Arquitetura de Computadores

PROF. ISAAC

Objetivos da matéria

Experimentar programação em baixo nível (**assembly**).

Pra que?

Introduzir o trabalho de um compilador;

Entender melhor como um computador funciona e o que uma linguagem de alto nível faz;

Consequências:

Não reclamar mais de linguagens de programação em alto nível;

Não reclamar mais da IDE;

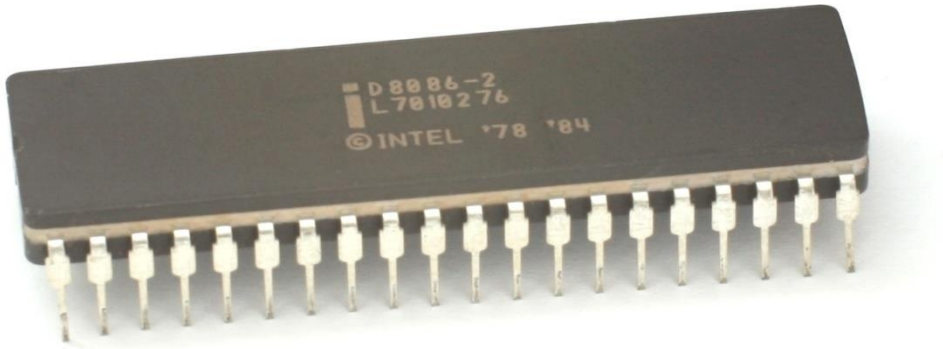
Microprocessador e Microcontrolador

O que é um microcontrolador?

• Qual a diferença entre um microcontrolador e um microprocessador?

Microprocessador

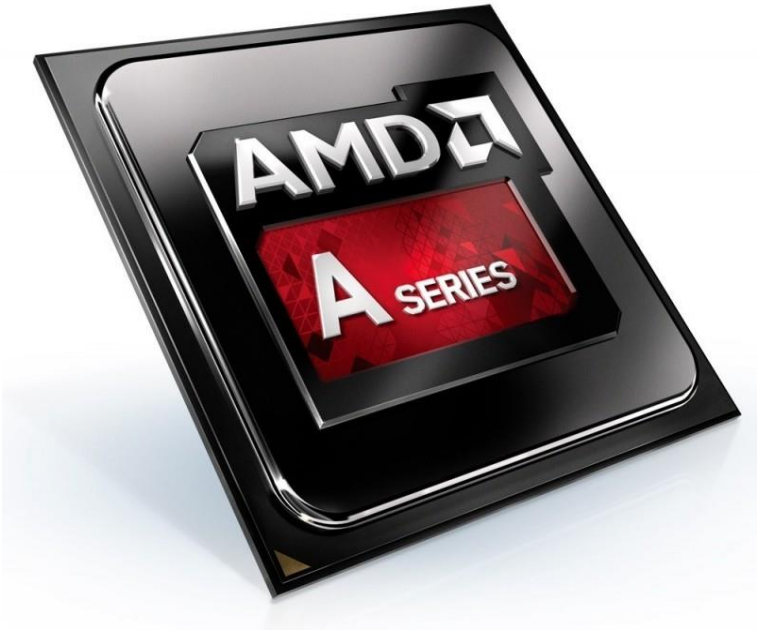
• Microprocessador é um circuito integrado responsável pela execução dos cálculos e das instruções solicitadas por um programa em um computador (tomada de decisão).



Microprocessador 8086

Microprocessador

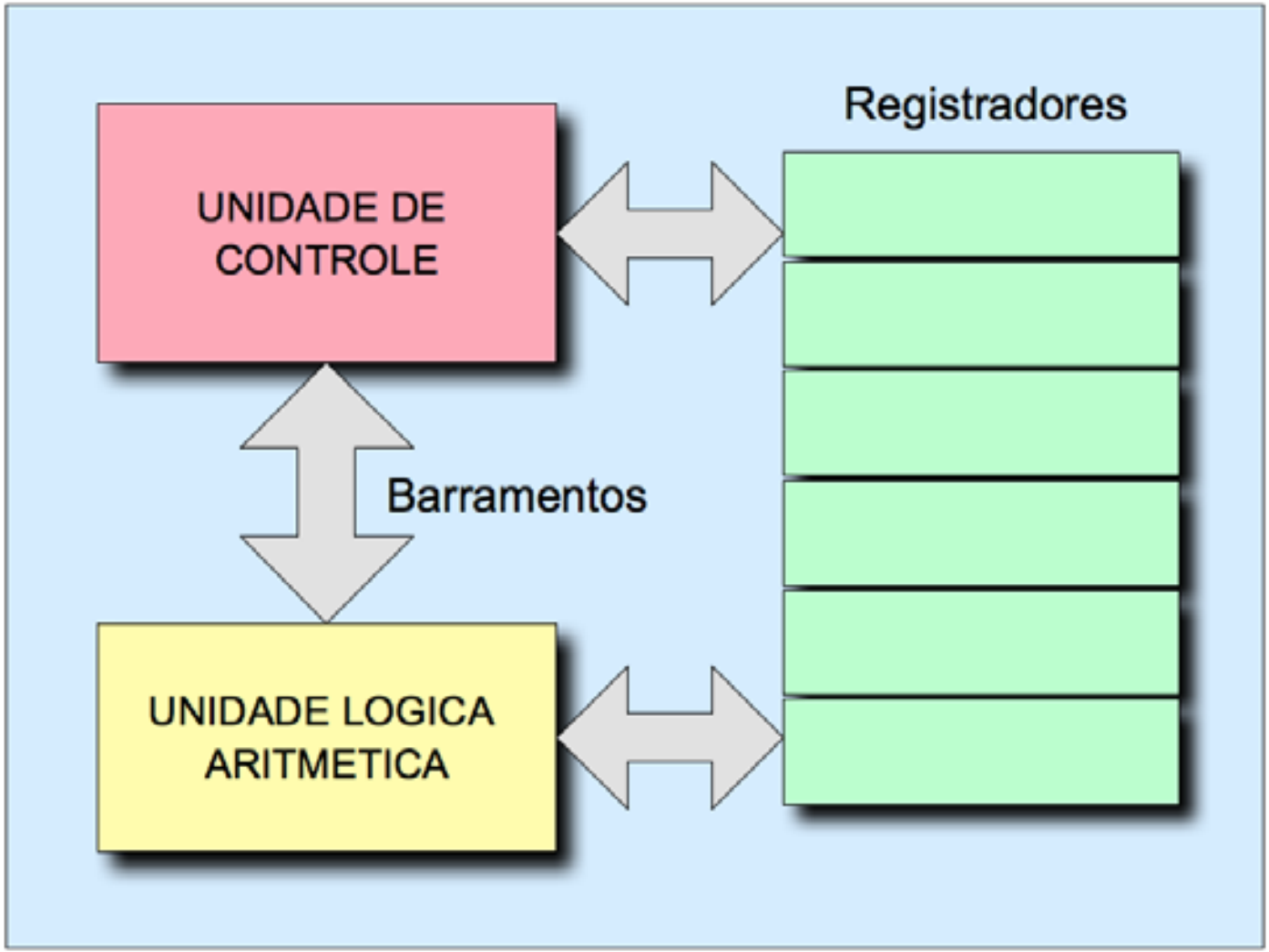
• **Microprocessador** geralmente chamado apenas de **processador** ou **CPU** (do inglês **Central Processing Unit -** Unidade Central de Processamento)



Microprocessador

• É constituído, basicamente, pela unidade lógica aritmética, unidade de controles, registradores e barramentos.

