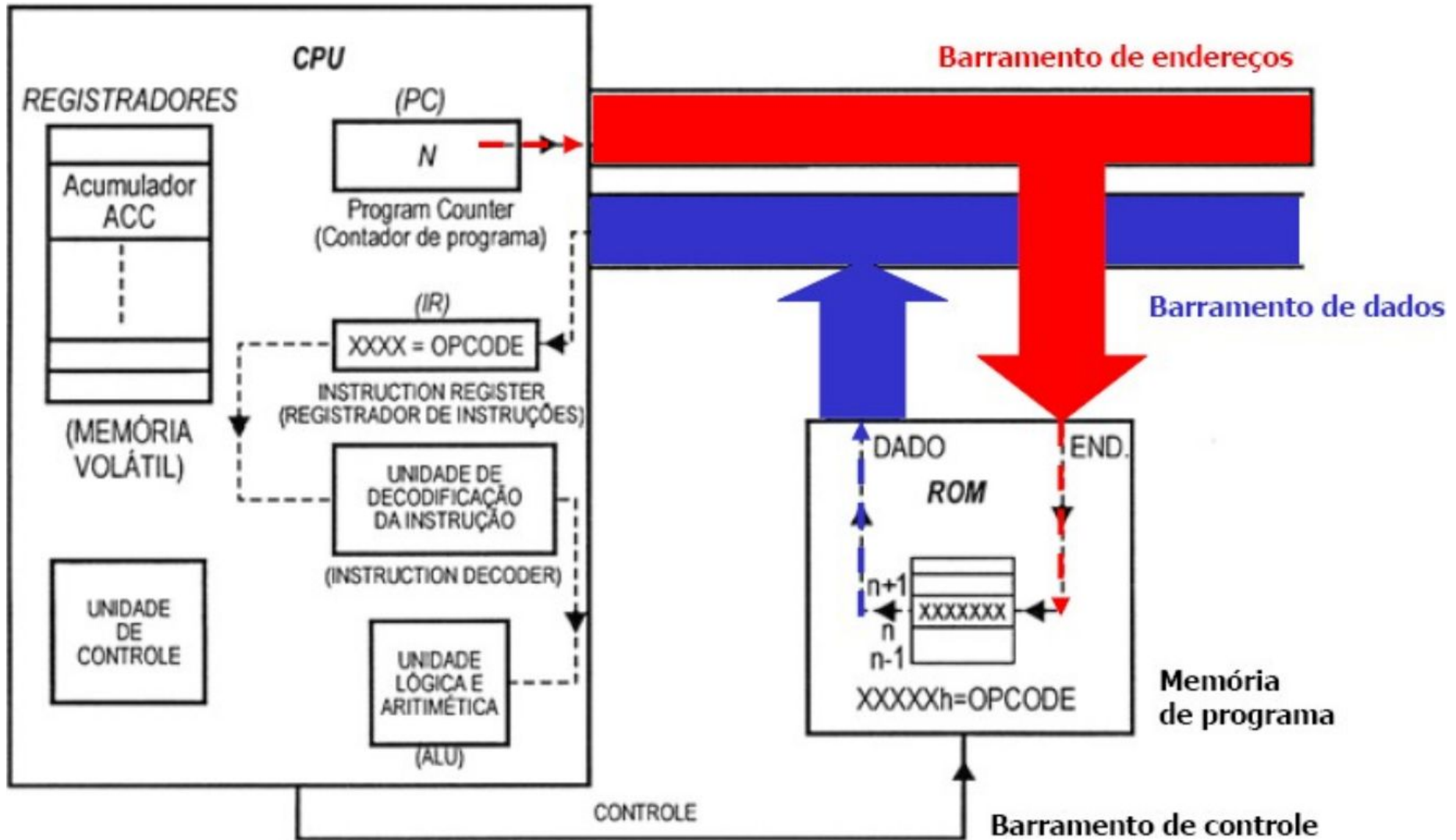
- a) Buscar instruções continuamente na ROM;
- b) Executar essas instruções;
- c) Executar funções lógicas;
- d) Executar funções aritméticas;
- e) Executar transferências de dados (internas e externas);
- f) Executar comparação de dados -> tomada de decisão;

**Meio “Tapada” => Só entende binário!!!**

# Funções

- 5) Oscilador: Tarefas internas e externas sincronizadas e com uma velocidade predeterminada;
- 6) Reset: Iniciar as rotinas e realizar a leitura no primeiro endereço 0000h;
- 7) Interrupções: Pinos de acesso externo que interrompem o microprocessador.

# CPU Internamente



# Funções

**Registradores:** Semelhante a RAM, só que interna a CPU. Armazena informações de “dentro” ou “de fora” da CPU.

Ex1: P0 - Proveniente da porta de I/O

Ex2: R0 ou XXh - Uso geral - utiliza as instruções do microprocessador.

# Funções

**Program Counter (PC):** “Onde eu estou e pra onde eu devo ir”. Endereça a próxima instrução a ser lida;

**Instruction Register (IR):** Preciso buscar uma instrução e colocar em algum lugar.

Onde a instrução está? ROM

Onde vou colocá-la? IR

**Passos: Busquei (ROM) -> Coloquei (IR) -> decodifica a instrução -> envia para ALU -> Executa a instrução**

# Funções

**Unidade de Decodificação:** Além do que vocês estão imaginando, são gerados os controles (Junto com a unidade de controle) que permitem a execução de uma instrução;

