Arquitetura de Computadores

PROF. DR. ISAAC

Unidades métricas nos sistemas computacionais

- Os computadores digitais processam a informação através de **bits**. Um bit é a **menor unidade** num sistema digital e pode assumir o valor 0 ou 1. O agrupamento de **8 bits** forma um **byte** e pode armazenar um valor numérico de 0 a 255 ou representar uma letra;

- Para medir o tamanho das memórias, discos, arquivos e banco de dados a unidade básica de medida é o **byte** e os seus múltiplos são 2^{10}.

Unidades métricas nos sistemas computacionais

Expoente	Unidade	Abreviatura	Valor explícito
2^0	*byte*	1 B	1 *byte* ou 8 *bits*
2^{10}	*kilobyte*	1 KB	1.024 *bytes*
2^{20}	*megabyte*	1 MB	1.048.576 *bytes* ou 1024 KB
2^{30}	*gigabyte*	1 GB	1.073.741.824 *bytes* ou 1024 MB
2^{40}	*terabyte*	1 TB	1.099.511.627.776 *bytes* ou 1024 GB
2^{50}	*petabyte*	1 PB	1.125.899.906.842.624 *bytes* ou 1024 TB
2^{60}	*exabyte*	1 EB	1.152.921.504.606.846.976 *bytes* ou 1024 PB
2^{70}	*zetabyte*	1 ZB	1.180.591.620.717.411.303.424 *bytes* ou 1024 EB
2^{80}	*yotabyte*	1 YB	1.208.925.819.614.629.174.706.176 *bytes* ou 1024 ZB

Diretivas Assembly

Diretiva	Significado
BIT	Especifica um bit da RAM interna
CODE	Define um símbolo para a memória de programa
DATA	Define um símbolo para a RAM interna
DB	Armazena um byte na memória de programa
DBIT	Reserva espaço na área de bits
DS	Reserva espaço na memória de dados
DW	Armazena uma constante de 16 bits na memória de programa
END	Indica o fim do programa fonte
EQU	Atribui um valor numérico a um símbolo
ORG	Define valor inicial do contador do segmento
XDATA	Define um símbolo para a RAM externa

Exercícios

Exercício 1:

Escreva na memória do programa um número inteiro positivo usando a diretiva DB.

Exemplo:

```
org 0050h   ;posição 50 da memória do programa
DB 10       ; número inteiro 10 em decimal
```

Verifique o valor que foi salvo na memória ao executar o programa.

Exercícios:

Exercício 2:

Escreva na memória do programa um número inteiro negativo usando a diretiva DB.

Verifique o valor que foi salvo na memória ao executar o programa.

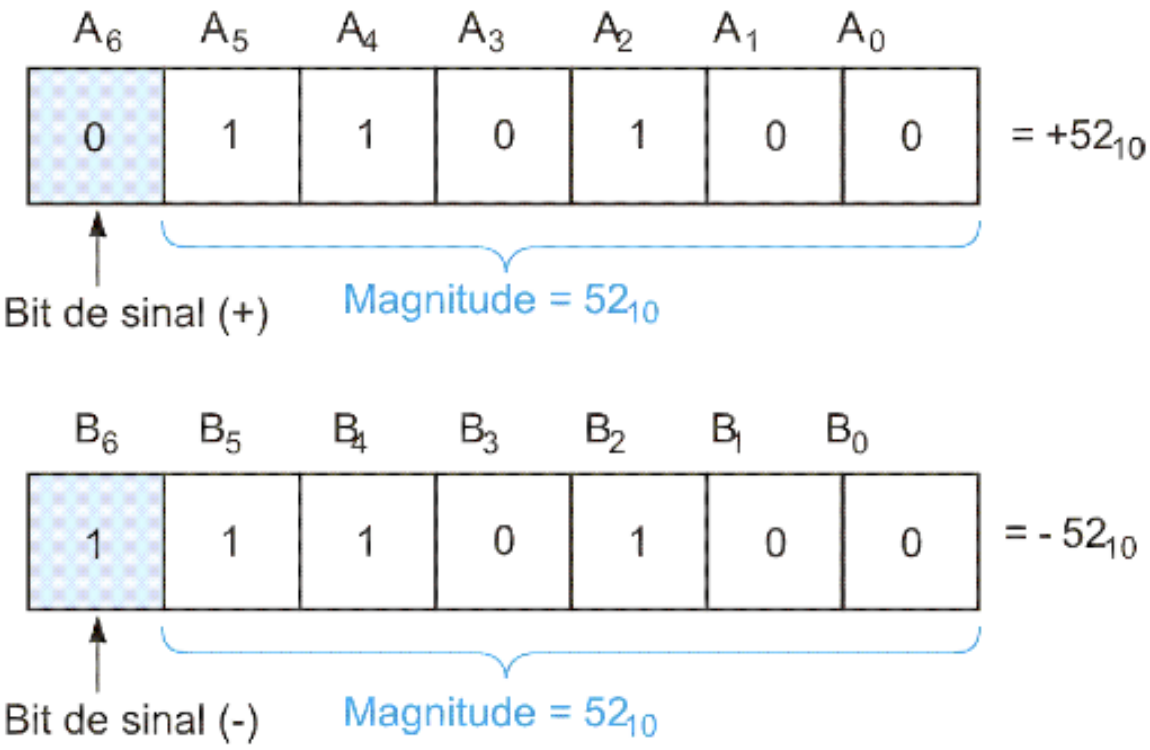
Esse número foi gravado na memória a representação de bit de sinal ou complemento de 2?

