



Instituto Federal de Santa Catarina  
Área de Telecomunicações

**MIC29004 – Microprocessadores**

Prof. Roberto de Matos  
roberto.matos@ifsc.edu.br

São José, agosto de 2013.

Aviso de direitos Autorais:  
Transparências baseadas no trabalho do  
[Prof. Eduardo Augusto Bezerra](#)

**Microcontroladores**

MIC29004 – Microprocessadores

**Microcontroladores**

Componentes básicos de sistemas computacionais processados:

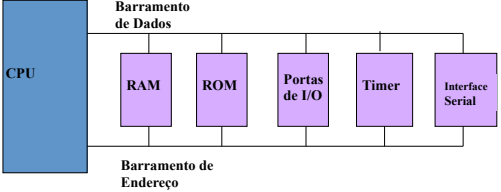
- CPU
- Memória de dados e programa
- Sistema de entrada/saída

MIC29004 – Microprocessadores

**Microcontroladores**

Microprocessadores de Propósito Geral

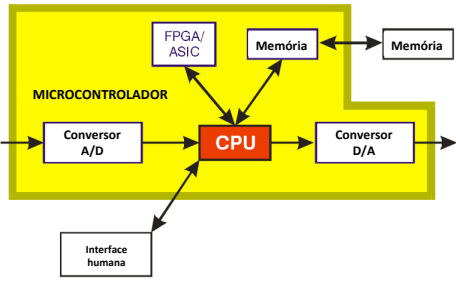
Sem RAM, ROM ou dispositivos de I/O



**Vantagem:** flexibilidade, sistema expansível ;  
**Desvantagem:** custo, roteamento de placa e dimensões do circuito.

MIC29004 – Microprocessadores

**Microcontroladores**



MIC29004 – Microprocessadores



### Microcontroladores: aplicações

MIC29004 – Microprocessadores

### FPGA: aplicações

**Figure 2. Where will the computers be?**

Where has computer science focused?

Where are the processors?

MIC29004 – Microprocessadores

### FPGA: aplicações

**Figure 3. Projected CPU shipments in 2000.**

Microcontroller Solutions 8,288,300,000				
Cores ???	Microcontroller units 7,257,000,000	Embedded Microprocessor units 281,300,000	Digital signal processor 600,000,000	Computational Microprocessor units 150,000,000
16 bit (ARM)	4 bit 1,680,000,000	8 bit 20,200,000	Texas Instruments	x86
32 bit (MIPS)	8 bit 4,770,000,000	16 bit 108,000,000	Advanced Micro Devices	PowerPC
ASSP	16 bit 764,000,000	32 bit 153,100,000		SPARC
ASIC	32 bit 43,000,000			

Source: DARPA

MIC29004 – Microprocessadores



### Microcontroladores: aplicações

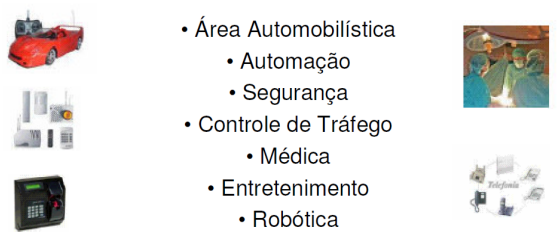
MIC29004 – Microprocessadores

### Microcontroladores: aplicações

MIC29004 – Microprocessadores

**Microcontroladores: aplicações**

- Área Automobilística
  - Automação
  - Segurança
- Controle de Tráfego
  - Médica
- Entretenimento
  - Robótica



MIC29004 – Microprocessadores

**Microcontroladores: aplicações**

- Embarcados em:
  - Sistemas automotivos
  - Aviônicos
  - Brinquedos
  - Dispositivos médicos
  - Eletrodomésticos
- Bilhões de unidades



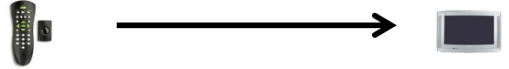
MIC29004 – Microprocessadores

**Microcontroladores: aplicações**


- Produtos de uso pessoal: Celulares, pagers, relógios, gravadores portáteis, calculadoras, câmeras fotográficas
- Laptops: mouse, teclado, modem, fax, placa de som, carregador de bateria
- Domótica: tranca eletromagnética, despertador, termostato, ar condicionado, controle remoto de TV, secador de cabelo, aparelho de DVD, geladeira, lavadora de roupa/louça, forno de microondas