**ALU ou ULA:** Faz o que o nosso cérebro faz.

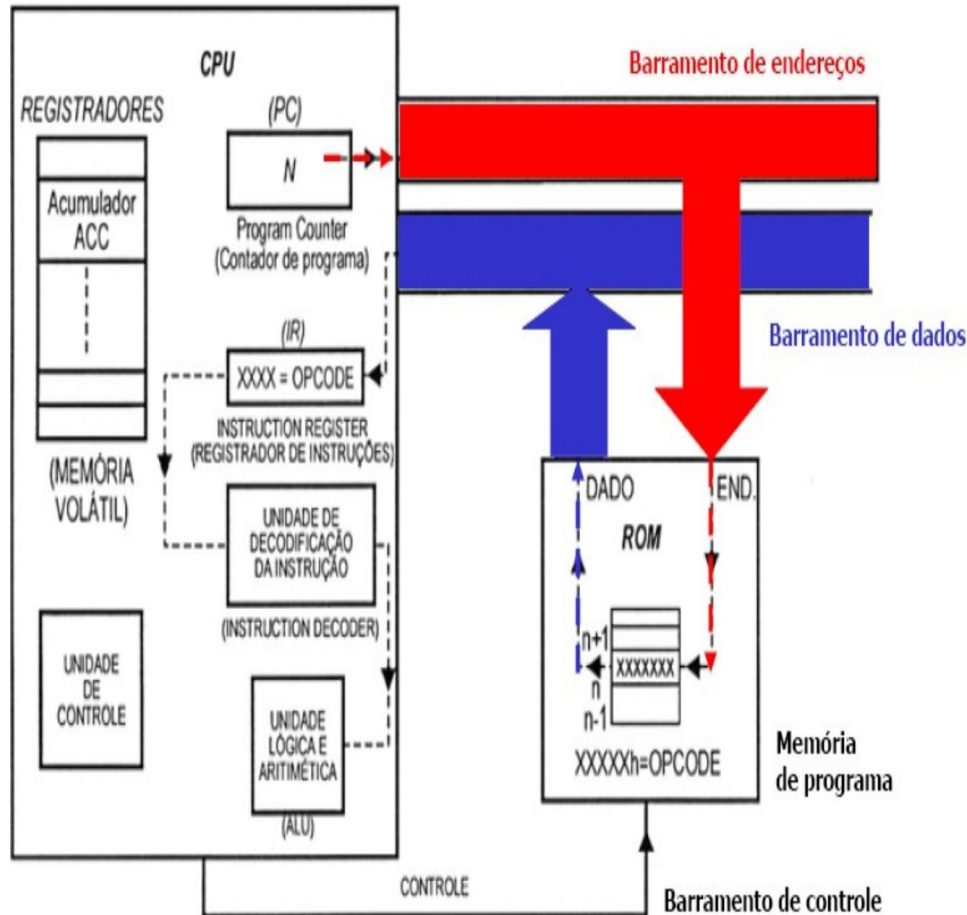
# Funções

**Acumulador:** The Best Registrator!!!

Várias instruções para ele!!!

**Unidade de Controle:** Organiza a zona que está ocorrendo dentro do microprocessador (Processa o fluxo das informações para que as instruções sejam executadas).

## Exemplo - O microprocessador vai ler uma instrução na ROM:



1.O endereço de PC vai p/ BUS

2.Ativa o sinal de controle da ROM

3.Ciclo de busca:

- lê da ROM;
- no endereço dado pelo PC;
- lê/transmite pelo BUS dados.

4.Instrução carregada e armazenada no IR;

5.Incrementa o PC (próxima inst.)

6.Inicia o ciclo de execução.



# Microprocessador vs Microcontrolador

Qual a diferença?

# SONO

De novo vou ter que pensar professor!!!!

Deixa eu dormir aqui!!!!



# Microprocessador vs Microcontrolador

1. Hardware interno diferente  
(microcontrolador + e microprocessador -)
2. Microprocessador contém: IR, PC, ALU, INT...
3. Microcontrolador: tudo o que o outro têm + TIMER, SERIAL, RAM, ROM...

# Periféricos

Exemplos:

1. Memórias;

2. Teclados;

3. Displays;

4. Impressoras;

5. Joysticks;

6. Sensores (Conversão);

7. Motores;

# 8051

- Já passou por várias transformações

Primeira família: 8048

Nova família:

8051, 8751, 8052...



**NÃO PARARAM A REPRODUÇÃO!!!!**

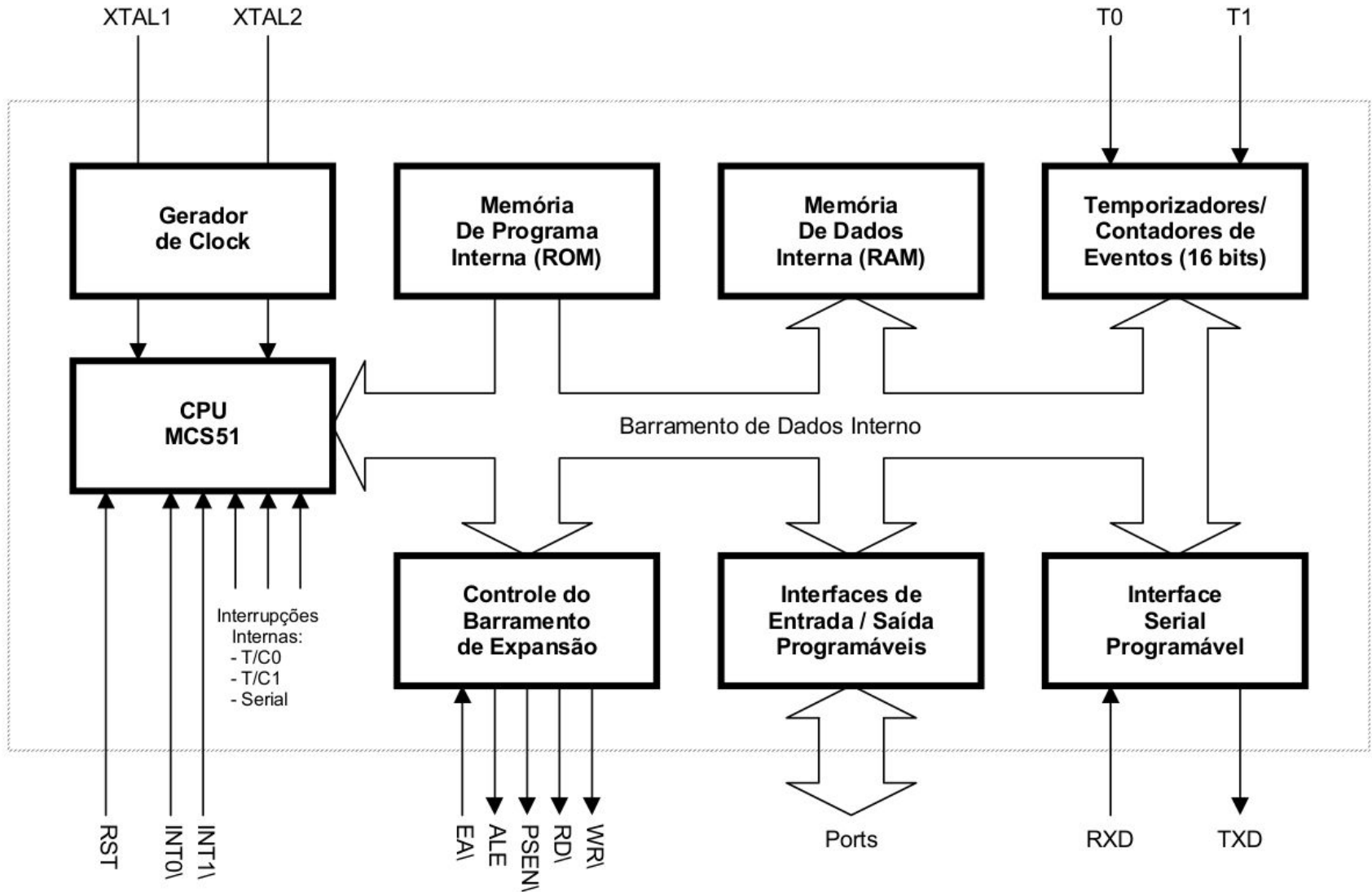
# 8051 - Características Básicas

- RAM interna de uso geral
  - a) 128 bytes (USO Geral);
  - b) 128 bytes (Registradores Especiais);
- ROM interna de 4 Kbytes;
- 4 Ports de I/O;
- 2 *Timers* de 16 bits;
- 1 Interface Serial;
- Capacidade de Endereçamento ROM: 64K
- Capacidade de Endereçamento RAM: 64K

