Números reais com partes inteiras muito grandes ou fracionárias muito pequenas não devem ser armazenados na representação ponto fixo.

Representação ponto flutuante A solução para manter a exatidão ou a precisão é utilizar a **representação ponto flutuante**. Essa representação permite que o ponto decimal *flutue*: podemos ter diferentes quantidades de dígitos à direita ou à esquerda do ponto decimal. O intervalo de números reais que pode ser armazenado utilizando-se esse método aumenta enormemente: números com grandes partes inteiras ou pequenas fracionárias podem ser armazenados na memória. Na representação ponto flutuante, um número, seja decimal ou binário, é composto de três seções, como mostra a Figura 3.10.

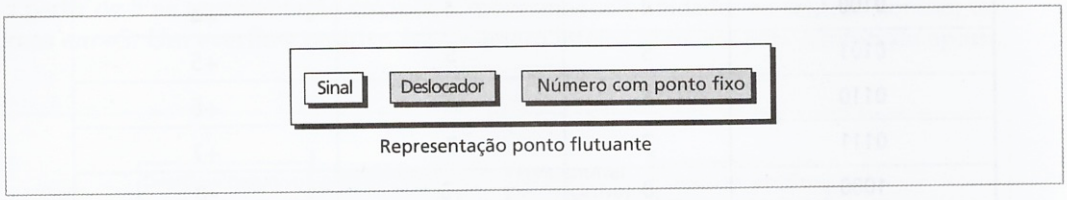

Figura 3.10 As três partes de um número real na representação ponto flutuante

A primeira parte é o sinal, positivo ou negativo. A segunda mostra quantas posições o ponto decimal deve mudar para a direita ou esquerda, para formar o número propriamente dito. A terceira é uma representação ponto fixo, em que a posição do ponto decimal é fixa.

Uma representação ponto flutuante de um número é composta de três partes: um sinal, um deslocador e um número com ponto fixo.

A representação ponto flutuante é utilizada nas ciências para representar números decimais muito pequenos ou muito grandes. Nessa representação, que também é chamada de *notação científica*, a parte com ponto fixo tem somente um dígito à esquerda do ponto decimal, e o deslocador é uma potência de base 10.

Exemplo 3.18

A seguir, veja o número decimal 7.425.000.000.000.000.000.000,00 em notação científica (representação ponto flutuante).

Solução

Número verdadeiro	→	+	7.425.000.000.000.000.000.000,00
Notação científica	→	+	$7{,}425 \times 10^{21}$

As três seções são o sinal (+), o deslocador (21) e a parte com ponto fixo (7,425). Observe que o deslocador é o expoente. Podemos, facilmente, perceber a vantagem disso. Mesmo se apenas quisermos escrever o número em um pedaço de papel, a notação científica é mais curta e ocupa menos espaço. A notação utiliza o conceito de ponto flutuante, porque a posição do ponto decimal, que está perto do lado direito, no exemplo, moveu 21 dígitos para a esquerda para formar a parte de ponto fixo do número. Algumas linguagens de programação e calculadoras mostram o número como +7,425E21, porque a base 10 é compreendida e não precisa ser mencionada.

Exemplo 3.19

Mostre o número –0,0000000000000232 em notação científica.

Solução

Utilizamos a mesma abordagem do exemplo anterior – movemos o ponto decimal depois do dígito 2, como é mostrado a seguir:

Número verdadeiro	→	–	0,0000000000000232
Notação científica	→	–	$2{,}32 \times 10^{-14}$

Note que o expoente, aqui, é negativo, porque o ponto decimal em 2,32 precisa ser movido para a esquerda (14 posições) para formar o número original. Novamente, podemos dizer que o número nessa notação é composto de três partes: sinal (–), o número real (2,32) e o inteiro negativo (14). Algumas linguagens de programação e calculadoras mostram isso como –2,32E–14.

Abordagens similares são utilizadas para representar números muito grandes ou muito pequenos (tanto inteiros como reais) em binário, para ser armazenados em computadores.

Exemplo 3.20

Mostre o número $(10100100000000000000000000000000,00)_2$ no formato ponto flutuante.

Solução

Utilizamos a mesma ideia, mantendo somente um dígito à esquerda do número.

