

# Microcontrolador Intel 8051 - Família MCS-51

Cristiano Cesar  
Danilo Alves de Jesus  
Denis Fournier  
Guilherme Augusto Macedo  
Matheus L. D. da Silva  
Victor Hugo Carlquist da Silva

Instituto Federal de São Paulo

21 de junho de 2012

## 1. Introdução

### 1.1 História

### 1.2 Arquitetura Von Neumann x Harvard

### 1.3 Microcontrolador x Microprocessador

### 1.4 CISC x RISC

## 2. Descrição Geral

## 3. Arquitetura do 8051

## 4. O Assembly do 8051

### 4.1 Tipos de Endereçamentos

### 4.2 Instruções *Assembly*

### 4.3 Exemplos em *Assembly*



# História

1. Herdeira da família 8048;



# História

1. Herdeira da família 8048;
2. 8051 faz parte de uma família chamada MCS-51;

# História

1. Herdeira da família 8048;
2. 8051 faz parte de uma família chamada MCS-51;
3. É a versão mais básica;



# História

1. Herdeira da família 8048;
2. 8051 faz parte de uma família chamada MCS-51;
3. É a versão mais básica;
4. Existem outras versões com características específicas

# História

1. Herdeira da família 8048;
2. 8051 faz parte de uma família chamada MCS-51;
3. É a versão mais básica;
4. Existem outras versões com características específicas
  - 4.1 8751 - Vem com EPROM;



# História

1. Herdeira da família 8048;
2. 8051 faz parte de uma família chamada MCS-51;
3. É a versão mais básica;
4. Existem outras versões com características específicas
  - 4.1 8751 - Vem com EPROM;
  - 4.2 8052 - Tem um timer a mais do que o 8051;



# História

1. Herdeira da família 8048;
2. 8051 faz parte de uma família chamada MCS-51;
3. É a versão mais básica;
4. Existem outras versões com características específicas
  - 4.1 8751 - Vem com EPROM;
  - 4.2 8052 - Tem um timer a mais do que o 8051;
  - 4.3 8052 - Basic - Vem com um interpretador BASIC.



# Arquitetura Von Neumann x Harvard

**Von Neumann** - Dados e programas na mesma memória;

**Harvard** - Dados e programas em memória separadas.

# Microcontrolador x Microprocessador

**Microcontrolador** - Vários circuitos juntos;

**Microprocessador** - Circuitos separados.



# CISC x RISC

**CISC** - *Complex Instruction Set Computer*;

**RISC** - *Reduced Instruction Set Computer*.

## 1. Introdução

### 1.1 História

### 1.2 Arquitetura Von Neumann x Harvard

### 1.3 Microcontrolador x Microprocessador

### 1.4 CISC x RISC

## 2. Descrição Geral

## 3. Arquitetura do 8051

## 4. O Assembly do 8051

### 4.1 Tipos de Endereçamentos

### 4.2 Instruções *Assembly*

### 4.3 Exemplos em *Assembly*

## Descrição Geral

- 256 *Bytes* RAM
  - 128 *bytes* de uso geral;
  - 128 *bytes* de registradores especiais;
- ROM interna de 4 *Kbytes*;
- 4 Pots de I O;
- 2 Timers de 16*bits*;
- 1 Interface serial;
- Capacidade de 64 *Kbytes* de endereçamento externo de ROM;
- Capacidade de 64 *Kbytes* de endereçamento externo de RAM;
- Ciclo típicos de instrução de 1 e 2  $\mu s$  a 12MHz;
- Instrução direta de divisão e multiplicação;
- Entradas de interrupção externa.

## 1. Introdução

### 1.1 História

### 1.2 Arquitetura Von Neumann x Harvard

### 1.3 Microcontrolador x Microprocessador

### 1.4 CISC x RISC

## 2. Descrição Geral

## 3. Arquitetura do 8051

## 4. O Assembly do 8051

### 4.1 Tipos de Endereçamentos

### 4.2 Instruções *Assembly*

### 4.3 Exemplos em *Assembly*

## Os *Ports* de *I/O*

- Possui 4 *Ports* de *I/O*:



## Os *Ports* de *I/O*

- Possui 4 *Ports* de *I/O*:
  - Os *Port* P0 e P2;

## Os *Ports* de *I/O*

- Possui 4 *Ports* de *I/O*:
  - Os *Port* P0 e P2;
  - O *Port* P3;

## Os *Ports* de *I/O*

- Possui 4 *Ports* de *I/O*:
  - Os *Port* P0 e P2;
  - O *Port* P3;
  - O *Port* P1.

