Capítulo 2

NÚMEROS, CÓDIGOS DE COMPUTADOR E ARITMÉTICA

2.1 NÚMEROS BINÁRIOS

Computadores digitais usam *números binários*. O sistema de numeração binário ou de *base 2* usa somente os dígitos 0 e 1. Esses dígitos binários são chamados *bits*. Nos circuitos eletrônicos do computador, um bit 0 é representado por uma tensão BAIXA, enquanto um bit 1 corresponde a uma tensão ALTA.

Os seres humanos são treinados para entender o *sistema de numeração decimal*. O sistema decimal ou de *base 10* tem 10 dígitos (0-9). O sistema de numeração decimal também possui uma característica de *valor de casa*. Por exemplo, a Fig. 2-1*a* mostra que o número decimal 1327 é igual a um 1000 mais três 100s mais dois 10s mais sete 1s (1000 + 300 + 20 + 7 = 1327).

O sistema de numeração binário também possui uma característica de valor de casa. O valor decimal para as primeiras quatro casas binárias é mostrado na Fig. 2-1*b*. O número binário 1001 (um zero, zero, um) é convertido para seu equivalente decimal 9. O bit dos 1s do número binário da Fig. 2-1*b* é chamado o *bit menos significativo* (LSB)*, enquanto o bit dos 8 é chamado o *bit mais significativo* (MSB)**.

Os equivalentes binários para os números decimais de 0 a 15 são mostrados na Fig. 2-1*c*. Pessoas que trabalham com computadores devem memorizar pelo menos estes números binários.

O procedimento para converter o número binário 10110110 (um, zero, um, um, zero, um, um, zero) para seu equivalente decimal é mostrado na Fig. 2-2*a*. Para cada bit 1 no número binário, o valor da casa decimal é escrito abaixo. Os valores decimais são somados a seguir (128 + 32 + 16 + 4 + 2 = 182), obtendo 182.

Os *índices* da Fig. 2-2*b* são usados para indicar a base do número. O número 10110110_2 é, pois, um número binário ou de base 2. O número 182_{10} é um número decimal ou de base 10.

* Do inglês *L*east *S*ignificant *B*it.

** Do inglês *M*ost *S*ignificant *B*it.

Potências de 10	10^3	10^2	10^1	10^0		
Valor da casa	1000s	100s	10s	1s		
Decimal	1	3	2	7		
Decimal	1000 +	300 +	20 +	7	=	1327

(*a*) Valores das casas em um número decimal

Potências de 2	2^3	2^2	2^1	2^0		
Valor da casa	8s	4s	2s	1s		
	MSB			LSB		
Binário	1	0	0	1		
Decimal	8 +	0 +	0 +	1	=	9

(*b*) Valores das casas em um número binário

Decimal		Binário				Decimal		Binário			
10s	1s	8s	4s	2s	1s	10s	1s	8s	4s	2s	1s
	0				0		8	1	0	0	0
	1				1		9	1	0	0	1
	2			1	0	1	0	1	0	1	0
	3			1	1	1	1	1	0	1	1
	4		1	0	0	1	2	1	1	0	0
	5		1	0	1	1	3	1	1	0	1
	6		1	1	0	1	4	1	1	1	0
	7		1	1	1	1	5	1	1	1	1

(*c*) Equivalentes decimais e binários

Fig. 2-1

A Fig. 2-3 mostra o processo para converter o número decimal 155 para binário. O número 155 é primeiramente dividido por 2 deixando um quociente de 77 e um resto de 1. Este resto torna-se o bit menos significativo do número binário e é transferido para esta posição na Fig. 2-3. O quociente (77) é, então, transferido conforme mostra a seta e torna-se o próximo dividendo. Os quocientes são repetidamente divididos por 2, até que o quociente torna-se 0 com um resto de 1. Isto é mostrado na penúltima linha da Fig. 2-3. A última linha mostra que o resultado da conversão é $155_{10} = 10011011_2$.

Potências de 2	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0		
Valor da casa	128s	64s	32s	16s	8s	4s	2s	1s		
Binário	**1**	**0**	**1**	**1**	**0**	**1**	**1**	**0**		
Decimal	128	+	32 +	16	+	4 +	2		=	182

a) Conversão de binário para decimal

$10110110_2 = 182_{10}$

b) Os índices indicam a base do número

Fig. 2-2

$155_{10} \div 2 = 77$	resto 1	LSB
$77 \div 2 = 38$	resto 1	
$38 \div 2 = 19$	resto 0	
$19 \div 2 = 9$	resto 1	
$9 \div 2 = 4$	resto 1	
$4 \div 2 = 2$	resto 0	
$2 \div 2 = 1$	resto 0	
$1 \div 2 = 0$	resto 1	MSB

$155_{10} = 1\ 0\ 0\ 1\ 1\ 0\ 1\ 1_2$

Fig. 2-3 Conversão de decimal para binário

Problemas Resolvidos

2.1 A maioria dos seres humanos entende o sistema decimal, enquanto os computadores digitais usam o sistema de base 2 ou de numeração ______________.

Solução:

Computadores digitais usam o sistema de base 2 ou de numeração binária.

2.2 Tratando com números binários o termo bit significa ______________.

Solução:

Bit significa dígito binário.

2.3 Escreva o número um, um, zero, zero, em base dois, usando o índice.

Solução.

1100_2

2.4 Tratando com números binários, as letras MSB significam ______________.

Solução:

As letras MSB significam bit mais significativo quando se trata de números binários.

2.5 De memória, converta os seguintes números binários em seus equivalentes decimais:
(*a*) 0001 (*b*) 0101 (*c*) 1000 (*d*) 1011 (*e*) 1111 (*f*) 0111

Solução:

Observe a Fig. 2-1*c*. Estes números binários devem ser memorizados. Os equivalentes decimais dos números binários são os seguintes:

(*a*) $0001_2 = 1_{10}$ (*c*) $1000_2 = 8_{10}$ (*e*) $1111_2 = 15_{10}$
(*b*) $0101_2 = 5_{10}$ (*d*) $1011_2 = 11_{10}$ (*f*) $0111_2 = 7_{10}$

2.6 Converta os seguintes números decimais em seus equivalentes binários:
(*a*) 23 (*b*) 39 (*c*) 55 (*d*) 48

Solução:

Siga o procedimento mostrado na Fig. 2-3. Os equivalentes binários dos números decimais são os seguintes:

(*a*)
$$23_{10} \div 2 = 11 \text{ resto } 1 \text{ (LSB)}$$
$$11 \div 2 = 5 \text{ resto } 1$$
$$5 \div 2 = 2 \text{ resto } 1$$
$$2 \div 2 = 1 \text{ resto } 0$$
$$1 \div 2 = 0 \text{ resto } 1 \text{ (MSB)}$$
$$23_{10} = 10111_2$$

(*b*)
$$39_{10} \div 2 = 19 \text{ resto } 1 \text{ (LSB)}$$
$$19 \div 2 = 9 \text{ resto } 1$$
$$9 \div 2 = 4 \text{ resto } 1$$
$$4 \div 2 = 2 \text{ resto } 0$$
$$2 \div 2 = 1 \text{ resto } 0$$
$$1 \div 2 = 0 \text{ resto } 1 \text{ (MSB)}$$
$$39_{10} = 100111_2$$

(*c*)
$$55_{10} \div 2 = 27 \text{ resto } 1 \text{ (LSB)}$$
$$27 \div 2 = 13 \text{ resto } 1$$
$$13 \div 2 = 6 \text{ resto } 1$$
$$6 \div 2 = 3 \text{ resto } 0$$
$$3 \div 2 = 1 \text{ resto } 1$$
$$1 \div 2 = 0 \text{ resto } 1 \text{ (MSB)}$$
$$55_{10} = 110111_2$$

(*d*)
$$48_{10} \div 2 = 24 \text{ resto } 0 \text{ (LSB)}$$
$$24 \div 2 = 12 \text{ resto } 0$$
$$12 \div 2 = 6 \text{ resto } 0$$
$$6 \div 2 = 3 \text{ resto } 0$$
$$3 \div 2 = 1 \text{ resto } 1$$
$$1 \div 2 = 0 \text{ resto } 1 \text{ (MSB)}$$
$$48_{10} = 110000_2$$

2.7 $204_{10} = \_\_\_\_\_\__2$

Solução:

Siga o procedimento mostrado na Fig. 2-3. $204_{10} = 11001100_2$.

2.8 $11101110_2 = \_\_\_\_\_\_\_\__{10}$

Solução:

Siga o procedimento mostrado na Fig. 2-1*b*. $11101110_2 = 238_{10}$.

2.2 NÚMEROS HEXADECIMAIS

Uma posição de memória típica de um microcomputador poderia guardar o número binário 10011110. Esta longa fila de 0s e 1s é difícil de lembrar e de registrar através de um teclado. O número 10011110_2 poderia ser convertido para decimal. Após a conversão, descobriríamos que 10011110_2 é igual a 158_{10}. Este processo de conversão é demasiado longo. A maioria dos sistemas de microcomputador usa a notação *hexadecimal* para simplificar a lembrança e o registro de números binários como 10011110.

O sistema de numeração hexadecimal ou de base 16 usa os 16 símbolos de 0 a 9, A, B, C, D, E e F. Os equivalentes decimais, hexadecimais e binários são mostrados na Fig. 2-4.

