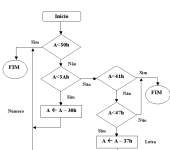


SEL-433 APLICAÇÕES DE MICROPROCESSADORES I

8051

AULA 2

Programação de Microprocessadores



Prof: Evandro L. L. Rodrigues

Como conhecer/estudar um
Microprocessador/Microcontrolador

As 5 partes fundamentais

- **Arquitetura interna**
- **Linguagem de Programação**
- **Pinagem**
- **Características Elétricas**
- **Ambiente de Desenvolvimento Integrado - IDE**

Programação de Microprocessadores



- **Microprocessadores** são **'Máquinas de Estado Seqüenciais Síncronas'** que operam mediante a execução de uma seqüência de códigos binários armazenados em memória.

• As ordens ou comandos compreendidos por um determinado Microprocessador, são **INSTRUÇÕES** seqüencialmente armazenadas na Memória.

• Ao conjunto de Instruções compreendidos por um determinado Microprocessador dá-se o nome de **"INSTRUCTION SET"**.

• Cada Microprocessador tem seu próprio Instruction Set que é em geral, diferente do Instruction Set de outro Microprocessador de fabricantes diferentes.

- Uma seqüência de Instruções do Instruction Set, armazenadas na memória e que realiza alguma operação, recebe o nome de **PROGRAMA**.

- Cada Instrução do Microprocessador é um código binário formada em geral por um ou mais Bytes.

- A cada código binário equivalente a uma Instrução está associado um **Mnemônico** para facilitar a compreensão da função que a Instrução executa.

- Ao conjunto de Instruções e seus Mnemônicos equivalentes dá-se o nome de **LINGUAGEM ASSEMBLY**.

Fluxograma

- Para a documentação lógica de um Programa em Assembly utiliza-se um **Fluxograma ou Diagrama de Blocos**.

- Cada bloco do Fluxograma equivale a um sub-conjunto do Instruction Set do Microprocessador.

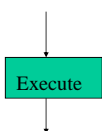
- O Fluxograma é uma forma de se implementar logicamente um programa, antes que o mesmo seja codificado na Linguagem Assembly do Microprocessador.

Fluxograma

- Linhas de Fluxo do Programa



- Bloco de Processo

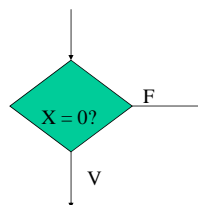


- Mostram a seqüência de execução das Instruções.
- Cada Bloco do Fluxograma possui apenas uma linha de Fluxo de Entrada e uma ou duas de saída

- Equivalem às Instruções que realizam alguma operação do tipo:
 - Movimento de Dados
 - Operação Aritmética
 - Operação Lógica

Fluxograma

- Bloco de Decisão

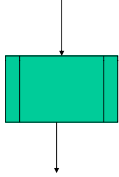


- Equivale às Instruções que decidem sobre o Fluxo do Programa.

- Se a função dentro do bloco for Verdadeira(V) o programa continua abaixo, se for Falsa(F) o programa muda o fluxo.

Fluxograma

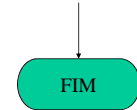
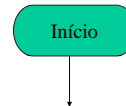
- Processo Pré-definido



- 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

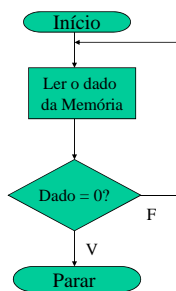
- Bloco de Início de Programa
- Bloco de Fim de Programa



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

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

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.

Codificação Assembly

- Para Codificar um Programa escrito através de um Fluxograma, deve-se escolher o Microprocessador, ou seja, conhecer seu Conjunto de Instruções.

- Os Microcontroladores da família MCS-51 serão os dispositivos a serem aplicados nesta disciplina.

Instruções do 8051

- O 8051 apresenta **111 tipos de instruções**, sendo **49 de 1 byte**, **45 de 2 bytes** e **17 de 3 bytes**.
- Considerando-se as variações de cada tipo, chega-se a **255 instruções**.

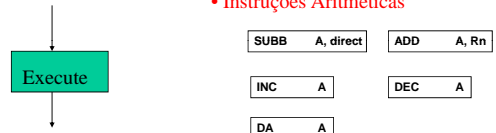
5 Grupos

- Transferência de dados,
- Operações lógicas,
- Operações aritméticas,
- Manipulação de variáveis booleanas (Bits)
- Controle de programa (Desvios)

Codificação Assembly do 8051

- Instruções equivalentes ao Bloco de Processo

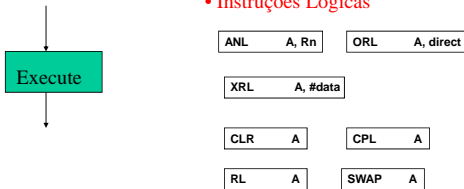
Instruções Aritméticas



Codificação Assembly do 8051

- Instruções equivalentes ao Bloco de Processo

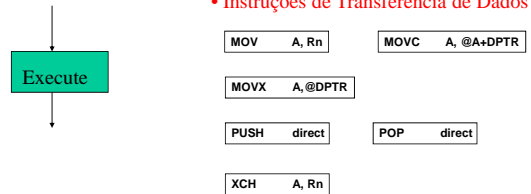
Instruções Lógicas



Codificação Assembly do 8051

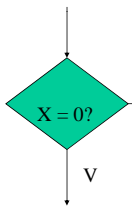
- Instruções equivalentes ao Bloco de Processo

Instruções de Transferência de Dados



Codificação Assembly do 8051

• Instruções equivalentes ao Bloco de Decisão

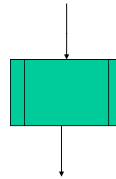


• Instruções de Desvio

- 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

2. Endereçamento Direto

- Opera sobre o dado cujo endereço está na instrução

• Exemplos:

```
ADD A,30
MOV A,P3
MOV 30h,A
```

Este tipo de instrução é codificada em 2 bytes.