## Pinagem

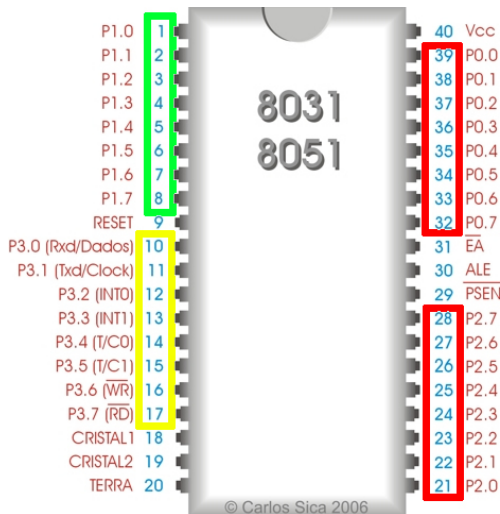


Figura 1: Ilustra a disposição dos pinos do 8051.

## Os Ports P0 e P2

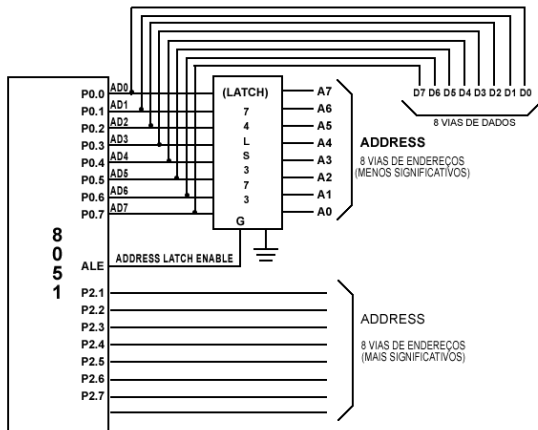


Figura 2: Demultiplexação de dados e endereços no Port P0.

# O *Port* P3

1. *Port* de propósito geral;

## O *Port* P3

1. *Port* de propósito geral;
2. Conforme a necessidade de se utilizar os periféricos do chip, alguns *bits* desse *port* vão sendo comprometidos.

## O Port P3 - Ilustração

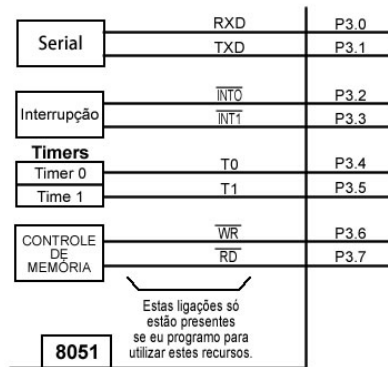


Figura 3: Detalhamento do P3.



# Os pinos de controle do Chip

1.  $\overline{PSEN}$  (Program Store Enable);

## Os pinos de controle do Chip

1.  $\overline{PSEN}$  (Program Store Enable);
2. ALE (Address Latch Enable);

## Os pinos de controle do Chip

1.  $\overline{PSEN}$  (Program Store Enable);
2. ALE (Address Latch Enable);
3.  $\overline{EA}$  (External Access);

## Os pinos de controle do Chip

1.  $\overline{PSEN}$  (Program Store Enable);
2. ALE (Address Latch Enable);
3.  $\overline{EA}$  (External Access);
4. RST (Reset).