# 8051 - Características Básicas

- Alguns já possuem conversor A/D;
- Alguns possuem saída PWM;

# Arquitetura Interna





# Pinagem



# Ports

- **P0**

- Compartilha I/O e a multiplexação dos barramentos de dados e endereços (LSB)

- **P1**

- Somente I/O

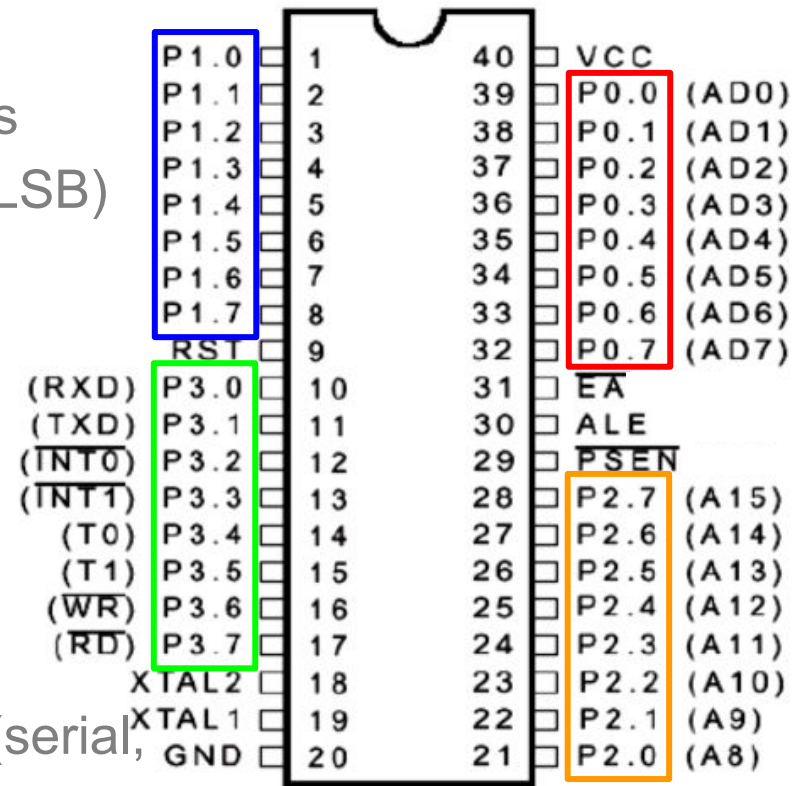
- **P2**

- Barramento de endereços (MSB)

- **P3**

- Compartilha I/O e funções especiais (serial, *timer*, interrupções, acesso a memórias externas)

AT89C51 (DIP40)



# Port P0

- Quando não for utilizada memória externa:
  - De propósito geral
- Quando for utilizada memória externa:
  - Multiplexado no tempo entre dados e endereços (menos significativos)
    - Maneira de economizar pinos no chip

# Port P0

- Quando não for utilizada memória externa:
  - De propósito geral
- Quando for utilizada memória externa:
  - Multiplexado no tempo entre dados e endereços (menos significativos)
    - Maneira de economizar pinos no chip
  - Mas como é feita a multiplexação?





- [illegible]

# Port P2

- Quando não for utilizada memória externa:
  - De propósito geral.
- Quando for utilizada memória externa:
  - Utilizado para endereçar os MSBs da RAM externa ou ROM/EPROM/EEPROM externa.

# Port P1

- De propósito geral como Entrada e Saída (I/O)
  - Pode-se ler ou escrever neste port
- Em microcontroladores que possuem o terceiro *timer* e for utilizado, 2 pinos são comprometidos.
- Em microcontroladores que possuem A/D interno e este for utilizado, 1 pino do port P1 por canal de A/D é comprometido.

# Port P3

- De propósito geral como Entrada e Saída (I/O)
  - Se nenhum **periférico interno** for utilizado;
  - Se nenhuma **interrupção externa** for utilizada;
  - Se não for utilizada **RAM externa**
    - \WR
    - \RD
- Caso contrário estará comprometido conforme a tabela a seguir:



# Port P3

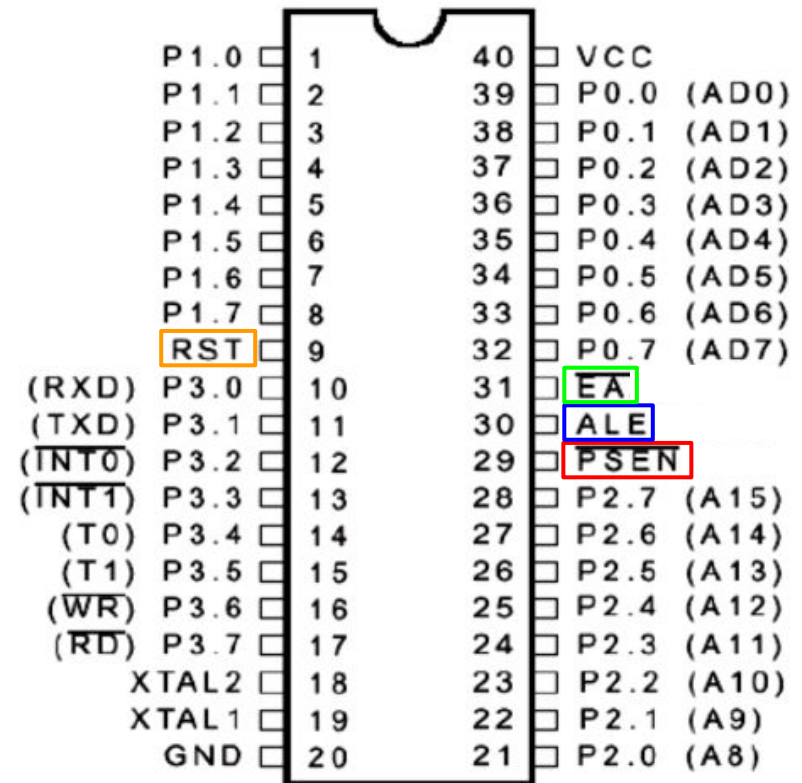
- Periféricos do P3

Periférico	Pino do Periférico	Pino	Função
Serial	RXD	P3.0	Receptor de dados serial
	TXD	P3.1	Transmissor de dados serial
Interrupção	INT0	P3.2	Para algum evento externo interromper o uC
	INT1	P3.3	Para algum evento externo interromper o uC
Timer 0	T0	P3.4	Quando se quer que o timer0 se torne um contador de eventos externos
Timer 1	T1	P3.5	Quando se quer que o timer1 se torne um contador de eventos externos
Controle de Memória	WR	P3.6	Sinaliza que o uC irá escrever na RAM,
	RD	P3.7	Sinaliza que o uC irá ler da RAM,

