



SEL 0415 – INTROD. À ORGANIZAÇÃO DE COMPUTADORES

Aula 9 – Microcontrolador Intel 8051

Prof. Dr. Marcelo A. C. Vieira



MICROCONTROLADOR 8051

Microcontrolador 80C51

- Membro da família MCS-51
- Núcleo de todos os dispositivos MCS-51 (Atmel)
- Sistema de um *chip* único, que contém, **além do microprocessador de 8 bits**:
 - Memória de Programa e Memória de Dados
 - Portas de I/O
 - Comunicação Serial (UART)
 - Contadores/ “Timers”
 - Lógica para Controle de Interrupção

Microcontrolador 80C51

1. Características do Núcleo (Core)

- CPU de 8 bits otimizada para aplicações de controle;
- Capacidade de processamento booleano (lógica de um único bit);
- Endereçamento de até 64 Kbytes de memória de programa externa;
- Endereçamento de até 64 Kbytes de memória de dados externa;
- 4 Kbytes de memória de programa (FLASH ROM) interna;
- 256 bytes de memória de dados (SRAM) interna, sendo 128 bytes para registradores especiais e 128 bytes disponíveis para uso geral;

Microcontrolador 80C51

1. Características do Núcleo (Core)

- 4 portas paralelas de 8 bits (32 linhas de I/O bidirecionais endereçadas individualmente)
- UART full duplex (*Universal Asynchronous Receiver Transmitter*)
- 2 Contadores / Temporizadores de 16 bits cada
- Estrutura de interrupção com níveis de prioridade
- Oscilador interno
- Versões disponíveis de 12 a 30 MHz (instruções de um ciclo, de 1 μ s a 400 ns)

Microcontrolador 80C51

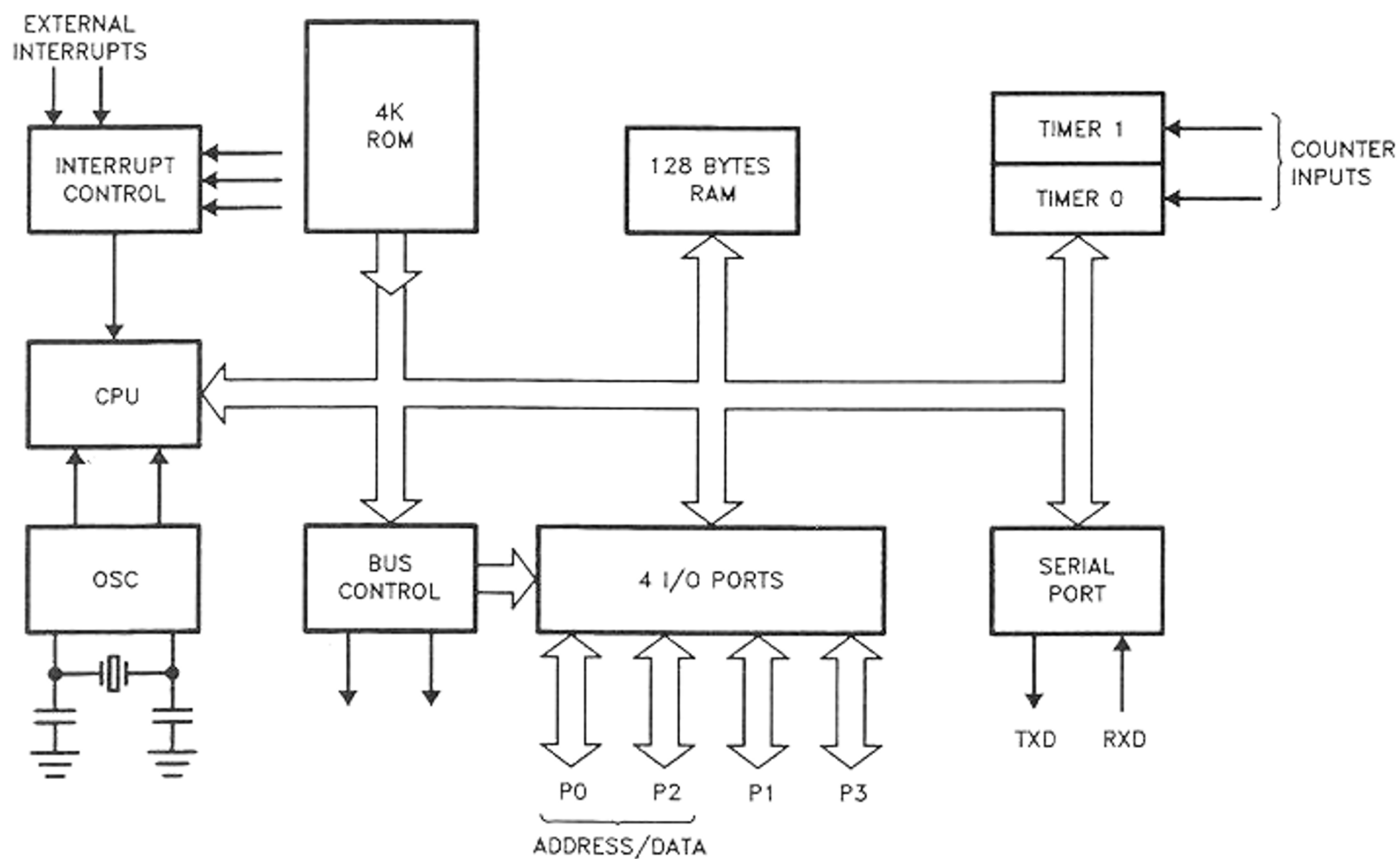
2. Arquitetura

- Arquitetura Von Neumann
- Set de instruções do tipo CISC
- 111 instruções
- O conjunto de instruções inclui:
 - ✓ Multiplicação e Divisão
 - ✓ *Bit set, reset, e test* (Instruções Booleanas).