## Os pinos de controle do Chip - Ilustração

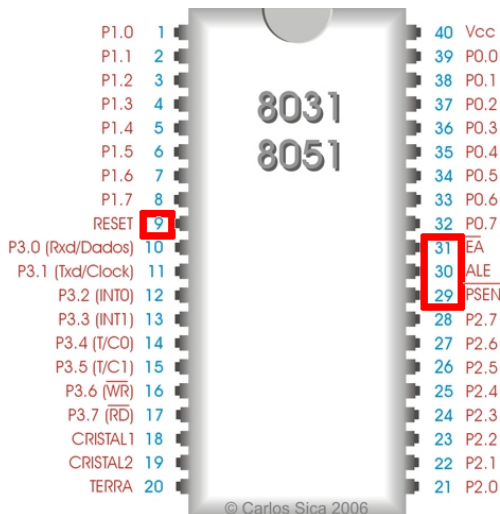


Figura 4: Os pinos de controle.

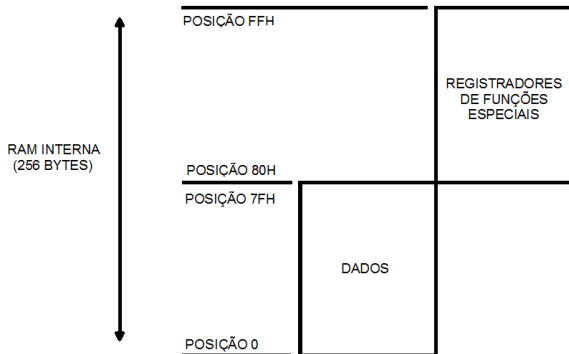


Figura 5: Funcionamento da Memória Interna do Chip.

## Mapa dos Registradores

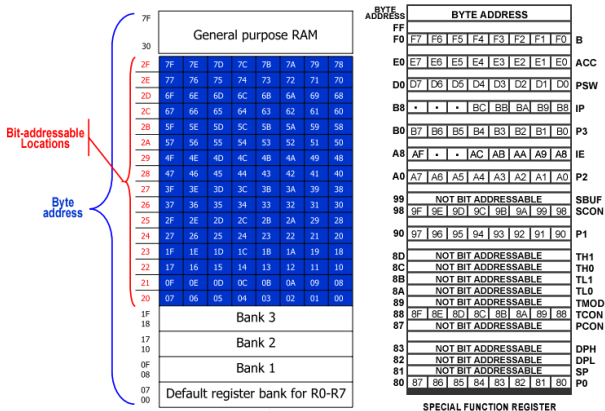


Figura 6: Mapa dos Registradores.

# Os Registradores de Funções Especiais

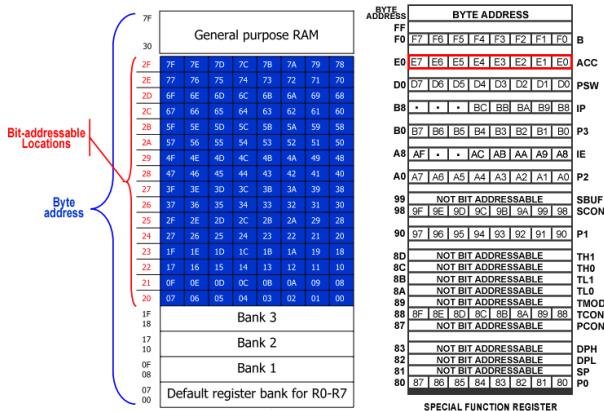


Figura 7: Registrador Acumulador.



## Ports de I/O

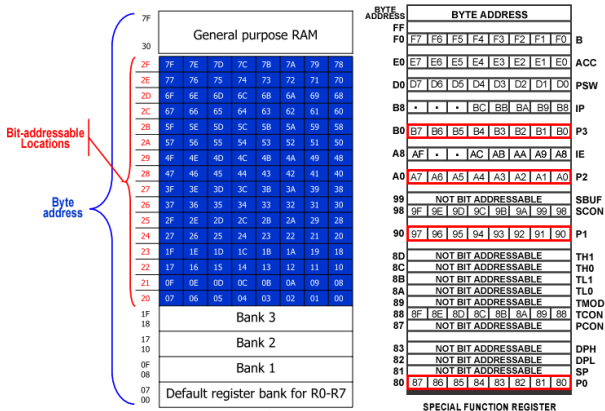


Figura 8: Registrador dos Ports I/O.