# Pinos de controle

- **\PSEN**
  - Memória externa de dados ou de programa
- **ALE**
  - Controle do multiplex do P0
- **RST**
  - Reset
- **\EA**
  - Memória de programa interna ou externa

AT89C51 (DIP40)



# Pinos de Controle

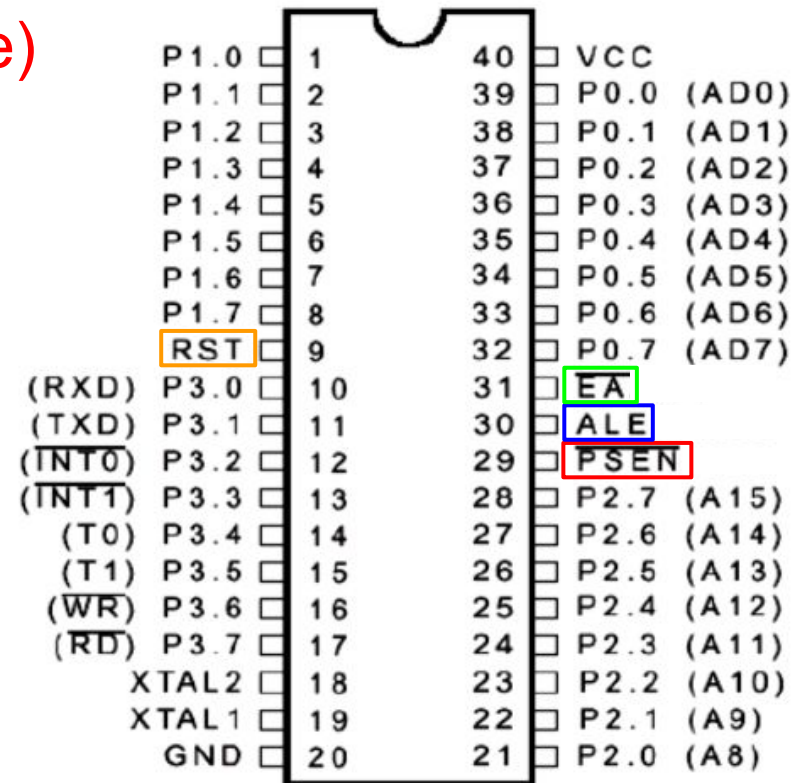
- **\PSEN (Program Strobe Enable)**

- Aciona memória ROM externa
  - Buscar instruções
  - Buscar dados em tabelas

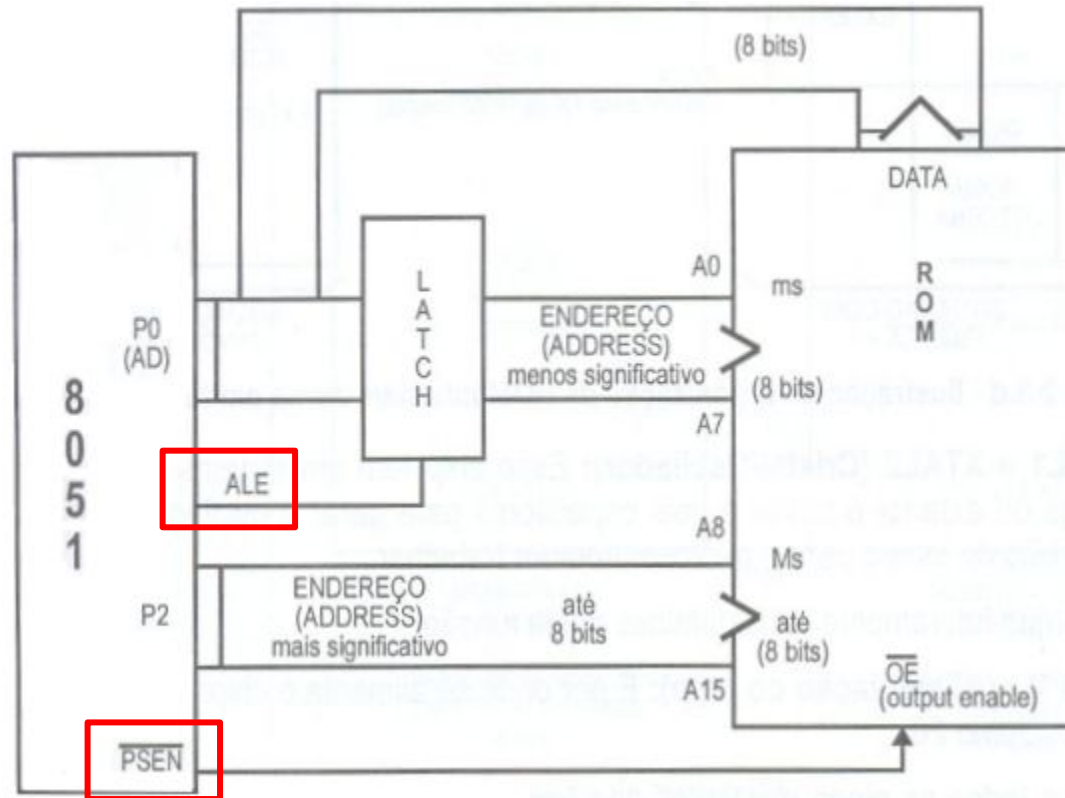
- **ALE (Address Latch Enable)**

- Comanda a demultiplexação de dados e endereços (LSB) do P0.

AT89C51 (DIP40)



# Pinos de Controle



# Pinos de controle

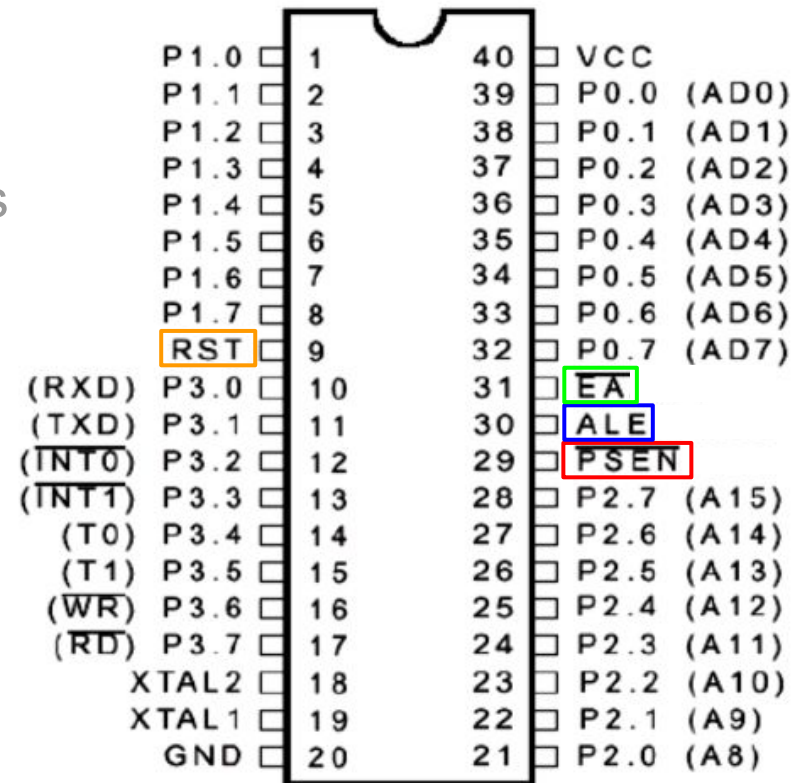
AT89C51 (DIP40)