Número verdadeiro	→	+	$(10100100000000000000000000000000,00)_2$
Notação científica	→	+	$1{,}01001 \times 2^{32}$

Observe que não temos de nos preocupar com todos os 0s à direita do 1 que está mais à direita, porque eles não são significativos quando utilizamos o número real $(1{,}01001)_2$. O expoente é mostrado como 32, mas, na verdade, é armazenado no computador em binário, como veremos brevemente. Também mostramos o sinal como positivo, mas ele será armazenado como um bit.

Exemplo 3.21

Mostre o número $-(0{,}00000000000000000000000101)_2$ no formato ponto flutuante.

Solução

Utilizamos a mesma ideia, mantendo somente um dígito diferente de zero à esquerda do ponto decimal.

Número verdadeiro	→	–	$(0{,}00000000000000000000000101)_2$
Notação científica	→	–	$1{,}01 \times 2^{-24}$

Note que o expoente é armazenado como um binário negativo no computador.

Normalização Para tornar uniforme a parte fixa da representação, os métodos científico (para o sistema decimal) e de ponto flutuante (para o sistema binário) utilizam somente um dígito diferente de zero à esquerda do ponto decimal. Isso é chamado de **normalização**. No sistema decimal, esse dígito pode ser de 1 a 9, ao passo que no binário somente 1. Veja, a seguir, que d é um dígito diferente de zero, x é um dígito e y é 0 ou 1.

Decimal	→	±	d,xxxxxxxxxxxxx	Note: *d* é de 1 a 9 e cada *x* é de 0 a 9
Binário	→	±	1,yyyyyyyyyyyyy	Note: cada *y* é 0 ou 1

Sinal, expoente e mantissa Depois que um número binário é normalizado, somente três partes da informação sobre o número são armazenadas: sinal, expoente e mantissa (os bits à direita do ponto decimal). Por exemplo, +1000111,0101 se torna:

+	2^6	×	1,0001110101
+	6		0001110101
↑	↑		↑
Sinal	Expoente		Mantissa

Note que o ponto e o bit 1 à esquerda da seção de ponto fixo não são armazenados – eles estão implícitos.

Sinal

O sinal do número pode ser armazenado utilizando 1 bit (0 ou 1).

Expoente

O expoente (potência de 2) define o deslocamento do ponto decimal. Observe que a potência pode ser negativa ou positiva. A **representação Excesso** (discutida posteriormente) é o método utilizado para armazenar o expoente.

Mantissa

Mantissa é o número inteiro binário à direita do ponto decimal. Ela define a precisão do número e é armazenada em notação ponto fixo. Se considerarmos a mantissa e o sinal juntos, podemos dizer que essa combinação é armazenada como um número inteiro no formato sinal-magnitude. Contudo, precisamos lembrar que essa não é um número inteiro, mas a parte fracionária que é armazenada como um número inteiro. Enfatizamos esse aspecto porque em uma mantissa, se inserirmos 0s extras à *direita* do número, o valor não será modificado, ao passo que em um número inteiro real, se inserirmos 0s extras à *esquerda* do número, o valor não será modificado.

Mantissa é uma parte fracionária que, juntamente com o sinal, é tratada como um número inteiro armazenado na representação com sinal-magnitude.

O sistema Excesso A mantissa pode ser armazenada como um número inteiro sem sinal. O expoente, a potência que mostra quantos bits o ponto decimal deverá ser movido para a esquerda ou a direita, é um número com sinal. Embora ele possa ser armazenado utilizando a representação complemento de dois, em vez disso, uma nova representação, chamada sistema Excesso, é utilizada. Nesse sistema, os números inteiros positivos e negativos são armazenados como números inteiros sem sinal. Para representar um número inteiro positivo ou negativo, um número inteiro positivo (chamado *bias*) é adicionado a cada número para deslocá-los uniformemente para o lado não negativo. O valor desse bias é $2^{m-1}-1$, onde m é o tamanho da localização de memória para armazenar o expoente.

Exemplo 3.22

Podemos expressar 16 números inteiros em um sistema numérico com alocação de 4 bits. Utilizando uma localização para 0 e dividindo os outros 15 (não totalmente iguais), podemos expressar números inteiros no intervalo de –7 a 8, como mostra a Figura 3.11. Adicionando sete unidades a cada número inteiro nesse intervalo, podemos transladar uniformemente todos os números inteiros à direita e tornar todos eles positivos, sem modificar a posição relativa dos números inteiros, um em relação ao outro, como mostra a figura. O novo sistema é chamado *Excesso_7*, ou representação com bias, com valor de bias igual a 7.