# Registrador B

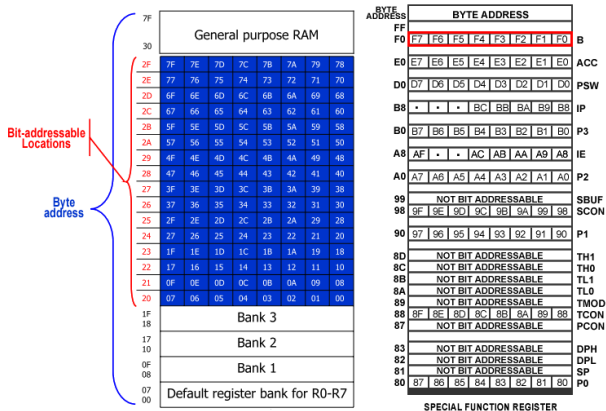


Figura 9: Registrador B.

# Registrador de Estado PSW

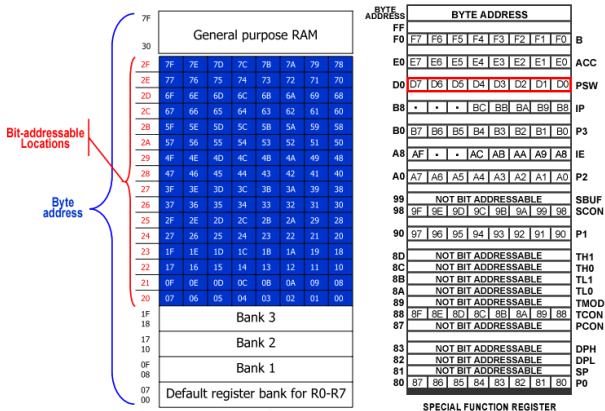


Figura 10: Registrador PSW.

## Registrador IP e IE

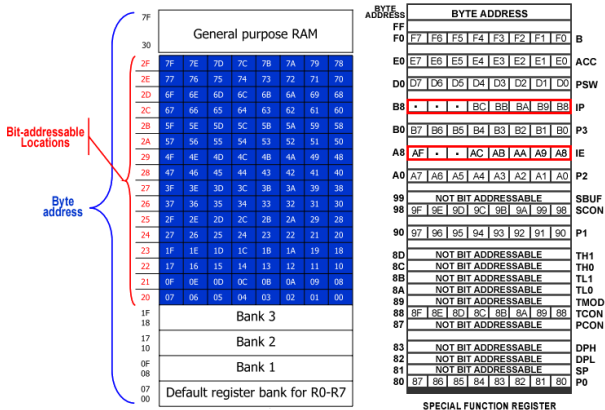


Figura 11: Registrador IP e IE.

# Registrador DPH e DPL

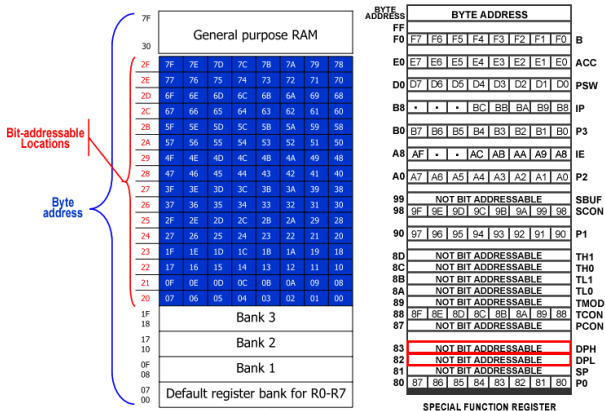


Figura 12: Registrador DPH e DPL.

# Registrador SP

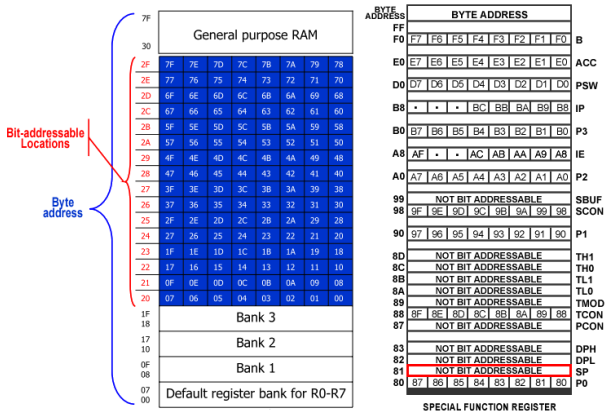


Figura 13: Registrador SP.