Microprocessador



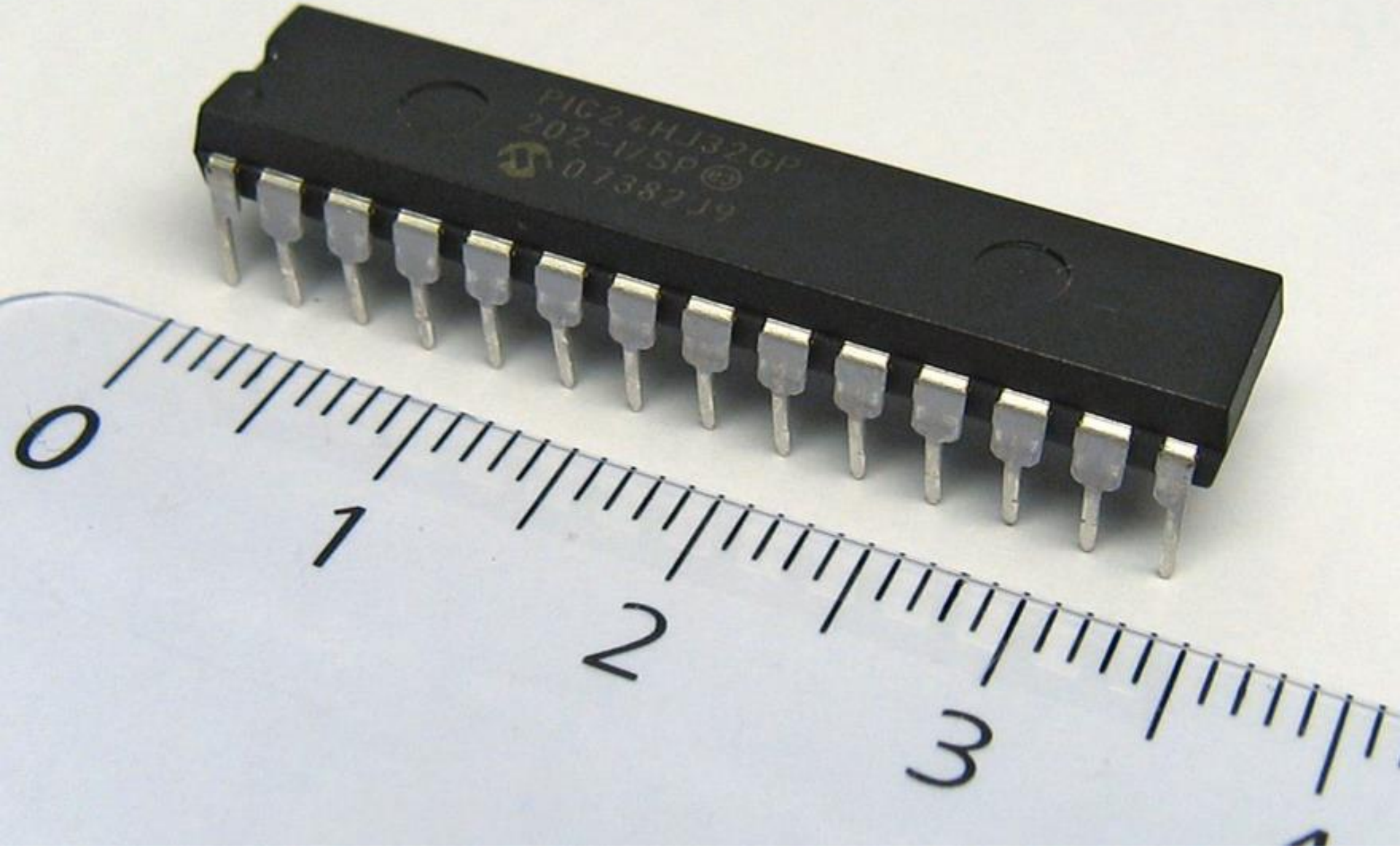
Microprocessador

- Microprocessador não trabalha sozinho.
- Não pode ser programado.
- Apenas executa funções que lhes são solicitadas...

Microprocessador

Microprocessador não trabalha sozinho.