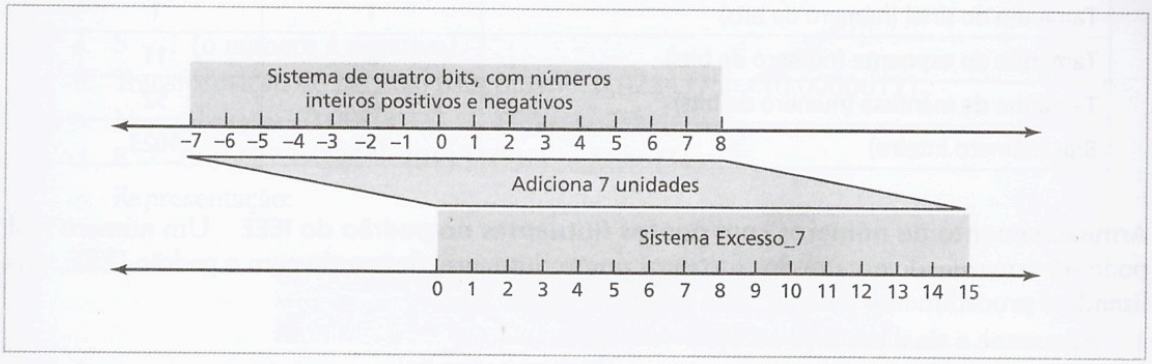

Figura 3.11 Deslocamento na representação Excesso

A vantagem dessa nova representação, em comparação com aquela antes da translação, é que todos os números inteiros no sistema Excesso são positivos, por isso, não há necessidade de nos preocuparmos quanto ao sinal quando compararmos ou realizarmos operações com os números inteiros. Para a alocação de quatro bits, o bias é $2^{4-1} - 1 = 7$, como era de se esperar.

Padrões do IEEE O IEEE (Institute of Electrical and Electronics Engineers [Instituto dos Engenheiros Eletricistas e Eletrônicos]) definiu diversos padrões para o armazenamento de números pontos flutuantes. Discutimos aqui os dois mais comuns, de precisão simples e de precisão dupla. Esses formatos são mostrados na Figura 3.12. Os números sobre as caixas são números de bits para os campos correspondentes.

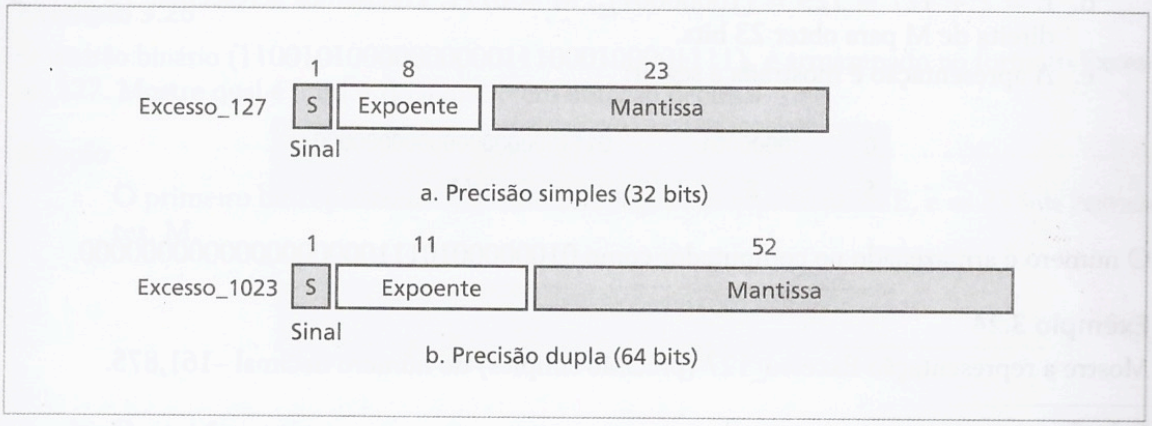

Figura 3.12 Padrões IEEE para a representação ponto flutuante

O formato com precisão simples utiliza um total de 32 bits para armazenar um número real na representação ponto flutuante. O sinal ocupa um bit (0 para positivo e 1 para negativo); o expoente, oito bits (utilizando um bias de 127); e a mantissa, 23 bits (número sem sinal). Esse padrão, algumas vezes, é chamado **Excesso_127**, porque o bias é 127.