# Registrador PCON

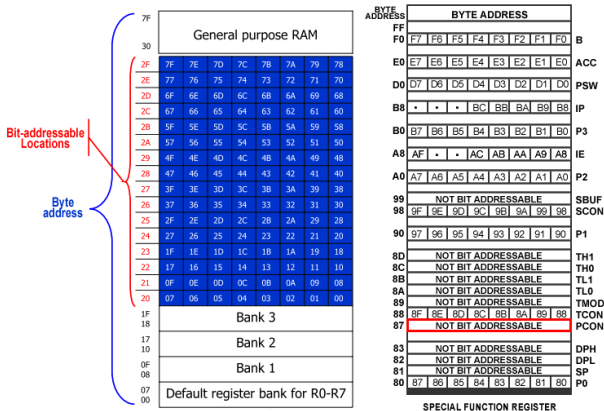


Figura 14: Registrador PCON.

# Registrador SBUF

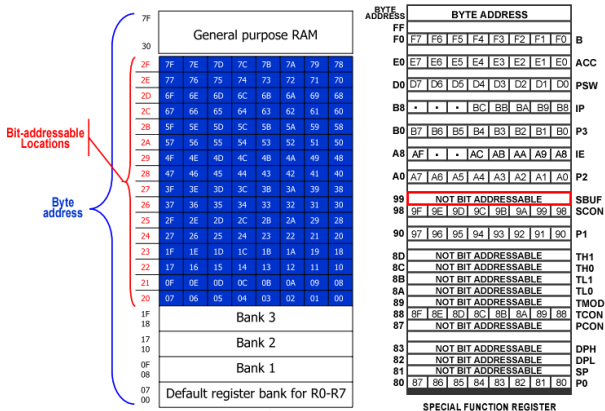


Figura 15: Registradores SBUF.



# Registrador SCON

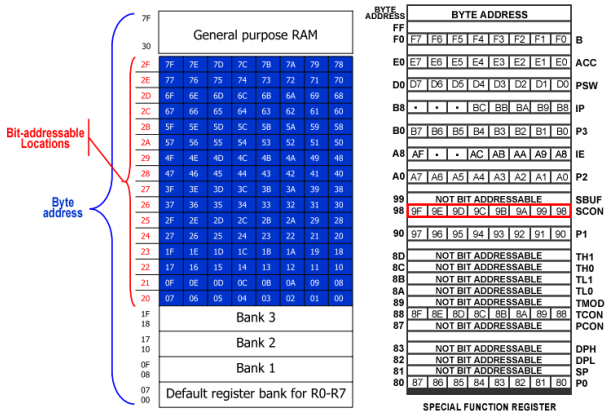


Figura 16: Registrador SCON.

# Registrador TH0, TL0, TH1 e TL1

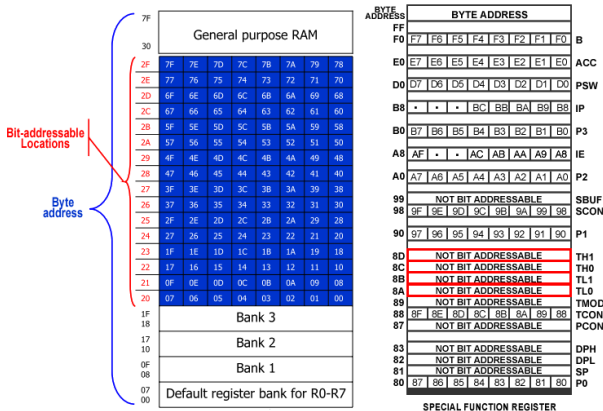


Figura 17: Registradores TH0, TL0, TH1 e TL1.