- **RST**

- Disparador de *reset* do uC.
- Deve estar em 1 por ao menos dois ciclos de clock

- **\EA**

- Determina se será usada somente memória ROM externa ou memória ROM interna + memória de ROM externa
  - Pino = 1: primeiro memória de programa interna, depois externa
  - Pino = 0: somente memória de programa externa

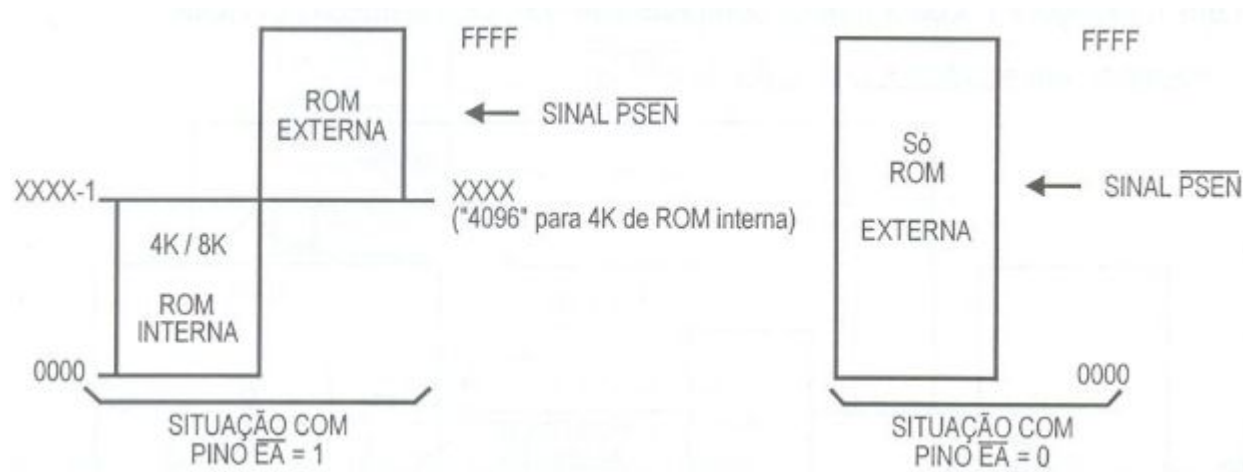


# Pinos de controle

- $\overline{EA}$  → Memória de programa
  - Interna:
    - Pode ser Flash ou ROM: 0, 4, 8, 12, 20, **32\*** ou 64 Kbytes
  - Externa
    - Até 64 Kbytes

\*AT89C5131A

Ilustração:



# Demais Pinos

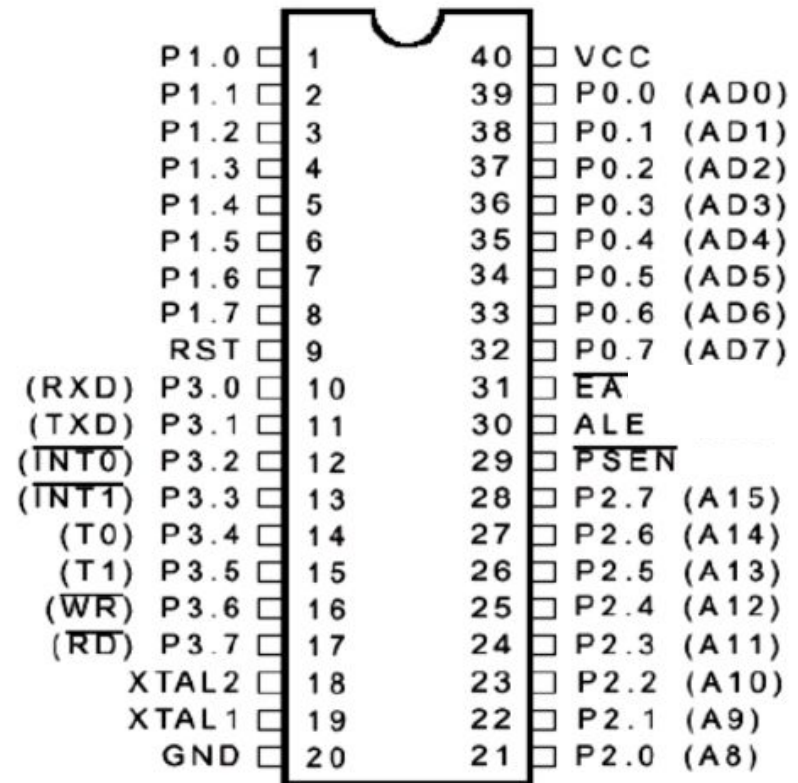
- Alimentação

- Vcc
- GND (ou Vss)

- Cristal Oscilador

- XTAL1
- XTAL2

AT89C51 (DIP40)





***Have a break,  
have a KitKat®.***



# Organização das Memórias

# Memórias

- Como o rascunho e o livro de receitas estão organizados?

# Memórias

- Memória de programas separada da memória de dados
- Existe memória RAM interna (MOV) -> Família 8051
  - Podendo gerenciar RAM externa (MOVX)
- Pode existir memória ROM interna (MOVC)
  - Podendo gerenciar ROM externa (MOVC)