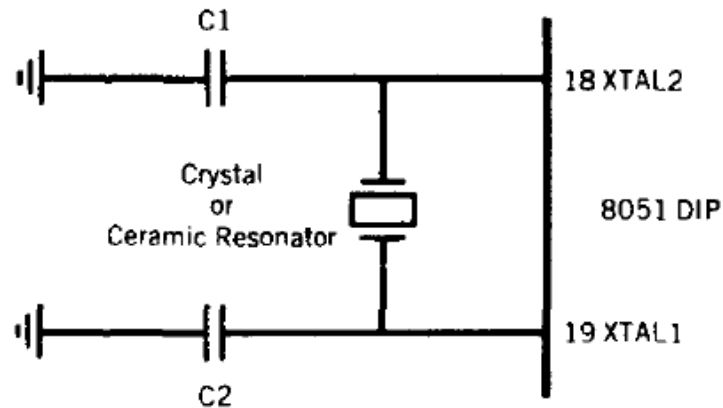
Instruções - CISC

- 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%

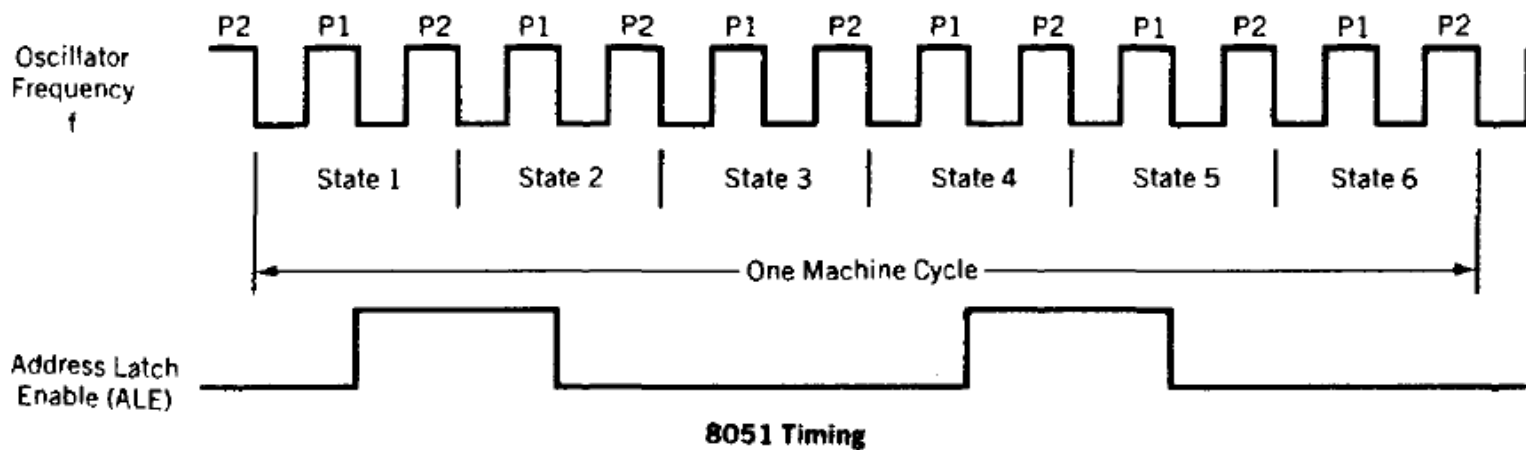
Microcontrolador 80C51



$$\text{Ciclo de Busca} + \text{Ciclo de Execução} = f_{\text{cristal}}/12$$

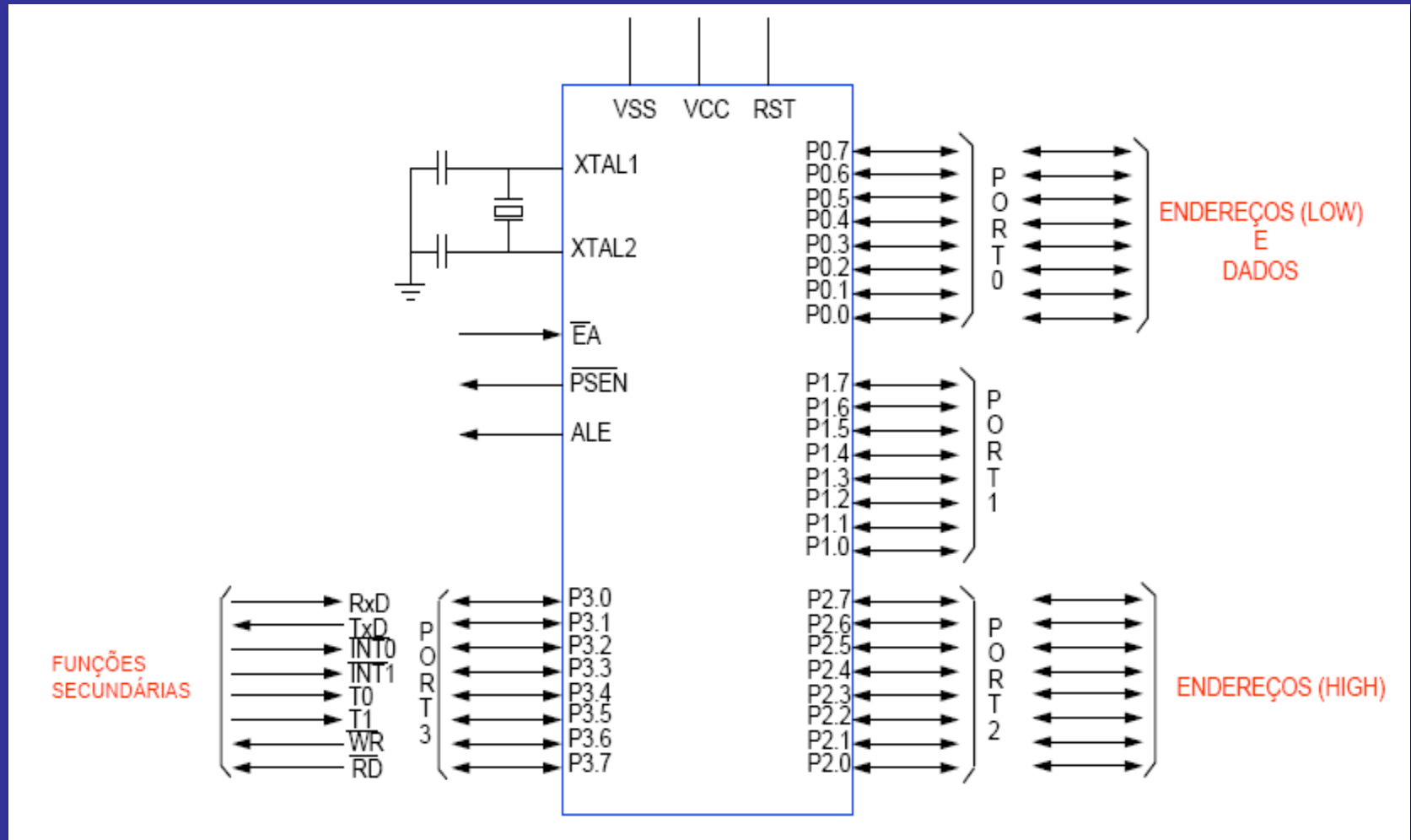


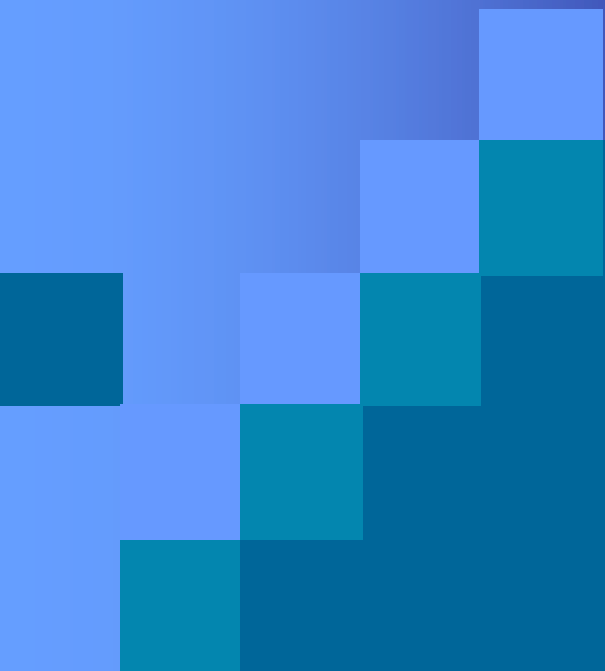
Crystal or Ceramic Resonator Oscillator Circuit



Microcontrolador 80C51

Configuração dos pinos