# Registrador TCON

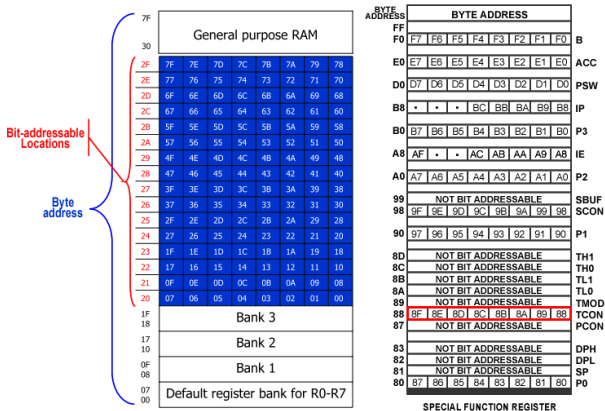


Figura 18: Registrador TCON.

# Registrador TMOD

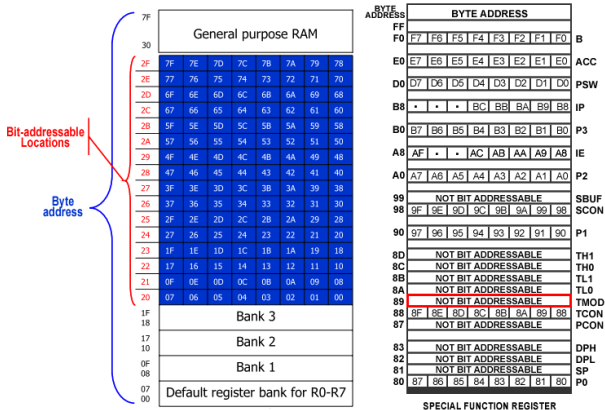


Figura 19: Registrador TMOD.



## Cada Bit do PSW

### PSW.7 - (CY - Carry)

#### PSW

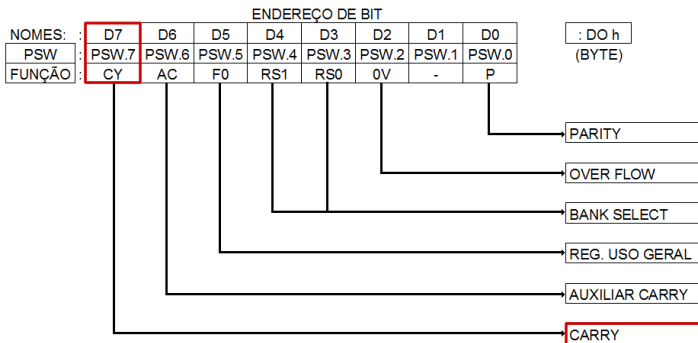


Figura 21: Bit de carry.

## Cada Bit do PSW

### PSW.6 - (AC - Auxiliar Carry)

#### PSW

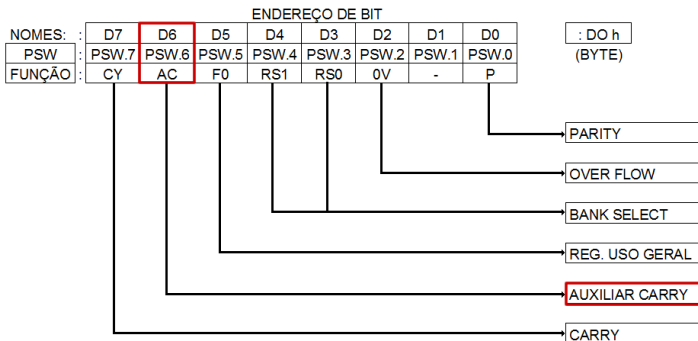


Figura 22: Bit de auxiliar do carry.

## Cada Bit do PSW

### PSW.5 - (F0 - Reg. Uso Geral)

#### PSW

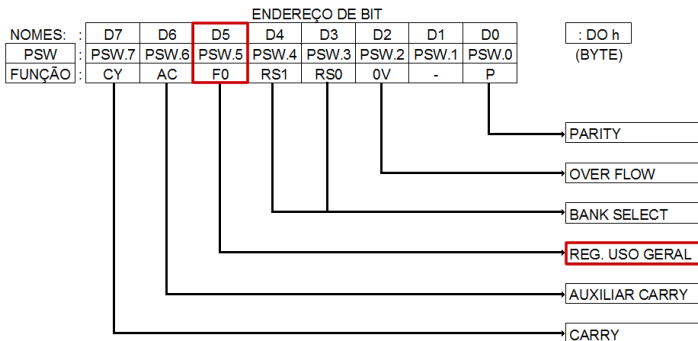


Figura 23: Bit de reg. de uso geral.