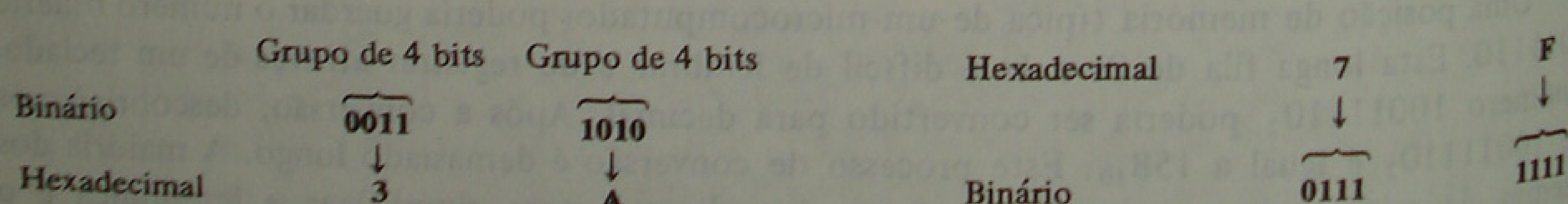
Note, na Fig. 2-4, que cada símbolo hexadecimal pode representar uma única combinação de 4 bits. O número binário 10011110 poderia, portanto, ser representado em hexadecimal por 9E.

Decimal	Hexadecimal	Binário			
		8s	4s	2s	1s
0	0	0	0	0	0
1	1	0	0	0	1
2	2	0	0	1	0
3	3	0	0	1	1
4	4	0	1	0	0
5	5	0	1	0	1
6	6	0	1	1	0
7	7	0	1	1	1
8	8	1	0	0	0
9	9	1	0	0	1
10	A	1	0	1	0
11	B	1	0	1	1
12	C	1	1	0	0
13	D	1	1	0	1
14	E	1	1	1	0
15	F	1	1	1	1

Fig. 2-4 Contando nos sistemas de numeração decimal, hexadecimal e binário

Isto é; a parte 1001 do número binário é igual a 9, de acordo com a Fig. 2-4 e a parte 1110 do número binário é igual a E em hexadecimal. 10011110_2, portanto, é igual a $9E_{16}$. Lembre-se de que os índices dão a base do número.

Para converter o número binário 111010 para hexadecimal (hex), começamos do LSB e dividimos o número binário em grupos de 4 bits, conforme mostra a Fig. 2-5*a*. Substituímos cada grupo de 4 bits pelo seu dígito hex equivalente. 1010_2 é igual a A em hex (ver Fig. 2-4). 0011_2 é igual a 3 em hex. 111010_2 é, pois, igual a $3A_{16}$.



(*a*) Conversão de binário para hexadecimal (*b*) Conversão de hexadecimal para binário

Fig. 2-5

Para converter o número hexadecimal 7F em seu equivalente binário, observamos na Fig. 2-5 que cada dígito hex é substituído por seu equivalente binário de 4 bits. Neste exemplo, 0111 substitui o hex 7 e 1111_2 substitui F_{16}. Conseqüentemente, $7F_{16}$ é igual a 01111111_2.

A notação hexadecimal é muito usada para representar números binários. As pessoas que usam a notação hexadecimal devem memorizar a tabela mostrada na Fig. 2-4.

O processo de conversão do número hexadecimal 2C6E em seu equivalente decimal é mostrado na Fig. 2-6*a*. Os valores das casas para os quatro primeiros dígitos hexadecimais são mostrados como sendo 4096, 256, 16 e 1. O número hexadecimal contém quatorze (E_{16}) 1s, seis 16s, doze (C_{16}) 256s e dois 4096s. O valor de cada uma das casas é multiplicado e os produtos são somados dando como resultado 11.374_{10}.

O processo de conversão do número decimal 15.797 em seu equivalente hexadecimal é mostrado na Fig. 2-6*b*. A primeira linha mostra 15.797_{10} sendo dividido por 16, dando um quociente de 987_{10} e um resto de 5. O resto é convertido para seu equivalente hexadecimal. 5_{10} é igual a 5_{16}. O resto hex (5_{16}) torna-se o dígito menos significativo (LSD) do número hexadecimal. O primeiro quociente (987) torna-se o dividendo da segunda linha e é dividido por 16. O segundo quociente é 61 com um resto de 11_{10} ou B em hexadecimal. A linha 3 mostra 61 dividido por 16 resultando em um quociente de 3 com resto de 13_{10} ou D_{16}. A quarta linha da Fig. 2-6*b* mostra o dividendo (3) sendo dividido por 16, dando um quociente 0 e um resto 3_{10} ou 3_{16}. Quando o quociente torna-se 0, como na linha 4, o cálculo está completo. 3_{16} é o dígito mais significativo (MSD). O procedimento mostrado na Fig. 2-6*b* converte o número decimal 15.797 em seu equivalente hexadecimal $3DB5_{16}$.

Potências de 16	16^3	16^2	16^1	16^0
Valor da casa	4096s	256s	16s	1s

Hexadecimal	**2**		**C**		**6**		**E**		
	↓		↓		↓		↓		
	4096		256		16		1		
	×2		×12		×6		×14		
Decimal	8192	+	3072	+	96	+	14	=	$11,374_{10}$

(*a*) Conversão de hexadecimal para decimal

$15,797_{10} \div 16 = 987$ resto $5_{10} = 5_{16}$ (LSD)

$987 \div 16 = 61$ resto $11_{10} = B_{16}$

$61 \div 16 = 3$ resto $13_{10} = D_{16}$

$3 \div 16 = 0$ resto $3_{10} = 3_{16}$ (MSD)

$15,797_{10} =$ 3 D B 5_{16}

(*b*) Conversão de decimal para hexadecimal

Fig. 2-6

Problemas Resolvidos

2.9 A notação hexadecimal é muito usada no trabalho com microcomputadores como um método "taquigráfico" de representação de números ________________ (binários, decimais).

Solução:

A notação hexadecimal é muito usada para representar números binários.

2.10 Converta os seguintes números hexadecimais em seus equivalentes binários:

(*a*) C	(*c*) F	(*e*) 1A	(*g*) A0	(*i*) 45
(*b*) 6	(*d*) E2	(*f*) 3D	(*h*) 8B	(*j*) D7

Solução:

Usando a tabela da Fig. 2-4 siga o procedimento mostrado na Fig. 2-5*b*. Os equivalentes binários dos números hexadecimais são os seguintes:

(*a*) $C_{16} = 1100_2$	(*f*) $3D_{16} = 00111101_2$
(*b*) $6_{16} = 0110_2$	(*g*) $A0_{16} = 10100000_2$
(*c*) $F_{16} = 1111_2$	(*h*) $8B_{16} = 10001011_2$
(*d*) $E2_{16} = 11100010_2$	(*i*) $45_{16} = 01000101_2$
(*e*) $1A_{16} = 00011010_2$	(*j*) $D7_{16} = 11010111_2$

2.11 Converta os seguintes números binários em seus equivalentes decimais:

(*a*) 1001	(*c*) 1101	(*e*) 10000000	(*g*) 00010101
(*b*) 1100	(*d*) 1111	(*f*) 01111110	(*h*) 11011011

Solução:

Usando a tabela da Fig. 2-4 siga o procedimento mostrado na Fig. 2-5*a*. Os equivalentes hexadecimais dos números binários são os seguintes:

(*a*) $1001_2 = 9_{16}$	(*e*) $10000000_2 = 80_{16}$
(*b*) $1100_2 = C_{16}$	(*f*) $01111110_2 = 7E_{16}$
(*c*) $1101_2 = D_{16}$	(*g*) $00010101_2 = 15_{16}$
(*d*) $1111_2 = F_{16}$	(*h*) $11011011_2 = DB_{16}$

2.12 $217_{10} = \underline{\qquad\qquad}_{16}$

Solução:

Usando a tabela da Fig. 2-4 siga o procedimento mostrado na Fig. 2-6*b*.

$$217_{10} \div 16 = 13 \text{ resto } 9_{10} = 9_{16} \text{ (LSD)}$$
$$13 \div 16 = 0 \text{ resto } 13_{10} = D_{16} \text{ (MSD)}$$
$$217_{10} = D9_{16}$$

2.13 $48.373_{10} = \underline{\qquad\qquad\qquad}_{16}$

Solução:

$$\begin{aligned}
48.373_{10} \div 16 &= 3023 \text{ resto } 5_{10} = 5_{16} \text{ (LSD)}\\
3023 \div 16 &= 188 \text{ resto } 15_{10} = F_{16}\\
188 \div 16 &= 11 \text{ resto } 12_{10} = C_{16}\\
11 \div 16 &= 0 \text{ resto } 11_{10} = B_{16} \text{ (MSD)}\\
48.373_{10} &= BCF5_{16}
\end{aligned}$$

2.3 NÚMEROS OCTAIS

A *notação octal*, como a hexadecimal, é usada para representar números binários. Octal usa os símbolos de 0 a 7 e é, portanto, chamado de sistema de numeração de *base 8*. A tabela da Fig. 2-7 mostra os equivalentes decimais, octais e binários.

O processo de conversão do número binário 11111000100 em seu equivalente octal é mostrado na Fig. 2-8*a*. Começando com o LSB do número binário divide-se o número em grupos de 3 bits. A seguir, usando a Fig. 2-7, converte-se cada grupo de 3 bits em seu dígito octal equivalente. Desta maneira, o número binário 11111000100 da Fig. 2-8*a* foi convertido em seu equivalente octal 3704_8.