MIC29004 – Microprocessadores


**Microcontroladores: aplicações**



Outro exemplo corriqueiro é o despertador

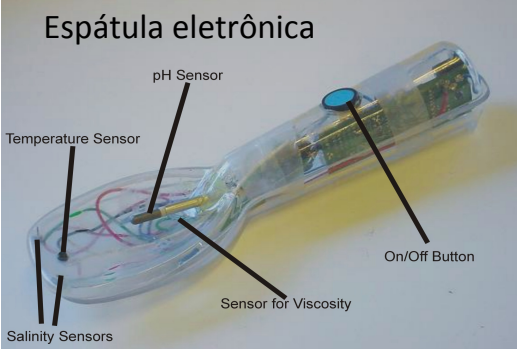


Outro exemplo é a parte de segurança



**Microcontroladores: aplicações**

**Espátula eletrônica**



**Microcontroladores: aplicações**

**Espátula eletrônica**



**Microcontroladores: aplicações**



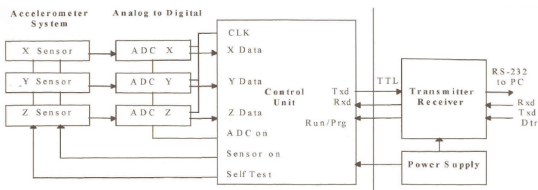
MIC29004 – Microprocessadores

**Microcontroladores: aplicações**

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

MIC29004 – Microprocessadores

**Microcontroladores: aplicações**



MIC29004 – Microprocessadores

**Microcontroladores: aplicações**

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

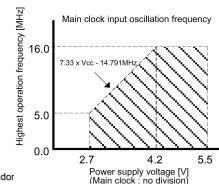
MIC29004 – Microprocessadores

**Microcontroladores: Limitações**

MIC29004 – Microprocessadores

**Limitações**

- Limitação importante: velocidade de processamento
- Não adequados para aplicações com tempo de resposta abaixo de poucos microssegundos
- Ambiente de desenvolvimento (compiladores, montadores, linkers, bibliotecas, plataformas de software e hardware, ...) – pode ser de uso complexo e custo elevado
- Tamanho dos programas e dados (recursos de memória escassos)
- Programas sequenciais



MIC29004 – Microprocessador

### Limitações

**PIC**

- Disponibilidade em encapsulamento DIP para uso direto em placas de prototipação
- Valores na ordem de US\$1 a US\$9

• **Limitação:** Custo das ferramentas – Compilador ~US\$200; Debug ~US\$150.

**AVR**

- Ferramentas gratuitas (gcc)
- IDE disponível para Windows, Mac e Linux, incluindo debug
- AVR-Dragon da Atmel custa em torno de US\$50 e pode ser utilizado para programação e depuração
- **Limitação:** poucas famílias de dispositivos disponíveis (pouca variedade) ao se comparar com o PIC

MIC29004 – Microprocessadores

### Limitações

**Microcontrolador:**

**Vantagens:**

- Mais versátil que CPLD, especialmente para aplicações analógicas (A/D, D/A).
- Facilidade para implementar algoritmos complexos e funções densas

**Desvantagens:**

- Temporização difícil de ser determinada para aplicações mais complexas (em C)
- Normalmente, menos desempenho em tempo de execução do que CPLD

**CPLD:**

**Vantagens:**

- Temporização eficiente e precisa

- Normalmente, melhor desempenho em tempo de execução do que microcontroladores

**Desvantagens:**

- Limitação para aplicações complexas e lógicas densas

MIC29004 – Microprocessadores

### Limitações

#### Programa Exemplo: Loop

```
/* pulses pin PORTB<3>
eight times */

pulse:
movlw 0x08
movwf counter

pulse_lp0:
bsf PORTB, 3
bcf PORTB, 3
decfsz counter, F
goto pulse_lp0
return
```

Assembly

```
/* pulses pin PORTB<3>
eight times */

void pulse ()
{
int i;
for (i=0; i<8; i++)
{ output_high(PIN_B3);
output_low(PIN_B3);
}
return;
}
```

C

MIC29004 – Microprocessadores

### Limitações

#### Compilador Ineficiente

```
/* pulses pin PORTB<3>
eight times */

0000: movlw 0x8
0001: movwf 0x20
0002: bsf 0x6,0x3
0003: bcf 0x6,0x3
0004: decfsz 0x20
```

Assembly escrito pelo desenvolvedor

```
/* pulses pin PORTB<3> eight
times */
0005: CLRF 21
0006: MOVF 21,W
0007: SUBLW 07
0008: BTSS 03,0
0009: GOTO 014
000A: BSF 03,5
000B: BCF 06,3
000C: BCF 03,5
000D: BSF 06,3
000E: BSF 03,5
000F: BCF 06,3
0010: BCF 03,5
0011: BCF 06,3
0012: INCF 21,F
0013: GOTO 006
```

Assembly gerado pelo compilador

MIC29004 – Microprocessadores

### 8051: Introdução

MIC29004 – Microprocessadores

### 8051 – Introdução

Características Básicas:

- CPU de 8 bits;
- endereça 64 Kb de memória de programa externa;
- endereça 64 Kb de memória de dados externa;
- 4 kbytes de memória ROM interna para programas;
- 128 bytes de memória RAM interna para dados;
- 4 portas de entrada e saída (8 pinos cada);
- 5 vetores de interrupção:
  - 2 níveis de prioridade
  - 2 externas, 2 dos timers/counters e 1 da porta serial