## Cada Bit do PSW

## PSW.4 - (RS1/RS0 - Bank Select)

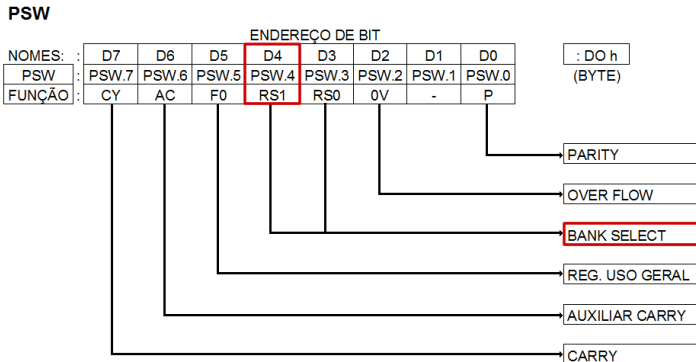


Figura 24: *Bit* de chaveador entre os bancos.

## Cada Bit do PSW

### PSW.3 - (RS1/RS0 - Bank Select)

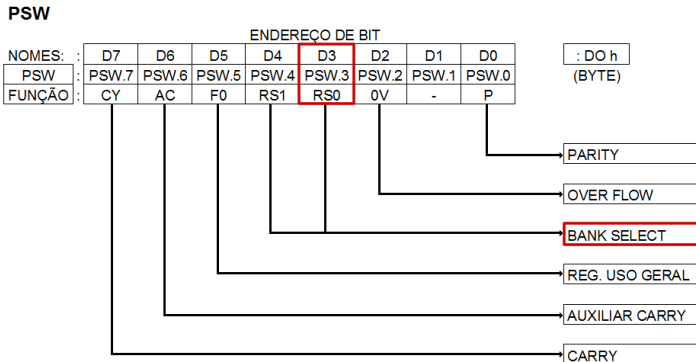


Figura 25: *Bit* de chaveador entre os bancos.

## Cada Bit do PSW

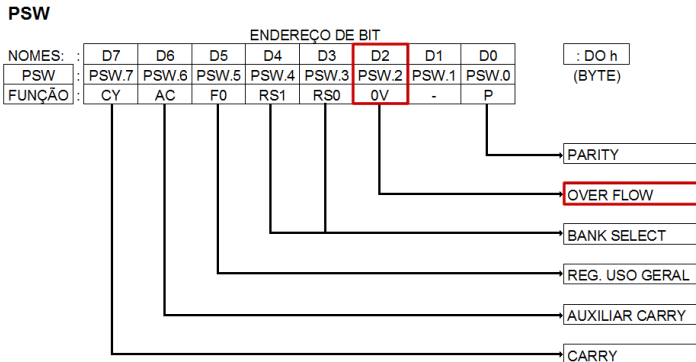
PSW.2 - (OV - *O*verflow)

Figura 26: *Bit de overflow.*

## Cada Bit do PSW

### PSW.1

#### PSW

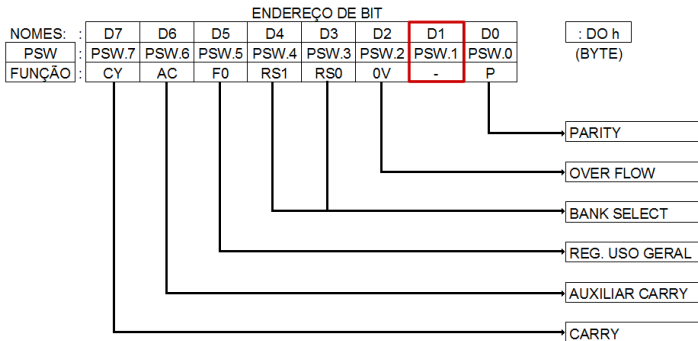


Figura 27: *Bit* definido pelo usuário.