Decimal	Octal	Binário 8s	2s	1s
0	0	0	0	0
1	1	0	0	1
2	2	0	1	0
3	3	0	1	1
4	4	1	0	0
5	5	1	0	1
6	6	1	1	0
7	7	1	1	1

Fig. 2-7 Contando nos sistemas de numeração decimal, octal e binário

Grupos de 3 bits

$$\begin{array}{lcccc}
\text{Binário} & \overbrace{011} & \overbrace{111} & \overbrace{000} & \overbrace{100}\\
 & \downarrow & \downarrow & \downarrow & \downarrow\\
\text{Octal} & 3 & 7 & 0 & 4
\end{array}$$

(*a*) Conversão de binário para octal

$$\begin{array}{lcccc}
\text{Octal} & 6 & 5 & 2 & 1\\
 & \downarrow & \downarrow & \downarrow & \downarrow\\
\text{Binário} & \overbrace{110} & \overbrace{101} & \overbrace{010} & \overbrace{001}
\end{array}$$

(*b*) Conversão de octal para binário

Fig. 2-8

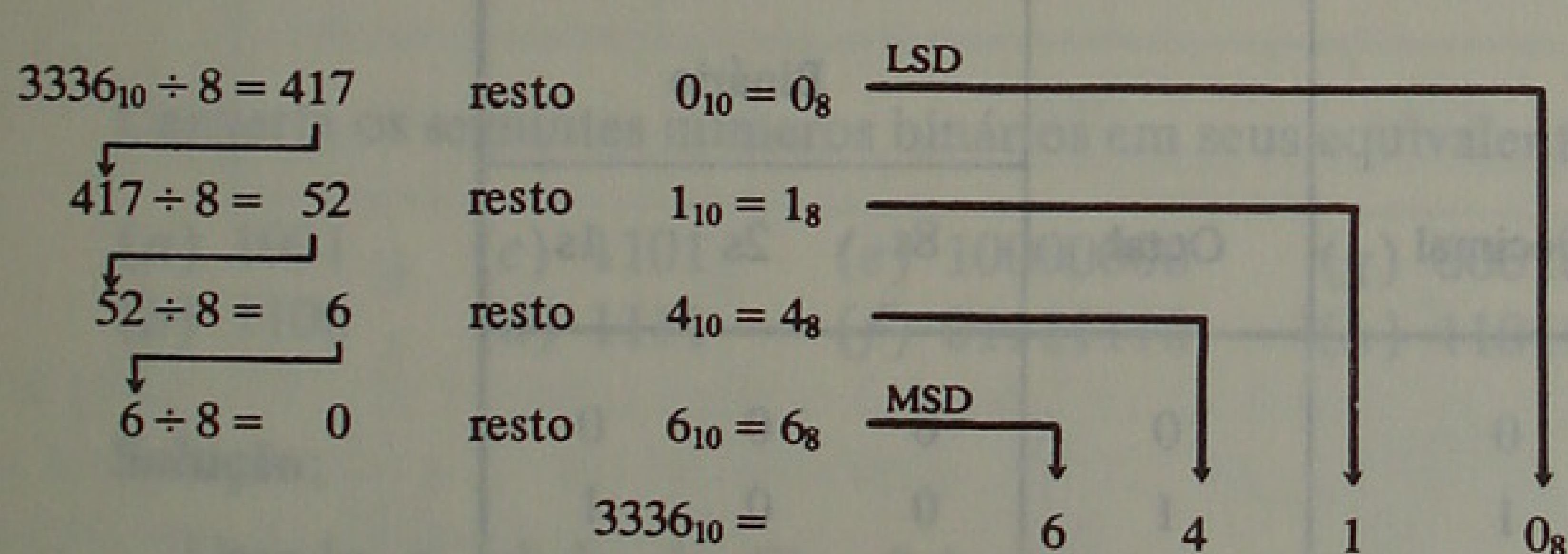
O procedimento para converter o número octal 6521 em seu equivalente binário é mostrado na Fig. 2-8*b*. Cada dígito octal é traduzido para seu grupo binário de 3 bits. A Fig. 2-8*b* mostra que 6521_8 é igual a 110101010001_2.

O procedimento já familiar para converter o número octal 2357 em seu equivalente decimal é detalhado na Fig. 2-9*a*. Os valores das quatro primeiras casas em um número octal são mostrados como 512, 64, 8 e 1. Note-se que neste exemplo há sete 1s, cinco 8s, três 64s e dois 512s. A seguir os produtos são somados (1024 + 192 + 40 + 7), resultando 1263_{10}. O exemplo da Fig. 1-9*a* mostra que 2357_8 é igual a 1263_{10}.

Potências de 8	8^3	8^2	8^1	8^0
Valor da casa	512s	64s	8s	1s

Octal	2		3		5		7		
	↓		↓		↓		↓		
	512		64		8		1		
	×2		×3		×5		×7		
Decimal	1024	+	192	+	40	+	7	=	1263_{10}

(*a*) Conversão de octal para decimal



(*b*) Conversão de decimal para octal

Fig. 2-9

O procedimento para converter o número decimal 3336 em seu equivalente octal é mostrado na Fig. 2-9*b*. A primeira linha mostra o número decimal 3336 sendo dividido por 8. O quociente é 417 e o resto é 0_{10}. O decimal é igual a 0_8. O octal 0 torna-se o LSD do número octal. O primeiro quociente (417) é transferido para a posição de dividendo na linha 2 e dividido por 8. O quociente é 52 com um resto de 1_{10} ou 1_8. 1_8 torna-se o segundo dígito do número octal. O segundo quociente (52) torna-se o dividendo na linha 3 e é dividido por 8. O quociente é 6 com um resto de 4_{10} ou 4_8˜4_8 é o próximo dígito octal. O terceiro quociente (6) torna-se o dividendo na linha 4 e é dividido por 8. O quociente é 0 com resto 6_{10} ou 6_8. O cálculo agora está completo, pois o último quociente (o quarto quociente) é 0. 6_8 torna-se o MSD do número octal. O procedimento da Fig. 2-9*b* mostra que 3336_{10} é igual a 6410_8.

A maioria dos microprocessadores e microcomputadores processa grupos de 4, 8 ou 16 bits. Por esta razão é típico o uso da notação hexadecimal em vez da octal. A notação octal é conveniente quando o grupamento de bits é divisível por 3, como nos grupos de 12 bits.

Problemas Resolvidos

2.14 A literatura de um microprocessador de 8 bits provavelmente usaria a notação ____________________ (hexadecimal, octal) para representar números binários.

Solução:

A literatura de um microprocessador de 8 bits provavelmente usaria a notação hexadecimal para representar números binários em virtude de sua eficiência.

2.15 O sistema de numeração octal é, às vezes, chamado de sistema ____________________.

Solução:

O sistema de numeração octal é, às vezes, chamado de sistema de base 8 por causa do uso de oito símbolos.

2.16 Converta os seguintes números binários em seus equivalentes octais:

(*a*) 101 (*c*) 010 (*e*) 1011000111
(*b*) 110 (*d*) 111000101010 (*f*) 100110100101

Solução:

Siga o procedimento mostrado na Fig. 2-8*a*. Os equivalentes octais dos números binários são os seguintes:

(*a*) $101_2 = 5_8$ (*c*) $010_2 = 2_8$ (*e*) $1011000111_2 = 1307_8$
(*b*) $110_2 = 6_8$ (*d*) $111000101010_2 = 7052_8$ (*f*) $100110100101_2 = 4645_8$

2.17 $6724_8 =$ ______________ $_{10}$

Solução:

Siga o procedimento mostrado na Fig. 2-9*a*.

$$6724_8 = (512 \times 6) + (64 \times 7) + (8 \times 2) + (1 \times 4) = 3540_{10}$$

2.18 $2648_{10} =$ ______________ $_8$

Solução:

Siga o procedimento mostrado na Fig. 2-9*b*.

$$\begin{aligned} 2648_{10} \div 8 &= 331 \text{ resto } 0 \text{ (LSD)} \\ 331 \div 8 &= 41 \text{ resto } 3 \\ 41 \div 8 &= 5 \text{ resto } 1 \\ 5 \div 8 &= 0 \text{ resto } 5 \text{ (MSD)} \end{aligned}$$

$$2648_{10} = 5130_8$$

2.4 NÚMEROS BCD

Números binários puros são representados em notação octal ou hexadecimal em virtude da facilidade de conversão. A conversão de binário para decimal, todavia, é bastante difícil. Em calculadoras, jogos e instrumentos digitais, onde há freqüentes entradas e saídas em decimal, um código especial é usado para representar esses números decimais. Este código é chamado *código BCD* (*B*inary-*C*oded-*D*ecimal ou *decimal codificado em binário*). Os equivalentes decimais e BCD são mostrados na tabela da Fig. 2-10*a*. Tecnicamente, esta tabela detalha o *código BCD 8421*. A qualificação 8421 dá os valores das casas para os quatro bits do código BCD. Outros códigos BCD também são usados, como o código BCD 5421 e o código excesso-3.

	BCD			
Decimal	8s	4s	2s	1s
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

(*a*) O código BCD 8421

Decimal	3	6	9	1
	↓	↓	↓	↓
BCD	0011	0110	1001	0001

(*b*) Conversão de decimal para BCD

BCD	1000	0000	0111	0010
	↓	↓	↓	↓
Decimal	8	0	7	2

(*c*) Conversão de BCD para decimal

Fig. 2-10

O procedimento para converter o número decimal 3691 em seu equivalente BCD 8421 é mostrado na Fig. 2-10*b*. Cada dígito decimal é traduzido diretamente para seu equivalente BCD de 4 bits. O exemplo mostra que 3691_{10} é igual a $0011\ 0110\ 1001\ 0001_{BCD}$.

O procedimento para converter o número BCD $1000\ 0000\ 0111\ 0010_{BCD}$ em seu equivalente decimal é detalhado na Fig. 2-10*c*. Cada grupo de 4 bits é traduzido diretamente para seu equivalente decimal. Este exemplo mostra que $1000\ 0000\ 0111\ 0010_{BCD}$ é igual a 8072_{10}.

Microprocessadores somam números binários puros. Muitos microprocessadores, todavia, possuem instruções especiais para transformar os resultados das somas para notação BCD. O número BCD é facilmente interpretado como um número decimal usando os procedimentos simples mostrados nas Figs. 2-10*b* e *c*.

Problemas Resolvidos

2.19 As letras BCD representam ____________________

Solução:

BCD representa decimal codificado em binário.