O formato com precisão dupla utiliza um total de 64 bits para armazenar um número real na representação ponto flutuante. O sinal ocupa 1 bit; o expoente, 11 bits (utilizando um bias de 1023); e a mantissa, 52 bits. Algumas vezes, o padrão é chamado **Excesso_1023**, porque o bias é 1023. A Tabela 3.2 resume a especificação dos dois padrões.

Tabela 3.2 Especificações dos dois padrões IEEE para ponto flutuante

Parâmetro	*Precisão simples*	*Precisão dupla*
Tamanho da localização de memória (número de bits)	32	64
Tamanho do sinal (número de bits)	1	1
Tamanho do expoente (número de bits)	8	11
Tamanho da mantissa (número de bits)	23	52
Bias (número inteiro)	127	1023

Armazenamento de números com pontos flutuantes no padrão do IEEE Um número real pode ser armazenado em um dos formatos ponto flutuante, de acordo com o padrão IEEE, utilizando o procedimento a seguir, com relação à Figura 3.12:

1. Armazene o sinal em S (0 ou 1).
2. Mude o número para binário.
3. Normalize.
4. Encontre os valores de E e M.
5. Concatene S, E e M.

Exemplo 3.23

Mostre a representação Excesso_127 (precisão simples) do número decimal 5,75.

Solução

a. O sinal é positivo, de modo que $S = 0$.
b. Transformação de decimal para binário: $5{,}75 = (101{,}11)_2$.
c. Normalização: $(101{,}11)_2 = (1{,}0111)_2 \times 2^2$.
d. $E = 2 + 127 = 129 = (10000001)_2$, $M = 0111$. Precisamos acrescentar 19 zeros à direita de M para obter 23 bits.
e. A apresentação é mostrada a seguir:

0	10000001	01110000000000000000
S	E	M

O número é armazenado no computador como 01000000101110000000000000000000.

Exemplo 3.24

Mostre a representação Excesso_127 (precisão simples) do número decimal –161,875.

Solução

a. O sinal é negativo, de modo que $S = 1$.
b Transformação de decimal para binário: $161{,}875 = (10100001{,}111)_2$.
c. Normalização: $(10100001{,}111)_2 = (1{,}0100001111)_2 \times 2^7$.
d. $E = 7 + 127 = 134 = (10000110)_2$ e $M = (0100001111)_2$.
e. Representação:

1	10000110	01000011110000000000000
S	E	M

O número é armazenado no computador como 11000011010000111100000000000000.

Exemplo 3.25

Mostre a representação Excesso_127 (precisão simples) do número decimal –0,0234375.

Solução

a. S = 1 (o número é negativo).
b. Transformação de decimal para binário: $0{,}0234375 = (0{,}0000011)_2$.
c. Normalização: $(0{,}0000011)_2 = (1{,}1)_2 \times 2^{-6}$.
d. $E = -6 + 127 = 121 = (01111001)_2$ e $M = (1)_2$.
e. Representação:

1	01111001	10000000000000000000000
S	E	M

O número é armazenado no computador como 10111100110000000000000000000000.

Recuperando números armazenados no formato ponto flutuante, de acordo com o padrão IEEE Um número armazenado em um dos formatos ponto flutuante do padrão IEEE pode ser recuperado utilizando o método a seguir:

1. Encontre o valor de S, E e M.
2. Se S = 0, defina o sinal para positivo; do contrário, defina o sinal para negativo.
3. Encontre o deslocador (E – 127).
4. Desnormalize a mantissa.
5. Modifique o número desnormalizado para binário, a fim de encontrar o valor absoluto.
6. Acrescente o sinal.

Exemplo 3.26

O padrão binário $(11001010000000000111000100001111)_2$ é armazenado no formato Excesso_127. Mostre qual é o valor do número em notação decimal.

Solução

a. O primeiro bit representa S, os oito bits seguintes representam E, e os 23 bits restantes, M.

1	10010100	00000000111000100001111
S	E	M

b. O sinal é negativo.
c. O deslocador = E – 127 = 148 – 127 = 21.
d. A desnormalização nos dá $(1{,}00000000111000100001111)_2 \times 2^{21}$.
e. O número binário é $(100000000111000100011{,}11)_2$.
f. O valor absoluto é 2.104.378,75.
g. O número é –2.104.378,75.

