

# Microprocessadores

---

## Instruções Lógicas e Aritméticas

# Roteiro

---

- Introdução
- Formato das instruções
- Problemas Lógicos
- Problemas Aritméticos
- Referências

# Introdução

---

## Linguagem Assembly

# Introdução

---

A linguagem assembly é o nível mais baixo para o desenvolvimento de software conhecido atualmente, pois atua diretamente na melhoria contínua de desempenho do hardware em nível de microinstruções.

# Introdução

---

Sempre iniciaremos qualquer programa definindo o **nome** e o local do **projeto**, a escolha da **CPU 8051** e, em seguida, **habilitando as memórias** de código e de dados externas com a maior capacidade possível de 65536b e 61440b, respectivamente.

# Introdução

---

Em seguida salve o arquivo do projeto com extensão **(.asm)** com o nome escolhido por você, de tal modo que você não se esqueça dele futuramente (boas práticas de programação)!

# Instruções

---

Formato na IDE

# Instruções

---

As instruções na linguagem contém quatro partes bem definidas, a saber:

- **Rótulo** (opcional)
- **Mnemônico** (obrigatório)
- **Operandos** (depende do opcode)
- **Comentários** (opcional)



# Instruções

## Exemplo

CONT:

MOV A, P1

INC R0

MOV B, R0

ADD A, B

; R0 ← R0 + 1

RÓTULO

MNEMÔNICO

OPERANDO

COMENTÁRIOS

# Instruções

---

## Rótulo

Forma de “apelido” de uma posição do programa para o ponteiro de instrução (PC), o qual identifica uma posição específica do programa que poderá ser vinculada a um procedimento ou função.

# Instruções

## Rótulo

**SOMA:**      **MOV** A, P1

**MOV** B, R0

**ADD** A, B

*; soma dois bytes.*



**RÓTULO**

# Instruções

---

## Mnemônico

É a instrução propriamente dita, consta no *opcode* do processador, ou seja, é bem definida e executa determinada ação após decodificada pela Unidade de Controle.

# Instruções

## Mnemônico

**SOMA:**      **MOV** A, P1      ; *MOV* faz movimentação  
                 **MOV** B, R0      ; *de dados internos à RAM.*  
                 **ADD** A, B

↓

MNEMÔNICO

# Instruções

---

## Operando

É sobre quem se deseja realizar operação.

# Instruções

## Operando

**OPR:**      **MOV** *A*, *P0*    ;  $A \leftarrow P0$

**PUSH** *A*

**CLR** *A*

SÓ PODE USAR “A” COMO OPERANDO

# Instruções

---

## Comentários

São utilizados para facilitar o desenvolvimento na linguagem, não importa para fins de compilação e código.



# Instruções

## Comentários

**OPR:**

**MOV** *A*, *P0*

**PUSH** *A*

**CLR** *A*

*; A recebe P0*

*; joga P0 na pilha.*

*; limpa A.*

BOAS PRÁTICAS DE PROGRAMAÇÃO.

# Problemas Lógicos

---

Linguagem Assembly

# Problemas

## Aritméticos

1. Some **P2** e **SP** e exiba-o em **P1**.
2. Faça o produto entre **R0** e **P3** e exiba-o em **P2**.
3. Subtraia dois valores da RAM interna e envie para **B**.
4. Realize divisão de dois números provenientes de portas duas portas distintas e envie os resultados pelas mesmas portas que foram lidas.
5. Realize a seguinte operação:  $P0 = SP * (P1 - B) + R3$

# Problemas

## Lógicos

1. Verificar se **P1.0** e **A.5** estão habilitados!
2. Verificar se a soma de dois números resultou em um número de nove bits.
3. Como verificar se dois valores são iguais?
4. Movimente três bits quaisquer de **P1** para posições bit endereçáveis da memória RAM.
5. Verifique se @0020h da RAM externa contém 00h.

# Referências

OSMERA, Martin. **MCU 8051 IDE handbook draft**. 2012. Disponível em <[http://www2.sorocaba.unesp.br/professor/marilza/Simulador%20MCU8051/MANUAL\\_MCU8051IDE.pdf](http://www2.sorocaba.unesp.br/professor/marilza/Simulador%20MCU8051/MANUAL_MCU8051IDE.pdf)>. Acesso em Ago/2016.

NICOLOSI, D. E. C. **Microcontrolador 8051 detalhado**. São Paulo: Érica, 2000. 221p. ISBN 9788571947214.

MANZANO, J. A. N. G. **Programação ASSEMBLY padrão IBM-PC 8086/8088**. São Paulo: Érica, 2013. ISBN 9788536516547.

GIMENEZ, S. P. **Microcontroladores 8051**. São Paulo: Prentice Hall, 2002. 270p. ISBN 8587918281.