Modos de Endereçamento do 8051

3. Endereçamento Indireto – instrução usa o símbolo @

- Opera sobre o dado cujo endereço está armazenado em um dos Registradores apontados na instrução.
- R0 e R1 para RAM interna e DPTR para RAM externa.

• Exemplos:

```
ADD A,@R0
```

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

```
MOVX A,@DPTR
```

O dado armazenado no endereço de RAM externa apontado por DPTR é movido para o Acumulador.

Modos de Endereçamento do 8051

3. Endereçamento Indireto

```
ADD A,@R0
```

Registrador A

00



```
ADD A,@R0
```

Registrador R0

30



00

+

20



Registrador A

20

Conteúdo do
Endereço 30

20



Modos de Endereçamento do 8051

4. Endereçamento de bits

- As instruções que manipulam bits individuais especificam este bit de forma direta.

• Exemplos: SETB 0Fh
CLR 20h

Este tipo de instrução é codificada em 2 bytes.

Modos de Endereçamento do 8051

5. Endereçamento indexado

- Este modo serve para endereçar a memória de programa (instrução MOVC).

• Exemplos: MOVC A,@A+DPTR
MOVC A,@A+PC

Este tipo de instrução é codificada em 1 byte.

Modos de Endereçamento do 8051

6. Endereçamento dos registradores

- Os opcodes das instruções de acesso aos registradores R0 a R7 utilizam três dos seus oito bits para especificar o registrador endereçado. São “mais curtas” que o modo direto.

• Exemplos: MOV A, R3
DEC R1

Este tipo de instrução é codificada em 1 byte.
Atenção para qual Banco de Registrador está ativo.

Modos de Endereçamento do 8051

7. Endereçamento implícito de registradores

- Instruções que trabalham sempre com um ou dois registradores específicos, e portanto nenhum byte é necessário para endereçar o operando.

• Exemplos: DAA - RRA - INCA - SWAPA
MUL AB - DIV AB

Modos de Endereçamento do 8051

8. Endereçamento relativo

- Instruções permitem realizar um salto em relação a posição atual do PC. O endereço relativo (Offset) é um dado de 8 bits sinalizado - realiza um salto relativo ao endereço que está o PC de -128 a +127.

• Exemplos: SJMP LOOP
SJMP \$

Este tipo de instrução é codificada em 2 bytes.

Modos de Endereçamento do 8051

9. Endereçamento Absoluto

- Instruções **ACALL** e **AJMP**, permitem desvios de até 2 Kbytes pois utilizam 11 linhas para endereçamento.

Este tipo de instrução é codificada em 2 bytes.

10. Endereçamento Longo

- Instruções **LCALL** e **LJMP**, permitem desvios que mapeiam todo espaço disponível de endereçamento (64K).

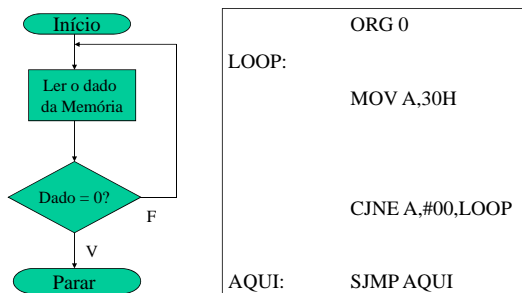
Este tipo de instrução é codificada em 3 bytes.

Modos de Endereçamento do 8051

Endereçamento Combinado ou Misto

MOV R0,10h	;Registrador e Direto
MOV 40h,R3	;Direto e Registrado
MOV R2,#40h	;Registrador e Imediato
MOV 50h,#55h	;Direto e Imediato
MOV @R1,#34h	;Indireto e Imediato
MOV 50h,@R0	;Direto e Indireto
MOV @R0,50h	;Indireto e Direto
MOV 30h,60h	;Direto e Direto

Exemplo de um Programa Assembly do 8051



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	30	LOOP	MOV A,30h
0002	B4 00 FB	FB		CJNE A,#00h,LOOP
0005	80 FE	FE	AQUI	SJMP AQUI

Exemplo de um Programa Assembly do 8051

Memória de Programa

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

↑ ↑
Endereço Conteúdo

Addr	Opcodes	ASC	Label	Disassembly
0000	E5 30	â0	LOOP	MOV A,30h
0002	B4 00 FB	ï0		CJNE A,#00h,LOOP
0005	80 FE	èp	AQUI	SJMP AQUI

Exemplo de um Programa para zerar a RAM interna

```

ORG 0
MOV R0,#0FFH
LOOP:
MOV @R0,#0
DJNZ R0,LOOP
SJMP LOOP
END
    
```

Exercício para casa:

Faça um programa para zerar os 64K bytes da RAM externa.