3.10 CONJUNTO DE PRÁTICAS

Questões para revisão

1. Cite cinco tipos de dados que um computador pode processar.
2. Qual é o comprimento do padrão binário relacionado ao número de símbolos que o padrão binário pode representar?
3. Como o método gráfico bitmap representa uma imagem como um padrão binário?
4. Qual é a vantagem do método gráfico vetorial em relação ao bitmap? Qual é a desvantagem?
5. Quais etapas são necessárias para converter dados de áudio para padrões binários?
6. Compare e faça a distinção de números inteiros positivos nos formatos sem sinal, sinal-magnitude e complemento de dois.
7. Compare e faça a distinção da representação de números inteiros negativos nos formatos sinal-magnitude e complemento de dois.
8. Compare e faça a distinção da representação de zero nos formatos sinal-magnitude, complemento de dois e com Excesso.
9. Comente o papel do bit mais à esquerda nos formatos sinal-magnitude e complemento de dois.
10. Responda às seguintes perguntas sobre representações ponto flutuante de números reais:
    a. Por que a normalização é necessária?
    b. Qual é a mantissa?
    c. Depois que um número é normalizado, que tipo de informação faz que o computador o armazene na memória?

Questões de múltipla escolha

11. Um byte consiste de _____ bits.
    a. 2
    b. 4
    c. 8
    d. 16
12. Em um conjunto de 64 símbolos, cada símbolo exige um comprimento de padrão binário igual a _____ bits.
    a. 4
    b. 5
    c. 6
    d. 7
13. Quantos símbolos podem ser representados por um padrão binário com dez bits?
    a. 128
    b. 256
    c. 512
    d. 1024
14. Se o código ASCII para 'E' for 1000101, então, este código para 'e' será _____. Preencha a lacuna sem consultar a tabela de código ASCII.
    a. 1000110
    b. 1000111
    c. 0000110
    d. 1100101
15. Um código de 32 bits chamado _____ representa símbolos em todas as linguagens.
    a. ANSI
    b. Unicode
    c. EBCDIC
    d. ASCII expandido
16. Uma imagem pode ser representada no computador utilizando o método _____.
    a. Gráfico bitmap
    b. Gráfico vetorial
    c. Sistema Excesso
    d. (Alternativas a ou b)
17. No método gráfico _____, para representar uma imagem no computador, cada pixel é atribuído a um ou mais padrões binários.
    a. Bitmap
    b. Vetorial
    c. Quantizado
    d. Binário
18. No método gráfico _____, para representar uma imagem no computador, a imagem é decomposta em uma combinação de figuras geométricas.
    a. Bitmap
    b. Vetorial
    c. Quantizado
    d. Binário
19. No método gráfico _____, para representar uma imagem nó computador, o rescalonamento da imagem cria uma imagem irregular ou granulada.
    a. Bitmap
    b. Vetorial
    c. Quantizado
    d. Binário
20. Quando queremos armazenar música em um computador, o sinal de áudio deve ser _____.
    a. Amostrado
    b. Quantizado
    c. Codificado
    d. (Todas as anteriores)
21. Se o bit mais à esquerda na representação numérica _____ for zero, o número decimal é não negativo.
    a. Complemento de dois
    b. Ponto flutuante
    c. Sistema Excesso
    d. (Alternativas a ou b)
22. Se o bit mais à esquerda na representação numérica _____ for 1, o número decimal é negativo.
    a. Complemento de dois

b. Ponto flutuante
c. Sistema Excesso
d. (Alternativas a e b)

23. Qual método de representação numérica geralmente é utilizado para armazenar o valor exponencial de uma parte fracionária?
a. Números inteiros sem sinal
b. Complemento de dois
c. Sistema Excesso
d. (Nenhuma das anteriores)

24. Em uma conversão com Excesso, temos que _____ o número que corresponde ao bias ao número a ser convertido.
a. Adicionar
b. Subtrair
c. Multiplicar
d. Dividir

25. Quando uma parte fracionária é normalizada, o computador armazena o(a) _____.
a. Sinal
b. Expoente
c. Mantissa
d. (Todas as anteriores)

26. A precisão da parte fracionária de um número armazenado em um computador é definida pelo(a) _____.
a. Sinal
b. Expoente
c. Mantissa
d. (Qualquer uma das anteriores)

27. A combinação entre sinal e mantissa de um número real no formato ponto flutuante, de acordo com o padrão IEEE, é armazenada como um número inteiro na representação _____.
a. Sem sinal
b. Sinal-magnitude
c. Complemento de dois
d. (Nenhuma das anteriores)