MIC29004 – Microprocessadores

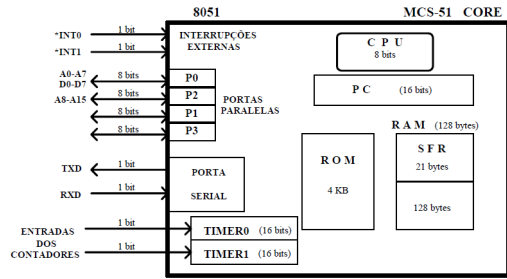
### 8051 – Introdução

#### Características Básicas:

- 111 Instruções:
  - 1 ciclo → 64 → 58%
  - 2 ciclos → 45 → 40%
  - 4 ciclos → 2 → 2%
  - 1 byte → 49 → 44%
  - 2 bytes → 46 → 41%
  - 3 bytes → 16 → 15%
- 98% de 1 ou 2 ciclos → velocidade
- 85% de 1 ou 2 bytes → compacto

MIC29004 – Microprocessadores

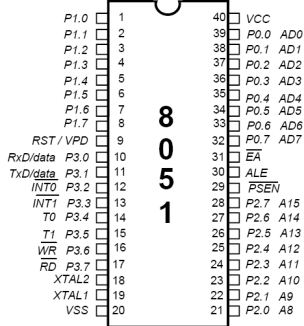
### 8051 – Introdução



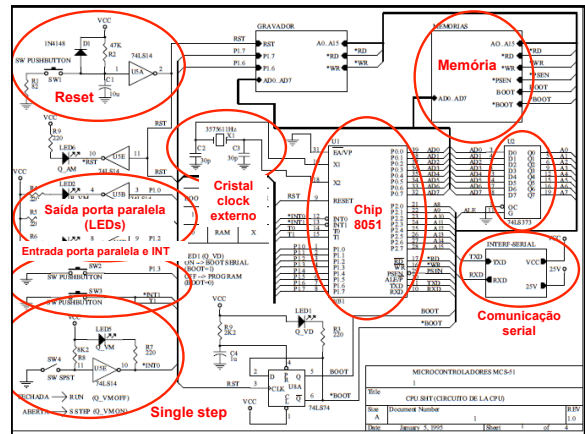
• 5 Interrupções (2 externas, 2 dos timers/counters e 1 da porta serial)

MIC29004 – Microprocessadores

### 8051 – Introdução



MIC29004 – Microprocessadores



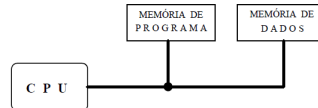
## 8051: Memória

MIC29004 – Microprocessadores

### 8051 – Memória

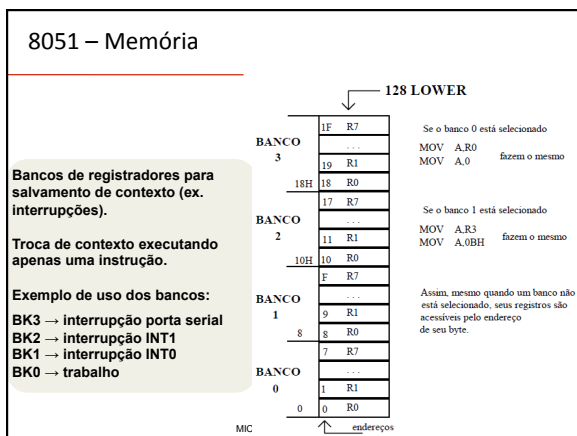
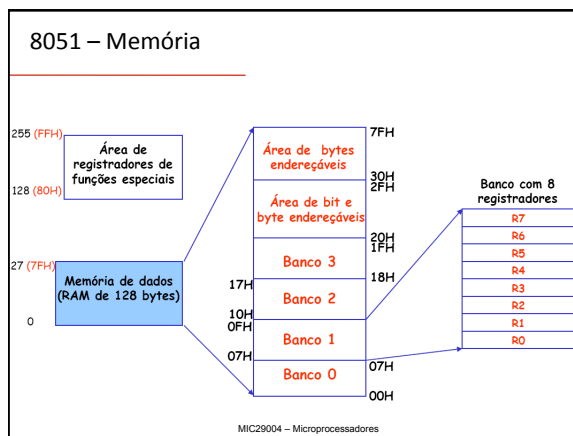
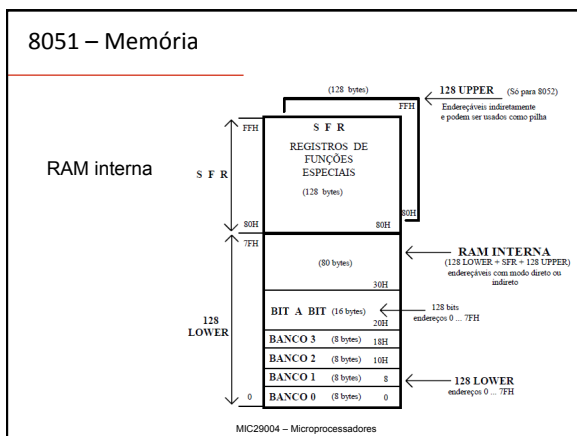
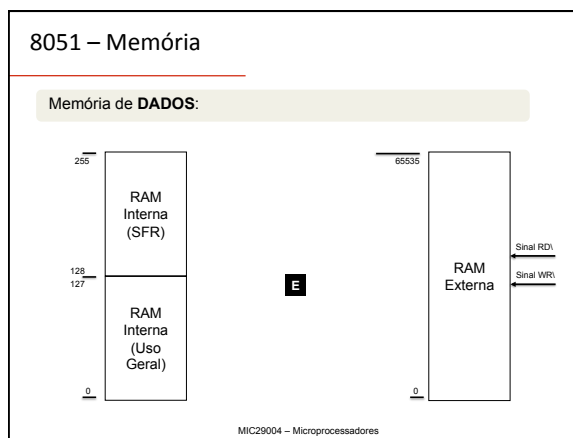
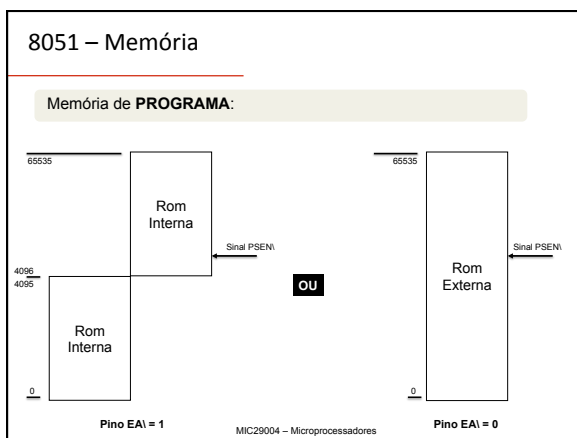
#### Arquitetura Harvard