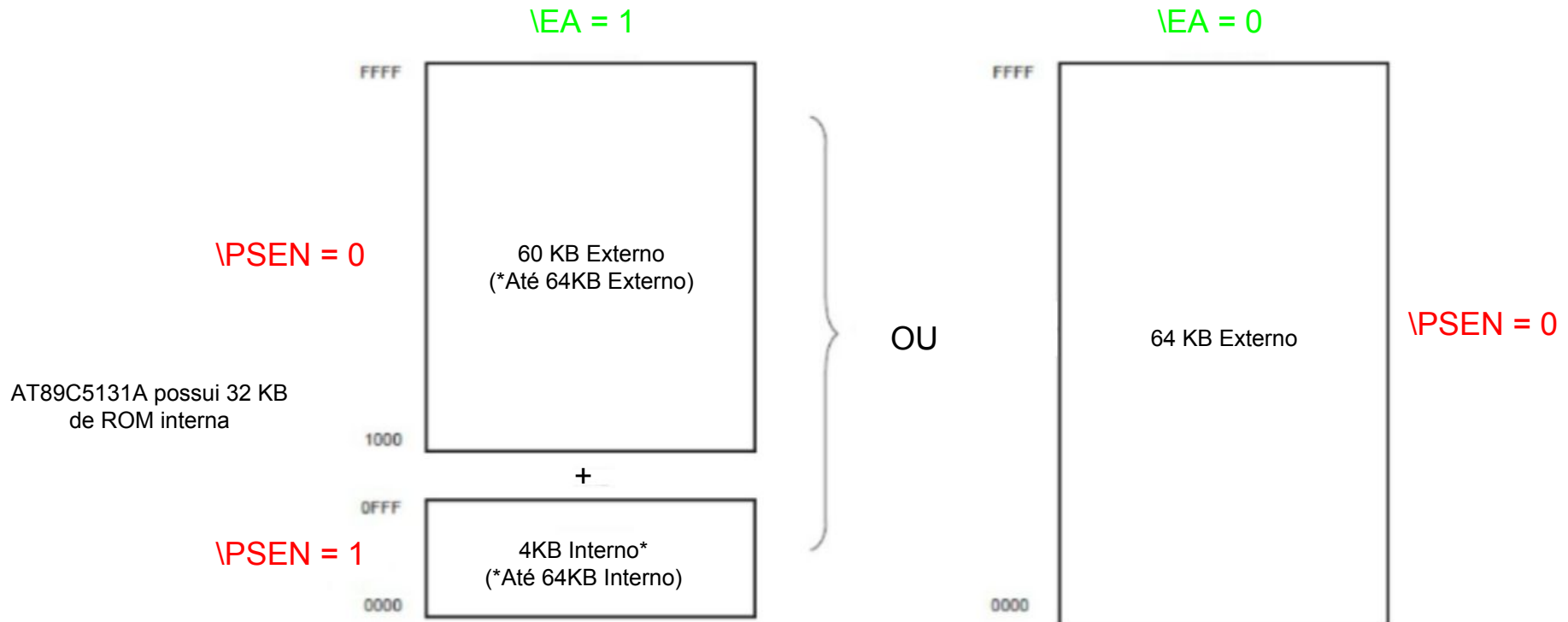
# Memória de Programa (ROM)

Pino **\EA** (External Access - Entrada)

- **\EA = 1** => Mem Interna + Externa
- **\EA = 0** => Mem Externa Apenas

Pino **\PSEN** (Program Strobe Enable - Saída)

- **\PSEN = 1** => Mem Interna
- **\PSEN = 0** => Mem Externa



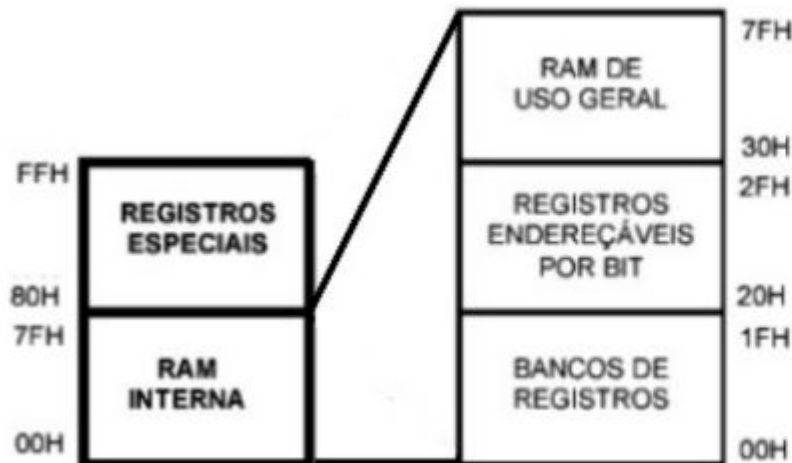
# Memória de Dados (RAM)

Pino **RD** (Read - Saída)

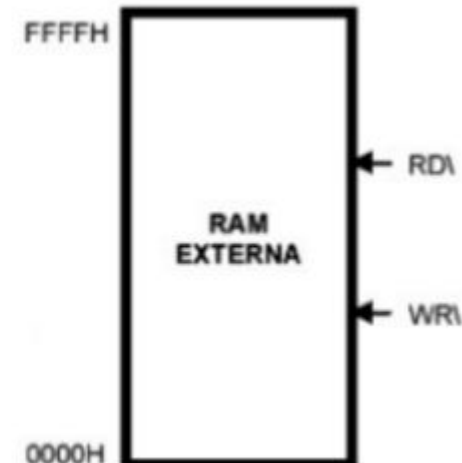
- Leitura da RAM Externa

Pino **WR** (Write - Saída)

- Escrita da RAM Externa



MOV



MOVX

# Memória RAM interna

- Espaço de 256 bytes (ou mais dependendo o modelo) dividido em:

1) Registradores de uso geral

2) Bit e/ou byte endereçável

3) Byte endereçável

RAM de uso geral



4) Registradores de Funções Especiais (SFR)

# Registradores de Uso Geral

- 4 bancos de registradores (0..3), cada um com 8 registradores de 8 bits (R0..R7)
- A seleção do banco de registradores é feita através dos bits RS0 e RS1 do PSW
- Também pode ser byte endereçável
- Varia de 0 a 1Fh



RS1	RS2	Banco	R0..R7
1	1	3	18h..1Fh
1	0	2	10h..17h
0	1	1	08h..0Fh
0	0	0	00h..07h



# Região Bit-endereçável

- Entre 20h a 2Fh (acima dos bancos de registradores)
- Acesso por byte (SETB 24.0) ou bit (SETB 20h)

Registro	Endereços Individuais dos Bits							
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
20H	07H	06H	05H	04H	03H	02H	01H	00H
21H	0FH	0EH	0DH	0CH	0BH	0AH	09H	08H
22H	17H	16H	15H	14H	13H	12H	11H	10H
23H	1FH	1EH	1DH	1CH	1BH	1AH	19H	18H
24H	27H	26H	25H	24H	23H	22H	21H	20H
25H	2FH	2EH	2DH	2CH	2BH	2AH	29H	28H
26H	37H	36H	35H	34H	33H	32H	31H	30H
27H	3FH	3EH	3DH	3CH	3BH	3AH	39H	38H
28H	47H	46H	45H	44H	43H	42H	41H	40H
29H	4FH	4EH	4DH	4CH	4BH	4AH	49H	48H
2AH	57H	56H	55H	54H	53H	52H	51H	50H
2BH	5FH	5EH	5DH	4CH	5BH	5AH	59H	58H
2CH	67H	66H	65H	64H	63H	62H	61H	60H
2DH	6FH	6EH	6DH	6CH	6BH	6AH	69H	68H
2EH	77H	76H	75H	74H	73H	72H	71H	70H
2FH	7FH	7EH	7DH	7CH	7BH	7AH	79H	78H