## Cada Bit do PSW

PSW.0 - (P - *Parity*)

### PSW

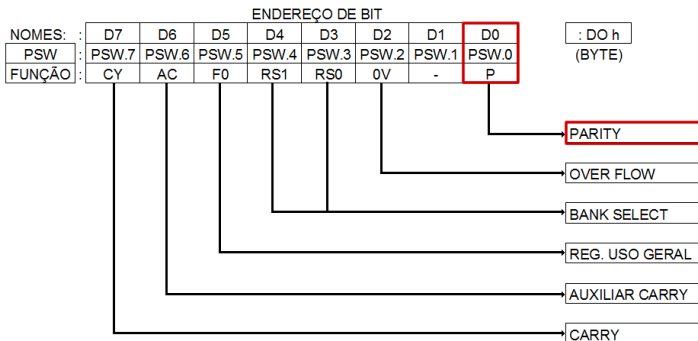


Figura 28: *Bit de parity.*

## 1. Introdução

### 1.1 História

### 1.2 Arquitetura Von Neumann x Harvard

### 1.3 Microcontrolador x Microprocessador

### 1.4 CISC x RISC

## 2. Descrição Geral

## 3. Arquitetura do 8051

## 4. O Assembly do 8051

### 4.1 Tipos de Endereçamentos

### 4.2 Instruções *Assembly*

### 4.3 Exemplos em *Assembly*

# Tipos de Endereçamentos

- Endereçamento por Registrador  
MOV A, R0

## Tipos de Endereçamentos

- Endereçamento por Registrador  
MOV A, R0
- Endereçamento Direto  
MOV A, 40H





## Tipos de Endereçamentos

- Endereçamento por Registrador  
MOV A, R0
- Endereçamento Direto  
MOV A, 40H
- Endereçamento Indireto  
MOV A, @R0



## Tipos de Endereçamentos

- Endereçamento por Registrador  
MOV A, R0
- Endereçamento Direto  
MOV A, 40H
- Endereçamento Indireto  
MOV A, @R0
- Endereçamento Imediato  
MOV A, #12



## Tipos de Endereçamentos

- Endereçamento por Registrador  
MOV A, R0
- Endereçamento Direto  
MOV A, 40H
- Endereçamento Indireto  
MOV A, @R0
- Endereçamento Imediato  
MOV A, #12
- Endereçamento Indexado  
JMP @A+DPTR



## Tipos de Endereçamentos

- Endereçamento por Registrador  
MOV A, R0
- Endereçamento Direto  
MOV A, 40H
- Endereçamento Indireto  
MOV A, @R0
- Endereçamento Imediato  
MOV A, #12
- Endereçamento Indexado  
JMP @A+DPTR
- Endereçamento de Bits  
MOV E0.0H, C

# Instruções *Assembly*

- Aritméticas;
- Lógicas;
- Transferência de Dados;
- Desvio.

## Exemplos de Assembly - 1

Trocar o conteúdo da memória interna com a externa

```
MOV    DPTR,    #0FF00h
MOVX   A,        @DPTR
MOV     R0,      0FFh
MOV     0FFh,    A
MOV     A,        R0
MOVX   @DPTR,    A
```

## Exemplos de Assembly - 2

Armezenar o nibble mais significativo de R7 nos dois nibbles de R6

```
MOV    A,    R7
ANL    A,    #0F0h
MOV    R6,    A
SWAP   A
ORL    A,    R6
MOV    R6,    A
```



## Exemplos de Assembly - 3

Dividir o conteúdo do R0 pelo de R1. Armazenar o quociente em R2 e o resto em R3. Depois restaurar o conteúdo original de R0.

```
MOV  A,    R0
MOV  B,    R1
DIV   AB
MOV  R2,   A
MOV  R3,   B
MOV  B,    R1
MUL  AB
ADD  A,    R3
MOV  R0,   A
```



## Exemplos de Assembly - 4

Transferir um bloco de dados de 20h até 30h para uma memória externa na posição de 1020h até 1030h.

```
      MOV    R7,      #0Ah
      MOV    R0,      #20h
      MOV    DPTR,    #1020h
nxt:  MOV    A,        @R0
      MOVB  @DPTR,    A
      INC   R0
      INC   DPTR
      DJNZ  R7,      nxt
```



# Conclusão