Microcontrolador



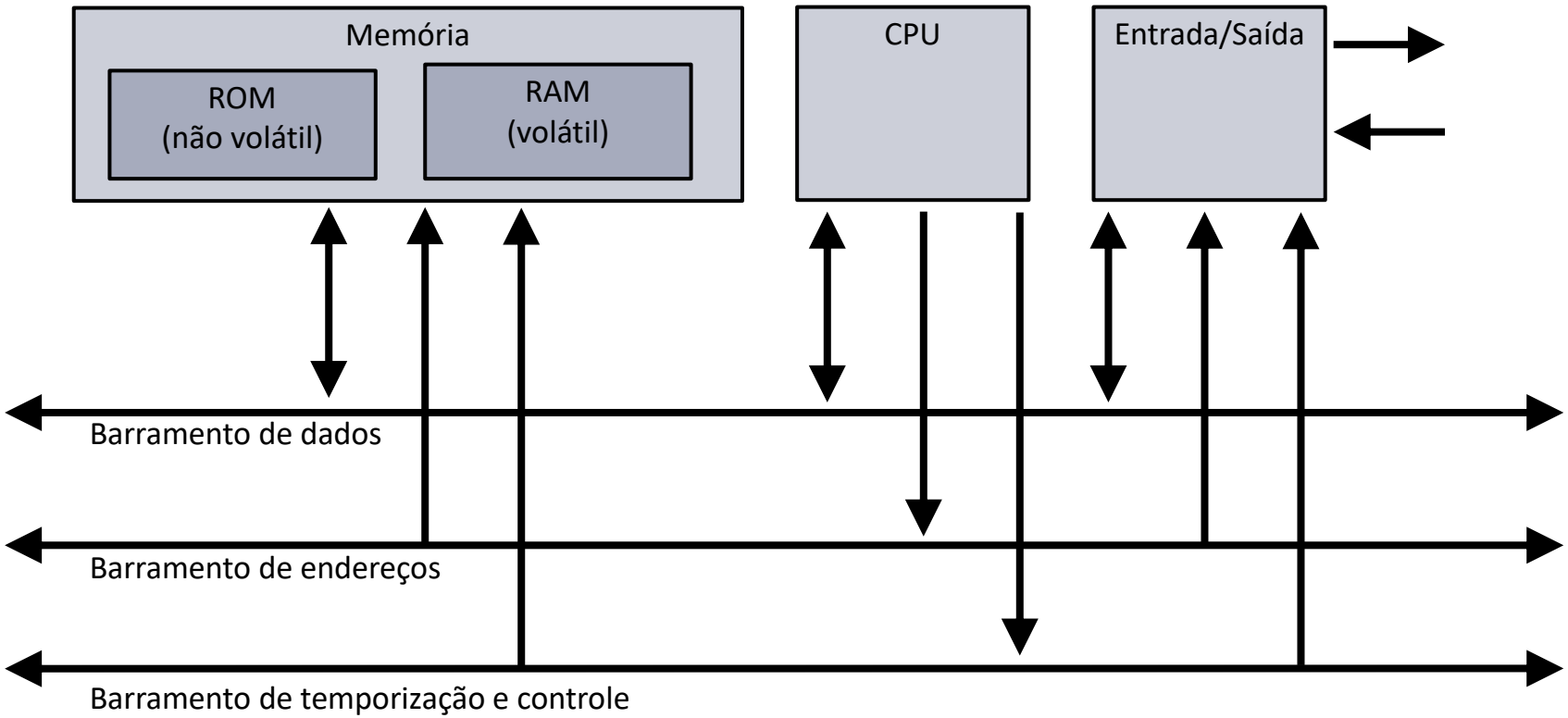
Microcontrolador

Não é o mesmo que um microprocessador;

Um microcontrolador possui:

1. CPU: executar operações em valores;
2. ROM: armazenar programa que deve ser executado;
3. RAM: armazenar valores de variáveis;
4. Pinos de entrada e saída: para leitura e escrita de valores em dispositivos externos;
5. Temporizadores, contadores, interrupção,...

Microcontrolador



Microcontrolador

Além de possuir um **microprocessador** em seu interior:

• Possui uma série de periféricos que permitem que ele realize diversas tarefas, como a comunicação com dispositivos de entrada e saída externos.

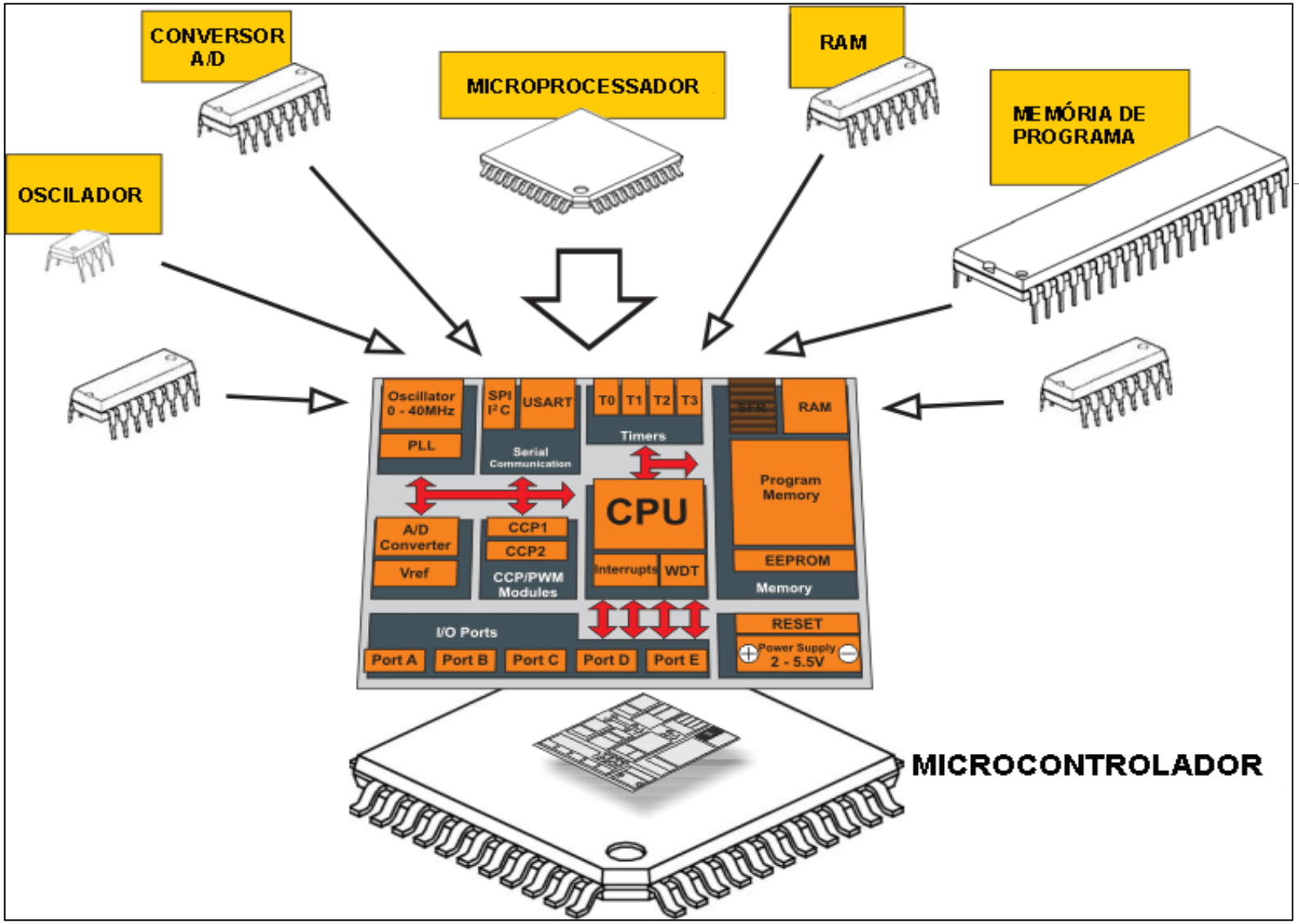
• Além de possuir memória RAM e outros sistemas que o transformam.

Microcontrolador

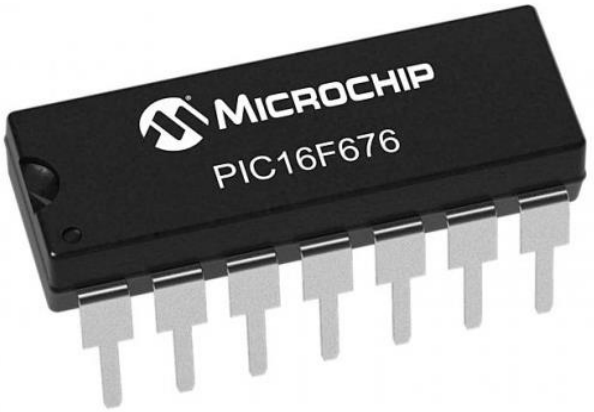
• Os microcontroladores são componentes ideais para serem utilizados em sistemas embarcados

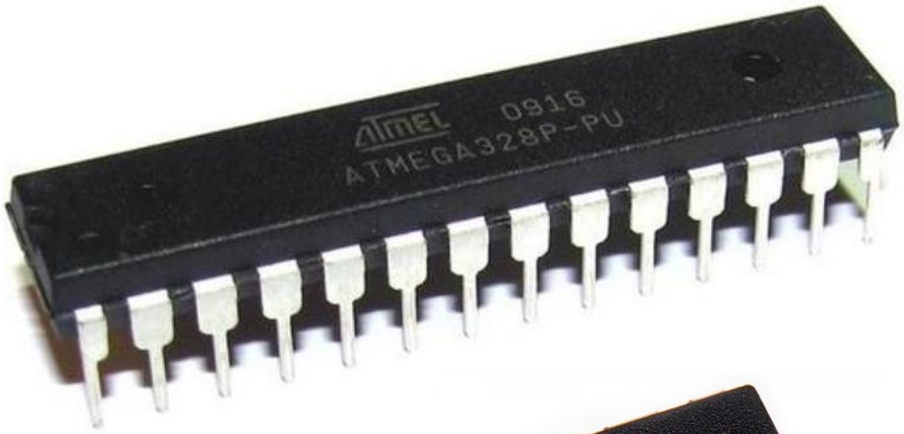
• Microcontroladores reduzem a eletrônica externa necessária para o funcionamento dos equipamentos nos quais estão implantados.