# Região Byte-Endereçável

- Entre 30h a 7Fh (acima da região bit-endereçável)
- Região interna de uso geral

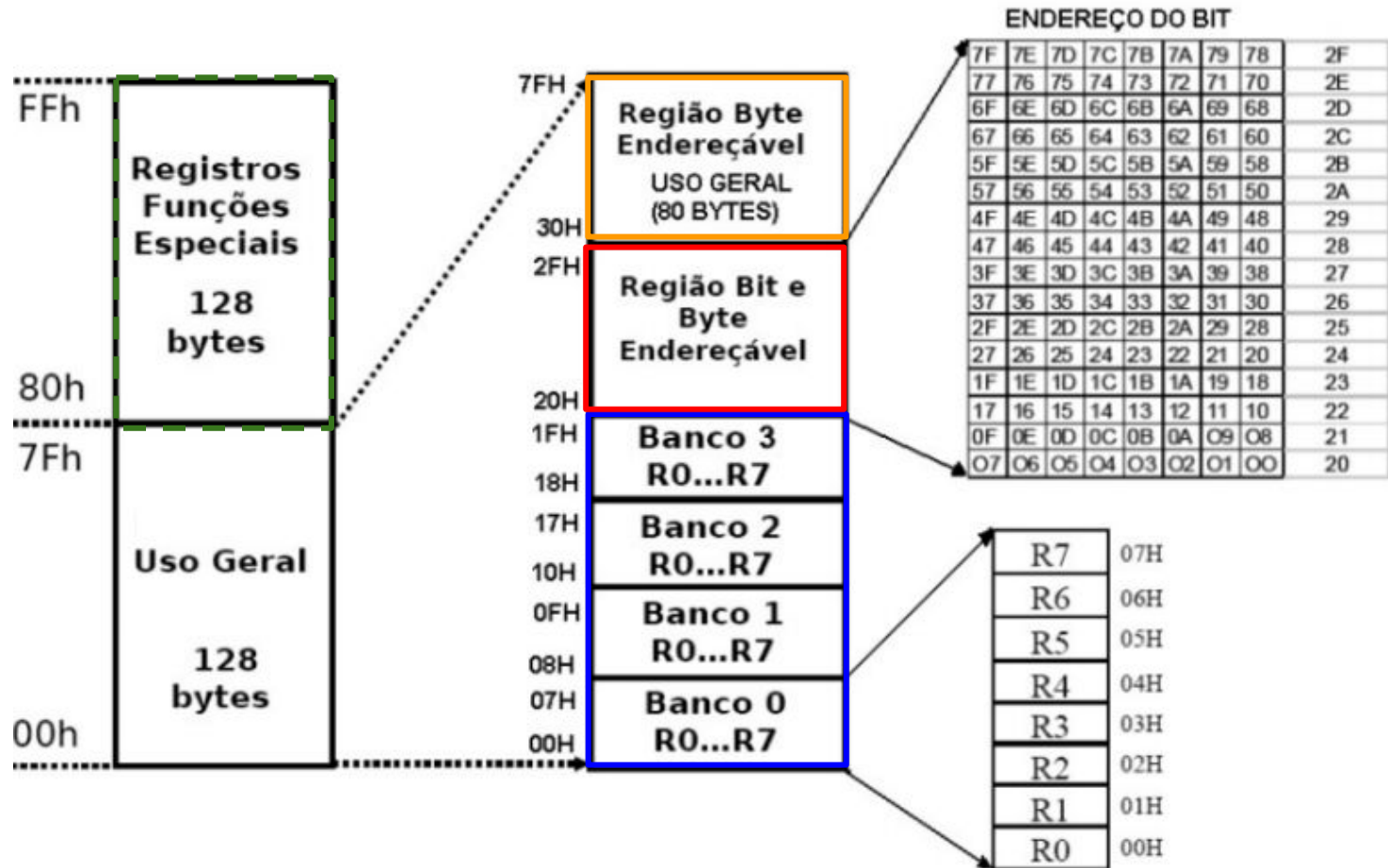
# Memória RAM interna

- E agora onde nós estamos???



# Memória RAM interna

- Mapa da memória interna



# Registradores de Funções Especiais

- Têm apelido
- Têm endereço absoluto
- Têm função específica dentro do uC

## Exemplo:

O **port P1** tem o endereço do byte **90h** e é o espelho do que acontece externamente com o **port P1**. Ver algo no **port P1** é olhar no registrador **P1**; alterar os valores do **port P1** é alterar os valores deste registrador;

# Registradores de Funções Especiais

- Agora eu vou ter que decorar todos os registradores de funções especiais e suas funções?
- E ainda seus endereços?