- 64 KB de Memória de Programa (PC=16 bits)
- 64 KB de Memória de Dados



4 KB de ROM interna ativada pelo pino EA (*External Access Enable*):  
 → se EA=0 → 64 KB de programa externo  
 → se EA=1 → 4 KB de ROM interna e 60 KB de programa externo

MIC29004 – Microprocessadores



### 8051 – Memória

**RAM interna com 256 bytes com nomes simbólicos para acesso direto**

**128 bytes (dos 256) reservados para Registros de Funções Especiais (SFR).**

**8051 possui conjunto mínimo, outros '51 podem possuir mais SFRs**

BIT	SÍMBOLO	NOME	ENDEREÇO
*	ACC	Acumulador	E0
*	B	Registro B	F0
*	PSW	Palavra de Status (Program Status Word)	D0
*	SP	Ponteiro da Pilha (Stack Pointer)	81
*	DPH	Ponteiro de Dados (high) (Data Pointer High)	63
*	DPL	Ponteiro de Dados (low) (Data Pointer Low)	62
*	P3	Porta 3	80
*	P2	Porta 2	A0
*	P1	Porta 1	90
*	P0	Porta 0	80
*	IP	Prioridade de Interrupção (Interrupt Priority)	88
*	IE	Habilitação de Interrupção (Interrupt Enable)	A8
*	TMOD	Timer/Counter Mode	89
*	TCN	Timer/Counter Control	88
*	TH1	Timer/Counter 1 (MSB)	8D
*	TL1	Timer/Counter 1 (LSB)	8B
*	TH0	Timer/Counter 0 (MSB)	8C
*	TL0	Timer/Counter 0 (LSB)	8A
*	SCON	Controle da Porta Serial (Serial Control)	98
*	SBUF	Buffer da Porta Serial (Serial Buffer)	99
*	PCON	Controle de Energia (Power Control)	87

MIC2900



8051 – Memória

---

**PSW - Program Status Word - Bit Addressable**

7	6	5	4	3	2	1	0	
CY	AC	F0	RS1	RS0	OV	F1	P	D0H

CY	PSW.7	Carry flag
AC	PSW.6	Auxiliary carry flag
F0	PSW.5	Definido pelo usuário
RS1	PSW.4	Bit 1 do seletor de Register Bank
RS0	PSW.3	Bit 0 do seletor de Register Bank
OV	PSW.2	Overflow flag
F1	PSW.1	Definido pelo usuário
P	PSW.0	Flag de paridade. 1 = ímpar,