Microcontrolador

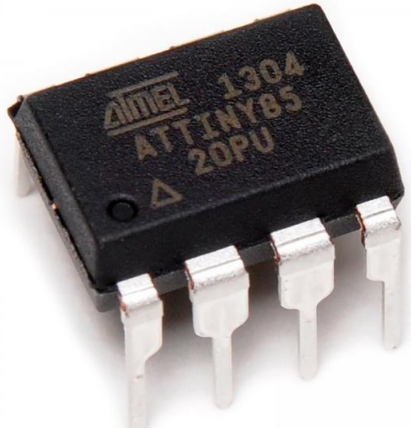


Microcontroladores





Microprocessador 8051



Microcontrolador

- O microcontrolador pode ser programado.
- Executa apenas aquilo que está em seu programa, para executar outras funções deve ser reprogramado.
- **Vantagem**: baixo custo.
- Microcontroladores são baratos porque eles têm recursos limitados.

Microprocessador e Microcontrolador

Qual a diferença entre um microcontrolador e um microprocessador ?

8051
(MSC-51)

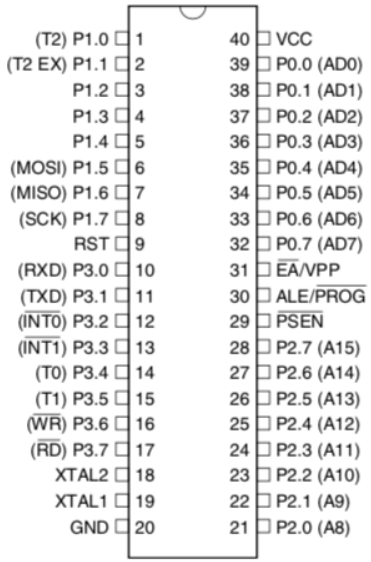
8051

É um microcontrolador CISC projetado pela Intel;

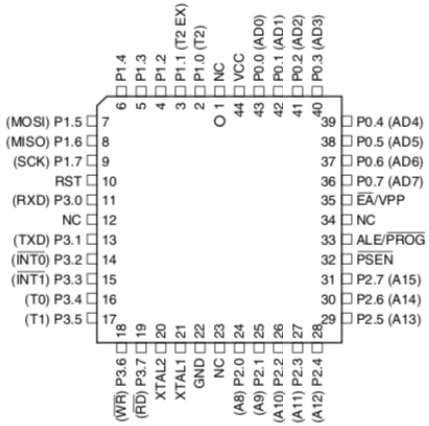
Atualmente possui diversos fabricantes (Philips, ATMel, NXP, etc);

Conjunto de instruções pequeno;

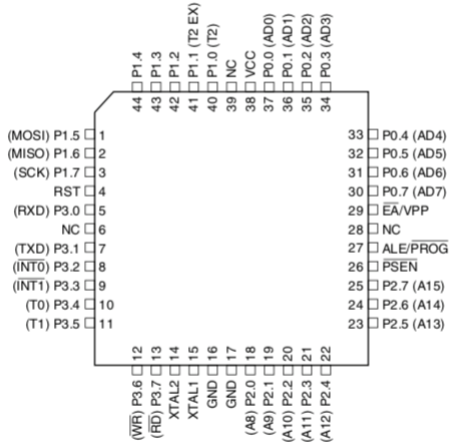
40-lead PDIP



44-lead PLCC

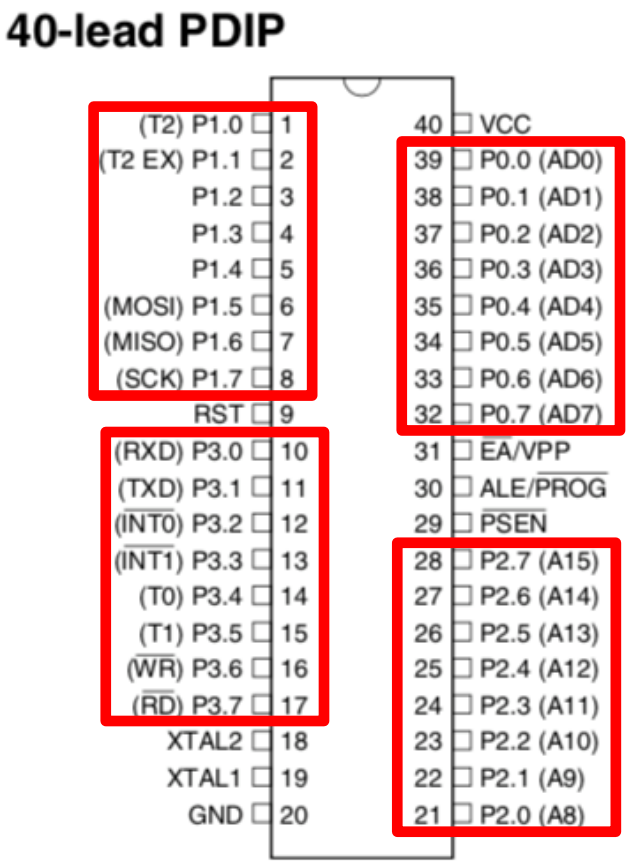


44-lead TQFP



Detalhes do 8051

- Trabalha com 8-bits;
- 4 ports de 8-bits;
- Endereçamento de 16-bits;
- ROM até 64KB;
- RAM interna até 256B:
  - 128B para programa;
  - 128B para registradores especiais de funções (SFR);
- É possível conectar uma RAM externa de até 64KB.