Números Inteiros

Formas de representar:

- 1. Bit de sinal (Sign-Magnitude).
- 2. Complemento de 2 (mais utilizado).

Números Inteiros – Bit de Sinal



Números Inteiros – Bit de Sinal

Exemplo:

Como é representado o número **6** utilizando a representação por bit de Sinal em máquinas de 8 bits?

Solução:

- **00000110**

Números Inteiros – Bit de Sinal

Exemplo:

Como é representado o número **-6** utilizando a representação por bit de Sinal em máquinas de 8 bits?

Solução:

- **10000110**

Complemento de 2

Passos:

1. Obter o complemento de 1 (inversão dos bits).

2. Adicionar1.

$1\,0\,1\,1\,0\,1_2$	Número binário original
↓ ↓ ↓ ↓ ↓ ↓	
$0\,1\,0\,0\,1\,0_2$	Complemento de 1
$+\,1$	
$0\,1\,0\,0\,1\,1_2$	Complemento de 2

Números inteiros: Complemento de 2 com Sinal

Utiliza-se um bit de sinal acrescido da representação em complemento de 2 se o número for negativo.

Números inteiros:
Complemento de 2 com Sinal

Exemplo:

Como é representado o número **5** utilizando a representação por complemento de 2 em máquinas de 8 bits?

Números inteiros:
Complemento de 2 com Sinal

Exemplo:

Como é representado o número **5** utilizando a representação por complemento de 2 em máquinas de 8 bits?

Solução:

- **00000101**

Números inteiros:
Complemento de 2 com Sinal

Exemplo:

Como é representado o número **-5** utilizando a representação por complemento de 2 em máquinas de 8 bits?

Números inteiros:
Complemento de 2 com Sinal

Exemplo:

Como é representado o número **-5** utilizando a representação por complemento de 2 em máquinas de 8 bits?

Solução:

- **11111011**

Inteiro sem sinal: exemplo de 4 bits

Esta é a representação mais simples, os números de 0 a 15 podem ser representados. Não tem números negativos.

0000 0	0001 1	0010 2	0011 3
0100 4	0101 5	0110 6	0111 7
1000 8	1001 9	1010 10	1011 11
1100 12	1101 13	1110 14	1111 15

Números Inteiros – Bit de Sinal
exemplo de 4 bits

Bit mais significativo: 0 - positivo; 1 - negativo.

Números de [-7 a 7]. 0 é duplicado.

0000 0	0001 1	0010 2	0011 3
0100 4	0101 5	0110 6	0111 7
1000 0	1001 -1	1010 -2	1011 -3
1100 -4	1101 -5	1110 -6	1111 -7

Números Inteiros – Complemento de 2
exemplo de 4 bits

Números de [-8 a 7].

0000 0	0001 1	0010 2	0011 3
0100 4	0101 5	0110 6	0111 7
1000 -8	1001 -7	1010 -6	1011 -5
1100 -4	1101 -3	1110 -2	1111 -1

Exercícios

Exercício 3:

Escreva na memória do programa um caractere (letra) usando a diretiva DB.

Verifique o valor que foi salvo na memória ao executar o programa.

ASCII

Os caracteres são representados em apenas 7 bits.