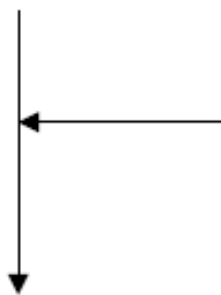




Introdução à Programação em Assembly

Fluxograma

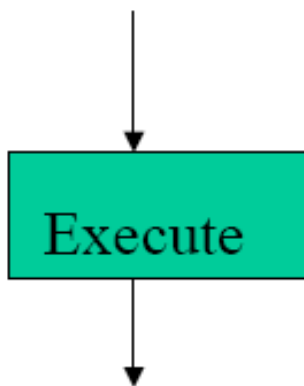
- Linhas de Fluxo do Programa



- Mostram a seqüência de execução das Instruções.

- Cada Bloco do Fluxograma possui apenas uma linha de Fluxo de Entrada e uma ou duas de saída

- Bloco de Processo

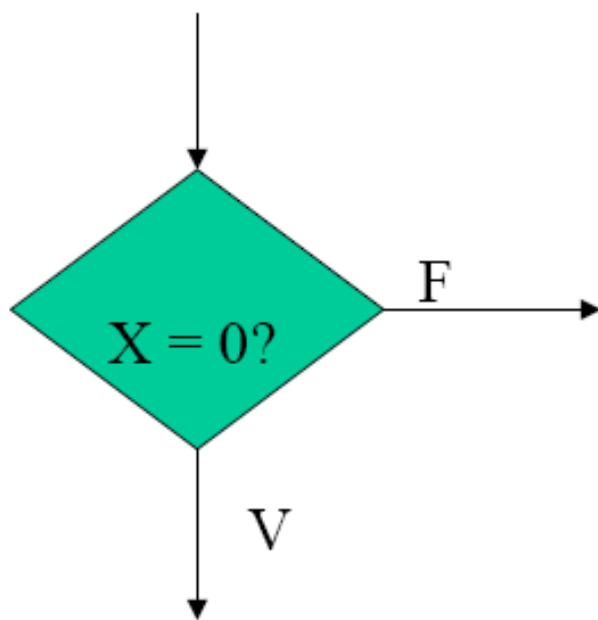


- Equivalem às Instruções que realizam alguma operação do tipo:

- Movimento de Dados
- Operação Aritmética
- Operação Lógica

Fluxograma

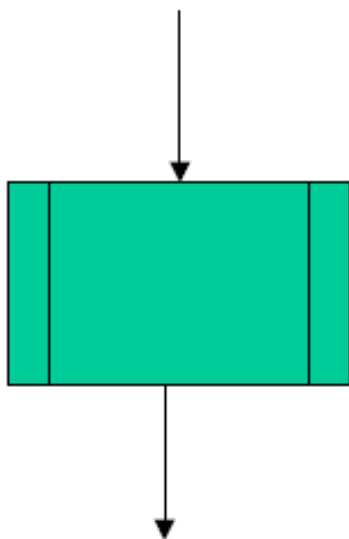
- Bloco de Decisão



- Equivale às Instruções que decidem sobre o Fluxo do Programa.
- Se a função dentro do bloco for Verdadeira(V) o programa continua abaixo, se for Falsa(F) o programa muda o fluxo.

Fluxograma

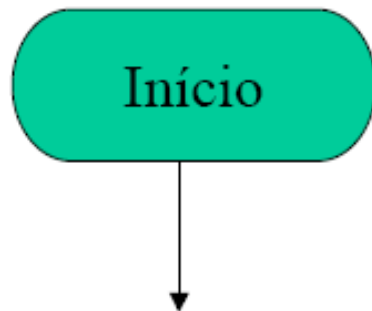
- Processo Pré-definido



- Equivale às Instruções que mandam executar uma Sub-rotina armazenada em outro lugar da Memória.
- Observe que quando a sub-rotina termina, o fluxo do programa continua normalmente.

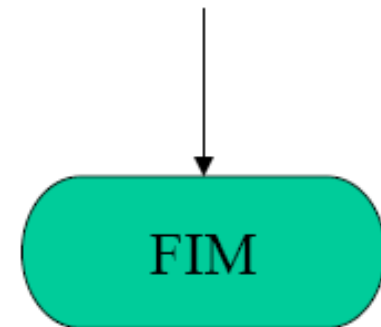
Fluxograma

- Bloco de Início de Programa



- O Bloco de Início de Programa não equivale a uma Instrução específica do Instruction SET.

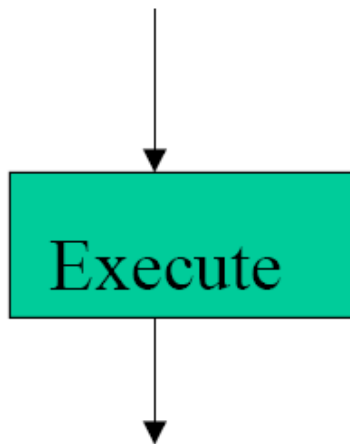
- Bloco de Fim de Programa



- O Bloco de FIM equivale a uma instrução que termina o Programa. É chamado de **FIM LÓGICO** do Programa.