MIC29004 – Microprocessadores

8051 – Memória

---

**Mapa da RAM interna**

S F B															
FFH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	FFH
FEH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	FEH
FDH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	FDH
FCH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	FCH
FBH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	FBH
FAH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	FAH
F9H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	F9H
F8H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	F8H
F7H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	F7H
F6H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	F6H
F5H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	F5H
F4H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	F4H
F3H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	F3H
F2H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	F2H
F1H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	F1H
F0H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	F0H
0FH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0FH
0EH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0EH
0DH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0DH
0CH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0CH
0BH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0BH
0AH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0AH
09H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	09H
08H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	08H
07H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	07H
06H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	06H
05H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	05H
04H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	04H
03H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	03H
02H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	02H
01H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	01H
00H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	00H
13S LOWER															
7FH	7FH	7EH	-	-	-	-	-	-	-	-	-	-	-	-	7FH
7EH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	7EH
7DH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	7DH
7CH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	7CH
7BH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	7BH
7AH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	7AH
79H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	79H
78H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	78H
77H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	77H
76H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	76H
75H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	75H
74H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	74H
73H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	73H
72H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	72H
71H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	71H
70H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	70H
6FH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	6FH
6EH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	6EH
6DH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	6DH
6CH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	6CH
6BH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	6BH
6AH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	6AH
69H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	69H
68H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	68H
67H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	67H
66H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	66H
65H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	65H
64H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	64H
63H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	63H
62H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	62H
61H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	61H
60H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	60H

MIC29004 – Microprocessadores

8051 – Memória

---

- Endereçamento por Registro
- Endereçamento Imediato (#)
- Endereçamento Direto
  - IDATA
- Endereçamento Indireto (@)
  - IDATA
  - XDATA

8051 – Memória

---

- Endereçamento Registro:

MOV A, Rn

- R0 a R7 selecionado no banco (RS1 e RS0)
- Instrução de apenas 1 byte

8051 – Memória

---

- Endereçamento Imediato (#):

MOV A, #data  
MOV DPTR, #data16

- Valor constante acompanha a instrução

8051 – Memória

---

- Endereçamento Direto – IDATA:

MOV A, direct

- Endereçamento direto, ou seja, *direct* é o endereço de uma posição de memória RAM interna.

### 8051 – Memória

- Endereçamento Indireto (@) – IDATA:
   
$$\text{MOV A, @Ri}$$
- Endereçamento indireto a uma posição de memória RAM interna Ri = R0 ou R1

### 8051 – Memória

- Endereçamento Indireto (@) – XDATA:
   
$$\text{MOVX A, @DPTR}$$

### 8051 – Memória

#### Mnemônicos

MOV A, Rn
MOV A, direct
MOV A, @Ri
MOV A, #data
MOV DPTR, #data16

**Rn** - registrador R0 a R7 do banco correntemente selecionado.

**direct** - endereçamento direto, ou seja, direct é o endereço de uma posição de memória RAM interna.

**@Ri** - endereçamento indireto a uma posição de memória RAM interna Ri=R0 ou R1

**#data** - endereçamento imediato, onde data é uma constante de 8 bits incluída no corpo da instrução.

**#data16** - endereçamento imediato, onde data é uma constante de 16 bits incluída no corpo da instrução.

MIC29004 – Microprocessadores

### 8051: Diagramas de Tempo

MIC29004 – Microprocessadores

### 8051 – Diagramas de Tempo

#### Clock

Opções de ligação:

a) Cristal

b) Oscilador Externo

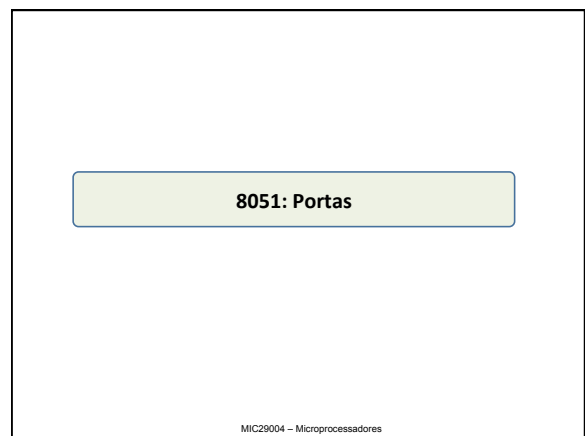
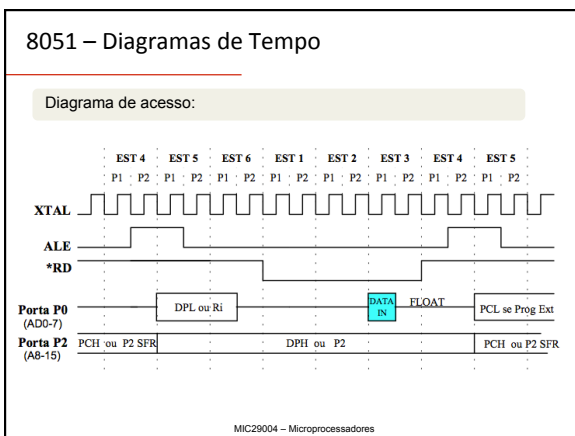
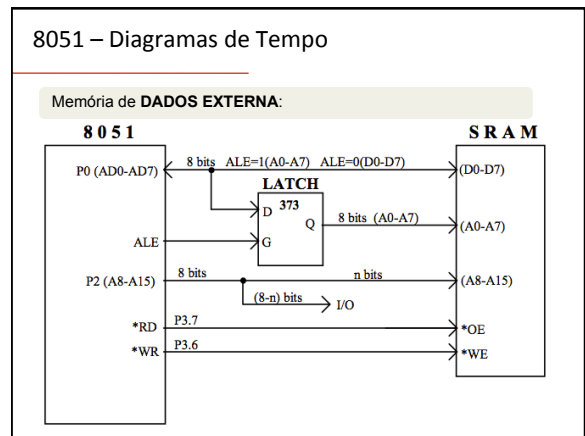
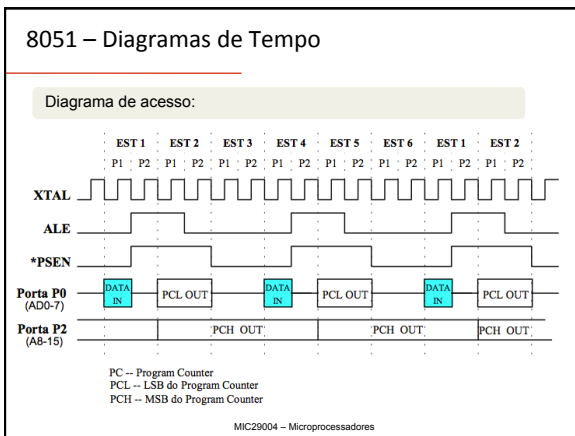
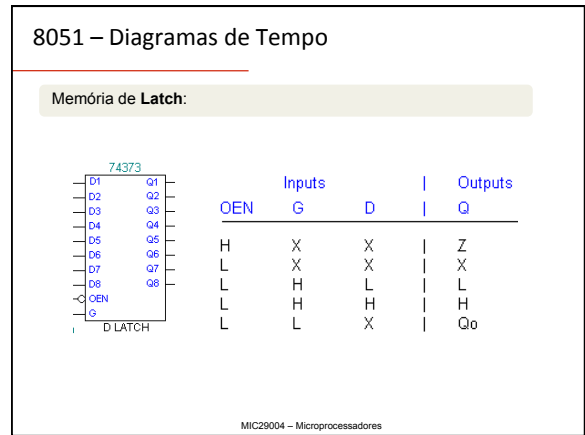
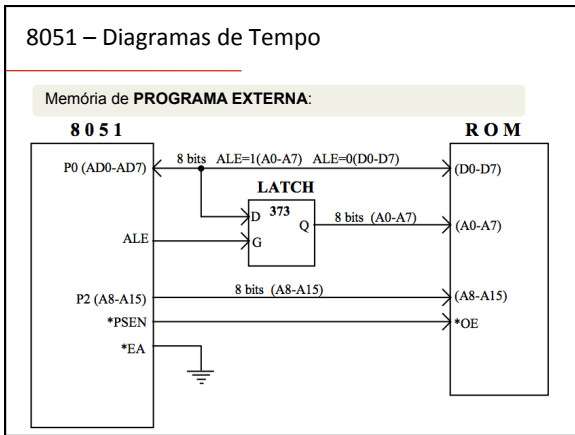
Observação: Os capacitores podem ser variados de  $\pm 10$  pF.