2.20 A notação BCD mais comum é o código __________ (5421, 8421).

Solução:

A notação BCD mais comum é o código 8421.

2.21 Converta os seguintes números BCD 8421 em seus equivalentes decimais:

(*a*) 1000 0000 (*c*) 1001 0010 (*e*) 0100 0011
(*b*) 0000 0001 (*d*) 0111 0110 (*f*) 0101 0101

Solução:

Siga o procedimento mostrado na Fig. 2-10*c*. Os equivalentes decimais dos números BCD são os seguintes:

(*a*) $1000\,0000_{BCD} = 80_{10}$ (*c*) $1001\,0010_{BCD} = 92_{10}$ (*e*) $0100\,0011_{BCD} = 43_{10}$
(*b*) $0000\,0001_{BCD} = 1_{10}$ (*d*) $0111\,0110_{BCD} = 76_{10}$ (*f*) $0101\,0101_{BCD} = 55_{10}$

2.5 ARITMÉTICA BINÁRIA

A adição, a subtração e a multiplicação de números binários são efetuadas de modo similar ao da aritmética decimal. A maioria dos microprocessadores possui instruções para somar e subtrair números binários. Uns poucos microprocessadores possuem até instruções para multiplicar e dividir como, por exemplo, o 8086 e o 8088.

As regras para a adição binária são mostradas na Fig. 2-11*a*. As primeiras duas regras, à esquerda, são óbvias. A terceira regra mostra que, em binário, 1 + 1 = 10, com o 1 mais significativo indo para a casa mais alta seguinte. A quarta regra mostra que, em binário, 1 + 1 + 1 = 11. Aqui as duas parcelas e o vem-um são todos 1s. O resultado é uma soma de 1 com um vai-um de 1.

A soma dos números binários 00111011 e 00101010 é ilustrada na Fig. 2-11*b*. Confira seu entendimento deste processo. Para maior conveniência, os equivalentes decimais dos números binários são mostrados à direita. A soma de 00111011 e 00101010 é mostrada na Fig. 2-11*b* como sendo 01100101_2.

As regras para subtração binária são mostradas na Fig. 2-12*a*. As três primeiras regras são idênticas às da subtração decimal. A última regra requer um *empréstimo* da casa mais significativa seguinte (a casa dos 2s). O minuendo é, neste caso, 10 e o subtraendo é 1 com uma diferença de 1.

		^{1}	^{1} ^{1}	
1ª parcela	0	0	1	1
2ª parcela	+0	+1	+1	+1
Soma	0	1	10	11

Vem-um da casa menos significativa seguinte

Vai-um para a casa mais significativa seguinte

Vai-um

(*a*) Regras para soma binária

Vai-uns

	1 1 1 1	1
1ª parcela	00 1 1 10 11	59
2ª parcela	+ 00 1 0 10 10	+ 42
Soma	01 1 0 01 01$_2$	101$_{10}$

(*b*) Problema de soma binária

Fig. 2-11

				^{0} $\curvearrowright$ ^{10}
Minuendo	0	1	1	$\cancel{1}$ 0
Subtraendo	−0	−1	−0	−1
Diferença	0	0	1	1

(*a*) Regras para subtração binária

^{1} ^{10} / ^{0} $^{\cancel{10}}$ ^{0} ^{10}	^{7} ^{1}
0 $\cancel{1}$ 0 $\cancel{1}$ 0 1 0 1	$\cancel{8}$ 5
− 0 0 1 1 1 0 0 1	− 5 7
0 0 0 1 1 1 0 0$_2$	2 8$_{10}$

(*b*) Problema de subtração binária

Fig. 2-12

A subtração do número binário 00111001 de 01010101 é detalhada na Fig. 2-12*b*. As colunas dos 1s, 2s e 4s da subtração binária são simples de seguir usando as três primeiras regras da Fig. 2-12*a*. A coluna dos 8s mostra 1 sendo subtraído de 0. Um 1 é tomado emprestado da coluna dos 16s. A seguir 1 é subtraído de 10_2 dando uma diferença de 1 de acordo com a quarta regra da Fig. 2-12*a*. Após o empréstimo, a coluna dos 16s mostra1 sendo subtraído do novo minuendo que é 0. Baseado na regra nº 4, um 1 deve ser tomado emprestado da casa mais significativa seguinte (a casa dos 32s). O minuendo na casa dos 32s é 0. A casa dos 32s deve tomar emprestado da casa dos 64s onde existe um 1. Finalmente, a casa dos 16s pode tomar 1 emprestado da casa dos 32s. O minuendo na casa dos 16s passa a ser 10_2 e o subtraendo é 1 com uma diferença de 1. A casa dos 32s agora mostra 1 − 1 dando uma diferença de 0. A casa dos 64s mostra 0 − 0 resultando em 0. A casa dos 128s mostra 0 − 0 também resultando em 0. Resumindo, o problema ilustrativo da Fig. 2-12*b* mostra o número binário 00111001 sendo subtraído de 01010101_2, dando um resultado de 00011100_2. O problema é também mostrado em forma decimal à direita.

As regras para multiplicação binária são mostradas na Fig. 2-13*a*. As duas primeiras regras não necessitam explicação. Nas duas regras seguintes o multiplicador é 1. Na multiplicação binária, quando o multiplicador for 1, o *multiplicando* é *copiado* no produto. Quando o multiplicador for 0 o produto será sempre 0.

Multiplicando	0	1	0	1
Multiplicador	×0	×0	×1	×1
Produto	0	0	0	1

(*a*) Regras para multiplicação binária

Multiplicando	1101	13
Multiplicador	×101	×5
Primeiro produto parcial	1101	65_{10}
Segundo produto parcial	0000	
Terceiro produto parcial	1101	
Produto final	1000001_2	

(*b*) Problema de multiplicação binária

Fig. 2-13

O problema ilustrativo da multiplicação de 1101 por 101 é mostrado na Fig. 2-13*b*. Como na multiplicação decimal, o multiplicando é primeiro multiplicado pelo dígito menos significativo (neste caso, o bit dos 1s). O bit dos 1s do multiplicador é 1, logo o multiplicando é copiado como primeiro produto parcial. O bit dos 2s do multiplicador é 0, logo, o segundo produto parcial é 0000. Note que é *deslocada* uma casa para a esquerda. O bit dos 4s do multiplicador é 1, logo o multiplicando é copiado como o terceiro produto parcial. Note que o 1101 é copiado após um segundo deslocamento para a esquerda. O primeiro, segundo e terceiro produtos parciais são somados, resultando o produto final 1000001_2. Resumindo, a Fig. 2-13*b* mostra que $1101_2 \times 101_2 = 1000001_2$, ou seja, $13_{10} \times 5_{10} = 65_{10}$.

Problemas Resolvidos

2.22 Resolva os seguintes problemas de soma binária:

(*a*) $\begin{array}{r} 1010 \\ \underline{+0101} \end{array}$ (*b*) $\begin{array}{r} 1101 \\ \underline{+0101} \end{array}$ (*c*) $\begin{array}{r} 01011011 \\ \underline{+00001111} \end{array}$ (*d*) $\begin{array}{r} 00111111 \\ \underline{+00011111} \end{array}$

Solução:

Refira-se à Fig. 2-11. As somas binárias para os problemas são as seguintes: (*a*) 1111 (*b*) 10010 (*c*) 01101010 (*d*) 01011110

2.23 O número superior em um problema de multiplicação é chamado ____________, enquanto o número inferior é chamado multiplicador e o resultado é chamado ____________.

Solução:

O número superior em uma multiplicação é chamado multiplicando, enquanto o número inferior é chamado multiplicador e o resultado é chamado produto.

2.24 Resolva os seguintes problemas de multiplicação binária:

(*a*) $\begin{array}{r} 1001 \\ \underline{\times 11} \end{array}$ (*b*) $\begin{array}{r} 1101 \\ \underline{\times 1001} \end{array}$ (*c*) $\begin{array}{r} 1111 \\ \underline{\times 101} \end{array}$ (*d*) $\begin{array}{r} 1110 \\ \underline{\times 1110} \end{array}$

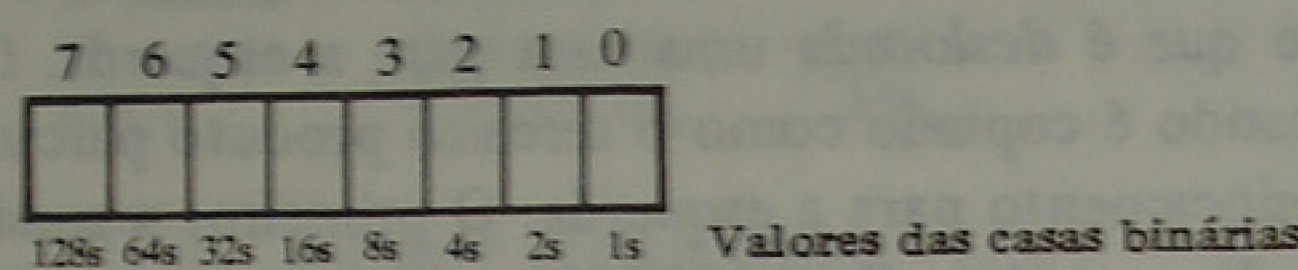
Solução:

Refira-se à Fig. 2-13. Os produtos binários para os problemas são os seguintes: (*a*) 11011 (*b*) 1110101 (*c*) 1001011 (*d*) 11000100

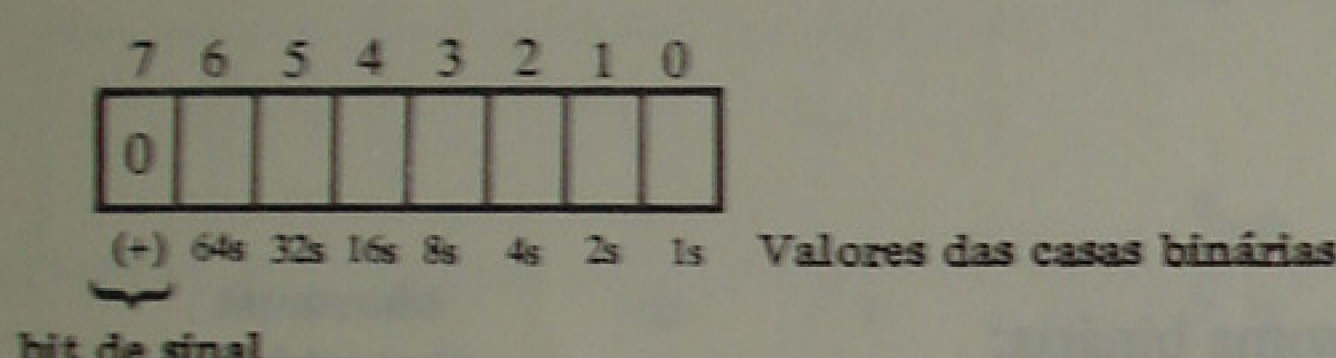
2.6 A NOTAÇÃO COMPLEMENTO 2

Números binários são geralmente usados nos computadores. Algumas vezes um código especial chamado *notação complemento 2* é usado quando são necessários números com sinal. O sistema simplifica os circuitos do computador.