Exercícios

28. Quantos diferentes padrões de cinco bits podemos ter?
29. Em alguns países, as placas de licença de veículos têm dois dígitos decimais (de 0 a 9). Quantas diferentes placas podemos ter? Se o dígito 0 não for permitido na placa de licença, quantas diferentes placas teremos?
30. Refaça o Exercício 29 considerando uma placa de licença com dois dígitos seguidos por três letras maiúsculas (de A a Z).
31. Uma máquina tem oito diferentes ciclos. Quantos bits são necessários para representar cada ciclo?
32. As notas de um aluno em um curso podem ser A, B, C, D, F, R (reprovado), ou I (incompleto). Quantos bits são necessários para representar as notas?
33. Uma empresa decidiu atribuir um único padrão binário a cada funcionário. Se a empresa tiver 900 funcionários, qual é o número mínimo de bits necessários para criar esse sistema de representação? Quantos padrões não foram atribuídos? Se a empresa contratar outros 300 funcionários, o número de bits deverá ser aumentado? Explique sua resposta.
34. Se utilizarmos um padrão de quatro bits para representar os dígitos de 0 a 9, quantos padrões serão utilizados?
35. Um sinal de áudio é amostrado 8.000 vezes por segundo. Cada amostra é representada por 256 diferentes níveis. Quantos bits por segundo são necessários para representar este sinal?
36. Transforme os seguintes números decimais em números inteiros sem sinal, com oito bits.
a. 23
b. 121
c. 34
d. 342
37. Transforme os seguintes números decimais em números inteiros sem sinal, com 16 bits.
a. 41
b. 411
c. 1.234
d. 342
38. Transforme os seguintes números decimais em números inteiros, com complemento de dois, com oito bits.
a. –12
b. –145
c. 56
d. 142
39. Transforme os seguintes números decimais em números inteiros, com complemento de dois, com 16 bits.
a. 102
b. –179
c. 534
d. 62.056
40. Transforme os seguintes números sem sinal de oito bits em decimais.
a. 01101011
b. 10010100
c. 00000110
d. 01010000
41. Transforme os seguintes números de oito bits, com complemento de dois, em decimais.
a. 01110111
b. 11111100
c. 01110100
d. 11001110
42. Os números seguintes são binários com complemento de dois. Mostre como mudar o sinal do número.

a. 01110111
b. 11111100
c. 01110111
d. 11001110

43. Se aplicarmos duas vezes a operação do complemento de dois em um número, deveremos obter o número original. Aplique essa operação a cada um dos seguintes números e verifique se é possível obter o número original.
a. 01110111
b. 11111100
c. 01110100
d. 11001110

44. Normalize os seguintes números binários com ponto flutuante. Mostre explicitamente o valor do expoente depois da normalização.
a. 1,10001
b. $2^3 \times 111,1111$
c. $2^{-2} \times 101,110011$
d. $2^{-5} \times 101101,00000110011000$

45. Converta os números a seguir no formato de 32 bits, de acordo com o padrão IEEE.
a. $-2^0 \times 1,10001$
b. $+2^3 \times 1,111111$
c. $+2^{-4} \times 1,01110011$
d. $-2^{-5} \times 1,01101000$

46. Converta os números a seguir no formato de 64 bits, de acordo com o padrão IEEE:
a. $-2^0 \times 1,10001$
b. $+2^3 \times 1,111111$
c. $+2^{-4} \times 1,01110011$
d. $-2^{-5} \times 1,01101000$

47. Converta os números a seguir no formato de 32 bits, de acordo com o padrão IEEE.
a. 7,1875
b. –12,640625
c. 11,40625
d. –0,375

48. Os números binários a seguir são representações sinal-magnitude em uma alocação de oito bits. Converta-os para decimais.
a. 01110111
b. 11111100
c. 01110100
d. 11001110

49. Converta os seguintes números inteiros decimais para uma representação sinal-magnitude com alocação de oito bits.
a. 53
b. –107
c. –5
d. 154

50. Um método para representar números com sinal em um computador é o com complemento de um, no qual, para representar um número positivo, armazenamos o número binário. Para representar um número negativo, aplicamos a operação do complemento de um ao número. Armazene os seguintes números inteiros decimais no formato complemento de um, com alocação de oito bits.
a. 53
b. –107
c. –5
d. 154

51. Os números a seguir são binários, na representação complemento de um, em uma alocação de oito bits. Converta-os em decimais (veja o Exercício 50).
a. 01110111
b. 11111100
c. 01110100
d. 11001110