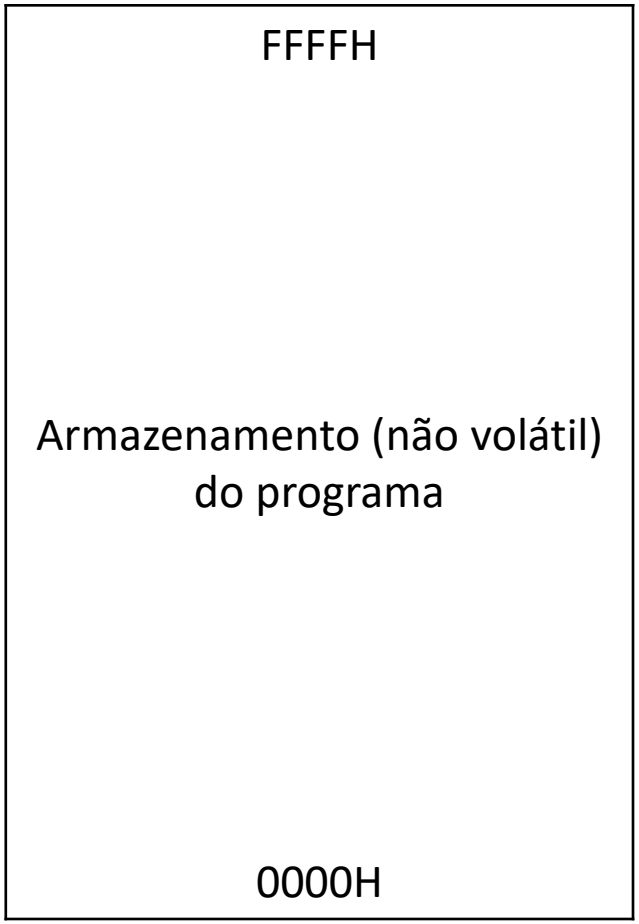
Memória do 8051 – ROM

Endereçamento de 16-bits;

- Início em 0000H (0);
- Final em FFFFH (65.535);

Cada posição aponta para 1 Byte;

No máximo, 64KB para armazenar o programa (podendo ser menor conforme o fabricante);



Memória do 8051 – RAM

Endereçamento de 16-bits;

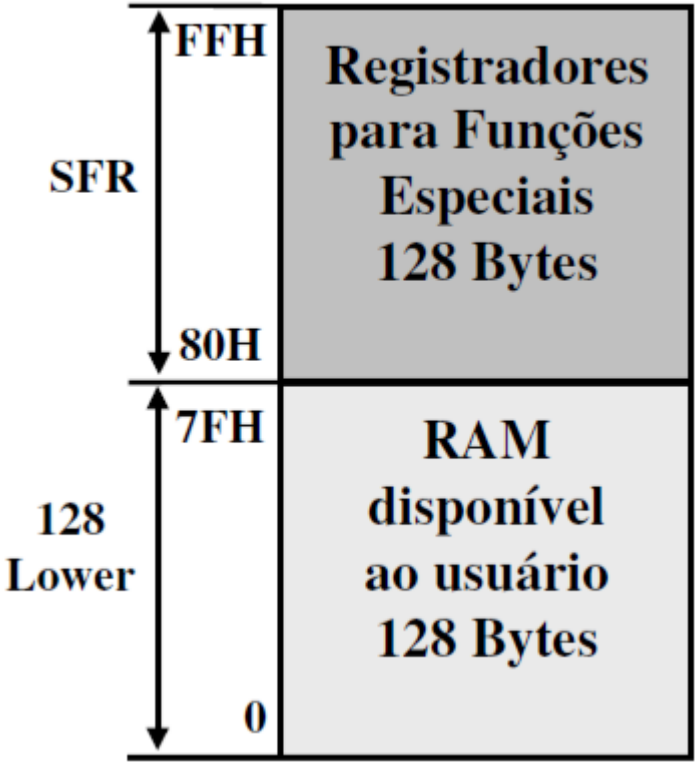
Para o programa:

- Início em 00H
- Final em 7FH
- Área que o programa pode usar para armazenar os dados

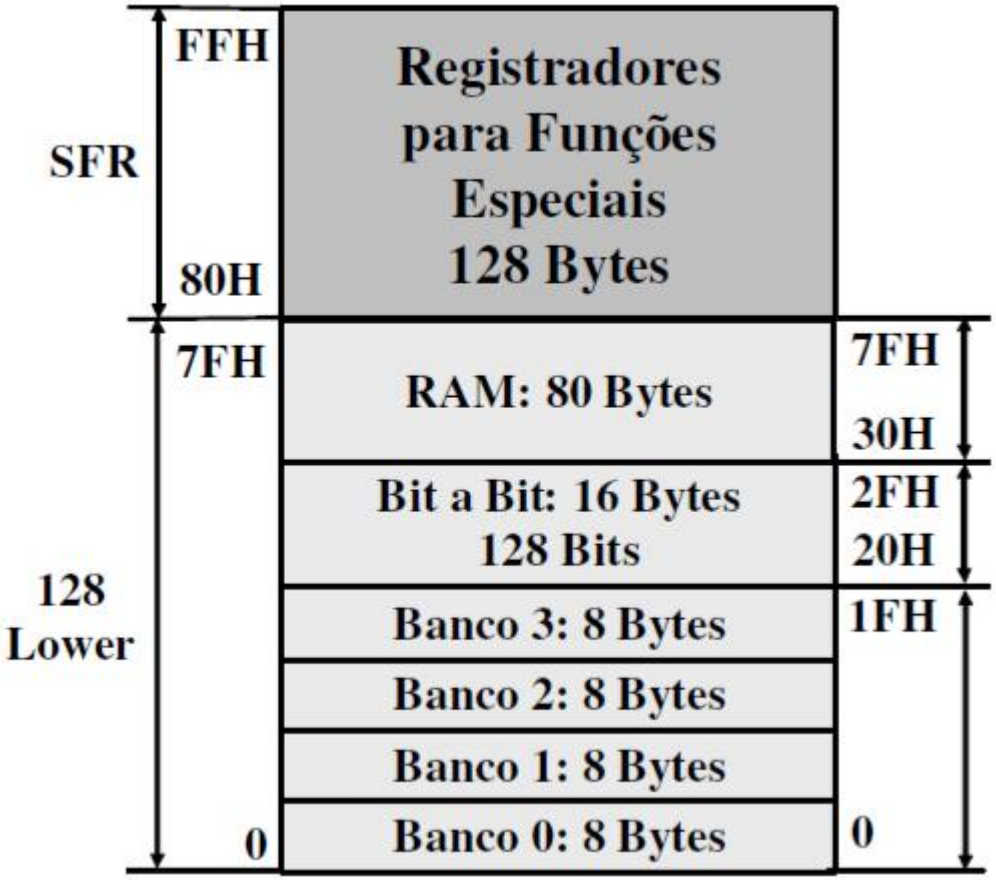
Uso interno:

- Início em 80H
- Final em FFH

Memória do 8051 – RAM



Divisão da RAM interna.



Subdivisões da metade inferior da RAM interna.