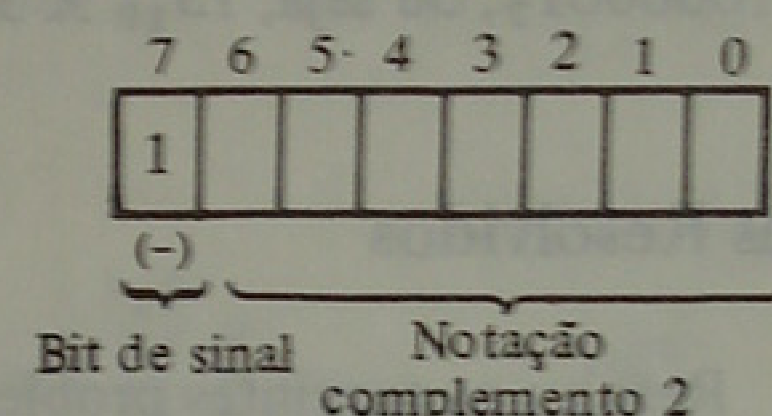
Um registrador ou posição de armazenamento típico em um microprocessador poderia parecer-se com o da Fig. 2-14*a*. Este registrador tem espaço para 8 bits de dados. As posições dos bits são numeradas de 7 a 0. Os valores das casas binárias são mostrados abaixo do registrador. O bit 7 seria a casa dos 128s, o bit 6 a casa dos 64s etc.



(*a*) Identificando as posições de armazenamento de um registrador de 8 bits



(*b*) Números positivos identificados por um 0 na posição do bit de sinal do registrador



(*c*) Números negativos identificados por um 0 na posição do bit de sinal do registrador

Fig. 2-14

A organização típica de um registrador de 8 bits usado para armazenar *números com sinal* é mostrada na Fig. 2-14*b* e *c*. O bit 7 em ambos os registradores é o *bit de sinal*. O bit de sinal informa se o número é (+) positivo ou (–) negativo. Um 0 na posição do bit de sinal significa que o número é positivo, enquanto um 1 na posição do bit 7 significa que o número é negativo.

Se o número for positivo, como na Fig. 2-14*b*, as posições de memória restantes (6–0) guardam um número binário de 7 bits. Por exemplo, se o registrador da Fig. 2-14*b* guardasse 01000001, isto significaria o decimal +65 (bit de sinal positivo +64 + 1). Se o registrador da Fig. 2-14*b* guardasse 01111111, seria igual a $+127_{10}$ (bit de sinal positivo +64 + 32 + 16 + 8 + 4 + 2 + 1). Este é o maior número positivo que pode ser representado neste registrador de 8 bits.

Se o número com sinal fosse *negativo*, como na Fig. 2-14*c*, o registrador manteria a *forma complemento 2* daquele número. A tabela da Fig. 2-15 mostra a notação complemento 2 para números positivos e negativos. Note que os números positivos têm um 0 como MSB, enquanto o resto do número corresponde a um número binário. Números negativos possuem um 1 como MSB. Observe a linha +0 na tabela da Fig. 2-15. A notação complemento 2 para +0 é 00000000. Na linha seguinte, note-se que 11111111 é a notação complemento 2 para –1. Pode-se pensar no sistema como sendo um adômetro que está sendo virado para trás enquanto se regride, na tabela, de 00000000 para 11111111.

Decimal	Representação de números com sinal	
+127	0111 1111	Números positivos representados da mesma maneira que em binário direto
⋮	⋮	
+8	0000 1000	
+7	0000 0111	
+6	0000 0110	
+5	0000 0101	
+4	0000 0100	
+3	0000 0011	
+2	0000 0010	
+1	0000 0001	
+0	0000 0000	
−1	1111 1111	Números negativos representados na forma complemento 2
−2	1111 1110	
−3	1111 1101	
−4	1111 1100	
−5	1111 1011	
−6	1111 1010	
−7	1111 1001	
−8	1111 1000	
⋮	⋮	
−128	1000 0000	

Fig. 2-15 Números decimais com sinal e seus equivalentes na notação complemento 2

Qual seria a notação complemento 2 para −9? Os passos para executar esta conversão são mostrados na Fig. 2-16*a*.

Passo 1 Listar o número decimal sem sinal. Neste exemplo seria 9.

Passo 2 Converter o decimal em número binário. Neste exemplo escreva o número binário 00001001.

Passo 3 Complemente cada bit formando o *complemento 1*. Neste exemplo escreva 11110110 que é o complemento 1.

Passo 4 Some 1 ao número complemento 1. Neste exemplo some 1 a 11110110.

O resultado é a notação complemento 2 do negativo do número decimal. No exemplo da Fig. 2-16*a*, −9 é igual a 11110111 na forma complemento 2. Note que o bit de sinal (11110111) é 1, o que significa que este número é negativo.

Decimal	9	Passo 1	Escreva o decimal
	↓	Passo 2	Converta para o binário
Binário	00001001		
	↓	Passo 3	Complemente cada bit
Complemento 1	11110110		
	+ 1	Passo 4	Some +1
Complemento 2	11110111 = −9		

(*a*) Formação do complemento 2 de um número negativo

Complemento 2	11110000	Passo 1	Escreva o Complemento 2
	↓	Passo 2	Complemente cada bit
Complemento 1	00001111		
	+ 1	Passo 3	Some +1
Binário	00010000 = 16		

(*b*) Cálculo do equivalente decimal de um número em complemento 2

Fig. 2-16

Qual é o valor do equivalente decimal do número em complemento 2 11110000? O procedimento para executar esta conversão é detalhado na Fig. 2-16*b*. O procedimento de complementação e adição de 1 é o mesmo usado para converter de binário para complemento 2. O procedimento da Fig. 2-16*b* mostra cada bit do número em complemento 2 sendo alterado formando o complemento 1. Um 1 é, então, somado ao número complemento 1 formando o número binário 00010000, que é igual a 16. Isto significa que a notação complemento 2 11110000 é igual a -16_{10}. O 16 deve ser negativo, pois o bit de sinal (MSB) do número complemento 2 é um 1.

Problemas Resolvidos

2.25 Quando números com sinal são armazenados em um registrador de um microprocessador de 8 bits o MSB (bit 7) é chamado bit de ____________________.

Solução:

Quando números com sinal são armazenados em um registrador de um microprocessador de 8 bits o MSB (bit 7) é chamado bit de sinal.

2.26 Usando a tabela da Fig. 2-15, encontre a notação complemento 2 para os seguintes números decimais com sinal:

(*a*) +1 (*b*) +5 (*c*) +127 (*d*) −1 (*e*) −2 (*f*) −128

Solução:

Refira-se à Fig. 2-15.

(*a*) +1 = 00000001 (notação complemento 2)
(*b*) +5 = 00000101 (notação complemento 2)
(*c*) +127 = 01111111 (notação complemento 2)
(*d*) −1 = 11111111 (notação complemento 2)
(*e*) −21 = 11111110 (notação complemento 2)
(*f*) −128 = 10000000 (notação complemento 2)

2.27 Usando o procedimento da Fig. 2-16*a*, converta os seguintes números decimais com sinal para sua forma complemento 2:

(*a*) −10 (*b*) −21 (*c*) −34 (*d*) −96

Solução:

Refira-se à Fig. 2-16*a*. As notações complemento 2 para os números decimais com sinal são as seguintes:

(*a*) Primeiro passo (converter de decimal para binário): $10_{10} = 00001010_2$
Segundo passo (complementar): $00001010_2 \rightarrow 11110101$
Terceiro passo (somar 1). 11110101 + 1 = 11110110 (complemento 2) = -10_{10}

(*b*) Primeiro passo (converter de decimal para binário): $21_{10} = 00010101_2$
Segundo passo (complementar): $00010101_2 \rightarrow 11101010$
Terceiro passo (somar 1): 11101010 + 1 = 11101011 (complemento 2) = -21_{10}

(*c*) Primeiro passo (converter de decimal para binário): $34_{10} = 00100010_2$
Segundo passo (complementar): $00100010_2 \rightarrow 11011101$
Terceiro passo (somar 1): 11011101 + 1 = 11011110 (complemento 2) = -34_{10}

(*d*) Primeiro passo (converter de decimal para binário): $96_{10} = 01100000_2$
Segundo passo (complementar): $01100000_2 \rightarrow 10011111$
Terceiro passo (somar 1): 10011111 + 1 = 10100000 (complemento 2) = -96_{10}

2.28 Usando o procedimento mostrado na Fig. 2-16*b*, converta as notações complemento 2 a seguir em seus equivalentes decimais com sinal:

(*a*) 11111011 (*b*) 00001111 (*c*) 10001111 (*d*) 01110111

Solução:

Os equivalentes decimais com sinais das notações complemento 2 são os seguintes:

(*a*) 11111011 (complemento 2) = -5_{10}

$$11111011 \xrightarrow{\text{complemento}} \begin{array}{r} 00000100 \\ +\quad\quad 1 \\ \hline 00000101 \end{array} = 5$$

(*b*) 00001111 (complemento 2) = +15

(*c*) 10001111 (complemento 2) = −113

$$10001111 \xrightarrow{\text{complemento}} \begin{array}{r} 01110000 \\ +\qquad\quad 1 \\ \hline 01110001 \end{array} = 113$$

(*d*) 01110111 (complemento 2) = +119

2.7 A ARITMÉTICA COMPLEMENTO 2

Um microprocessador pode usar números complemento 2, porque ele pode *complementar*, *incrementar* (somar +1 a um número) e *somar* números binários. Os microprocessadores não possuem circuitos de subtração. Em vez disso é usado um somador e números complemento 2 para executar a subtração.