MIC29004 – Microprocessadores

### 8051 – Diagramas de Tempo

#### Reset

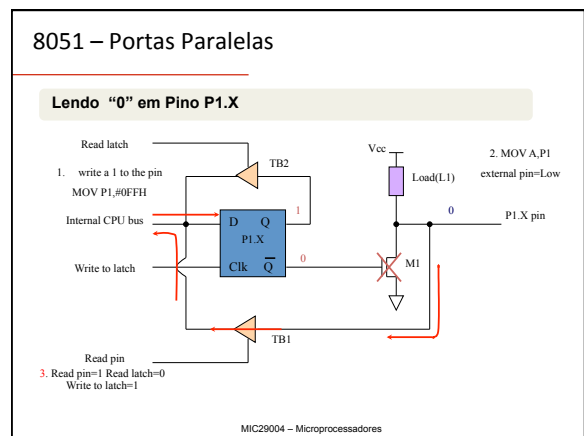
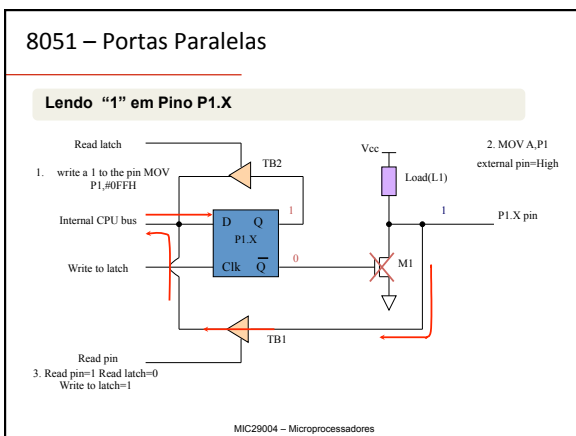
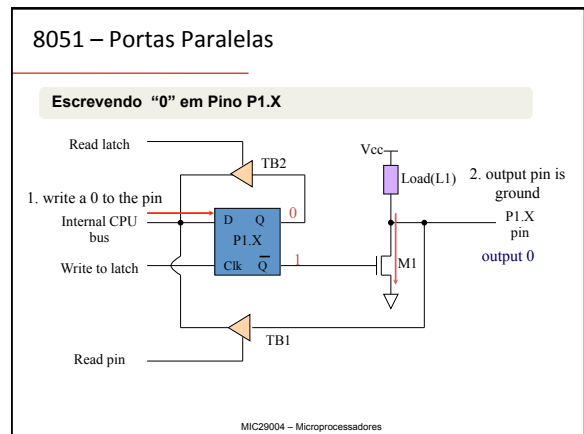
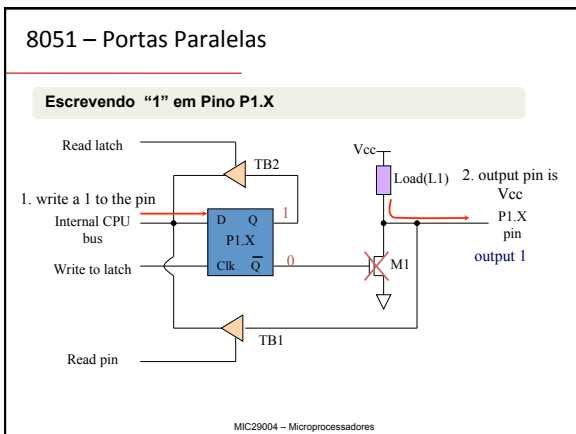
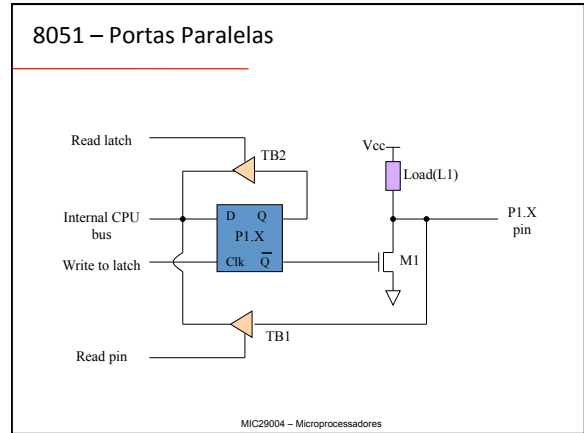
MIC29004 – Microprocessadores



### 8051 – Portas Paralelas

- Cada porta paralela é constituída por três partes:
  - um registro latch (SFR ==> P0,P1,P2,P3)
  - um driver de saída
  - um buffer de entrada

MIC29004 – Microprocessadores



### 8051 – Portas Paralelas

- Algumas instruções fazem a leitura do Pino (TB1) e outros da saída do Latch (TB2)
- Instruções que leem do latch (“lê-modifica-escreve”). Exemplos:

<b>ANL</b>	<b>AND lógico</b>	ex. ANL P1,A
<b>ORL</b>	<b>OR lógico</b>	ex. ORL P2,A
<b>XRL</b>	<b>XOR lógico</b>	ex. XRL P3,A
<b>CPL</b>	<b>complementa bit</b>	ex. CPL P3.0
<b>INC</b>	<b>incremento</b>	ex. INC P2
<b>DEC</b>	<b>decrementa</b>	ex. DEC P2
<b>DJNZ</b>	<b>decrementa e salta se não zero</b>	ex. DJNZ P3,LABEL
<b>MOV PX,Y,C</b>	<b>move bit de carry para bit Y da Port X</b>	
<b>CLR PX.Y</b>	<b>limpa bit Y da Port X</b>	
<b>SET PX.Y</b>	<b>seta bit Y da Port X</b>	

MIC29004 – Microprocessadores

### 8051 – Portas Paralelas

#### Lendo do Latch de Porta

Quando P1.0=1 → LED aceso  
Quando P1.0=0 → LED apagado

CPL P1.0 inverte o estado do led

MIC29004 – Microprocessadores

### 8051 – Portas Paralelas

#### Porta 0

a. Port 0 Bit

MIC29004 – Microprocessadores

### 8051: Interrupções

MIC29004 – Microprocessadores

### 8051 – Interrupções

- Tratar eventos assíncronos
- Mecanismo de Interrupção:
  - Salva o contexto atual do programa
  - Desvia Fluxo
  - Retorna ao fluxo normal
- Interrupções aninhadas (Prioridade)