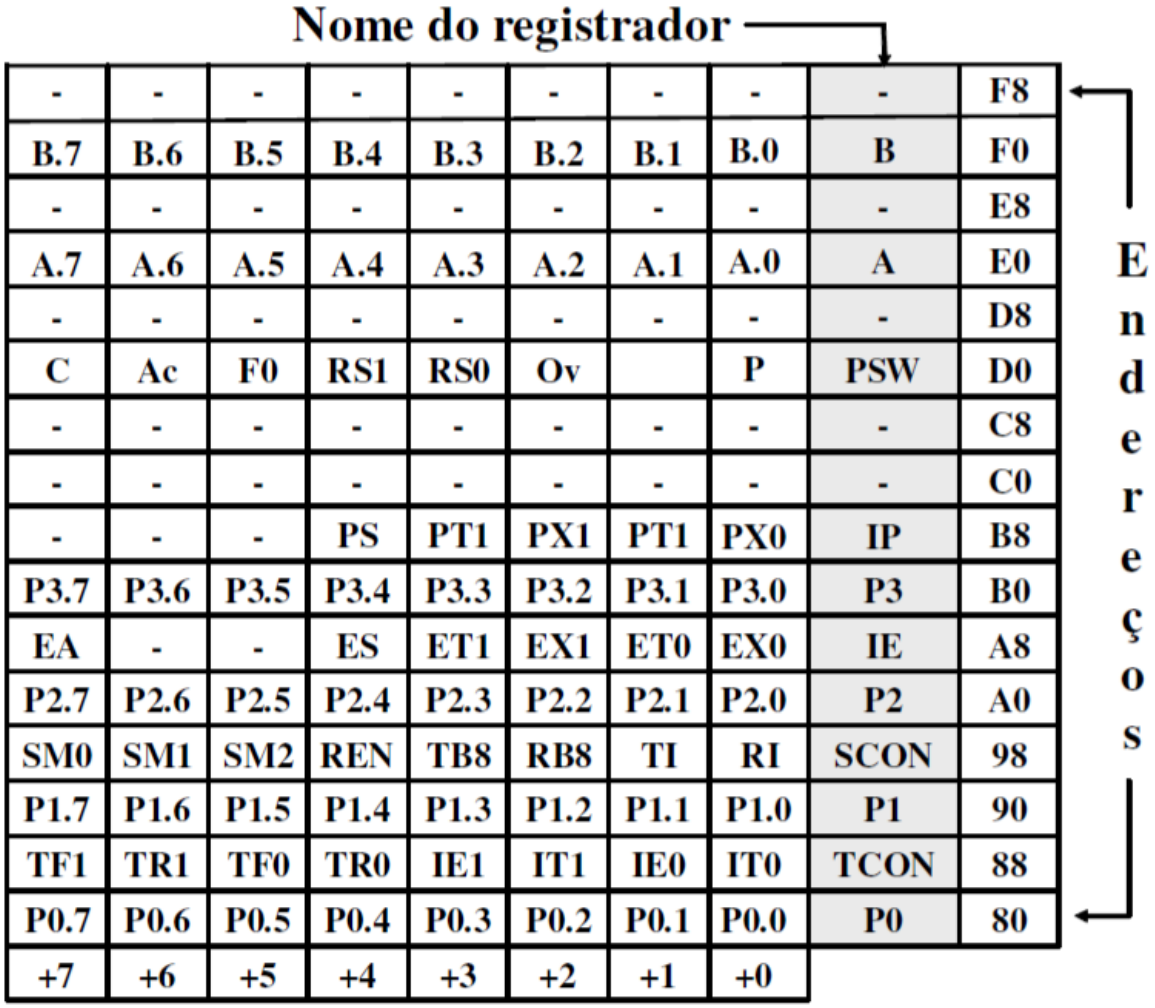
Registradores

Endereços de memória que possuem funções especiais;

No 8051:

- A (acumulador): utilizado em (quase) todas as operações aritméticas;
- R0 até R7: são 8 registradores que podem ser acessados diretamente, além dos 127B de RAM;
- R0 e R1: também podem ser acessados indiretamente;
- DPTR: ponteiro na RAM externa;
- PSW: Program Status Word.

Registradores



-	-	-	-	-	-	-	-	-	F8
B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0	B	F0
-	-	-	-	-	-	-	-	-	E8
A.7	A.6	A.5	A.4	A.3	A.2	A.1	A.0	A	E0
-	-	-	-	-	-	-	-	-	D8
C	Ac	F0	RS1	RS0	Ov		P	PSW	D0
-	-	-	-	-	-	-	-	-	C8
-	-	-	-	-	-	-	-	-	C0
-	-	-	PS	PT1	PX1	PT1	PX0	IP	B8
P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	P3	B0
EA	-	-	ES	ET1	EX1	ET0	EX0	IE	A8
P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	P2	A0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI	SCON	98
P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	P1	90
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	TCON	88
P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	P0	80
+7	+6	+5	+4	+3	+2	+1	+0		

O mapeamento dos 128 bits pertencentes aos registradores SFR.

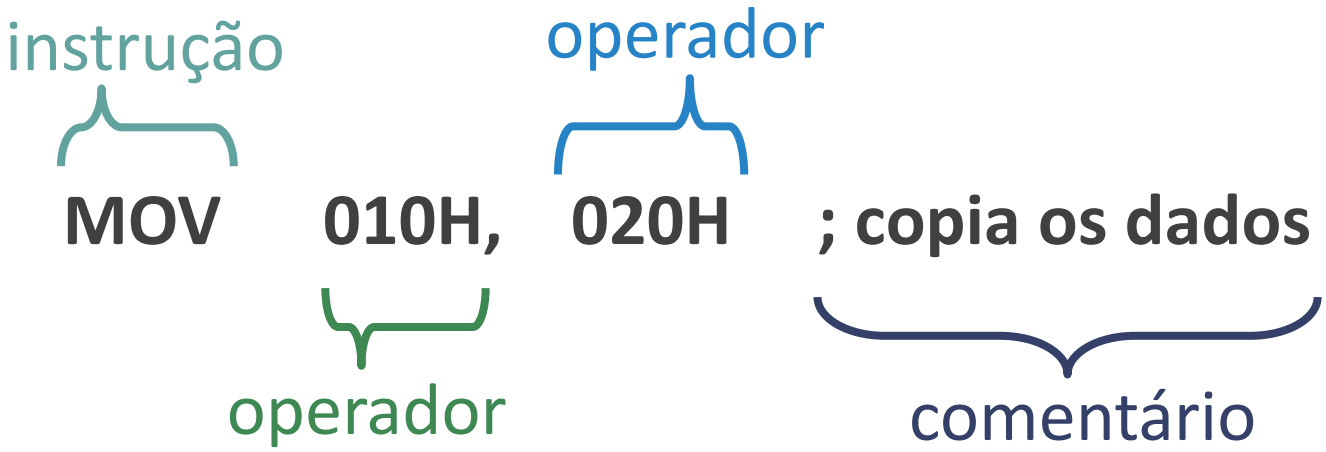
Assembly!

Assembly

A programação depende dos dados anteriores;

- No início da programação é necessário informar o endereço em que o programa será armazenado na ROM;
- Sabe-se que não será possível ter programas muito grandes por conta da limitação de 64KB;
- Sabe-se que teremos no máximo 127B para armazenar dados durante a execução do programa;
- Sabe-se que existem endereços da RAM que não devemos usar (acima do 7FH);
- Sabe-se que a linguagem vai oferecer instruções para execução de operações lógicas, aritméticas e de movimentação na memória;

Sintaxe



- O nome de uma instrução possui entre 2 e 5 letras;
- Instruções podem conter até 3 operadores;
- Um comentário é iniciado por ponto e vírgula depois da instrução;
- Os operadores podem ser valores, portas, endereços de memória, registradores, etc.