Some os números decimais com sinal +5 e +3. O processo é mostrado na Fig. 2-17*a* tanto para números decimais como para complemento 2. Da tabela da Fig. 2-15 sabemos que +5 é igual a 00000101 em complemento 2, ao passo que +3 = 00000011. Os números complemento 2 00000101 e 00000011 são somados como se fossem números binários normais, resultando a soma complemento 2 de 00001000. A soma 00001000 é igual a $+8_{10}$.

Some os números decimais com sinal +7 e −3. O procedimento usando números decimais e complemento 2 é mostrado na Fig. 2-17*b*. Da tabela da Fig. 2-15 obtemos +7 = 00000111 e −3 = 11111101 em notação complemento 2. Os números complemento 2 00000111 e 11111101 são somados como se fossem números binários normais resultando na soma 100000100. O MSB é um vai-um de sobra do registrador de 8 bits e é descartado. Isto deixa a soma complemento 2 de 00000100 ou $+4_{10}$.

1ª parcela	(+5)	00000101
2ª parcela	+ (+3)	+ 00000011
Soma	(+8)	00001000

(*a*) Problema de soma complemento 2

1ª parcela	(+7)	00000111
2ª parcela	+ (−3)	+ 11111101
Soma	(+4)	1 00000100

Descartar a sobra

(*b*) Problema de soma complemento 2

1ª parcela	(+3)	00000011
2ª parcela	+ (−8)	+ 11111000
Soma	(−5)	11111011

(*c*) Problema de soma complemento 2

1ª parcela	(−2)	11111110
2ª parcela	+ (−5)	+ 11111011
Soma	(−7)	1 11111001

Descartar a sobra

(*d*) Problema de soma complemento 2

Fig. 2-17

Some os números decimais com sinal +3 e −8. O procedimento é detalhado na Fig. 2-17*c*. Da tabela da Fig. 2-15 sabe-se que +3 = 00000011 e −8 = 11111000 em notação complemento 2. Os números complemento 2 são somados como se fossem números binários normais, dando uma soma 11111011. Novamente da Fig. 2-15 determinamos que a soma complemento 2 11111011 é igual a -5_{10}.

Some os decimais −2 e −5. O procedimento é detalhado na Fig. 2-17*d*. Da tabela da Fig. 2-15 obtemos que −2 = 11111110 (complemento 2) e −5 = 11111011 em notação complemento 2. Os números complemento 2 11111110 e 11111011 são somados como se fossem números binários regulares. A soma é 111111011. O MSB é um vai-um de sobra do registrador de 8 bits e é descartado. A soma dos números complemento 2 11111110 e 11111011 é, então, 11111001 (complemento 2). Usando a tabela da Fig. 2-15 determina-se que a soma complemento 2 11111001 é igual a -7_{10}.

Subtraia o número decimal com sinal +5 de +8. O procedimento é esboçado na Fig. 2-18*a*. O minuendo (+8) é igual a 00001000. O subtraendo é +5, ou seja, 00000101. O 00000101 deve ser convertido para a forma complemento 2 (complementar e somar 1) resultando 11111011. O "minuendo" 00001000 é somado ao complemento 2 do subtraendo 11111011 como se fossem números binários. A soma é 100000011. O MSB é um vai-um de sobra do registrador e é descartado deixando uma soma de 00000011. Da tabela da Fig. 2-15 determina-se que a soma complemento 2 00000011 é igual a $+3_{10}$. Note que, para subtrair, o subtraendo, é convertido para sua forma complemento 2 e somado ao minuendo. *Usando a representação complemento 2 e um somador, o microprocessador pode efetuar subtrações.*

Subtraia o maior número decimal de +6 de +2. O procedimento é mostrado na Fig. 2-28*b*. O minuendo +2 é igual a 00000010 (complemento 2). O subtraendo +6 é igual a 00000110. Este subtraendo é convertido em sua forma complemento 2 (complementar e somar 1) resultando 11111010. Os números complemento 2 00000010 e 11111010 são, então, somados como se fossem números binários, resultando uma soma 11111100. Da tabela da Fig. 2-15 determina-se que a soma complemento 2 11111100 é igual a 4_{10}.

$$\begin{array}{lrcr}
\text{Minuendo} & (+8) & & 00001000 \\
\text{Subtraendo} & -(+5) = 00000101 & \xrightarrow[\text{complemento 2}]{\text{Converter para}} & +\ 11111011 \\
\text{Diferença} & (+3) & & \underbrace{1}_{\text{Descartar a sobra}}\,00000011
\end{array}$$

(*a*) Problema de subtração em complemento 2 usando soma

$$\begin{array}{lrcr}
\text{Minuendo} & (+2) & & 00000010 \\
\text{Subtraendo} & -(+6) = 00000110 & \xrightarrow[\text{complemento 2}]{\text{Converter para}} & +\ 11111010 \\
\text{Diferença} & (-4) & & 11111100
\end{array}$$

(*b*) Problema de subtração em complemento 2 usando soma

Fig. 2-18

Problemas Resolvidos

2.29 Some os seguintes decimais com sinal usando números complemento 2:

(*a*) $\begin{array}{r} (+8) \\ +(-5) \\ \hline \end{array}$ (*b*) $\begin{array}{r} (+89) \\ +(-46) \\ \hline \end{array}$

Solução:

Siga o procedimento mostrado na Fig. 2-17*b*. Os resultados das somas são os seguintes:

$$(a)\quad \begin{array}{r} (+8) \\ +(-5) \\ \hline (+3) \end{array} \; = \; \begin{array}{r} 1\,000 \\ +\ 11111011 \\ \hline 1\ 00000011 \end{array} \quad \text{(complemento 2)}$$

$$(b)\quad \begin{array}{r} (+89) \\ +(-46) \\ \hline (+43) \end{array} \; = \; \begin{array}{r} 01011001 \\ +\ 11010010 \\ \hline 1\ 00101011 \end{array} \quad \text{(complemento 2)}$$

descartar a sobra

descartar a sobra

2.30 Some os seguintes decimais com sinal usando números complemento 2:

$$(a)\quad \begin{array}{r} (+1) \\ \hline +(-6) \end{array} \qquad (b)\quad \begin{array}{r} (+20) \\ \hline +(-60) \end{array}$$

Solução:

Siga o procedimento mostrado na Fig. 2-17*c*. Os resultados das somas são os seguintes:

$$(a)\quad \begin{array}{r} (+1) \\ +(-6) \\ \hline (-5) \end{array} = \begin{array}{r} 00000001 \\ +\,11111010 \\ \hline 11111011 \end{array} \quad \text{(complemento 2)}$$

$$(b)\quad \begin{array}{r} (+20) \\ +(-60) \\ \hline (-40) \end{array} = \begin{array}{r} 00010100 \\ +\,11000100 \\ \hline 11011000 \end{array} \quad \text{(complemento 2)}$$

2.31 Subtraia os seguintes decimais com sinal usando números complemento 2:

$$(a)\quad \begin{array}{r} (+7) \\ \hline -(+2) \end{array} \qquad (b)\quad \begin{array}{r} (+113) \\ \hline -\ (+50) \end{array}$$

Solução:

Siga o procedimento mostrado na Fig. 2-18*a*. Os resultados dos problemas de subtração são os seguintes:

$$(a)\quad \begin{array}{r} (+7) \\ -(+2) \\ \hline (+5) \end{array} = 00000010 \xrightarrow[\text{e soma}]{\text{complemento 2}} \begin{array}{r} 00000111 \\ +\,11111110 \\ \hline 1\ 00000101 \end{array} \quad \text{(complemento 2)}$$

sobra descartada

$$(b)\quad \begin{array}{r} (+113) \\ -\ (+50) \\ \hline (+63) \end{array} = 00110010 \xrightarrow[\text{e soma}]{\text{complemento 2}} \begin{array}{r} 01110001 \\ +\,11001110 \\ \hline 1\ 00111111 \end{array} \quad \text{(complemento 2)}$$

sobra descartada

2.32 Subtraia os seguintes decimais com sinal usando números complemento 2:

$$(a)\quad \begin{array}{r} (+3) \\ \underline{-(+8)} \end{array} \qquad (b)\quad \begin{array}{r} (+12) \\ \underline{-(+63)} \end{array}$$

Solução:

Siga o procedimento mostrado na Fig. 2-18*b*. Os resultados dos problemas de subtração são os seguintes:

$$(a)\quad \begin{array}{r} (+3) \\ \underline{-(+8)} = 00001000 \\ (-5) = \end{array} \xrightarrow[\text{e soma}]{\textbf{complemento 2}} \begin{array}{r} 00000011 \\ \underline{+\ 11111000} \\ 11111011 \end{array} \quad \text{(complemento 2)}$$

$$(b)\quad \begin{array}{r} (+12) \\ \underline{-(+63)} = 00111111 \\ (-51) = \end{array} \xrightarrow[\text{e soma}]{\textbf{complemento 2}} \begin{array}{r} 00001100 \\ \underline{+\ 11000001} \\ 11001101 \end{array} \quad \text{(complemento 2)}$$

2.8 AGRUPAMENTOS DE BITS

Um único dígito binário é chamado *bit*. Quatro bits agrupados são chamados *nibble*. Oito bits agrupados são chamados *byte*.