00	NUL	10	DLE	20	SP	30	0	40	@	50	P	60	`	70	p
01	SOH	11	DC1	21	!	31	1	41	A	51	Q	61	a	71	q
02	STX	12	DC2	22	"	32	2	42	B	52	R	62	b	72	r
03	ETX	13	DC3	23	#	33	3	43	C	53	S	63	c	73	s
04	EOT	14	DC4	24	$	34	4	44	D	54	T	64	d	74	t
05	ENQ	15	NAK	25	%	35	5	45	E	55	U	65	e	75	u
06	ACK	16	SYN	26	&	36	6	46	F	56	V	66	f	76	v
07	BEL	17	ETB	27	'	37	7	47	G	57	W	67	g	77	w
08	BS	18	CAN	28	(	38	8	48	H	58	X	68	h	78	x
09	HT	19	EM	29	)	39	9	49	I	59	Y	69	i	79	y
0A	LF	1A	SUB	2A	*	3A	:	4A	J	5A	Z	6A	j	7A	z
0B	VT	1B	ESC	2B	+	3B	;	4B	K	5B	[	6B	k	7B	{
0C	FF	1C	FS	2C	´	3C	<	4C	L	5C	\	6C	l	7C	|
0D	CR	1D	GS	2D	-	3D	=	4D	M	5D	]	6D	m	7D	}
0E	SO	1E	RS	2E	.	3E	>	4E	N	5E	^	6E	n	7E	~
0F	SI	1F	US	2F	/	3F	?	4F	O	5F	_	6F	o	7F	DEL

Exercícios:

Exercício 4:

Escreva na memória do programa uma palavra caractere usando a diretiva DB.

Verifique o valor que foi salvo na memória ao executar o programa.

Diretivas Assembly

ORG xxxxH : inicia “assemblagem” em xxxxH

Exemplo:

```
ORG 8000h
```

EQU : define uma constante

Exemplo:

```
minhaConst EQU 25
```

DB : define byte ou dado

Exemplo:

```
DB 28
DB “estudo na fei”
```

END : fim do arquivo assembly

Exercícios:

Exercício 5:

Atribua um valor numérico a um símbolo usando a diretiva EQU.

Exemplo:

```
DADO EQU 20h      ; Cria uma constante que vale 10
MOV DADO, #10h
```

Verifique o valor que foi salvo na memória de dados ao executar o programa.

Exercícios:

Exercício 6:

Atribua um valor numérico a um símbolo usando a diretiva EQU.

Exemplo:

```
DADO EQU 20h      ; Cria uma constante que vale 10
MOV DADO, #10h
MOV A, DADO
```

Verifique o valor que foi salvo no Acumulador ao executar o programa.

Exercícios:

Exercício 7:

Atribua um valor numérico a um símbolo usando a diretiva EQU.

Exemplo:

```
DADO EQU 20h      ; Cria uma constante que vale 10
MOV DADO, #10h
MOV A, #DADO
```

Verifique o valor que foi salvo no Acumulador ao executar o programa.

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 - MOVC

Operação: MOVC

Função: Transferências de Dados com a Memória de Programa.

Sintaxe: MOVC A,@A+*registrador*

Descrição : MOVC moves um byte da memória de programa para o acmuluador. O endereço da memória de programa do qual o byte será movido é calculado através da soma do valor do acumulador e o valor de DPTR ou PC. No caso de ser o PC, seu valor é incrementado de 1 antes de o mesmo ser adicionado ao acumulador.

Exemplo:

- movc A, @A+DPTR
- movc A, @A+PC

Instrução - MOVC

		bytes	MC	Op
MOVC A,	@A+DPTR	1	2	93
MOVC A,	@A+PC	1	2	83

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

Instrução - MOVX

Operação: MOVX

Função: Transferências de Dados com a Memória de Dados Externa.

Sintaxe: MOVX *operando1*,*operando2*

Descrição : move um byte da RAM externa para o acumulador ou vice-versa. Se o *operando1* é @DPTR, o conteúdo do acumulador é movido para o endereço de memória RAM externa indicado em DPTR. Esta instrução utiliza a Porta0 e a Porta2 para saída do endereço de 16-bits e do dado. Se o *operando2* é DPTR, então o byte é movido da RAM externa para o acumulador. Se *operando1* for @R0 ou @R1, o acumulador é movido para o endereço de memórica de 8-bits especificado no registrador. Esta instrução usa apenas a Porta0. Se o *operando2* for @R0 ou @R1 então o byte é movido da memória externa para o acumulador.