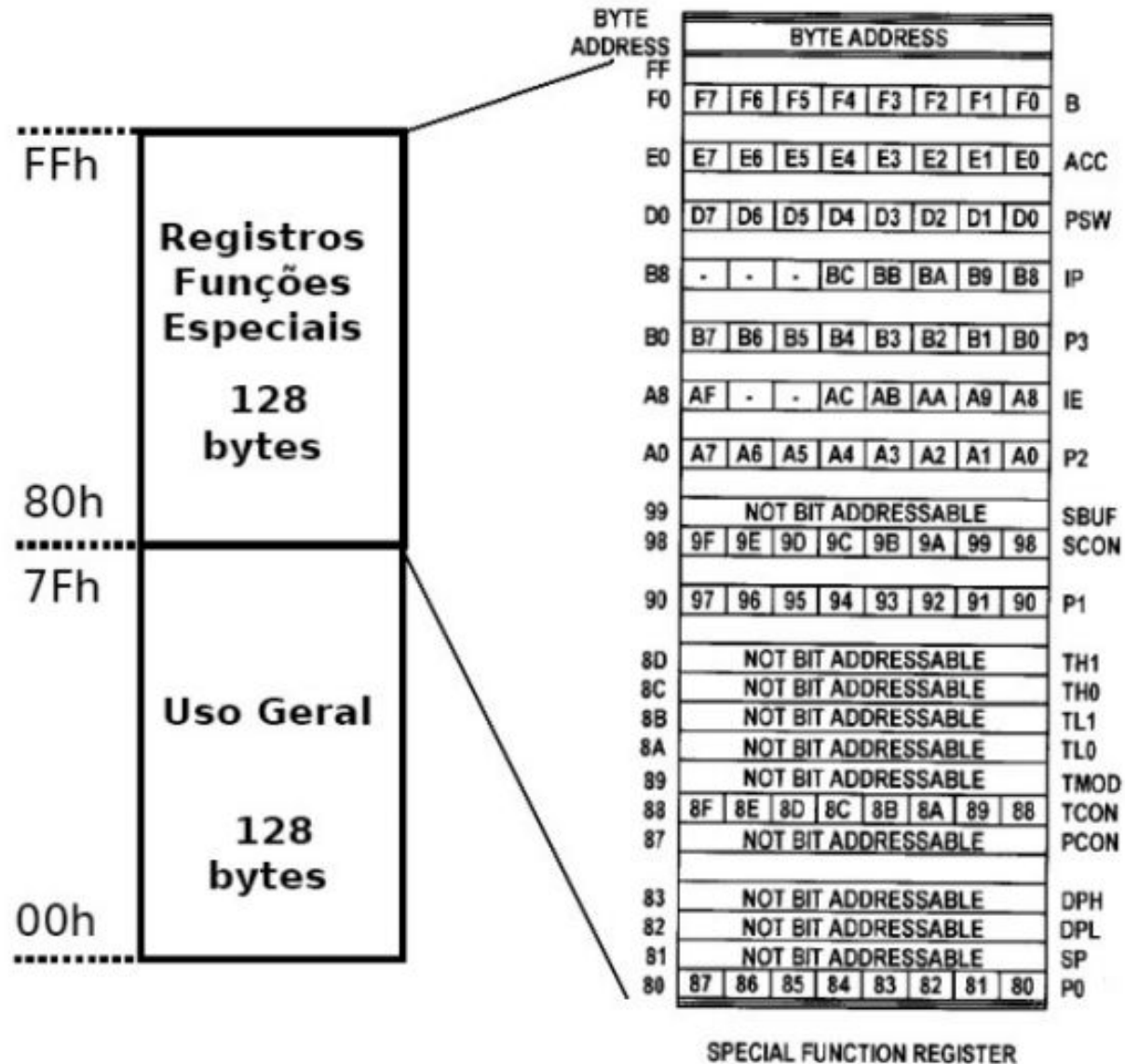


# Registradores de Funções Especiais

R: Quase isso! Apenas os nomes e as funções que cada um desempenha.

E quais são os SFRs?

# Registradores de Funções Especiais



# Registradores de Funções Especiais

- **Acc** (E0h): acumulador - registrador utilizado como operando em operações aritméticas e lógicas:

ADD        A, R1

ANL        A, #43h

MOV        A, R1



ACC	ACC7	ACC6	ACC5	ACC4	ACC3	ACC2	ACC1	ACC0
E0h	E7h	E6h	E5h	E4h	E3h	E2h	E1h	E0h



# Registradores de Funções Especiais

- **B (F0h)**: registrador de uso geral utilizado nas operações de multiplicação e divisão:

MUL      AB

DIV      AB

B	—	—	—	—	—	—	—	—
F0h	E7h	E6h	E5h	E4h	E3h	E2h	E1h	E0h

# Registradores de Funções Especiais

- **PSW** (D0h): registrador de estado da última operação lógica ou aritmética realizada

	CY	AC	F0	RS1	RS0	OV	-	P
PSW	PSW.7	PSW.6	PSW.5	PSW.4	PSW.3	PSW.2	PSW.1	PSW.0
D0h	D7h	D6h	D5h	D4h	D3h	D2h	D1h	D0h

# Registradores de Funções Especiais

## ● PSW (D0h):

- CY: *carry* ou “vai-um” quando a operação causa um valor maior do que FFh. Também é utilizado como acumulador para operações com bits:

ADD A, #01h ;CY=1 se A=FFh antes

ANL C, P1.0 ;CY=C ^ P1.0

- AC: *auxiliary carry* ou “vai-um” do nibble inferior do acumulador. Usado na aritmética BCD;
- F0: Flag 0 - flag de uso geral, sem função especial
- RS1 e RS0: *Registers Set* - seleciona o conjunto de registradores em uso
- OV: *Overflow* - bit ligado quando o valor da operação extrapola +128 ou -128
- P: *Parity* - paridade ímpar do acumulador

# Registradores de Funções Especiais

- **SP (81h): *Stack Pointer*** - ponteiro do topo da pilha de endereços de retorno
  - Utilizado nas subrotinas e atendimento a interrupções:
    - PUSH: SP é incrementado depois o dado é armazenado;
    - POP: o dado é lido e depois o SP é decrementado.



- Pilha definida na RAM interna
  - Após o reset,  $SP \leq 07h$
  - Na primeira busca estará apontando para o registrador R0 do banco 1 de registradores.
  - Funciona no sistema LIFO (*Last In First Out*)

# Registradores de Funções Especiais

- **DPH** (83h) e **DPL** (82h): *Data Pointer High & Low* - formam o ponteiro de 16 bits (**DPTR**) que aponta um endereço para acesso às memórias externas

```
MOV      DPTR, #2000h
```

```
MOVBX    A, @DPTR
```

# Registradores de Funções Especiais

- **P0** (80h), **P1** (90h), **P2** (A0h): *ports* de I/O de uso geral.

P0	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
80h	87h	86h	85h	84h	83h	82h	81h	80h

P1	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
90h	97h	96h	95h	94h	93h	92h	91h	90h

P2	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0
A0h	A7h	A6h	A5h	A4h	A3h	A2h	A1h	A0h

# Registradores de Funções Especiais

- **P3 (B0h):** Port de I/O de uso geral ou com funções especiais:
  - P3.0 - RxD/data - recepção serial assíncrona ou E/S de dados síncronos;
  - P3.1 - TxD/clock - transmissão porta serial assíncrona ou saída de clock para dados síncronos;
  - P3.2 - INT0 - entrada da interrupção 0 ou bit de controle para temporizador/contador 0;
  - P3.3 - INT1 - entrada da interrupção 1 ou bit de controle para temporizador/contador 1;
  - P3.4 - T0 - entrada de clock externo para o temporizador/contador 0;
  - P3.5 - T1 - entrada de clock externo para o temporizador/contador 1;
  - P3.6 - WR - sinal de escrita na memória de dados externa;
  - P3.7 - RD - sinal de leitura na memória de dados externa.

P3	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0
B0h	B7h	B6h	B5h	B4h	B3h	B2h	B1h	B0h

# Registradores de Funções Especiais

- **PCON** (87h): *Power Control Register* - Só tem na versão CMOS, controla o consumo de energia do uC e outras informações:
  - **SMOD**: emprestado para o periférico serial para ajuste do *baud-rate*
  - **GF1** e **GF2**: *General Flags* 0 e 1 - bits de uso geral
  - **PD**: *Power Down Mode* - o uC para, mantendo os valores da RAM interna e dos ports só saindo quando o *reset* for ativado
  - **IDL**: *Idle Mode* - semelhante ao *PD*, mas sai por interrupções já liberadas

PCON	SMOD	-	-	-	GF1	GF0	PD	IDL
87h								





Professor, diga que  
acabou???

Na verdade não! Mas os  
outros SFRs serão vistos  
quando os respectivos  
periféricos forem  
estudados.