Uma característica muito importante de qualquer microprocessador é o tamanho do acumulador. Os microprocessadores usam, comumente, acumuladores de 8 bits. O *tamanho da palavra* do microprocessador é, então, de 8 bits. Neste caso, um byte perfaz uma palavra. Os microprocessadores têm palavras de 4, 8, 16 ou mesmo 32 bits. Um microprocessador de 16 bits tem uma palavra cujo tamanho é de 2 bytes ou 16 dígitos binários. Uma *palavra* é um grupo de bits que é processado como um único número ou instrução pelo microprocessador. Um microprocessador de 8 bits transfere e armazena todos os dados em grupos de 8 bits através de oito condutores em paralelo chamados barramento de dados .

O conteúdo da memória de um microprocessador de 8 bits poderia ser como o da Fig. 2-19*a*. Note que cada posição de endereço (denominada conteúdo) guarda um grupo de 8 bits de informação. Cada byte é chamado *palavra de memória*, pois o microprocessador é uma unidade de 8 bits. Cada palavra de memória tem um significado específico quando é buscada e decodificada pelo microprocessador. O conteúdo binário da memória da Fig. 2-19*a* poderia representar uma das seguintes quantidades:

1. um número binário
2. um número binário com sinal
3. um número BCD
4. um caracter (uma letra do alfabeto)
5. uma instrução
6. um endereço de memória
7. o endereço de uma porta de acesso de entrada ou saída

Consideremos a posição mais alta de memória (01100100_2) na Fig. 2-19*a*. O conteúdo desta posição de memória é 11011011. Esta palavra de memória poderia ser interpretada como:

1. um número binário: $11011011_2 = 219_{10}$
2. um número binário com sinal: 11011011 (complemento 2) = -37_{10}
3. um número BCD – não poderia ser um número decimal codificado em binário pois nem 1101 nem 1011 são códigos BCD
4. um caracter – não representa nenhum caracter ASCII (ASCII é um código alfanumérico especial muito popular)
5. uma instrução – 11011011 = a instrução INPUT do popular microprocessador 8080/8085 da Intel
6. um endereço de memória: $11011011_2 = DB_{16}$ = posição de memória 219_{10}
7. um endereço de uma porta de acesso de entrada ou saída – 11011011_2 = porta 219_{10}

Resumindo, a palavra mais alta da memória da Fig. 2-19*a* poderia ser o número binário representando 219_{10}, o número binário com sinal representando -37_{10}, a instrução INPUT do microprocessador 8085, o endereço da posição de memória DB_{16}, ou o endereço da porta de acesso de entrada ou saída 219_{10}. Neste exemplo a palavra mais alta da memória (11011011) poderia representar qualquer uma de cinco diferentes posições, instruções ou quantidades.

Se o operador do microprocessador 8085 inicializasse o contador de programa no endereço 100_{10} (01100100_2), o microprocessador buscaria e decodificaria a palavra de memória 11011011 como uma instrução INPUT de dados. O microprocessador, então, prosseguiria para o próximo endereço 101_{10} (01100101_2). O conteúdo da memória da Fig. 2-19*a* é o mesmo da memória da Fig. 1-4. Lembre-se de que o programa mostrado na Fig. 1-4 executa as seguintes instruções:

1. INPUT dados da porta 1
2. STORE dados na posição de memória 200
3. OUTPUT dados para a porta 10

A interpretação pelo microprocessador do conteúdo da memória é detalhada na Fig. 2-19*b*. As instruções do programa são listadas nas seis posições mais altas de memória ($100–105_{10}$). A posição mais baixa de memória (200_{10}) é a posição de armazenamento de dados. Neste caso, o código ASCII da letra A é armazenado nesta posição de memória.

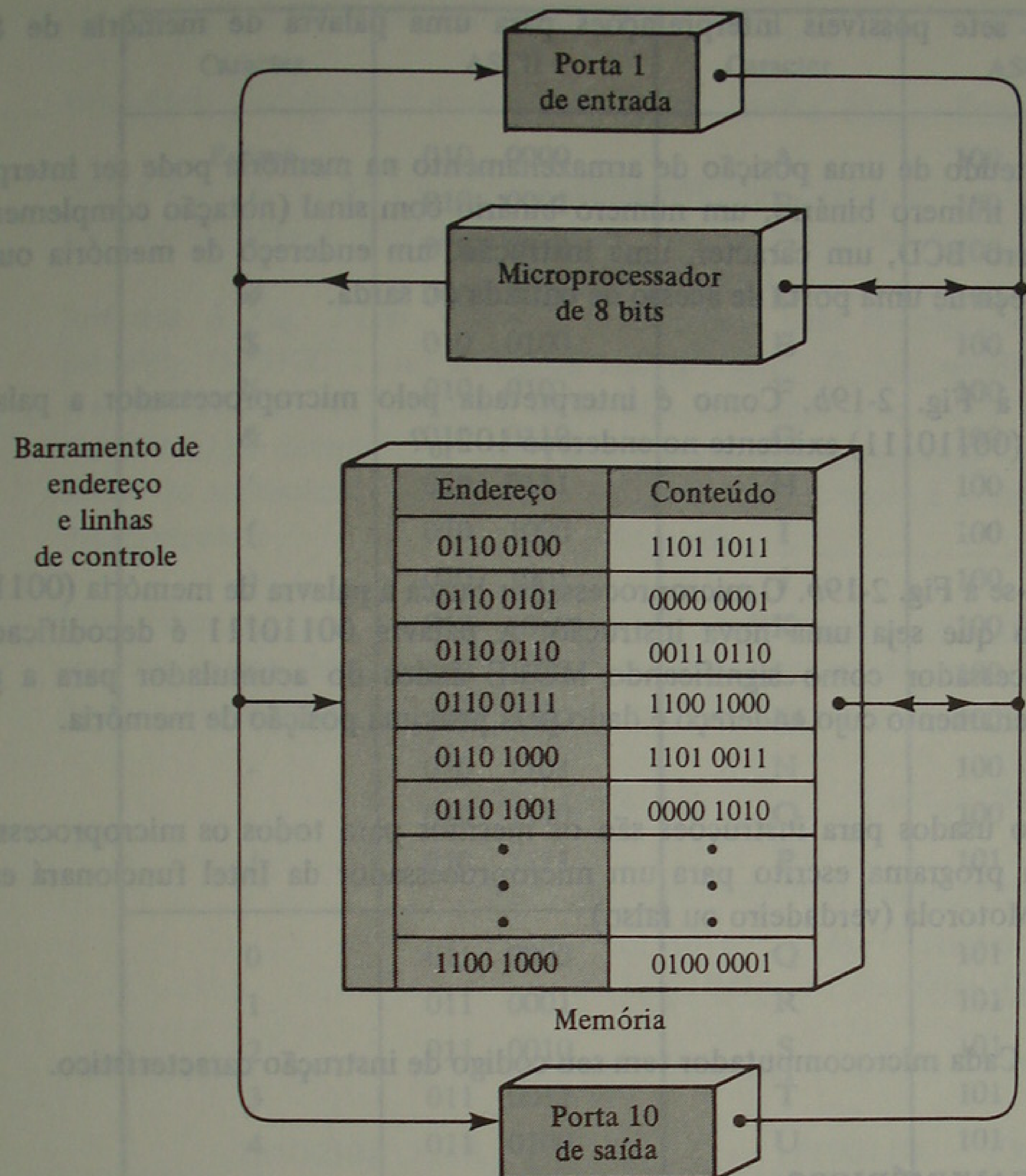
Resumindo, é importante notar que nos microprocessadores os bits são agrupados em palavras. Estas palavras da memória de programa são interpretadas pelo microprocessador em seqüência, *uma de cada vez*. É importante que o programador entenda como o microprocessador seqüencia e interpreta dados. Cada microprocessador tem seus próprios códigos de instrução, mas todos os microprocessadores seqüenciam as posições de memória de modo similar.