Exemplo:

- movx A, @R0
- movx A, @DPTR
- movx @DPTR, A

Instrução - MOVX

		bytes	MC	Op1
MOVX A,	@Ri	1	2	E2+i
MOVX A,	@DPTR	1	2	E0

Instruções para ler a memória de dados externa.

		bytes	MC	Op1
MOVX @Ri,	A	1	2	F2+i
MOVX @DPTR,	A	1	2	F0

Instruções para escrever na memória de dados externa.

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

Instrução - XCH

Operação: XCH

Função: Permutação de Bytes

Sintaxe: XCH A, *operando*

Descrição : Troca o valor do acumulador pelo valor contido em um operando.

Exemplo:

- XCH A, @R0
- XCH A, 45h

Instrução - XCHD

Operação: XCHD

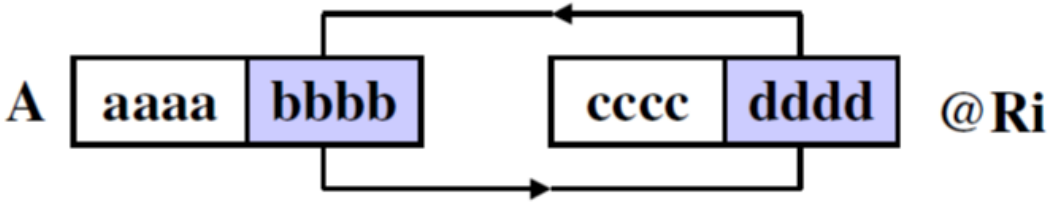
Função: Permutação de Nibbles.

Sintaxe: XCHD A,*[@R0/@R1]*

Descrição : Troca os bits 0-3 do acumulador com os bits 0-3 de um endereço da RAM interna apontado indiretamente por R0 ou R1. Os demais bits não são afetados.

Exemplo:

- XCHD A, @R1



Instrução – XCH e XCHD

		bytes	MC	Op1	Op2
XCH A,	Rn	1	1	C8+n	-
	end8	2	1	C5	end8
	@Ri	1	1	C6+i	-

Instruções para permutação de bytes.

		bytes	MC	Op1
XCHD	A,@Ri	1	1	D6+i

Instrução para a permutação de nibbles.

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

Exercícios

Exercícios:

Exercício 8:

Escreva na memória do programa um número inteiro usando a diretiva DB.

Leia o valor que você escreveu na memória do programa usando a instrução movc, e carregue o conteúdo para o registrador R1.

Exemplo:

```
mov DPTR, #0050h
clr A
movc A, @A+DPTR
```

Resposta:

Exercício 8:

```
org 0050h
DB 10

org 0060h
MOV DPTR, #0050h
CLR A
MOVC A, @A+DPTR
MOV R1, A
```

Exercícios:

Exercício 9:

Escreva na memória do programa um número inteiro negativo usando a diretiva DB no endereço 60h.

Leia o valor que você escreveu na memória do programa usando a instrução movc, faça a soma com o valor 0Fh e carregue o conteúdo para o registrador R1.

Verifique o valor que foi salvo na memória ao executar o programa.

Resposta:

Exercício 9:

```
org 0060h
DB -2

org 0080h
MOV DPTR, #0060h
CLR A
MOVC A, @A+DPTR
ADD A, #0Fh
MOV R1, A
```

Exercícios.

Exercício 10:

Se DPTR=1000h e no endereço 1000h temos o conteúdo #0EFh e A = #01h, qual o conteúdo de A após as instruções abaixo:

```
mov    A,   #2h
movx   A,  @DPTR
Add    A,  #01h
```

Exercícios.

Exercício 11:

Se DPTR=1000h, no endereço 1000h temos o conteúdo #0EFh, no endereço 1001h temos o conteúdo #33h e A = #01h, qual o conteúdo de A após a instrução:

```
movc   A,  @A + DPTR
```

Exercícios:

Exercício 12:

Atribua o valor numérico 20h a um símbolo usando a diretiva EQU; atribua um valor na memória de dados usando o símbolo; atribua o valor #0FFh ao Acumulador; troque os conteúdos entre o símbolo e o Acumulador usando XCH.

Bibliografia

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

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