Codificação Assembly do 8051

- Instruções equivalentes ao Bloco de Processo



- Instruções Aritméticas

SUBB **A, direct**

ADD **A, Rn**

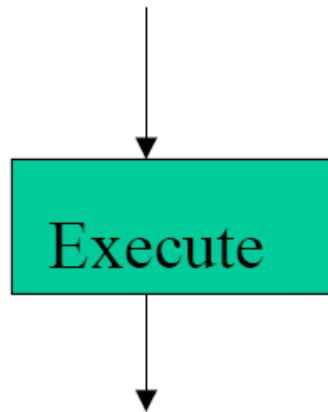
INC **A**

DEC **A**

DA **A**

Codificação Assembly do 8051

- Instruções equivalentes ao Bloco de Processo



- Instruções de Transferência de Dados

MOV A, Rn

MOVC A, @A+DPTR

MOVX A,@DPTR

PUSH direct

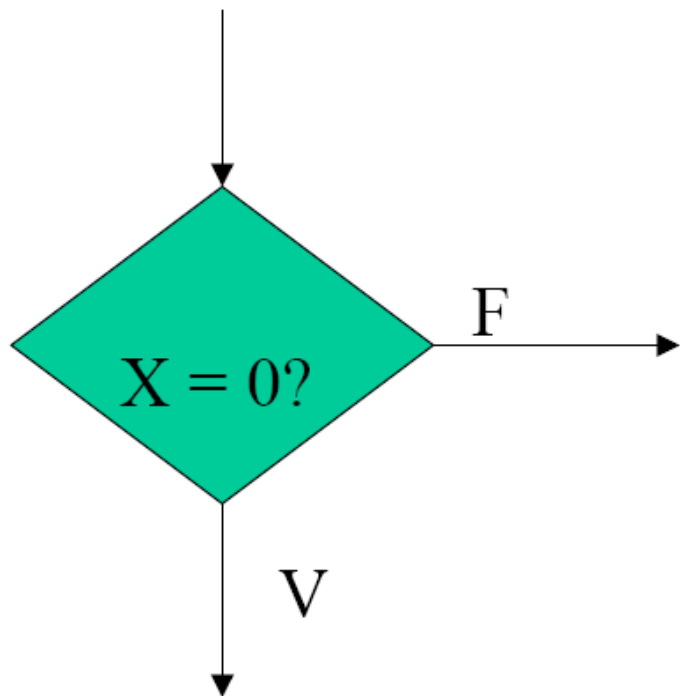
POP direct

XCH A, Rn

Codificação Assembly do 8051

- Instruções equivalentes ao Bloco de Decisão

- Instruções de Desvio



JZ rel

JNZ rel

CJNE A, direct, rel

JC rel

JNC rel

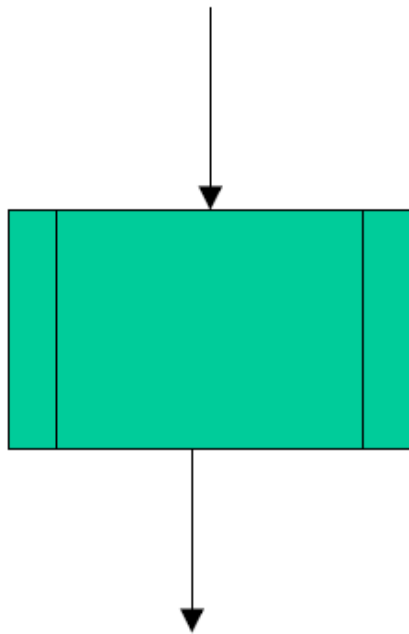
JB bit, rel

JNB bit, rel

DJNZ Rn, rel

Codificação Assembly do 8051

- Instruções equivalentes ao Bloco de Processo Pré-definido



- Instruções de Sub-Rotina

LCALL addr16

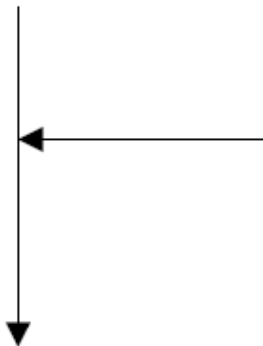
ACALL addr11

RET

RETI

Codificação Assembly do 8051

- Instruções equivalentes a Mudança de Fluxo



- Instruções de Saltos

LJMP **addr16**

AJMP **addr11**

SJMP **rel**

JMP **@A+DPTR**

Modos de Endereçamento do 8051

1. Endereçamento Imediato

- Opera sobre o dado localizado na própria instrução

• Identificado através do sinal #

• Exemplo:

ADD A,#30

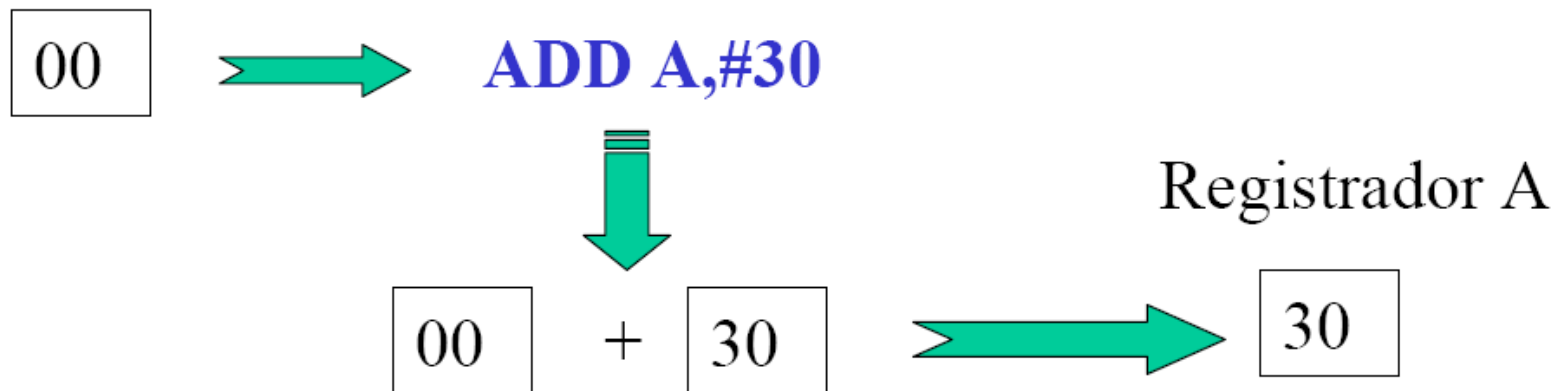
O dado 30 é somado ao Registrador A

Modos de Endereçamento do 8051

1. Endereçamento Imediato

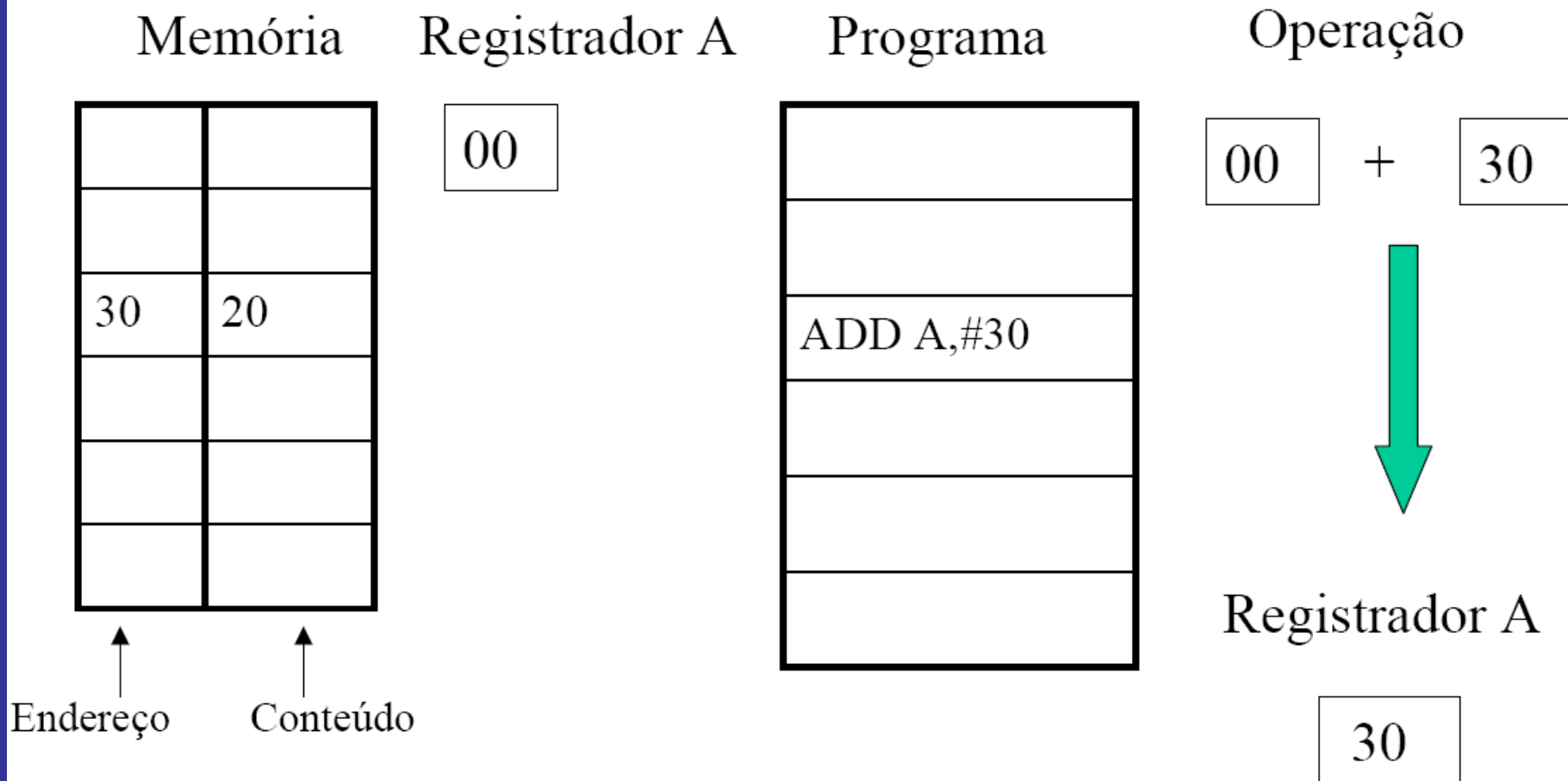
ADD A,#30

Registrador A



Modos de Endereçamento do 8051

ADD A,#30



Modos de Endereçamento do 8051

2. Endereçamento Direto

- Opera sobre o dado cujo endereço está na instrução

• Exemplo:

ADD A,30

O dado armazenado no endereço 30 é somado ao Registrador A

Modos de Endereçamento do 8051

2. Endereçamento Direto

ADD A,30

Registrador A

00



ADD A,30



Conteúdo do
Endereço 30

00

+

20



Registrador A

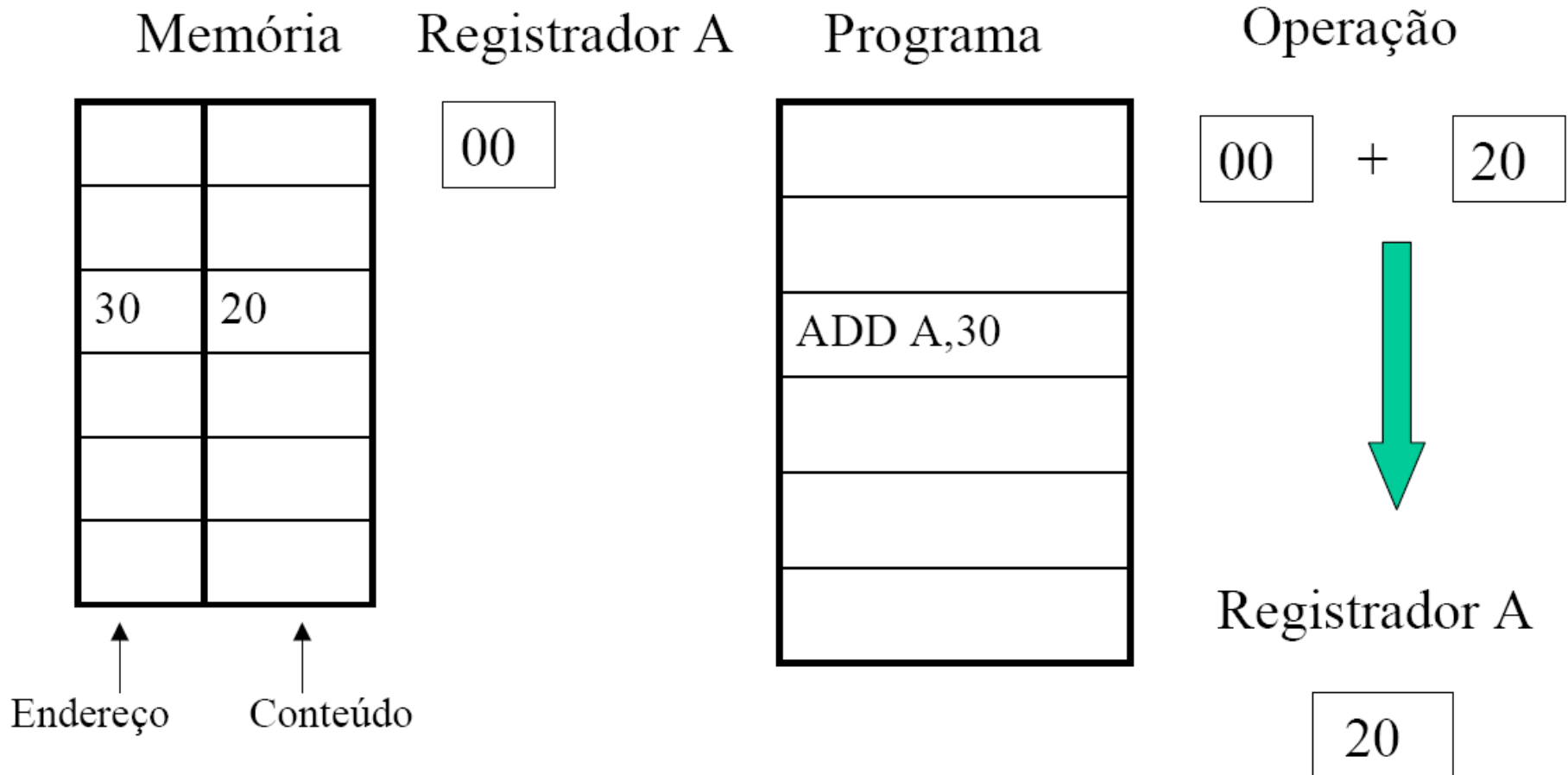
20

20



Modos de Endereçamento do 8051

ADD A,30



Modos de Endereçamento do 8051

2. Endereçamento Indireto

- Opera sobre o dado cujo endereço está armazenado em um Registrador apontado na instrução

• Identificado através do sinal @

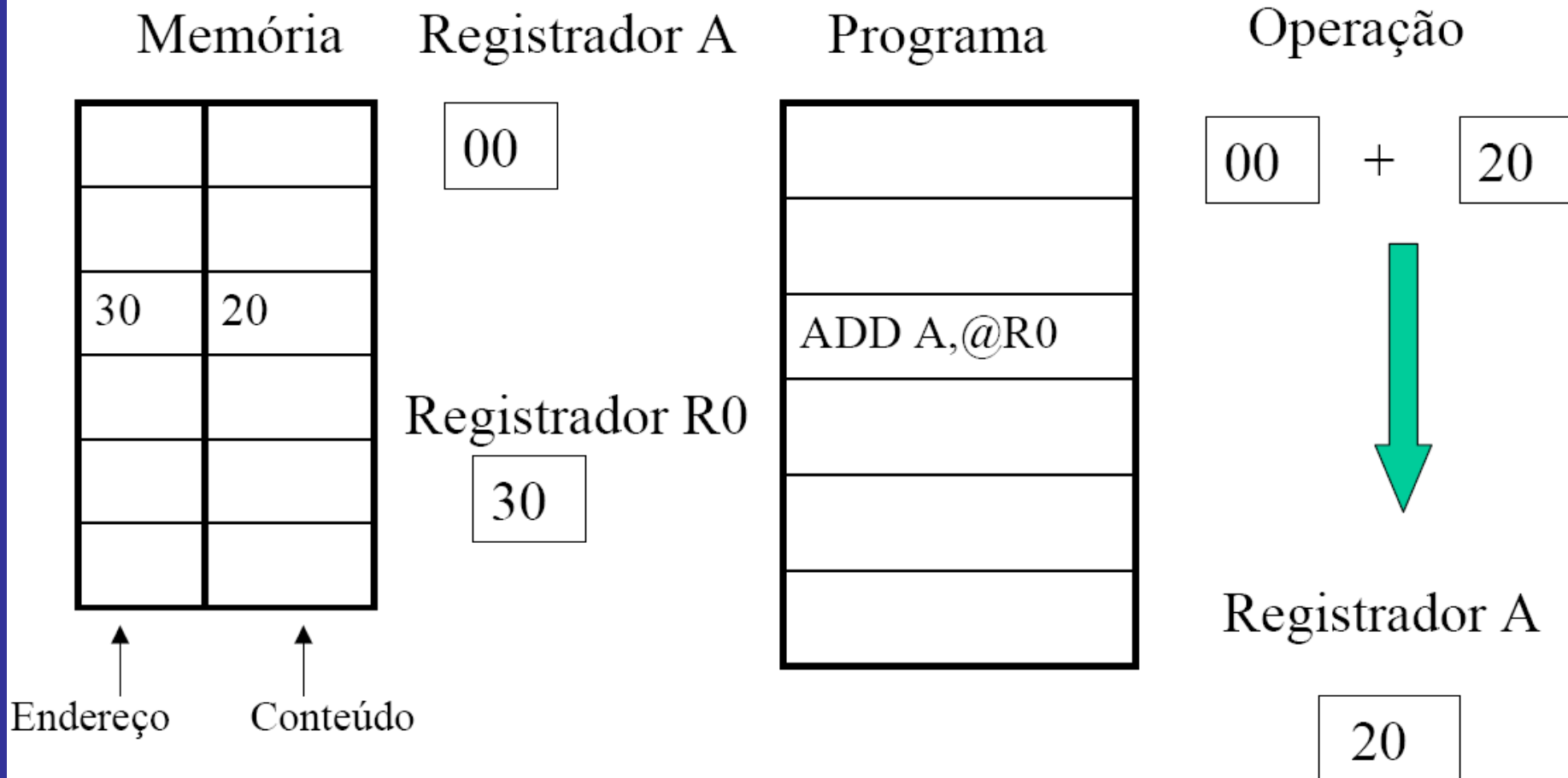
• Exemplo:

ADD A,@R0

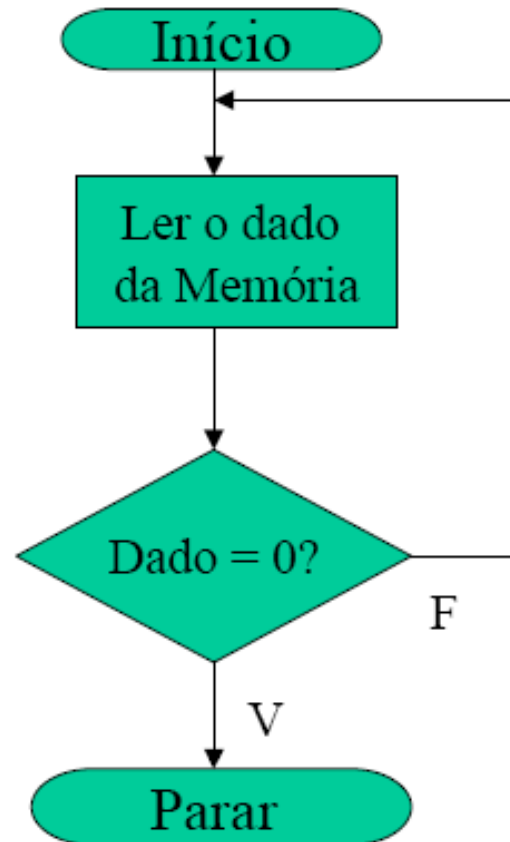
O dado armazenado no endereço apontado pelo Registrador R0 é somado ao Registrador A

Modos de Endereçamento do 8051

ADD A,@R0

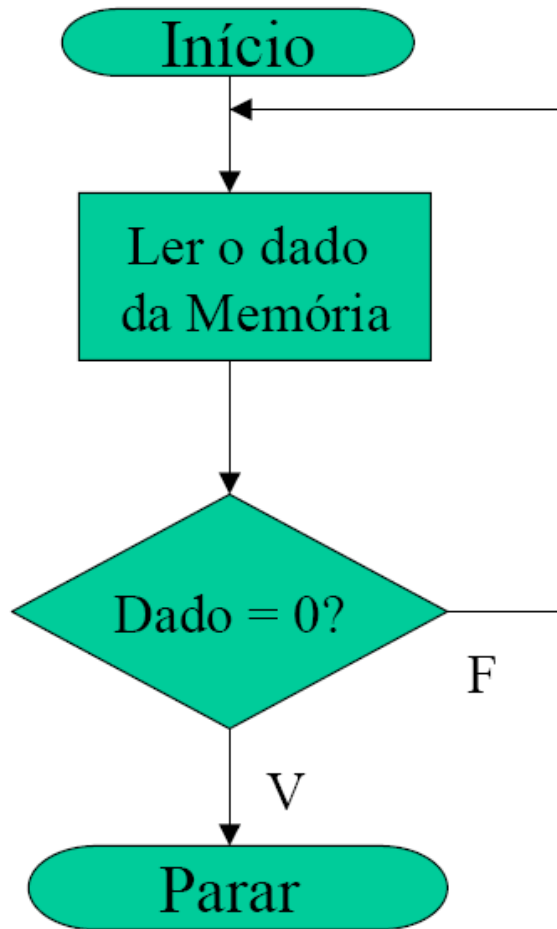


Exemplo de Fluxograma de um Programa de Microprocessador



- O programa ao lado deve Ler um Dado da memória, verificar se é igual a zero. Se não for zero, continua em LOOP. Se for zero para o programa.

Exemplo de um Programa Assembly do 8051



```
ORG 0  
  
LOOP:  
  
MOV A,30H  
  
CJNE A,#00,LOOP  
  
AQUI:  
SJMP AQUI
```

Exemplo de um Programa Assembly do 8051

Mnemônicos (Programa Assembly)

```
                ORG 0
LOOP:           MOV A,30H
                CJNE A,#00,LOOP
AQUI:           SJMP AQUI
```

COMPILADOR

Código Compilado (Opcode)

Addr	Opcodes	ASC	Label	Disassembly
0000	E5 30	â0	LOOP	MOV A,30h
0002	B4 00 FB	ÿ0		CJNE A,#00h,LOOP
0005	80 FE	€p	AQUI	SJMP AQUI

Exemplo de um Programa Assembly do 8051

Memória

00	E5
01	30
02	B4
03	00
04	FB
05	80
06	FE

↑
Endereço

↑
Conteúdo

Addr	Opcodes	ASC	Label	Disassembly
0000	E5 30	ã0	LOOP	MOV A,30h
0002	B4 00 FB	îû		CJNE A,#00h,LOOP
0005	80 FE	€þ	AQUI	SJMP AQUI

Exemplo de Programa:

Considere o programa abaixo: qual o valor final armazenado em cada um dos registradores (em hexadecimal) após o término de sua execução?

```
ORG 0000h
MOV A, #00100001b
MOV 30h, #9Bh
MOV 31h, 30h
MOV 32h, #30h
MOV R0, #32h
MOV R1, 32h
ADD A, @R0
ANL A, @R1
INC A
DEC R0
ADD A, @R0
MOV @R1, A
SJMP $
END
```

Registrador	Valor Final
A	
30h	
31h	
32h	
R0	
R1	

Exemplo de Programa:

Considere o programa abaixo: qual o valor final armazenado em cada um dos registradores (em hexadecimal) após o término de sua execução?

```
ORG 0000h
MOV A, #00100001b
MOV 30h, #9Bh
MOV 31h, 30h
MOV 32h, #30h
MOV R0, #32h
MOV R1, 32h
ADD A, @R0
ANL A, @R1
INC A
DEC R0
ADD A, @R0
MOV @R1, A
SJMP $
END
```

Registrador	Valor Final
A	ADh
30h	ADh
31h	9Bh
32h	30h
R0	31h
R1	30h

Algumas Instruções do 8051

Ver lista de instruções do 8051
na página da disciplina!

- MOV
- ADD
- SETB
- CLR
- SJMP
- ACALL
- INC
- DEC
- CPL
- NOP

- JB
- JNB
- JZ
- JNZ
- RET
- RETI
- DJNZ
- PUSH
- POP

- ORG
- END

} diretivas do
compilador

FIM