MIC29004 – Microprocessadores

### 8051 – Interrupções

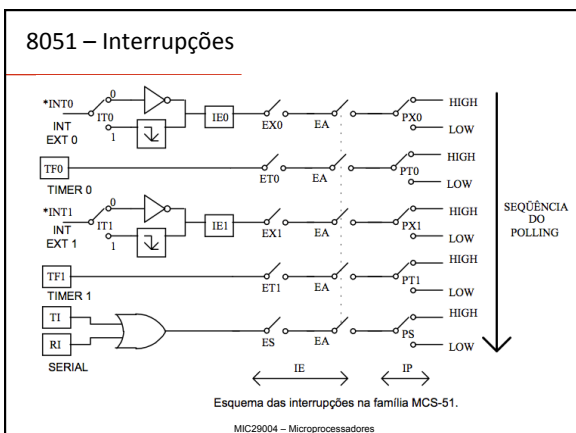
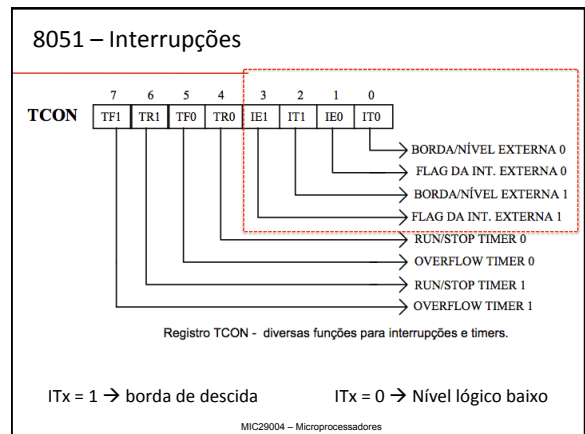
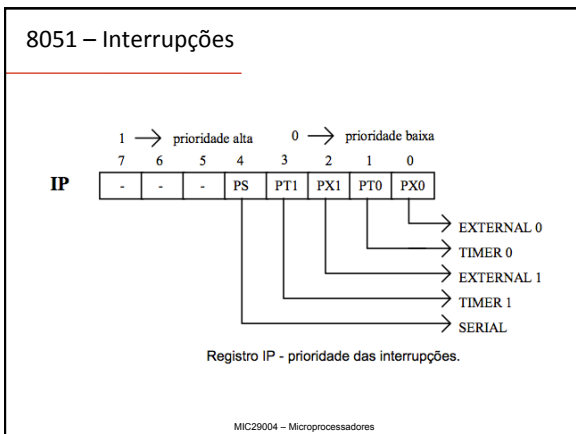
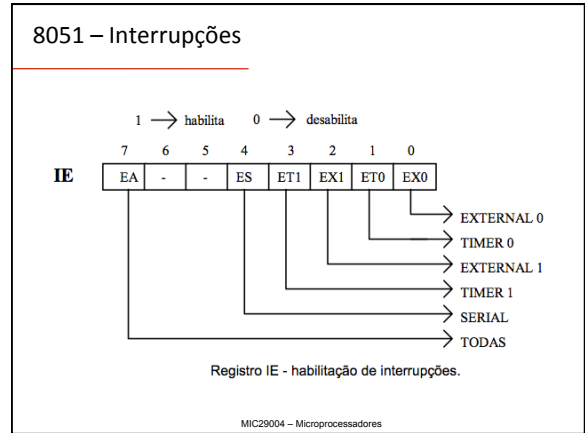
- Interrupções no 8051
  - 2 externas
  - 2 timers
  - 1 serial
- Duas prioridades (alta ou baixa)

MIC29004 – Microprocessadores

### 8051 – Interrupções

Interrupção	Endereço
Interrupção externa 0	0003H
Timer 0	000BH
Interrupção externa 1	0013H
Timer 1	001BH
Canal serial	0023H

MIC29004 – Microprocessadores



### 8051 – Interrupções

Faça um programa que aceite int0 (acionada por borda). Qdo a int0 for solicitada, escrever na porta P1, caractere por caractere, a cadeia de 16 caracteres: 'Microcontrolador'.

MIC29004 – Microprocessadores

```

reset      equ 00h
ltint0    equ 03h ; local tratador
state     equ 20h

org reset      ;PC=0 depois de reset
jmp inicio

org ltint0
jmp handler

inicio:
mov ie,#1000001b ; habilita int
mov tcon,#0000001b ; borda

mov state,#0h ;inicialização
mov r0,# state
mov dptr,#tabela
mov r1,#0

volta:    cjne @r0,#1,volta
mov      a,r1
mov      state,#0h
mov      a,r1
movc    a,@a+dptr
mov      p1,a
inc     r1
cjne    r1,#16,volta
jmp     $

handler:  mov      state,#1h
reti

tabela:  db 'Microcontrolador'
end

```

MIC29004 – Microprocessadores