Sintaxe – Operadores

Tipo de Operador	Sintaxe
Registrador	MOV A, R0
Endereço	MOV A, 010H
Valor	MOV A, #010H
Valor (acesso indireto)	MOV A, @R0

Montagem

Para que possam ser executados, esses programas devem ser convertidos em uma lista de **códigos de operação** ou **opcodes**, que serão armazenados na memória.

Essa tarefa de converter uma lista de mnemônicos em opcodes recebe o nome de montagem, logo após a montagem se denomina programa montado ou programa executável.

8051 – Código objeto

Instrução	Cod. HEX	num. bytes
nop	00	1
inc R0	08	1
mov A, #55H	74 55	2
mov R0, #0AAH	78 AA	2
mov R0, 0AAH	A8 AA	2
mov DPTR, #55AAH	90 55 AA	3

8051 – Código objeto

Instrução	Cod. HEX	num. bytes
nop	00	1
inc R0	08	1
mov A, #55H	74 55	2
mov R0, #0AAH	78 AA	2
mov R0, 0AAH	A8 AA	2
mov DPTR, #55AAH	90 55 AA	3

		Bytes	MC	Op1	Op2
INC	A	1	1	04	-
	Rn	1	1	08+n	-
	end8	2	1	05	end8
	@Ri	1	1	06+i	-

Instruções de incremento de 8 bits.

8051 – Código objeto

Hex	Assembly
0000:	.equ cout, 0x0030
0000:	.equ cin, 0x0032
0000:	.equ esc, 0x004E
8000:	.org 0x8000
8000: 79 61	mov r1, #'a'
8002: 78 1A	mov r0, #26
	next_char:
8004: E9	mov A, r1
8005: 12 00 30	lcall cout
8008: 09	inc r1
8009: D8 F9	djnz r0, next_char
800B: 12 00 32	lcall cin
800E: 02 00 00	ljmp 0x0000

Address	Value
8000	79
8001	61
8002	78
8003	1A
8004	E9
8005	12
8006	00
8007	30
8008	09
8009	D8
800A	F9
800B	12
800C	00
800D	32
800E	02
800F	00
8010	00

Instruções do 8051

Lógica	Aritmética	Memória	Outros
ANL	ADD	MOV	NOP
ORL	ADDC	MOVC	RET e RETI
XRL	SUBB	MOVX	ACALL e LCALL
CLR	MUL	PUSH	JMP
CPL	DIV	POP	AJMP
RL	INC	XCH	LJMP
RLC	DEC	XCHD	SJMP
RR	DA		JB e JNB
RRC			JZ e JNZ
SWAP			JC e JNC
SETB			JBC
			DJNZ
			CJNE

Instruções do 8051

Lógica	Aritmética	Memória	Outros
ANL	ADD	MOV	NOP
ORL	ADDC	MOVC	RET e RETI
XRL	SUBB	MOVX	ACALL e LCALL
CLR	MUL	PUSH	JMP
CPL	DIV	POP	AJMP
RL	INC	XCH	LJMP
RLC	DEC	XCHD	SJMP
RR	DA		JB e JNB
RRC			JZ e JNZ
SWAP			JC e JNC
SETB			JBC
			DJNZ
			CJNE

Instrução - MOV

Operação: MOV

Função: Transferências na RAM Interna.

Sintaxe: MOV *operando1*,*operando2*

Descrição: MOV copia o valor do *operando2* no *operando1*, sem que o valor do *operand2* seja modificado. Ambos os operandos devem estar na RAM interna. Nenhuma flag é afetada, a não ser que a instrução esteja movendo um valor para PSW diretamente.

Exemplo:

- mov A, #32
- mov A, R0
- mov R2, A

Instrução - MOV

		bytes	MC	Op1	Op2	Op3
MOV A,	Rn	1	1	E8+n	-	-
	end8	2	1	E5	end8	-
	@Ri	1	1	E6+i	-	-
	#dt8	2	1	74	dt8	-
MOV Rn,	A	1	1	F8+n	-	-
	end8	2	2	A8+n	end8	-
	#dt8	2	1	78+n	dt8	-
MOV end8,	A	2	1	F5	end8	-
	Rn	2	2	88+n	end8	-
	end8	3	2	85	end8 (fonte)	end8 (destino)
	@Ri	2	2	86+i	end8	-
	#dt8	3	2	75	end8	dt8
MOV @Ri	A	1	1	F6+i	-	-
	end8	2	2	A6+i	end8	-
	#dt8	2	1	76+i	dt8	-
MOV DPTR	#dt16	3	2	90	MSB(dt16)	LSB(dt16)

***MC (Machine Cycle) ciclos de máquina.**

Instrução - MOV

MOV	A	,	A
	Rn	,	Rn
	Rn	,	@Ri
	@Ri	,	@Ri
	@Ri	,	Rn

Combinações de operandos que são proibidas.

***MC (Machine Cycle) ciclos de máquina.**

Instrução - ADD

Operação: ADD

Função: Adiciona o valor do operando ao valor do acumulador.

Sintaxe: ADD A, *operando*

Descrição : ADD adiciona o valor do operando ao valor do acumulador.

- **Carry bit (C)** – setado se existe um carry saindo do bit 7. Em outras palavras, se a soma do valor no acumulador e o operando excede 255.
- **Overflow (OV)** – setado se existir um carry saindo do bit 6 ou do bit 7, mas não dos dois. Em outras palavras, se a soma do acumulador e operando excede a faixa do valor armazenado em um byte com sinal (-128 até +127) o OV é setado, caso contrário, seu valor estará em 0.

Exemplo:

- ADD A, #03h
- ADD A, R0
- ADD A, @R1

Instrução - ADD

		Bytes	MC	Op1	Op2
ADD A,	Rn	1	1	28+n	-
	end8	2	1	25	end8
	@Ri	1	1	26+i	-
	#dt8	2	1	24	dt8

***MC (Machine Cycle) ciclos de máquina.**

Instrução - DEC

Operação: DEC

Função: Decrementa o operando em 1

Sintaxe: DEC *operando*

Descrição : DEC decrementa o valor do *operando* em1. Se o valor inicial do *operando* for 0, decrementar seu valor significará um reset para 255 (0xFF H). Obs: o C não é setado quando o valor muda de 0 to 255.

Exemplo:

- DEC A
- DEC R0

Instrução - INC

Operação: INC

Função: Incrementa o operando

Sintaxe: INC *operando*

Descrição : INC incrementa o valor do *operando* em 1. Se o valor inicial do *operando* for 255 (0xFF Hex), o incremento causará um reset para 0.

Obs: **Carry (C)** não será setado quando o valor passar de 255 para 0.