52. Se aplicarmos duas vezes a operação complemento de um a um número, deveremos obter o número original. Aplique duas vezes a operação complemento de um a cada um dos seguintes números, e veja se é possível obter o número original (veja o Exercício 50).
a. 01110111
b. 11111100
c. 01110100
d. 11001110

53. Um método alternativo para encontrar o complemento de dois de um número é, primeiro, obter o complemento de um do número (veja o Exercício 50) e então acrescentar 1 ao resultado. (A adição de números inteiros binários será explicada no Capítulo 4). Experimente ambos os métodos utilizando os seguintes números. Compare e contraste os resultados.
a. 01110111
b. 11111100
c. 01110100
d. 11001110

54. O equivalente do complemento de um (veja o Exercício 50) no sistema binário é o complemento de nove no sistema decimal ($1 = 2 - 1$ e $9 = 10 - 1$). Com a alocação de n dígitos, podemos representar os números do complemento de nove, no intervalo de:

$$-[(10^n/2) - 1] \quad \text{a} \quad +[(10^n/2 - 1)]$$

O complemento de nove de um número com alocação de n dígitos é assim obtido: se o número for positivo, o complemento de nove do número é ele mesmo. Se for negativo, subtraímos cada dígito de 9. Responda às seguintes perguntas para a alocação de três dígitos:
a. Qual é o intervalo dos números que podemos representar utilizando o complemento de nove?
b. Nesse sistema, como podemos determinar o sinal de um número?
c. Temos dois zeros nesse sistema?

d. Se a resposta do item (c) for sim, qual é a representação para +0 e –0?

55. Assumindo a alocação de três dígitos, encontre o complemento de nove dos seguintes números decimais (veja o Exercício 54):
    a. +234
    b. +560
    c. –125
    d. –111

56. O equivalente do complemento de dois no sistema binário é o complemento de dez no sistema decimal (no sistema binário, 2 é a base; no decimal, 10). Utilizando a alocação de *n* dígitos, podemos representar os números no intervalo de:

$$-(10^n/2) \quad \text{a} \quad +(10^n/2-1)$$

no formato complemento de dez. Este complemento de um número com alocação de *n* dígitos é obtido, primeiro, encontrando o complemento de nove do número (como foi descrito no Exercício 54) e, então, adicionando 1 ao resultado. Responda às questões a seguir para a alocação de três dígitos.
    a. Qual é o intervalo dos números que podemos representar utilizando o complemento de dez?
    b. Nesse sistema, como podemos determinar o sinal de um número?
    c. Temos dois zeros nesse sistema?
    d. Se a resposta do item (c) for sim, qual é a representação para +0 e –0?

57. Assumindo a alocação de três dígitos, encontre o complemento de dez dos seguintes números decimais. (As informações para resolver este problema são apresentadas no Exercício 56).
    a. +234
    b. +560
    c. –125
    d. –111

58. O equivalente do complemento de um (Exercício 50) no sistema binário é o complemento de 15 no sistema hexadecimal (1 = 2 – 1 e 15 = 16 – 1). Leia a explicação apresentada para o Exercício 54 para responder às seguintes questões:
    a. Qual intervalo de números podemos representar com alocação de três dígitos no complemento de 15?
    b. Explique como o complemento de 15 de um número é obtido no sistema hexadecimal.
    c. Temos dois zeros nesse sistema?
    d. Se a resposta do item (c) for sim, qual é a representação para +0 e –0?

59. Assumindo a alocação de três dígitos, encontre o complemento de 15 dos seguintes números hexadecimais (veja o Exercício 58):
    a. +B14
    b. +FE1
    c. –1A
    d. –1E2

60. O equivalente do complemento de dois no sistema binário é o complemento de 16 no sistema hexadecimal. Leia a explicação apresentada para o Exercício 56 para responder às seguintes questões:
    a. Qual intervalo de números podemos representar com alocação de três dígitos no complemento de 16?
    b. Explique como um complemento de 16 de um número é obtido no sistema hexadecimal.
    c. Temos dois zeros nesse sistema?
    d. Se a resposta do item (c) for sim, qual é a representação para +0 e – 0?

61. Assumindo uma alocação de três dígitos, encontre o complemento de 16 dos seguintes números hexadecimais (veja o Exercício 60):
    a. +B14
    b. +FE1
    c. –1A
    d. –1E2