Problemas Resolvidos

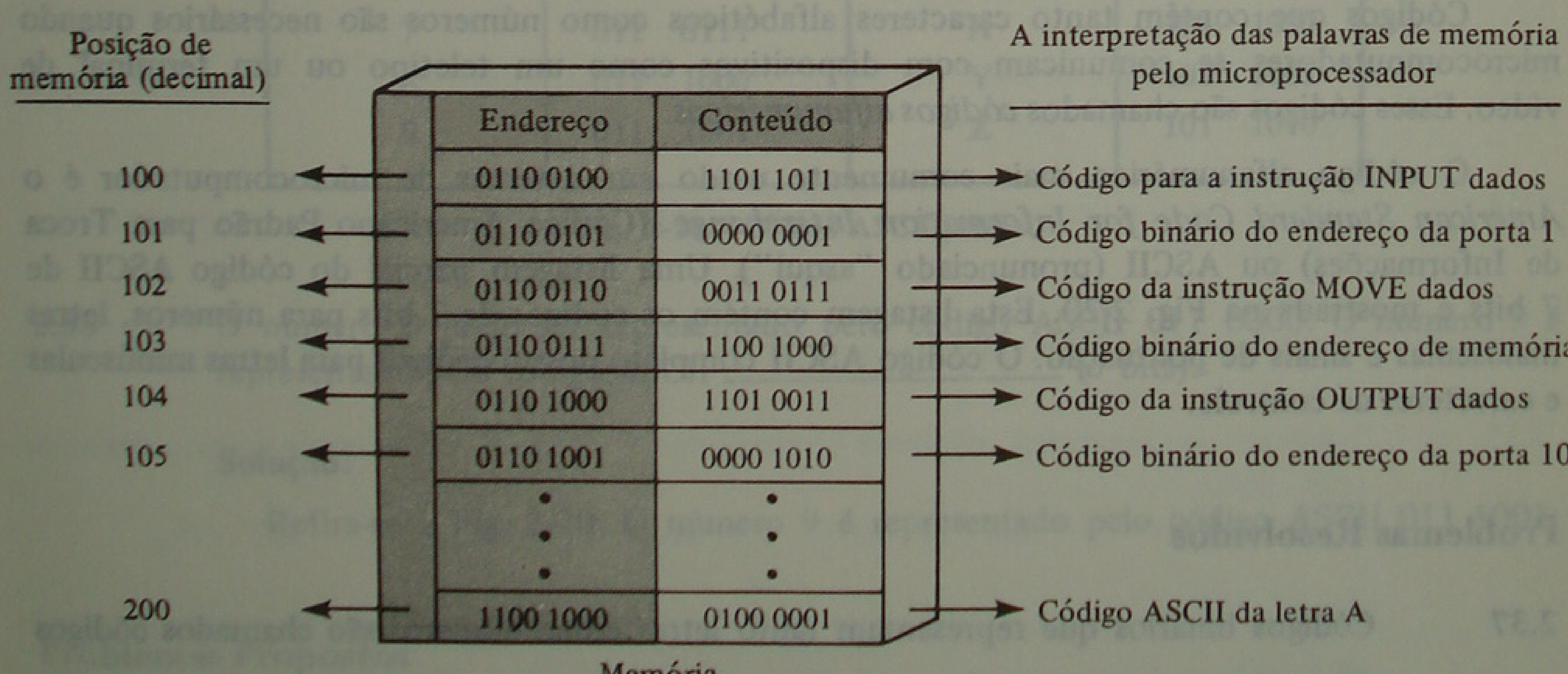
Quatro bits agrupados são chamados de nibble, enquanto um grupo de 8 bits é chamado de ____________________.

Solução:

Um grupamento de 8 bits é chamado byte. Em circunstâncias especiais, este grupo de 8 bits também pode ser chamado de palavra.



(*a*) Conteúdo binário de memória típico de um microcomputador



(*b*) A interpretação do conteúdo da memória pelo microprocessador

Fig. 2-19

2.34 Enumere sete possíveis interpretações para uma palavra de memória de 8 bits.

Solução:

O conteúdo de uma posição de armazenamento na memória pode ser interpretado como um número binário, um número binário com sinal (notação complemento 2), um número BCD, um caracter, uma instrução, um endereço de memória ou como um endereço de uma porta de acesso de entrada ou saída.

2.35 Refira-se à Fig. 2-19*b*. Como é interpretada pelo microprocessador a palavra de memória (00110111) existente no endereço 102_{10}?

Solução:

Refira-se à Fig. 2-19*b*. O microprocessador busca a palavra de memória (00110111), esperando que seja uma nova instrução. A palavra 00110111 é decodificada pelo microprocessador como significando MOVE dados do acumulador para a posição de armazenamento cujo endereço é dado pela próxima posição de memória.

2.36 Os código usados para instruções são os mesmos para todos os microprocessadores; logo, um programa escrito para um microprocessador da Intel funcionará em uma unidade Motorola (verdadeiro ou falso).

Solução:

Falso. Cada microcomputador tem seu código de instrução característico.

2.9 CÓDIGOS ALFANUMÉRICOS

Códigos que contêm tanto caracteres alfabéticos como números são necessários quando microcomputadores se comunicam com dispositivos como um teletipo ou um terminal de vídeo. Esses códigos são chamados *códigos alfanuméricos.*

O código alfanumérico mais comumente usado em sistemas de microcomputador é o *American Standard Code for Information Interchange* (Código Americano Padrão para Troca de Informações) ou ASCII (pronunciado "asqui"). Uma listagem parcial do código ASCII de 7 bits é mostrada na Fig. 2-20. Esta listagem contém os códigos de 7 bits para números, letras maiúsculas e sinais de pontuação. O código ASCII completo possui códigos para letras minúsculas e caracteres de controle.

Problemas Resolvidos

2.37 Códigos binários que representam tanto letras como números são chamados códigos ____________________ .

Solução:

Códigos alfanuméricos são usados para representar tanto letras como números.

Caracter	ASCII	Caracter	ASCII
Espaço	010 0000	A	100 0001
!	010 0001	B	100 0010
"	010 0010	C	100 0011
#	010 0011	D	100 0100
$	010 0100	E	100 0101
%	010 0101	F	100 0110
&	010 0110	G	100 0111
'	010 0111	H	100 1000
(	010 1000	I	100 1001
)	010 1001	J	100 1010
*	010 1010	K	100 1011
+	010 1011	L	100 1100
,	010 1100	M	100 1101
-	010 1101	N	100 1110
.	010 1110	O	100 1111
/	010 1111	P	101 0000
0	011 0000	Q	101 0001
1	011 0001	R	101 0010
2	011 0010	S	101 0011
3	011 0011	T	101 0100
4	011 0100	U	101 0101
5	011 0101	V	101 0110
6	011 0110	W	101 0111
7	011 0111	X	101 1000
8	011 1000	Y	101 1001
9	011 1001	Z	101 1010

Fig. 2-20 Código ASCII parcial

2.38 O número 0 pode ser representado pelo código ASCII 011 0000. O número 9 é representado pelo código ASCII ________ (7 bits).

Solução:

Refira-se à Fig. 2-20. O número 9 é representado pelo código ASCII 011 1001.

Problemas Propostos

2.39 De memória, converta os seguintes números binários em seus equivalentes decimais:
(*a*) 0000 (*b*) 0010 (*c*) 0011 (*d*) 0111 (*e*) 1001 (*f*) 1100
Resp.: (*a*) 0 (*b*) 2 (*c*) 3 (*d*) 7 (*e*) 9 (*f*) 12

2.40 $01101001_2 = \_\_\_\_\_\__{10}$ *Resp.:* 105_{10}

2.41 $8D_{16} = \_\_\_\_\_\__2$ *Resp.:* 10001101_2

2.42 $01011111_2 = \_\_\_\_\_\__{16}$ *Resp.:* $5F_{16}$

2.43 $AE_{16} = \_\_\_\_\_\__2$ *Resp.:* 10101110_2

2.44 $10111100_2 = \_\_\_\_\_\__{16}$ *Resp.:* BC_{16}

2.45 $3C_{16} = \_\_\_\_\_\__{10}$ *Resp.:* 60_{10}

2.46 $57_8 = \_\_\_\_\_\__{10}$ *Resp.:* 47_{10}

2.47 $63_{10} = \_\_\_\_\_\__8$ *Resp.:* 77_8

2.48 $92_{10} = \_\_\_\_\_\__{BCD}$ *Resp.:* $1001\ 0010_{BCD}$

2.49 $1000\ 0110_{BCD} = \_\_\_\_\_\__{10}$ *Resp.:* 86_{10}

2.50 Resolva os seguintes problemas de soma binária:

(*a*) $\begin{array}{r} 11000011 \\ +\ 00111100 \\ \hline \end{array}$ (*b*) $\begin{array}{r} 01101110 \\ +\ 00111101 \\ \hline \end{array}$

Resp.: (*a*) 11111111_2 (*b*) 10101011_2

2.51 $11011000_2 - 00110011_2 = \_\_\_\_\_\__2$ *Resp.:* 10100101_2

2.52 $1001_2 \times 1101_2 = \_\_\_\_\_\__2$ *Resp.:* 1110101_2

2.53 A notação complemento 2 01111110 representa um número ________ (positivo, negativo). *Resp.:* positivo

2.54 Traduza os seguintes números decimais com sinal para sua forma complemento 2 de 8 bits:
(*a*) +12 (*b*) −12
Resp.: (*a*) 00001100 (complemento 2) (*b*) 11110100 (complemento 2)

2.55 Traduza os seguintes números complemento 2 para seus equivalentes decimais com sinal:
(*a*) 01110100 (*b*) 11011101 *Resp.:* (*a*) $+116_{10}$ (*b*) -35_{10}

2.56 Subtraia os seguintes números decimais com sinal usando números complemento 2 de 8 bits:

(*a*) $\begin{array}{r} (+13) \\ -\ (+5) \\ \hline \end{array}$ (*b*) $\begin{array}{r} (+19) \\ -\ (+29) \\ \hline \end{array}$

Resp.: (*a*) 00001000 (complemento 2) (*b*) 11110110 (complemento 2)

2.57 Um byte é um grupo que contém ______________ bits. *Resp.:* 8

2.58 Um nibble é um grupo que contém ______________ bits. *Resp.:* 4

2.59 Refira-se à Fig. 2-19*a*. Este é o diagrama de blocos básico de um ______________ (microcomputador, microprocessador). *Resp.:* microcomputador

2.60 Refira-se à Fig. 2-19*b*. Como é interpretada pelo microprocessador a palavra de memória (11010011) no endereço 104_{10}? *Resp.:* O microprocessador busca a palavra de memória (11010011) esperando que seja uma nova instrução. A palavra (11010011) é decodificada pelo microprocessador significando que ele deve transferir dados do acumulador para a porta de saída, cujo número é dado pela posição de memória seguinte.

2.61 Um código ______________________ provavelmente seria usado para traduzir um dispositivo de entrada do tipo teclado para um sistema de microcomputador. *Resp.:* alfanumérico ou ASCII