Exemplo:

- INC A
- INC R0

Instruções – INC e DEC

		Bytes	MC	Op1	Op2
INC	A	1	1	04	-
	Rn	1	1	08+n	-
	end8	2	1	05	end8
	@Ri	1	1	06+i	-

		Bytes	MC	Op1	Op2
DEC	A	1	1	14	-
	Rn	1	1	18+n	-
	end8	2	1	15	end8
	@Ri	1	1	16+i	-

***MC (Machine Cycle) ciclos de máquina.**

Instrução - INC

Operação: INC

Função: Incrementa o registrador DPTR.

Sintaxe: INC *DPTR*

Descrição : No conjunto de instruções do 8051, existe apenas um único incremento de 16 bits e envolve o DPTR, que é o registrador de 16 bits dessa arquitetura. Deve-se notar que não existe uma instrução para decremento de 16 bits.

Exemplo:

- INC DPTR

Instrução - INC

	Bytes	MC	Op
INC DPTR	1	2	A3

Instrução de incremento de 16 bits.

***MC (Machine Cycle) ciclos de máquina.**

Instruções do 8051

Lógica	Aritmética	Memória	Outros
ANL	ADD ✓	MOV ✓	NOP
ORL	ADDC	MOVC	RET e RETI
XRL	SUBB	MOVX	ACALL e LCALL
CLR	MUL	PUSH	JMP
CPL	DIV	POP	AJMP
RL	INC ✓	XCH	LJMP
RLC	DEC ✓	XCHD	SJMP
RR	DA		JB e JNB
RRC			JZ e JNZ
SWAP			JC e JNC
SETB			JBC
			DJNZ
			CJNE

Modos de Endereçamento:

Modo de endereçamento	Operandos (Registradores e Memória)
Endereçamento imediato	Memória de Programa
Por registrador	R0-R7 e A (ACC), B, C (*carry-bit*) e DPTR
Direto	Os 128 *bytes* menos significativos da RAM Interna e Registradores de Funções Especiais
Indireto ou indexado por registrador	RAM interna (@R0, @R1 e SP) e Memória de Dados Externa (@R0, @R1 e @DPTR)
Registrador base mais indireto ou indexado por registrador	Memória de Programa (@DPTR+A e @PC+A)

1- Endereçamento por Registrador:

MOV A,R1

Representação simbólica da instrução: (A) ← (R1)

Significado da instrução: o conteúdo do registrador R1 será copiado para o conteúdo do acumulador (A) (operação de escrita no acumulador).

2- Endereçamento Direto:

MOV A,30h

Representação simbólica da instrução: (A) ← (30h)

Significado da instrução: o conteúdo da posição de memória cujo endereço é 30h é copiado para o conteúdo do registrador acumulador (operação de escrita no registrador acumulador).

Modos de Endereçamento:

3- Endereçamento Indireto ou Indexado por Registrador:

MOV A,@R0
Representação simbólica: (A) ← ((R0)). Observe que nessa instrução existe um símbolo de arroba e também existem dois entre parênteses no segundo operando.
Significado da instrução: o conteúdo da posição de memória cujo endereço é fornecido pelo conteúdo do registrador R0 é copiado para o conteúdo do registrador acumulador (operação de leitura de uma posição de memória RAM interna para o registrador acumulador).

4- Endereçamento Imediato:

MOV A,#3Ah
Representação simbólica: (A) ← #3Ah (repare que existe o símbolo de sustenido "#" nessa instrução)
Significado da instrução: o conteúdo do registrador acumulador será inicializado (operação de inicialização do acumulador) com o valor constante 3Ah.

5- Endereçamento Indireto por Registrador Base mais registrador Indexado:

MOVC A, @A+DPTR
Representação simbólica: (A) ← ((A) + (DPTR))
Significado da instrução: o conteúdo da posição de memória de programa (ROM/EPROM/EEPROM/*Flash*) cujo endereço é dado pela soma do conteúdo do registrador acumulador (A ou ACC) e o conteúdo do registrador *data pointer* (DPTR) é copiado para o conteúdo do registrador acumulador (A) (operação de leitura da memória de programa externa para o acumulador).

6- Endereçamento Combinado ou Misto:

MOV R0,20h
Representação simbólica: (R0) ← (20h)
Modo de endereçamento combinado (misto): por registrador e direto.

MOV @R1,#33h
Representação simbólica: ((R1)) ← #33h

Exemplos/Exercícios

1) Escreva as instruções para os seguintes casos:

a) Mova para o registrador R0 o valor de 10h;
b) Mova para o acumulador o valor que está no registrador R0;
c) Mova para o endereço 50h o valor de 20h;
d) Incremente o valor que está no registrador R2;
e) Decremente o valor do acumulador;
f) Deixe o valor do acumulador como zero;

Exercícios

2) Escreva um código para as seguintes equações:

a) A = A + R0;

b) A = R3 + R2;

c) A = R5 + 30;

d) A = R1 – 1;

e) R2 = 31 + R1;

f) R2 = R0 + R1 (com acesso indireto aos valores de R0 e R1);

Exercícios

3) Escreva um código que coloque cada digito do seu número de matrícula na memória de dados iniciando no endereço 20h.

Exemplo: 22113083-4

Endereço : valor

30h : 02
31h : 02
32h : 01
33h : 01
34h : 03
35h : 00
36h : 08
37h : 03
38h : 04

Instruções

		Bytes	MC	Op1	Op2
ADD A,	Rn	1	1	28+n	-
	end8	2	1	25	end8
	@Ri	1	1	26+i	-
	#dt8	2	1	24	dt8

		Bytes	MC	Op1	Op2
INC	A	1	1	04	-
	Rn	1	1	08+n	-
	end8	2	1	05	end8
	@Ri	1	1	06+i	-

		Bytes	MC	Op1	Op2
DEC	A	1	1	14	-
	Rn	1	1	18+n	-
	end8	2	1	15	end8
	@Ri	1	1	16+i	-

		bytes	MC	Op1	Op2	Op3
MOV A,	Rn	1	1	E8+n	-	-
	end8	2	1	E5	end8	-
	@Ri	1	1	E6+i	-	-
	#dt8	2	1	74	dt8	-
MOV Rn,	A	1	1	F8+n	-	-
	end8	2	2	A8+n	end8	-
	#dt8	2	1	78+n	dt8	-
MOV end8,	A	2	1	F5	end8	-
	Rn	2	2	88+n	end8	-
	end8	3	2	85	end8 (fonte)	end8 (destino)
	@Ri	2	2	86+i	end8	-
	#dt8	3	2	75	end8	dt8
MOV @Ri	A	1	1	F6+i	-	-
	end8	2	2	A6+i	end8	-
	#dt8	2	1	76+i	dt8	-
MOV DPTR	#dt16	3	2	90	MSB(dt16)	LSB(dt16)

Respostas exercício 1

a) MOV R0, #10H

b) MOV A, R0

c) MOV 50h, #20H

d) INC R2

e) DEC A

f) MOV A, #0H

Bibliografia

Gimenez, Salvador P. Microcontroladores 8051 - Teoria e Prática, Editora Érica, 2010.

ZELENOVSKY, R.; MENDONÇA, A. Microcontroladores Programação e Projeto com a Família 8051. MZ Editora, RJ, 2005.